



**UNIVERSIDAD NACIONAL
AUTONOMA DE MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES
ACATLÁN**

**ANÁLISIS DEL SALTO VERTICAL Y PARABÓLICO EN EL
BÁSQUETBOL UNIVERSITARIO MEDIANTE DESARROLLO DE
SOFTWARE DE REGRESIÓN Y GRAFICACIÓN EN 3D.**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN MATEMÁTICAS APLICADAS Y COMPUTACIÓN**

PRESENTA

ARIEL AGUILAR GUILLEN

Asesor: Lic. Ernesto Baltazar Martínez

Fecha: Mayo de 2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Al Dios Jehová, por prestarme vida.

A mis padres y hermanos.

A todos mis compañeros y amigos de la carrera, especialmente a Víctor y Ramfis.

A todos los profesores que me dieron la inspiración para escribir este libro.

A Manuel, Yair, Richard, Pibe, Cobi y Pablo, excompañeros de la selección de básquetbol de Pumas Acatlán.

A la ciencia.

Quiero agradecer a los profesores que me dieron el conocimiento y la disciplina física y mental para iniciar y concluir esta tesis. En especial a los profesores: Maricarmen Villar, Jaime Ramírez, Sara Camacho, Angel Roldan, Reyes Laurencio, Ruth Cuellar, Araceli Álvarez, Javier Rosas, Georgina Eslava, Francisco Patiño, Manuel Junco. Al prof. de Ingles Héctor Leal. A los entrenamientos recibidos en la selección de básquetbol Pumas Acatlán que al mismo tiempo me permitieron analizar a fondo el tema del presente proyecto. A la disposición de las instalaciones del gimnasio de pesas para poder estudiar la respuesta biomecánica. A mi asesor por su compañerismo y accesibilidad durante todo este tiempo. A las universidades que me permitieron realizar las investigaciones y a la Facultad de Estudios Superiores Acatlán por haber creado esta Carrera.

A todos ellos: Muchísimas gracias.



 INDICE

Introducción	1
Objetivo.....	2
Hipótesis.....	2
1 Antecedentes.....	3
1.1 Antecedentes deportivos	4
1.2 Entrenamientos más comunes	7
1.3 Factores genéticos.....	12
1.3.1 El dominio de la raza afro.....	12
1.3.2 Diferencia entre los sexos.....	13
1.3.3 La prominencia de los brazos de palanca	14
1.4 Factores externos.....	15
1.4.1 Altitud y Latitud.....	15
1.4.2 El calzado deportivo.....	18
2 Tipos De Salto.....	19
2.1 Salto Vertical	20
2.1.1 Salto de altura olímpico	20
2.1.2 Salto de altura basquetbolístico.....	21
2.1.3 Salto vertical en suspensión.....	22
2.2 Salto parabólico.....	23
2.2.1 Salto parabólico en suspensión.....	25
2.2.2 Cálculo del salto parabólico.....	27
2.2.2.1 Definiciones.....	27
2.2.3 Cálculo de longitud máxima	35
2.2.4 Cálculo de altura máxima.....	36
2.2.5 Tiempo de vuelo	37
3 Biomecánica.....	39
3.1 Conceptos de biomecánica	40
3.1.1 Definiciones.....	41
3.1.1.1 Flexión y extensión.....	43
3.1.1.2 Brazos de palanca	44
3.1.2 Transmisión de fuerzas	46
3.2 Conceptos de Centro de Gravedad.....	47
3.2.1 Centro de gravedad en el cuerpo humano.....	48
3.2.1.1 Cálculo del centro de gravedad en una recta	48
3.2.1.2 Cálculo del centro de gravedad en un plano.....	49
3.2.1.3 Cálculo del centro de gravedad en el espacio.....	51

3.2.1.4	La plataforma equilátera.....	52
3.2.1.5	El método segmentario.....	56
3.3	Brazos de palanca en el salto vertical.....	58
3.3.1	Hiperextensión de la cabeza.....	58
3.3.2	Flexión de las piernas.....	59
3.3.3	Extensión de los brazos.....	59
3.3.4	Flexión del antebrazo.....	60
3.3.5	Elevación de los brazos.....	61
3.3.6	Extensión de la espalda alta.....	61
3.3.7	Extensión de la espalda media.....	62
3.3.8	Extensión de la pelvis.....	63
3.3.9	Flexión de la espalda baja.....	63
3.3.10	Extensión del muslo.....	64
3.3.11	Extensión de la pierna.....	65
3.3.12	Extensión del pie.....	65
3.3.13	Origen e Inserción.....	67
3.4	Generación de vectores de fuerza.....	68
3.5	Dificultad en los grados de inclinación.....	75
3.5.1	Posición incorrecta.....	79
3.6	Vectores en el salto parabólico.....	79
3.7	Fuerza explosiva.....	83
4	Regresión lineal y desarrollo de sistema para análisis de entrenamientos.....	85
4.1	Regresión lineal en los entrenamientos de salto.....	86
4.1.1	Mínimos cuadrados.....	87
4.1.2	Ajuste exponencial.....	96
4.1.3	Ajuste logarítmico.....	99
4.1.4	Ajuste cuadrático.....	102
4.1.5	Investigación de Campo.....	105
4.2	Desarrollo de sistema en Visual Basic y Crystal Reports.....	108
4.2.1	Diseño de tablas en Access.....	108
4.2.2	Desarrollo de reportes gráficos de regresión.....	111
4.2.2.1	Grafica/reporte de dispersión.....	114
4.2.2.2	Grafica/reporte de regresión.....	123
4.2.2.3	Grafica/reporte de ajuste exponencial.....	127
4.2.2.4	Grafica/reporte de ajuste logarítmico.....	129
4.2.2.5	Grafica/reporte de ajuste cuadrático.....	131
4.2.3	Desarrollo de sistema encuestador y grafico-reportador.....	134
4.2.3.1	Módulo de llenado de encuestas.....	135
4.2.3.2	Módulo de graficación de regresión y ajuste.....	138
4.2.3.3	Armado del sistema.....	138

5 Graficación del modelo humano en 3D.....	143
5.1 Transformaciones geométricas en 2D.....	144
5.1.1 Traslación 2D.....	144
5.1.2 Escalamiento 2D.....	145
5.1.3 Rotación 2D.....	145
5.2 Transformaciones geométricas en 3D.....	146
5.2.1 Traslación 3D.....	146
5.2.2 Escalamiento 3D.....	146
5.2.3 Rotación 3D.....	147
5.3 Diseño de un modelo humano.....	147
5.3.1 Medidas y pesos corporales.....	148
5.3.2 Diseño de segmentos del cuerpo en una cuadrícula.....	149
5.3.3 Ensamblado de Segmentos.....	152
5.4 Instalación del Lenguaje Visual C++ y OpenGL.....	153
5.5 Programación en OpenGL y Visual C++ del Centro de Gravedad.....	153
5.5.1 Rotación de los segmentos del modelo.....	156
5.5.2 Calculando el centro de gravedad.....	158
5.5.3 Codificación de matrices en Visual C++/OpenGL.....	161
5.5.4 Codificación de Interfaz de teclas de dirección.....	167
5.5.5 Teclas de dirección.....	168
5.5.6 Ejecución de escena "Cálculo de Centro de gravedad".....	169
5.6 Programación en OpenGL y VC++ de Salto Vertical/Parabólico.....	171
5.6.1 Codificación del salto vertical/parabólico.....	171
5.6.2 Transformación de objetos predefinidos.....	173
5.6.3 Codificación del tablero de básquetbol.....	174
5.6.4 Codificación de la malla.....	174
5.6.5 Codificación de datos iniciales.....	175
5.6.6 Teclas de dirección.....	176
5.6.7 Lecturas del salto vertical/parabólico.....	177
5.6.8 Ejecución de escena "Cálculo de Salto Vertical/Parabólico" ..	178
Resultados.....	179
Conclusiones.....	181
Bibliografía.....	182
Anexo A. Código Visual Basic (Sistema Encuestador Reporteador).....	A[1]
Anexo B. Código Visual C++/OpenGL (Programa del Centro de Gravedad).....	B[1]
Anexo C. Código Visual C++/OpenGL (Programa de Salto Vertical/Parabólico)....	C[1]
Anexo D. Sistema coordinado del cuerpo humano.....	D[1]

Introducción

Una vez dijo Albert Einstein que: "Dios no juega a los dados con el cosmos" que "se manifiesta en la armonía de lo que existe" [007], surge la idea de que el cosmos así como cada pequeño universo no es producto de un suceso azaroso sino mas bien determinístico o bien determinado por cierto orden. El cuerpo humano, está diseñado para permanecer atraído por la fuerza de gravedad de la tierra, donde cada vez que se ejecuta un salto se intenta escapar de dicha fuerza por unos instantes, para llegar a gravedad cero y finalmente volver al suelo. Como es de esperarse, hay posibilidad de estudiar el cuerpo humano para entender sus fenómenos, como lo que sucede al despegar del suelo por unos instantes y al que se le llama dependiendo de la trayectoria "salto vertical o salto parabólico".

Por siglos el hombre ha querido siempre superarse a sí mismo, correr más rápido, saltar más alto, romper su propio récord. De allí que se presente la situación de querer saltar a una mayor magnitud en un deporte como el basquetbol en donde en ocasiones se le ha llamado vuelo, que aunque no sea lo mismo volar y saltar, si despierta la imaginación de quienes observan a un buen atleta.

En el basquetbol es necesario tener un buen salto vertical y parabólico (mejor conocido como salto estacionario y salto en carrera) para las exigencias de este deporte. Es probable que alguna vez se haya planteado la pregunta de porqué el hombre no pueda saltar más alto en proporción a su tamaño a pesar de tener las piernas tan largas en comparación con muchos animales, los que de manera sobresaliente parecen estar mejor condicionados para correr o para saltar, aunque es verdad que mientras que para el ser humano sea un asunto mayormente deportivo, para muchos de aquellos animales sea un asunto de vida o muerte.

En la ciudad de México existen diversas instituciones deportivas donde se imparte esta disciplina. El básquetbol profesional tiene sus raíces en el básquetbol amateur y básquetbol universitario. Es en este último donde se presta atención a los entrenamientos que emergen de dichas instituciones educativas. Un problema consiste en que los jugadores de basquetbol universitario no saltan lo suficientemente alto como para ejecutar diversos movimientos requeridos por este deporte. Existe también la necesidad de compensar las diferencias de estaturas entre los jugadores de básquetbol. Mejorando su salto se podrá recuperar un mayor número de rebotes, mejorar el tiro en suspensión y entre otras cosas meter el balón directamente en la canasta (mejor conocida como clavada o volcada) sin importar mucho la estatura del atleta. Cada institución tiene sus propios entrenamientos, sin embargo estos no difieren mucho entre una y otra. También es notorio que el tipo de entrenamiento no varía mucho con respecto a los de décadas pasadas. Pues bien, se pretende mejorar esta situación haciendo un análisis biomecánico, estadístico y gráfico tridimensional de entrenamientos deportivos para obtener aquellos que tengan mejores resultados. Los datos estadísticos se obtienen en la temporada deportiva de 6 Universidades del D.F. y área metropolitana. Tomando en cuenta que la mayor concentración de la población está en la ciudad de México y área metropolitana, la investigación extrae datos de universidades que radican en estas áreas. Dicha investigación se efectúa en temporada de torneos estudiantiles.

Se incluye un análisis biomecánico deportivo y una graficación por computadora para conocer las posiciones de los centros de gravedad y observar las parábolas de los saltos que se forman en base a la velocidad así como los grados de inclinación.

Es verdad que para los amantes de otras disciplinas deportivas no sea tan importante este detalle deportivo, sin embargo para la mayoría de los jugadores de basquetbol es tan importante como el juego mismo. La presente investigación tiene como fin abstraer la esencia de los entrenamientos desde el punto de vista biomecánico-matemático y en base a los análisis y resultados de regresión, seleccionar los mejores ejercicios para salto. Finalmente se observará a través de un desarrollo de animación gráfica en 3D, el funcionamiento del cuerpo en movimiento al ejecutar dichos saltos.

La solución del problema va enfocada a los jugadores de básquetbol universitario y a los entrenadores. Sin embargo el alcance busca ir más allá de dichas instituciones superiores, puesto que sentará más bases para que en el futuro se llegue a tener un mejor desempeño como podría ser el campo olímpico o profesional.

Objetivo

Recabar información biomecánica del funcionamiento de las partes del cuerpo que intervienen en el salto vertical y en el salto parabólico e información estadística para observar los resultados de diversos entrenamientos para salto. En base a dicha información desarrollar un software de apoyo a entrenadores de básquetbol de educación superior para analizar y mejorar la altura y suspensión de los saltos.

Hipótesis

Es posible mejorar la altura y suspensión del salto vertical y parabólico por medio de la información obtenida por el análisis biomecánico del funcionamiento de los segmentos del cuerpo que intervienen al saltar. Se podrán seleccionar los entrenamientos que den mejores resultados al observar las estadísticas de regresión lineal. Por otro lado se podrá entender mejor el funcionamiento biomecánico de salto por medio de una animación gráfico-tridimensional y principalmente, se espera que sea de gran utilidad para los deportistas y sus entrenadores el hacer uso de dicha información que brindará el software desarrollado.

*"Usted cree en un Dios que juega a los dados,
y yo,
en la ley y el orden absolutos
en un mundo que existe objetivamente,
y el cual, de forma insensatamente especulativa,
estoy tratando de comprender."*

*"Dios no juega a los dados con el cosmos."
"...se manifiesta en la armonía de lo que existe"*

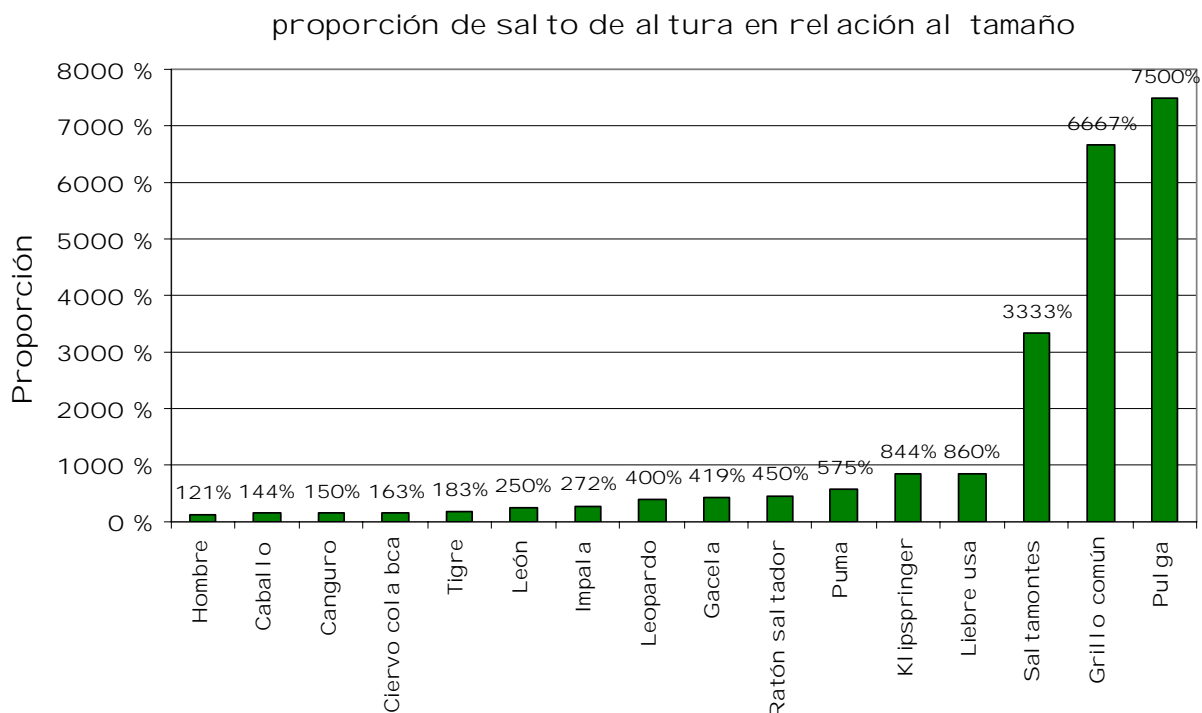
ALBERT EINSTEIN

1. Antecedentes

1.1 Antecedentes Deportivos

Usualmente para saltar y librar una barra horizontal situada a unos 2.28 metros sobre el suelo. Un hombre tiene que tomar suficiente impulso y dar un salto apoyándose sobre una de las piernas. Sin embargo la mayoría de los animales saltan haciendo fuerza con las dos patas posteriores.

En el caso de los animales frecuentemente el saltar puede ser un asunto de vida o muerte como ya se ha comentado, el medio para ganar su sustento o salvar la vida, mientras que para el ser humano sea más usado para fines deportivos. Haciendo una comparación en proporciones con los animales se observa el rendimiento del humano entrenado para salto de altura olímpico. Enseguida se muestran cuáles son las criaturas más sobresalientes en el salto de altura.



salto de altura			
Nombre	tamaño en mts	salto máximo en mts	proporción
 Hombre	1.9	2.3	1.21
 Caballo	1.8	2.6	1.44
 Canguro	2	3	1.50
 Ciervo cola bca	1.5	2.45	1.63
 Tigre	3	5.5	1.83
 León	1.2	3	2.50
 Impala	0.9	2.45	2.72
 Leopardo	0.75	3	4.00
 Gacela	0.8	3.35	4.19
 Ratón saltador	0.2	0.9	4.50
 Puma	0.8	4.6	5.75
 Klipspringer	0.9	7.6	8.44
 Liebre usa	0.25	2.15	8.60
 Saltamontes	0.06	2	33.33
 Grillo común	0.03	2	66.67
 Pulga	0.004	0.3	75.00

Como se observa en la tabla, "La pulga puede elevarse de un salto a una distancia superior a cien veces su tamaño, lo que equivaldría a que un ser humano realizara un salto de más de 100 metros de alto. Además, la pulga puede dar esos brinco durante 78 horas sin descansar.

Por lo general no cae sobre sus patas después del quinto salto, sino sobre su cabeza o espalda sin tener vértigos o lastimarse debido a la conformación especial de su cuerpo.

El esqueleto de este insecto, formado por numerosas placas a la manera de una coraza, está constituido por un compuesto duro llamado "esclerotina" (producto del entrecruzamiento de la quitina con cadenas de proteínas) que se ubica en el exterior del cuerpo y lo envuelve por completo. Esa estructura absorbe y neutraliza el impacto de cada salto.

La pulga no posee vasos sanguíneos sino que su interior está lleno de una sangre fluida y clara que actúa de amortiguador de los órganos interiores que flotan en ese medio. A ello se debe que la presión abrupta producida por cada brinco no le afecte para nada. La sangre se purifica a través de aberturas de aire esparcidas en todo el cuerpo y elimina la necesidad de un elemento que bombee oxígeno continuamente. El corazón tiene la forma de un tubo y late tan lentamente que los saltos no producen ningún inconveniente al órgano.

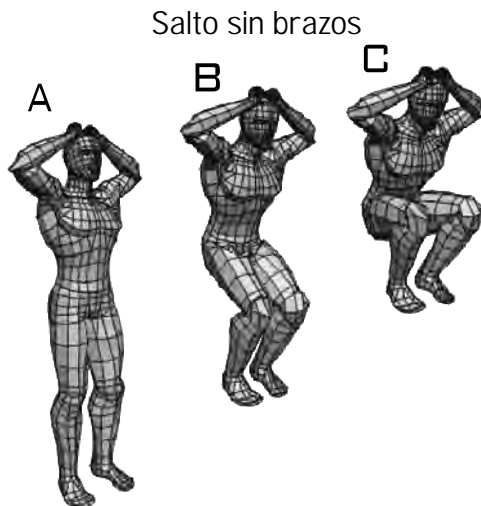
Al descubrir los científicos que los músculos de las patas no resultaban ser tan fuertes como era de esperarse, investigaron qué era lo que posibilitaba la altura alcanzada. Así se enteraron que posee un sistema de resortes adicionado a las extremidades que trabaja gracias a una proteína con propiedades elásticas llamada "resilina", donde se almacena energía mecánica. La sorprendente propiedad de esa sustancia radica en su capacidad de liberar hasta el 97% de la energía acumulada al momento de estirarse. El material más flexible conocido hoy día en el mercado, llega a liberar solamente 95% de la energía acumulada. La resilina está ubicada en la base de las largas patas traseras, en almohadillas diminutas.

En unas pocas décimas de segundo la pulga se prepara para el brinco comprimiendo ese material al mismo tiempo que contrae las patas. Un mecanismo tipo cremallera sostiene las patas recogidas hasta el momento en que un músculo se relaja y la estructura tipo resorte da impulso a un salto extraordinario a través de la energía acumulada en la resilina" (022).

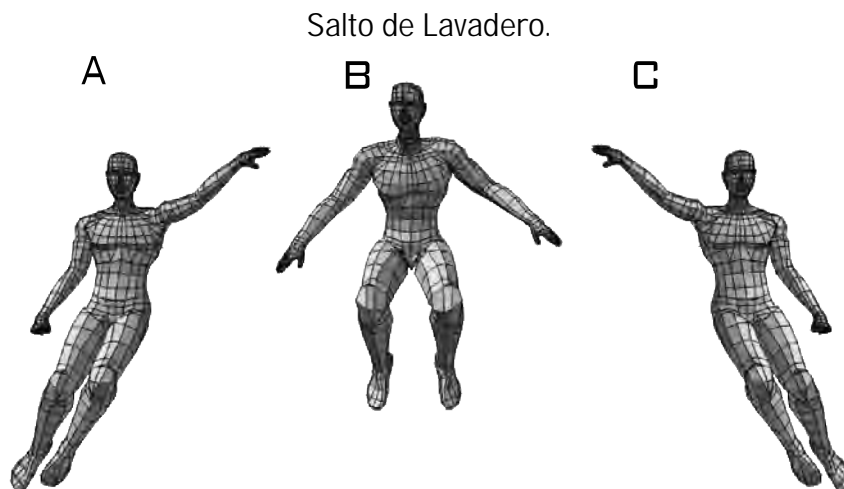
El mayor salto sin impulso previo es el de la gacela saltarina, (*Antidorcas marsupialis*), que estando parada puede dar saltos de 3 metros de altura.

1.2 Entrenamientos más comunes

Ahora bien, como antecedente. Entre los entrenamientos más comunes conocidos se encuentran los siguientes, mismos que tienen el objetivo de aumentar la fuerza y velocidad en las piernas.

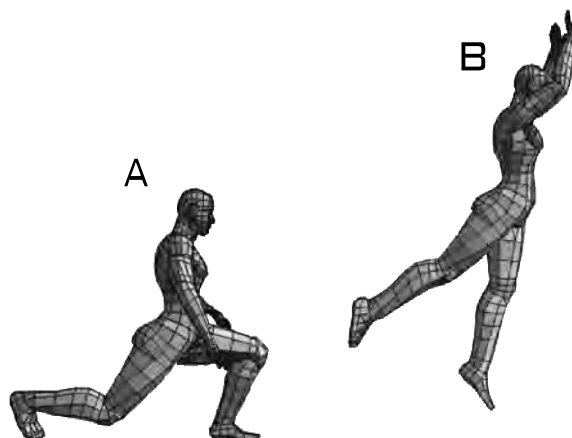


Son series de saltos muy intensas. Los saltos sin brazos ayudan al aumento explosivo. Descripción: Manos en la cabeza. Empezar saltando hacia arriba y levantando las piernas hacia delante, subir las piernas en el ascenso hasta tocar el pecho, expulsar las piernas hacia abajo hasta que hayan tocado el suelo.



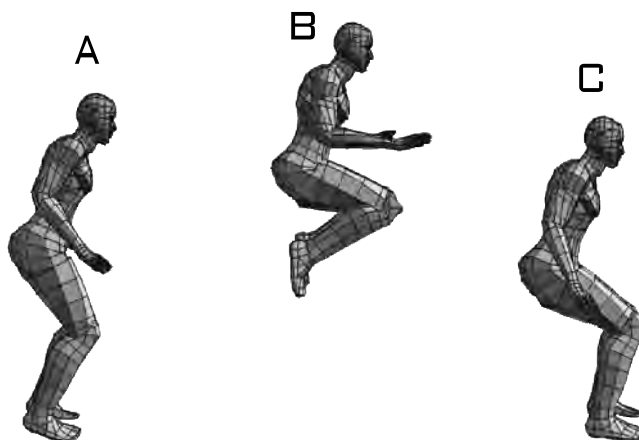
Aumenta los movimientos explosivos laterales, especialmente para mejorar la habilidad y la velocidad alta al cambiar de direcciones. Descripción: Empezar usando un cono, bolsa de papel, u otro objeto rompible elevado de 20 a 60cm de altura. Saltar lateralmente sobre el objeto. Procurar saltar tan lejos y tan alto como se pueda atrás y adelante sobre el objeto.

Salto de Pierna Flexionada



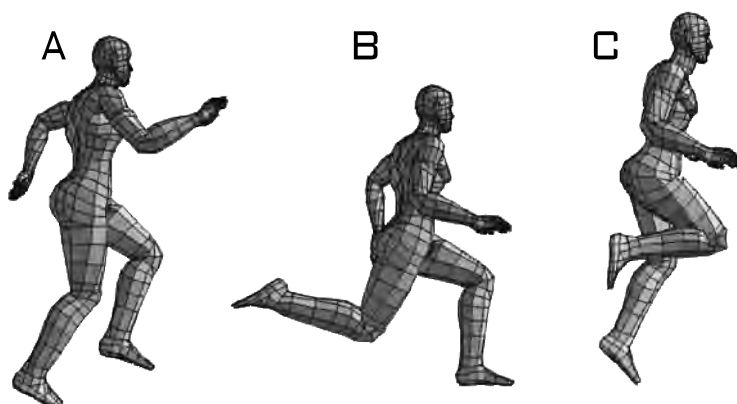
Ayuda a mejorar la explosión de una sola pierna. Los Saltos de pierna hendida usan una prolongación del movimiento corriendo y ejecutandolo en ese orden para trabajar mas la explosión de los músculos más bajos del cuerpo. Descripción: El Salto de pierna flexionada es un ejercicio simple que usa un movimiento prolongado. Cuando se aterriza hay que tener la pierna doblada a unos 90 grados. Hay que procurar alcanzar la máxima altura y distancia.

Salto a Doble Pierna



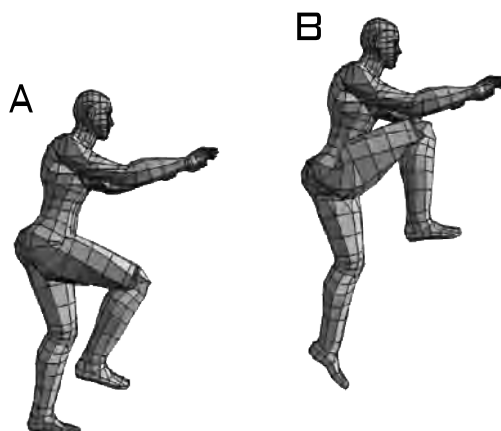
Es similar al salto de una sola pierna pero con menos tensión en el cuerpo. Descripción: Usar las ilustraciones como referencia al ejecutar el salto de pierna doble. Procurar la distancia máxima al ejercitarlo.

Limitando Una Sola Pierna



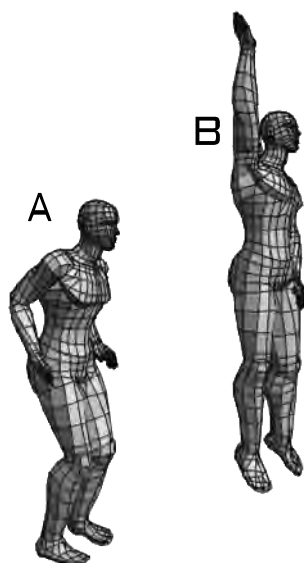
Algo similar al salto de la pierna flexionada, pero dando énfasis al saltar horizontalmente y acelerando en la mayor extensión de la pierna. Descripción: Es sencillamente un ejercicio que consiste en dar múltiples saltos largos. Hay que procurarlo para la distancia máxima y las explosiones. Usar una superficie acolchonada como una alfombra o usar plantillas acolchonadas en los tenis para ayudar a absorber la tensión del cuerpo.

Saltos de Poder



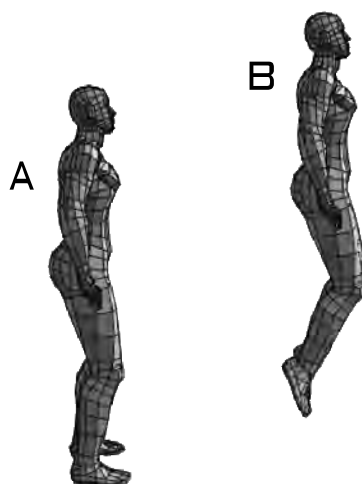
Hacer contacto desde muy abajo al empezar el ejercicio. Trabaja el músculo flexor de la cadera y los glúteos de manera explosiva. Es similar al de una sola pierna. Se debe de tomar la fuerza necesaria para mejorar en la carrera. Descripción: Es un salto de poder que usa un movimiento prolongado al saltar. Se debe llevar una pierna hasta el pecho cuando se esté elevando. Hay que balancear ambos brazos hacia arriba cuando una pierna esté junto a la otra. Se deben alternar las planteadas.

Salto de Borde



Comenzar a intensidad baja. Trabaja los músculos necesarios para los saltos verticales explosivos. Descripción: Salto de borde es un simple ejercicio que al saltar se utiliza un tablero u otro objeto alto. Hay que empezar saltando hacia arriba y estabilizar el máximo salto tratando de alcanzar el borde con la punta de una mano. Alternar las manos.

Salto de Tobillo.



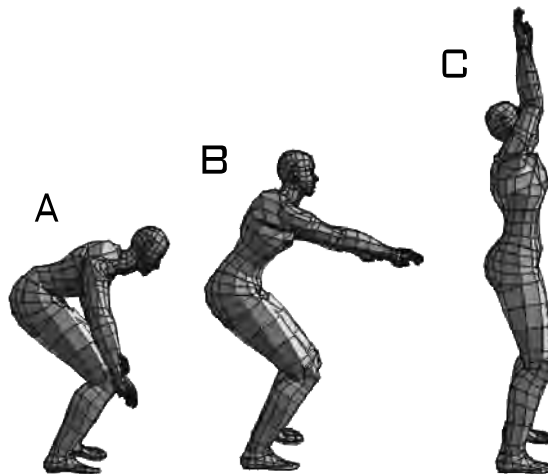
Quizás es el más seguro y simple de todos los ejercicios. Los saltos de tobillo definen exclusivamente los músculos gemelos. Descripción: Saltar hacia arriba usando sólo los músculos gemelos, usando un movimiento completo.

Aceleraciones.



Las aceleraciones consisten en correr a la 2ª velocidad de sprint máximo y entonces acelerar a máxima velocidad para luego volver a la 1ª velocidad. Descripción: Empezar sprintando a 2ª velocidad máxima, luego acelerar a velocidad completa. Tan pronto como se complete la velocidad máxima, golpear lentamente el suelo para desacelerar y repetir.

Lanzamiento Hacia Atrás.



Trabaja las explosiones del músculo deltoides. El lanzamiento hacia atrás, ayuda a mejorar la velocidad de los brazos que son girados en el empuje cuando se está saltando. Descripción: Apretar una pelota, o llenar un balón de basquetbol viejo con agua (aprox. 2.26kg) o usar bolsa de arena (aprox. 2.26kg) y apretarla con ambas manos. Empezar agachado usando un movimiento corto y mostrando la pelota antes de iniciar el lanzamiento hacia arriba y hacia atrás.

Para todos los ejercicios se debe usar una superficie acolchonada como una alfombra o usar plantillas acolchonadas en los tenis para ayudar a absorber la tensión del cuerpo [035].

1.3 Factores genéticos

A continuación se mencionan los factores que afectan de manera directa la ejecución de un salto vertical y parabólico como pueden ser algunos propios de la raza, la fisonomía del atleta y el sexo, de allí que algunos ejercicios para salto no tengan los mismos resultados en todos los atletas (tanto en la rama varonil como en la femenil). Es importante hacer notar que en los deportes una raza domine más que la otra dependiendo de la prueba que se trate, en este caso la que nos ocupa es el salto basquetbolístico en el que se destaca la prominencia de la raza negra así como en las carreras de velocidad. Sin embargo es necesario distinguir un sexo del otro por separado puesto que existen diferencias en la distribución de masas corporales alterando la posición del centro de gravedad principal. Por otro lado se observa también la diferencia de brazos de palanca en la estructura ósea de los atletas en donde hay desigualdad de rendimiento entre una morfología y otra.

1.3.1 El dominio de la raza afro

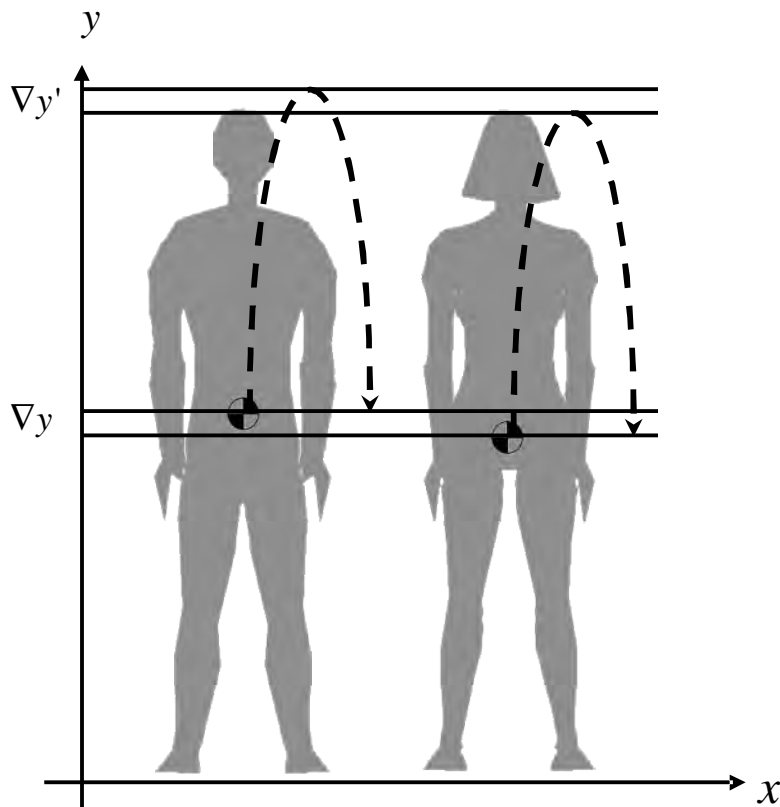
La lista de los basquetbolistas así como velocistas negros que han sobresalido en el mundo del atletismo es interminable, pero al observar este dominio invariable, surge la pregunta del porqué la raza negra tiene este predominio. Andrezej Piotrowsky, entrenador de velocistas mexicanos y originario de Polonia, manifestó que se habla mucho de la influencia del biotipo en el caso de la velocidad de los corredores negros, quienes son de piernas largas y su centro de gravedad es alto.

Pero eso no es todo, Piotrowsky indicó que hace tiempo científicos de Finlandia realizaron nuevos estudios para observar cómo es el organismo en su estructura muscular. Para el estudio se realizaron pruebas a 300 atletas, de los cuales retiraron una muestra de músculos de pierna, y se observó que existe un determinado porcentaje de fibras blancas y fibras rojas. En la investigación se descubrió que cuando el individuo tiene un mayor porcentaje de fibras blancas, es más apto para la velocidad, y quien tiene mayor porcentaje de fibras rojas es más apto para desarrollarse en las pruebas de fondo. Andrezej Piotrowsky especificó que las fibras blancas son más grandes y permiten movimientos más rápidos en la estructura muscular. En lo referente a las rojas, son más pequeñas, permiten movimientos lentos, al igual que las contracciones, y tienen alta calidad de cambio del oxígeno [003].

De esto se comprende el porqué en el básquetbol profesional de otros países predomine este tipo racial como los atletas más sobresalientes.

1.3.2 Diferencia entre los sexos

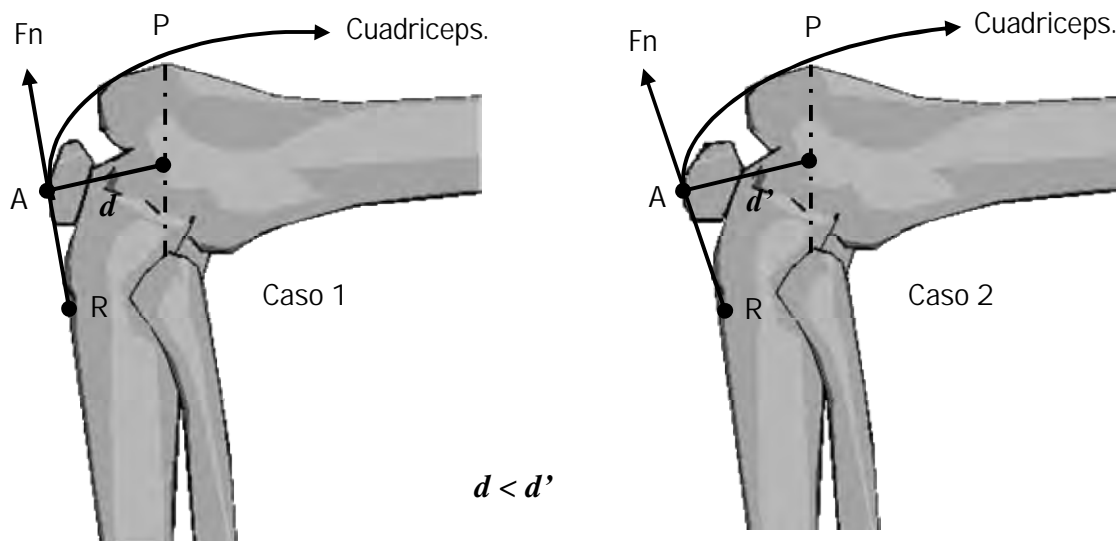
Existe una diferencia en cuanto al rendimiento debido a la fuerza ejercida y por otro lado debido a que la posición del centro de gravedad en las mujeres está más abajo que en el de los hombres [027]. Las caderas aumentan peso en la parte inferior del cuerpo femenino y hay menos peso en los hombros y en la espalda. El salto ejecutado difiere entre uno y el otro sexo. Al ejecutar un salto parabólico y elevando los brazos para subir el CG, el origen y la parábola formada en la rama femenil será de menor altura que en la rama varonil en atletas de la misma estatura, compleción y peso. No obstante, aunque las parábolas coincidan, la diferencia estará en el grado de fuerza.



Diferencias en la posición del centro de gravedad en ambos sexos, en el supuesto de que ambos tengan la misma estatura, mismo peso y misma compleción.

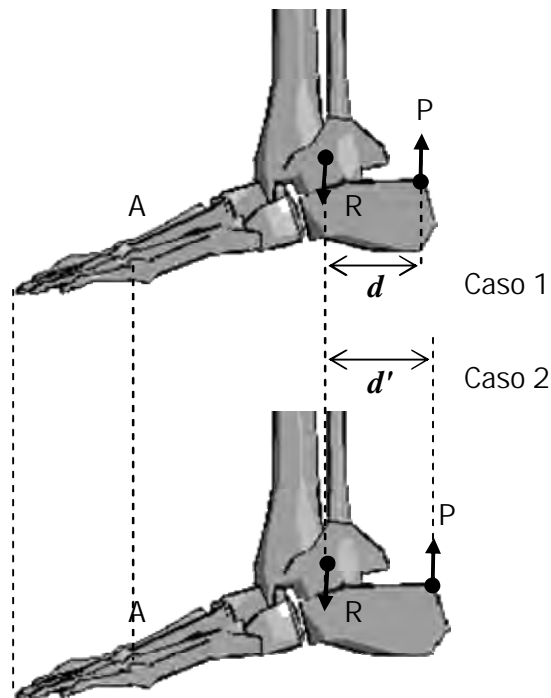
1.3.3 La prominencia de los brazos de palanca

En algunos casos la ejecución de un salto tendrá mejor rendimiento en un atleta que en otro -a pesar de llevar el mismo entrenamiento-, debiéndose en parte a la genética de los huesos de cada atleta. Se tiene un ejemplo relacionado con la eficacia mecánica del músculo cuádriceps cuando actúa sobre la pierna (F_n). La dirección que adopta dicha fuerza está condicionada por la dirección seguida del tendón de inserción y ésta por la mayor o menor protuberancia de la rótula. En el caso 2, se puede decir que se trata de un atleta que tiene una rótula prominente y, consecuentemente, su brazo de palanca es relativamente mayor que el presentado en el caso 1, cuya rodilla tiende a ser más plana. Con esta situación se puede decir que la eficacia del músculo cuádriceps es mayor en el caso 2 que en el caso 1, el que se ve determinado por la distancia del brazo de palanca d y d' [O15]. Si la fuerza que el músculo ejerce se realiza lo más lejos que se pueda del punto de apoyo de la palanca se obtendrá una mayor eficacia.



Vector resultante (F_n) de la fuerza de actuación del músculo cuádriceps sobre la pierna, donde A=Apoyo, P=Potencia, R=Resistencia, d =Distancia del brazo de palanca al punto de apoyo.

Cuando el brazo de palanca de la potencia es mayor, la fuerza resultante también es mayor en el caso 2 que en el caso 1, debido a que genéticamente el atleta tiene un calcáneo más largo en proporción a otros atletas de la misma estatura. En estos caso se ejerce una mejor potencia de salto. El origen de los músculos gemelos se encuentra en la parte inferior y posterior del fémur y están insertados en el calcáneo por medio del tendón de Aquiles. Cuando estos se contraen extienden el pie y flexionan la pierna. Si existiese un brazo de palanca más prominente como el ya citado, se podría obtener un mejor salto en el atleta con el mismo esfuerzo que en el caso 1.



1.4 Factores externos

Algunos factores que afectan indirectamente a la altura obtenida de un salto vertical o parabólico se encuentran determinados por el sitio geográfico en nuestro planeta. Esto significa que el atleta depende del lugar en que se encuentra en cuanto a latitud y altitud se refiere para poder tener un bajo o alto rendimiento de su entrenamiento y de sus resultados.

1.4.1 Altitud y Latitud

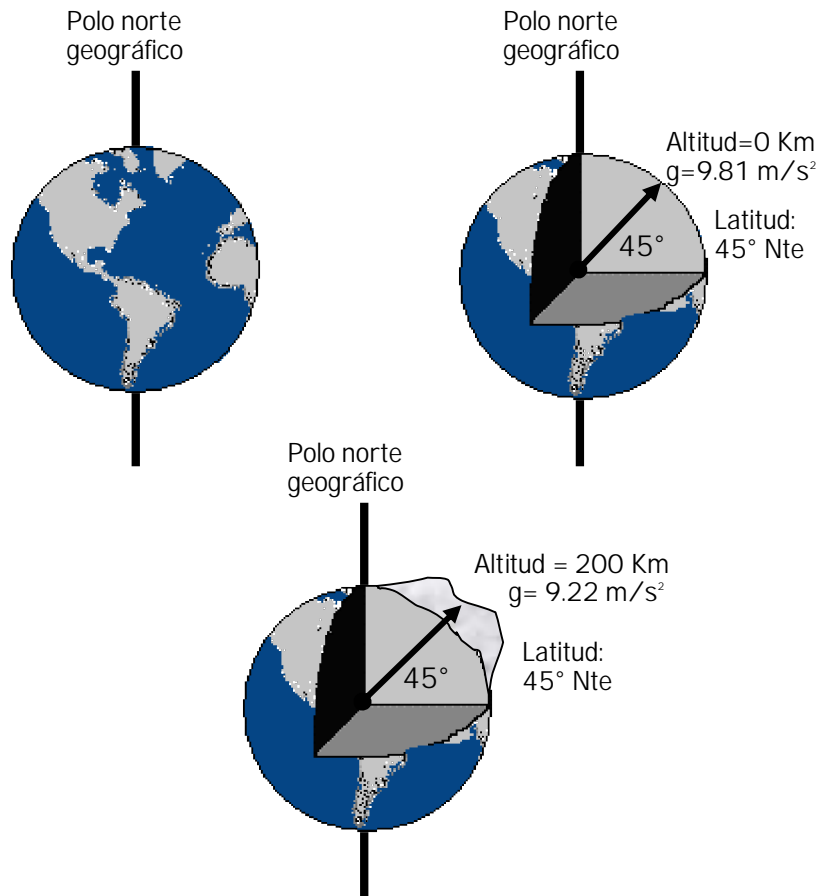
Altitud es la distancia vertical de un punto sobre el nivel del mar. La altitud es inversamente proporcional al valor de la gravedad. A mayor altitud, menor gravedad.

Un deportista no saltará a la misma altura en la cima de una montaña que en las costas o en los valles como puede ocurrir en la ciudad de México. El basquetbolista saltará menos en una costa por la altitud baja, puesto que la gravedad es más fuerte en estos sitios, provocando que su peso aumente ligeramente, sin embargo en un lugar alto como las montañas el atleta será más ligero y saltará más alto.

Aunque es poca la diferencia, en la tabla se puede apreciar que entre mayor sea la altitud, menor será la gravedad debido al cambio de la presión atmosférica y lejanía del centro de gravedad de la tierra [001].

Variación de la gravedad con la altitud
(Tomando la latitud 45° Norte como constante)

Altitud kmts	$g(m/s^2)$
0	9.81
20	9.75
40	9.69
60	9.63
80	9.57
100	9.51
200	9.22

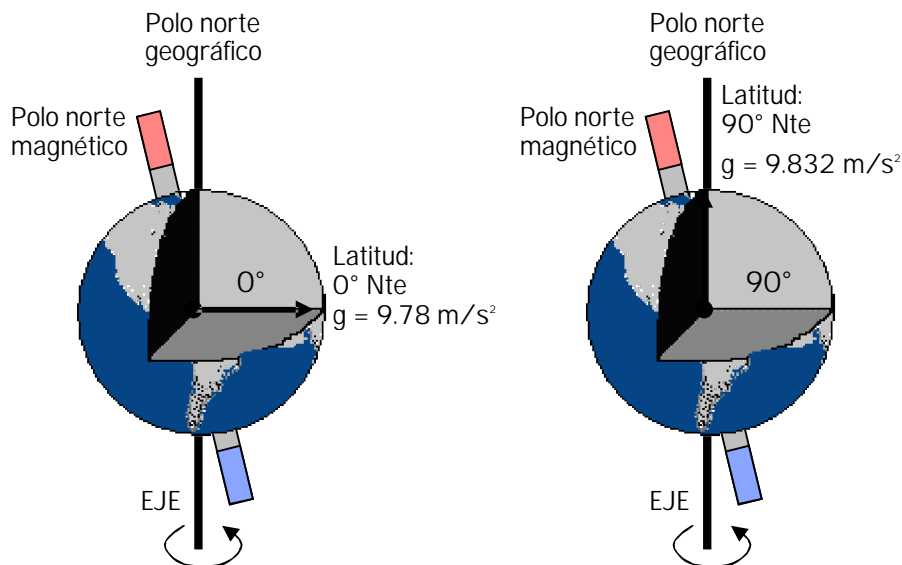


Latitud es el ángulo que forman la vertical (del centro de la tierra a la superficie en el nivel del mar) y el plano ecuatorial terrestre. Se mide hacia el norte o hacia el sur. El valor mínimo es 0 y el máximo es 90° norte o sur. A mayor latitud, mayor gravedad, en otras palabras que tan cerca o que tan alejados se está del ecuador y de los polos geográficos y magnéticos de la tierra. Aunque no lo parezca este factor también altera el resultado en las marcaciones de saltos de altura y de longitud, y por ende, el rendimiento de un entrenamiento y una ejecución de salto.

Es una variación muy pequeña, sin embargo importante, esto se debe en parte a la rotación de la tierra que crea una fuerza centrífuga mientras más alejado se encuentre un cuerpo de los ejes de rotación (polos geográficos). Y Por otra parte se debe a que la fuerza de gravedad es mas fuerte en los polos magnéticos que lejos de ellos [009].

Variación de la gravedad con la latitud
(Tomando como constante la altitud al nivel del mar)

Latitud Norte	$g(m/s^2)$
0°	9.78
20°	9.786
40°	9.802
60°	9.819
80°	9.831
90°	9.832



El valor de la aceleración de la gravedad influye en el resultado de un salto de altura en distintos lugares. Una persona cerca de los polos magnéticos de la tierra tiene un mayor peso que en el ecuador [001].

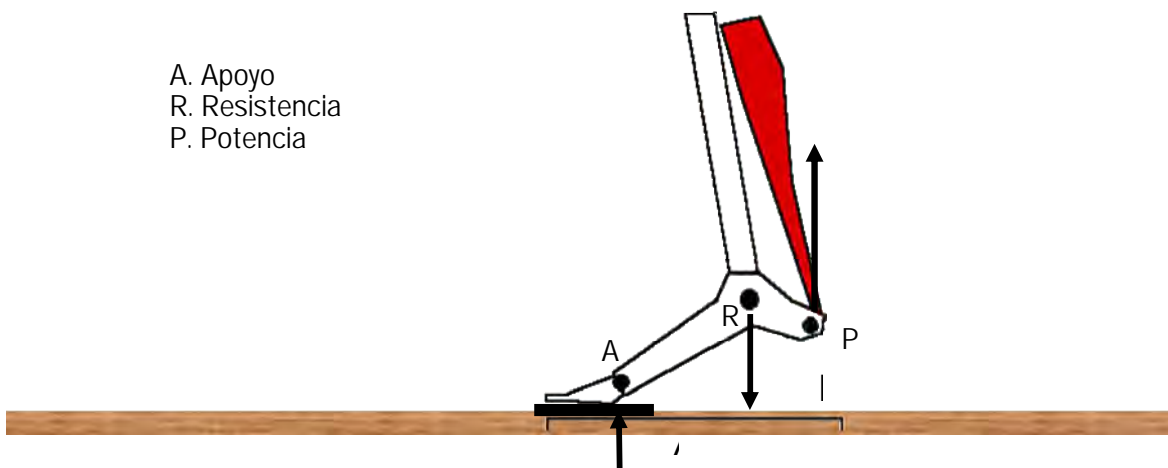
1.4.2 El calzado deportivo

La fuerza que impulsa a un corredor al igual que a un saltador basquetbolista, es la que ejerce el suelo contra él. Para que exista eficacia en el salto dicha fuerza debe ser la misma que ejerce el saltador contra el suelo (3ra ley de Newton). Esto significa que con la misma fuerza que un corredor empuja al suelo hacia atrás, el suelo empuja al corredor hacia delante. De la misma manera, con la fuerza que un saltador empuja el suelo hacia abajo, el suelo empuja al saltador hacia arriba.¹



Si la suela del tenis es demasiado deformable o muy suave en esta parte anterior (empeine), parte de la fuerza de acción se desperdiciará en deformar la suela del tenis, produciendo una reducción en la fuerza de reacción del suelo contra el saltador [O15].

A. Apoyo
R. Resistencia
P. Potencia



La parte anterior del pie soporta todo el apoyo donde se concentra la acción dando como resultado la reacción en el mismo punto de apoyo. Dicha reacción se verá afectada por el grado de deformación de la suela.

¹ Los tenis con amortiguadores en la parte posterior (el talón) tienen la función de absorber el golpe y las fuerzas del impacto -en la caída de un salto o en la carrera- que causan el dolor del talón, de rodilla y de la espalda en la zona lumbar.

*“Cuando puedes medir aquello de lo que hablas,
y expresarlo con números,
sabes algo acerca de ello;
pero cuando no lo puedes medir,
cuando no lo puedes expresar con números,
tu conocimiento es pobre e insatisfactorio:
puede ser el principio del conocimiento,
pero apenas has avanzado en tus pensamientos
a la etapa de ciencia.”*

WILLIAM THOMSON KELVIN

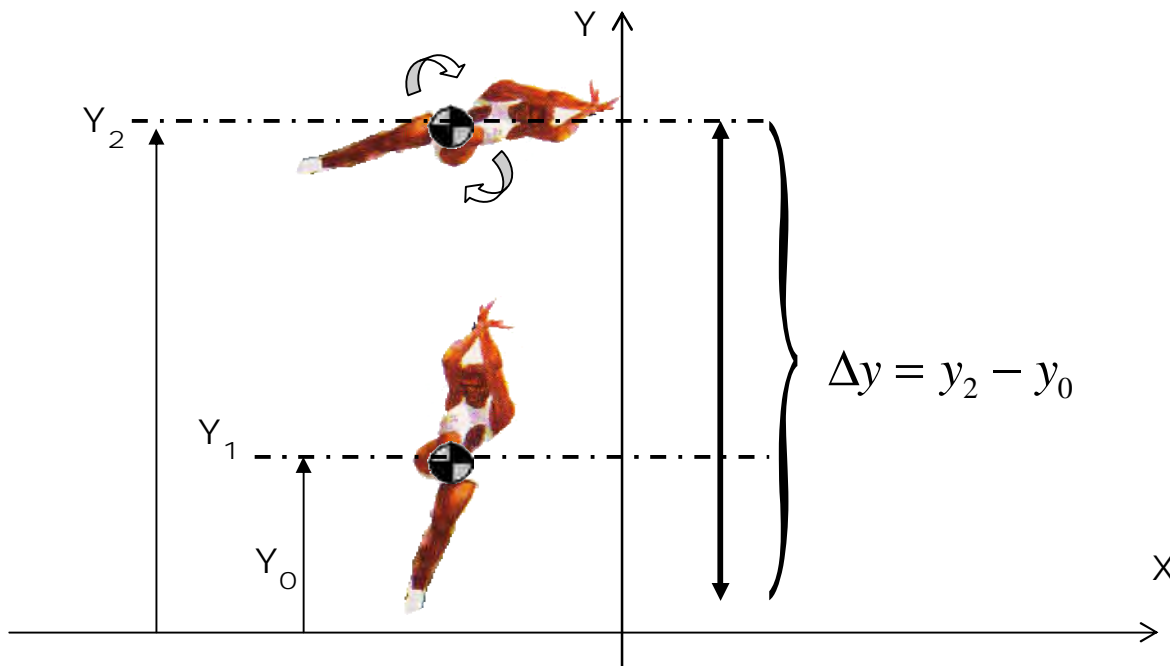
2. Tipos de salto

2.1 Salto Vertical

Salto es la acción de saltar, palabra que viene del latín saltare y significa: Levantarse alguien o algo del suelo o del sitio donde se encuentra con impulso, quedando suspendido por un momento, para volver a caer en el mismo sitio o en otro [023]. El salto vertical es el salto donde la trayectoria que describe el centro de gravedad del atleta es vertical en el ascenso y descenso de la ejecución y donde sólo existe la componente vertical del vector resultante. Este es muy usado dentro del básquetbol para recuperar rebotes, hacer tiros de suspensión y clavadas con salto explosivo en casos especiales. Ahora bien, ¿Cuánto es lo máximo alcanzado en salto vertical hasta el momento? Es verdad que en las marcaciones olímpicas en salto de altura se ha calificado arriba de 2.28 metros como promedio y en el básquetbol 1.25m, no obstante, hay una diferencia en la medición con el salto de altura en el básquetbol. En el caso del salto de altura olímpico, una vez alcanzado el máximo de altura, el cuerpo tiene que girar con respecto al centro de gravedad para poder atravesar la barra horizontal, entonces se toma la medida desde el suelo hasta la altura de la barra. Por otro lado el salto del basquetbolista se mide a partir de la posición inicial del centro de gravedad antes de saltar y la máxima altura alcanzada del centro de gravedad una vez saltado. En el caso práctico se puede tomar la medida con la punta de la mano alzada antes de saltar y la altura de la mano alzada una vez saltado al máximo [018].

2.1.1 Salto de altura olímpico

De acuerdo a la siguiente gráfica el nivel del suelo la representa y_0 y la posición de la barra en el máximo alcanzado la representa y_2 . Entonces $\Delta y = y_2 - y_0$. De manera que el hombre ha saltado con el cuerpo completo hasta la barra horizontal, por dos tipos de movimiento: 1) de traslación y 2) de rotación, en donde la altura máxima de la barra se toma a partir del suelo y no a partir de su centro de gravedad. Así se toma la medida de Δy .

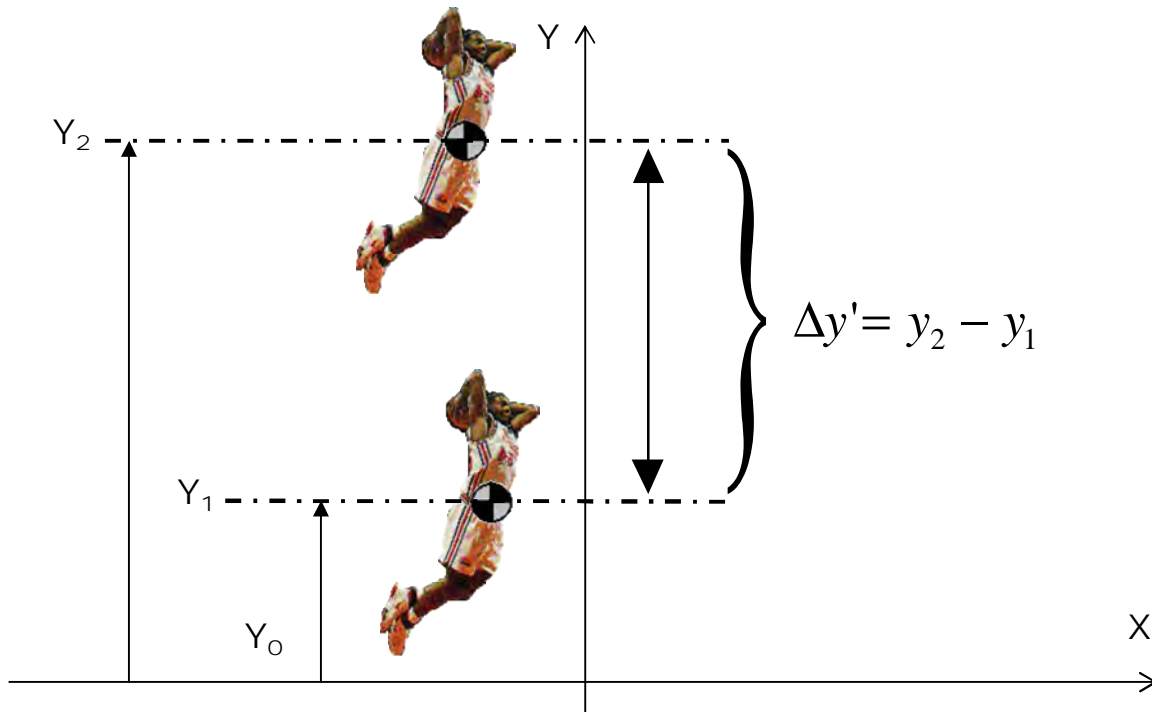


2.1.2 Salto de altura basquetbolístico

De acuerdo a la siguiente gráfica la posición original del centro de gravedad la representa y_1 y la posición final del máximo alcanzado la representa y_2 . Entonces $\Delta y' = y_2 - y_1$

De manera que en este caso el hombre no ha saltado menos de la marca de la barra horizontal olímpica, puesto que sólo ha efectuado un movimiento de traslación sin haber hecho una rotación del cuerpo, además de que la medida de su altura se hace de las siguientes dos maneras:

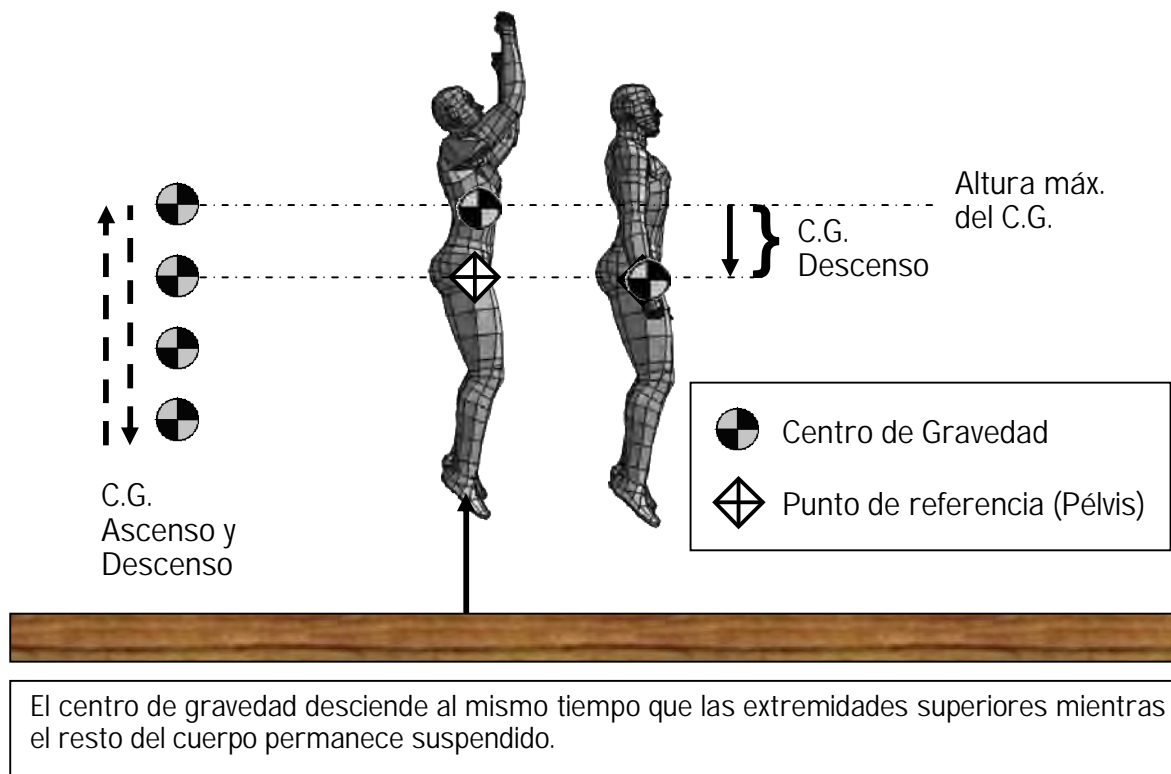
- 1) Tomando la altura de su centro de gravedad con respecto del suelo antes de saltar (con los brazos en alto) y después tomando la altura de su centro de gravedad con respecto del suelo una vez saltando (con los brazos en alto). A esto se saca la diferencia de la primera y la segunda medida. En este caso se considera la medida de $\Delta y'$
- 2) Tomando la altura de la punta de su mano alzada con respecto del suelo antes de saltar y después tomando nuevamente la altura de la punta de su mano alzada con respecto del suelo una vez saltando. A esto se saca la diferencia de la primera y la segunda medida.



2.1.3 Salto vertical en suspensión

Después de varias observaciones en los jugadores y videos de básquetbol. Se puede ver que un tiro al salto bien entrenado trae como consecuencia el efecto de suspensión, es decir que parece que el cuerpo humano se suspende por unos instantes en el aire antes de caer. Es verdad que la trayectoria un proyectil es siempre una parábola, y esto sucede con todos los cuerpos rígidos. Cuando se observa a un saltador, su centro de gravedad es siempre una parábola. No obstante hay que recordar que el cuerpo humano es un sistema articulado que puede cambiar la posición del centro de gravedad total con la rotación de sus segmentos. Entonces al momento de saltar si el CG forma una parábola y en el aire se rotan algunos segmentos, el resultado es que el cuerpo se traslada y/o rota con respecto al CG sin cambiar la trayectoria parabólica. Es decir, que si se intenta descender el CG en el aire con los brazos y/o piernas, el resto del cuerpo se eleva. Y si se intenta elevar el CG en el aire con los brazos y/o piernas, el resto del cuerpo descende (3ª ley de Newton). De esta forma se conserva la trayectoria parabólica del CG.

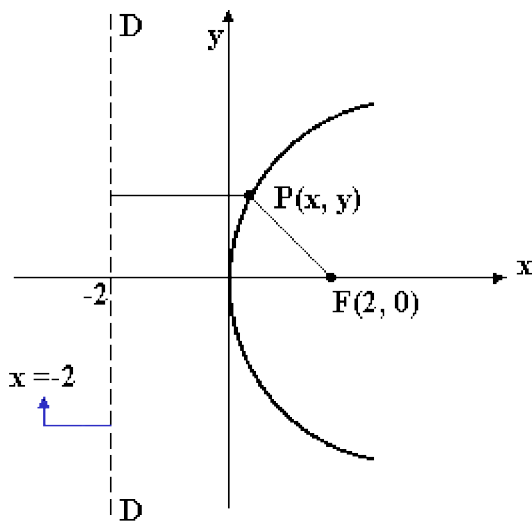
Al estar de pie y levantar los brazos, el centro de gravedad sube porque ahora hay mas masa distribuida en la parte superior del cuerpo haciendo que el punto de equilibrio suba un poco más. Si se vuelve a bajar los brazos el centro de gravedad volverá a descender. El efecto de suspensión se produce cuando sólo se observa un punto de referencia del cuerpo humano al saltar, como puede ser la pelvis. Cuando esto ocurre la pelvis conserva la suspensión porque los miembros superiores o inferiores del cuerpo están compensando los movimientos ascendentes y descendentes del CG, mientras que por otro lado, la desaceleración constante de la gravedad va creando un efecto retardante al aproximarse a la cima de la trayectoria.



2.2 Salto parabólico

El salto parabólico es el salto donde la trayectoria que describe el centro de gravedad del atleta es una parábola desde el ascenso hasta el descenso de la ejecución, en donde existen las componentes vertical y horizontal en el vector resultante.

Ahora bien, una parábola es el conjunto de todos los puntos de un plano que equidistan de un punto fijo F (llamado foco) y de una recta fija D (llamada directriz), ambos contenidos en un plano [030].



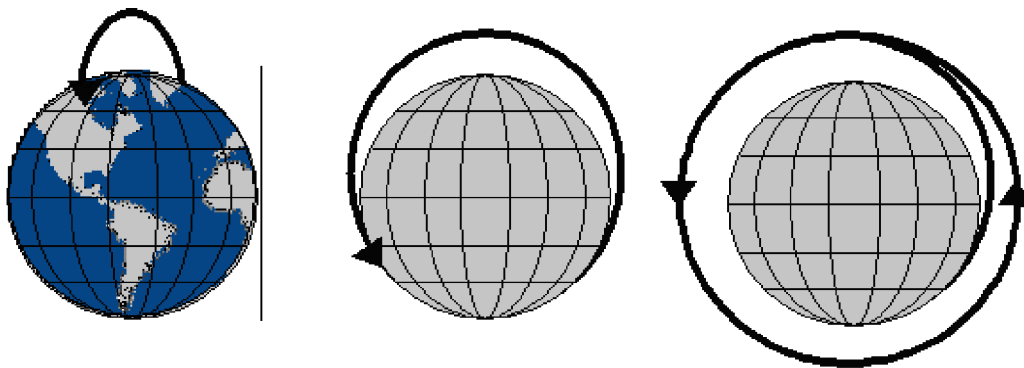
- F Foco.
- D Directriz.
- P Parámetro. La distancia del punto P al foco coincide con la distancia a la directriz
- X Eje Recta perpendicular a la directriz que pasa por el foco.
- V Vértice. Punto de intersección de la parábola con su eje.

Desde el punto de vista geométrico (matemático) la parábola pertenece a la familia de las Cónicas. Se trata de una curva plana, abierta, que se obtiene al cortar una superficie cónica mediante un plano que no pasa por el vértice pero corta la base dejando "fuera" un ángulo menor de 180° [036].



Todos los cuerpos son atraídos al centro de la tierra. La parábola de un proyectil es parte de una elipse, al igual que el salto parabólico es también una parte de aquella. Un proyectil lanzado desde la tierra a velocidad moderada, sube hasta cierto punto y vuelve a caer a la tierra. A una mayor velocidad y con el ángulo justo el proyectil iría mas lejos en torno a la tierra sin volver a caer al suelo, puesto que en realidad el proyectil siempre está cayendo, si no existiera la gravedad, el proyectil seguiría en línea recta por el espacio. Sin embargo la gravedad atrae constantemente al proyectil provocando una desaceleración en la elevación hasta detenerse, para luego comenzar una aceleración en el descenso, al igual que sucede en un salto con impulso de carrera formando una trayectoria parabólica [030].

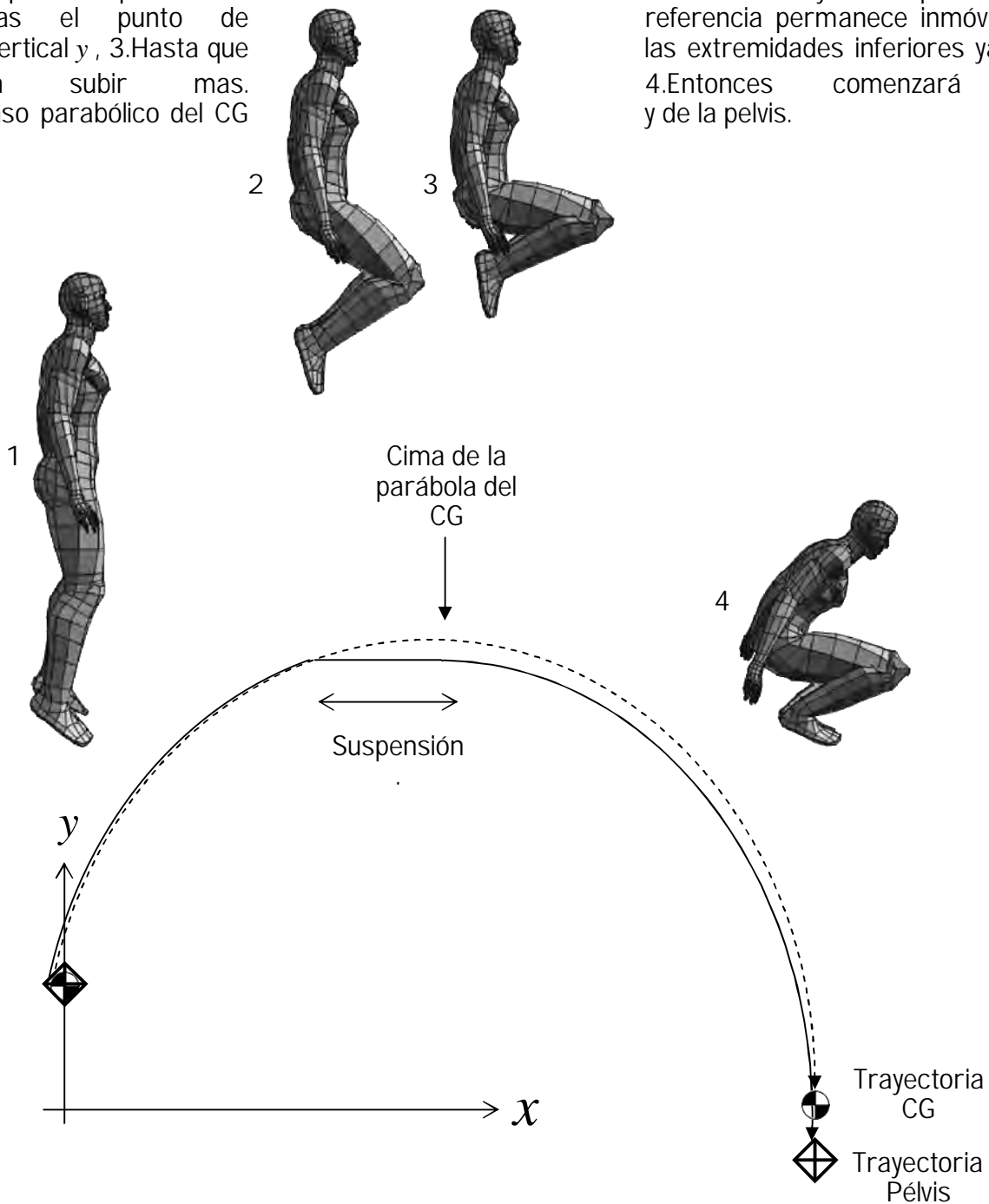
Tomando como fijo el movimiento de traslación y rotación de la tierra, dependiendo de la inclinación y velocidad del proyectil la parábola seguirá movimientos como estos:



El tercer dibujo es una órbita típica de un punto que viaja a una velocidad llamada de fuga, que le es necesaria para librarse del campo gravitacional. Las sondas espaciales interplanetarias realizan órbitas parabólicas que deben escapar al campo gravitacional de la Tierra, con el fin de dirigirse hacia los planetas.

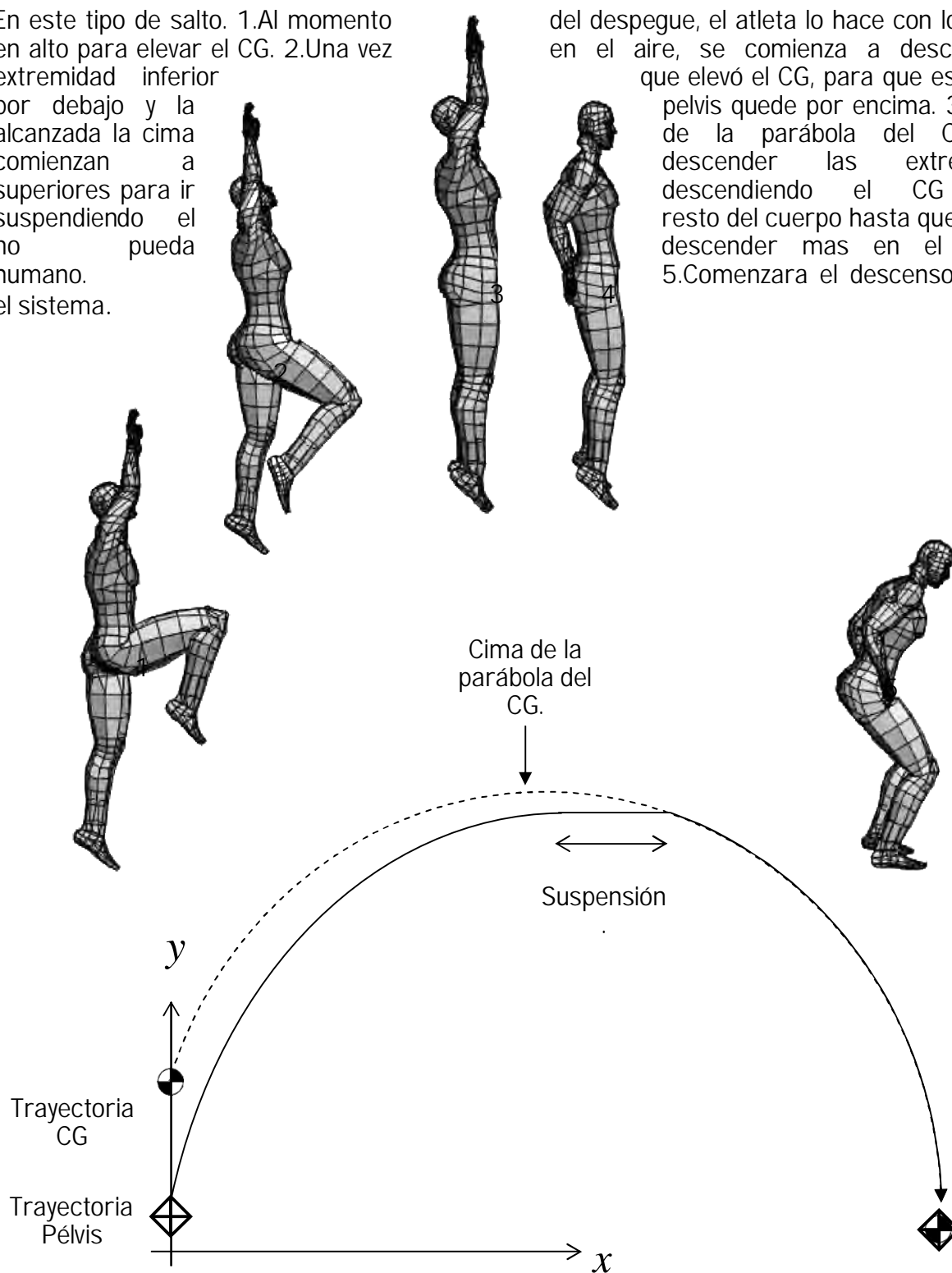
2.2.1 Salto parabólico en suspensión

Usando como punto de referencia la pelvis. 1. Cuando el cuerpo se encuentra de pie y los brazos caídos, el CG se encuentra aproximadamente a la altura del centro geométrico de la pelvis. Ahora bien, una vez saltando como se muestra abajo, 2. El atleta se suspende antes de llegar a la cima de la parábola del CG. flexionando las extremidades inferiores para elevar el CG por encima de la pelvis, por lo que el CG continúa su trayectoria parabólica mientras el punto de referencia permanece inmóvil en las extremidades inferiores ya no pueden subir más. 3. Hasta que puedan subir más. 4. Entonces comenzará el descenso parabólico del CG y de la pelvis.



En este tipo de salto. 1. Al momento en alto para elevar el CG. 2. Una vez alcanzada la cima comienzan a suspenderse las extremidades inferiores por debajo y la superior para ir suspendiendo el resto del cuerpo hasta que el CG ya no pueda moverse más en el sistema humano.

del despegue, el atleta lo hace con los brazos en el aire, se comienza a descender la que elevó el CG, para que este quede por encima de la pelvis. 3. Una vez de la parábola del CG. 4. Se comienzan a descender las extremidades inferiores descendiendo el CG e ir descendiendo el resto del cuerpo hasta que el CG ya no pueda moverse más en el sistema humano. 5. Comenzará el descenso de todo el sistema.



2.2.2 Cálculo del salto parabólico

2.2.2.1 Definiciones

Velocidad: Es una medida de que tan aprisa se mueve un objeto. Se mide siempre en términos de una unidad de distancia dividida entre una unidad de tiempo. Es una rapidez en una dirección determinada

Velocidad Constante:
$$v = \frac{d}{t}$$

Velocidad Instantánea:
$$v = \lim_{\Delta t \rightarrow 0} \frac{\Delta d}{\Delta t}$$

Velocidad instantánea: (si velocidad inicial = 0 y la aceleración es constante): $v = at$

Velocidad Promedio:
$$\bar{v} = \frac{v_1 + v_2}{2}$$

Aceleración: Es la razón de cambio de la velocidad. Es una medida de cómo cambia la velocidad respecto al tiempo. Se aplica tanto a cambios de rapidez como a cambios de dirección en la velocidad.

$$a = \frac{v}{t} \quad \text{o bien} \quad a = \frac{v_2 - v_1}{t_2 - t_1}$$

Desaceleración: Es la aceleración negativa o aceleración retardante. Puede producir un gran decremento por segundo de la rapidez.

Distancia: Es la velocidad por un intervalo de tiempo. $d = v \cdot t$

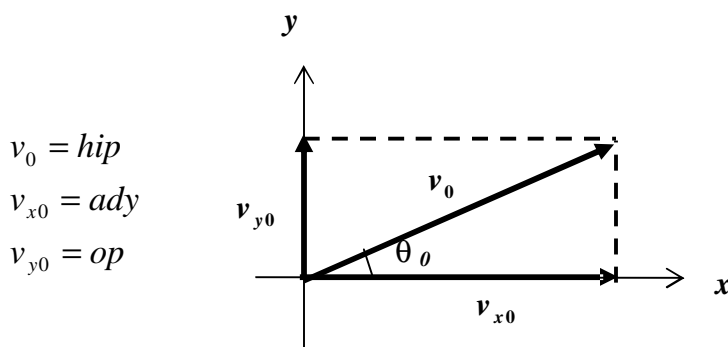
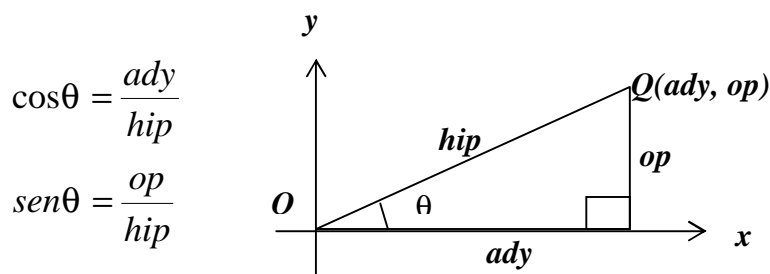
Distancia recorrida: Es la velocidad promedio por un intervalo de tiempo. $d = \bar{v} \cdot t$

<p>Aceleración de la gravedad</p> $a = \frac{v_2 - v_1}{t_2 - t_1}$ $a = \frac{9.8m/s - 0m/s}{1s - 0s}$ $a = \frac{9.8m/s}{1s}$ $a = 9.8m/s^2$ <p>Constante de gravedad g:</p> $g = 9.8m/s^2$	<p>Velocidad Instantánea</p> $v = at$ <p>Velocidad Final</p> $v = gt$	<p>Distancia recorrida al caer</p> $d = \bar{v} \cdot t$ $d = \frac{v_1 + v_2}{2} \cdot t$ $d = \frac{v_1 + v_2}{2} \cdot t$ $d = \frac{0 + gt}{2} \cdot t$ $d = \frac{1}{2}gt^2$ <p>(ec. 2.0) Distancia recorrida</p>
---	--	--

El tipo de salto parabólico tiene la forma de una parábola perfecta, siempre y cuando los factores externos no intervengan en su trayectoria, como puede ser el viento, el cual se considera como despreciable, tratándose de la resistencia del aire en un salto parabólico (a altas velocidades la resistencia del aire si es considerable). Este tipo de salto se ejecuta cuando el atleta termina de correr para dar un salto y aprovechando la velocidad de la carrera para incrementar la elevación y la longitud. La parábola que se forma con un salto en carrera, tiene la forma de un proyectil, el tiro parabólico, que es la composición de dos movimientos:

- 1) Uniforme a lo largo del eje horizontal x .
 - 2) Uniformemente acelerado a lo largo del eje vertical y .
- Este describe el movimiento del centro de gravedad del atleta. Es parecida a una función de y como la que se muestra a continuación en el desarrollo.

Se sabe por definición que: Sea θ un ángulo en la posición normal sobre un sistema de coordenadas rectangulares y sea $Q(a,b)$ cualquier punto distinto de O sobre el lado final de θ . De modo que si $d(O, Q) = r$, entonces por identidades trigonométricas:



$$\cos\theta_0 = \frac{v_{x0}}{v_0} \quad v_{x0} = v_0 \cos\theta_0$$

$$\text{sen}\theta_0 = \frac{v_{y0}}{v_0} \quad v_{y0} = v_0 \text{sen}\theta_0$$

Los componentes iniciales de la velocidad están dados por su despeje, y las ecuaciones del movimiento de un proyectil bajo la desaceleración constante de la gravedad son:

$$v_{x0} = v_0 \cos\theta_0 \quad a_x = 0 \quad v_x = v_{x0} \quad x = v_{x0}t$$

$$v_{y0} = v_0 \text{sen}\theta_0 \quad a_y = -g \quad v_y = v_{y0} \quad y = v_{y0}t + d$$

De la ecuación (2.0) de la distancia recorrida en el componente vertical ya desarrollada, se tiene:

$$d = \frac{1}{2}gt^2$$

$$y = v_{y0}t + d$$

$$y = v_{y0}t + \frac{1}{2}gt^2$$

Desaceleración constante: $a_y = -g$

$$y = v_{y0}t + \frac{1}{2}(-g)t^2$$

$$y = v_{y0}t - \frac{1}{2}gt^2$$

Entonces se pueden saber las coordenadas del proyectil en cualquier tiempo obteniendo los componentes de la velocidad y de la posición:

$$v_x = v_{x0} = v_0 \cos\theta_0$$

(ec. 2.1) Componente horizontal de la velocidad

$$v_x = v_0 \cos\theta_0$$

$$x = v_{x0}t = (v_0 \cos\theta_0)t$$

(ec. 2.2) Componente de la posición horizontal

$$x = (v_0 \cos\theta_0)t$$

$$v_y = v_{y0} = v_0 \text{sen}\theta_0$$

$$v_y = v_{y0} - gt = v_0 \text{sen}\theta_0 - gt$$

(ec. 2.3) Es el componente vertical de la velocidad

$$v_y = v_0 \text{sen}\theta_0 - gt$$

$$y = v_{y0}t - \frac{1}{2}gt^2 = (v_0 \text{sen}\theta_0)t - \frac{1}{2}gt^2$$

(ec. 2.4) Es el componente de la posición vertical

$$y = (v_0 \text{sen}\theta_0)t - \frac{1}{2}gt^2$$

Dado que no existe componente horizontal de la aceleración, la ecuación 2.1 permanece como constante a lo largo de la trayectoria. No obstante en la ecuación 2.3 se debe tomar en cuenta la fuerza de gravedad en el componente vertical.

Eliminado el tiempo en las ecuaciones que nos dan las posiciones x e y , se obtiene la ecuación de la trayectoria, que tiene la forma $y = ax^2 + bx + c$, lo que representa una parábola. Se obtiene la altura máxima, cuando la componente vertical de la velocidad v_y es cero; y el alcance horizontal máximo x cuando el cuerpo retorna al suelo $y = 0$.

Ahora bien, se puede encontrar la trayectoria parabólica despejando la t (t es el tiempo en segundos) de la ecuación 2.2 que nos dice la posición horizontal y sustituyéndola en el componente de la posición vertical en la ecuación 2.4 de la siguiente manera:
Se despeja a t de la ecuación 2.2:

$$x = (v_0 \cos \theta_0) t$$

$$t = \frac{x}{(v_0 \cos \theta_0)}$$

Se Sustituye t en la ecuación 2.4:

$$y = (v_0 \sin \theta_0) t - \frac{1}{2} g t^2$$

$$y = (v_0 \sin \theta_0) \left(\frac{x}{(v_0 \cos \theta_0)} \right) - \frac{1}{2} g \left(\frac{x}{(v_0 \cos \theta_0)} \right)^2$$

$$y = \frac{(v_0 \sin \theta_0) x}{(v_0 \cos \theta_0)} - \frac{1}{2} g \left(\frac{x^2}{v_0^2 \cos^2 \theta_0} \right)$$

Por identidades trigonométricas se sabe que: $\tan \theta = \frac{\sin \theta}{\cos \theta}$

$$\tan \theta_0 = \frac{(v_0 \sin \theta_0)}{(v_0 \cos \theta_0)}$$

$$y = (\tan \theta_0) x - \frac{1}{2} g \left(\frac{x^2}{v_0^2 \cos^2 \theta_0} \right)$$

$$y = (\tan \theta_0) x - \left(\frac{g x^2}{2 v_0^2 \cos^2 \theta_0} \right)$$

$$y = (\tan \theta_0) x - \left(\frac{g}{2 v_0^2 \cos^2 \theta_0} \right) x^2$$

Ahora se le dan valores a x desde 0 generándose la siguiente gráfica:

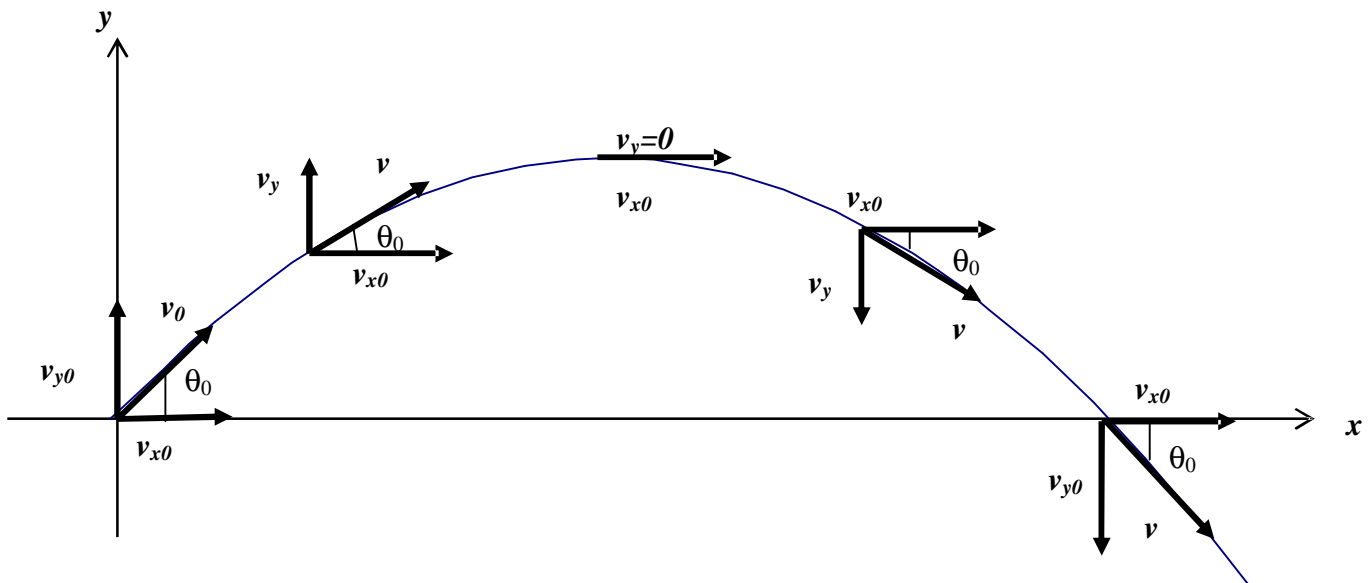
$$y = (\tan\theta_0)x - \left(\frac{g}{2v_0^2 \cos^2\theta_0}\right)x^2$$

v_0 = Velocidad inicial

θ_0 = Grados de inclinación

g = Constante de gravedad

$x \in \mathbb{Z}$



El hombre corre cerca de los 40 km/hora de velocidad, 35 km/hora equivalen a 9.7 m/seg. Si el humano llegase a alcanzar esa velocidad en el vector resultante v_0 . Al darle valores a la fórmula obtenida para la trayectoria parabólica se encontraría lo siguiente.

$$y = (\tan\theta_0)x - \left(\frac{g}{2v_0^2 \cos^2\theta_0} \right) x^2$$

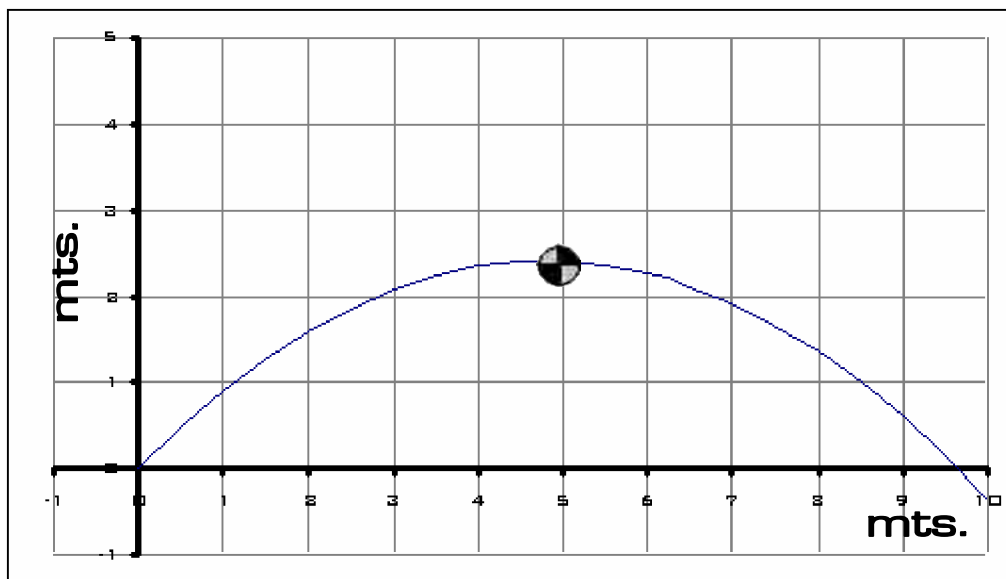
$$v_0 = 9.7m / seg$$

$$\theta_0 = 45^\circ$$

$$g = 9.8m / seg^2$$

$$0 \leq x \leq 10$$

Velocidad Km/hora	35
Velocidad m/s	9.72222222
grados inclinación	45
gravedad m/s ²	9.8
x	y1
0	0
1	0.89632
2	1.58528
3	2.06688
4	2.34112
5	2.408
6	2.26752
7	1.91968
8	1.36448
9	0.60192
10	-0.368



Diferentes ángulos en los que se ejecuta un salto parabólico. El alcance que se tiene en altura y longitud suponiendo que la velocidad es la misma constante de 9.7m/s (35 km/hora) para todos los ángulos, se tiene lo siguiente:

$$y = (\tan\theta_0)x - \left(\frac{g}{2v_0^2 \cos^2\theta_0}\right)x^2$$

$$v_0 = 9.7m/seg$$

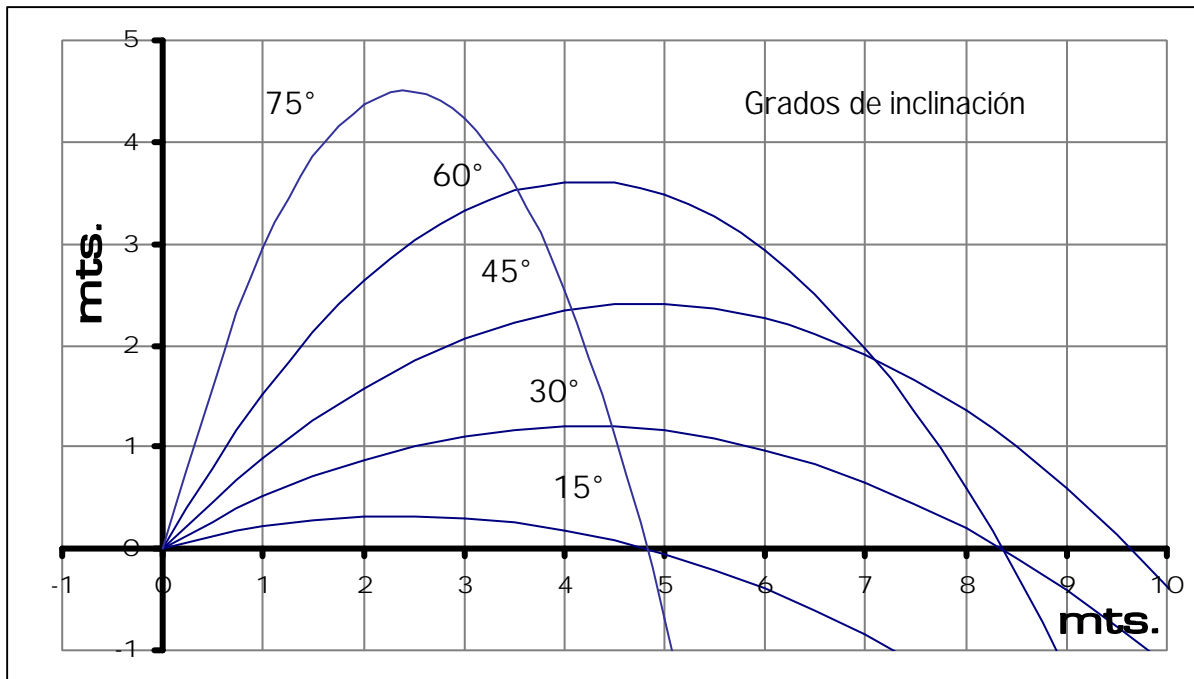
$$\theta = \{15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ\}$$

$$g = 9.8m/seg^2$$

$$0 \leq x \leq 10$$

$$x \in \mathbb{Z}$$

Velocidad Km/hora	35	35	35	35	35
Velocidad m/s	9.72222222	9.72222222	9.72222222	9.72222222	9.72222222
grados inclinación	15	30	45	60	75
gravedad m/s ²	9.8	9.8	9.8	9.8	9.8
x	y1	y2	y3	y4	y5
0	0	0	0	0	0
1	0.21238725	0.50823027	0.89632	1.52469081	2.95817275
2	0.31365061	0.87822054	1.58528	2.63466162	4.36858939
3	0.30379008	1.10997081	2.06688	3.32991242	4.23124992
4	0.18280566	1.20348108	2.34112	3.61044323	2.54615434
5	-0.04930265	1.15875135	2.408	3.47625404	-0.68669735
6	-0.39253485	0.97578162	2.26752	2.92734485	-5.46730515
7	-0.84689094	0.65457188	1.91968	1.96371565	-11.7956691
8	-1.41237091	0.19512215	1.36448	0.58536646	-19.6717891
9	-2.08897478	-0.40256758	0.60192	-1.20770273	-29.0956652
10	-2.87670253	-1.13849731	-0.368	-3.41549192	-40.0672975



Diferentes velocidades en las que se ejecutaría un salto parabólico. El alcance que se tiene en altura y longitud suponiendo que el ángulo es el mismo constante de 45° para todos las velocidades, se tiene lo siguiente:

$$y = (\tan\theta_0)x - \left(\frac{g}{2v_0^2 \cos^2\theta_0}\right)x^2$$

$$v = \{4.1m/s, 5.5m/s, 6.9m/s, 8.3m/s, 9.7m/s\}$$

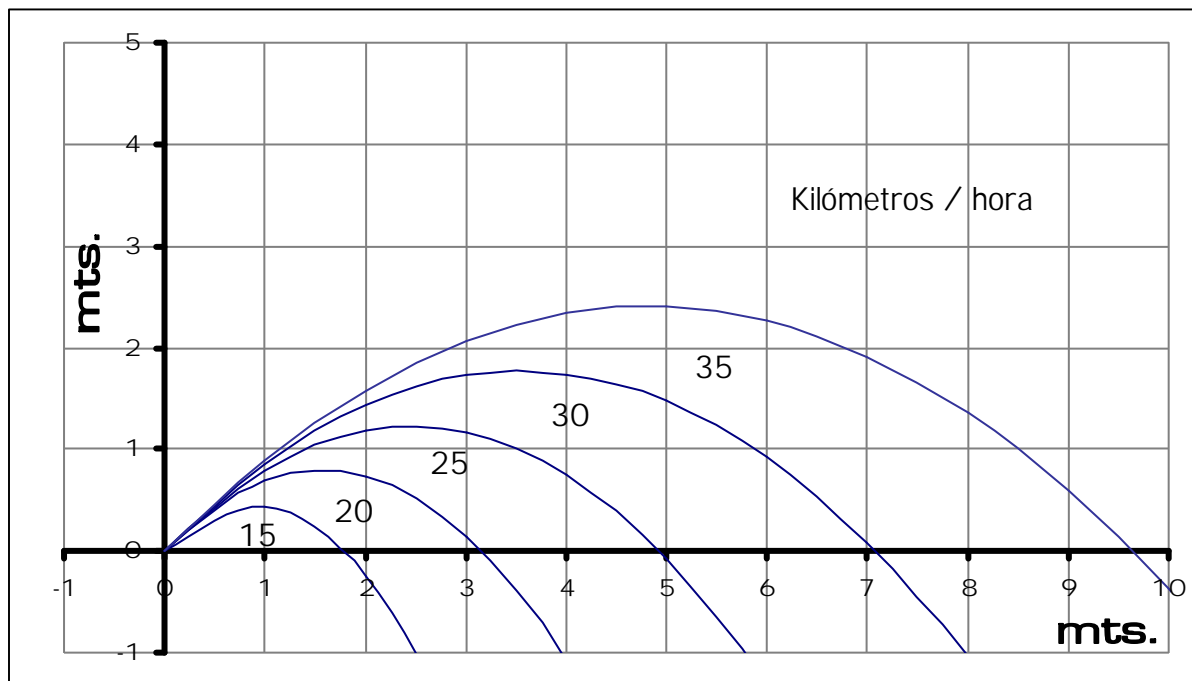
$$\theta = 45^\circ$$

$$g = 9.8/seg^2$$

$$0 \leq x \leq 10$$

$$x \in Z$$

Velocidad Km/hora	15	20	25	30	35
Velocidad m/s	4.16666667	5.55555556	6.94444444	8.33333333	9.72222222
grados inclinación	45	45	45	45	45
gravedad m/s ²	9.8	9.8	9.8	9.8	9.8
x	y1	y2	y3	y4	y5
0	0	0	0	0	0
1	0.43552	0.68248	0.7967872	0.85888	0.89632
2	-0.25792	0.72992	1.1871488	1.43552	1.58528
3	-2.08032	0.14232	1.1710848	1.72992	2.06688
4	-5.03168	-1.08032	0.7485952	1.74208	2.34112
5	-9.112	-2.938	-0.08032	1.472	2.408
6	-14.32128	-5.43072	-1.3156608	0.91968	2.26752
7	-20.65952	-8.55848	-2.9574272	0.08512	1.91968
8	-28.12672	-12.32128	-5.0056192	-1.03168	1.36448
9	-36.72288	-16.71912	-7.4602368	-2.43072	0.60192
10	-46.448	-21.752	-10.32128	-4.112	-0.368



2.2.3 Cálculo de longitud máxima

Como se ha podido observar hasta el momento, hay dos formas de lograr un máximo en la altura del salto parabólico, una es incrementando la velocidad de la carrera, y la otra es cambiando el grado de inclinación al momento de despegar del suelo. En el supuesto de que un atleta deja el suelo a un ángulo de 20° respecto al eje de las Xs y a una rapidez exagerada de 11 m/s . Su alcance horizontal se describe -tomando como base la ecuación 2.2 ya antes mencionada- por la siguiente ecuación:

$$x = (v_0 \cos\theta_0)t \qquad x = (11 \text{ m/s})(\cos 20^\circ)t$$

El valor de x se puede encontrar si se conoce t_1 , el tiempo de salto. Se puede utilizar t tomando como base la ecuación 2.3, utilizando la siguiente expresión:

$$v_y = v_0 \text{sen}\theta_0 - gt \qquad v_y = v_0 \text{sen}\theta - g\Delta t$$

Observando que en lo más alto del salto la componente de la vertical de la velocidad v_y es cero:

$$v_y = v_0 \text{sen}\theta_0 - gt = 0$$

$$v_0 = 11 \text{ m/s}$$

$$\theta_0 = 20^\circ$$

$$g = 9.8 \text{ m/s}^2$$

$$v_y = v_0 \text{sen}\theta_0 - gt = 0$$

$$v_y = (11 \text{ m/s})\text{sen}20^\circ - (9.8 \text{ m/s}^2)t_1 = 0$$

$$v_y = 0$$

$$(11 \text{ m/s})\text{sen}20^\circ - (9.8 \text{ m/s}^2)t_1 = 0$$

$$t_1 = \frac{-(11 \text{ m/s})\text{sen}20^\circ}{-(9.8 \text{ m/s}^2)}$$

$$t_1 = \frac{(11 \text{ m/s})(0.342)}{(9.8 \text{ m/s}^2)}$$

$$t_1 = \frac{3.762 \text{ m/s}}{9.8 \text{ m/s}^2}$$

$$t_1 = \frac{\frac{3.762 \text{ m}}{s}}{\frac{9.8 \text{ m}}{s^2}}$$

$$t_1 = \frac{3.762 \text{ ms}^2}{9.8 \text{ ms}}$$

$$t_1 = \frac{3.762 \text{ s}}{9.8}$$

$$t_1 = 0.384 \text{ s}$$

El valor t_1 es el intervalo de tiempo que tarda en alcanzar la altura máxima el salto. Por la simetría en el movimiento vertical, después de que el atleta regresa al suelo transcurre un intervalo de tiempo idéntico. Entonces el tiempo total t en el aire es:

$$t = 2t_1$$

$$t = 2(0.384s)$$

$$t = 0.768s$$

Sustituyendo esto en la expresión anterior: $x = (v_0 \cos\theta_0)t$ nos da como resultado:

$$x = [(11m/s)(\cos 20^\circ)] \cdot t$$

$$x = [(11m/s)(\cos 20^\circ)] \cdot (0.768s)$$

$$x = [(11m/s)(0.93)] \cdot (0.768s)$$

$$x = [10.33m/s] \cdot (0.768s)$$

$$x = \left[\frac{10.33m}{s} \right] \cdot (0.768s)$$

$$x = \left[\frac{7.94ms}{s} \right]$$

$$x = 7.94m$$

∴ La máxima longitud alcanzada para el atleta sería de 7.94 metros

2.2.4 Cálculo de altura máxima

Luego, para determinar la altura máxima para el atleta, se utiliza la ecuación 2.4 mencionada al principio:

$$y = (v_0 \text{sen}\theta_0)t - \frac{1}{2}gt^2$$

Se sabe que la mitad del tiempo de vuelo es el tiempo en el que se alcanza el punto más alto de la vertical, entonces se sustituye los valores de t_1 y los ya antes mencionados para encontrar y_{\max} :

$$t = t_1$$

$$y_{\max} = (v_0 \text{sen}\theta_0)t_1 - \frac{1}{2}gt_1^2$$

$$y_{\max} = [(11m/s)(\text{sen}20^\circ)] \cdot (0.384s) - \frac{1}{2}(9.8m/s^2)(0.384s)^2$$

$$y_{\max} = [(11m/s)(0.342)] \cdot (0.384s) - \frac{1}{2}(9.8m/s^2)(0.147s^2)$$

$$y_{\max} = [3.762m/s] \cdot (0.384s) - \frac{1}{2} (9.8m/s^2)(0.147s^2)$$

$$y_{\max} = \left[\frac{3.762m}{s} \right] \cdot (0.384s) - \frac{1}{2} \left[\frac{9.8m}{s^2} \right] \cdot (0.147s^2)$$

$$y_{\max} = \left[\frac{1.444ms}{s} \right] - \left[\frac{9.8m}{2s^2} \right] \cdot (0.147s^2)$$

$$y_{\max} = \left[\frac{1.444ms}{s} \right] - \left[\frac{1.440ms^2}{2s^2} \right]$$

$$y_{\max} = [1.444m] - \left[\frac{1.440m}{2} \right]$$

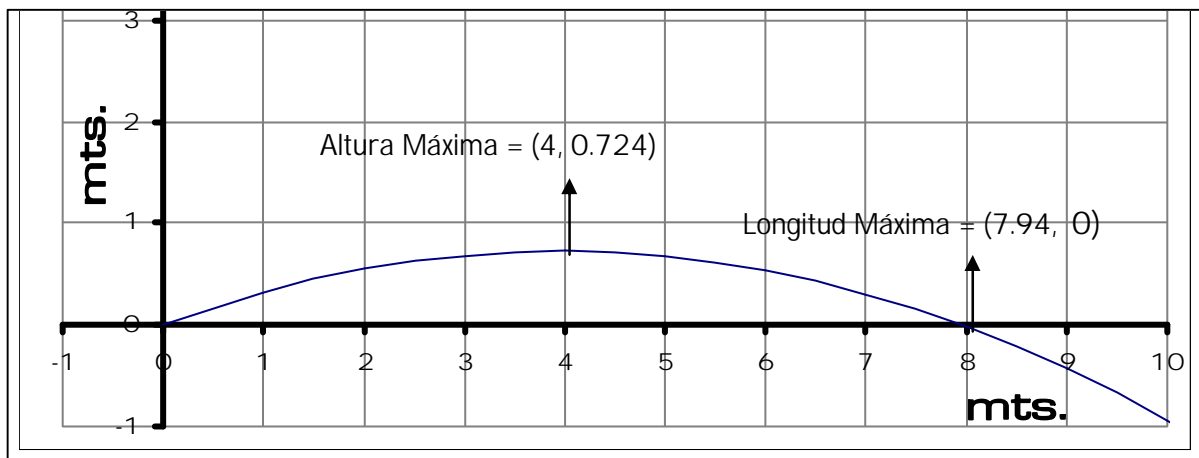
$$y_{\max} = [1.444m] - [0.720]$$

$$y_{\max} = 0.724m$$

∴ Los puntos máximos alcanzados por el atleta serían:

Longitud máxima $y_{\max} = 7.94$ metros

Altura máxima $x_{\max} = 0.724$ metros.



En esta situación se hace una simplificación excesiva de que el salto parabólico es idéntico al del proyectil. Los valores obtenidos son razonables.

2.2.5 Tiempo de vuelo

Las componentes horizontal y vertical del movimiento son independientes una de la otra. Las reglas del movimiento de proyectiles se aplican a los saltos. Una vez que los pies se despegan del suelo, si no se toma en cuenta la resistencia del aire, la única fuerza que actúa sobre el atleta que salta es la gravedad. El "tiempo de vuelo" depende sólo de la componente vertical de la velocidad de despegue. Sucede que la fuerza del salto se puede aumentar un poco mediante la acción de correr, por lo cual "el tiempo de vuelo" para un salto con carrera supera casi siempre al de un salto estacionario. No obstante, una vez que los pies dejan de tocar el suelo sólo la componente vertical de la velocidad de despegue determina el tiempo de vuelo [018].

Cuando se salta hacia arriba, la fuerza del impulso se aplica sólo mientras los pies permanecen en contacto con el suelo. Si la fuerza es mayor, mayor será la rapidez de despegue y mayor la altura que se alcance. Inmediatamente después de que los pies dejan de tocar el suelo, cualquiera que sea la rapidez de ascensión que se alcance, comenzará a disminuir de manera constante y sufriendo una desaceleración de la gravedad de $9.8m/s^2$ ó dependiendo de la latitud y altitud de la tierra donde se encuentre. Si se aterriza en la misma posición en la que se inició el salto, de pie y con las piernas extendidas, el tiempo del ascenso es igual al tiempo de caída. El llamado "tiempo de vuelo" será la suma de los tiempos de ascensión y de caída.

Mientras se está en el aire, al mover los miembros superiores e inferiores del cuerpo, no se podrá modificar el tiempo de vuelo con respecto al centro de gravedad. Si se cae con las piernas dobladas se aumentará ligeramente el tiempo de vuelo con respecto a los pies pero no con respecto al centro de gravedad del atleta, el que siempre describirá una trayectoria parabólica.

Ahora bien, de acuerdo a la ecuación [2.0] el tiempo del ascenso y la altura vertical es:

$$d = \frac{1}{2}gt^2 \text{ Si se tiene la altura } d, \text{ se puede conocer el tiempo de vuelo despejando: } t = \sqrt{\frac{2d}{g}}$$

El récord de altura de salto basquetbolístico lo tiene Spud Webb, quien ha alcanzado 1.25m de altura. Esta altura representa la altura máxima del centro de gravedad del atleta, misma que es la que más importa para determinar la capacidad de salto.

Ahora bien, si se conoce la distancia de la altura alcanzada del atleta $d = 1.25m$, y el valor de la gravedad promedio de $g = 9.8m/s^2$ se obtiene el tiempo del ascenso t_1 que será igual al tiempo del descenso t_2 :

$t_1 = \sqrt{\frac{2(1.25m)}{9.8m/s^2}}$	$t_1 = \sqrt{\frac{2.50ms^2}{9.8m}}$	Tiempo del ascenso $t_1 = 0.505s$
$t_1 = \sqrt{\frac{2.50m}{9.8m/s^2}}$	$t_1 = \sqrt{0.255s^2}$	Tiempo del descenso $t_2 = 0.505s$
$t_1 = \sqrt{\frac{2.50m}{\frac{1}{9.8m/s^2}}}$	$t_1 = 0.505s$	Tiempo de vuelo $t = t_1 + t_2$
		$\therefore t = 1 \text{ seg}$

El tiempo de vuelo del atleta es de 1 segundo.

*“Ahora bien,
de la existencia del Dios perfectísimo,
deriva inmediatamente la confianza
que yo puedo tener en que las cosas que están en el mundo,
entre ellas mi cuerpo,
y el mundo mismo,
como creadas por El,
son ciertas
y en mi contacto con ellas
no puedo engañarme.”*

RENE DESCARTES

3.1 Conceptos de biomecánica

Biomecánica: Estudia los movimientos del hombre y del animal desde el punto de vista de las leyes mecánicas [019]. El concepto de biomecánica (combinación de las palabras biología y mecánica) tiene que ver con los principios y métodos de la mecánica aplicados al estudio de la estructura y función de sistemas biológicos [039]. La mecánica es una rama de la física y la física tiene apoyo en lo matemático. ¿Puede la fuerza muscular desarrollada en los movimientos, ser calculada matemáticamente? En absoluto, aunque es necesario hacer las siguientes aclaraciones:

- Las partes que forman el organismo no son homogéneas, pues están formadas de tejidos de diversa calidad.
- Solamente en forma aproximada se pueden comparar con cuerpos geométricos.
- Las propiedades físicas de un mismo tejido varían en el mismo movimiento, por ejemplo cuando un músculo tiene distinta tensión según el ángulo de su inserción o aplicación [027].

La Biomecánica Deportiva es una disciplina científica que estudia el movimiento humano y la técnica del deportista considerando los principios básicos de la mecánica y las características del aparato locomotor. Este término así como el de kinesiología, se han utilizado para darle nombre a los estudios del movimiento del cuerpo humano. El objetivo básico es lograr que el deportista mejore su técnica deportiva y de esta forma, elevar su rendimiento en la competición [037].

“El cuerpo humano se puede entender como un sistema biológico sujeto a las leyes físicas, y como tal ser tratado. Si se considera que un movimiento deportivo implica movimiento, se puede tratar utilizando las leyes de la mecánica. Así, se puede decir que existe movimiento cuando la totalidad o alguno de los segmentos corporales ocupan una posición diferente en tiempos diferentes como consecuencia de la aplicación de unas fuerzas. Estas variables, que se derivan de la concepción mecánica del movimiento humano, han hecho que se desarrolle la Biomecánica de la Actividad Física y del Deporte.

Para el análisis biomecánico se utiliza básicamente la dimensión newtoniana de la mecánica. Perspectiva desde la cual el movimiento se entiende a partir de un sistema de referencias asociado a la tierra y las partículas poseen posiciones y velocidades definidas no comparables a la velocidad de la luz. Actualmente, tanto la mecánica relativista como la cuántica tienen poca relevancia en el análisis biomecánico del movimiento humano, especialmente en relación a los desplazamientos que el hombre realiza en el espacio.

La mecánica clásica, en interés propio, se subdivide en estática, cinemática y dinámica, según se dedique a estudiar los cuerpos en equilibrio, describir el movimiento o expresar las causas que lo producen, respectivamente. Así, la estática es la rama de la mecánica que trata las fuerzas que actúan sobre los cuerpos en equilibrio y donde no existe movimiento aparente. La Cinemática es una parte de la mecánica que estudia el movimiento de una partícula o sistemas de partículas sin considerar ni hacer referencia a las fuerzas que lo producen. Es el estudio de la geometría en movimiento, se dedica a su descripción, muy similar al concepto de “técnica” utilizado en el ámbito deportivo. La Dinámica se considera como la rama de la mecánica que se encarga de estudiar las causas físicas del movimiento, es decir, la fuerza.

3.1.1 Definiciones

Momento: Unidad de medida del efecto de rotación que una fuerza produce al ser aplicada sobre un cuerpo rígido. $M = F \cdot d$, en donde:

M = Momento

F = Fuerza ejercida

d = Distancia al eje de giro

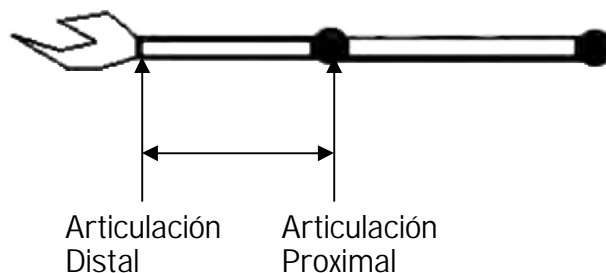
Newton: Unidad de medida de fuerza que (símbolo N) es equivalente a la unidad de momento dividida entre la unidad de tiempo. Fuerza que comunica a un cuerpo de 1 kilogramo de masa una aceleración de 1 metro por segundo al cuadrado.

$$N = \frac{\text{kg} \cdot \text{m}}{\text{s}^2}$$

Proximal: Es el término biomecánico usado para referirse a la articulación inicial del segmento del que se trate. Donde se inicial el segmento,

Distal: Es el término biomecánico usado para referirse a la articulación final del segmento del que se trate. Donde finaliza el segmento.

Por ejemplo, en el supuesto de que se esté tratando el segmento antebrazo, para referirse a las demás articulaciones que siguen al brazo, éstas se nombran como sigue:

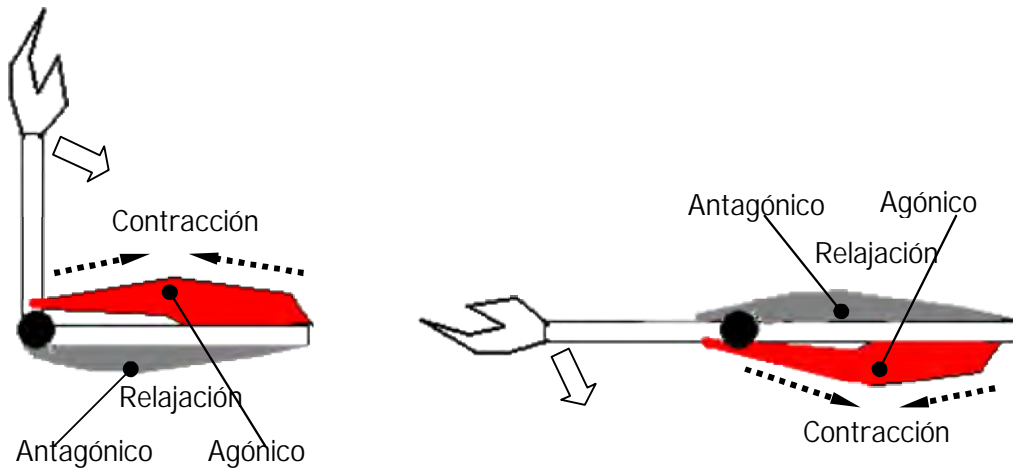


Músculo: Órgano formado por fibras excitables y contráctiles que sirve para producir el movimiento en el hombre y los animales [O23]. Los músculos poseen propiedad contráctil, es decir la propiedad de contraerse, no poseen empuje sino contracción de sus extremos, si se observa un movimiento en el cuerpo que aparentemente sea un empuje, en realidad se debe a la posición de las palancas de los huesos, las que producen un empuje al contraerse los músculos.

Agonista: Músculo que al contraerse produce un movimiento de rotación en una determinada articulación.

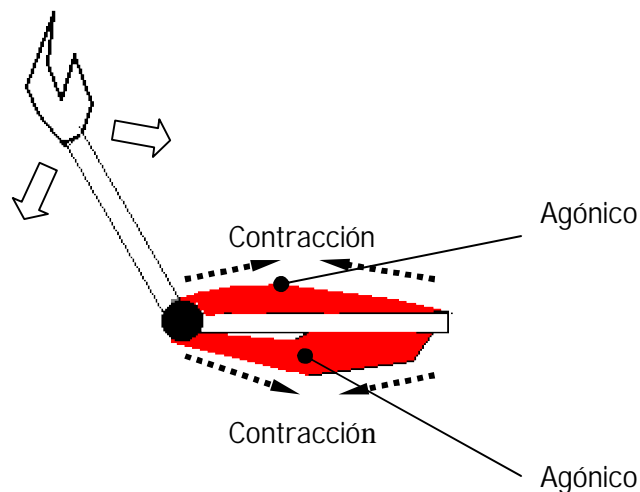
Antagonista: Músculo colocado en posición opuesta al músculo agonista que permanece relajado mientras aquel se contrae.

Por ejemplo: Si un músculo al contraerse flexiona (dobla) la pierna, otro músculo también al contraerse puede extenderla (desdoblarla). El músculo que actúa es el músculo agónico, y el músculo que se haya en la posición opuesta y relajado es el músculo antagonista. Recibirá el nombre de agónico y antagonista dependiendo de cual se contraiga y cual se relaje.



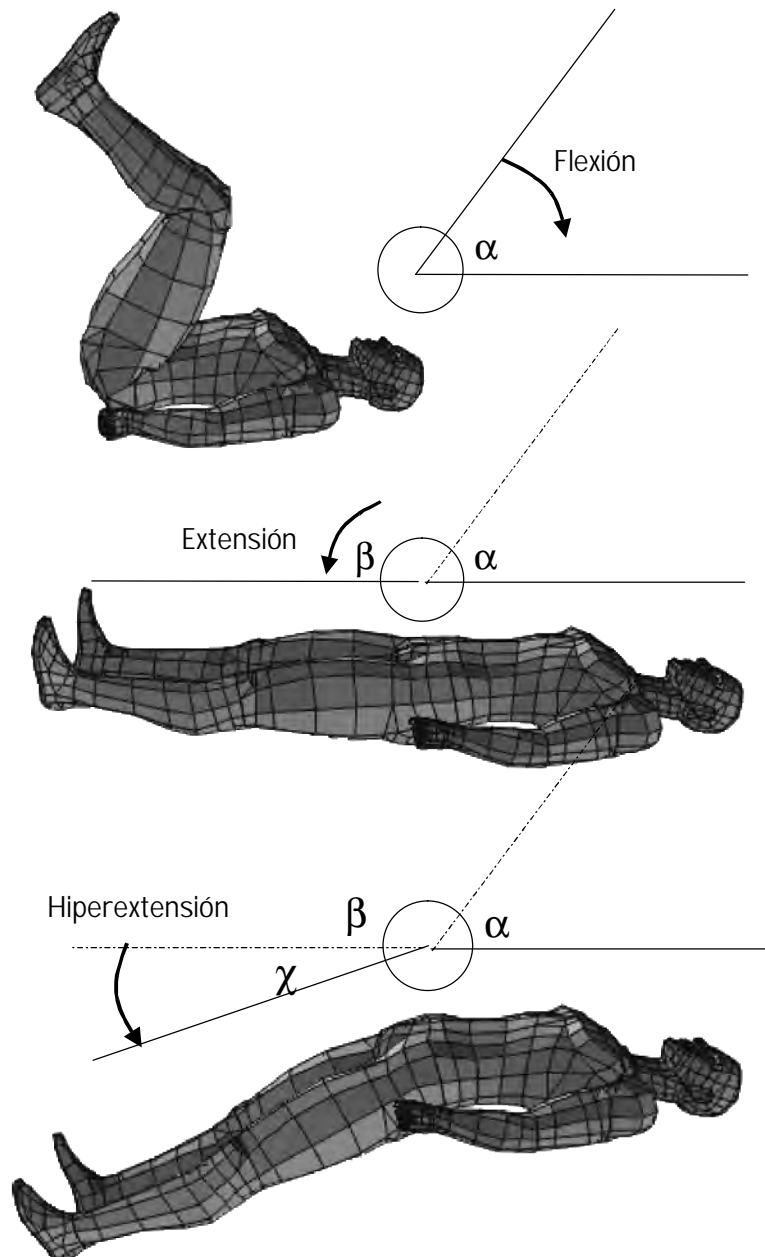
Cuando un músculo agónico se contrae, el antagonista se relaja para que el agónico lo pueda extender. En biomecánica de salto se estudiarán básicamente los músculos como agónicos en los brazos de palanca.

Cocontracción: Cuando dos músculos en posiciones opuestas uno del otro se contraen simultáneamente



3.1.1.1 Flexión y extensión

Normalmente se utilizan en el lenguaje deportivo los términos flexión y extensión. La flexión se refiere básicamente a la disminución del ángulo entre las superficies anteriores de los huesos que articulan, consisten en movimientos que doblan o pliegan dos huesos entre sí. La extensión se refiere a lo contrario de la flexión, todos los movimientos de enderezamiento se denominan de extensión, pues ésta restablece la parte del cuerpo a su posición anatómica partiendo desde la posición flexionada, pero si la extensión continúa más allá de la anatómica, entonces se denomina hiperextensión.



A continuación se enumeran los músculos extensores y flexores principales de las articulaciones para salto vertical y parabólico [002], [027].

Extensores del antebrazo:	Tríceps
Flexores del antebrazo:	Bíceps
Extensores del brazo:	Deltoides (parte trasera del hombro)
Flexores del brazo:	Deltoides (parte delantera del hombro)
Extensores del pie:	Gemelos
Flexores del pie:	Tibial anterior
Extensores de la pierna:	Cuadriceps crural
Flexores de la pierna:	Bíceps crural
Extensores del muslo:	Glúteo mayor
Flexores del muslo:	Psoas-iliaco
Extensores de la columna cervical-dorsal:	Trapezio y dorsal ancho,
Extensores de la columna lumbar:	Masa común y cuadrado lumbar
Flexores de la columna:	Abdominales

Generalizando, se puede decir que el movimiento humano se produce a partir de los momentos de fuerza que generan los músculos al contraerse, los cuales producen una rotación de los segmentos. La eficacia de un músculo, por tanto no depende sólo de la fuerza que genere el músculo al contraerse, es necesario entender que existe otro factor tanto o más importante que determina su eficacia, éste es el brazo de palanca o la distancia perpendicular desde el eje de rotación articular a la línea de aplicación de la fuerza.

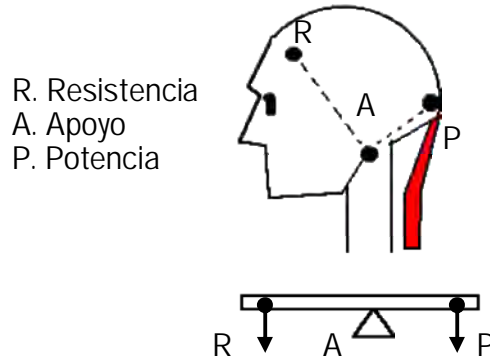
3.1.1.2 Brazos de palanca

A Arquímedes se le atribuye la primer postulación matemática formal del principio de la palanca. "Dadme un punto de apoyo, y moveré al mundo". Ahora bien, en la fisiología del ejercicio, se pueden encontrar tres tipos de palancas fundamentales que son utilizadas para el trabajo biomecánico [027].

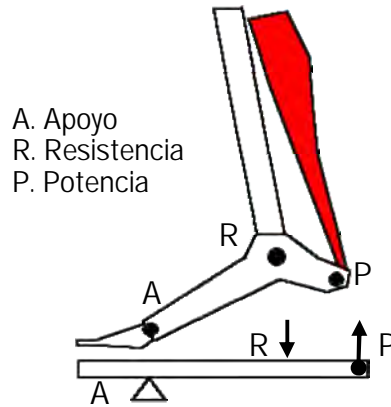
El punto de Apoyo:	Es el centro de giro de la palanca, en este caso se encuentra en el giro de la articulación.
El punto de Potencia:	Es el punto donde se concentra la fuerza que contrarresta al peso de la resistencia. En este caso se encuentra en la inserción del tendón del músculo.
El punto de Resistencia:	Es el punto donde se concentra el peso que se resiste a la potencia. En este caso es el centro de gravedad del miembro del cuerpo que se desea mover.

Los tres tipos de palanca son:

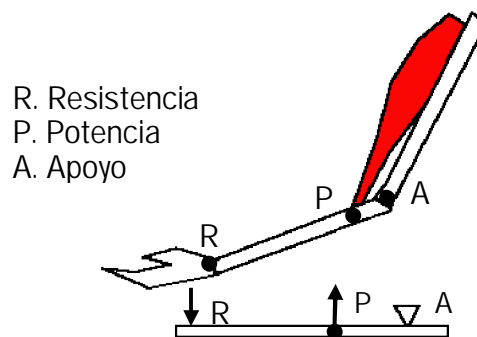
- a) Palanca de 1er género (ej. Extensión de la Cabeza)
 Cuando el Apoyo se encuentra entre la Resistencia y la Potencia



- b) Palanca de 2do género (ej. Extensión del pie)
 Cuando la Resistencia se encuentra entre el Apoyo y la Potencia.



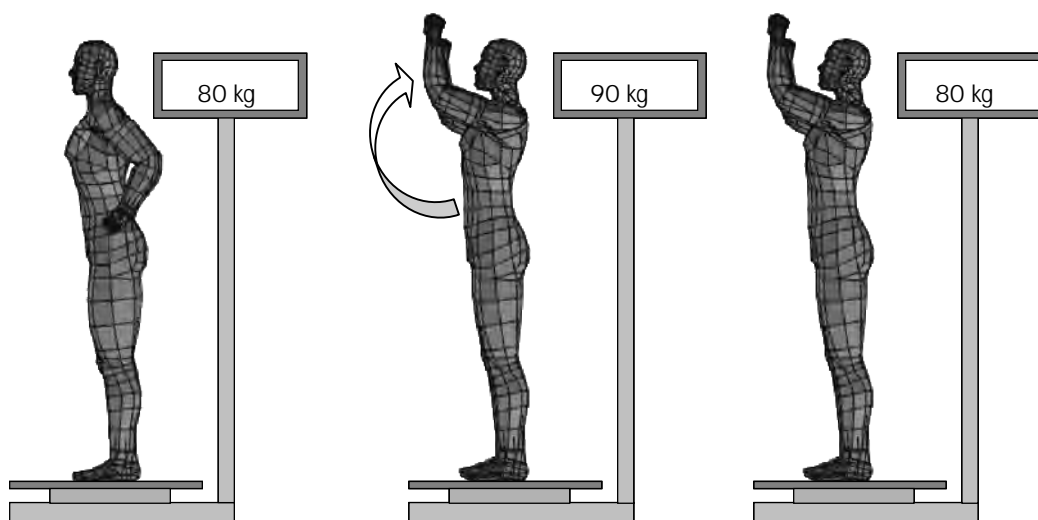
- c) Palanca de 3er género (Ej. Flexión del Brazo)
 Cuando la Potencia se encuentra entre la Resistencia y el Apoyo



3.1.2 Transmisión de fuerzas

“En un sistema coordinado como el del cuerpo humano, las fuerzas se transmiten de un segmento a otro. De forma similar a lo que ocurre cuando se golpean varias bolas de billar que están en contacto mutuo.

Por ejemplo considérese que se está sobre una báscula, como se muestra abajo, si se permanece quieto, la báscula registrará el peso, pero si se aceleran los brazos hacia arriba, aparece una fuerza o tensión de compresión hacia abajo, la cual se transmite de segmento en segmento para sumarse a la fuerza ejercida contra el suelo y, consecuentemente, la báscula incrementará su registro.



Efecto que se produce al transmitir la fuerza generada por la aceleración de los brazos. Dicha fuerza se transmite de segmento en segmento hasta llegar a los pies. La báscula registrará nuevamente el peso original cuando los brazos se detengan.

En este caso, como en otros donde la transmisión de las fuerzas a través de los segmentos es importante para el análisis del movimiento, las fuerzas internas juegan un papel estabilizador muy importante, de tal forma que es realmente difícil que exista habilidad distal sin estabilidad proximal.” (es difícil que exista habilidad de los segmentos distantes del cuerpo sin que haya estabilidad de los segmentos próximos) [015].

3.2 Conceptos de Centro de Gravedad

Frecuentemente se tienden a confundir los conceptos de Centro de Masa y Centro de Gravedad. No son exactamente lo mismo, aunque en la práctica se comporten de manera similar.

“Centro de masa (CM) Es aquel punto de un cuerpo material o de un sistema coordinado que se mueve como si la masa total del sistema se hallara en el citado punto y todas las fuerzas externas fuesen aplicadas al mismo. El movimiento de cualquier objeto o sistema coordinado puede describirse en función del movimiento del centro de masas y considerarse como el movimiento global del sistema, aunque también es necesario conocer el movimiento de las masas parciales que componen el sistema con respecto al centro de masas

Centro de gravedad (CG) se define como un punto fijo de un cuerpo material donde actúa la fuerza gravitatoria resultante. Cuando se somete un cuerpo a la acción de la gravedad, cada partícula de su masa experimenta una atracción hacia la tierra, lo que se ha denominado fuerza gravitatoria. El origen de la resultante de todas las fuerzas que actúan sobre ese cuerpo puede considerarse como el centro de gravedad de dicho cuerpo” [O15].

Se puede decir que el término centro de gravedad (CG) y el término centro de masas (CM) tienen diferencias conceptuales, “el primero se refiere a un punto que es origen de la fuerza resultante con la que atrae la gravedad a una masa, y el segundo se refiere a un punto relacionado exclusivamente a la masas” [O15]. Para diferenciarlos más aún se puede poner por ejemplo un cuerpo simétrico como un edificio suficientemente alto, en donde su centro de masa se sitúa en el centro geométrico del cuerpo, si la altura es considerable, el centro de gravedad se encontrará milímetros mas abajo del centro de masa debido a que la fuerza de gravedad es mayor en los pisos de abajo que en los pisos de arriba por encontrarse más lejos del suelo de la tierra. Por lo tanto, considerando que la gravedad es constante en todos los puntos del sistema de estudio, se usará el concepto de centro de gravedad (CG) por el uso genérico en las publicaciones sobre biomecánica deportiva y también por utilizarse al momento de hacer los cálculos, el momento que ejercen las fuerzas gravitatorias [O18].

“Las propiedades más significativas que pueden ayudarnos a entender el movimiento de un sistema, a partir del movimiento del centro de gravedad, son las siguientes:

- a) El análisis del movimiento del centro de gravedad equivale a estudiar el movimiento de una partícula en la cual se supone que se concentra toda la masa del sistema de estudio. Por lo tanto, es posible estudiar cualquier movimiento rectilíneo o circular del sistema a partir del movimiento del centro de gravedad.
- b) Cuando un sistema coordinado cualquiera se desplaza a través del aire, sin que actúen más fuerzas sobre él que el de gravedad, su centro de gravedad describirá una trayectoria parabólica igual a la que recorrería una partícula aislada, como en el caso de un gimnasta que realiza un salto mortal. Se puede observar que a pesar de los movimientos que realiza su cuerpo en el aire, su centro de gravedad constituye el centro de giro del sistema.
- c) Todas las fuerzas gravitatorias que actúan sobre las partículas de un cuerpo producen un momento resultante nulo, de tal forma que si se quiere localizar el centro de gravedad de un cuerpo, sólo habrá que balancear dicho cuerpo sobre un pivote. Si se balancea un cuerpo homogéneo, la balanza se equilibrará en un punto medio o centro geométrico del cuerpo, el cual coincidirá con el centro de gravedad.

- d) Cuando la dirección de una fuerza externa aplicada al sistema de estudio pasa por su centro de gravedad, esta fuerza sólo modificará el movimiento de traslación del cuerpo, sin modificar en absoluto su rotación. Por el contrario, si la dirección de esta fuerza no pasa por el centro de gravedad, la fuerza modificará el movimiento de traslación y el de rotación del cuerpo. Se puede decir que cuando la dirección de una fuerza pasa por el centro de gravedad su momento es nulo.
- e) En los cuerpos homogéneos que poseen un centro de simetría, éste será también el centro de gravedad. En cualquier objeto rígido y homogéneo, su centro de simetría coincidirá con el centro de gravedad.
- f) El centro de gravedad de un cuerpo o sistema coordinado puede estar situado fuera de la superficie del cuerpo. Como en el caso de un salto de altura en donde el atleta dobla el tronco hacia atrás y quedando el CG por debajo de la barra.
- g) Si se conoce el CG y el peso de todos los cuerpos que componen un sistema, se puede determinar el CG total del sistema, y éste será el punto de aplicación de la resultante de todas las fuerzas paralelas del mismo sentido" [015].

3.2.1 Centro de gravedad en el cuerpo humano

3.2.1.1 Cálculo del centro de gravedad en una recta

Sea l una recta coordenada y S un conjunto de n partículas de masas m_1, m_2, \dots, m_n localizadas en los puntos de coordenadas x_1, x_2, \dots, x_n respectivamente.

Sea $m = \sum_{i=1}^n m_i$ la masa total

(i) El momento de S con respecto al origen es $M_x = \sum_{i=1}^n m_i x_i$

(ii) El centro de gravedad de S es el punto con coordenada \bar{x} , tal que $m\bar{x} = M_x$

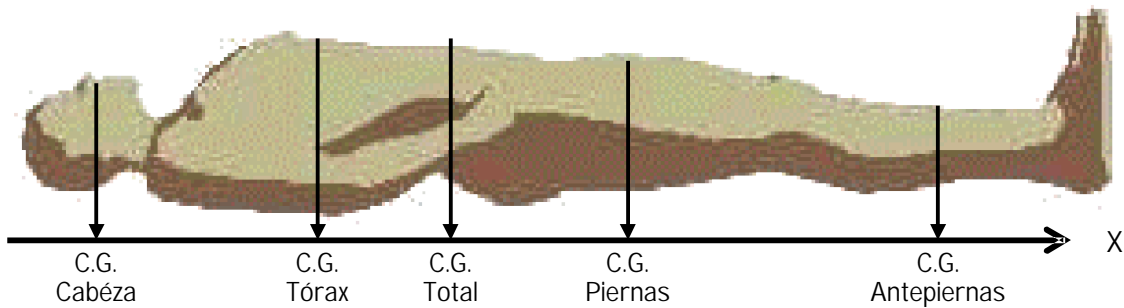
al sustituir M_x queda: $m\bar{x} = \sum_{i=1}^n m_i x_i$ al despejarse... $\bar{x} = \frac{\sum_{i=1}^n m_i x_i}{m}$

finalmente sustituyendo el valor de m quedará la fórmula:

$$\bar{x} = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}$$

Entrando a detalle en la fórmula mostrada los números $m_1 x_1, m_2 x_2, \dots, m_n x_n$ se denominan momentos de las masas. Un centro de gravedad (X, Y) se determina sumando los momentos de las masas y dividiéndola entre la masa total.

En el cuerpo humano los centros de gravedad se concentran en diferentes sitios, principalmente cuando este permanece tendido. Para determinar el centro de gravedad en un cuerpo tendido en el eje de las X_s , se toman los centros de gravedad de cada segmento:



3.2.1.2 Cálculo del centro de gravedad en un plano

Sea S un conjunto de n partículas de masas m_1, m_2, \dots, m_n ubicadas en los puntos $P_1(x_1, y_1), P_2(x_2, y_2), \dots, P_n(x_n, y_n)$ respectivamente de un plano coordenado,

y sea $m = \sum_{i=1}^n m_i$ la masa total.

(i) Los momentos M_x y M_y de S con respecto al eje X y al eje Y son

$$M_x = \sum_{i=1}^n m_i x_i \quad \text{y} \quad M_y = \sum_{i=1}^n m_i y_i$$

(ii) El centro de gravedad de S es el punto $P(\bar{x}, \bar{y})$ dado por:

$$m\bar{x} = M_x \quad \text{y} \quad m\bar{y} = M_y$$

Sustituyendo M_y , M_x se tiene lo siguiente:

$$m\bar{x} = \sum_{i=1}^n m_i x_i \quad \text{y} \quad m\bar{y} = \sum_{i=1}^n m_i y_i$$

Despejando \bar{x}, \bar{y}

$$\bar{x} = \frac{\sum_{i=1}^n m_i x_i}{m}$$

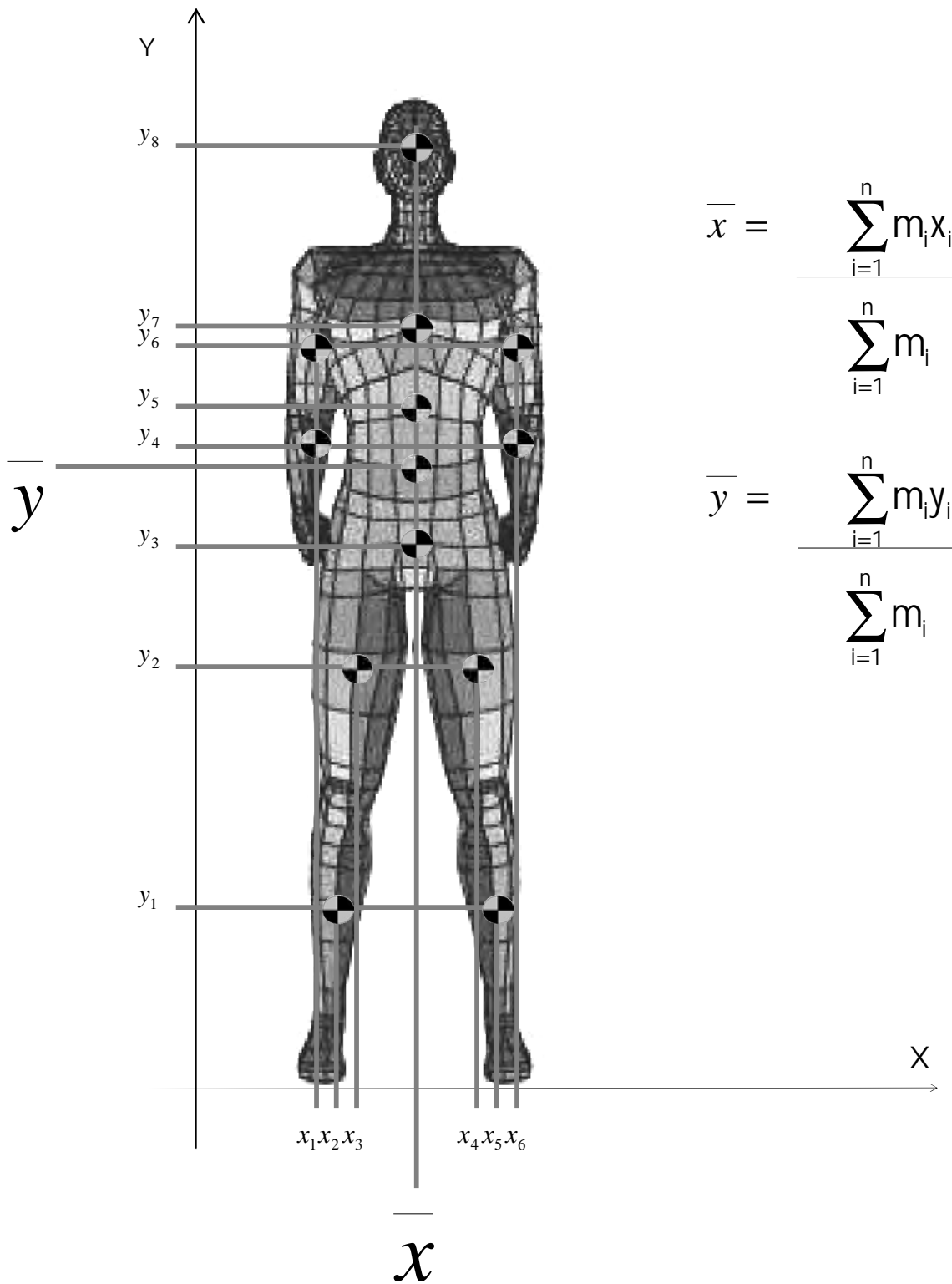
$$\bar{y} = \frac{\sum_{i=1}^n m_i y_i}{m}$$

Sustituyendo a la masa total de m se tiene:

$$\bar{x} = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}$$

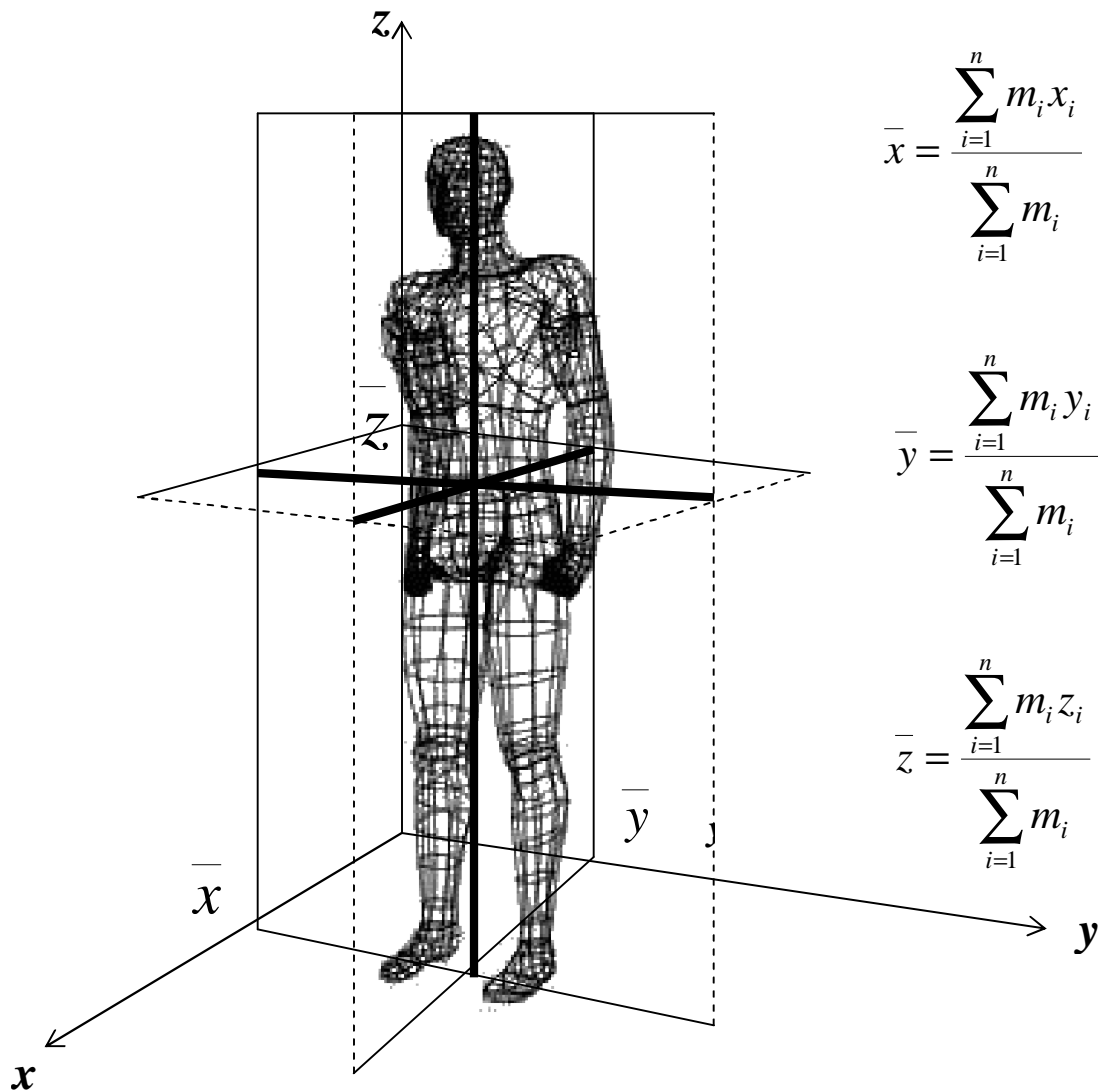
$$\bar{y} = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}$$

Los principales centros se pueden encontrar distribuidos a lo largo de todo el cuerpo, como en la intersección de todas las líneas de las X's y las Y's de acuerdo a la gráfica siguiente:



3.2.1.3 Cálculo del centro de gravedad en el espacio

Al igual que el centro de gravedad en un plano, se calcula de la misma manera para el eje de las Z's. Este centro de gravedad como ya se ha comentado constituye el centro de todas las fuerzas que controlan el movimiento del atleta. Así pues, si el atleta hace un movimiento en el espacio como lo que hace un gimnasta, para cualquier dirección este centro de gravedad será el centro de giro y origen de la traslación del sistema corporal.

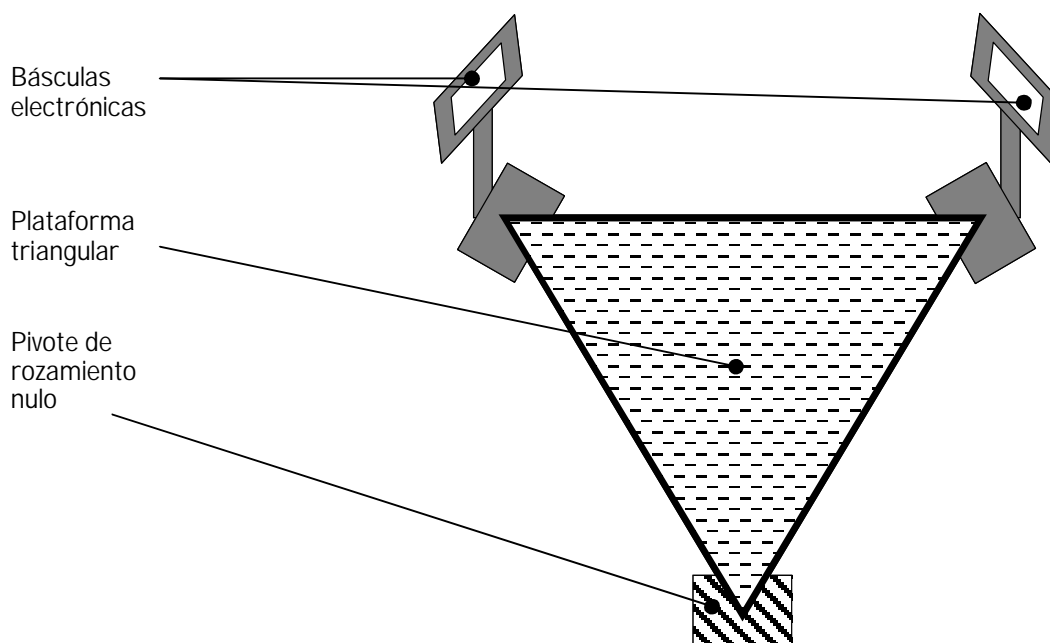


En los casos anteriores para reconocer la posición de los centros de gravedad de todos los miembros del cuerpo, se tendría que fraccionarlos independientemente para conocer el sitio de cada uno de ellos, para así poder hacer el cálculo, como lo que sucede cuando se hacen pruebas con cuerpos sin vida. Sin embargo también se puede obtener la posición del sitio de CG mediante otros métodos para hacer pruebas con atletas vivos. Estos son: La plataforma rectangular de momentos y la plataforma equilátera.

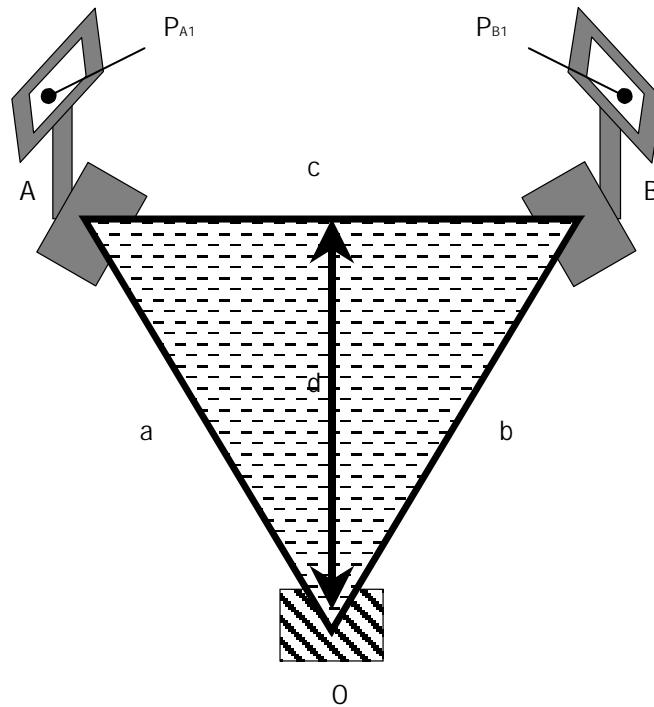
3.2.1.4 La plataforma equilátera

Se llama "Plataforma equilátera de Basler". Permite determinar el centro de gravedad del atleta en dos dimensiones. El sistema consiste en una plataforma equilátera homogénea de lados a , b y c de altura d , la cual descansa, por uno de sus vértices (o), sobre un pivote de rozamiento que se considerase nulo, un segundo vértice (P_{A1}) y un tercer vértice que descansa sobre una segunda balanza que registra la fuerza (P_{B1})..

Sobre la citada plataforma, el atleta al cual se pretende determinar el centro de gravedad, adopta una posición similar a la que adoptaría durante el ejercicio que se pretende analizar. Conociendo el peso del atleta (P) y una segunda lectura de las dos básculas que soportan los vértices A y B con el atleta sobre la plataforma (P_{A2} y P_{B2}) respectivamente, es posible determinar la intersección de dos líneas de aplicación, o punto de intersección, que correspondería a la posición del centro de gravedad.

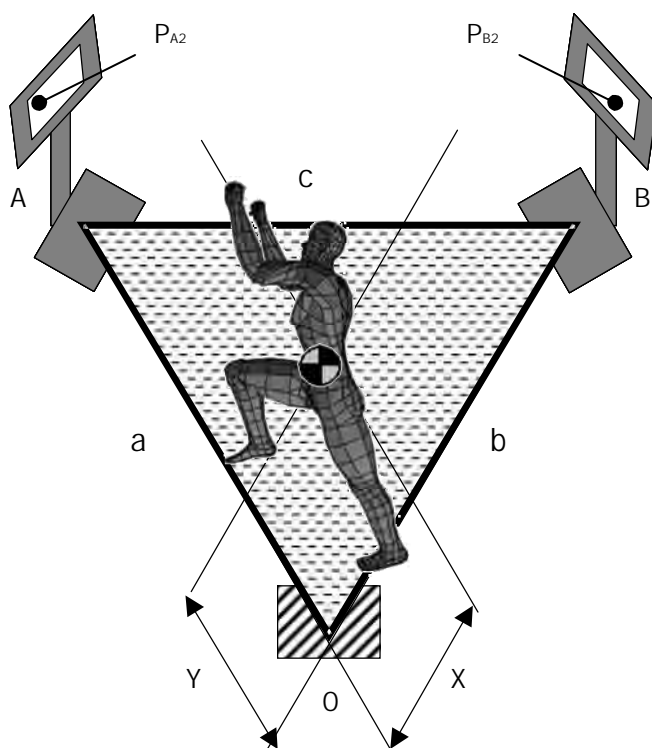


Elementos de una plataforma equilátera, vista superficial.



Variables para medir la posición del CG, antes de colocar al atleta.

PA1	Lectura del peso inicial de la báscula A
PA1	Lectura del peso inicial de la báscula B
A	BASCULA A
B	BASCULA B
O	PIVOTE RE ROZAMIENTO NULO
a	Longitud de OA
b	Longitud de OB
c	Longitud AB
d	Altura del triángulo (H)

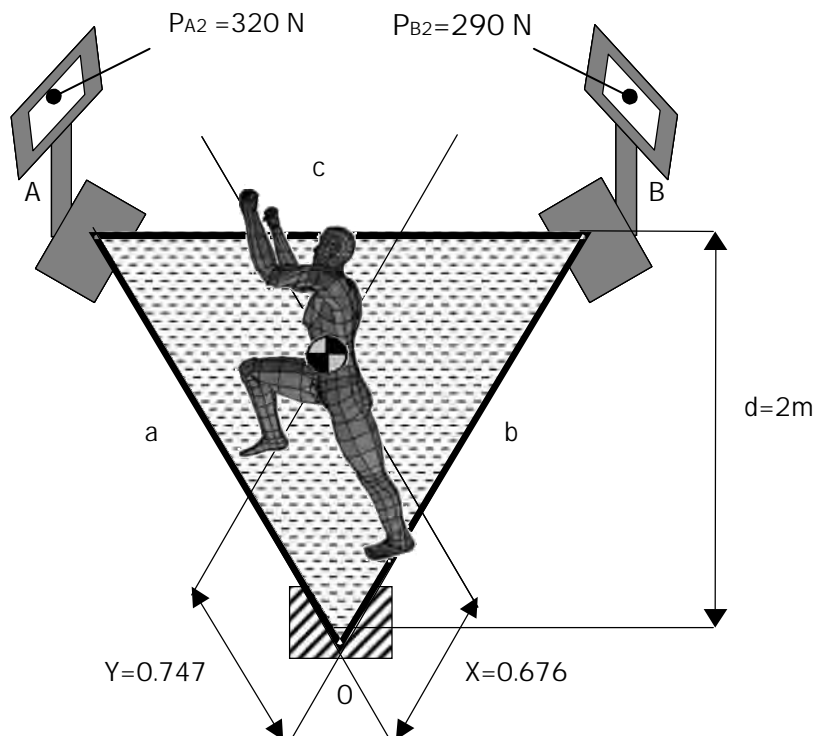


Posición adoptada por un atleta sobre la plataforma equilátera de Basler y las variables a tomar en cuenta para el cálculo del Centro de Gravedad.

Este método se ha utilizado para determinar el centro de gravedad de deportistas en posiciones concretas. El procedimiento solía comenzar con la obtención de una fotografía o la selección de imágenes procedentes de una película cinematográfica, a partir de las cuales se obtenían los puntos articulares y se situaban, a escala real, en la plataforma. Después de realizar las primeras lecturas P_{A1} y P_{B1} , el deportista adoptaba la posición representada en la plataforma por los puntos señalados y se realizaban las segundas lecturas P_{A2} y P_{B2} , lo que permitía calcular el centro de gravedad del deportista en la posición adoptada durante la ejecución deportiva [015].

Aunque este método es bastante eficaz y exacto, el principal inconveniente se desprende de su carácter experimental, ya que es necesario disponer del deportista durante todo el proceso de cálculo,

Por ejemplo, utilizando una plataforma triangular equilátera de 2 m de altura, se puede determinar la localización del centro de gravedad de un salto basquetbolístico en donde el atleta pesa $840N$, en el instante de perder contacto con el suelo. En primer lugar se anota la lectura de las básculas antes de subirse el atleta $P_{A1} = 6N$ y $P_{B1} = 6N$ después de señalar la posición de sus puntos articulares a escala, procedentes de una imagen real, el atleta adopta dicha posición sobre la plataforma y se anotan de nuevo las lecturas que tienen las básculas $P_{A2} = 320N$ y $P_{B2} = 290N$, la distancia desde el lado a se podría calcular utilizando la siguiente expresión:



Notaciones para el cálculo del CG de un saltador.

$$X = \frac{(P_{B2} - P_{B1}) \cdot d}{p} \text{ en donde los valores quedarían: } X = \frac{(290N - 6N) \cdot 2}{840} = 0.676m$$

$$Y = \frac{(P_{A2} - P_{A1}) \cdot d}{p} \text{ en donde los valores quedarían: } Y = \frac{(320N - 6N) \cdot 2}{840} = 0.747m$$

3.2.1.5 El método segmentario

Al hacer referencia a las propiedades del centro de gravedad, se puso de manifiesto que cuando se conoce el centro de gravedad y el peso de los segmentos que componen un sistema, se puede determinar el centro de gravedad total del mismo, ya que éste será el punto de aplicación de la resultante de todas las fuerzas paralelas que actúan en el sistema.

Para la determinación de un segmento corporal son imprescindibles, como mínimo, dos puntos que definen su eje longitudinal, que habitualmente se corresponden con los extremos de dicho eje y se les conoce como punto proximal (donde se inicia el segmento) y punto distal (donde finaliza el segmento).

Cronológicamente, los primeros datos que se obtuvieron, en relación a los parámetros inerciales del cuerpo humano, procedían del desmembramiento de cuerpos sin vida según planos de segmentación que pasaban por los puntos que definen el modelo, coincidentes, con los centros de rotación de las articulaciones más importantes. Después de ser congeladas, estas muestras eran equilibradas mediante plataformas de momentos para obtener el centro de gravedad y pesadas para calcular el porcentaje de peso con respecto al peso total del atleta. Dichos datos fueron expresados en forma de promedio y, en algunos casos, mediante ecuaciones de regresión.

A pesar de que los datos de Zatsiosky y Seluyanov están publicados desde el año de 1985, muchos investigadores no los han utilizado, debido a ciertos problemas de localizaciones de centros articulares o puntos que definen el modelo. Un intento de resolver este problema ha sido realizado por Leva (1996), el cual ha ajustado los parámetros inerciales propuestos por Zatsiorsky y Seluyanov a los puntos articulares anatómicos del modelo humano normalizado y más habitual en biomecánica deportiva. En los siguientes cuadros se presentan las adaptaciones llevadas a cabo por de Leva a los parámetros inerciales aportados por Zatsiorsky y Seluyanov (1985).

Finalmente, es necesario comentar los modelos basados en aproximaciones geométricas o modelos matemáticos del cuerpo humano. Dichos modelos representan los segmentos corporales mediante sólidos definidos geoméricamente y capaces de ser descritos matemáticamente. Este es el caso de los modelos matemáticos descritos por Whitsett (1963), Hanavan (1964), Jensen (1978,1986,1993,1994), Hatze (1980), Yeadon (1990), Safarty y Laidin (1993).

Pesos de los segmentos corporales, relativos al peso total del cuerpo (% de 1)		
Segmento	Hombres	Mujeres
Cabeza y cuello	0.0694	0.0668
Tronco	0.4346	0.4257
(Tórax)	0.1596	0.1545
(Abdomen)	0.1633	0.1465
(Pelvis)	0.1117	0.1247
Brazo	0.0271	0.0255
Antebrazo	0.0162	0.0138
Mano	0.0061	0.0056
Muslo	0.1416	0.1478
Pierna	0.0433	0.0481
Pie	0.0137	0.0129

Localización del vector CG de los segmentos corporales, expresado como porcentaje de la distancia total entre dos puntos, a partir del proximal		
Segmento	Hombres	Mujeres
Cabeza y cuello	59.76	58.94
Tronco	44.86	41.51
(Tórax)	29.99	20.77
(Abdomen)	45.02	45.12
(Pelvis)	61.15	49.20
Brazo	57.72	57.54
Antebrazo	45.74	45.59
Mano	79.00	74.74
Muslo	40.95	36.12
Pierna	44.59	44.16
Pie	44.15	40.14

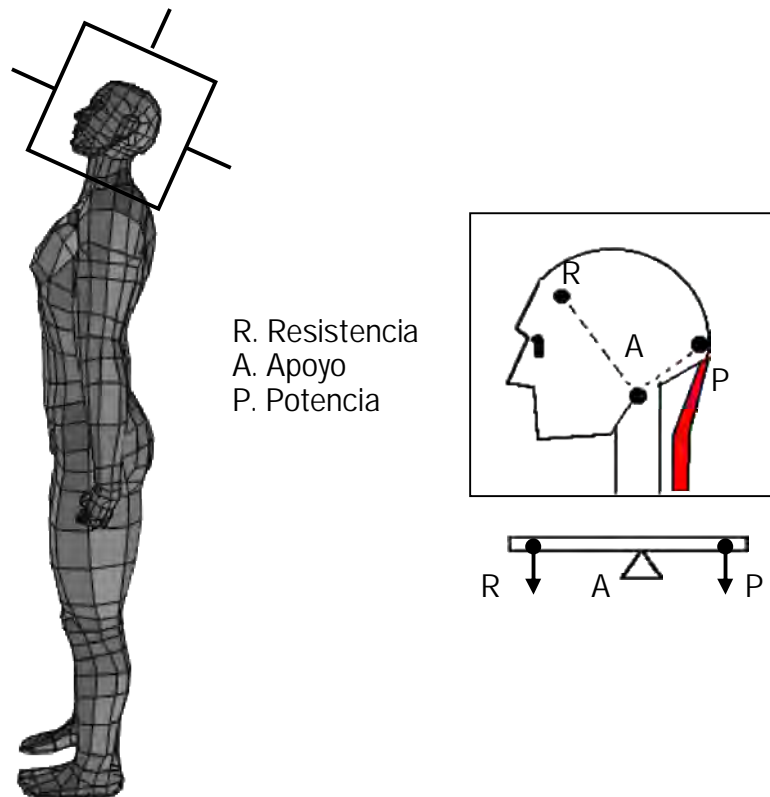
3.3 Brazos de palanca en el salto vertical

En todo el cuerpo existen numerosas series de músculos y palancas que trabajan para un salto vertical, sin embargo se delimitará el estudio a los más fundamentales.

Los huesos actúan como palancas, y las articulaciones son los puntos de apoyo de las palancas. Por definición, la palanca es cualquier barra rígida que tiene la facultad de moverse sobre un punto fijo llamado fulcro o punto de apoyo. Al contraerse el músculo, hace tracción de la palanca ósea en el sitio de la inserción en el hueso. Ello hace que el hueso de inserción se mueva sobre el fulcro articular [002].

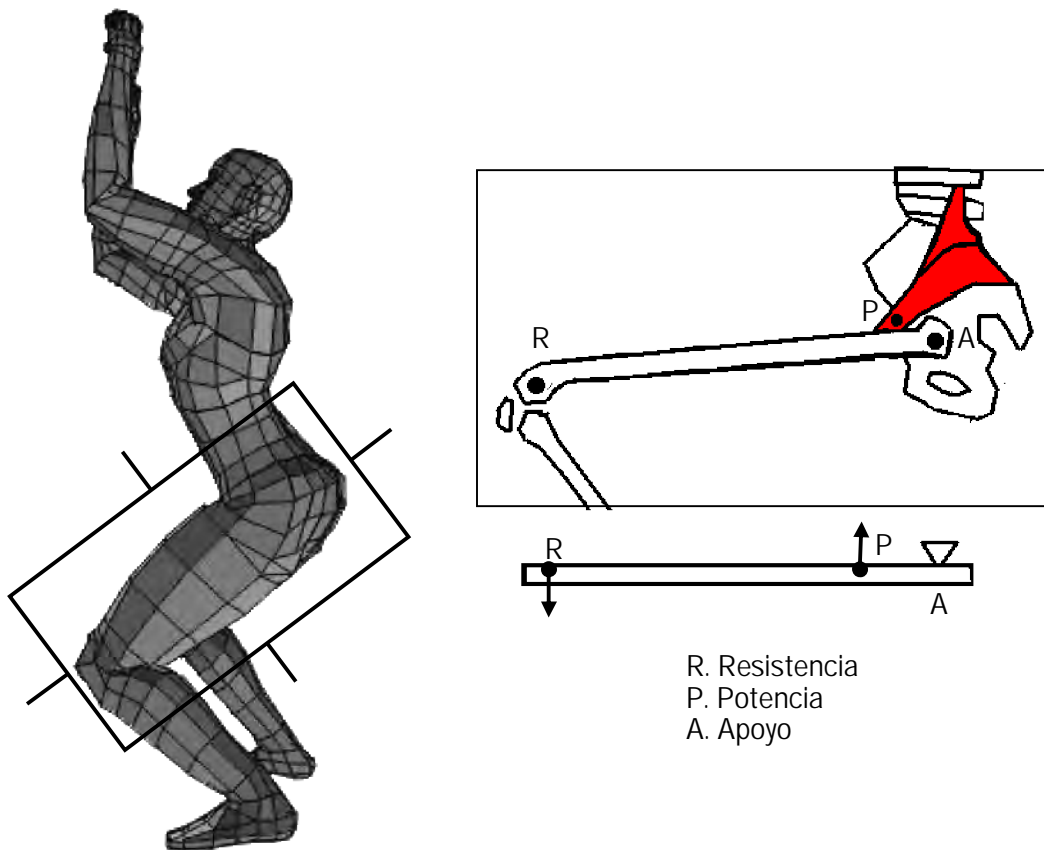
3.3.1 Hiperextensión de la cabeza

El músculo extensor de la articulación del cuello hacia atrás es el trapecio, el cual tira de la nuca hacia la espalda y teniendo como consecuencia el levantamiento de la cabeza por la parte frontal, aquí se aplica una palanca de 1er género en la articulación occipito-atloidea, misma que actúa como apoyo, El peso de la cabeza actúa como resistencia y el trapecio como potencia.



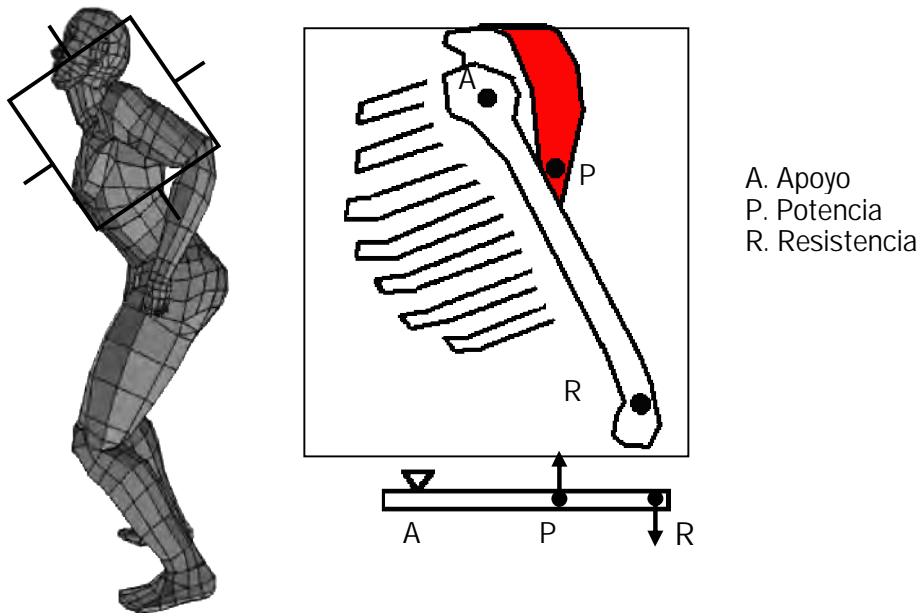
3.3.2 Flexión de las piernas

Los músculos flexores principales de las piernas son el psoas iliaco y el recto interno, los cuales elevan las piernas desde una posición erguida o de pie, teniendo como consecuencia (3ra ley de Newton) la flexión del dorsal cuando el fémur actúa como origen para colocar en posición de salto. Aquí se aplica una palanca de 3er género en la articulación de la cabeza del fémur con la pelvis, misma que actúa como apoyo, El peso de las piernas actúan como el peso de la resistencia y los músculos como potencia



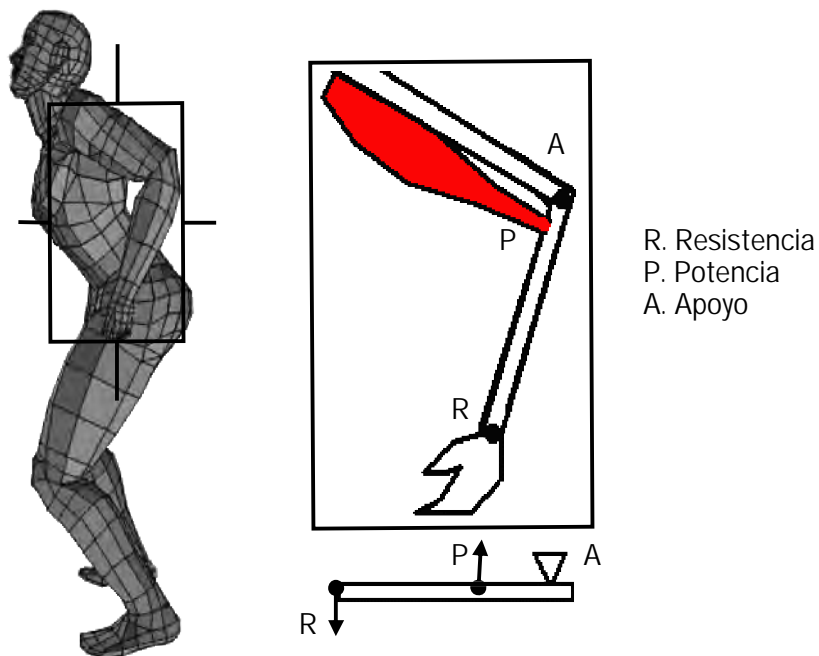
3.3.3 Extensión de los brazos

El músculo extensor del hombro y el brazo hacia la parte posterior es el deltoides, el cual tira del brazo hacia abajo y hacia atrás una vez que el brazo está en reposo y el cuerpo entero está de pie. Aquí se aplica una palanca de 3er género en la articulación del húmero con el omóplato. El extremo superior del húmero actúa como apoyo, el peso del brazo actúa como resistencia y el deltoides como potencia.



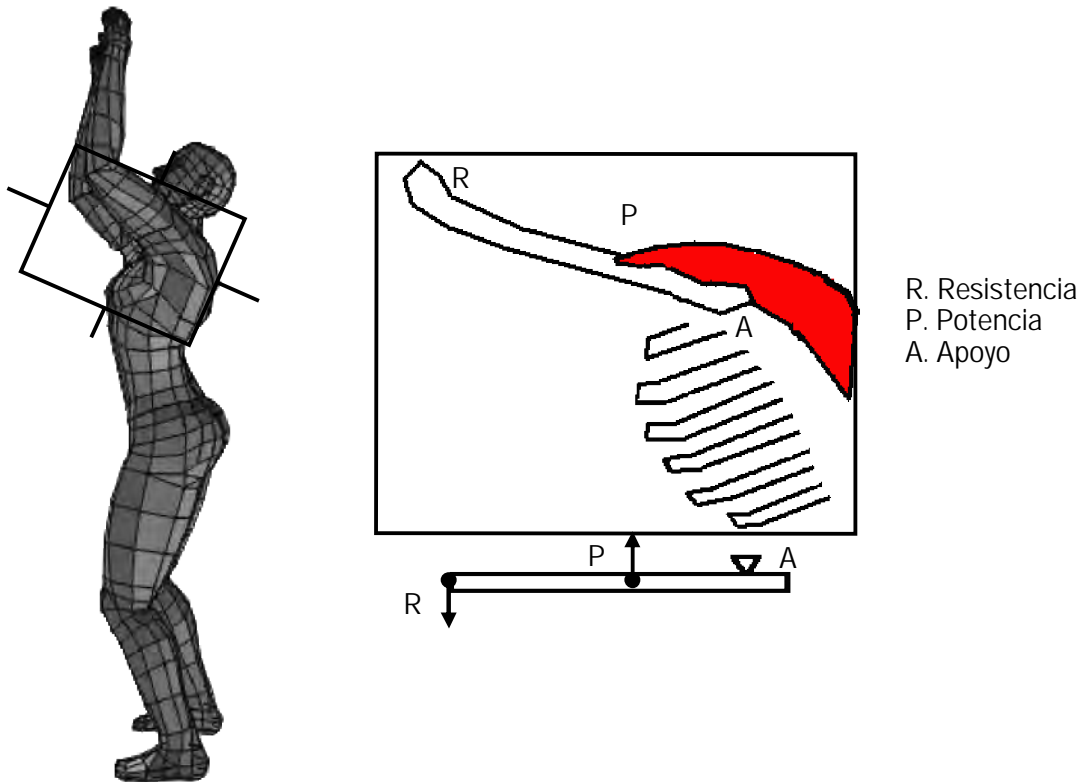
3.3.4 Flexión del antebrazo

El músculo flexor principal de los antebrazos es el bíceps, el que al contraerse tiene como consecuencia la flexión del antebrazo. Aquí se aplica una palanca de 3er género. De acuerdo a la gráfica. El peso del antebrazo como resistencia. El bíceps como potencia y el extremo inferior del húmero articulado con el cúbito y radio o sea el codo, actúa como apoyo.



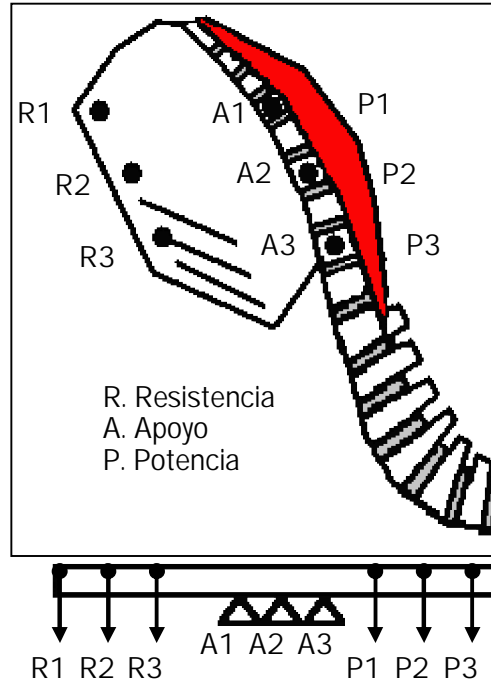
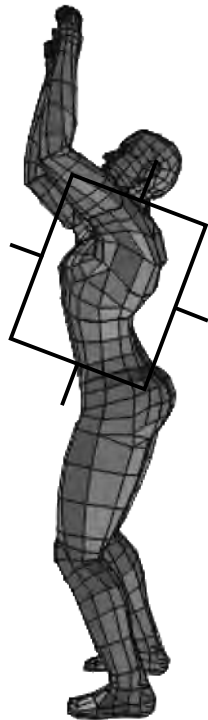
3.3.5 Elevación de los brazos

El músculo flexor principal de los brazos –que también es el extensor- es el deltoides, el que al contraerse por la parte frontal desde una posición de reposo, tiene como consecuencia la elevación de los brazos. Aquí se aplica una palanca de 3er género. De acuerdo a la gráfica, el extremo superior del húmero actúa como resistencia por el peso de los brazos, la articulación del húmero con el omóplato actúa como apoyo (hombro) y el deltoides como potencia.



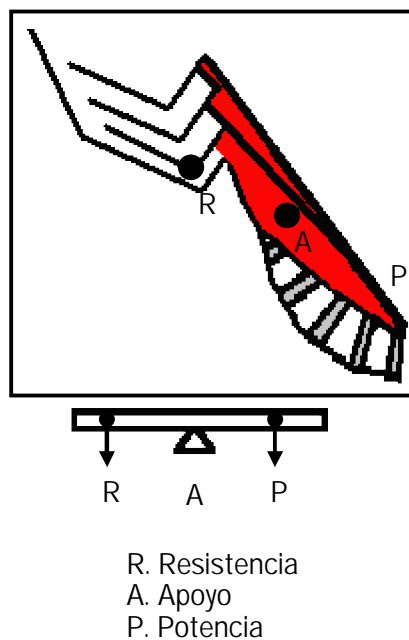
3.3.6 Extensión de la espalda alta

El músculo extensor de la espalda alta es también el trapecio, iliocostal cervical e iliocostal dorsal los cuales tiran de la sección alta de la espalda y base del cuello hacia la parte posterior, traen como consecuencia la elevación de la caja torácica y extensión de las vértebras cervicales y dorsales, aquí se aplica una palanca de 1er género en las articulaciones de las vértebras dorsales, las mismas que actúan como apoyo, el peso de la caja torácica y miembros superiores actúan como resistencia y los músculos citados como potencia.



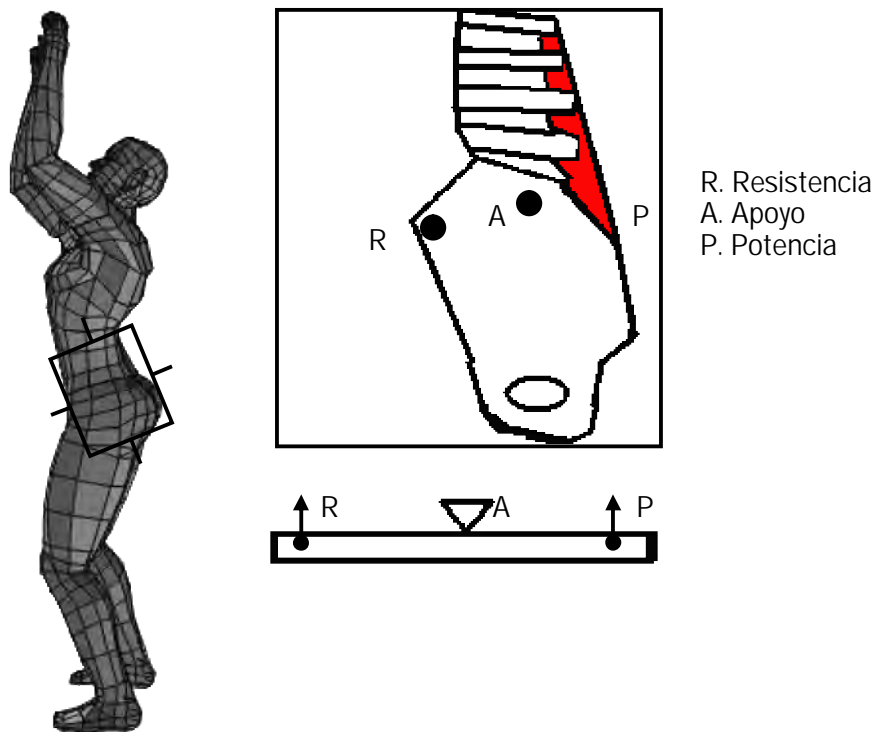
3.3.7 Extensión de la espalda media

Los músculos extensores de la espalda media son: el llamado iliocostal lumbar y el dorsal largo. Conservan la postura erguida del tronco. Aquí se aplica una palanca de 1er género en las articulaciones de las vértebras lumbares, la parte media de la región lumbar como apoyo, la parte superior de la columna y miembros superiores como resistencia y los músculos mencionados como potencia.



3.3.8 Extensión de la pelvis

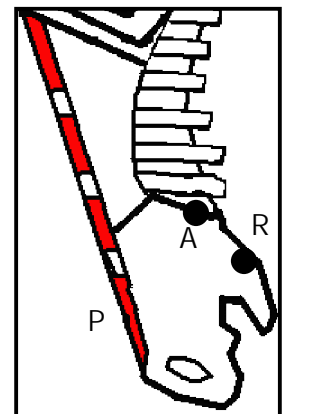
El músculo extensor de la espalda baja es el cuadrado lumbar o cuadrado de los lomos, que al contraerse extiende la columna vertebral tirando de la sección media de la espalda hacia la sección lumbar. Aquí se aplica una palanca de 1er género en la articulación de las últimas vértebras lumbares y la pelvis, la sección articular de la región lumbar actúa como apoyo, la flexión de la pelvis hacia la parte frontal y superior actúa como resistencia y los músculos mencionados como potencia



3.3.9 Flexión de la espalda baja

Por estar a la altura del centro de gravedad total del cuerpo, este es uno de los más importantes, la referencia se hace al músculo flexor de la espalda baja, el músculo recto abdominal, es también el centro motor del cuerpo, pues éste actúa antes que todo movimiento importante al tratarse de saltar. El recto abdominal tira de la pelvis hacia arriba y la caja torácica hacia abajo, dependiendo de cuál sección actué como origen y cuál como inserción. En este caso, cuando el cuerpo está despegando del suelo eleva la pelvis. Aquí se aplica una palanca de 1er género. En las articulaciones de las vértebras lumbares, la parte media de la región lumbar actúa como apoyo, la parte posterior de la pelvis -al flexionarse con las vértebras lumbares- actúan como resistencia y la inserción de los músculos abdominales en la parte frontal de la pelvis como potencia.

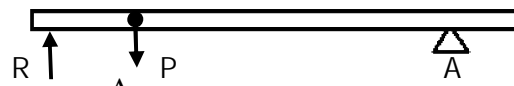
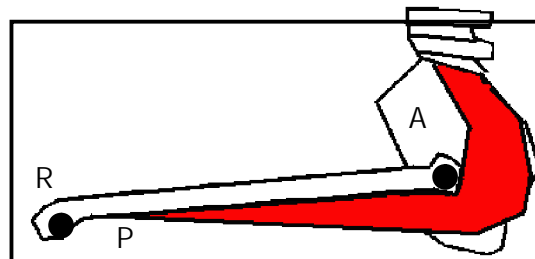
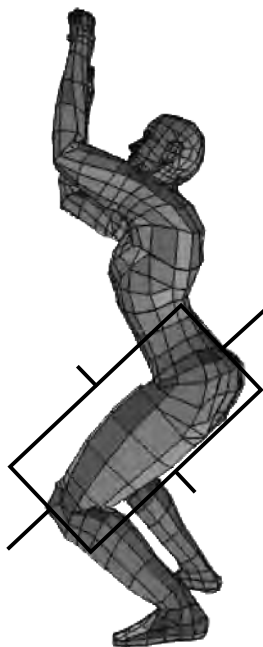
En el momento en que se extienden las piernas al saltar, si se contraen al mismo tiempo estos músculos abdominales agregarán un impulso mas al salto precisamente porque al flexionarse, se endereza la sección lumbar haciendo girar la pelvis hacia la parte frontal y al mismo tiempo dando un impulso hacia abajo a la cabeza del fémur. De esta manera se continúan transmitiendo fuerzas a las demás articulaciones subsecuentes. Este es el impulso de la sección media del cuerpo, sin este impulso se pierden gran parte de las fuerzas de transmisión provenientes de los miembros superiores del cuerpo.



R. Resistencia
A. Apoyo
P. Potencia

3.3.10 Extensión del muslo

Los músculos extensores de la articulación de la cadera y muslo son el glúteo mayor y menor, los cuales tiran de las piernas hacia la parte inferior y posterior una vez que ya están flexionadas y el atleta está de pie, teniendo como consecuencia la elevación del tronco. Aquí se aplica una palanca de 3er género en la articulación del fémur con la pelvis.

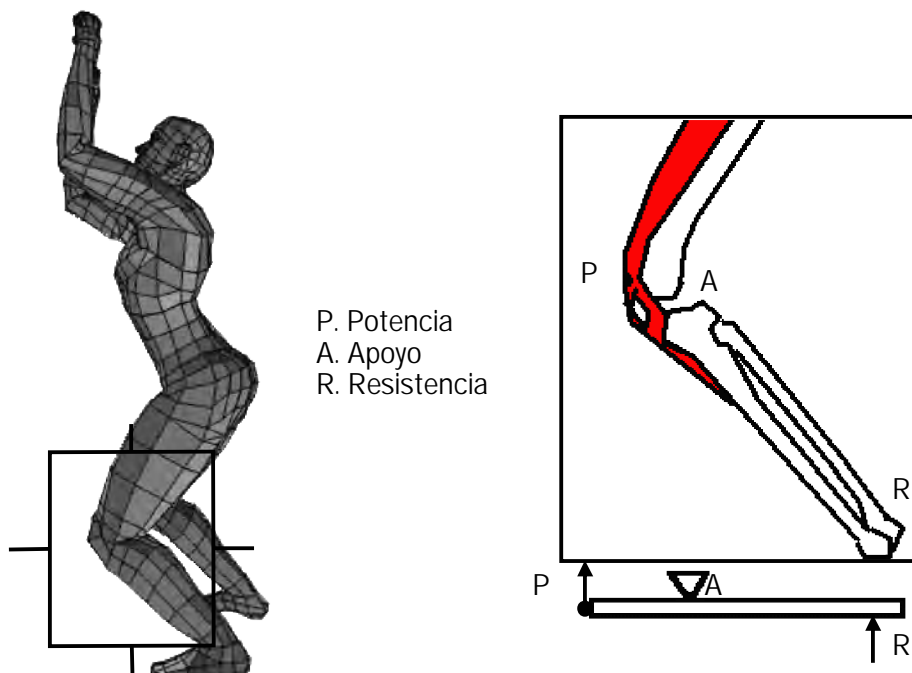


R. Resistencia
P. Potencia
A. Apoyo

El extremo del fémur actúa como resistencia al empujar la pierna hacia arriba por causa del peso del tronco, la articulación de la cabeza del fémur actúa como apoyo y el tendón de inserción en la parte inferior y superior del fémur como potencia. Este movimiento tiene como consecuencia la extensión de las piernas hacia la parte inferior y posterior.

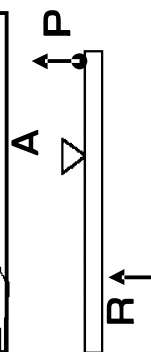
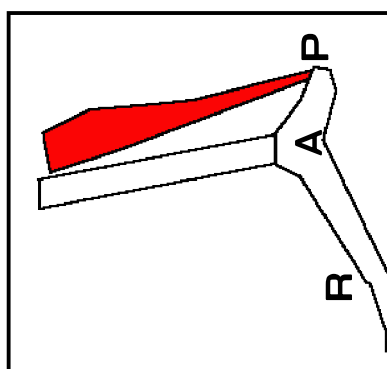
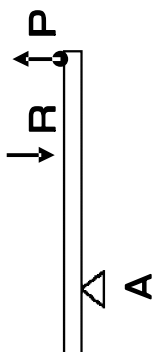
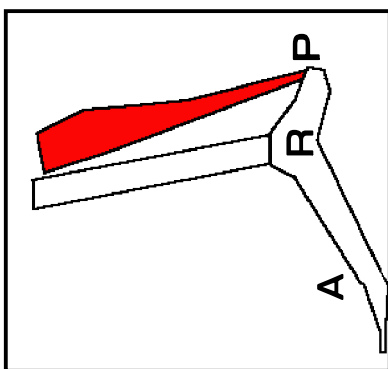
3.3.11 Extensión de la pierna

El músculo extensor de la articulación de la rodilla es el cuádriceps el cual tira de la antepierna hacia el lado frontal y hacia arriba una vez que ya está flexionada, teniendo como consecuencia el empuje del tronco hacia arriba al ser extendidas, aquí se aplica una palanca de 1er género en la articulación del fémur con la tibia, en la rodilla. El extremo inferior de la tibia y peroné actúan como resistencia debido al peso del cuerpo que flexiona la pierna, la articulación de la tibia con el fémur actúa como apoyo y la inserción en el tendón rotuliano como potencia. (La rótula es una extensión de la tibia y el peroné para la inserción del tendón de la potencia).



3.3.12 Extensión del pie

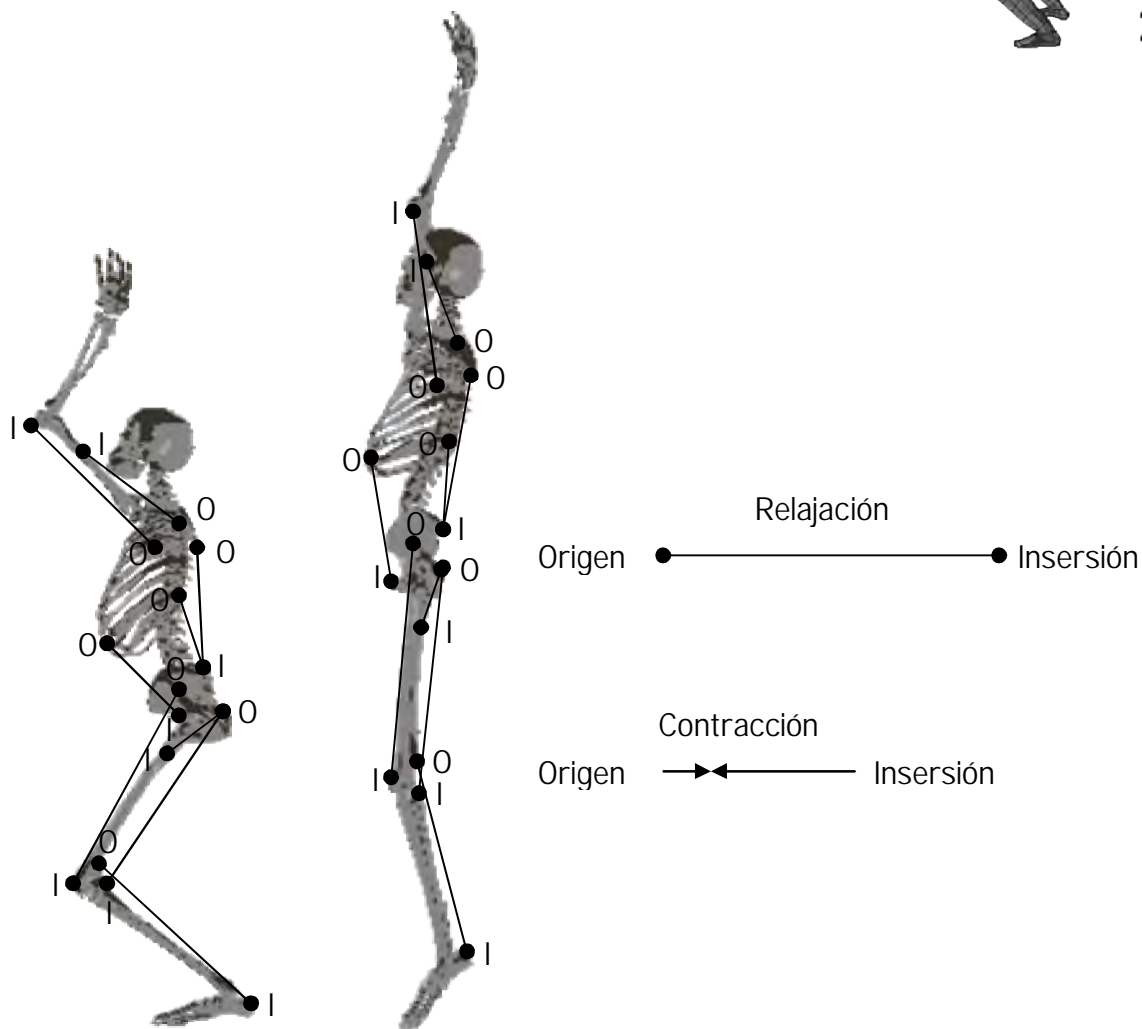
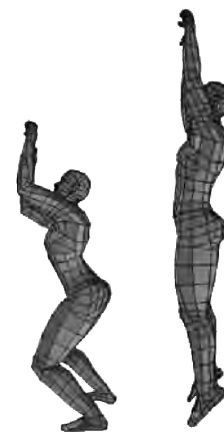
Los músculos extensores de la articulación del pie son los músculos gemelos, los cuales tiran del talón del pie hacia arriba una vez flexionado, teniendo como consecuencia la rotación del empeine del pie hacia abajo y la elevación del tobillo. Aquí se aplica una palanca de 1er género en la articulación del astrágalo con la tibia. La punta del pie actúa como resistencia debido a que el peso del cuerpo hace que el suelo empuje el empeine hacia arriba, en la cabeza de los huesos metatarsianos, la articulación del astrágalo con la tibia actúa como apoyo y el tendón de Aquiles como potencia. La mayoría de las ocasiones se ha dicho que esta palanca es una palanca de 2do género, donde la punta del pie es el apoyo y la articulación del tobillo es la resistencia. Sin embargo de acuerdo a las definiciones de otros autores, el fulcro o la articulación que se hace rotar es el punto de apoyo [002].



3.3.13 Origen e Inserción

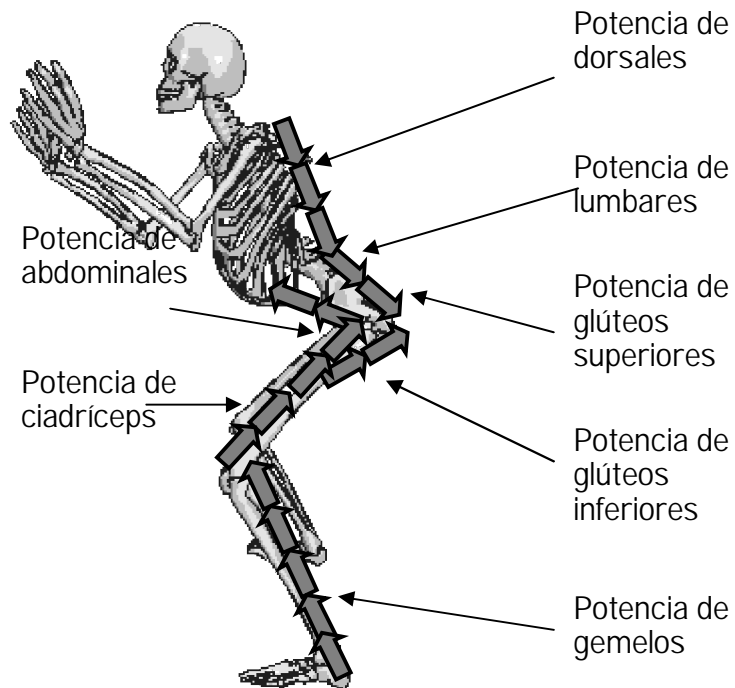
Como se ha comentado en otros capítulos, los músculos no poseen poder de empuje sino de contracción. Sin embargo al estar sujetos a un origen e insertados por medio de los tendones en cada segmento óseo, se produce el poder del empuje por medio de las palancas al hacer rotar las articulaciones. [002].

En la parte inferior se muestran los músculos principales del salto vertical al momento del despegue en la extensión de los segmentos. El origen está compuesto en su mayor proporción de masa muscular y la inserción mayormente tendinosa. Por esta razón el sentido de la contracción es de la inserción hacia el origen.

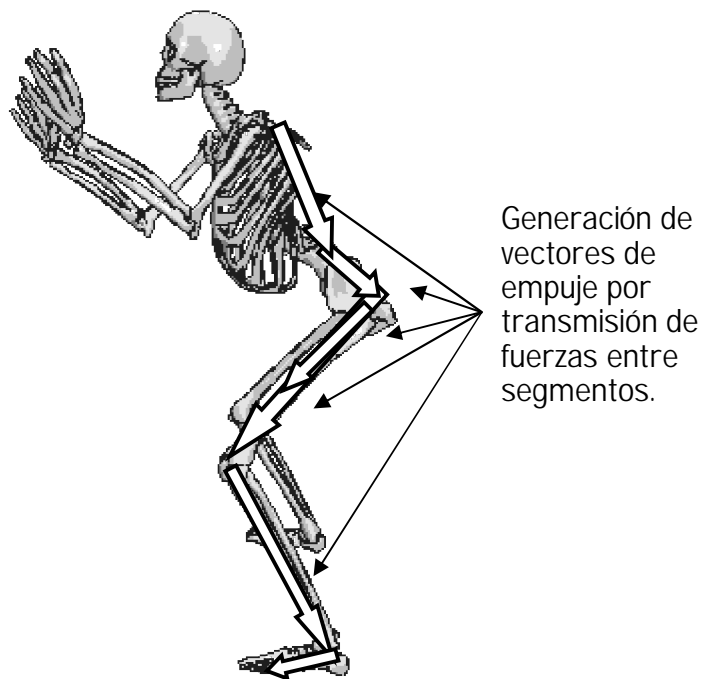


3.4 Generación de vectores de fuerza

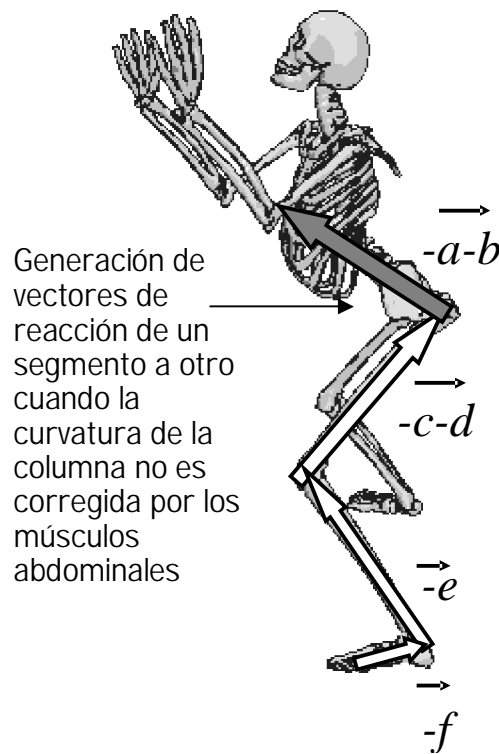
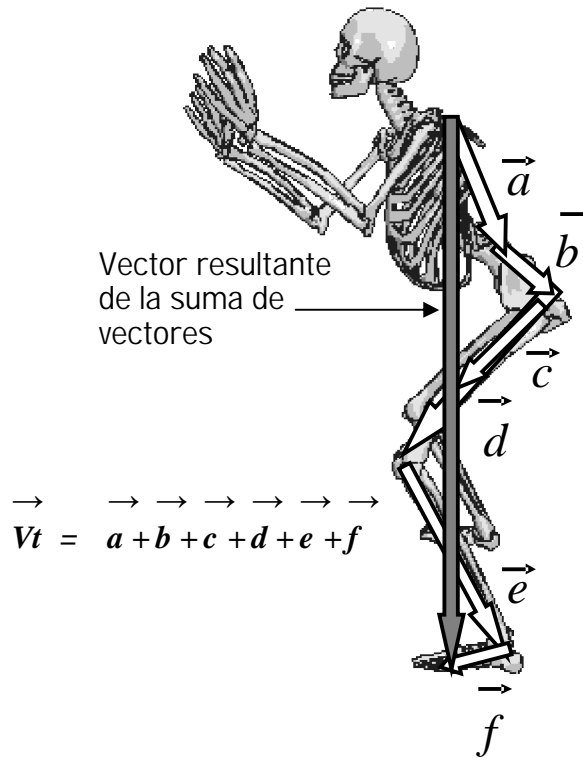
Como ya se había comentado en el caso de la báscula, al estar parado y levantar los brazos rápidamente, se puede registrar un cambio en el peso del cuerpo debido a la transmisión de las fuerzas. Dichas fuerzas se transmiten de un segmento a otro, y si uno de estos segmentos desperdicia parte del impulso o bien no ejerce nada de impulso en la cadena de transmisión de fuerzas se perderá gran parte de la transmisión teniendo como resultado un salto deficiente.



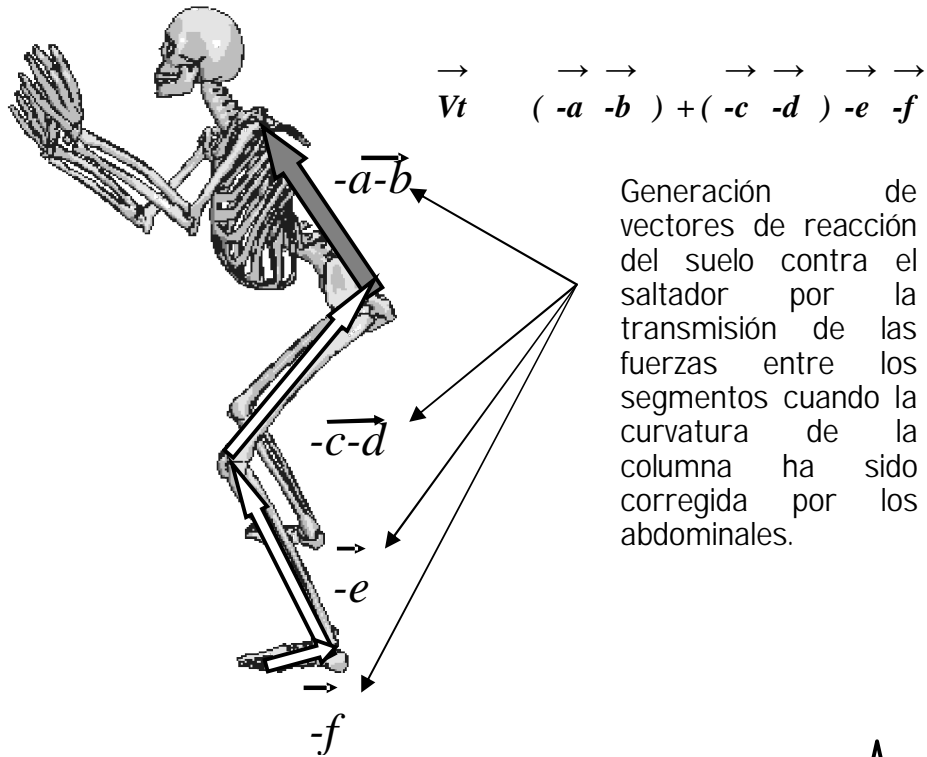
Cada músculo al contraerse genera una potencia a sus respectivos brazos de palanca, sumándose a las palancas adyacentes formando la transmisión de las fuerzas de un segmento a otro. En el caso de los abdominales, estos corrigen la excesiva curvatura de la columna vertebral impidiendo con esto que la fuerza del impulso no se desperdicie en la deformación de la columna al doblarse hacia atrás, al mismo tiempo los abdominales agregan potencia al brazo de palanca de los glúteos superiores.



En la figura de abajo se observa que la deformación de la columna vertebral causada por la falta de contracción en los músculos abdominales produce una desviación en el último vector de empuje al no llegar a su punto de origen de la fuerza que está en la parte superior de la espalda, el salto obtenido tendrá un desperdicio de impulso al irse parte de este en la deformación de la zona lumbar (espalda baja).



Por otro lado, si la contracción de los músculos abdominales es efectiva, no habrá desviación de fuerzas de reacción del suelo contra el saltador y el impulso será aprovechado al no deformarse la zona lumbar. "La función postural importante de todos los músculos del abdomen es hacer la tracción hacia arriba de la parte anterior de la pelvis, la cual aplanar la curvatura lumbar de la columna vertebral" (O13).

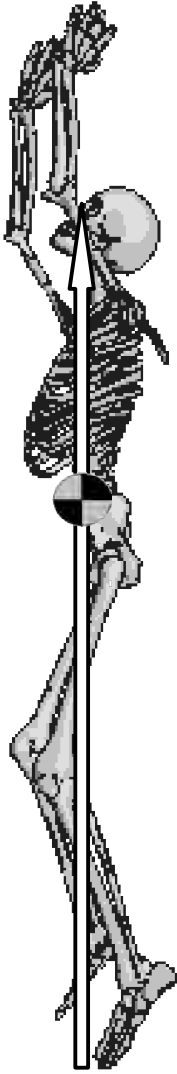


Generación de vectores de reacción del suelo contra el saltador por la transmisión de las fuerzas entre los segmentos cuando la curvatura de la columna ha sido corregida por los abdominales.

Finalmente en esta parte se aprecia el vector resultante de reacción del suelo contra el saltador, de acuerdo a la 3ra ley de Newton ("A toda acción corresponde una reacción de igual magnitud pero en sentido contrario".)

Vector de reacción del suelo contra el saltador, que pasa por el centro de gravedad





Los vectores resultantes pasan de un segmento a otro que al sumarse se obtiene el vector que pasa por el centro de gravedad moviendo todo el sistema. Los músculos que rodean al centro de gravedad corresponden a los rectos mayores, oblicuos y las fibras horizontales. Al momento de saltar los abdominales continúan contrayéndose para colocar el cuerpo listo para la caída. Este mismo abdominal que aumenta la altura del salto, protege las vísceras abdominales en caso de que dicha caída sea de una altura considerable. También hay un control del centro de gravedad por parte de los abdominales. Antes de cada movimiento siempre se contrae una sección de ellos para acomodar todos los brazos de palanca adyacentes y ejecutar mejor cada movimiento. En el caso de que se ejecuten movimientos en el aire todos estos girarán en torno a este centro de gravedad, y como ya se ha expuesto anteriormente puede obtenerse la suspensión del salto con el descenso de los brazos y de las piernas.

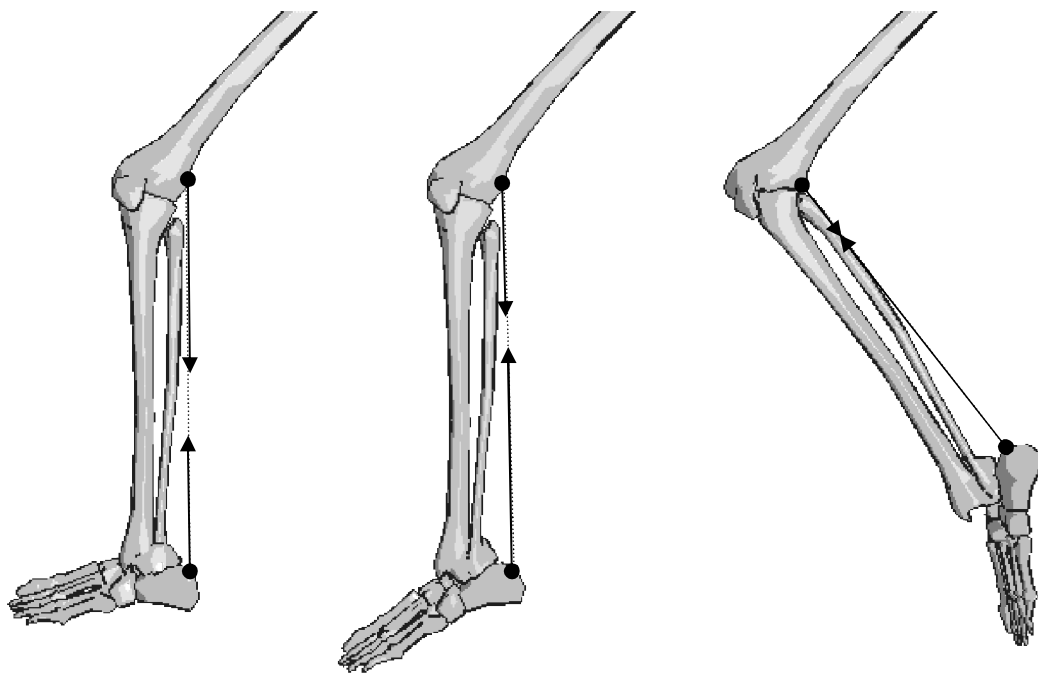
El C.G. es controlado inicialmente por la cavidad abdominal que "aloja a las vísceras y que esta cerrada por delante y en los costados por paredes blandas: los músculos abdominales. En el momento en que estas paredes se distienden, hay un desequilibrio de la estática general de cuerpo. Los antiguos griegos habían comprendido perfectamente la importancia de los músculos abdominales" [O13].

"Un conjunto de abdominales fuertes y bien definidos parece una red impresionante en engranes acoplados, en realidad, los abdominales son la fuerza impulsora de la máquina corporal. Sin ellos, la fuerza que se genera en los movimientos de esta naturaleza está sumamente limitada, independientemente de las demás fuerzas que intervengan" [O13]. Contrario a lo que muchos han de pensar, no toda la fuerza recae sobre las piernas al momento del impulso, por mucha fuerza que se llegue a tener en las piernas, si no se tienen unos abdominales fuertes y elásticos no se podrá mejorar el salto. Ahora bien como es sabido que a cada músculo agónico le corresponde un antagonista, cuando los cuadrado lumbares se contraen seguidos de los glúteos mayores, involuntariamente los músculos abdominales se distienden, provocando la curvatura excesiva en la zona lumbar. Y cuando los abdominales se contraen, aquellos se distienden, provocando así la deficiencia al brincar. Por lo tanto, para obtener un mejor resultado en el salto, al ejercitarse deberá buscarse el entrenamiento apropiado de manera que se contraigan al mismo tiempo en esta parte de transmisión de las fuerzas: Abdominales, cuadrado lumbar y glúteos. Debe haber una cocontracción entre todos ellos.

Para vencer un peso de 1N en un brazo de resistencia equidistante de un brazo de potencia con respecto al punto de apoyo se necesita una fuerza de 1N para equilibrar la palanca y una mayor a 1N para vencerla. Para levantar el peso total del cuerpo se necesita una fuerza superior al peso total del cuerpo para poder elevarlo.

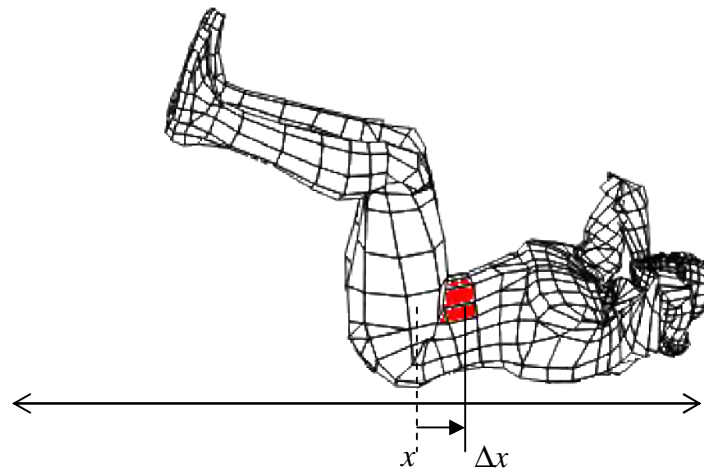
Cada segmento del cuerpo al irse extendiendo en las articulaciones, lo hace con un brazo de palanca que genera una fuerza mayor al peso del segmento a extender, sumándose en cada segmento subsiguiente la fuerza del segmento anterior, a lo que se puede llamar la transmisión de fuerzas. Una vez que llegan hasta el último brazo de palanca de los pies. Este brazo de palanca recibe toda la fuerza de todos los segmentos superiores, el cual al sumarse debe vencerla con una fuerza mayor al peso total del cuerpo. Dicha fuerza es enviada al suelo el que (3ra ley de Newton) responde con la misma fuerza pero en sentido contrario enviando el cuerpo hacia arriba e iniciando el despegue.

Por ello es importante que los músculos gemelos estén lo suficientemente entrenados en potencia para dar el último impulso que debe ser el más fuerte de todos. Esto es lo que sucede precisamente en el caso del canguro. El secreto está en su largo tendón de Aquiles. "Es como si el canguro estuviera saltando con un par de resortes en los pies". El tendón de Aquiles del canguro trabaja de manera similar al tendón que los seres humanos tienen unido al músculo de la pantorrilla: se estira al caer al suelo y se contrae al momento de saltar [O17].

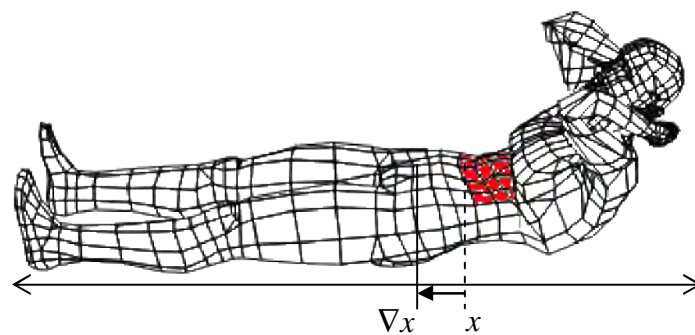


Algunos saltadores extraordinarios de nacimiento, experimentan cierto dolor en la parte superior de los gemelos después de haber saltado con mucha frecuencia. Aquí se muestra el sentido de la contracción muscular.

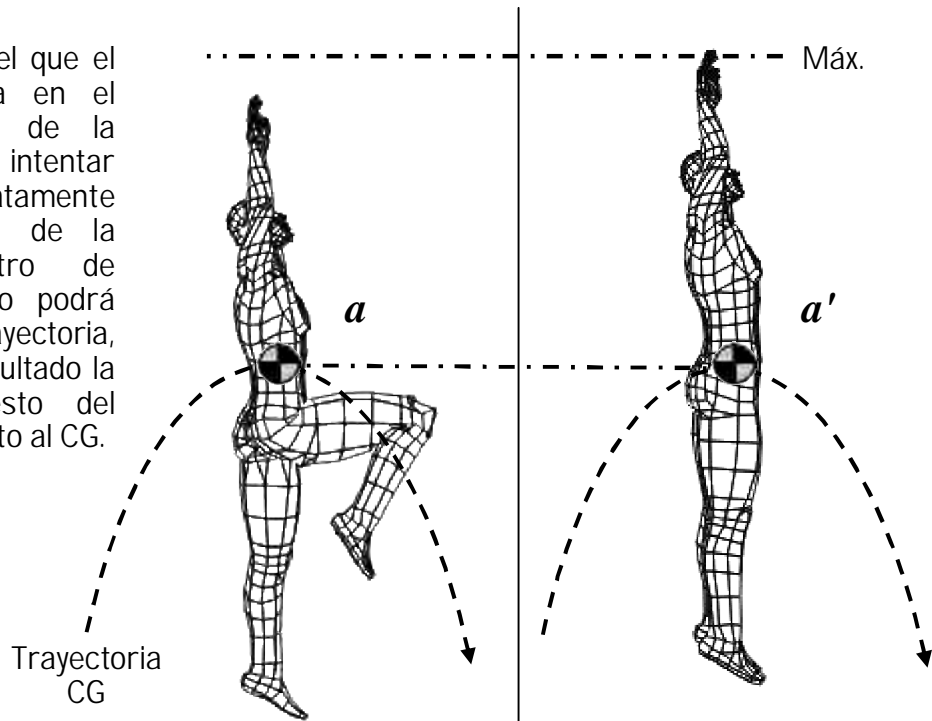
En los músculos rectos abdominales inferiores. Cuando las costillas actúan como el origen del músculo y la pelvis como la inserción, la pelvis se flexiona hacia las costillas, iniciando la flexión de las piernas con ayuda de los músculos psoas iliaco y recto interno, elevando el CG. Al saltar, en el momento del despegue, si el CG parte de un sitio alto, la parábola o la vertical formada también será alta. Sin embargo esta sección muscular es más utilizada en el salto para ganar potencia como ya se comentó.



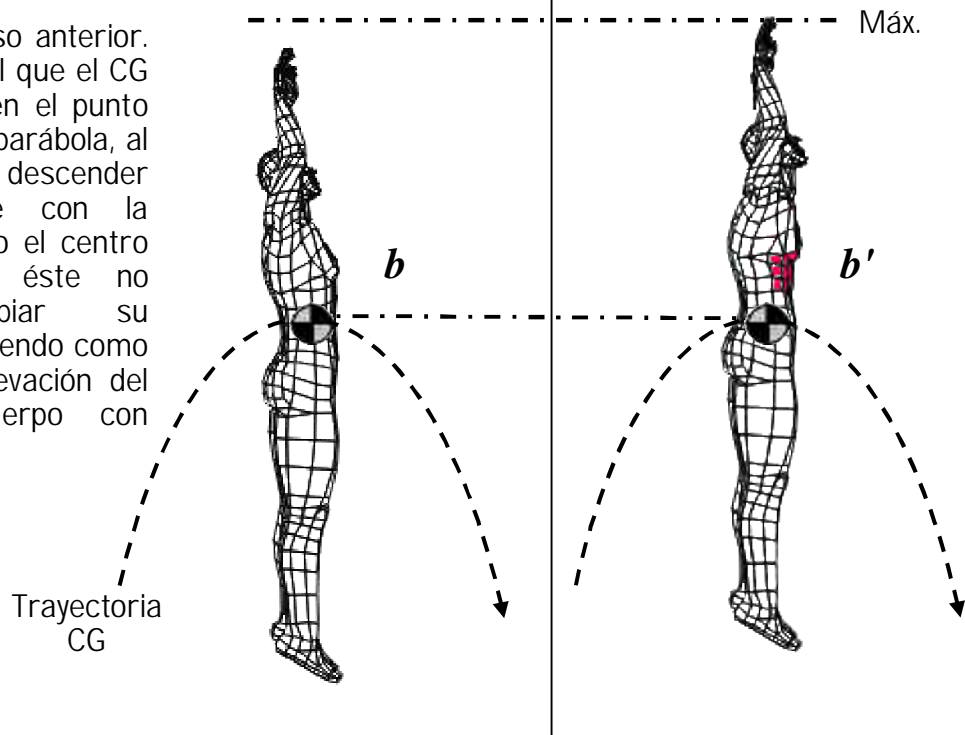
En los músculos rectos abdominales superiores. Cuando la pelvis actúa como origen del músculo y las costillas como la inserción. La caja torácica se flexiona hacia abajo y el centro de gravedad baja. Al momento de saltar y alcanzar gravedad cero, si se baja el CG con cualquiera de las extremidades superiores e inferiores en el descenso comenzando por los abdominales superiores, el resto del cuerpo sube puesto que no es posible cambiar en el aire la trayectoria del centro de gravedad dentro del mismo sistema. Por otro lado si se hace descender el CG al ir cayendo, el CG podrá continuar su trayectoria parabólica o vertical descendente y suspender el cuerpo en el aire por un instante.



En el instante en el que el CG se encuentra en el punto más alto de la parábola, al intentar descender inmediatamente con la extensión de la pierna el centro de gravedad, éste no podrá cambiar su trayectoria, teniendo como resultado la elevación del resto del cuerpo con respecto al CG.



Como en el caso anterior. En el instante el que el CG se encuentra en el punto mas alto de la parábola, al intentar descender inmediatamente con la flexión del torso el centro de gravedad, éste no podrá cambiar su trayectoria, teniendo como resultado la elevación del resto del cuerpo con respecto al CG.



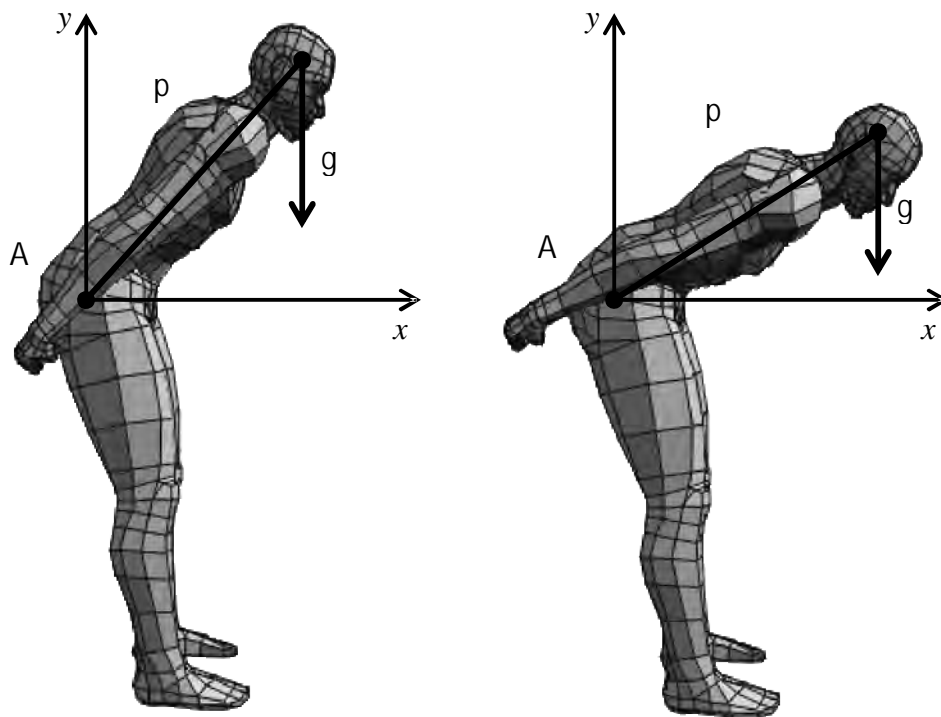
3.5 Dificultad en los grados de inclinación

Una vez entendido que es importante la utilización de los músculos abdominales para los brazos de palanca y el centro de gravedad, es necesario recalcar que los grados de inclinación de los brazos de palanca a lo largo de los segmentos al momento de saltar, deben de ser los apropiados para efectuar un entrenamiento con pesas aumentando la fuerza y por el otro lado reducir el esfuerzo al ejecutar el salto [O25].

En el supuesto de que la posición inicial sea con una inclinación hacia delante y los brazos a lo largo del tronco, se tiene lo siguiente:

El tronco inclinado p representa una palanca que tiende a girar alrededor de un eje A debido a la fuerza de gravedad g . Ahora bien, para poder permanecer en esta posición los grupos de músculos con la fuerza de x_1 tienen que esforzarse para sostener la posición de la primera figura y con la fuerza de x_2 para sostener la posición de la segunda figura.

Si $x_1 = x_2$, entonces no hay diferencia en el grado de dificultad entre las dos posiciones. Para determinar cuál de las proposiciones es verdadera. Si ambas posiciones requieren el mismo esfuerzo, si la primera requiere mas que la segunda o viceversa:



A primera vista la única diferencia es el ángulo de inclinación del tronco donde la longitud permanece constante para el mismo individuo y la fuerza de gravedad también permanece constante.

Ahora hay que determinar la relación existente entre los tres factores del individuo A, p, g. Se debe introducir un factor adicional a la fuerza de los brazos a_i , que son perpendiculares a la distancia del eje A a la línea de fuerza g.

La relación entre a_i y g se expresa como sigue:

$$M_1 = a_1 \cdot g \qquad M_2 = a_2 \cdot g$$

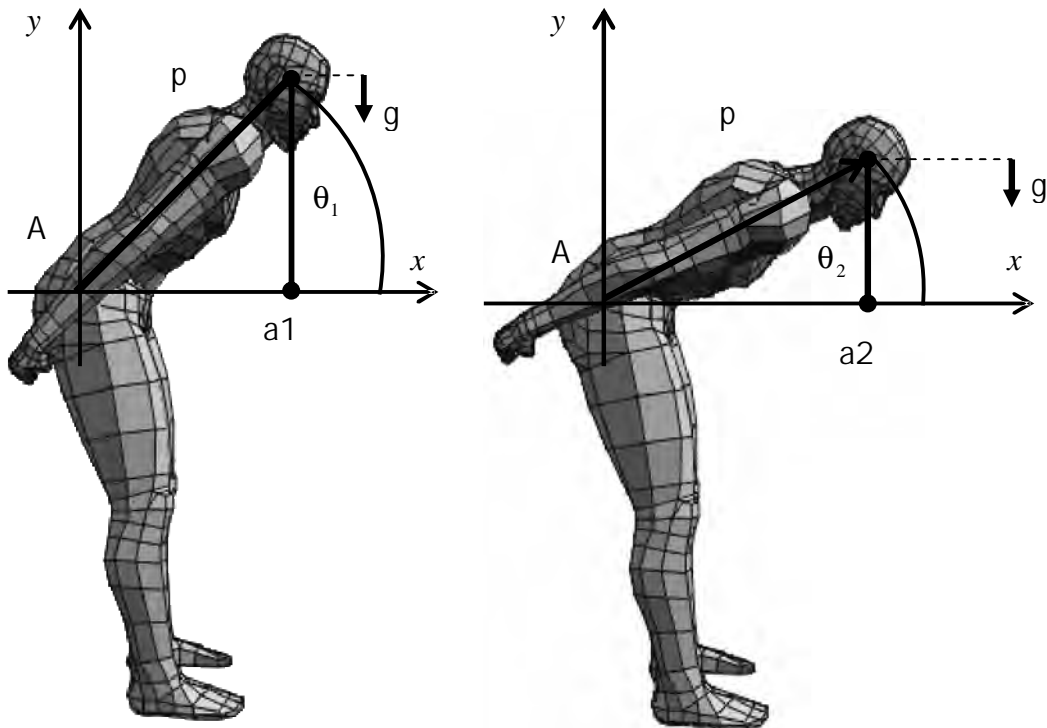
Donde M_i es el momento de fuerza que propende a hacer rotar el tronco. Como p y g permanecen constantes para el mismo individuo, la única diferencia está en el $\cos(\theta_i)$.

$$a_1 = p \cos\theta_1 \qquad a_2 = p \cos\theta_2$$

Luego se sustituye en las dos primeras ecuaciones los valores de a_i :

$$M_1 = p \cos\theta_1 \cdot g \qquad M_2 = p \cos\theta_2 \cdot g$$

Entre más pequeño es el ángulo, más grande es el cos, Como $\theta_2 < \theta_1 \Rightarrow a_2$ es más grande que a_1 ,



De esta relación se concluye que el grado de dificultad depende de los cambios, ya sea aumento o disminución que se efectúen en cualquiera de los factores siguientes:

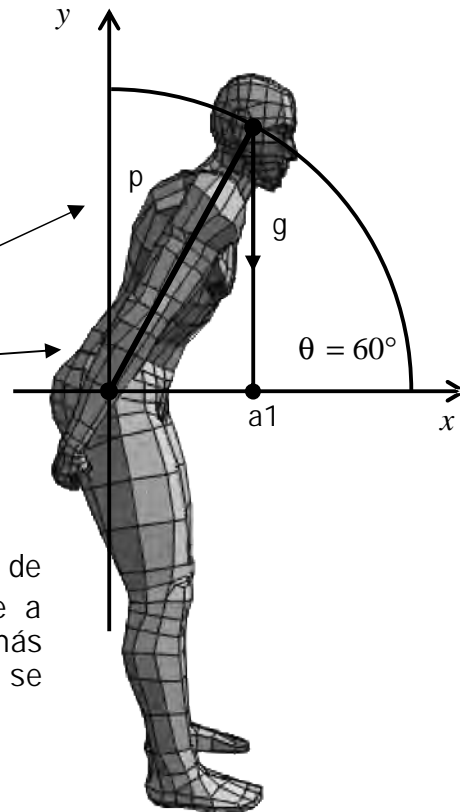
$$x_1 = p \cdot g \cdot \cos\theta$$

$$x_1 = p \cdot g \cdot \cos(60^\circ)$$

$$\cos(60^\circ) = 0.5$$

$$\therefore x_1 = p \cdot g \cdot (0.5)$$

Grupo de
músculos
con la fuerza
de x_1



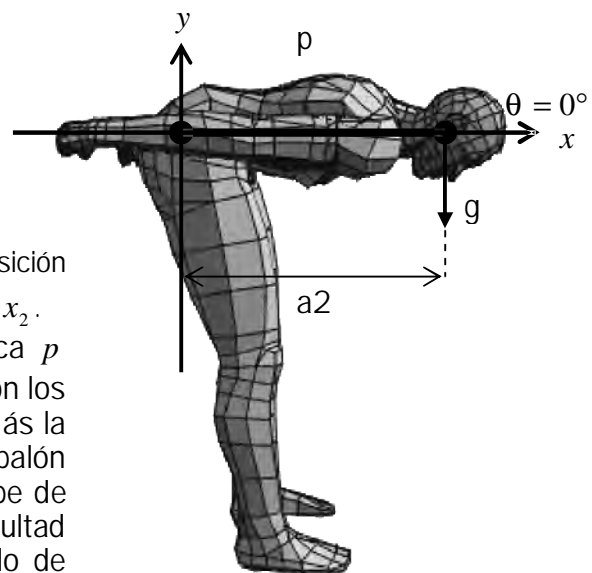
Fuerza de x_1 Al hacer cambio del ángulo de inclinación del tronco, que por supuesto que a mayor inclinación hacia la línea horizontal será más difícil. Si se inclina a 60° de la línea horizontal, se hace el cálculo para la fuerza de x_1 .

$$x_2 = p \cdot g \cdot \cos\theta$$

$$x_2 = p \cdot g \cdot \cos(0^\circ)$$

$$\cos(0^\circ) = 1$$

$$\therefore x_2 = p \cdot g \cdot 1$$



Fuerza de x_2 En la inclinación de la posición horizontal, se hace el cálculo para la fuerza de x_2 .

Si se cambia la longitud de la palanca p es más difícil sostenerla si ésta se hace con los brazos sobre la cabeza. Aumentará aún más la dificultad si detrás el cuello se coloca un balón medicinal. Por lo que, para ejercitarse debe de hacerse con el mayor grado de dificultad posible y para saltar con un menor grado de dificultad posible [025].

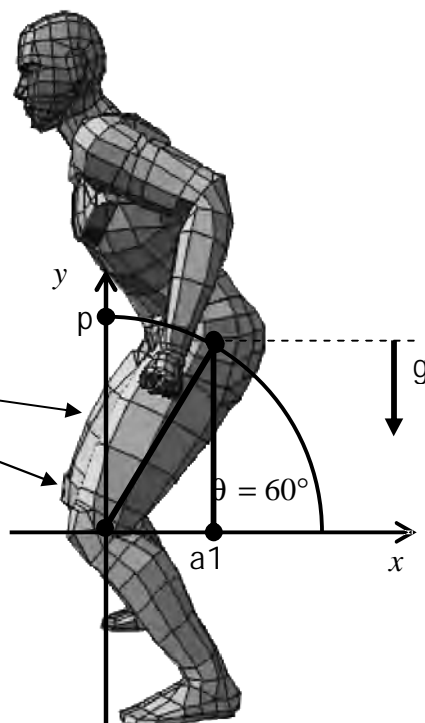
$$x_1 = p \cdot g \cdot \cos(\theta)$$

$$x_1 = p \cdot g \cdot \cos(60^\circ)$$

$$\cos(60^\circ) = 0.5$$

$$\therefore x_1 = p \cdot g \cdot (0.5)$$

Grupo de
músculos
con la fuerza
de x_1



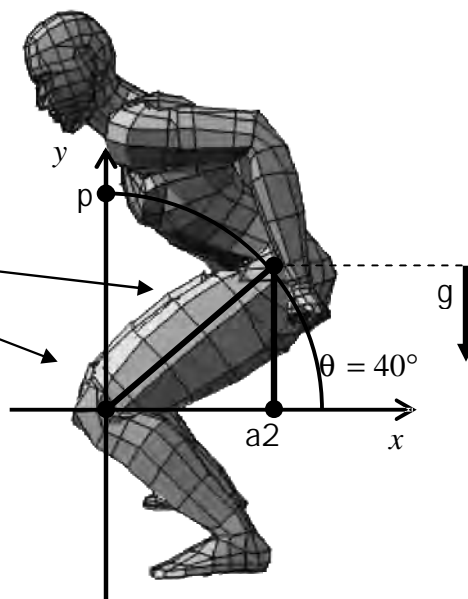
$$x_2 = p \cdot g \cdot \cos(\theta)$$

$$x_2 = p \cdot g \cdot \cos(40^\circ)$$

$$\cos(40^\circ) = 0.76$$

$$\therefore x_2 = p \cdot g \cdot (0.76)$$

Grupo de
músculos
con la fuerza
de x_2



3.5.1 Posición incorrecta

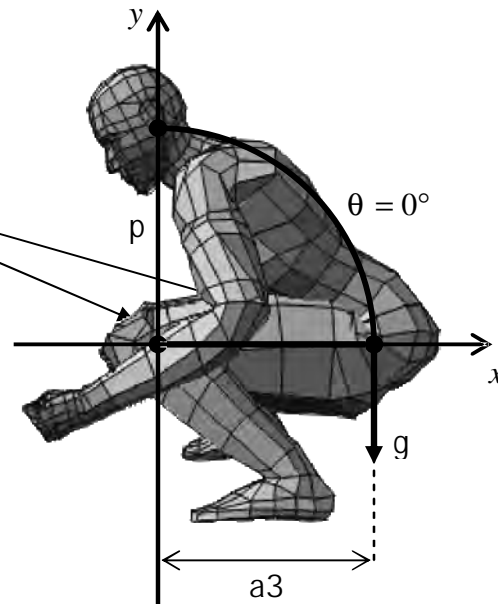
$$x_3 = p \cdot g \cdot \cos(\theta)$$

$$x_3 = p \cdot g \cdot \cos(0^\circ)$$

$$\cos(0^\circ) = 1$$

$$\therefore x_3 = p \cdot g \cdot (1)$$

Grupo de
músculos
con la fuerza
de x_3

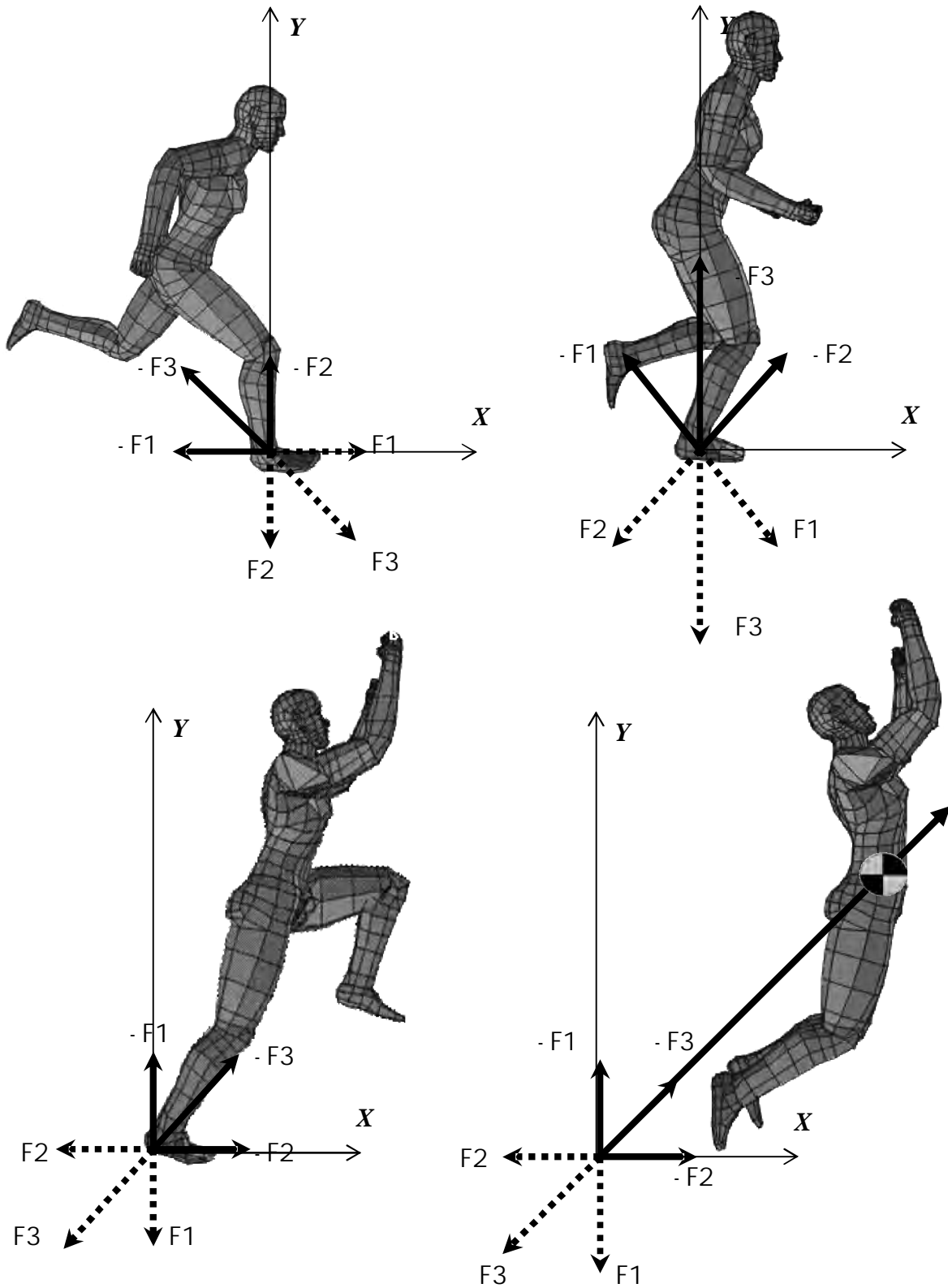


Esta posición no es recomendable para los entrenamientos de salto debido a que se ejerce una gran presión sobre la rótula de las rodillas que es donde se concentra la potencia de los músculos cuádriceps. Algunos de estos entrenamientos pueden ser las sentadillas pronunciadas como se muestra en la figura.

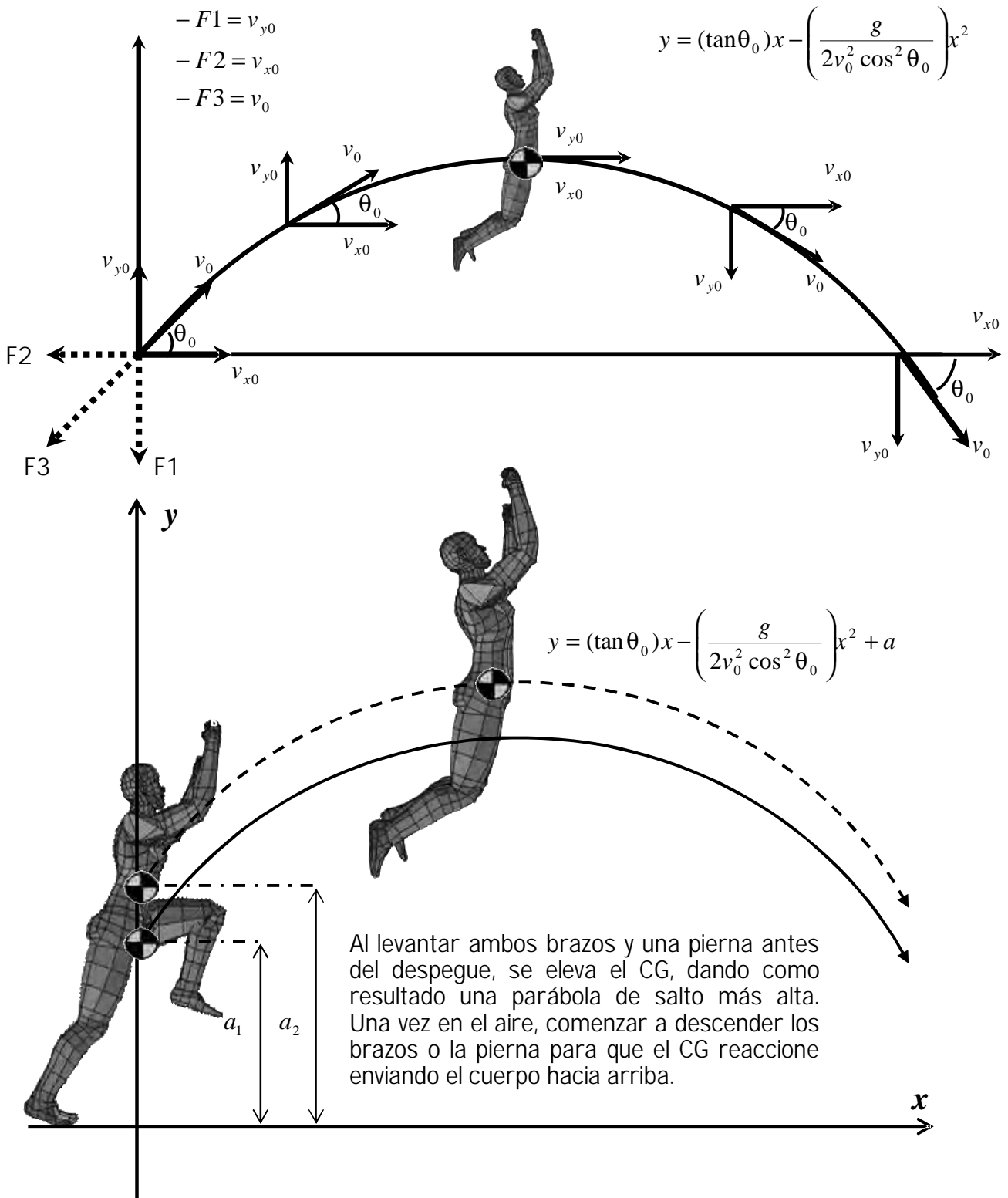
3.6 Vectores en el salto parabólico

A veces interesa que el sistema coordinado de estudio esté compuesto sólo por ciertos segmentos corporales, como sucede en el ejemplo presentado en la figura siguiente, donde se representa el diagrama de fuerzas del miembro inferior derecho de un saltador en el instante de despegar del suelo. En este caso el cuerpo del saltador se considera un elemento externo al sistema que está en contacto con él, lo que constituye una fuerza externa. La fuerza que ejerce el suelo contra el sistema es $-F_3$, la cual se ha descompuesto en sus dos componentes a lo largo de los ejes \mathbf{X} e \mathbf{Y} , siendo $-F_1$ la fuerza en el eje \mathbf{X} y $-F_2$ la fuerza en el eje \mathbf{Y} . En este caso pueden obviarse el resto de las fuerzas por no ser relevantes para su estudio, como las aerodinámicas. A continuación se muestran las fuerzas que intervienen:

- F1: Vector de acción de la carrera.
- F2: Vector de acción de la pierna contra el suelo.
- F3: Vector de acción resultante.
- F1: Vector reacción de la carrera.
- F2: Vector de reacción del suelo contra la pierna.
- F3: Vector de reacción resultante.



En capítulos anteriores se mostró el desarrollo para la parábola del proyectil, en donde:



Al darle valores a la fórmula de la trayectoria parabólica citada anteriormente, donde el origen de las coordenadas está en el punto más alto del centro de gravedad al despegar del suelo:

$$y = (\tan\theta_0)x - \left[\frac{g}{2v_0^2 \cos^2 \theta_0} \right] x^2$$

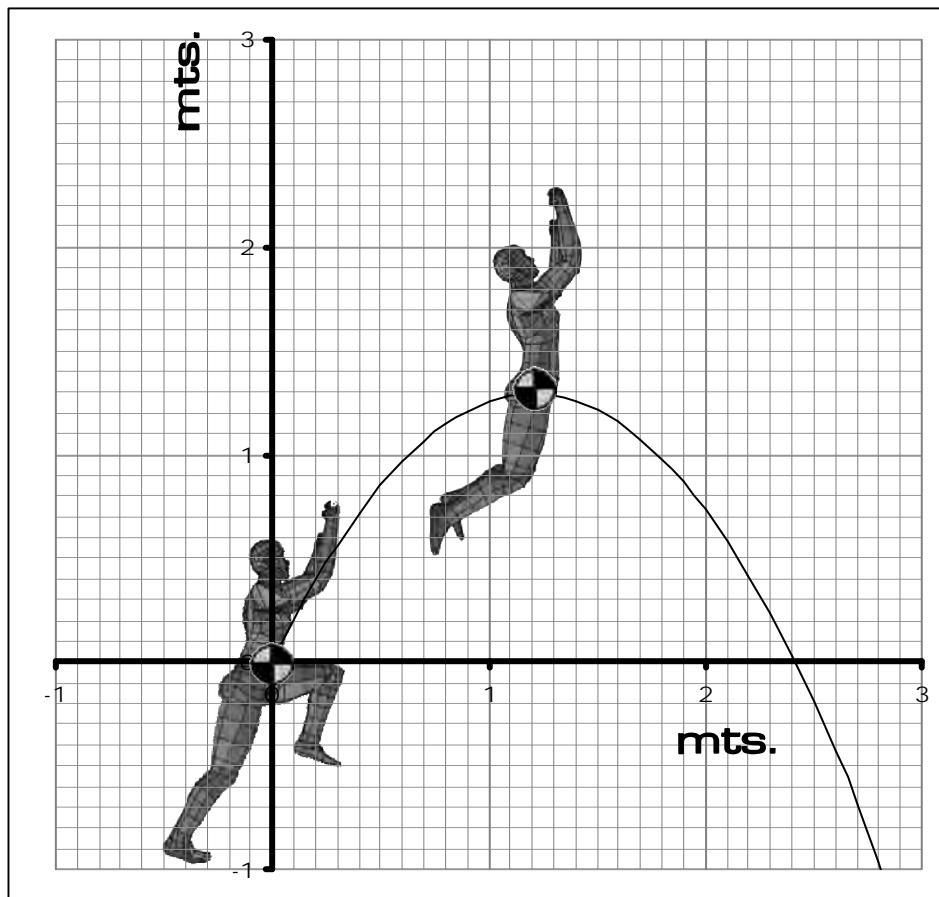
$$v_0 = 20\text{km/hr} = 5.5\text{m/seg}$$

$$\theta_0 = 65^\circ$$

$$g = 9.8\text{m/seg}^2$$

$$0 \leq x \leq 4$$

Velocidad Km/hora	20
Velocidad m/s	5.56
grados inclinación	65
gravedad m/s ²	9.8
x	y1
0.0	0.00
1.0	1.26
2.0	0.73
3.0	-1.57
4.0	-5.64



3.7 Fuerza Explosiva

El impulso mide la acción de la fuerza durante cierto intervalo de tiempo. Es el producto de la intensidad de esta fuerza por el tiempo durante el cual actúa. Para ganar impulso explosivo debe haber fuerzas muy grandes que actúen durante intervalos de tiempo muy cortos. Ahora bien, el impulso está dado por:

$$I = F\Delta t$$

Donde: I es el impulso en unidades de N.s
 F Fuerza
 Δt Intervalo de tiempo

Es importante incrementar la fuerza de las contracciones musculares y reducir el tiempo aumentando la velocidad para lograr un salto explosivo. Supóngase que el atleta de masa $m = 75\text{kg}$, se impulsa en línea recta bajo la acción resultante de la fuerza F , que es constante.

En un instante dado $t_1 = 0\text{s}$, la velocidad $v_1 = 0\text{m/s}$
 y en el instante $t_2 = 0.5\text{s}$ la velocidad $v_2 = 4\text{m/s}$
 $\Rightarrow \Delta t = t_2 - t_1$.

$a = \frac{v_2 - v_1}{\Delta t}$	$F = ma$	$I = F\Delta t$
$a = \frac{v_2 - v_1}{t_2 - t_1}$	$F = (75\text{kg})(8\text{m/s}^2)$	$I = (600\text{N})(0.5\text{s})$
$a = \frac{4\text{m/s} - 0\text{m/s}}{0.5\text{s} - 0\text{s}}$	$F = 600\text{kg} \cdot \text{m/s}^2$	$I = 300\text{N} \cdot \text{s}$
$a = \frac{4\text{m/s}}{0.5\text{s}}$	$F = 600\text{N}$	
$a = 8\text{m/s}^2$		

Si se aumentara la velocidad $v_2 = 5\text{m/s}$

$a = \frac{v_2 - v_1}{\Delta t}$	$F = ma$	$I = F\Delta t$
$a = \frac{v_2 - v_1}{t_2 - t_1}$	$F = (75\text{kg})(10\text{m/s}^2)$	$I = (750\text{N})(0.5\text{s})$
$a = \frac{5\text{m/s} - 0\text{m/s}}{0.5\text{s} - 0\text{s}}$	$F = 750\text{kg} \cdot \text{m/s}^2$	$I = 375\text{N} \cdot \text{s}$
$a = \frac{5\text{m/s}}{0.5\text{s}}$	$F = 750\text{N}$	Este será el impulso.
$a = 10\text{m/s}^2$		

*“Las ciencias no tratan de explicar,
incluso apenas tratan de interpretar,
construyen modelos principalmente.*

Por modelo,

*se entiende una construcción matemática que,
con la adición de ciertas interpretaciones verbales,
describe los fenómenos observados.*

La justificación de tal construcción matemática es

sólo y precisamente que

se espera que funcione.”

JOHN VON NEUMANN

4 Regresión lineal y desarrollo de sistema para análisis de entrenamientos

4.1 Regresión lineal en los entrenamientos de salto

En estadística se utiliza el modelo de regresión lineal, además de los histogramas para determinar el comportamiento de los datos. En este caso la aplicación es a los entrenamientos llevados a cabo por los atletas y los efectos producidos en la ejecución de los saltos vertical y parabólico.

La palabra regresión fue usada la primera vez por Francis Galton (1822-1911) en los estudios que realizó sobre herencia biológica.

En dichos estudios se notó que, tomando las características de un grupo en particular de la población, en su próxima generación, dichas características mostraban una tendencia a moverse en dirección a las características de toda la población, o bien, regresaban a las características promedio de la población en general. Esta tendencia fue referida como una regresión hacia la media de la población.

La regresión es el ajuste de diferentes valores de "X" al valor promedio de la población gracias a una variable independiente que hace que cualquier valor de "X" tienda a ese valor promedio.

A pesar de que no se encuentra problema alguno con las designaciones comunes de variable independiente y dependiente para X e Y, respectivamente, se preferiría denominarlas como variables de predicción y de respuesta. Puesto que en la regresión sólo puede asociarse un valor de X con uno de respuesta Y. No es posible establecer una relación causa-efecto de la X a la Y, y al mismo tiempo de la Y a la X. Por ejemplo, se sabe que existe una relación entre la altura y el peso de los seres humanos, pero esto no necesariamente significa que pueda cambiar la altura de una persona si se modifica su peso. El análisis de regresión sólo descubre una asociación entre la variable de predicción X y la variable de respuesta Y.

En estadística se usa el término "causalidad", la causalidad estadística es la dependencia mutua de dos variables, sin embargo la regresión lineal no puede caer en la causalidad, ya que en este modelo, se encuentra una variable independiente que influye en otra dependiente, y la segunda no influye en la primera. En otras palabras, X es la variable independiente e Y es la variable dependiente, Y depende de X, pero X no depende de Y.

La regresión o el modelo de regresión no es determinístico (cuando los datos siempre tienen una misma relación), ya que el conjunto de datos obtenidos son variables aleatorias y por lo tanto nos lleva a un proceso estocástico (cuando se desconoce el resultado), lo cual implica un acarreo de error.

4.1.1 Mínimos cuadrados

Aquí se estudia la estimación por el método de mínimos cuadrados para el modelo lineal simple en el que sólo se tiene una variable de predicción, y se supone una ecuación de regresión lineal. Por ejemplo: Los atletas que tienen mayor entrenamiento en trabajar los músculos gemelos son por lo tanto los que tienen un mejor resultado en cuanto a salto de altura se trate, en el ejemplo se toman los primeros 20 elementos de muestra de la encuesta.

Los datos que se encuentran en la siguiente tabla representan la frecuencia de sus entrenamientos y los resultados obtenidos respectivamente, donde para x la frecuencia del entrenamiento es:

4=(Diario),

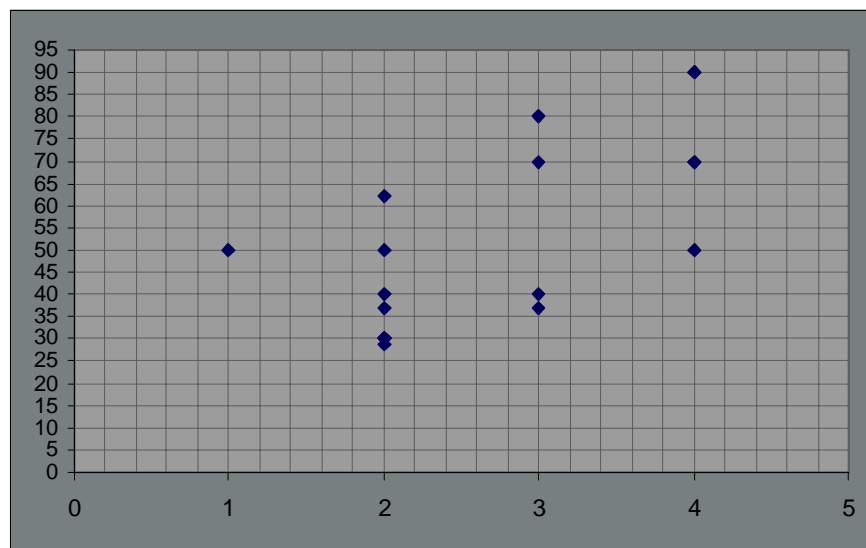
3=(3 veces x semana).

2=(1o2 veces x semana).

1=(nunca).

y para Y la altura obtenida es de: 0 a 1.20 cm.

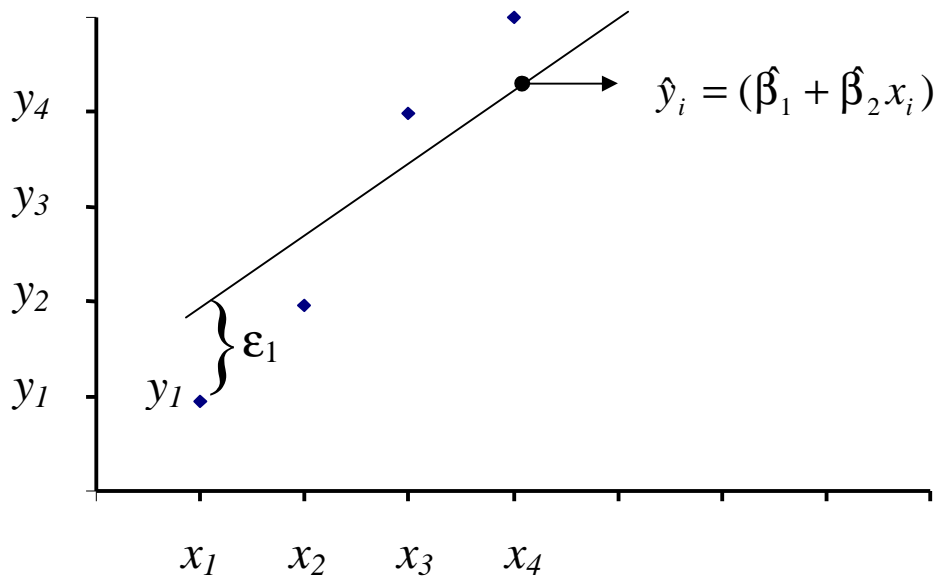
GEMELOS	SALTA
x_i	y_i
4	70
3	80
4	90
2	30
3	70
4	90
3	80
2	50
2	30
4	70
2	30
3	40
2	30
1	50
2	40
4	50
2	29
3	37
2	37
2	62



A pesar de que se muestra una dispersión, también se observa un comportamiento lineal hacia arriba y hacia la derecha. De acuerdo a esto se utilizará un modelo de regresión lineal de la forma $\hat{y}_i = \hat{\beta}_1 + \hat{\beta}_2 x_i + \varepsilon$ tal que $i = 1, 2, \dots, n$

Dicho modelo se conoce como modelo lineal simple porque es lineal en los parámetros, o sea que ningún parámetro aparece como exponente. También es lineal debido a que se tiene sólo una variable de predicción, la cual corresponde a las x_i .

La i -ésima observación de la variable respuesta es y_i , la cual corresponde al i -ésimo valor de x_i de la variable de predicción. ε_i es el error aleatorio que no se observa y que está determinado por la diferencia de y_i y \hat{y}_i , mientras más pequeña sea la diferencia, mejor aproximación tendrá la línea de regresión. β_1 y β_2 representan la intersección y la pendiente respectivamente, sin embargo sus valores son desconocidos hasta el momento.



Se sabe que $\hat{y}_i = (\beta_1 + \beta_2 x_i)$ (ec 4.1) Estimación de y_i

La i -ésima desviación o error es: $\varepsilon_i = y_i - \hat{y}_i$

Sustituyendo a \hat{y}_i , se tiene $\varepsilon_i = y_i - (\beta_1 + \beta_2 x_i)$

Y la suma de los cuadrados de los errores es:

$$\sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - \beta_1 - \beta_2 x_i)^2$$

Se deriva para simplificar la ecuación con respecto β_1 e igualando a cero:

$$\frac{\partial \sum_{i=1}^n \varepsilon_i^2}{\partial \beta_1} = -2 \sum_{i=1}^n (y_i - \beta_1 - \beta_2 x_i) = 0$$

Para simplificar dividir entre 2.

$$-2 \sum_{i=1}^n (y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i) \left(\frac{1}{2} \right) = 0 \left(\frac{1}{2} \right)$$

$$\frac{-2 \sum_{i=1}^n (y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i)}{2} = 0$$

$$-\sum_{i=1}^n (y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i) = 0$$

$$-\sum_{i=1}^n y_i + \hat{\beta}_1 \sum_{i=1}^n 1 + \hat{\beta}_2 \sum_{i=1}^n x_i = 0$$

$$\hat{\beta}_1 \sum_{i=1}^n 1 + \hat{\beta}_2 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i$$

$$\hat{\beta}_1 n + \hat{\beta}_2 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \quad (\text{ec. 4.1.1})$$

Para encontrar a $\hat{\beta}_1$ dividir entre n:

$$\left(\hat{\beta}_1 n + \hat{\beta}_2 \sum_{i=1}^n x_i \right) \frac{1}{n} = \left(\sum_{i=1}^n y_i \right) \frac{1}{n}$$

$$\frac{\hat{\beta}_1 n}{n} + \frac{\hat{\beta}_2 \sum_{i=1}^n x_i}{n} = \frac{\sum_{i=1}^n y_i}{n}$$

$$\hat{\beta}_1 + \frac{\hat{\beta}_2 \sum_{i=1}^n x_i}{n} = \frac{\sum_{i=1}^n y_i}{n}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n y_i}{n} - \frac{\hat{\beta}_2 \sum_{i=1}^n x_i}{n} \quad (\text{ec. 4.2})$$

Por definición se sabe que $\frac{\sum_{i=1}^n y_i}{n} = \bar{y}$ de la misma manera sucede con la suma de X_i , quedando la simplificación como sigue:

$$\hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x}$$

Desarrollo para encontrar a $\hat{\beta}_2$ procediendo de la misma forma anterior se tiene:

$$\epsilon_i = y_i - \hat{y}_i$$

se sustituye el valor de \hat{y}_i , de la ecuación 4.1

$$\epsilon_i = y_i - (\hat{\beta}_1 + \hat{\beta}_2 x_i)$$

Se suman los cuadrados del error

$$\sum_{i=1}^n \varepsilon_i^2 = \sum_{i=1}^n (y_i - \beta_1 - \beta_2 x_i)^2$$

Se saca la parcial con respecto de β_2

y para simplificar la ecuación se iguala a 0:

$$\frac{\partial \sum_{i=1}^n \varepsilon_i^2}{\partial \beta_2} = -2 \sum_{i=1}^n x_i (y_i - \beta_1 - \beta_2 x_i)$$

$$-2 \sum_{i=1}^n x_i (y_i - \beta_1 - \beta_2 x_i) = 0$$

Para simplificar dividir entre 2:

$$-2 \sum_{i=1}^n x_i (y_i - \beta_1 - \beta_2 x_i) \left(\frac{1}{2} \right) = 0 \left(\frac{1}{2} \right)$$

$$-\sum_{i=1}^n x_i (y_i - \beta_1 - \beta_2 x_i) = 0$$

Por distribución se multiplican los términos

$$-\sum_{i=1}^n x_i y_i + \beta_1 \sum_{i=1}^n x_i + \beta_2 \sum_{i=1}^n x_i^2 = 0$$

Se tiene que el despeje es:

$$\beta_1 \sum_{i=1}^n x_i + \beta_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

Sustituyendo $\hat{\beta}_1$ de la ec. 4.2 se tiene:

$$\left(\frac{\sum_{i=1}^n y_i}{n} - \frac{\hat{\beta}_2 \sum_{i=1}^n x_i}{n} \right) \sum_{i=1}^n x_i + \hat{\beta}_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

Haciendo la multiplicación queda:

$$\frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n} - \frac{\hat{\beta}_2 \left(\sum_{i=1}^n x_i \right)^2}{n} + \hat{\beta}_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i$$

$$-\frac{\hat{\beta}_2 \left(\sum_{i=1}^n x_i \right)^2}{n} + \hat{\beta}_2 \sum_{i=1}^n x_i^2 = \sum_{i=1}^n x_i y_i - \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n}$$

Se saca el factor común $\hat{\beta}_2$ del lado izquierdo de la ecuación:

$$\hat{\beta}_2 \left(-\frac{\left(\sum_{i=1}^n x_i\right)^2}{n} + \sum_{i=1}^n x_i^2 \right) = \sum_{i=1}^n x_i y_i - \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n}$$

Se multiplica por n ambas partes de la igualdad para simplificar:

$$\left[\hat{\beta}_2 \left(-\frac{\left(\sum_{i=1}^n x_i\right)^2}{n} + \sum_{i=1}^n x_i^2 \right) \right] (n) = \left[\sum_{i=1}^n x_i y_i - \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n} \right] (n)$$

Se efectua la distribución multiplicativa:

$$\hat{\beta}_2 n \left(-\frac{\left(\sum_{i=1}^n x_i\right)^2}{n} + \sum_{i=1}^n x_i^2 \right) = n \left(\sum_{i=1}^n x_i y_i \right) - n \left(\frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n} \right)$$

$$\hat{\beta}_2 \left(n \left[-\frac{\left(\sum_{i=1}^n x_i\right)^2}{n} \right] + n \left[\sum_{i=1}^n x_i^2 \right] \right) = n \sum_{i=1}^n x_i y_i - \frac{n \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n}$$

$$\hat{\beta}_2 \left(-\frac{n \left(\sum_{i=1}^n x_i\right)^2}{n} + n \sum_{i=1}^n x_i^2 \right) = n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i$$

$$\hat{\beta}_2 \left(-\left(\sum_{i=1}^n x_i\right)^2 + n \sum_{i=1}^n x_i^2 \right) = n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i$$

$$\hat{\beta}_2 \left(- \left(\sum_{i=1}^n x_i \right)^2 + n \sum_{i=1}^n x_i^2 \right) = n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i$$

Reordenando el lado izquierdo de la ecuación:

$$\hat{\beta}_2 \left(n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 \right) = n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i$$

$\hat{\beta}_2 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{\left(n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2 \right)}$	$\hat{\beta}_2 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}$
--	---

Por simplicidad, se pueden omitir los subíndices:

$$\hat{\beta}_2 = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2}$$

∴ Los estimadores son:

$$\hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x} \quad \text{y} \quad \hat{\beta}_2 = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2}$$

Retomando el caso del ejemplo inicial, se necesita sacar algunos totales y medias los cuales son:

$$n, \sum_{i=1}^n x_i, \sum_{i=1}^n y_i, \bar{x}, \bar{y}, \sum_{i=1}^n x_i y_i, \sum_{i=1}^n x_i^2, \sum_{i=1}^n y_i^2$$

o bien:

$$n, \sum x, \sum y, \bar{x}, \bar{y}, \sum xy, \sum x^2, \sum y^2$$

GEMELOS	SALTA			
x_i	y_i	$x_i y_i$	x_i^2	y_i^2
4	70	280	16	4900
3	80	240	9	6400
4	90	360	16	8100
2	30	60	4	900
3	70	210	9	4900
4	90	360	16	8100
3	80	240	9	6400
2	50	100	4	2500
2	30	60	4	900
4	70	280	16	4900
2	30	60	4	900
3	40	120	9	1600
2	30	60	4	900
1	50	50	1	2500
2	40	80	4	1600
4	50	200	16	2500
2	29	58	4	841
3	37	111	9	1369
2	37	74	4	1369
2	62	124	4	3844

$\sum x$	54
$\sum y$	1065
\bar{x}	2.7
\bar{y}	53.25
$\sum xy$	3127
$\sum x^2$	162
$\sum y^2$	65423
n	20

$$\hat{\beta}_2 = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2} \quad \text{y} \quad \hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x}$$

Sustituyendo valores para los estimadores encontrados $\hat{\beta}_1$ y $\hat{\beta}_2$ se tiene:

$$\hat{\beta}_2 = \frac{(20)(3127) - (54)(1065)}{(20)(162) - (54)^2}$$

$$\hat{\beta}_2 = \frac{62540 - 57510}{3240 - 2916}$$

$$\hat{\beta}_2 = \frac{5030}{324}$$

$$\hat{\beta}_2 = 15.5246914$$

$$\hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x}$$

$$\hat{\beta}_1 = 53.25 - (15.5246914)(2.7)$$

$$\hat{\beta}_1 = 53.25 - 41.91666678$$

$$\hat{\beta}_1 = 11.3333333$$

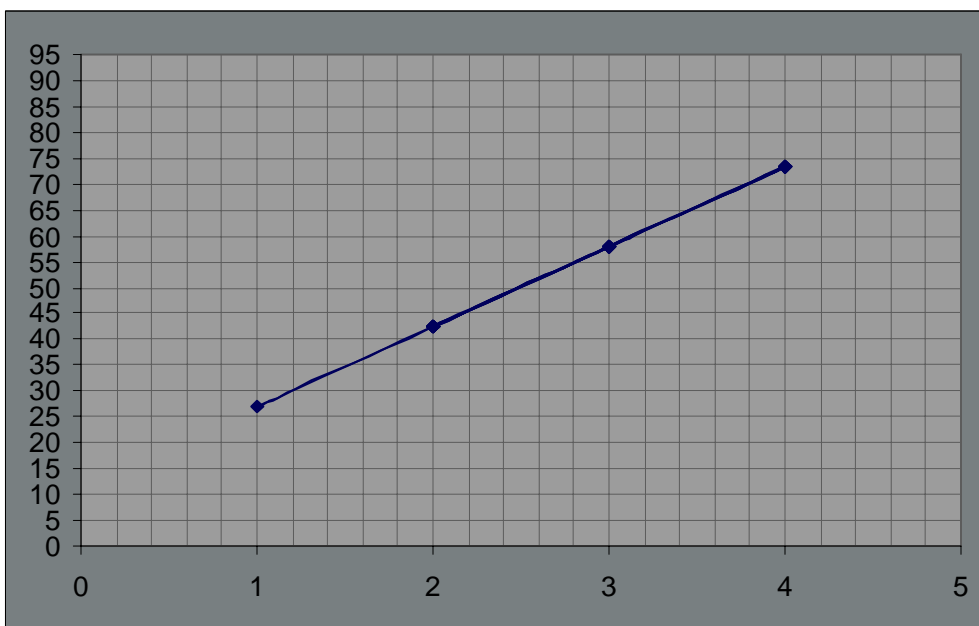
Encontrando los valores de \hat{y}_i estimado, en cada una de las x_i 's se encuentra la columna de \hat{y}_i . Posteriormente se grafica las x_i como independientes y \hat{y}_i estimado como dependientes para encontrar la línea de regresión como se muestra más abajo.

$$\hat{y}_i = (\beta_1 + \beta_2 x_i)$$

$$\hat{y}_i = (11.3333 + 15.524x_i)$$

GEMELOS	SALTA				ESTIMADO
x_i	y_i	$x_i y_i$	x_i^2	y_i^2	\hat{y}_i
4	70	280	16	4900	73.4321
3	80	240	9	6400	57.9074
4	90	360	16	8100	73.4321
2	30	60	4	900	42.3827
3	70	210	9	4900	57.9074
4	90	360	16	8100	73.4321
3	80	240	9	6400	57.9074
2	50	100	4	2500	42.3827
2	30	60	4	900	42.3827
4	70	280	16	4900	73.4321
2	30	60	4	900	42.3827
3	40	120	9	1600	57.9074
2	30	60	4	900	42.3827
1	50	50	1	2500	26.858
2	40	80	4	1600	42.3827
4	50	200	16	2500	73.4321
2	29	58	4	841	42.3827
3	37	111	9	1369	57.9074
2	37	74	4	1369	42.3827
2	62	124	4	3844	42.3827

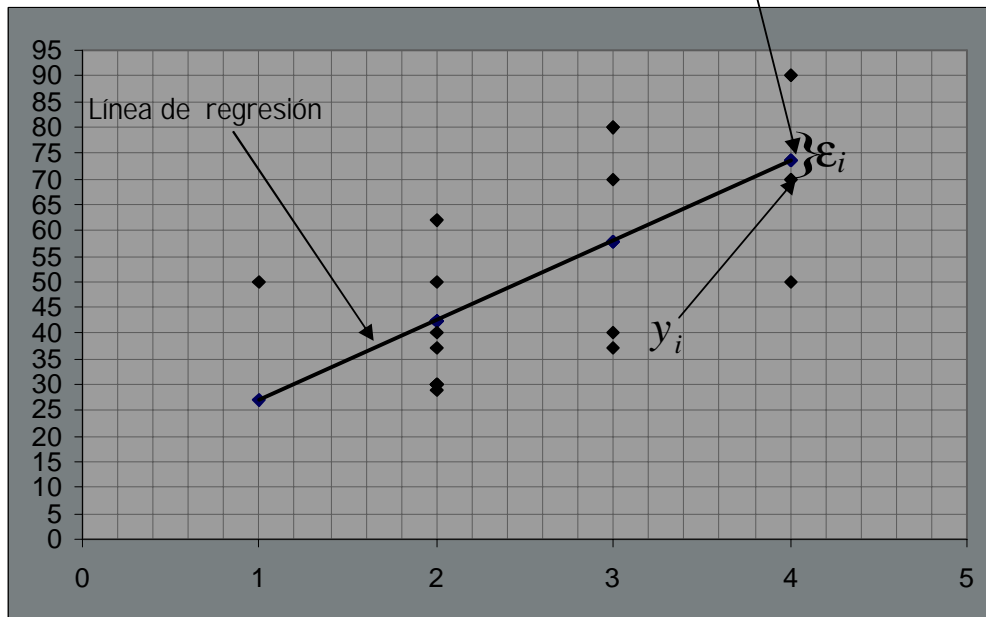
$\sum x$	54
$\sum y$	1065
\bar{x}	2.7
\bar{y}	53.25
$\sum xy$	3127
$\sum x^2$	162
$\sum y^2$	65423
n	20
β_1	11.3333
β_2	15.5247



Ahora se saca el residuo de las y_i menos las \hat{y}_i y al resultado se le saca su cuadrado para encontrar el error de la distancia vertical entre las y_i y la línea de regresión como se muestra en la siguiente tabla:

GEMELOS	SALTA				ESTIMADO	RESIDUO	CUADRAD RESIDUO
x_i	y_i	$x_i y_i$	x_i^2	y_i^2	\hat{y}_i	$y_i - \hat{y}_i$	$(y_i - \hat{y}_i)^2$
4	70	280	16	4900	73.4321	-3.4321	11.7793
3	80	240	9	6400	57.9074	22.0926	488.083
4	90	360	16	8100	73.4321	16.5679	274.495
2	30	60	4	900	42.3827	-12.383	153.332
3	70	210	9	4900	57.9074	12.0926	146.231
4	90	360	16	8100	73.4321	16.5679	274.495
3	80	240	9	6400	57.9074	22.0926	488.083
2	50	100	4	2500	42.3827	7.61728	58.023
2	30	60	4	900	42.3827	-12.383	153.332
4	70	280	16	4900	73.4321	-3.4321	11.7793
2	30	60	4	900	42.3827	-12.383	153.332
3	40	120	9	1600	57.9074	-17.907	320.675
2	30	60	4	900	42.3827	-12.383	153.332
1	50	50	1	2500	26.858	23.142	535.551
2	40	80	4	1600	42.3827	-2.3827	5.67734
4	50	200	16	2500	73.4321	-23.432	549.063
2	29	58	4	841	42.3827	-13.383	179.097
3	37	111	9	1369	57.9074	-20.907	437.12
2	37	74	4	1369	42.3827	-5.3827	28.9736
2	62	124	4	3844	42.3827	19.6173	384.838

$$\hat{y}_i = (11.3333333 + 15.5246914 x_i)$$



Este método será utilizado para la investigación de campo en todos los entrenamientos de salto y poder determinar cuales son los más eficaces en la ejecución deportiva.

4.1.2 Ajuste Exponencial

Para determinar si los entrenamientos tienen un comportamiento en sus resultados de forma exponencial, se ajusta la muestra a una curva exponencial, teniendo primeramente que la función exponencial es:

$$\hat{y}_i = ab^{x_i} \text{ la que con los estimadores } \hat{\beta}_1 \text{ y } \hat{\beta}_2 \text{ se expresa: } \hat{y}_i = \hat{\beta}_1 \hat{\beta}_2^{x_i}$$

Como lo sugiere el nombre exponencial, la variable x aparece como exponente, luego se toman logaritmos para transformar la ecuación y encontrar posteriormente los antilogaritmos para encontrar los valores de los estimadores.

Se procede como sigue:

$$\log \hat{y} = \log \hat{\beta}_1 + (\log \hat{\beta}_2)x$$

Partiendo de las ecuaciones de mínimos cuadrados ya desarrolladas se tiene que:

$$\hat{\beta}_2 = \frac{n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2}$$

$$\log \hat{\beta}_2 = \frac{n \sum x \log y - \sum x \sum \log y}{n \sum x^2 - [\sum x]^2}$$

$$\hat{\beta}_2 = \text{anti log} \left(\frac{n \sum x \log y - \sum x \sum \log y}{n \sum x^2 - [\sum x]^2} \right)$$

$$\hat{\beta}_2 = 10^{\left(\frac{n \sum x \log y - \sum x \sum \log y}{n \sum x^2 - [\sum x]^2} \right)}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n y_i}{n} - \frac{\hat{\beta}_2 \sum_{i=1}^n x_i}{n}$$

$$\log \hat{\beta}_1 = \frac{\sum \log y}{n} - \log \hat{\beta}_2 \frac{\sum x}{n}$$

$$\hat{\beta}_1 = \text{anti log} \left(\frac{\sum \log y}{n} - \log \hat{\beta}_2 \frac{\sum x}{n} \right)$$

$$\hat{\beta}_1 = 10^{\left(\frac{\sum \log y}{n} - \log \hat{\beta}_2 \frac{\sum x}{n} \right)}$$

GEMELOS	SALTA					
x_i	y_i	x_i^2	$\log y_i$	$x_i \log y_i$	$\hat{y}_i = \beta_1 \beta_2^{x_i}$	
1	50	1	1.6990	1.6990	30.2208	$\sum x$ 54
2	30	4	1.4771	2.9542	40.2852	$\sum y$ 1065
2	50	4	1.6990	3.3979	40.2852	$\sum x^2$ 162
2	30	4	1.4771	2.9542	40.2852	$\sum \log y_i$ 33.850667
2	30	4	1.4771	2.9542	40.2852	$\sum x_i \log y_i$ 93.419206
2	30	4	1.4771	2.9542	40.2852	n 20
2	40	4	1.6021	3.2041	40.2852	
2	29	4	1.4624	2.9248	40.2852	
2	37	4	1.5682	3.1364	40.2852	
2	62	4	1.7924	3.5848	40.2852	
3	80	9	1.9031	5.7093	53.7014	
3	70	9	1.8451	5.5353	53.7014	
3	80	9	1.9031	5.7093	53.7014	
3	40	9	1.6021	4.8062	53.7014	
3	37	9	1.5682	4.7046	53.7014	
4	70	16	1.8451	7.3804	71.5855	
4	90	16	1.9542	7.8170	71.5855	
4	90	16	1.9542	7.8170	71.5855	
4	70	16	1.8451	7.3804	71.5855	
4	50	16	1.6990	6.7959	71.5855	

$$\hat{\beta}_2 = 10^{\left(\frac{n \sum x \log y - \sum x \sum \log y}{n \sum x^2 - [\sum x]^2} \right)}$$

$$\hat{\beta}_2 = 10^{\left(\frac{(20)(93.419206) - (54)(33.850667)}{(20)(162) - [54]^2} \right)}$$

$$\hat{\beta}_2 = 10^{(0.12483982)}$$

$$\hat{\beta}_2 = 1.33302969$$

$$\hat{\beta}_1 = 10^{\left(\frac{\sum \log y - \log \hat{\beta}_2 \sum x}{n} \right)}$$

$$\hat{\beta}_1 = 10^{\left(\frac{33.850667 - (0.12483982) \frac{54}{20}}{20} \right)}$$

$$\hat{\beta}_1 = 10^{(1.3554659)}$$

$$\hat{\beta}_1 = 22.67075$$

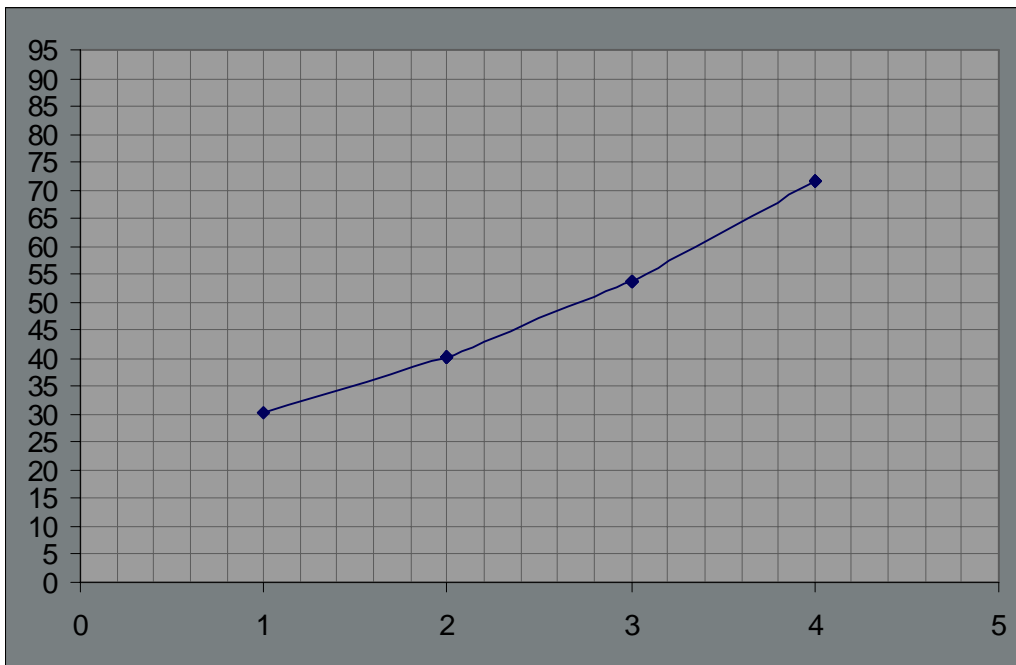
El ajuste $\hat{y}_i = \hat{\beta}_1 \hat{\beta}_2^{x_i}$

toma los valores

$$\hat{y}_i = (22.6)(1.33)^{x_i}$$

La curva exponencial queda de la siguiente forma

GEMELOS	SALTA					
x_i	y_i	x_i^2	$\log y_i$	$x_i \log y_i$	$\hat{y}_i = \beta_1 \beta_2^{x_i}$	
1	50	1	1.6990	1.6990	30.2208	$\sum x$ 54
2	30	4	1.4771	2.9542	40.2852	$\sum y$ 1065
2	50	4	1.6990	3.3979	40.2852	$\sum x^2$ 162
2	30	4	1.4771	2.9542	40.2852	$\sum \log y_i$ 33.850667
2	30	4	1.4771	2.9542	40.2852	$\sum x_i \log y_i$ 93.419206
2	30	4	1.4771	2.9542	40.2852	n 20
2	40	4	1.6021	3.2041	40.2852	$\log \beta_2$ 0.1248398
2	29	4	1.4624	2.9248	40.2852	$\log \beta_1$ 1.3554659
2	37	4	1.5682	3.1364	40.2852	β_2 1.3330296
2	62	4	1.7924	3.5848	40.2852	β_1 22.67075
3	80	9	1.9031	5.7093	53.7014	
3	70	9	1.8451	5.5353	53.7014	
3	80	9	1.9031	5.7093	53.7014	
3	40	9	1.6021	4.8062	53.7014	
3	37	9	1.5682	4.7046	53.7014	
4	70	16	1.8451	7.3804	71.5855	
4	90	16	1.9542	7.8170	71.5855	
4	90	16	1.9542	7.8170	71.5855	
4	70	16	1.8451	7.3804	71.5855	
4	50	16	1.6990	6.7959	71.5855	



4.1.3 Ajuste logarítmico

Ahora bien, para observar si los entrenamientos se ajustan a una curva logarítmica, se sabe que la función logarítmica se comporta de manera inversa de la exponencial. La función logarítmica tiene la siguiente forma:

$$\hat{y}_i = a + b \log x_i \quad \text{la que con los estimadores queda: } \hat{y}_i = \hat{\beta}_1 + \hat{\beta}_2 \log x_i$$

Como en el caso anterior, se toma primeramente la fórmula de mínimos cuadrados para los estimadores:

$$\hat{\beta}_2 = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2}$$

$$\hat{\beta}_2 = \frac{n \sum (\log x)(y) - \sum \log x \sum y}{n \sum (\log x)^2 - (\sum \log x)^2}$$

$$\hat{\beta}_1 = \bar{y} - \hat{\beta}_2 \bar{x}$$

$$\hat{\beta}_1 = \frac{\sum y}{n} - \hat{\beta}_2 \frac{\sum x}{n}$$

$$\hat{\beta}_1 = \frac{\sum y}{n} - \hat{\beta}_2 \frac{\sum \log x}{n}$$

gemelos	salta					
x_i	y_i	$\log x_i$	$(\log x_i)^2$	$(\log x_i)(y_i)$	$\sum (\log x)(y)$	470.9866
1	50	0.0000	0.0000	0.0000	$\sum x$	54
2	30	0.3010	0.0906	9.0309	$\sum y$	1065
2	50	0.3010	0.0906	15.0515	$\sum \log x$	8.1052
2	30	0.3010	0.0906	9.0309	$\sum (\log x)^2$	3.7662
2	30	0.3010	0.0906	9.0309	n	20
2	30	0.3010	0.0906	9.0309		
2	40	0.3010	0.0906	12.0412		
2	29	0.3010	0.0906	8.7299		
2	37	0.3010	0.0906	11.1381		
2	62	0.3010	0.0906	18.6639		
3	80	0.4771	0.2276	38.1697		
3	70	0.4771	0.2276	33.3985		
3	80	0.4771	0.2276	38.1697		
3	40	0.4771	0.2276	19.0849		
3	37	0.4771	0.2276	17.6535		
4	70	0.6021	0.3625	42.1442		
4	90	0.6021	0.3625	54.1854		
4	90	0.6021	0.3625	54.1854		
4	70	0.6021	0.3625	42.1442		
4	50	0.6021	0.3625	30.1030		

$$\hat{\beta}_2 = \frac{n \sum (\log x)(y) - \sum \log x \sum y}{n \sum (\log x)^2 - (\sum \log x)^2}$$

$$\hat{\beta}_2 = \frac{(20)(470.9866) - (8.1052)(106.5)}{(20)(3.7662) - (8.1052)^2}$$

$$\hat{\beta}_2 = \frac{787.719}{9.62964}$$

$$\hat{\beta}_2 = 81.80144$$

$$\hat{\beta}_1 = \frac{\sum y}{n} - \hat{\beta}_2 \frac{\sum \log x}{n}$$

$$\hat{\beta}_1 = \frac{1065}{20} - \hat{\beta}_2 \frac{8.1052}{20}$$

$$\hat{\beta}_1 = \frac{1065}{20} - (81.801) \frac{8.1052}{20}$$

$$\hat{\beta}_1 = 20.09925$$

gemelos	salta					
x_i	y_i	$\log x_i$	$(\log x_i)^2$	$(\log x_i)(y_i)$	$\hat{y} = \hat{\beta}_1 + \hat{\beta}_2 \log x$	
1	50	0.0000	0.0000	0.0000	20.0992	
2	30	0.3010	0.0906	9.0309	44.7239	
2	50	0.3010	0.0906	15.0515	44.7239	
2	30	0.3010	0.0906	9.0309	44.7239	
2	30	0.3010	0.0906	9.0309	44.7239	
2	30	0.3010	0.0906	9.0309	44.7239	
2	40	0.3010	0.0906	12.0412	44.7239	
2	29	0.3010	0.0906	8.7299	44.7239	
2	37	0.3010	0.0906	11.1381	44.7239	
2	62	0.3010	0.0906	18.6639	44.7239	
3	80	0.4771	0.2276	38.1697	59.1285	
3	70	0.4771	0.2276	33.3985	59.1285	
3	80	0.4771	0.2276	38.1697	59.1285	
3	40	0.4771	0.2276	19.0849	59.1285	
3	37	0.4771	0.2276	17.6535	59.1285	
4	70	0.6021	0.3625	42.1442	69.3486	
4	90	0.6021	0.3625	54.1854	69.3486	
4	90	0.6021	0.3625	54.1854	69.3486	
4	70	0.6021	0.3625	42.1442	69.3486	
4	50	0.6021	0.3625	30.1030	69.3486	

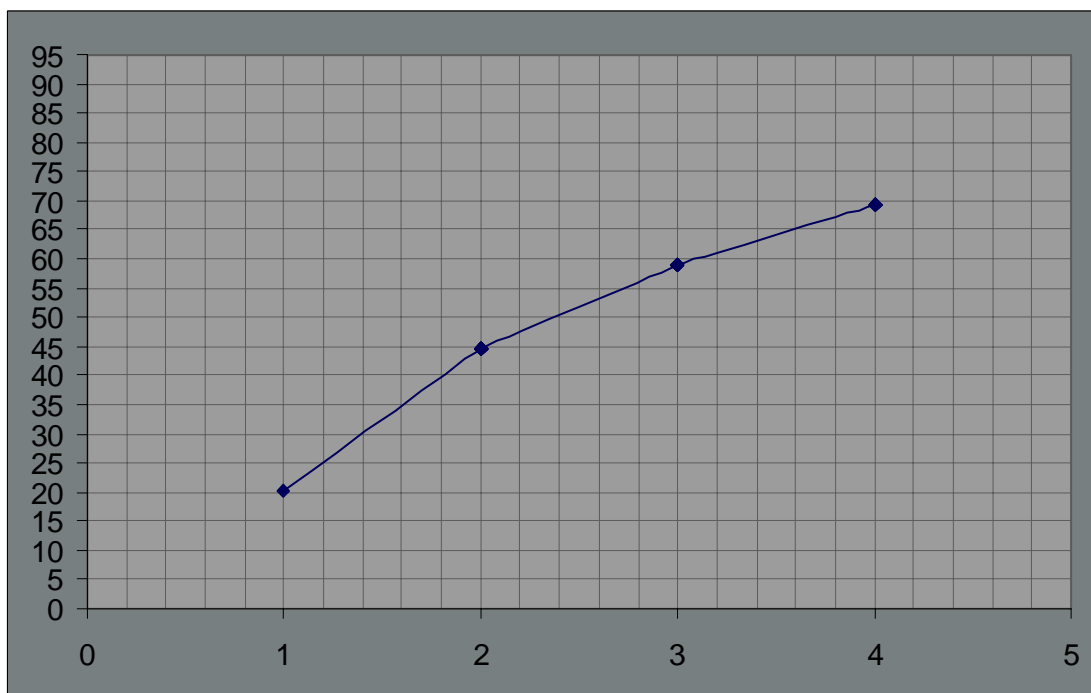
El ajuste $\hat{y}_i = \hat{\beta}_1 + \hat{\beta}_2 \log x_i$

toma los valores

$$\hat{y}_i = (20.09) + (81.80) \log x_i$$

$\sum (\log x)(y)$	470.9866
$\sum x$	54
$\sum y$	1065
$\sum \log x$	8.1052
$\sum (\log x)^2$	3.7662
n	20
$\hat{\beta}_2$	81.80144
$\hat{\beta}_1$	20.09925

El ajuste a la curva logarítmica queda como sigue:



4.1.4 Ajuste cuadrático

Como se puede observar en el modelo anterior, los datos observados pueden no ajustarse necesariamente a la curva del modelo logarítmico, puesto que el cuerpo humano no responde siempre de manera logarítmica al recibir entrenamientos cada vez más frecuentes, como es el entrenamiento diario de un mismo ejercicio, no tiende a estabilizarse. Ahora bien cuando se sufre un sobreentrenamiento el cuerpo tiende a responder de manera negativa. Es decir que el sobreentrenamiento es contraproducente. Por esta razón se ajustarán los datos obtenidos a una curva de modelo cuadrático. Procediendo como en la regresión lineal, se ajusta la muestra como sigue en el modelo cuadrático:

$$\hat{y}_i = a + bx_i + cx_i^2 \text{ en donde los estimadores son: } \hat{y}_i = \hat{\beta}_1 + \hat{\beta}_2 x_i + \hat{\beta}_3 x_i^2$$

Los estimadores se calculan por el método de mínimos cuadrados partiendo de la ec. 4.1.1:

$$\hat{\beta}_1 n + \hat{\beta}_2 \sum_{i=1}^n x_i = \sum_{i=1}^n y_i \text{ simplificando subíndices: } \hat{\beta}_1 n + \hat{\beta}_2 \sum x = \sum y$$

Agregando el 3er término cx^2 donde el estimador es: $\hat{\beta}_3 x^2$, se usa un sistema de ecuaciones:

$$\hat{\beta}_1 n + \hat{\beta}_2 \sum x + \hat{\beta}_3 \sum x^2 = \sum y$$

$$\hat{\beta}_1 20 + \hat{\beta}_2 54 + \hat{\beta}_3 162 = 1065$$

$$\hat{\beta}_1 \sum x + \hat{\beta}_2 \sum x^2 + \hat{\beta}_3 \sum x^3 = \sum xy$$

$$\hat{\beta}_1 54 + \hat{\beta}_2 162 + \hat{\beta}_3 528 = 3127$$

$$\hat{\beta}_1 \sum x^2 + \hat{\beta}_2 \sum x^3 + \hat{\beta}_3 \sum x^4 = \sum x^2 y$$

$$\hat{\beta}_1 162 + \hat{\beta}_2 528 + \hat{\beta}_3 1830 = 10085$$

Utilizando una matriz para encontrar los valores de los estimadores por el método de Gauss-Jordan:

$$\left[\begin{array}{ccc|c} 20 & 54 & 162 & 1065 \\ 54 & 162 & 528 & 3127 \\ 162 & 528 & 1830 & 10085 \end{array} \right] 1/20R_1$$

$$\left[\begin{array}{ccc|c} 1 & 2.7 & 8.1 & 53.25 \\ 54 & 162 & 528 & 3127 \\ 162 & 528 & 1830 & 10085 \end{array} \right] \begin{array}{l} -54R_1 + R_2 \\ -162R_1 + R_3 \end{array}$$

$$\left[\begin{array}{ccc|c} 1 & 2.7 & 8.1 & 53.25 \\ 0 & 16.2 & 90.6 & 251.5 \\ 0 & 90.6 & 517.8 & 1458.5 \end{array} \right] 1/16.2R_2$$

$$\left[\begin{array}{ccc|c} 1 & 2.7 & 8.1 & 53.25 \\ 0 & 1 & 5.59259 & 15.5247 \\ 0 & 90.6 & 517.8 & 1458.5 \end{array} \right] \begin{array}{l} -2.7R_2 + R_1 \\ -90.6R_2 + R_3 \end{array}$$

$$\left[\begin{array}{ccc|c} 1 & 0 & -7 & 11.3333 \\ 0 & 1 & 5.59259 & 15.5247 \\ 0 & 0 & 11.1111 & 51.963 \end{array} \right] 1/11.1111R_3$$

$$\left[\begin{array}{ccc|c} 1 & 0 & -7 & 11.3333 \\ 0 & 1 & 5.59259 & 15.5247 \\ 0 & 0 & 1 & 4.67667 \end{array} \right] \begin{array}{l} 7R_3 + R_1 \\ -5.59259R_3 + R_2 \end{array}$$

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 44.07 \\ 0 & 1 & 0 & -10.63 \\ 0 & 0 & 1 & 4.67667 \end{array} \right] \quad \left[\begin{array}{ccc|c} \beta_1 & 0 & 0 & 44.07 \\ 0 & \beta_2 & 0 & -10.63 \\ 0 & 0 & \beta_3 & 4.67667 \end{array} \right]$$

$$\hat{\beta}_1 = 44.07 \quad \hat{\beta}_2 = -10.63 \quad \hat{\beta}_3 = 4.67667$$

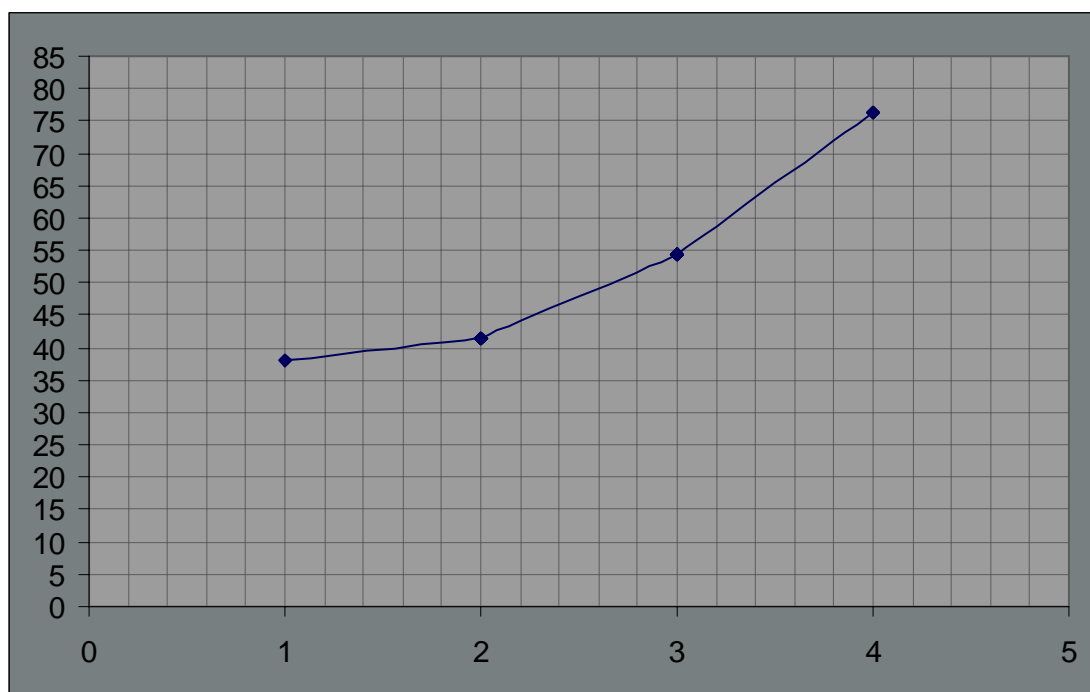
$$\hat{y} = \hat{\beta}_1 + \hat{\beta}_2 x + \hat{\beta}_3 x^2$$

$$\therefore \hat{y} = 44.07 - 10.63x + 4.67667x^2$$

gemelos	salta						
x_i	y_i	x_i^2	x_i^3	x_i^4	$x_i y_i$	$x_i^2 y_i$	$\hat{y} = \hat{\beta}_1 + \hat{\beta}_2 x + \hat{\beta}_3 x^2$
1	50	1	1	1	50	50	38.1167
2	30	4	8	16	60	120	41.5167
2	50	4	8	16	100	200	41.5167
2	30	4	8	16	60	120	41.5167
2	30	4	8	16	60	120	41.5167
2	30	4	8	16	60	120	41.5167
2	40	4	8	16	80	160	41.5167
2	29	4	8	16	58	116	41.5167
2	37	4	8	16	74	148	41.5167
2	62	4	8	16	124	248	41.5167
3	80	9	27	81	240	720	54.2700
3	70	9	27	81	210	630	54.2700
3	80	9	27	81	240	720	54.2700
3	40	9	27	81	120	360	54.2700
3	37	9	27	81	111	333	54.2700
4	70	16	64	256	280	1120	76.3767
4	90	16	64	256	360	1440	76.3767
4	90	16	64	256	360	1440	76.3767
4	70	16	64	256	280	1120	76.3767
4	50	16	64	256	200	800	76.3767

$\sum x$	54
$\sum y$	1065
$\sum x^2$	162
$\sum x^3$	528
$\sum x^4$	1830
$\sum xy$	3127
$\sum x^2y$	10085
n	20
$\hat{\beta}_3$	4.67667
$\hat{\beta}_2$	-10.63
$\hat{\beta}_1$	44.07


La función cuadrática ajustada a la línea de regresión quedará como se muestra.



4.1.5 Investigación de Campo

En la investigación se toma el apoyo de una encuesta llevada a cabo en seis diferentes universidades de la Ciudad de México y área metropolitana, tomando como fuente de información a los seleccionados de basquetbol de cada institución. Así pues, se podrá estudiar el comportamiento de los entrenamientos que contribuyan a mejorar el salto vertical y determinar cuáles dan un mejor resultado.

La encuesta tiene el siguiente formato:

<p>Σ @</p>	<p>Esta investigación pretende encontrar métodos alternativos para mejorar el salto vertical (en suspensión) y parabólico (en carrera) en el baloncesto universitario, con el trabajo biomecánico y del centro de gravedad.</p>			
1 Sexo	<input type="checkbox"/> 1 M	<input type="checkbox"/> 0 F		
2 Tipo racial	<input type="checkbox"/> 3 Afro	<input type="checkbox"/> 2 Moreno	<input type="checkbox"/> 1 Blanco	
3 Compleción	<input type="checkbox"/> 2 Hectomorfo (delgado)	<input type="checkbox"/> 3 Mesomorfo (fornido)	<input type="checkbox"/> 1 Endomorfo (robusto)	
4 Edad	<input type="checkbox"/> 3 26-30	<input type="checkbox"/> 2 21-25	<input type="checkbox"/> 1 18-20	
5 Peso en Kgs.	<input type="text"/> Kgs...			
6 Estatura en metros.	<input type="text"/> Mts.			
7 ¿Te gustaría mejorar tu salto?	<input type="checkbox"/> 1 SI	<input type="checkbox"/> 0 NO		
8 ¿Has logrado retacar un balón en un aro colocado a 3.05 mts?	<input type="checkbox"/> 1 SI	<input type="checkbox"/> 0 NO		
9 ¿aproximadamente cuanto saltas en centímetros?	<input type="text"/> Cm.			
10 Como saltador te consideras	<input type="checkbox"/> 3 excelente	<input type="checkbox"/> 2 Bueno	<input type="checkbox"/> 1 malo	
11 ¿Que tipo de calzado usas?	<input type="checkbox"/> 3 suela normal	<input type="checkbox"/> 1 suela aire	<input type="checkbox"/> 2 nike shox	

En esta primera sección se piden los datos personales de cada encuestado para posteriormente efectuar la filtración de los resultados como por ejemplo: Graficar sólo la rama varonil o sólo la rama femenil, graficar sólo un tipo racial, graficar sólo un tipo de compleción, etc.

En la segunda sección se piden los datos de frecuencia de entrenamientos.

Frecuencia de ejecución de ejercicios					
12	Del toides del antero (Hombros Del anteros)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
13	Gemelos (pantorrillas)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
14	Salto al tablero	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
15	abdominales cortos (flexión corta)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
16	abdominales completos (flexión completa)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
17	Trapezio (Espalda alta)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
18	Dorsales (Espalda baja)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
19	Subir Escaleras	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
20	Trotar con polainas	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
21	Subir Escaleras con polainas	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
22	aparatos p' glúteos	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
23	aparatos p' Cuadriceps (pierna del antera)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
24	aparatos p' bíceps crural (pierna trasera)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
25	Carreras de velocidad - (arrancones)	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
26	Salto con vallas	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
27	Correr sobre las puntas	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
28	Elasticidad	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
29	Isométricos	<input type="checkbox"/> 4 diario	<input type="checkbox"/> 3 días x sem	<input type="checkbox"/> 2 1 o 2 x sem	<input type="checkbox"/> 1 Nunca
gracias por tu colaboración.					a.a.g

Los números que aparecen en las casillas son para identificar las respuestas en cada pregunta y posteriormente capturarlas en el sistema desarrollado para este fin. Una vez graficadas en las regresiones lineales el entrenador podrá tomar ideas métodos para saltar y conocer cuáles rutinas son contraproducentes o bien no hacen ningún efecto en el atleta.

Las encuestas se efectuaron en las siguientes Universidades, en particular a los seleccionados de básquetbol de cada institución:



Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Acatlán
Coordinación de Actividades deportivas
Prof. Víctor González



Universidad Autónoma Metropolitana
Cámpus Azcapotzalco
Actividades deportivas
Prof. Oscar Inclan



Instituto Politécnico Nacional
Cámpus Zacatenco
Actividades deportivas
Prof. David Moreno



Universidad del Valle de México
Cámpus San Rafael
Coordinación de deportes
Prof. Agustín Navarro



Universidad Tecnológica de México
Cámpus Atizapán
Actividades deportivas
Prof. Javier Moctezuma



Tecnológico de Monterrey
Cámpus Ciudad de México
Deportes
Prof. Eugenio Ugalde

4.2 Desarrollo de sistema en Visual Basic y Crystal Reports

4.2.1 Diseño de tablas en Access

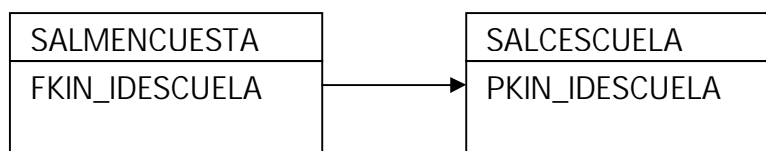
Para alimentar la información de la base de datos de las encuestas se desarrolla un sistema en lenguaje Visual Basic versión 6.0. Para Analizar las curvas de regresión y los ajustes respectivos a dichas curvas se desarrollan también reportes gráficos en Crystal Reports 9.0. Estos dos desarrollos trabajan como un sólo sistema, mismo que al tiempo de irse alimentando con mas encuestas realizadas, podrá mostrar en pantalla e impresora las gráficas de regresión por cada una de las rutinas de ejercicios para salto. La base de datos se crea en el manejador Access 2000 por razón de no ser necesario un manejador Cliente Servidor como SQL Server. La aplicación se desarrolla para trabajar localmente.

Las tablas se componen como sigue:

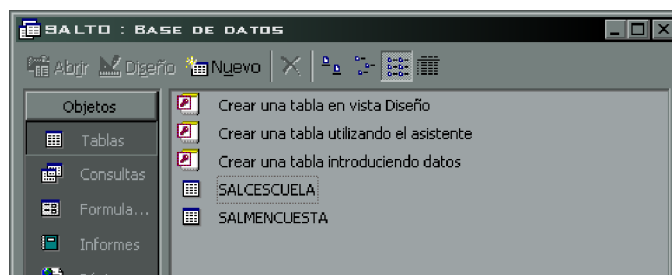
SALMENCUESTA	TABLA MAESTRO DE ENCUESTAS DE SALTO
PKIN_IDENCUEST	PRIMARY KEY INTEGER (LLAVE PRIMARIA VALOR ENTERO) IDENTIFICADOR UNICO DE FOLIO
FKIN_IDESCUELA	FOREIGN KEY INTEGER (LLAVE FORÁNEA VALOR ENTERO) IDENTIFICADOR DE ESCUELA
NNIN_SEXO	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE SEXO
NNIN_RAZA	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE RAZA
NNIN_COMPLEX	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE COMPLEXIÓN
NNIN_EDAD	NOT NULL INTEGER (NO NULO VALOR ENTERO) EDAD EN AÑOS
NNNU_PESO	NOT NULL NUMERIC (NO NULO VALOR NUMERICO CON 3 DECIMALES) PESO EN KILOGRAMOS
NNNU_ESTAT	NOT NULL NUMERIC (NO NULO VALOR NUMERICO CON 3 DECIMALES) ESTATURA EN METROS
NNIN_MEJORAM	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE DESEOS DE MEJORAR
NNIN_ALCANCE	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE ALCANCE DE ARO
NNNU_SALTOAPROX	NOT NULL NUMERIC (NO NULO VALOR NUMERICO CON 3 DECIMALES) SALTO APROXIMADO EN CENTIMETROS
NNIN_SALTADOR	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE TIPO DE SALTADOR
NNIN_CALZADO	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE TIPO DE CALZADO
NNIN_DELTOIDES	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA MUSCULOS DELTOIDES
NNIN_GEMELOS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA MUSCULOS GEMELOS
NNIN_TABLERO	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA SALTOS AL TABLERO
NNIN_ABDMCORT	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA ABDOMINALES CORTOS
NNIN_ABDMCOMP	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA ABDOMINALES COMPLETOS

NNIN_TRAPECIO	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA MÚSCULO TRAPECIO
NNIN_DORSAL	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA MUSCULOS DORSALES
NNIN_ESCALERAS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA SUBIR ESCALERAS
NNIN_POLAINAS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA TROTAR CON POLAINAS
NNIN_ESCPOL	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA ESCALERAS CON POLAINAS
NNIN_GLÚTEOS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA MUSCULOS GLÚTEOS
NNIN_CUADRICEPS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA MÚSCULOS CUADRICEPS
NNIN_BÍCEPS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA MUSCULOS BÍCEPS
NNIN_VELOCIDAD	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA CARRERAS DE VELOCIDAD
NNIN_VALLAS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA SALTOS CON VALLAS
NNIN_PUNTAS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA CORRER SOBRE LAS PUNTAS
NNIN_ELASTICIDAD	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA ELASTICIDAD
NNIN_ISOMETRICOS	NOT NULL INTEGER (NO NULO VALOR ENTERO) IDENTIFICADOR DE FRECUENCIA PARA ISOMÉTRICOS
SALCESCUOLA	TABLA CATALOGO DE ESCUELAS ENCUESTADAS PARA SALTO
PKIN_IDESCUELA	PRIMARY KEY (LLAVE PRIMARIA) IDENTIFICADOR DE ESCUELA
NNVC_NOMBRE	NOT NULL VARCHAR (NO NULO CADENA DE CARACTERES) NOMBRE DE UNIVERSIDAD

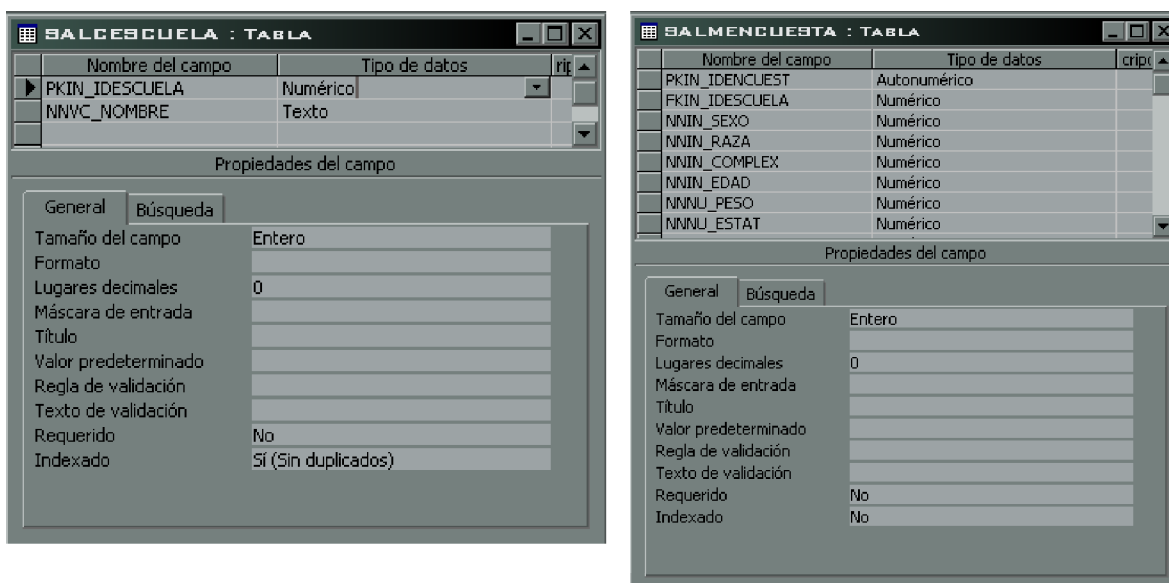
La relación de las tablas en el modelo Entidad-Relación se encuentra de la forma:



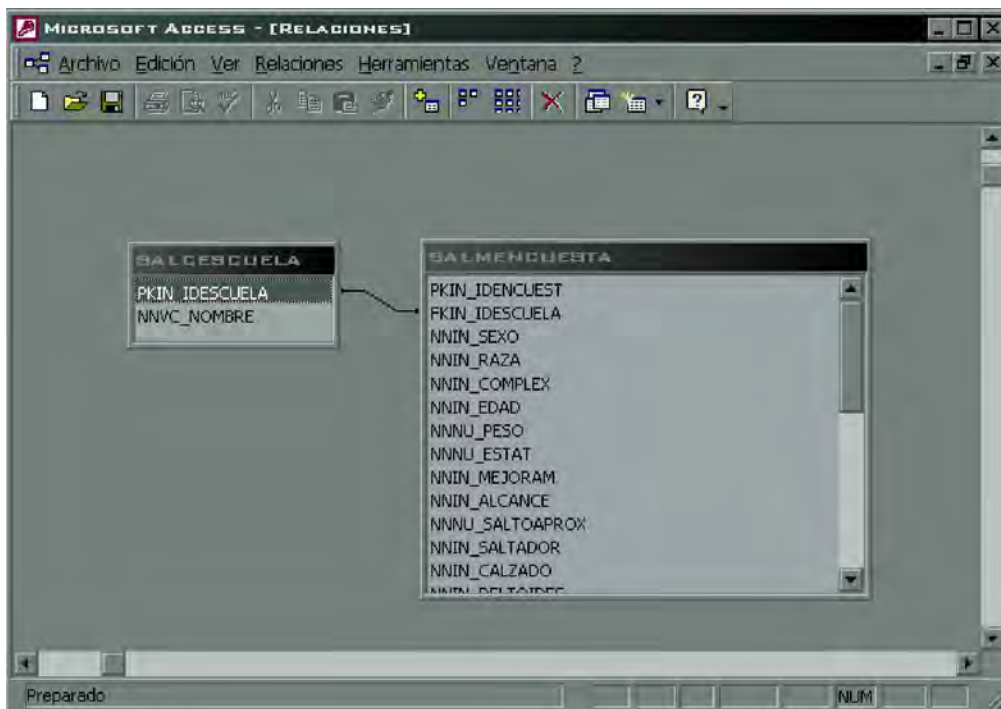
Cada registro encuestado tiene una llave foránea que hace referencia a la escuela de donde proviene. En la parte inferior se muestra la base de datos de SALTO:



El diseño de la tabla de catálogo de escuelas y el diseño de la tabla de maestro de encuestas:

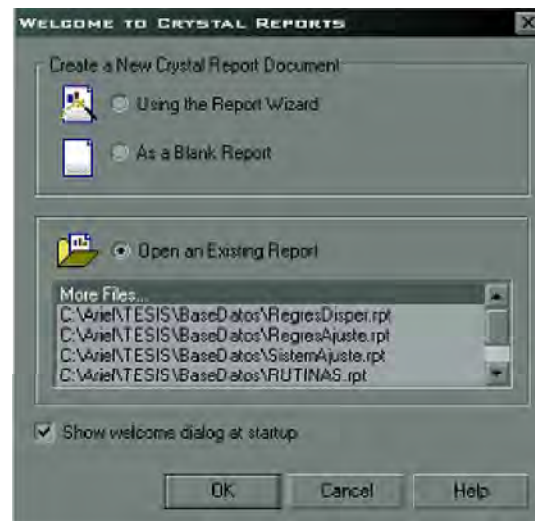
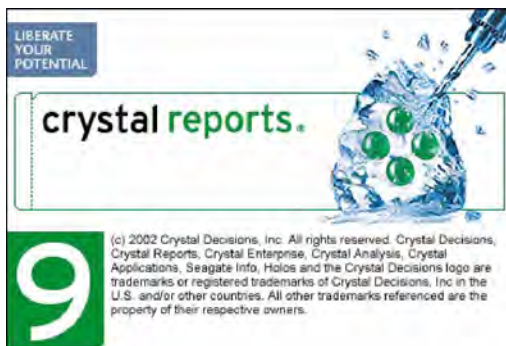


La relación de las tablas:

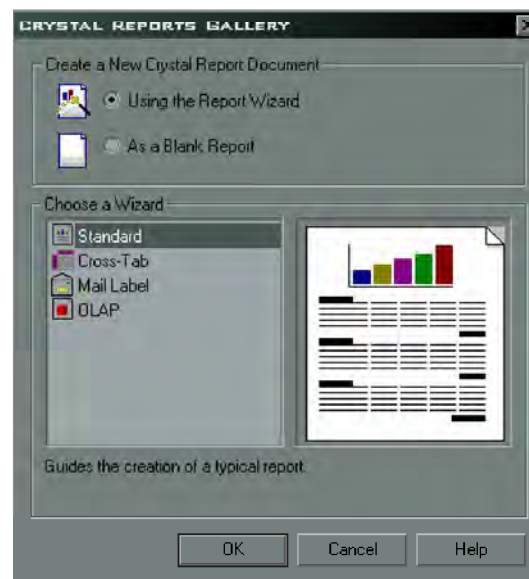
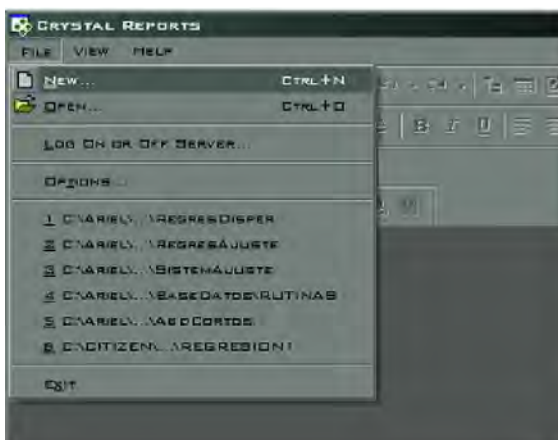


4.2.2 Desarrollo de reportes gráficos de regresión

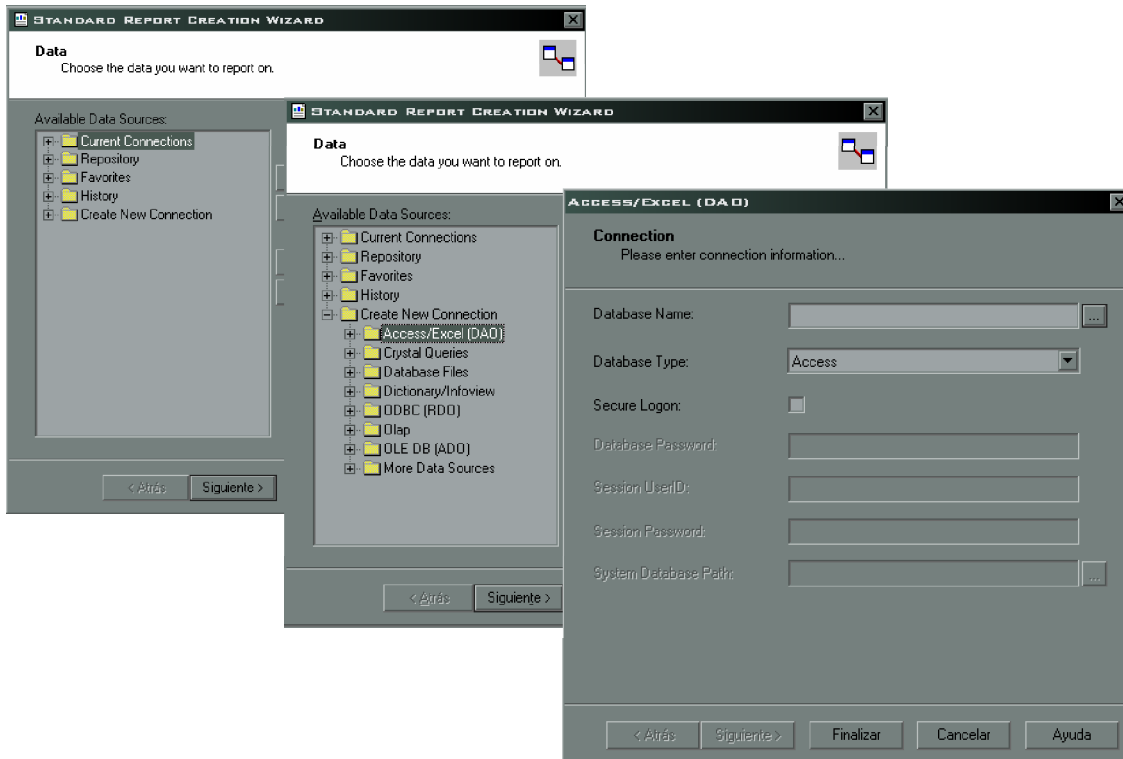
Los reportes graficadores de la dispersión, regresión y el ajuste de los datos como ya es sabido en este tema, se desarrollan en Crystal Reports 9.0. Ahora bien, Crystal no tiene gráficos prediseñados para la regresión, tampoco para los ajustes lineales, por lo que cada una de las fórmulas se hace dentro del propio desarrollo parte por parte, y una vez teniendo todas las fórmulas terminadas sólo se elegirán dos columnas a graficar x e y para la dispersión y por otro lado x e \hat{y} para la regresión lineal y los ajustes. Como dato adicional Crystal Reports sólo maneja logaritmos naturales por lo que en las fórmulas del desarrollo se hacen algunas conversiones y equivalencias. Para comenzar a enlazar el reporte de dispersión de los datos primeramente se abre Crystal Reports y aparece la pantalla de desarrollos recientes:



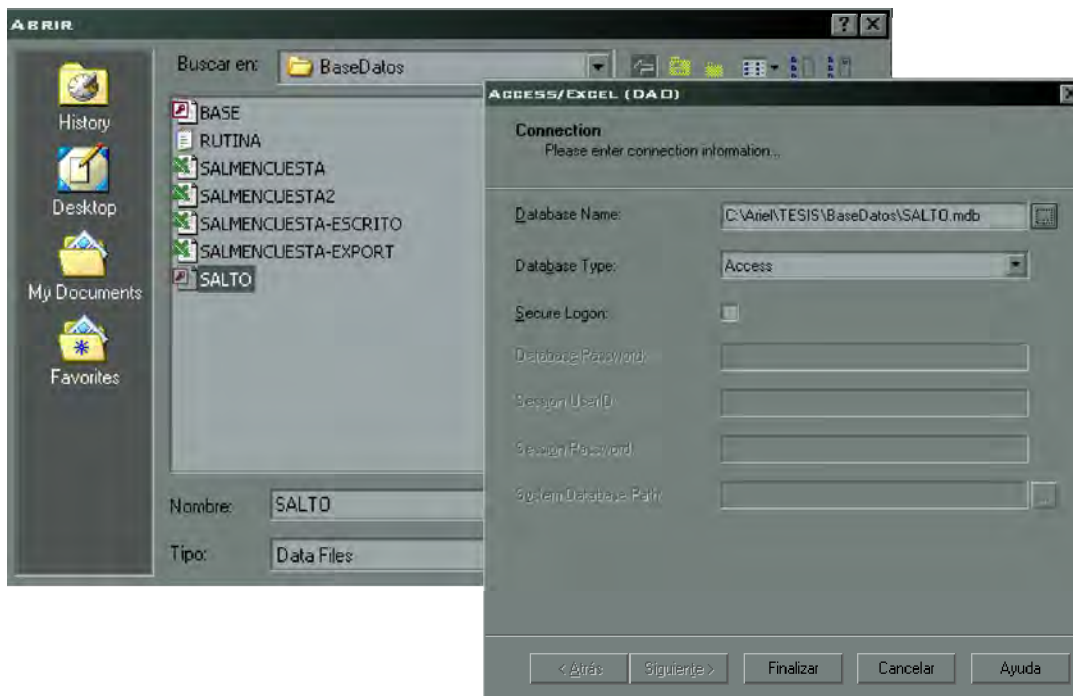
Donde se oprime el botón cancelar (Cancel) y posteriormente se selecciona de la barra de menús la opción [File+New]. Luego se selecciona: [Using the Report Wizard].



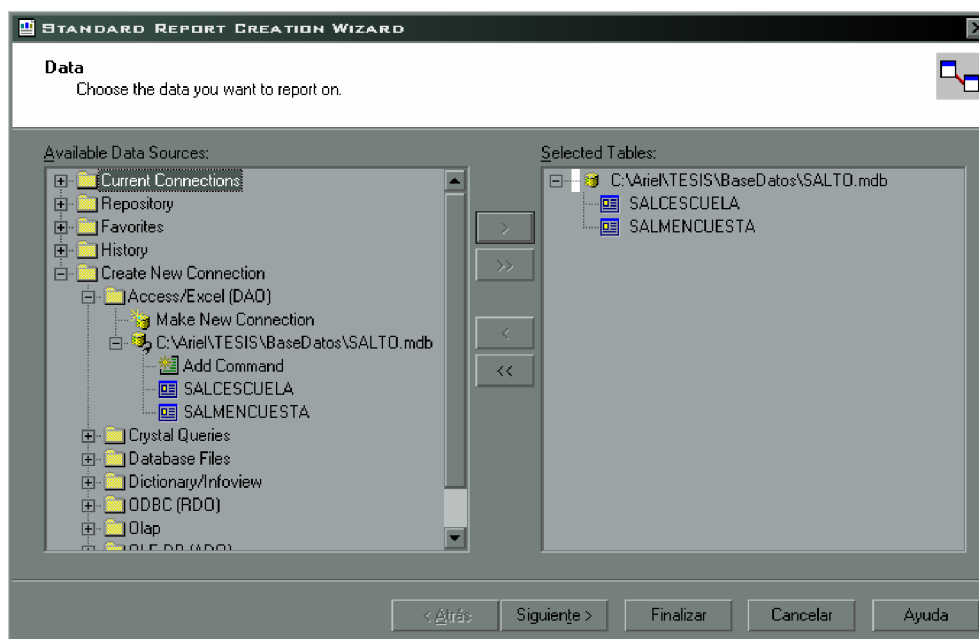
Se selecciona Crear nueva conexión de Access/Excel [Create New Connection]



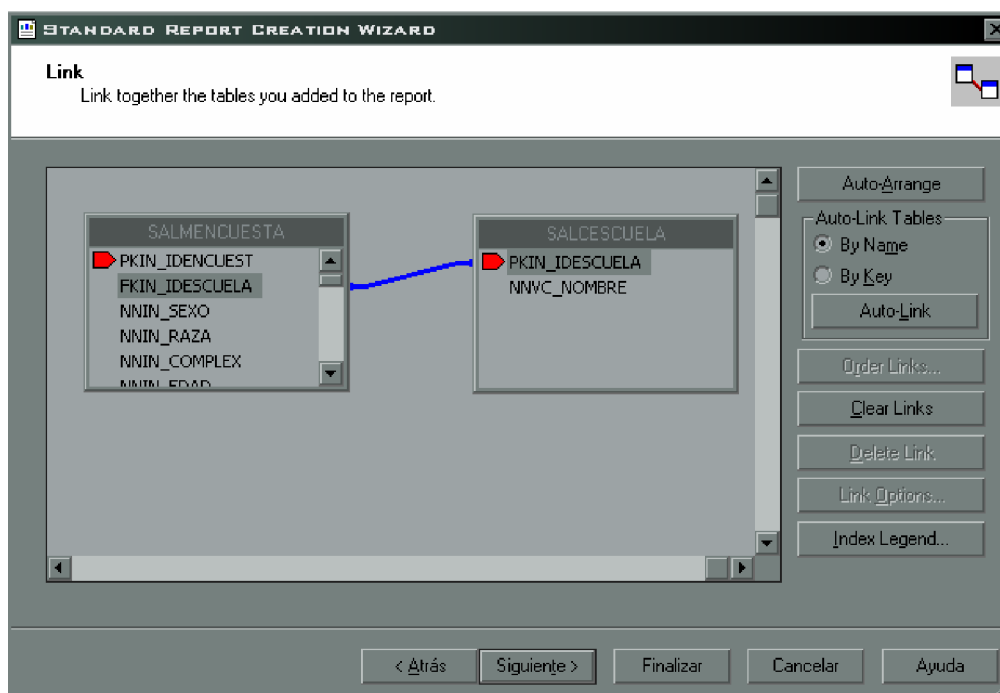
Se abre la base de datos de Access, que en este caso es SALTO.MDB



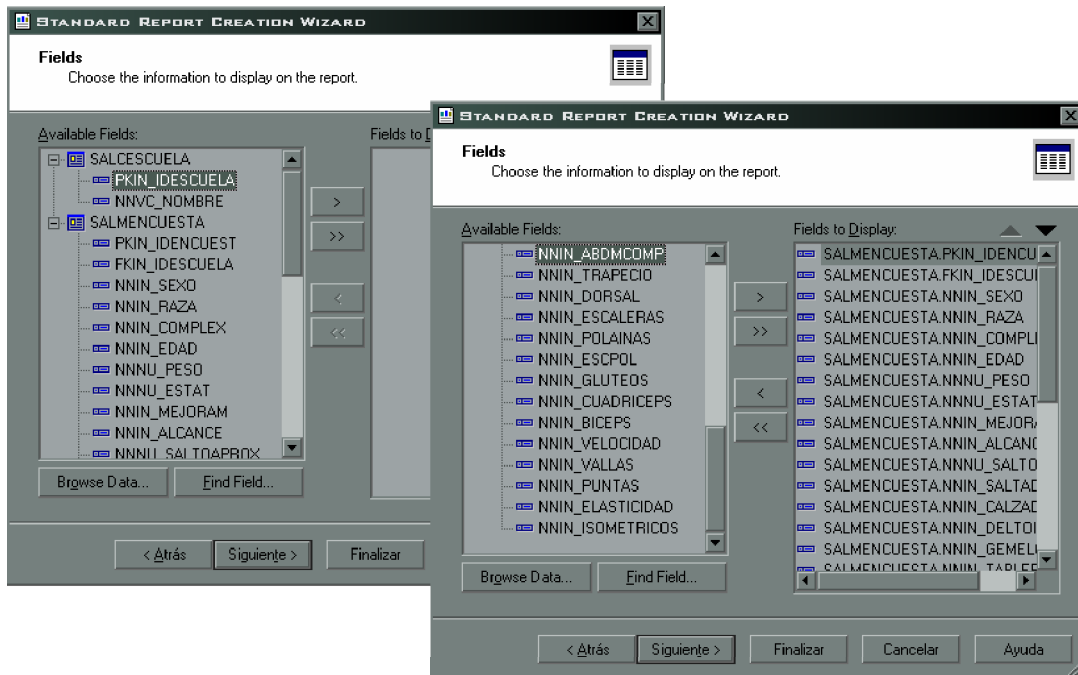
Se seleccionan las tablas de la Base de Datos que corresponden a la entidad educativa y a las encuestas realizadas pasándolas hacia el lado derecho de la ventana, luego se oprime siguiente:



Después se unen las dos tablas con la llave foránea de la tabla de encuesta que apunte a la llave primaria de tabla de la entidad educativa:

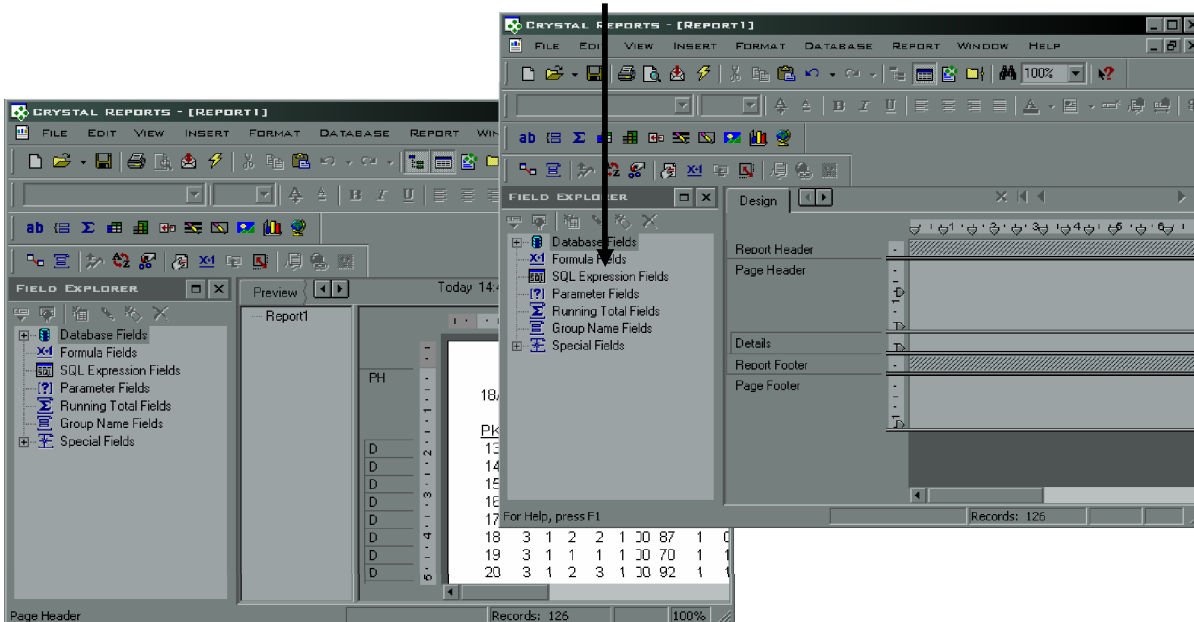


Se seleccionan todos los campos de la tabla de las encuestas y se pasan a la sección derecha. Luego se oprime finalizar:

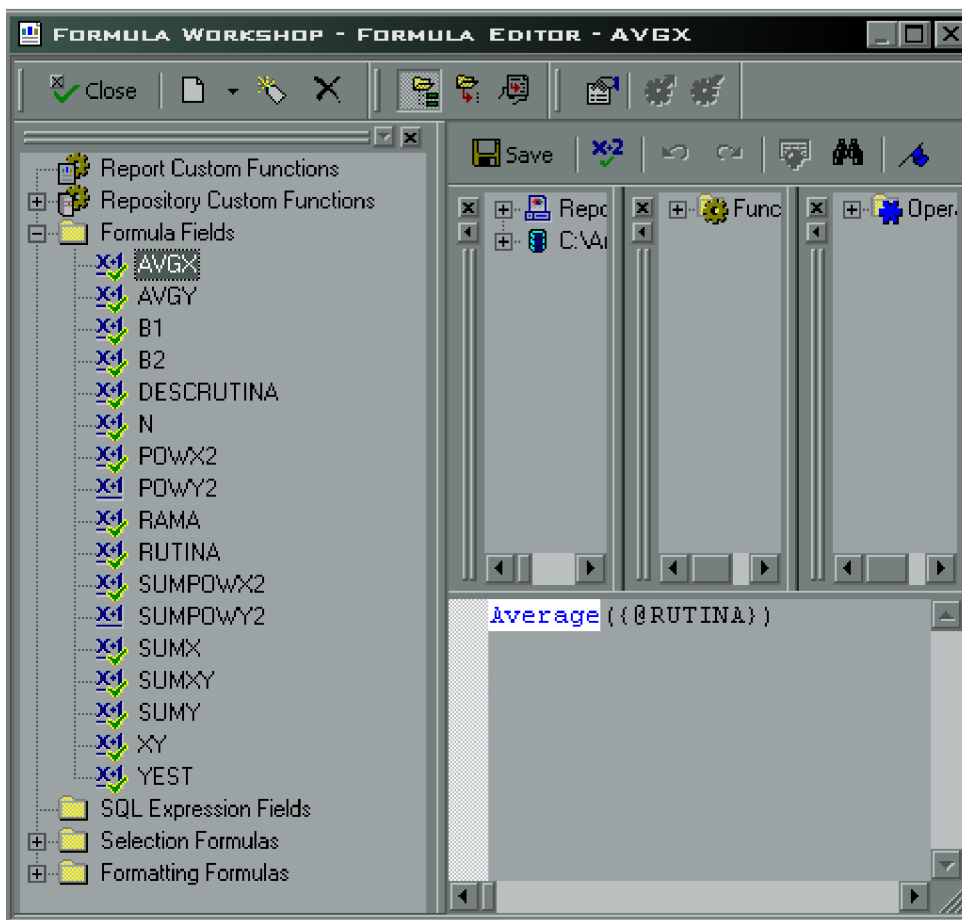
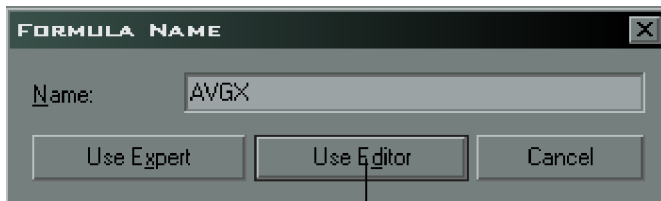


4.2.2.1 Grafica/reporte de dispersión

Una vez apareciendo todos los campos en el reporte, se seleccionan y se suprimen para elaborar el desarrollo de graficación comenzando por la creación de fórmulas, en esta sección se oprime botón derecho y luego [New] para crear una nueva fórmula:



Dentro de esta opción se selecciona el botón editor [Use Editor] para editar la Nueva fórmula y escribir el código, que en este caso será \bar{x} (x media), que en código será: Average({RUTINA})



AVGX = Average({@RUTINA})

Si aparecen mensajes de que hay un error en la fórmula será porque todavía no están completas todas las fórmulas del desarrollo, por lo que será necesario terminar el código de desarrollo de cada una antes de graficar.

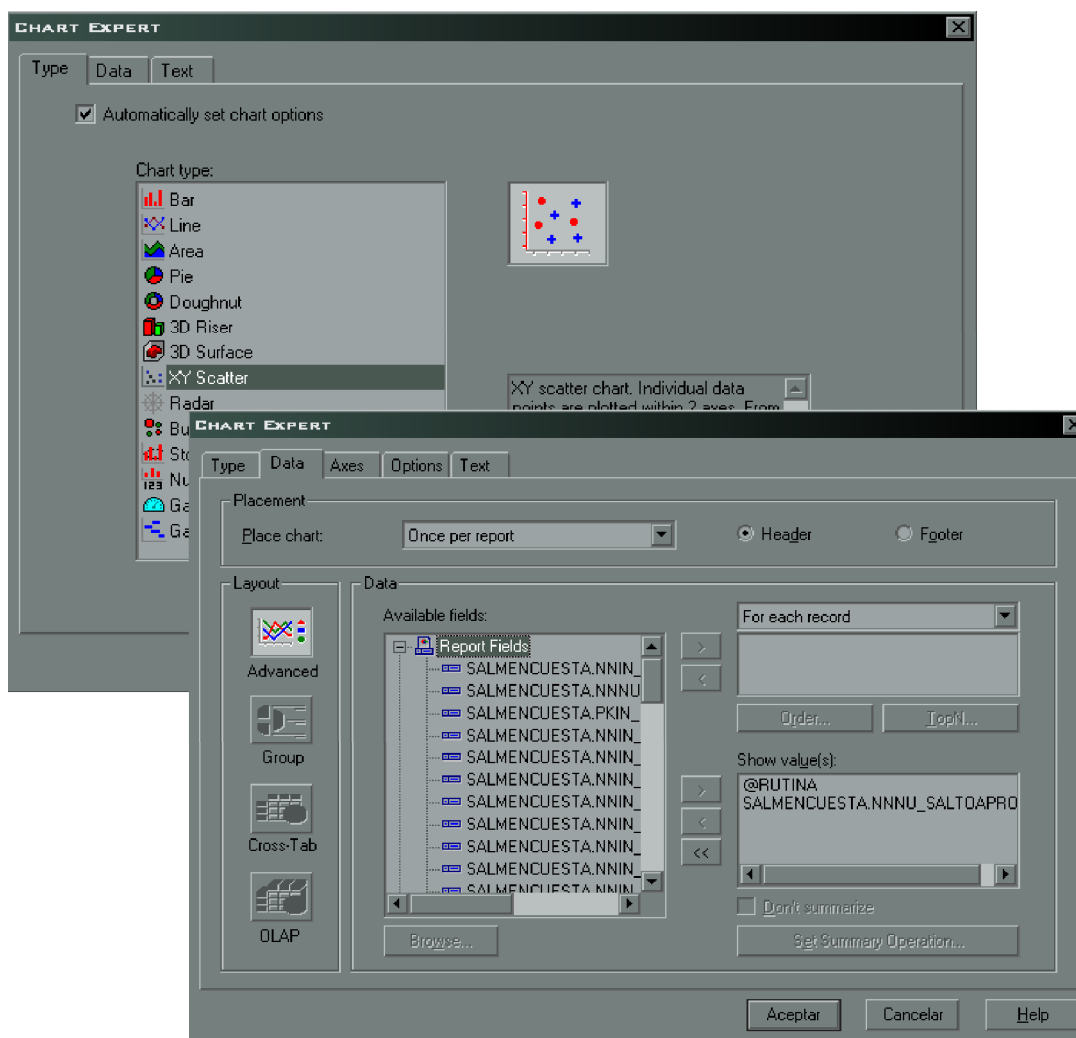
Parámetro	Fórmula	Código
\bar{y}	@AVGY	Average ({SALMENCUESTA.NNNU_SALTOAPROX})
$\hat{\beta}_1$	@B1	{@AVGY}-{@B2}*{@AVGX}

Parámetro	Fórmula	Código
β_2	@B2	(({@N}*{@SUMXY})-{@SUMX}*{@SUMY})/(({@N}*{@SUMPOWX2})-{@SUMX}*{@SUMX})
Rutina	@DESCRUTINA	SELECT TONUMBER({?RUTINA}) CASE 1:"DELTOIDES" CASE 2:"GEMELOS" CASE 3:"TABLEROS" CASE 4:"ABDOM-CORT" CASE 5:"ABDOM-COMP" CASE 6:"TRAPECIO" CASE 7:"DORSALES" CASE 8:"ESCALERAS" CASE 9:"TROTE-POLAINAS" CASE 10:"ESCAL-POLAINAS" CASE 11:"GLUTEOS" CASE 12:"CUADRICEPS" CASE 13:"BICEPS-CRURAL" CASE 14:"ARRANCONES" CASE 15:"VALLAS" CASE 16:"PUNTAS" CASE 17:"ELASTICIDAD" CASE 18:"ISOMETRICOS" DEFAULT: "NINGUNA"
n	@N	COUNT({SALMENCUESTA.PKIN_IDENCUEST})
x^2	@POWX2	{@RUTINA}*{@RUTINA}
y^2	@POWY2	{SALMENCUESTA.NNNU_SALTOAPROX}* {SALMENCUESTA.NNNU_SALTOAPROX}
Rama	@RAMA	SELECT TONUMBER({?SEXO}) CASE 0: "FEMENIL" CASE 1: "VARONIL" CASE 2: "VARONIL Y FEMENIL"
Rutina	@RUTINA	SELECT TONUMBER({?RUTINA}) CASE 1: {SALMENCUESTA.NNIN_DELTOIDES} CASE 2: {SALMENCUESTA.NNIN_GEMELOS} CASE 3: {SALMENCUESTA.NNIN_TABLERO} CASE 4: {SALMENCUESTA.NNIN_ABDMCORT} CASE 5: {SALMENCUESTA.NNIN_ABDMCOMP} CASE 6: {SALMENCUESTA.NNIN_TRAPECIO} CASE 7: {SALMENCUESTA.NNIN_DORSAL} CASE 8: {SALMENCUESTA.NNIN_ESCALERAS} CASE 9: {SALMENCUESTA.NNIN_POLAINAS} CASE 10: {SALMENCUESTA.NNIN_ESCPOL} CASE 11: {SALMENCUESTA.NNIN_GLUTEOS} CASE 12: {SALMENCUESTA.NNIN_CUADRICEPS} CASE 13: {SALMENCUESTA.NNIN_BICEPS} CASE 14: {SALMENCUESTA.NNIN_VELOCIDAD} CASE 15: {SALMENCUESTA.NNIN_VALLAS} CASE 16: {SALMENCUESTA.NNIN_PUNTAS} CASE 17: {SALMENCUESTA.NNIN_ELASTICIDAD} CASE 18: {SALMENCUESTA.NNIN_ISOMETRICOS} DEFAULT: 0
$\sum x^2$	@SUMPOWX2	SUM({@POWX2})
$\sum y^2$	@SUMPOWY2	SUM({@POWY2})
$\sum x$	@SUMX	SUM({@RUTINA})
$\sum xy$	@SUMXY	SUM({@XY})
$\sum y$	@SUMY	SUM({SALMENCUESTA.NNNU_SALTOAPROX})
xy	@XY	{@RUTINA}*{SALMENCUESTA.NNNU_SALTOAPROX}
\bar{y}	@YEST	{@B1}+({@B2}*{@RUTINA})

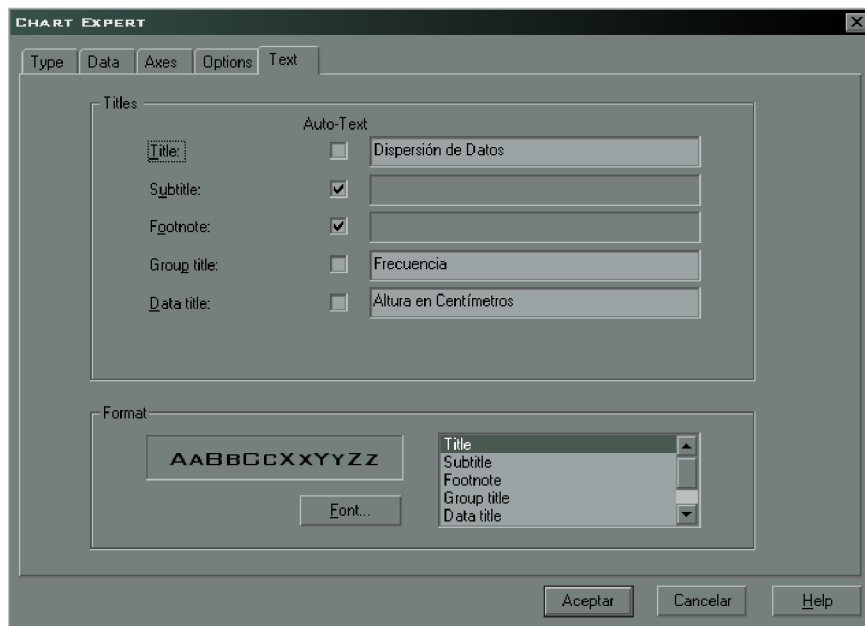
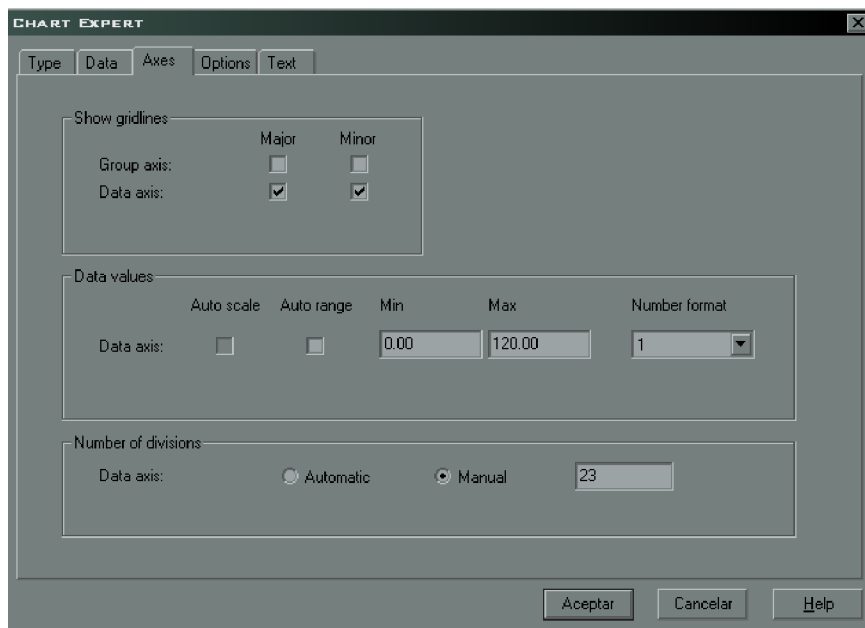
Una vez obtenido todas las fórmulas de la regresión, se comienza a elaborar el gráfico de dispersión oprimiendo el botón de gráfico (Chart)



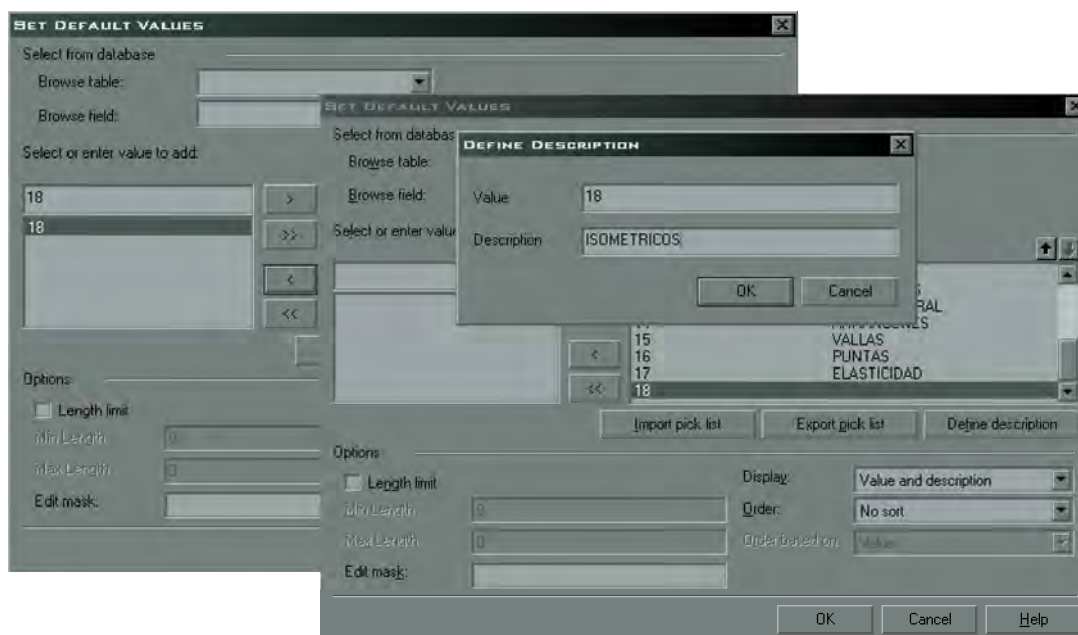
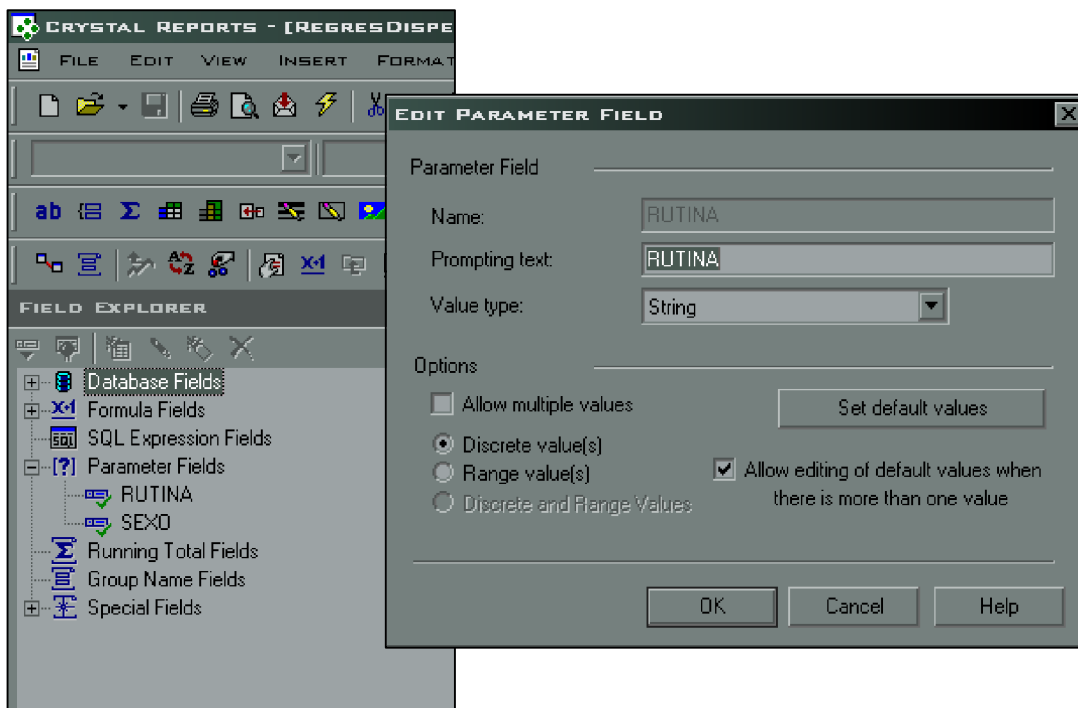
Se selecciona la opción de XY Scatter (Dispersión XY) y luego se oprime Aceptar. Posteriormente se seleccionan la fórmula @RUTINA y el campo NNUU_SALTOAPROX en la sección de Mostrar valores [Show Value(s)] con las opciones que aparecen en la pantalla: Una vez por reporte [Once per Report] encabezado [Header] y por cada registro [For each record].



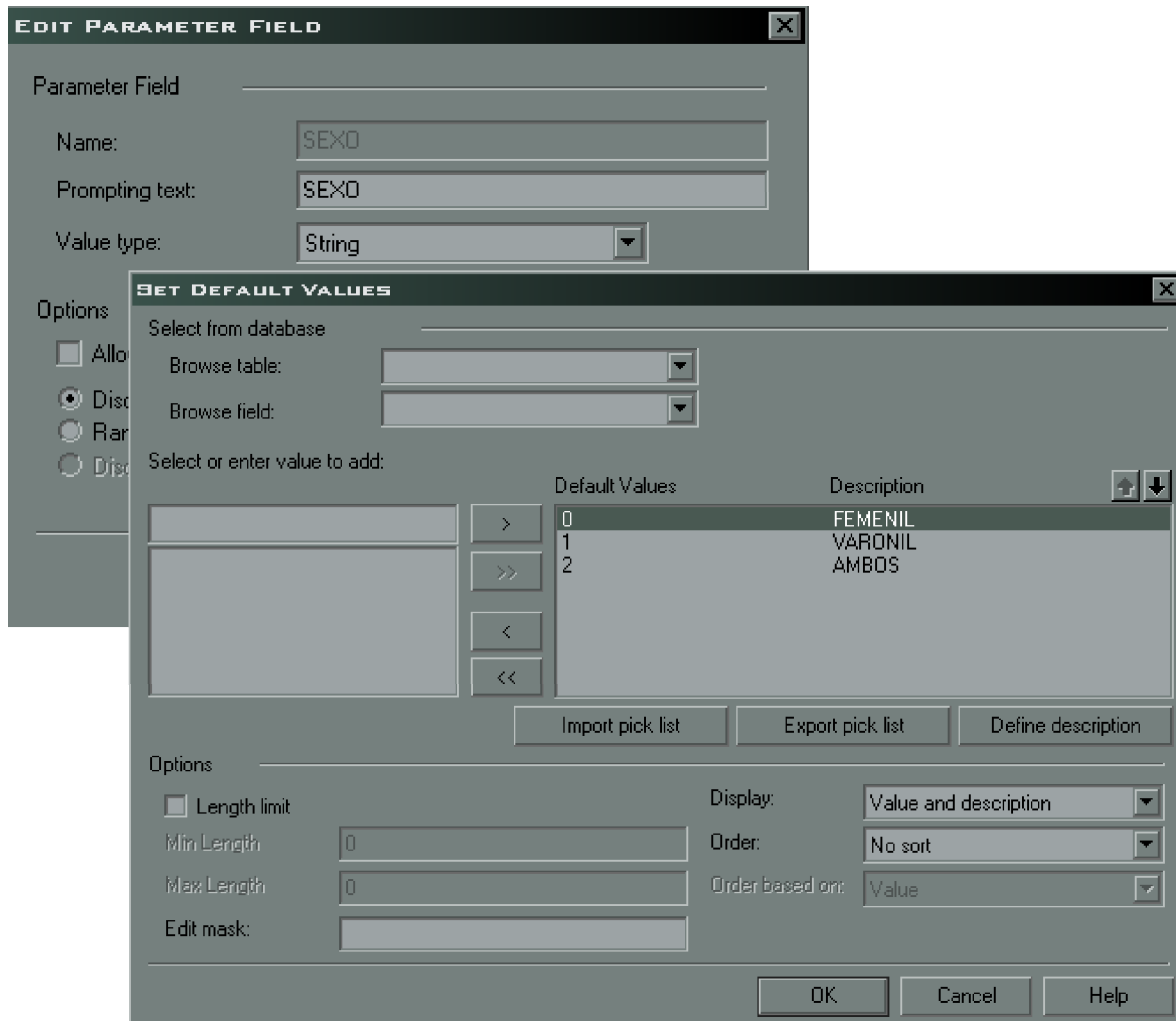
En la pestaña de Axes y Text se seleccionan las opciones de las pantallas siguientes:



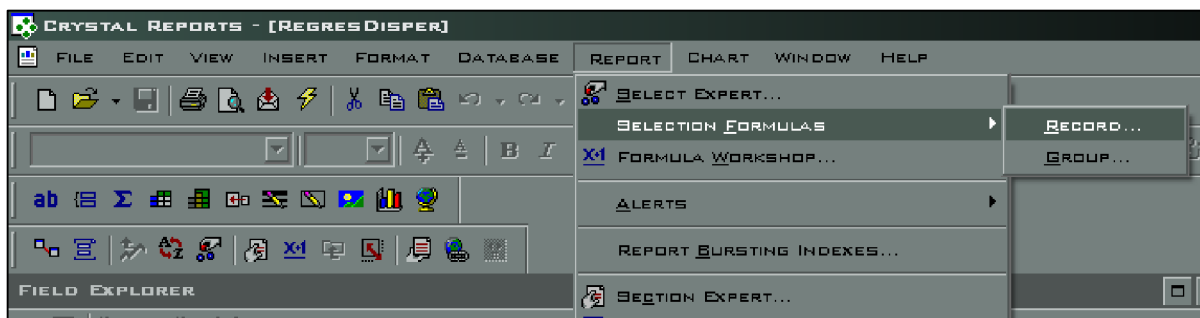
Una vez introducidos y seleccionados los valores del gráfico se acepta el cambio. Después se selecciona dentro del Explorador de campos [Field Explorer] la opción de Campos parámetro [Parameter Fields]. Esta opción es para que al ejecutarse el reporte se le filtren los datos ya sea por rama varonil y/o femenil y tipo de rutina que se quiere graficar. En este caso sólo se pondrán 2 filtros para elegir por parte del usuario: el parámetro RUTINA y el parámetro SEXO. Luego se oprime Establecer valores por default [Set default Values]. En la tercer ventana se capturan los valores por omisión por número y descripción de la rutina:



Se lleva a cabo el mismo procedimiento para el parámetro de SEXO con 3 opciones Varonil, Femenil y Ambos:



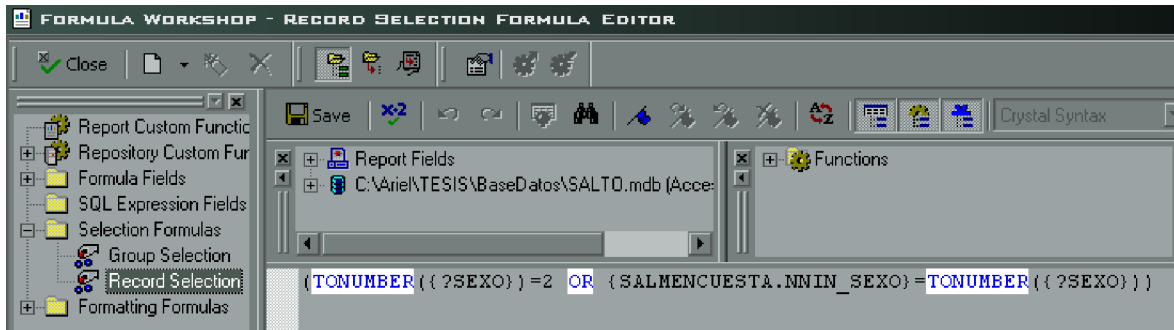
Después de oprimir [Ok], se selecciona la opción del menú: [Report+Selection Formulas+Record]. Esta opción es para filtrar el reporte gráfico en base a los parámetros introducidos por el usuario.



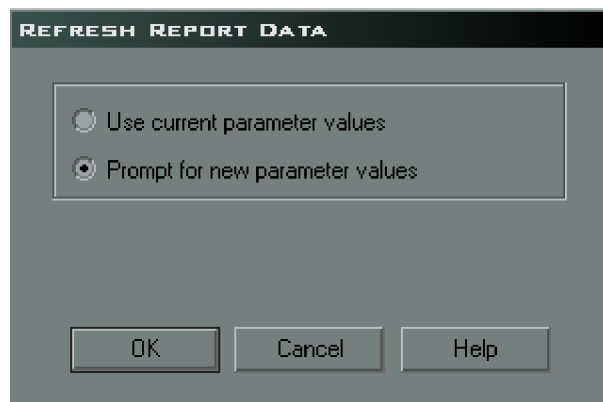
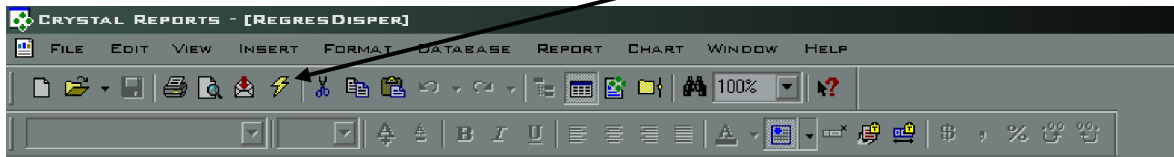
Una vez abierta la ventana de edición se procede a introducir la fórmula de filtrado de los registros, donde se introduce el siguiente código:

```
(TONUMBER({?SEXO})=2 OR {SALMENCUESTA.NNIN_SEXO}=TONUMBER({?SEXO}))
```

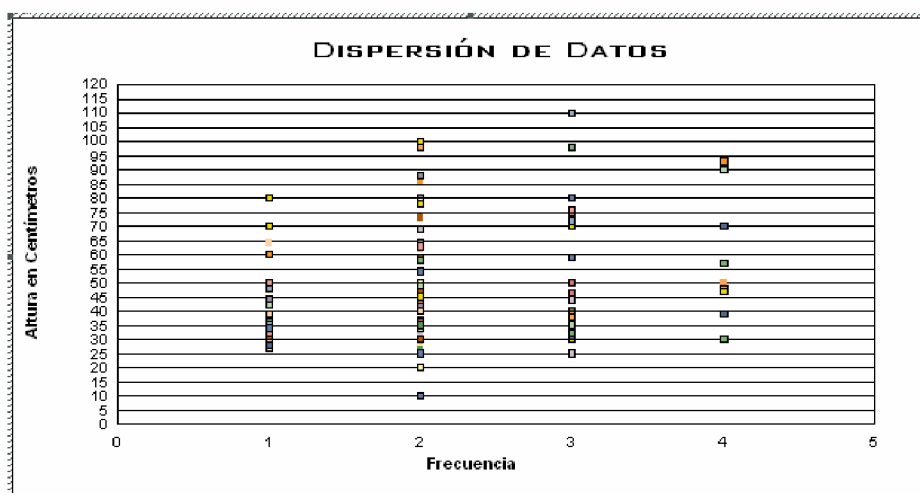
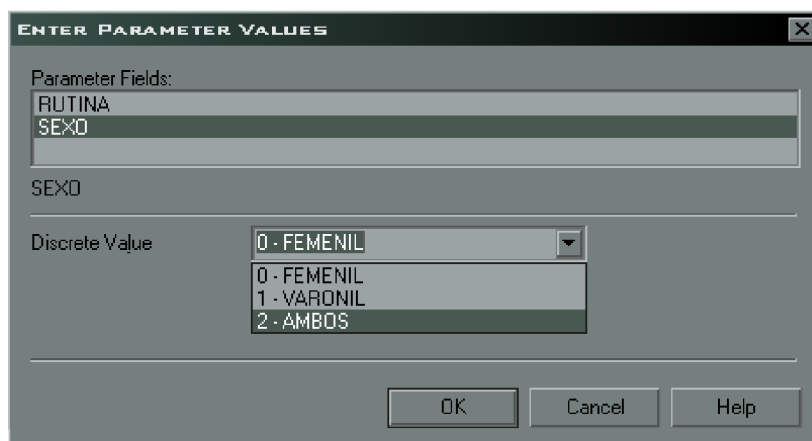
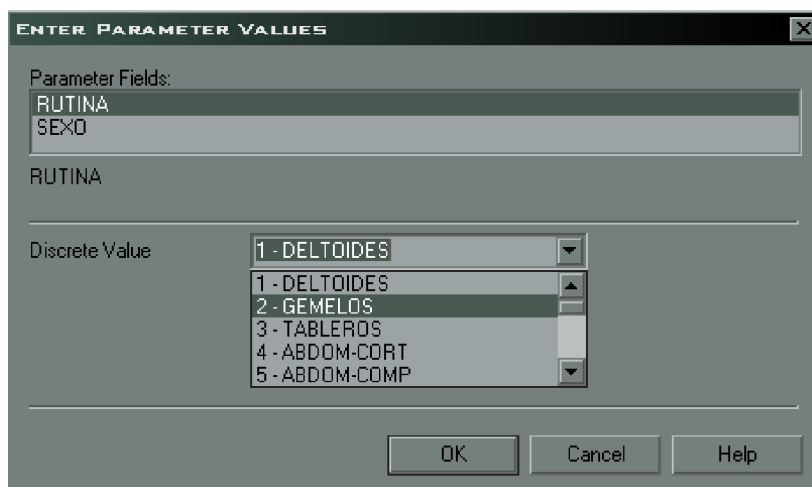
En donde el Sexo por default será 2 (ambos Sexos) o donde el Sexo seleccionado por el usuario sea el del parámetro. (Las fórmulas en Crystal Reports vienen precedidas por el [@] y los parámetros preceden por el signo de interrogación [?])



Una vez terminado el proceso de desarrollo, se guarda el archivo y se ejecuta la corrida por medio del Icono [Refresh] contenido en la barra de herramientas: Se selecciona Apuntador para introducir nuevos valores [Prompt for new parameter values]:

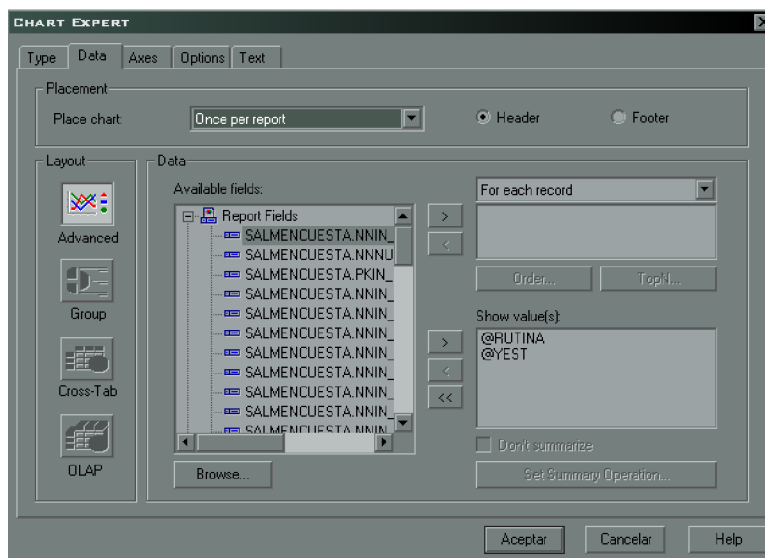


Para este ejemplo se ha seleccionado la rutina de los músculos Gemelos en Ambos Sexos:

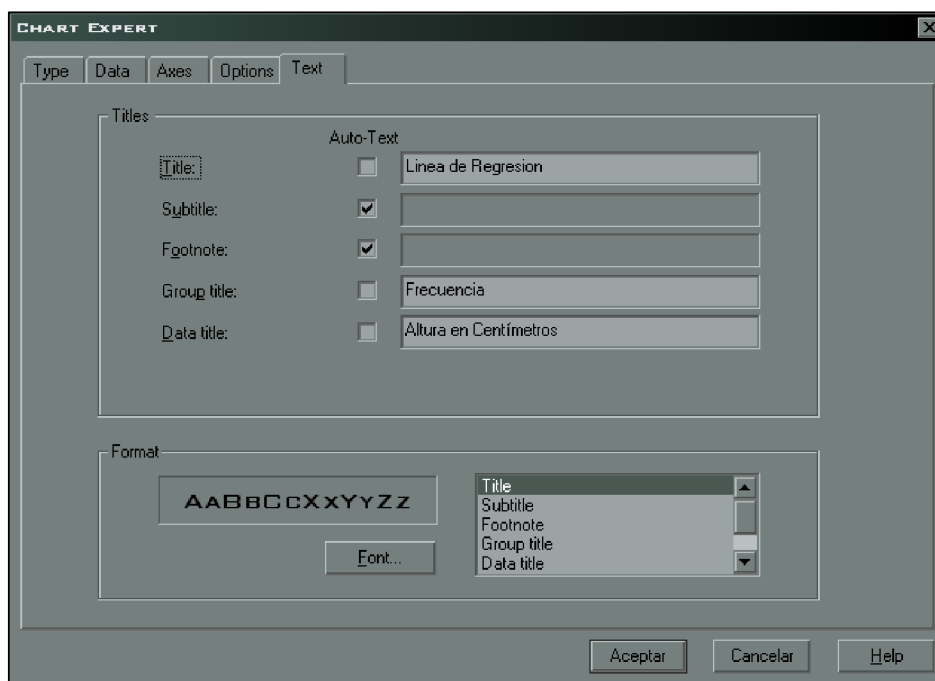


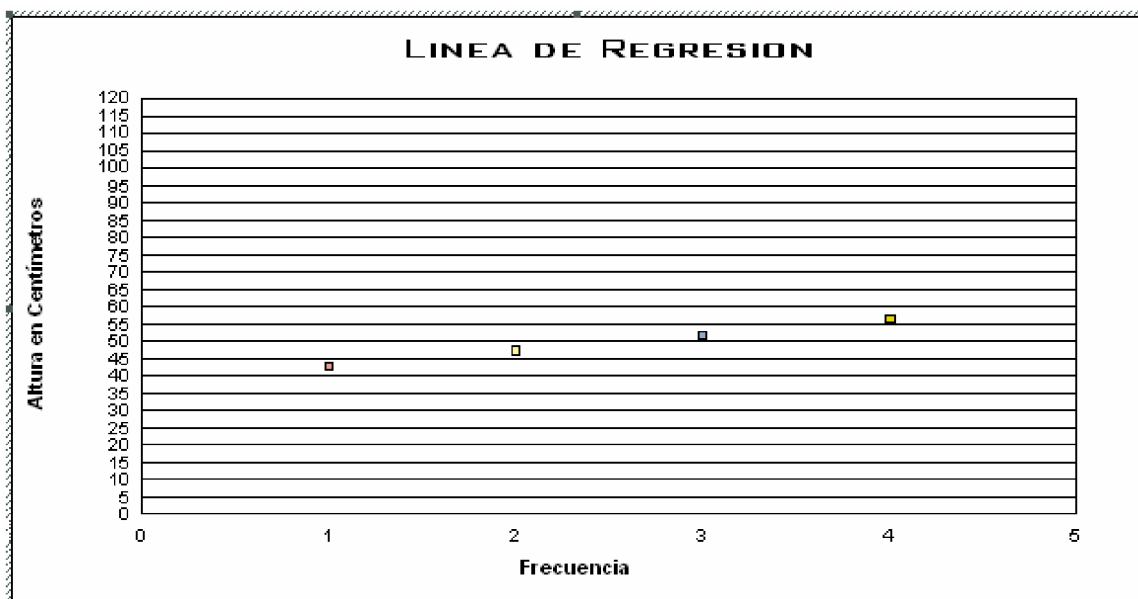
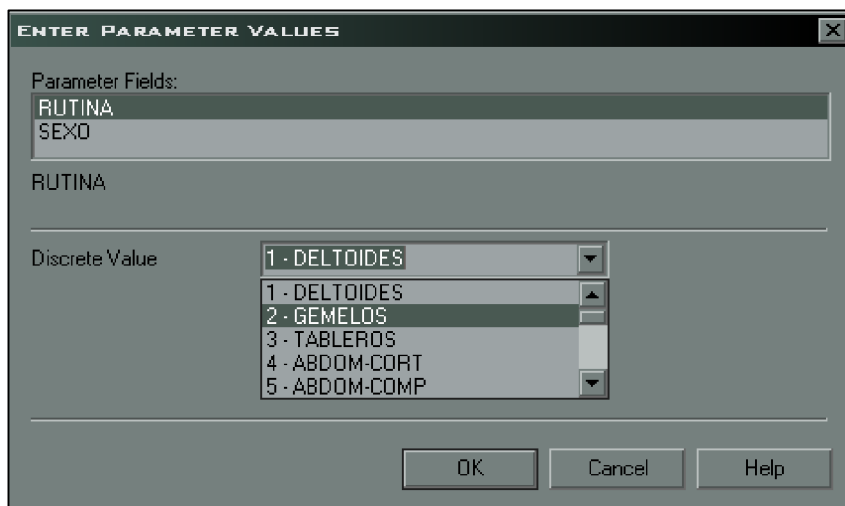
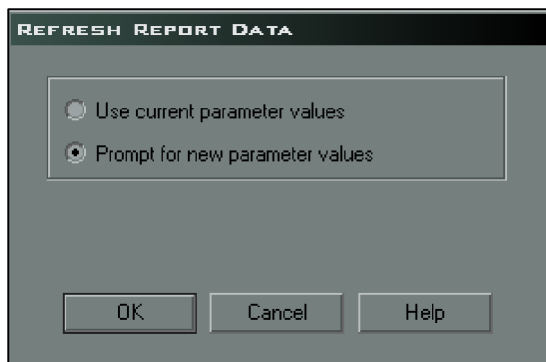
4.2.2.2 Grafica/reporte de regresión

Para el gráfico de regresión se procede de la misma forma, sólo que en este caso los valores a mostrar serán la rutina y el salto estimado de la regresión.

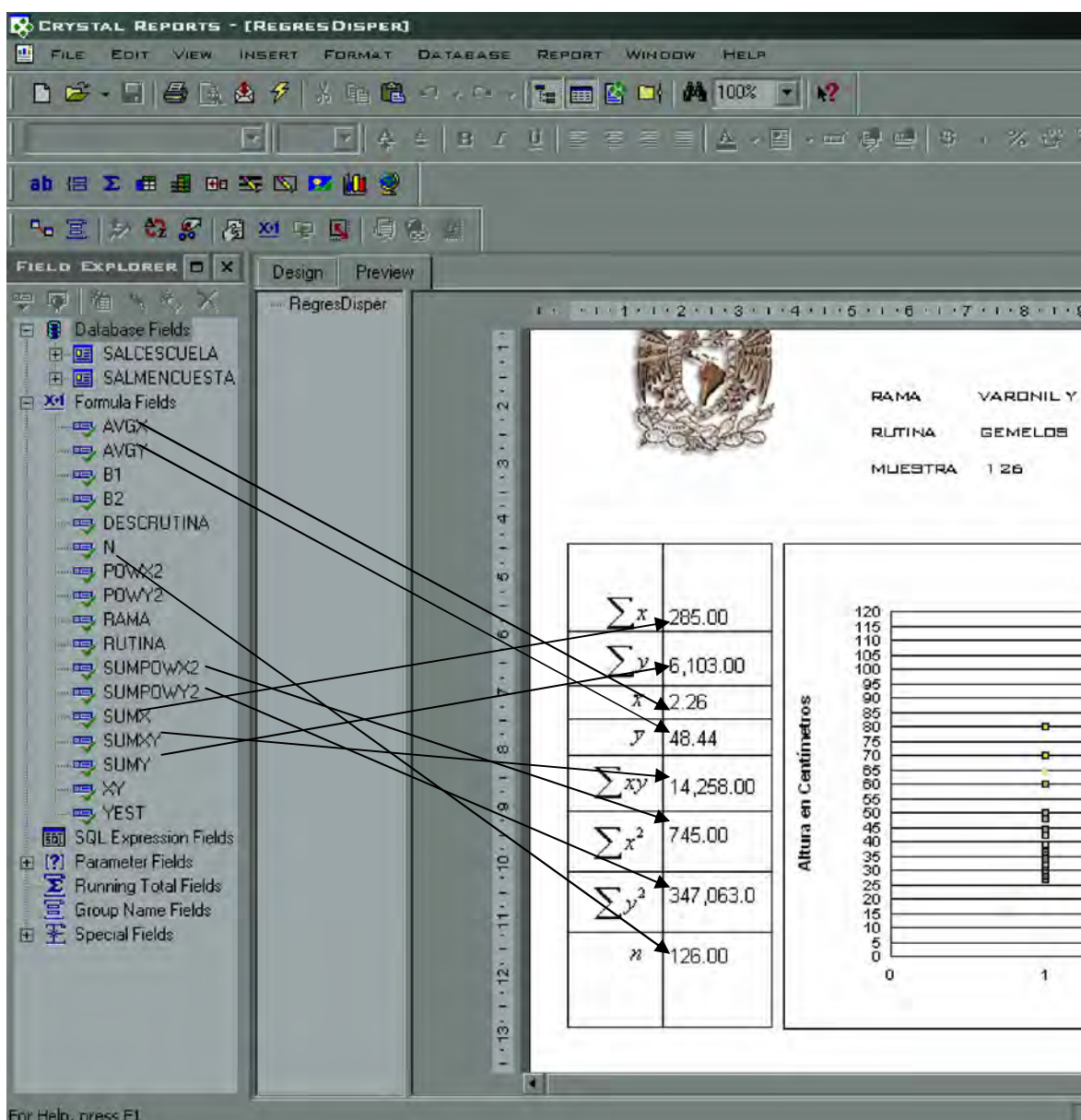


Los datos de texto serán con las leyendas de Línea de Regresión. Una vez Aceptado, se procede a ejecutar el reporte con el botón Refresh para actualizar los datos y mostrarse el segundo gráfico:

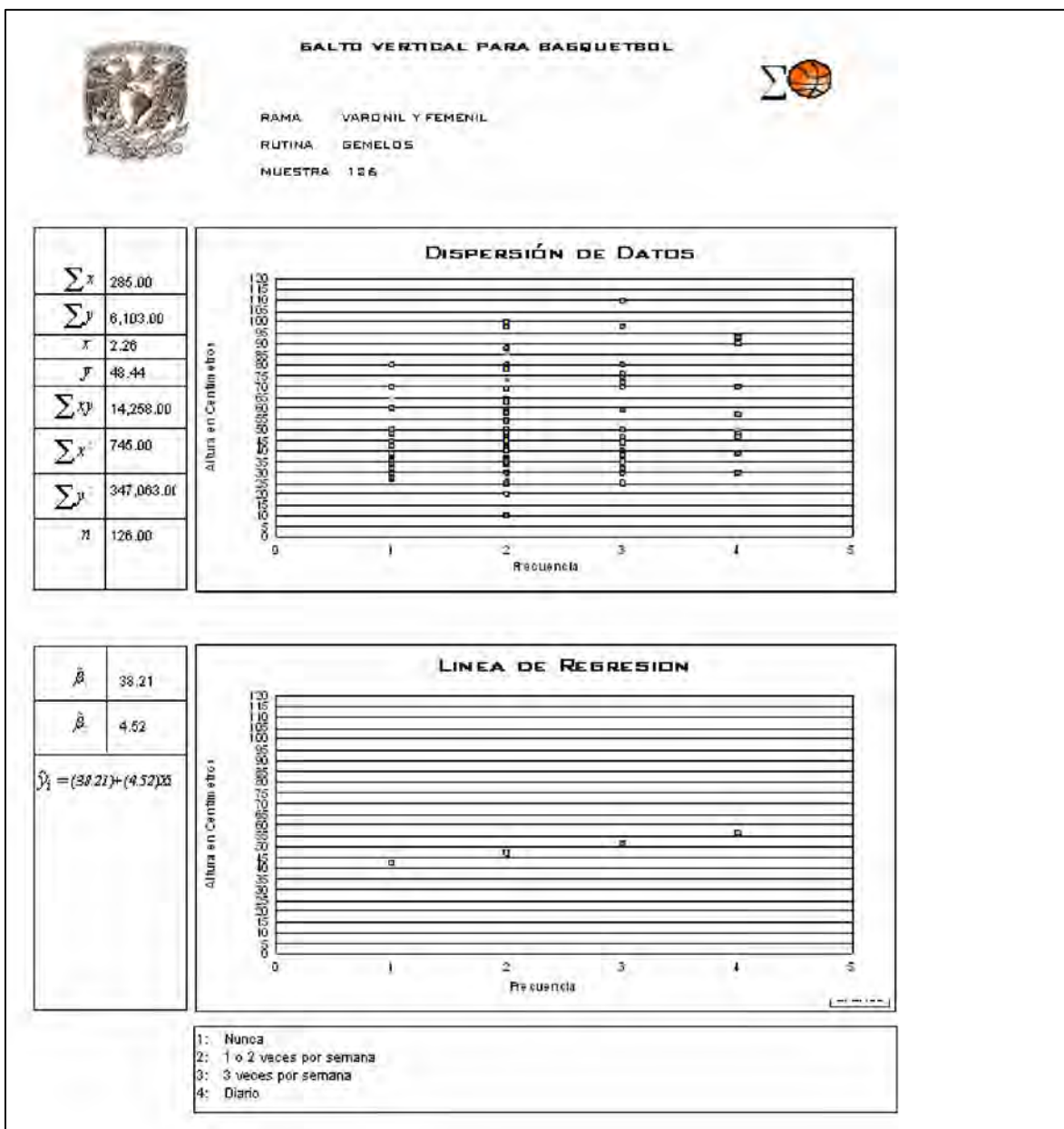




Después de escribir las fórmulas matemáticas en el editor de ecuaciones del Office e insertarlas en el diseño del reporte. Se procede a insertar las fórmulas de código escritas en el desarrollo.



Se efectúa el mismo procedimiento para el segundo gráfico y el reporte queda como se muestra a continuación.



4.2.2.3 Grafica/reporte de ajuste exponencial

Para el gráfico de Ajuste exponencial los procedimientos para desarrollar el reporte graficador son los mismos, sin embargo el desarrollo de las fórmulas son distintas por razones obvias de tratarse en este caso de Ajuste exponencial.

Parámetro	Fórmula	Código
β_1	@a	10^{@Loga}
β_2	@b	10^{@Logb}
Rutina	@DESCRUTINA	<pre> SELECT TONUMBER({?RUTINA}) CASE 1: "DELTOIDES" CASE 2: "GEMELOS" CASE 3: "TABLEROS" CASE 4: "ABDOMINALES CORTOS" CASE 5: "ABDOMINALES COMPLETOS" CASE 6: "TRAPECIO" CASE 7: "DORSALES" CASE 8: "ESCALERAS" CASE 9: "TROTE CON POLAINAS" CASE 10:"ESCALERAS CON POLAINAS" CASE 11:"GLUTEOS" CASE 12:"CUADRICEPS" CASE 13:"BICEPS CRURAL" CASE 14:"ARRANCONES" CASE 15:"VALLAS" CASE 16:"PUNTAS" CASE 17:"ELASTICIDAD" CASE 18:"ISOMETRICOS" DEFAULT: "NINGUNA" </pre>
$\log \beta_1$	@Loga	$(\{ @SUMLogY \} / \{ @N \}) - (\{ @Logb \}) * (\{ @SUMX \} / \{ @N \})$
$\log \beta_2$	@Logb	$((\{ @N \} * \{ @SUMXLogY \}) - (\{ @SUMX \} * \{ @SUMLogY \})) / ((\{ @N \} * \{ @SUMX^2 \}) - (\{ @SUMX \}^2))$
$\log y$	@LogY	0.4343*Log({@ Y})
n	@N	COUNT({SALMENCUESTA.PKIN_IDENCUEST})
Rama	@RAMA	<pre> SELECT TONUMBER({?SEXO}) CASE 0:"FEMENIL" CASE 1:"VARONIL" CASE 2:"VARONIL Y FEMENIL" </pre>
Rutina	@RUTINA	<pre> SELECT TONUMBER({?RUTINA}) CASE 1: {SALMENCUESTA.NNIN_DELTOIDES} CASE 2: {SALMENCUESTA.NNIN_GEMELOS} CASE 3: {SALMENCUESTA.NNIN_TABLERO} CASE 4: {SALMENCUESTA.NNIN_ABDMCORT} CASE 5: {SALMENCUESTA.NNIN_ABDMCOMP} CASE 6: {SALMENCUESTA.NNIN_TRAPECIO} CASE 7: {SALMENCUESTA.NNIN_DORSAL} CASE 8: {SALMENCUESTA.NNIN_ESCALERAS} CASE 9: {SALMENCUESTA.NNIN_POLAINAS} CASE 10: {SALMENCUESTA.NNIN_ESCPOL} CASE 11: {SALMENCUESTA.NNIN_GLUTEOS} CASE 12: {SALMENCUESTA.NNIN_CUADRICEPS} CASE 13: {SALMENCUESTA.NNIN_BICEPS} CASE 14: {SALMENCUESTA.NNIN_VELOCIDAD} CASE 15: {SALMENCUESTA.NNIN_VALLAS} CASE 16: {SALMENCUESTA.NNIN_PUNTAS} CASE 17: {SALMENCUESTA.NNIN_ELASTICIDAD} CASE 18: {SALMENCUESTA.NNIN_ISOMETRICOS} DEFAULT: 0 </pre>

Parámetro	Fórmula	Código
$\sum \log y$	@SUMLogY	SUM({@LogY})
$\sum x$	@SUMX	SUM({@X})
$\sum x^2$	@SUMX^2	SUM({@X^2})
$\sum x \log y$	@SUMXLogY	SUM({@XLogY})
$\sum y$	@SUMY	SUM({@Y})
x	@X	{@RUTINA}
x^2	@X^2	{@X} ^ 2
$x \log y$	@XlogY	{@X}*{@LogY}
y	@Y	{SALMENCUESTA.NNNU_SALTOAPROX}
\hat{y}	@YEST	{@a}*{@b}^{@X}

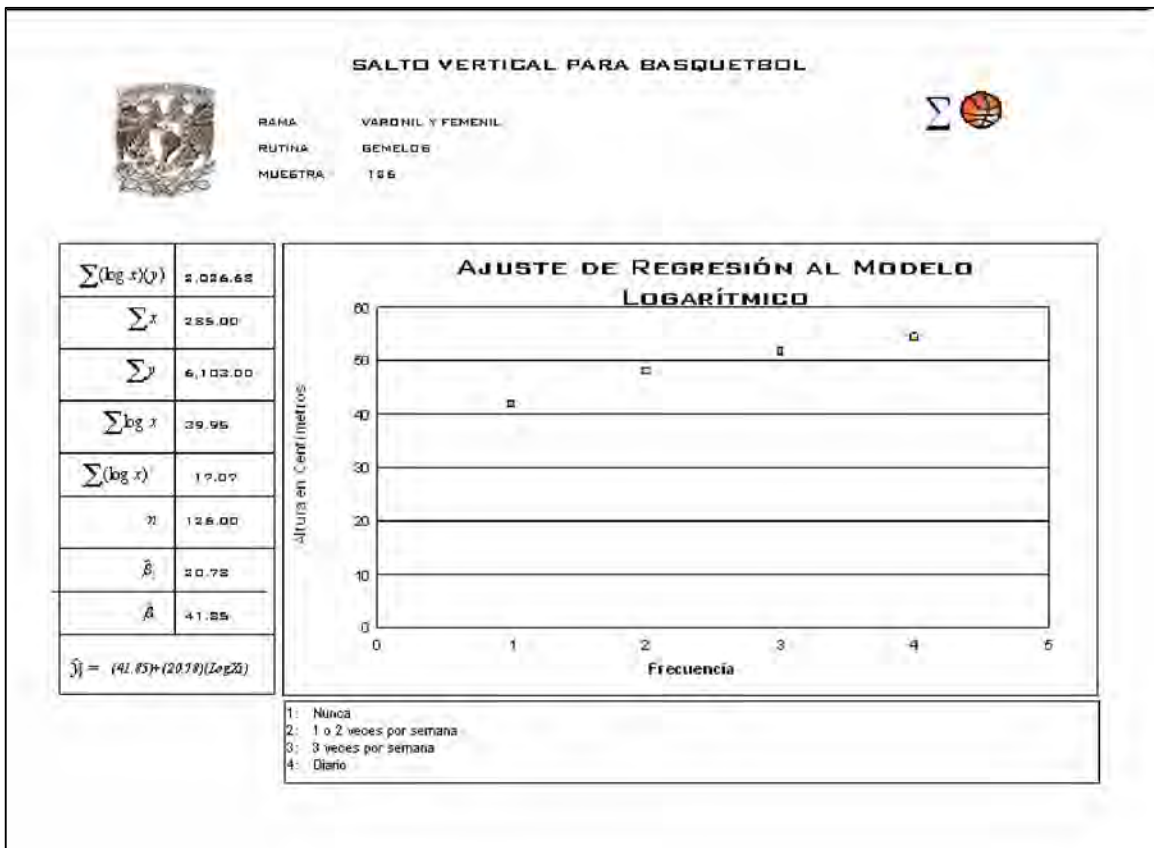


4.2.2.4 Grafica/reporte de ajuste logaritmico

Para el gráfico de Ajuste logaritmico los procedimientos para desarrollar el reporte graficador también son los mismos, pero en este caso el desarrollo de las fórmulas son distintas por tratarse de Ajuste logaritmico.

Parámetro	Fórmula	Código
$\hat{\beta}_1$	@a	$((@SumY)/{@N})-{@b}*{@SumLogX}/{@N}$
$\hat{\beta}_2$	@b	$(({@N}*{@SUMLogXY})-{@SumLogX}*{@SumY})/({{@N}*{@SumLogX^2})-({@SumLogX})^2$
DescRutina	@DESCRUTINA	<pre> SELECT TONUMBER({?RUTINA}) CASE 1: "DELTOIDES" CASE 2: "GEMELOS" CASE 3: "TABLEROS" CASE 4: "ABDOMINALES CORTOS" CASE 5: "ABDOMINALES COMPLETOS" CASE 6: "TRAPECIO" CASE 7: "DORSALES" CASE 8: "ESCALERAS" CASE 9: "TROTE CON POLAINAS" CASE 10:"ESCALERAS CON POLAINAS" CASE 11:"GLUTEOS" CASE 12:"CUADRICEPS" CASE 13:"BICEPS CRURAL" CASE 14:"ARRANCONES" CASE 15:"VALLAS" CASE 16:"PUNTAS" CASE 17:"ELASTICIDAD" CASE 18:"ISOMETRICOS" DEFAULT: "NINGUNA" </pre>
$\log x$	@LogX	$\text{Log}({@X})*0.4343$
$\log x^2$	@LogX^2	$\{{@LogX}\}^2$
$\log xy$	@LogXY	$\{{@LogX}\}*{\@Y}$
n	@N	$\text{COUNT}(\{\text{SALMENCUESTA.PKIN_IDENCUEST}\})$
Rama	@RAMA	<pre> SELECT TONUMBER({?SEXO}) CASE 0: " FEMENIL" CASE 1: " VARONIL" CASE 2: " VARONIL Y FEMENIL" </pre>
Rutina	@RUTINA	<pre> SELECT TONUMBER({?RUTINA}) CASE 1: {SALMENCUESTA.NNIN_DELTOIDES} CASE 2: {SALMENCUESTA.NNIN_GEMELOS} CASE 3: {SALMENCUESTA.NNIN_TABLERO} CASE 4: {SALMENCUESTA.NNIN_ABDMCORT} CASE 5: {SALMENCUESTA.NNIN_ABDMCOMP} CASE 6: {SALMENCUESTA.NNIN_TRAPECIO} CASE 7: {SALMENCUESTA.NNIN_DORSAL} CASE 8: {SALMENCUESTA.NNIN_ESCALERAS} CASE 9: {SALMENCUESTA.NNIN_POLAINAS} CASE 10: {SALMENCUESTA.NNIN_ESCPOL} CASE 11: {SALMENCUESTA.NNIN_GLUTEOS} CASE 12: {SALMENCUESTA.NNIN_CUADRICEPS} CASE 13: {SALMENCUESTA.NNIN_BICEPS} CASE 14: {SALMENCUESTA.NNIN_VELOCIDAD} CASE 15: {SALMENCUESTA.NNIN_VALLAS} CASE 16: {SALMENCUESTA.NNIN_PUNTAS} CASE 17: {SALMENCUESTA.NNIN_ELASTICIDAD} CASE 18: {SALMENCUESTA.NNIN_ISOMETRICOS} DEFAULT: 0 </pre>

Parámetro	Fórmula	Código
$\sum \log x$	@SumLogX	Sum({@LogX})
$\sum (\log x)^2$	@SumLogX^2	Sum({@LogX^2})
$\sum (\log x)(y)$	@SUMLogXY	Sum({@LogXY})
$\sum x$	@SumX	Sum({@X})
$\sum y$	@SumY	Sum({@Y})
x	@X	{@RUTINA}
y	@Y	{SALMENCUESTA.NNNU_SALTOAPROX}
\hat{y}	@YEST	{@a}+{@b}*(LOG({@X})*0.4343)




4.2.2.5 Gráfica/reporte de ajuste cuadrático

Para el gráfico de Ajuste cuadrático los procedimientos para desarrollar el reporte graficador también son los mismos, no obstante el desarrollo de las fórmulas son diferentes como en cada ajuste realizado.


Parámetro	Fórmula	Código
β_1	@a	$\frac{(((\sum Y)/\sum X^2) - ((\sum XY)/\sum X^3)) / ((\sum X^2)/\sum X^4 - ((\sum X)^2/\sum X^4))}{(((\sum XY)/\sum X^3) - ((\sum X^2)/\sum X^4)) / ((\sum X^2)/\sum X^4 - ((\sum X)^2/\sum X^4))} - \frac{((\sum XY)/\sum X^3) - ((\sum X^2)/\sum X^4)}{((\sum X^2)/\sum X^4 - ((\sum X)^2/\sum X^4))}$
β_2	@b	$\frac{((\sum Y)/\sum X^2) - ((\sum XY)/\sum X^3)}{((\sum X^2)/\sum X^4) - ((\sum X)^2/\sum X^4)} - a * \frac{((\sum N)/\sum X^2) - ((\sum X)/\sum X^3)}{((\sum X^2)/\sum X^4) - ((\sum X)^2/\sum X^4)}$
β_3	@c	$((\sum Y)/\sum X^2) - (a * ((\sum N)/\sum X^2) + b * ((\sum X)/\sum X^2))$
DescRutina	@DESCRUTINA	<pre>SELECT TONUMBER(?RUTINA) CASE 1: "DELTOIDES" CASE 2: "GEMELOS" CASE 3: "TABLEROS" CASE 4: "ABDOMINALES CORTOS" CASE 5: "ABDOMINALES COMPLETOS" CASE 6: "TRAPECIO" CASE 7: "DORSALES" CASE 8: "ESCALERAS" CASE 9: "TROTE CON POLAINAS" CASE 10:"ESCALERAS CON POLAINAS" CASE 11:"GLUTEOS" CASE 12:"CUADRICEPS" CASE 13:"BICEPS CRURAL" CASE 14:"ARRANCONES" CASE 15:"VALLAS" CASE 16:"PUNTAS" CASE 17:"ELASTICIDAD" CASE 18:"ISOMETRICOS" DEFAULT: "NINGUNA"</pre>
n	@N	COUNT({SALMENCUESTA.PKIN_IDENCUEST})
x^2	@POWX2	{@RUTINA}^2
$x^2 y$	@POWX2Y	{@RUTINA}^2*{SALMENCUESTA.NNNU_SALTOAPROX}
x^3	@POWX3	{@RUTINA}^3
x^4	@POWX4	{@RUTINA}^4
Rama	@RAMA	<pre>SELECT TONUMBER(?SEXO) CASE 0: "FEMENIL" CASE 1: "VARONIL" CASE 2: "VARONIL Y FEMENIL"</pre>

Parámetro	Fórmula	Código
Rutina	@RUTINA	<pre> SELECT TONUMBER({?RUTINA}) CASE 1: {SALMENCUESTA.NNIN_DELTOIDES} CASE 2: {SALMENCUESTA.NNIN_GEMELOS} CASE 3: {SALMENCUESTA.NNIN_TABLERO} CASE 4: {SALMENCUESTA.NNIN_ABDMCORT} CASE 5: {SALMENCUESTA.NNIN_ABDMCOMP} CASE 6: {SALMENCUESTA.NNIN_TRAPECIO} CASE 7: {SALMENCUESTA.NNIN_DORSAL} CASE 8: {SALMENCUESTA.NNIN_ESCALERAS} CASE 9: {SALMENCUESTA.NNIN_POLAINAS} CASE 10: {SALMENCUESTA.NNIN_ESCPOL} CASE 11: {SALMENCUESTA.NNIN_GLUTEOS} CASE 12: {SALMENCUESTA.NNIN_CUADRICEPS} CASE 13: {SALMENCUESTA.NNIN_BICEPS} CASE 14: {SALMENCUESTA.NNIN_VELOCIDAD} CASE 15: {SALMENCUESTA.NNIN_VALLAS} CASE 16: {SALMENCUESTA.NNIN_PUNTAS} CASE 17: {SALMENCUESTA.NNIN_ELASTICIDAD} CASE 18: {SALMENCUESTA.NNIN_ISOMETRICOS} DEFAULT: 0 </pre>
$\sum x^2$	@SUMPOWX2	SUM({@POWX2})
$\sum x^2 y$	@SUMPOWX2Y	SUM({@POWX2Y})
$\sum x^3$	@SUMPOWX3	SUM({@POWX3})
$\sum x^4$	@SUMPOWX4	SUM({@POWX4})
$\sum x$	@SUMX	SUM({@RUTINA})
$\sum xy$	@SUMXY	SUM({@XY})
$\sum y$	@SUMY	SUM({SALMENCUESTA.NNNU_SALTOAPROX})
xy	@XY	{@RUTINA}*{SALMENCUESTA.NNNU_SALTOAPROX}
\hat{y}	@YEST	{@a}+{@b}*{@RUTINA}+{@c}*{@RUTINA}*{@RUTINA}}



SALTO VERTICAL PARA BASKETBOL

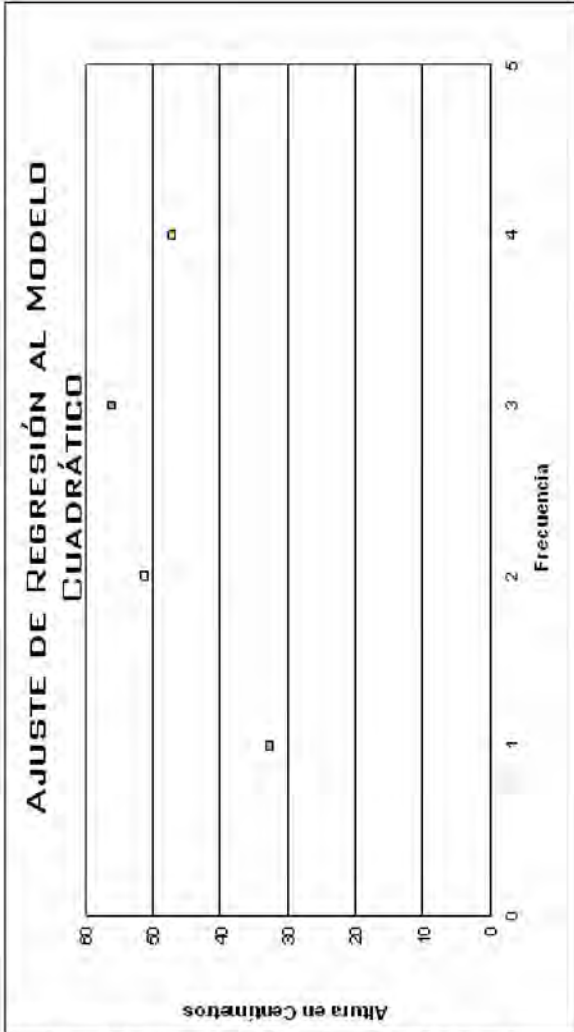
RAMA: VARONIL Y FEMENIL
 RUTINA: GEMELOS
 MUESTRA: 126



Σ

Σx	255.00
Σy	6,103.00
Σx^2	745.00
Σx^3	2,169.00
Σx^4	6,841.00
Σxy	14,252.00
Σx^2y	32,302.00
n	126.00
$\hat{\beta}_1$	-6.85
$\hat{\beta}_2$	39.07
$\hat{\beta}_3$	0.57

AJUSTE DE REGRESIÓN AL MODELO CUADRÁTICO



$$\hat{y}_i = (0.57)(39.07)\Sigma^2 + (-6.85)\Sigma^3 + 0.57$$

1: Nunca
 2: 1 o 2 veces por semana
 3: 3 veces por semana
 4: Diario

4.2.3 Desarrollo de sistema encuestador y gráfico-reportador

Para el desarrollo del sistema encuestador y reportador se utiliza el lenguaje de programación Visual Básico 6.0 del Visual Studio. Ahora bien, el sistema se compone de:

6 Formularios

- frmEncuesta Formulario de llenado de encuestas
- frmGráficos Formulario de selección de gráfica
- frmRegresAjusteExp Formulario del visor Crystal Reports para la gráfica de ajuste exponencial.
- frmRegresAjusteLog Formulario del visor Crystal Reports para la gráfica de ajuste logarítmico.
- frmRegresAjusteSqr Formulario del visor Crystal Reports para la gráfica de ajuste cuadrático
- frmRegresDisperLin Formulario del visor Crystal Reports para la gráfica de dispersión y regresión lineal.

1 Módulo

- modGeneral Módulo de procedimientos generales para los formularios

4 Reportes

- rptRegresAjusteExp Reporte de Crystal Reports para al gráfica exponencial
- rptRegresAjusteLog Reporte de Crystal Reports para al gráfica logarítmica
- rptRegresAjusteSqr Reporte de Crystal Reports para al gráfica cuadrática
- rptRegresDisperLin Reporte de Crystal Reports para al gráfica lineal

4.2.3.1 Módulo de llenado de encuestas..

El formulario es similar al formato de encuesta estadística, sólo que es interactivo. La institución universitaria depende de donde se haya aplicado la encuesta. El folio es automático y el llamado del formulario es mediante selección de opción múltiple y captura de pesos y medidas de cada atleta encuestado.

Institución de la encuesta

Folio de la encuesta

Datos de la encuesta

Nuevo registro

Guardar registro

Imprimir Grafico

Créditos

Cerrar Ventana

En la segunda pestaña del formulario principal se ha puesto cada una de las rutinas mas usadas en las instituciones universitarias en donde se selecciona la opción tachada en la encuesta manual. No se debe de guardar el registro hasta haber concluido de capturar y seleccionar todas las opciones múltiples de todas las pestañas del formulario.

Rutina	DIARIO	3 DIAS X SEM	1 o 2 X SEM	NUNCA
DELTOIDES	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
GEMELOS	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
SALTOS AL TABLERO	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
ABDOMINALES CORTOS	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
ABDOMINALES COMPLETOS	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
TRAPECIO	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
DORSALES	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
ESCALERAS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
POLAINAS	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
ESCALERAS CON POLAINAS	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
GLUTEOS	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Rutinas de la encuesta

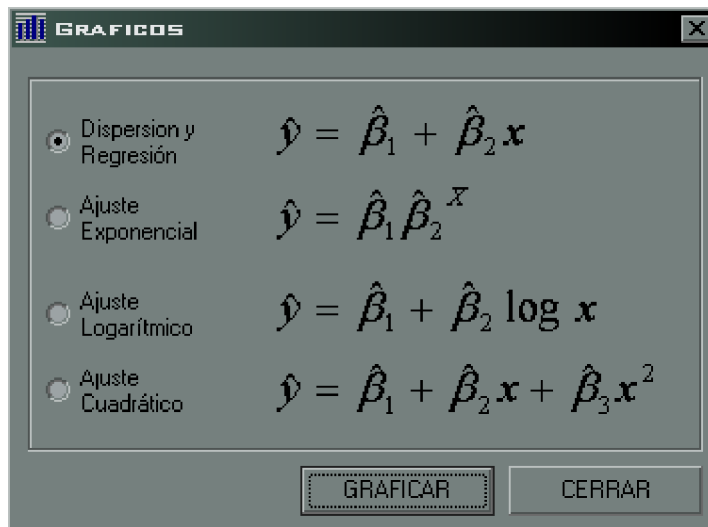
Después de que se hayan capturado todas y cada una de las rutinas en el formulario entonces si se procede a guardar el registro. Una vez que se haya guardado, el consecutivo aumentará en $n+1$ y todas las rutinas se inicializarán en la opción [Nunca] y los demás datos aparecerán en blanco en la primera pestaña.

Actividad	DIARIO	3 DIAS X SEM	1 o 2 X SEM	NUNCA
CUADRICEPS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
BICEPS CRURAL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
ARRANCONES	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
SALTOS CON VALLAS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
CORRER SOB LAS PUNTAS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
ELASTICIDAD	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
ISOMETRICOS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Rutinas de la encuesta

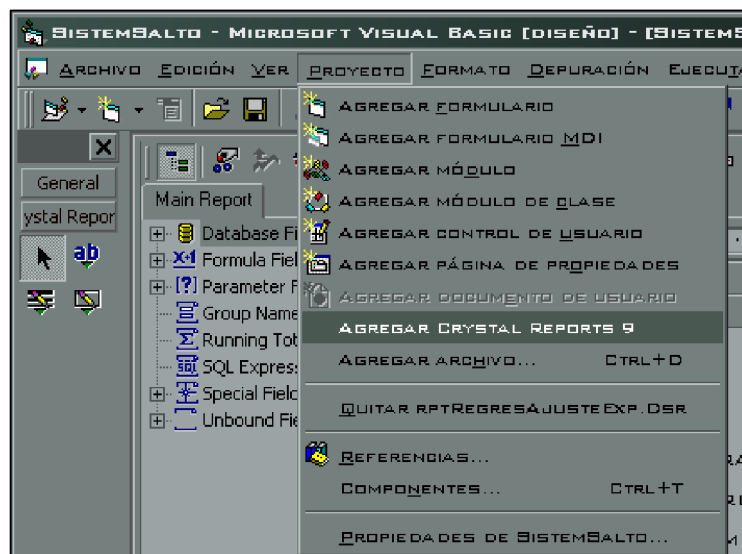
4.2.3.2 Módulo de graficación de regresión y ajuste

Dentro de este formulario se han colocado las opciones de Dispersión y Regresión, Ajustes: Exponencial, Logarítmico y Cuadrático. De manera que sólo hay que seleccionar el gráfico que se desea analizar seguido del botón [Graficar]. Este Formulario llama los visores de Crystal Reports 9.0 insertados en los formularios de Visual Basic, dependiendo de la selección es el gráfico de reporte que se ejecuta. El código fuente se encuentra en el Anexo A.

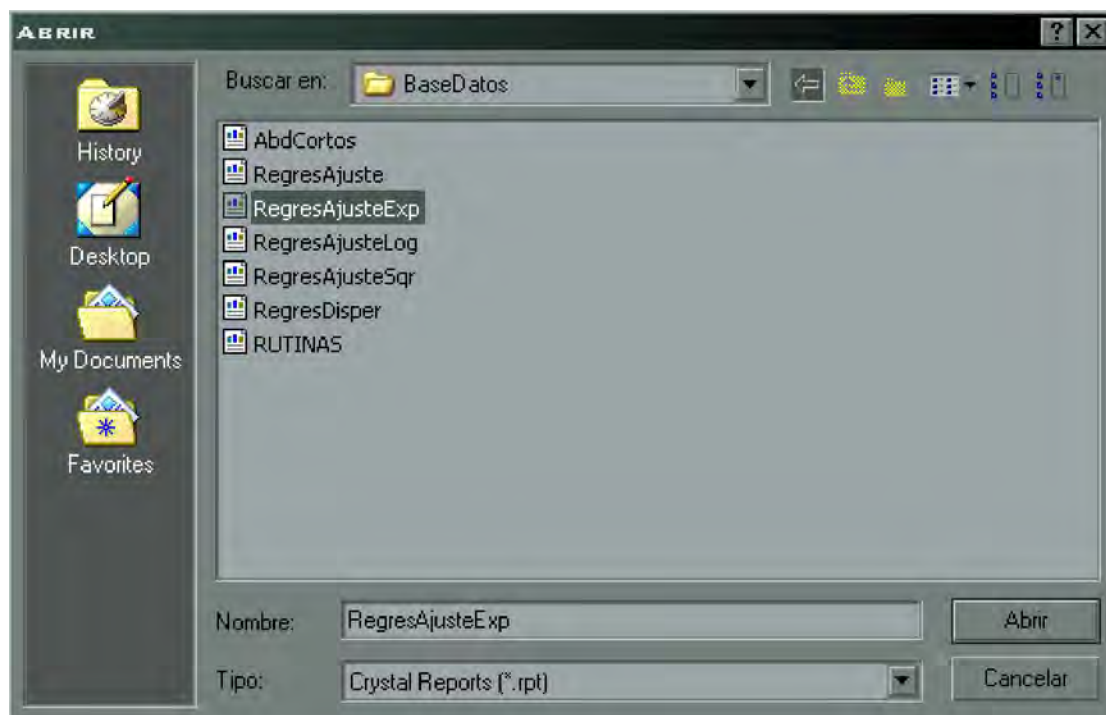
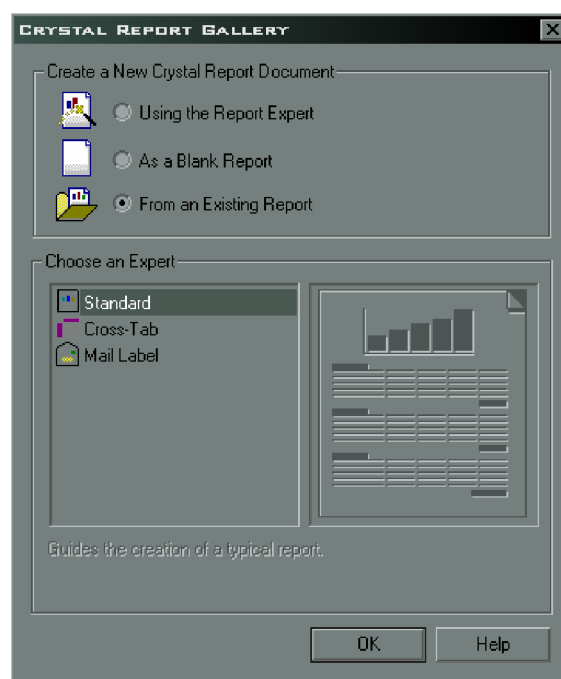


4.2.3.3 Armado del sistema

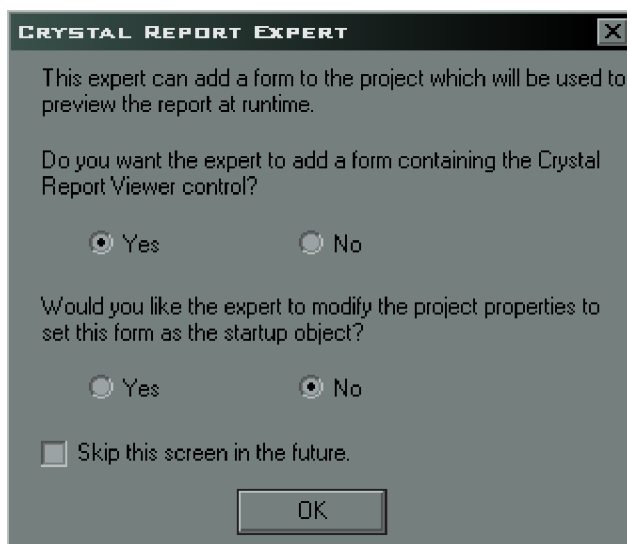
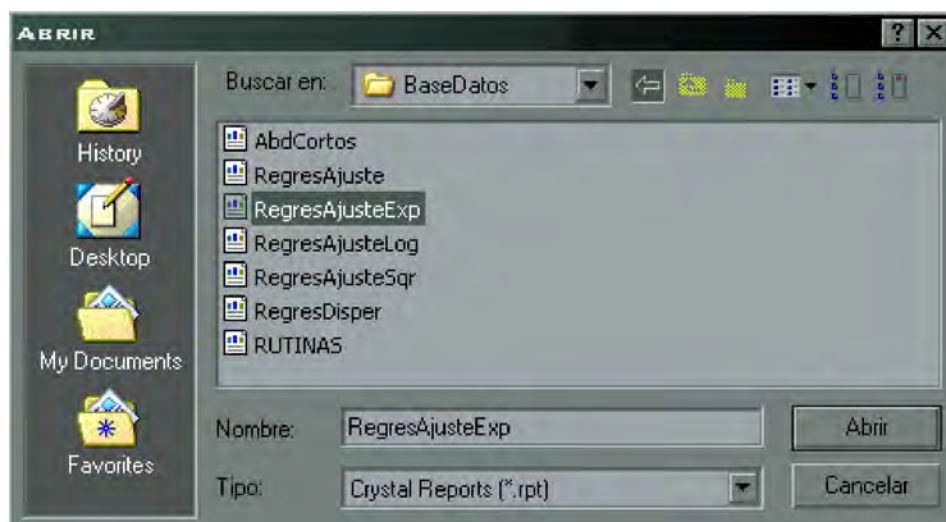
Para agregar al desarrollo de Visual Basic los reportes que ya se desarrollaron se selecciona de la barra de menús la siguiente opción para incluir un formulario y visor de Crystal Reports en la aplicación.



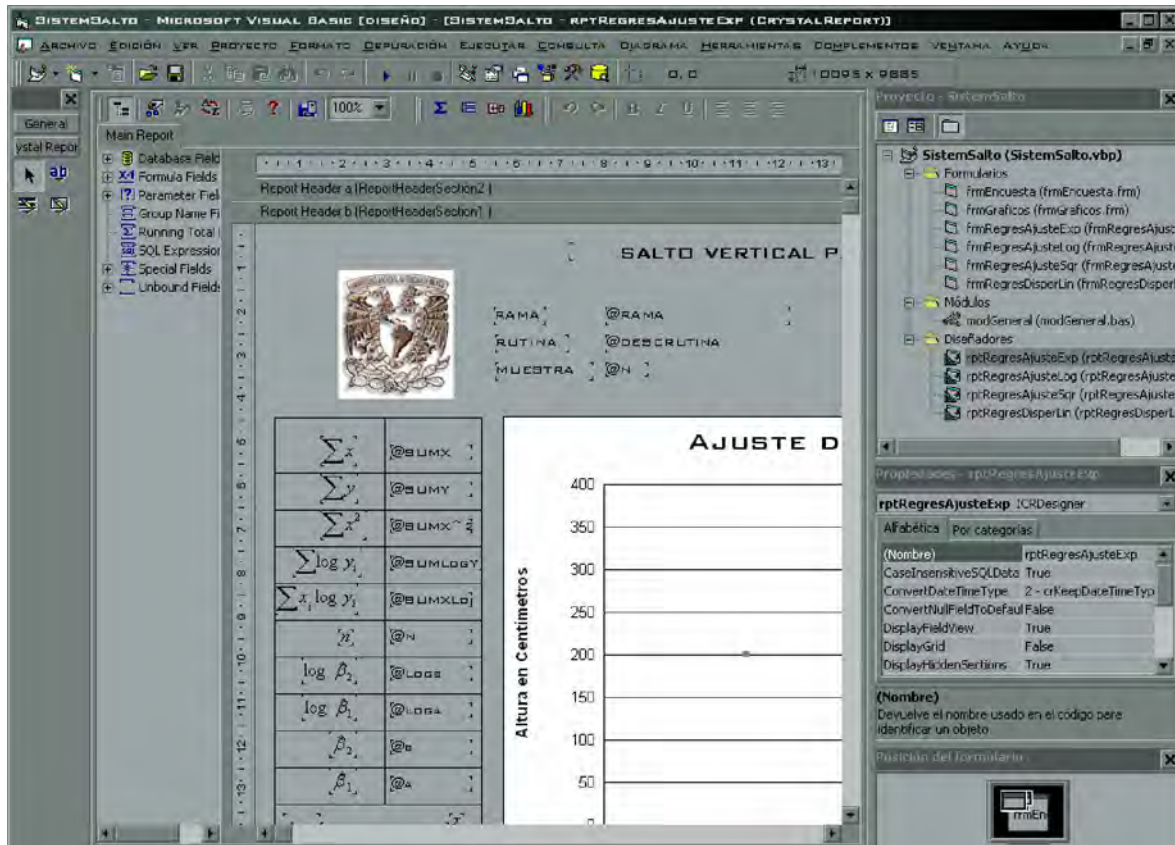
Se selecciona un reporte existente, puesto que ya se ha desarrollado el reporte de Crystal Reports. Se selecciona la ruta donde quedó guardado el reporte.



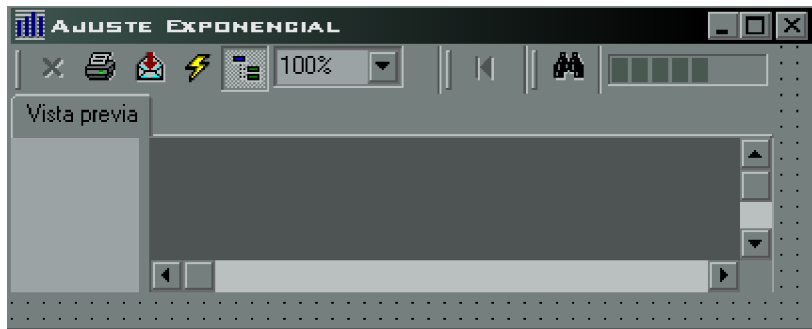
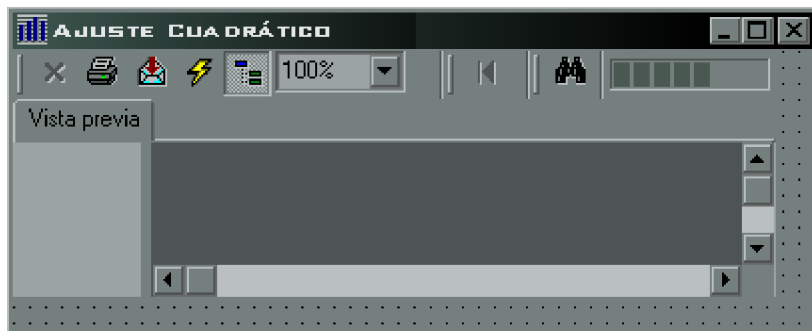
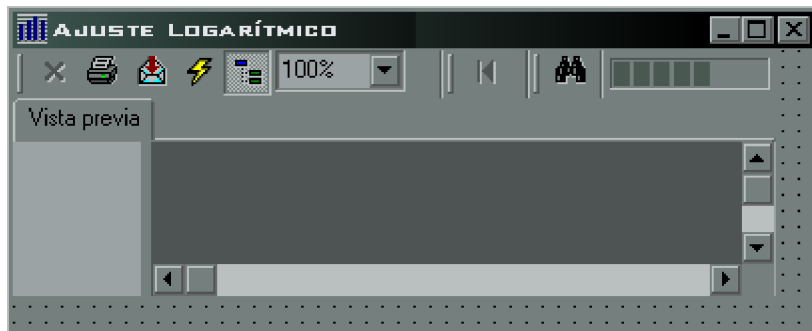
Al momento de importar el desarrollo de Crystal Reports al desarrollo de Visual Basic no debe de interferir con el formulario que se ejecuta al inicio de la aplicación. Por ese motivo en la opción que aparece en la parte posterior como [Startup Object], se debe seleccionar No.



Después el desarrollo del reporte de Crystal Reports se incluye en un formulario que se abre automáticamente. Dentro de este formulario sólo hay que escribir el código fuente para controlar el acceso al mismo y las propiedades de la ventana. No hay que olvidar que este desarrollo ya se hizo en capítulos anteriores para cada una de las gráficas de regresión y ajustes.



Los formularios contienen un visor incluido donde se muestra el reporte de Crystal. Por omisión se genera el nombre del reporte como CrystalReports1, CrystalReports2, . . . ,CrystalReports(n). De la misma forma los formularios se generarán como form1, form2, ..., form(n). Aquí se le dan formatos a los nuevos formularios, tamaños, comportamientos, etcétera.



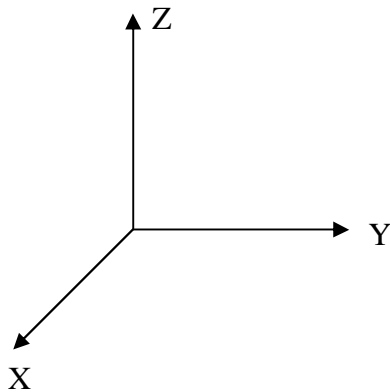
*"Las matemáticas
son el alfabeto con el cual
Dios ha escrito el Universo."
"...está escrito en el lenguaje matemático
y los símbolos son triángulos,
círculos y otras figuras geométricas
sin la ayuda de las cuales
es imposible concebir una sola palabra de él,
y sin las cuales
uno vaga inútilmente
por un oscuro laberinto."*

GALILEO GALILEI

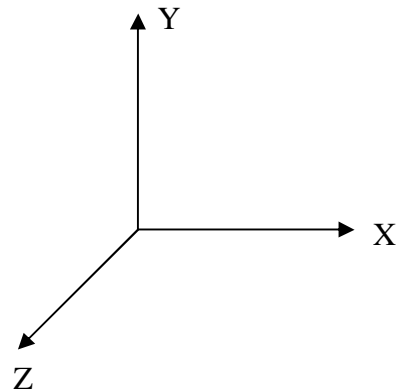
5. Graficación del modelo humano en 3D

Como se había analizado en los capítulos anteriores, existen varios métodos para encontrar el centro de gravedad principal en un atleta. Algunos de ellos usados en biomecánica han sido los del tipo de básculas triangulares ó el método segmentario. Sin embargo se pueden idear métodos alternativos para encontrar el centro de gravedad principal, como es el caso de graficación en 3D. Por otro lado se pueden visualizar los gráficos de los saltos parabólicos de forma espacial acordes a la velocidad y grados de inclinación. En este capítulo se desarrolla un modelo humano con sus 14 segmentos utilizados normalmente en biomecánica. Los desarrollos se efectúan en el lenguaje de programación Visual C++ versión 6.0 con librerías en OpenGL 1.2.

En OpenGL los ejes coordenados se encuentran colocados en diferente posición que en el espacio vectorial usado normalmente. Esto sirve para poder visualizar gráficos en 2 dimensiones y al mismo tiempo graficar en el espacio tridimensional sin rotar los ejes principales.



Ejes coordenados normales



Ejes en OpenGL

5.1 Transformaciones geométricas en 2D

Las principales transformaciones geométricas usadas en OpenGL son: Traslación, Rotación y Escalamiento.

5.1.1 Traslación 2D

Se puede trasladar un punto en el plano XY a una nueva posición sumando una cantidad de x paralela al eje de las $X's$ y una cantidad de y paralela al eje de las $Y's$ para obtener un nuevo punto $P'(x', y')$

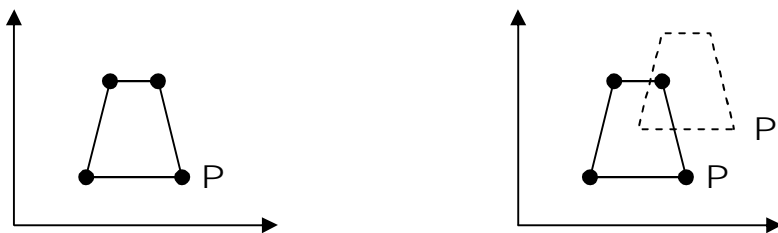
$$x' = x + dx$$

$$y' = y + dy$$

Al expresar estos valores en vector columna se tiene:

$$P = \begin{bmatrix} x \\ y \end{bmatrix} \quad T = \begin{bmatrix} dx \\ dy \end{bmatrix} \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix}$$

La traslación de un objeto se obtiene aplicando la siguiente fórmula a cada uno de los puntos que la forman: $P' = P + T$



5.1.2 Escalamiento 2D

Los puntos se pueden escalar con respecto al eje x y/o al eje y con las siguientes multiplicaciones:

$x' = S_x \cdot x$ $y' = S_y \cdot y$ que en forma matricial se tiene que:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad P' = S \cdot P$$



El escalamiento es con respecto al origen, la proporción del objeto cambia cuando el escalamiento de x es diferente de y : $S_x \neq S_y$. Si alguno de los valores de S es negativo, se obtiene un efecto de reflexión.

5.1.3 Rotación 2D

Los puntos se pueden rotar en un ángulo θ con respecto al origen usando la fórmula matricial:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\text{sen}\theta \\ \text{sen}\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad P' = RP$$



Se pueden manejar todas las transformaciones en forma homogénea. Mientras que la traslación es una suma y el escalamiento y rotación es una multiplicación, se pueden manejar todas las transformaciones en forma multiplicativa expresándolas como matrices homogéneas.

$$\begin{array}{l} \text{Traslación} \\ \text{Escalamiento} \\ \text{Rotación} \end{array} \begin{array}{l} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \\ \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\text{sen}\theta & 0 \\ \text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \end{array}$$

5.2 Transformaciones geométricas en 3D

En las transformaciones tridimensionales también se pueden usar coordenadas homogéneas, las cuáles son representadas por matrices de 4x4 y un punto (x, y, z) se representa con (x, y, z, w) . Las matrices de transformación para 3 dimensiones son:

5.2.1 Traslación 3D

$$\text{Traslación} \quad T(d) = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

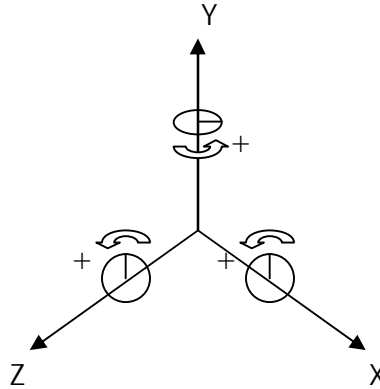
5.2.2 Escalamiento 3D

$$\text{Escalamiento} \quad S(s) = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.2.3 Rotación 3D

$$\begin{array}{l}
 \text{Rotación en el eje } X \\
 \text{Rotación en el eje } Y \\
 \text{Rotación en el eje } Z
 \end{array}
 \quad
 \begin{array}{l}
 R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\text{sen}\theta & 0 \\ 0 & \text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \text{sen}\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sen}\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 R_z(\theta) = \begin{bmatrix} \cos\theta & -\text{sen}\theta & 0 & 0 \\ \text{sen}\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{array}$$

El sentido de la rotación siguen la regla de la mano derecha, de tal manera que las rotaciones de los ejes hacia el lado positivo de esta regla se comporta de esta forma:



5.3 Diseño de un modelo humano

El cuerpo humano es uno de los diseños más complejos de imitar, especialmente al momento de crear un modelo tridimensional. Sin embargo en este caso el modelo de los segmentos del mismo, se pueden asemejar a figuras geométricas aproximadas al cuerpo natural. Eso debido a que nuestro objeto de estudio es en cuanto a la posición de los centros de gravedad en el espacio y el centro de gravedad total del sistema humano.

5.3.1 Medidas y pesos corporales

Ahora bien, cada centro de gravedad de cada segmento del cuerpo humano ha sido calculado por el método segmentario y mejorado por otro método que consiste en un escáner de rayos gamma, cuya muestra fue calculada con 100 personas: 56 estudiantes de educación física, 26 estudiantes de carreras técnicas y 18 fueron personas que realizaban una actividad normal.

Localización del CG de los segmentos corporales expresado como porcentaje de la distancia entre dos puntos a partir del proximal.		
Segmento	Hombres	Mujeres
Cabeza	59.76	58.94
Torzo	44.86	41.51
Brazo	57.72	57.54
Antebrazo	45.74	45.59
Mano	79	74.74
Pierna	40.95	36.12
AntePierna	44.59	44.16
Pie	44.15	40.14

Localización del CG de los segmentos corporales expresado en porcentaje de la distancia entre dos puntos: Proximal y Distal (Varonil)					
Segmentos en el modelo de 3D a desarrollar		Porcentaje de distancia al CG		Distancia al CG	
Segmento	Longitud	% del proximal	% del distal	Del Proximal	Del Distal
Cabeza	4.25	59.76	40.24	2.5398	1.7102
Torzo	8.5	44.86	55.14	3.8131	4.6869
Brazo	5.5	57.72	42.28	3.1746	2.3254
Antebrazo	5	45.74	54.26	2.287	2.713
Mano	2.5	79	21	1.975	0.525
Pierna	8	40.95	59.05	3.276	4.724
AntePierna	7	44.59	55.41	3.1213	3.8787
Pie	3.5	44.15	55.85	1.54525	1.95475

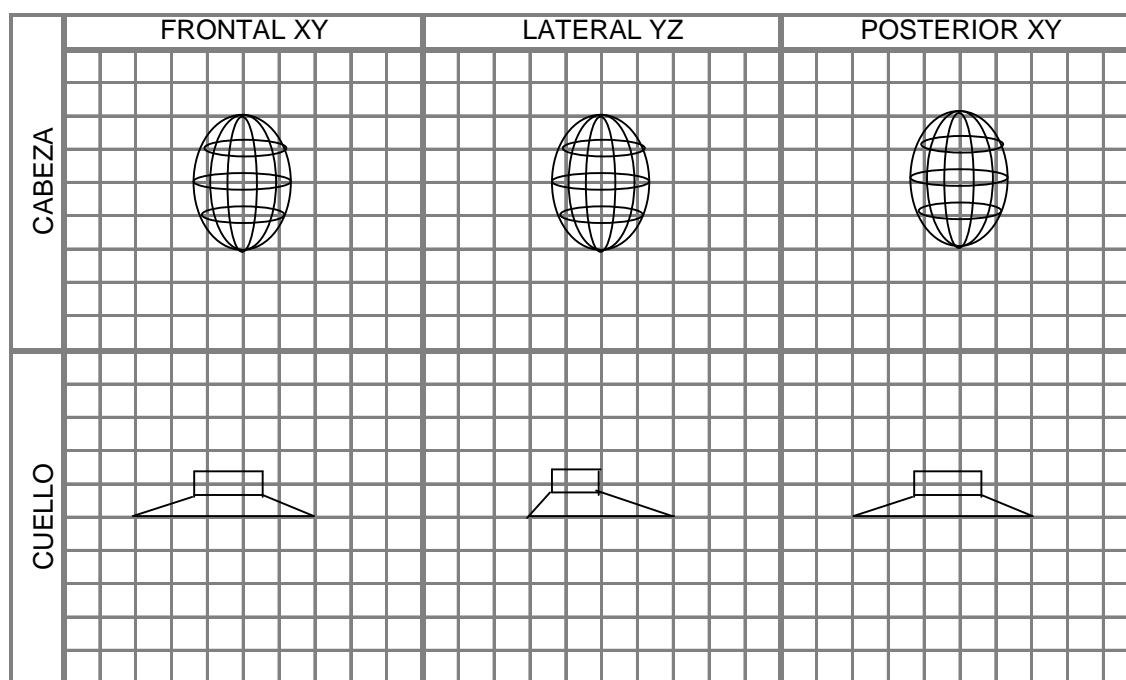
Localización del CG de los segmentos corporales expresado en porcentaje de la distancia entre dos puntos: Proximal y Distal (Femenil)					
Segmentos en el modelo de 3D a desarrollar		Porcentaje de distancia al CG		Distancia al CG	
Segmento	Longitud	% del proximal	% del distal	Del Proximal	Del Distal
Cabeza	4.25	58.94	41.06	2.50495	1.74505
Torzo	8.5	41.51	58.49	3.52835	4.97165
Brazo	5.5	57.54	42.46	3.1647	2.3353
Antebrazo	5	45.59	54.41	2.2795	2.7205
Mano	2.5	74.74	25.26	1.8685	0.6315
Pierna	8	36.12	63.88	2.8896	5.1104
AntePierna	7	44.16	55.84	3.0912	3.9088
Pie	3.5	40.14	59.86	1.4049	2.0951



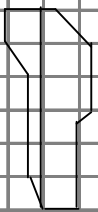



El peso de cada segmento de acuerdo a las pruebas que se han hecho con el método segmentario está expresado en porcentajes del peso total del cuerpo como sigue.

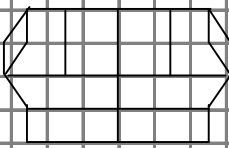
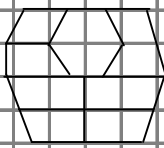
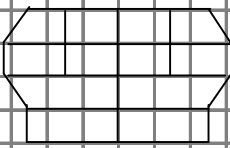

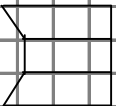
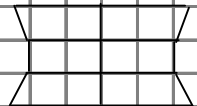
Pesos de los segmentos corporales relativos al peso total del cuerpo en % de 1		
Segmento	Hombres	Mujeres
Cabeza y cuello	0.0694	0.0668
Tronco	0.4346	0.4257
Brazo	0.0271	0.0255
Antebrazo	0.0162	0.0138
Mano	0.0061	0.0056
Muslo	0.1416	0.1478
Pierna	0.0433	0.0481
Pie	0.0137	0.0129

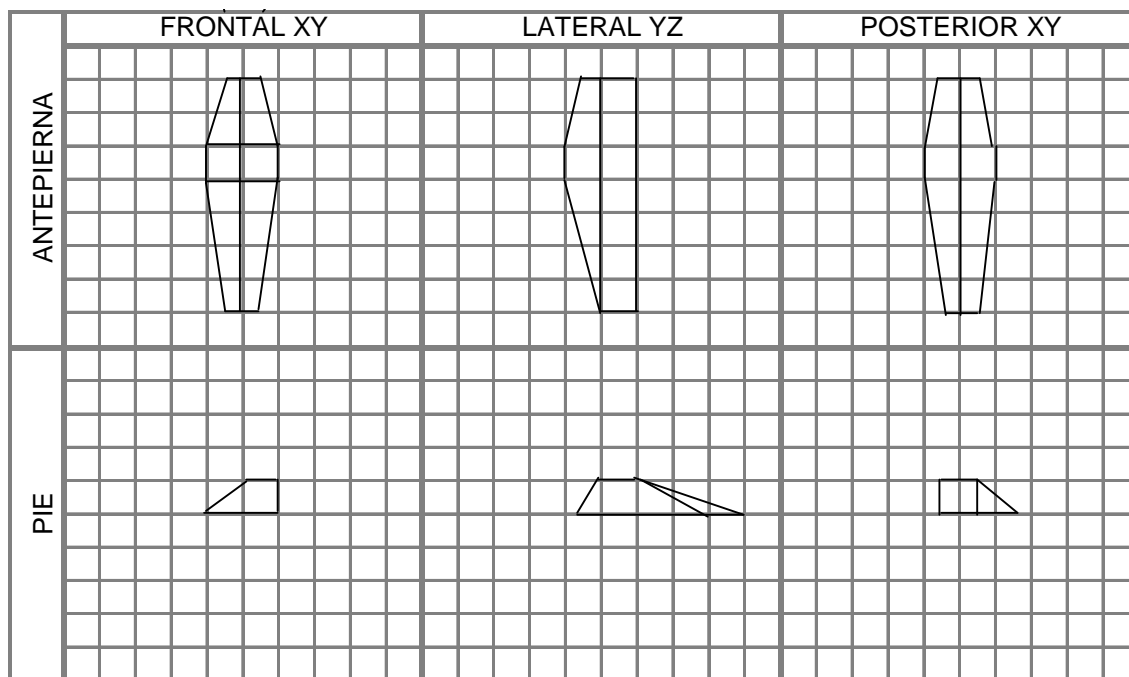
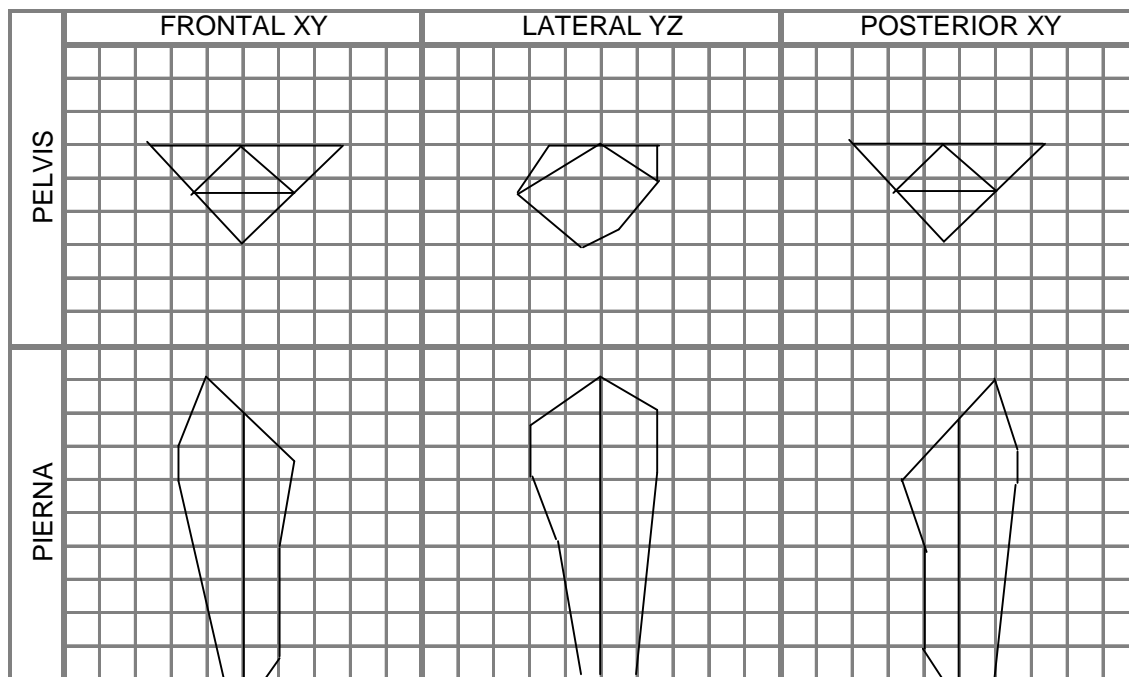
5.3.2 Diseño de segmentos del cuerpo en una cuadrícula

El modelo se ha diseñado de tal manera que 1 cm equivale a 0.159 de la unidad entera en la graficación, por lo que el modelo de 1.80 mts mide 28.75 unidades graficas. Ahora bien, para desarrollar el modelo del cuerpo, en este caso se ha elegido diseñar por separado las figuras de los segmentos en el plano XY (Vista Frontal y Posterior) y en el plano YZ (Vista lateral derecha). A continuación se muestra el diseño del modelo de los 14 segmentos que conforman el cuerpo humano en una cuadrícula también hecha en borrador en hojas milimétricas:

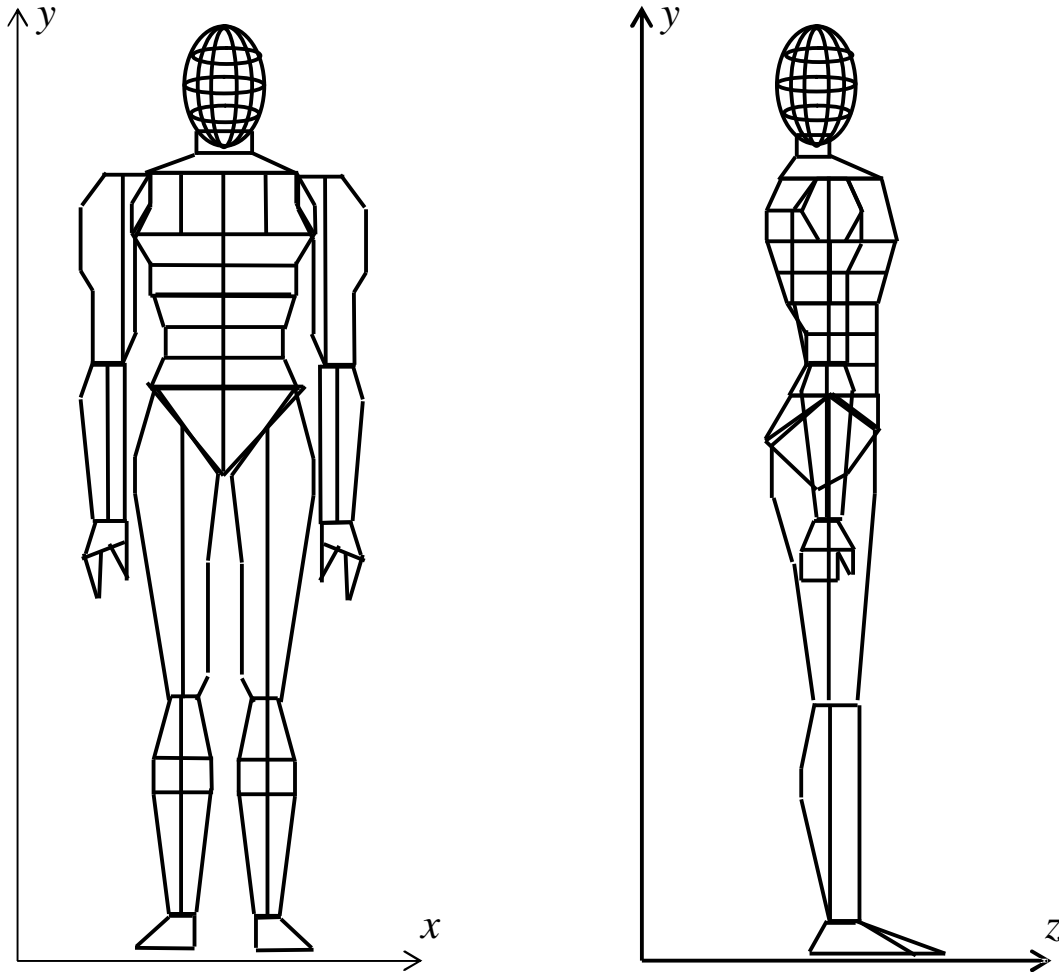


	FRONTAL XY	LATERAL YZ	POSTERIOR XY
BRAZO			
ANEBRAZO / MANO			

	FRONTAL XY	LATERAL YZ	POSTERIOR XY
TORZO			
ABDOMEN			



5.3.3 Ensamblado de Segmentos



El único segmento que se sustituye con una primitiva gráfica de OpenGL, es la cabeza, la cual una forma más aproximada (una esfera escalada de los lados para darle la forma ovalada). El modelo humano una vez en la proyección de OpenGL tendrá una apariencia transparente debido a que en este caso es necesario conocer las posiciones de los centros de gravedad de todos los segmentos del cuerpo y al mismo tiempo conocer la posición que constantemente cambia de lugar del CG total al articular cualquiera de los miembros del modelo. Dichos centros de gravedad están dentro de los segmentos y el CG total estará dentro y fuera del cuerpo, dependiendo de las posiciones deportivas adoptadas por éste.

5.4 Instalación del Lenguaje Visual C++ y OpenGL

Para la instalación del lenguaje Visual C++ 6.0, sólo se instala el Visual Studio 6.0 y se selecciona la opción de Visual C++, luego se siguen las instrucciones para terminar.

Para la instalación de las librerías de OpenGL 1.2, las cuáles están en los archivos: glut.h, glut32.lib, glut32.dll, copiar a las diferentes carpetas como sigue:

Copiar glut.h a la carpeta:

C:\Archivos de programa\Microsoft Visual Studio\VC98\Include\GL

Copiar glut32.lib a la carpeta:

C:\Archivos de programa\Microsoft Visual Studio\VC98\Lib

Copiar glut32.dll a la carpeta:

C:\WINDOWS\system

5.5 Programación en OpenGL y Visual C++ del Centro de Gravedad

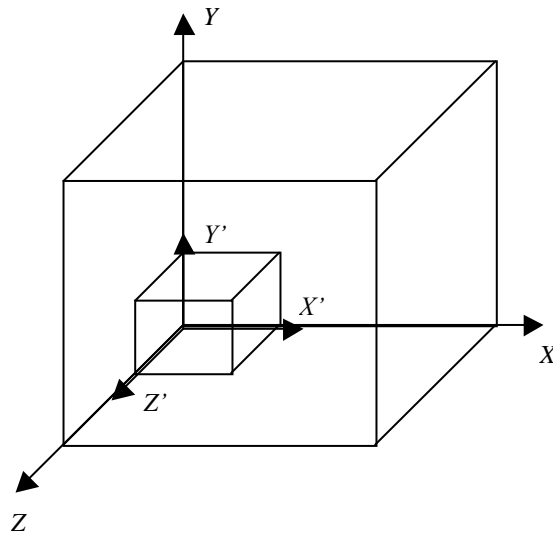
OpenGL contiene funciones para dibujar primitivas gráficas como puntos, líneas, etc, en 2 y 3 dimensiones. También contiene otras para graficar figuras básicas prediseñadas como cubos, esferas, aros, poliedros y demás. En el modelo del cuerpo humano se ocupan puntos, líneas unidas por varios puntos en el espacio que forman los segmentos corporales y sólo una figura prediseñada para la cabeza del cuerpo, en este caso es una esfera escalada para cambiar la proporción de los lados y darle una forma alargada. Como se había mencionado al principio, OpenGL contiene transformaciones como Traslación, Escalamiento y Rotación, las que se usan en este modelo para hacer rotar la cabeza, los brazos, antebrazos, tronco, piernas y antepiernas. En el modelado del cuerpo humano para centros de gravedad se considera despreciable sólo en este caso la rotación de las manos.

OpenGL utiliza funciones matriciales para hacer las transformaciones geométricas que se mencionan al principio, de la misma manera que la parte teórica traslada, escala y rota los objetos en el espacio para poderlos visualizar. Las transformaciones se van concatenando y si se quiere rotar un brazo del cuerpo esto hace rotar toda la escena. Para que esto no suceda se deben manejar varias matrices de modelo, las cuales se almacenan en una pila de nombre "modelview matrix stack" de manera que las transformaciones que afecten a una matriz de modelo se apliquen únicamente a la que se encuentra arriba de la pila de matrices agregadas, esto significa que en 3D se comportaría como un espacio tridimensional nuevo dentro del que ya existe, si se agrega otra matriz a la pila de matrices, en 3D se parecerá a abrir otro espacio dependiente del último espacio tridimensional al momento de crear la escena. Los comandos para manipular la pila de matrices son:

glPushMatrix(); Para introducir elementos a la matriz.
glPopMatrix(); Para sacarlos de la matriz.

Las matrices en OpenGL desde el punto de vista gráfico se comportan como espacios vectoriales dentro de otros al momento de dibujar una escena.

```
glPushMatrix(); // Espacio grafico (X,Y,Z)
    glPushMatrix(); // Subespacio gráfico (X',Y',Z')
        //Objetos de dibujo
    glPopMatrix();
glPopMatrix();
```



En el modelo humano se tienen segmentos cuyos movimientos dependen de otros segmentos, por ejemplo: en la graficación la mano está en un subespacio vectorial sujeta al espacio del antebrazo, el que a su vez depende del superespacio brazo y a su vez éste subespacio depende del espacio tronco del cuerpo humano. Eso quiere decir que si se mueve de lugar el tronco, se mueve también sus subespacios: brazo-antebrazo-mano, porque estos dependen del tronco. Si se mueve de lugar el brazo se mueven también sus subespacios antebrazo-mano. Y cuando se mueve el antebrazo se mueve el subespacio mano.

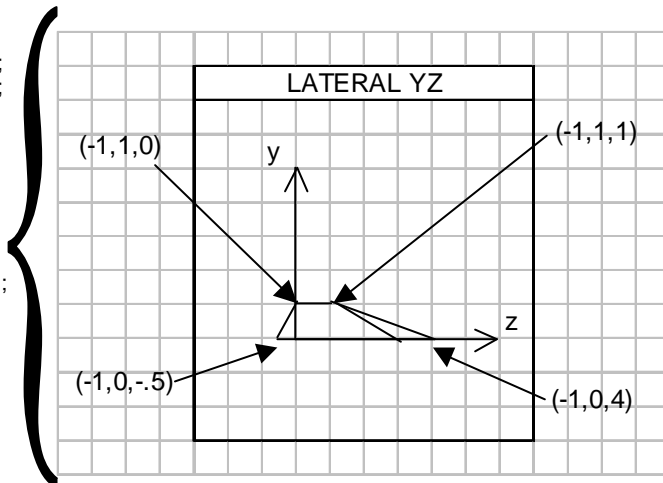
En el modelo, las matrices nos ayudan para poder articular cada uno de los segmentos del cuerpo al momento de desarrollarlo en OpenGL. En este caso las transformaciones de traslación y rotación de los segmentos son la más utilizadas.

Los comandos más utilizados para la graficación del modelo además de las matrices de OpenGL son:

```
glVertex3f(x,y,z);           // Grafica un punto en la coordenada especificada.
glBegin(GL_LINE_LOOP);     // Une varios puntos por medio de líneas cíclicas.
glTranslatef(x,y,z);       // Traslación de objetos en las coordenadas especificadas.
glScalef(x,y,z);          // Escala el objeto en la dimensión especificada de cada eje.
glRotatef ((GLfloat) grados, x, y, z); // Rota en los grados y ejes especificados.
```

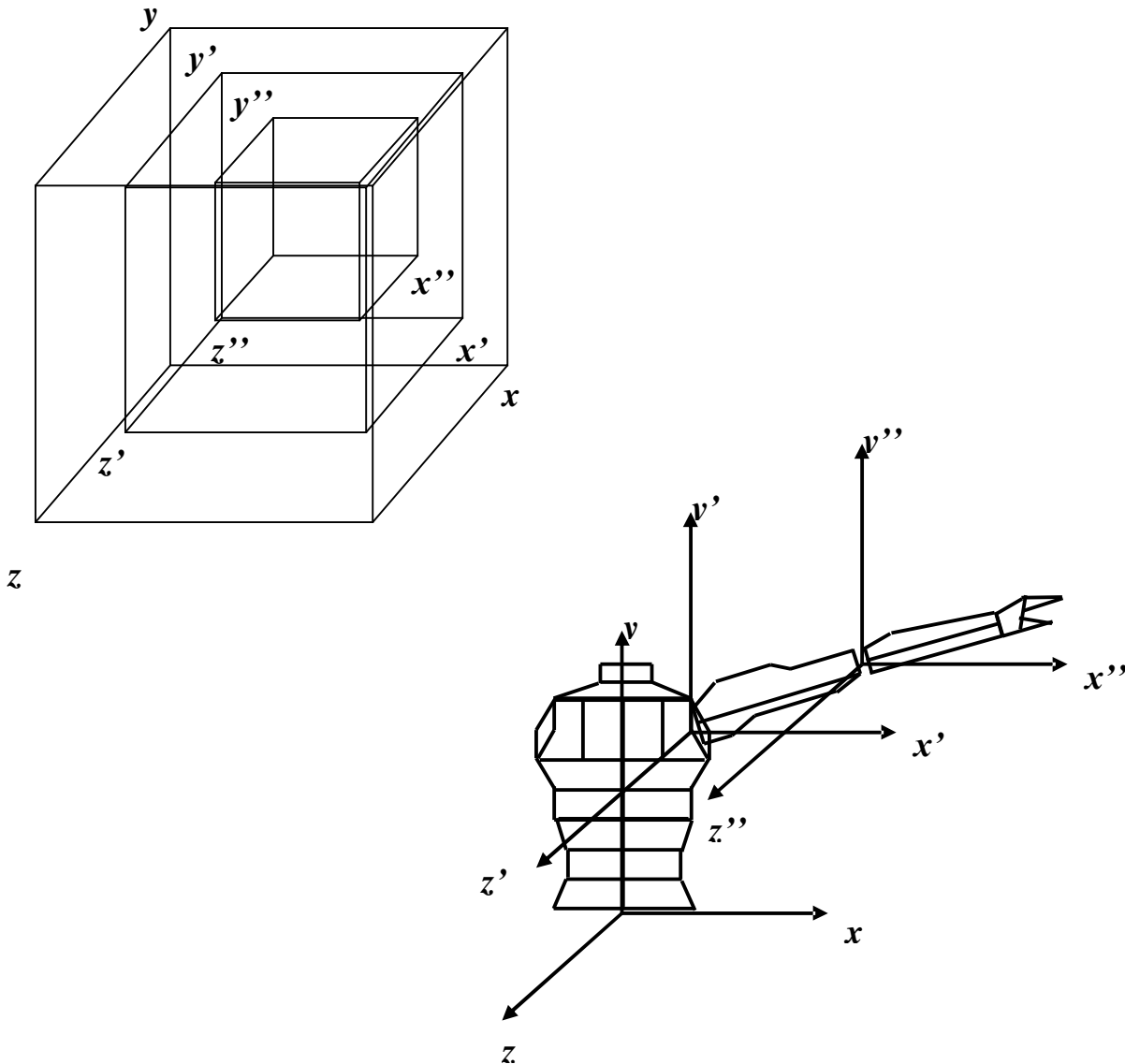
A continuación se muestra como graficar el segmento del cuerpo pie derecho con código OpenGL, vista lateral basándose en el diseño ya efectuado en la hoja milimétrica:

```
//PIE DERECHO
glPushMatrix();
  glBegin(GL_LINE_LOOP);
    glVertex3f(-1,0,4);
    glVertex3f(-1.5,0,4);
    glVertex3f(-2.5,0,3);
    glVertex3f(-2,1,1);
    glVertex3f(-1,1,1);
  glEnd();
  glBegin(GL_LINE_LOOP);
    glVertex3f(-2,1,1);
    glVertex3f(-2.5,0,3);
    glVertex3f(-2,0,-.5);
    glVertex3f(-2,1,0);
  glEnd();
  glBegin(GL_LINE_LOOP);
    glVertex3f(-1,0,-.5);
    glVertex3f(-2,0,-.5);
    glVertex3f(-2,1,0);
    glVertex3f(-1,1,0);
  glEnd();
  glBegin(GL_LINE_LOOP);
    glVertex3f(-1,1,0);
    glVertex3f(-1,1,1);
    glVertex3f(-1,0,4);
    glVertex3f(-1,0,-.5);
  glEnd();
glPopMatrix();
//FIN PIE DERECHO
```



5.5.1 Rotación de los segmentos del modelo

La rotación de los segmentos del modelo del cuerpo humano, como ya se había comentado se parece mas a un subespacio dentro de otro espacio, y esto permite obtener una rotación de cada uno de los miembros del modelo sin afectar a todo el sistema coordinado. Esto ocurre al hacer una traslación y luego una rotación de los ejes.

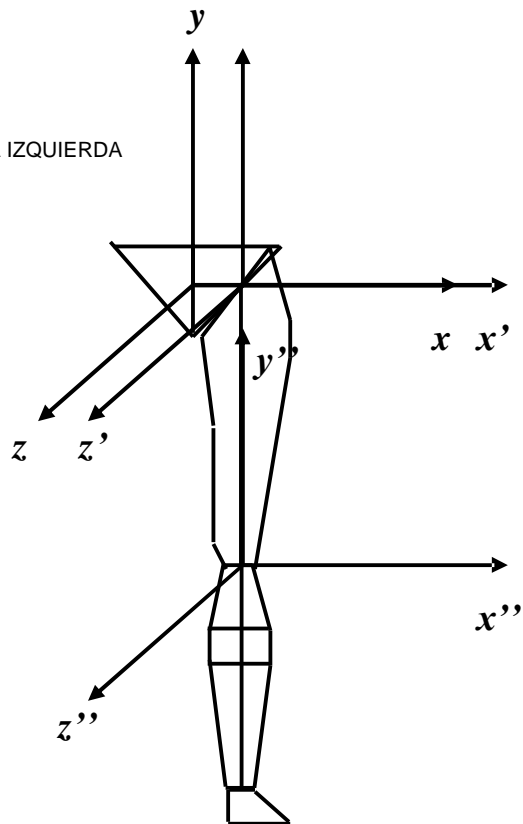


```

glRotatef ((GLfloat) globalx, 1.0, 0.0, 0.0);
glRotatef ((GLfloat) globaly, 0.0, 1.0, 0.0);
glRotatef ((GLfloat) globalz, 0.0, 0.0, 1.0);
//PELVIS
glPushMatrix();
//...

glPushMatrix();
glTranslatef(0,16, 0);
glRotatef ((GLfloat) Pier Izq
glTranslatef(0, 16, 0);
glPushMatrix();
//...
glPopMatrix();
//ANTEPIERNA IZQUIERDA
glPushMatrix();
glTranslatef(0,8, 0);
glRotatef ((GLfloat) AntePier Izq, 1.0, 0.0, 0.0);
glTranslatef(0, -8, 0);
glPushMatrix();
//...
glPopMatrix();
//PIE IZQUIERDO
glPushMatrix();
//...
glPopMatrix();//FIN PIE
glPopMatrix();//FIN ANTEPIERNA IZQUIERDA
glPopMatrix();//FIN PIERNA IZQUIERDA
//...
//...
glPopMatrix();//FIN PELVIS
//...
//...

```



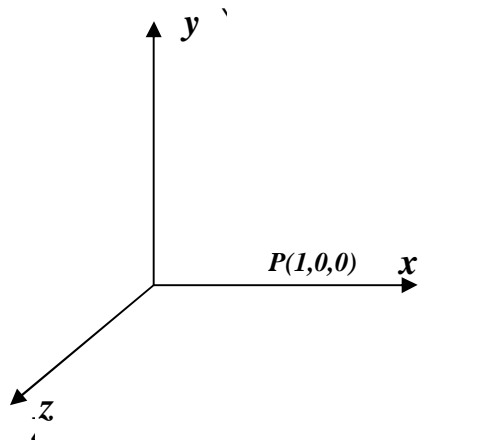
5.5.2 Calculando el centro de gravedad

Como es de esperarse, al momento de trasladar los ejes a otra posición y después rotarlos a cierta inclinación, aunque se hayan especificado los puntos exactos del dibujo, estos han sido trasladados y rotados en la proyección. En otras palabras, cuando se hace una transformación, las constantes originales no cambian de valor, sin embargo en la proyección aparecen en otro sitio porque han sido transformadas. Ahora bien, para obtener el punto exacto en el espacio aunque ya hayan sido trasladados, escalados y rotados en los sub-ejes, es necesario saber la coordenada exacta en el espacio tridimensional principal. Esto es para poder hacer los cálculos en el desarrollo y obtener las coordenadas exactas de los centros de gravedad. Por esto se toma apoyo en las matrices citadas al principio del capítulo.

Por ejemplo, si se tiene el punto $P(x, y, z)$ cuyos valores son $x = 1$; $y = 0$; $z = 0$.

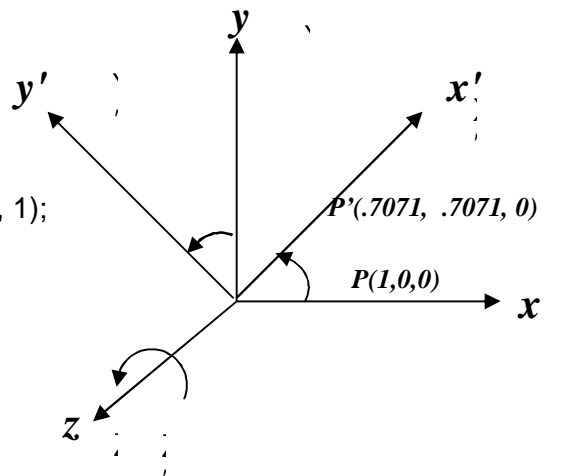
En OpenGL se graficaría con el código:

```
glPushMatrix();
    glVertex3f(1, 0, 0);
glPopMatrix();
```



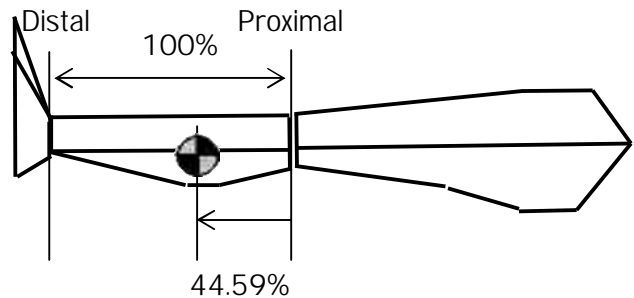
Si se rota este punto a 45° con respecto al eje Z , el punto seguirá siendo el mismo dentro de las coordenadas originales escritas en OpenGL, pero es transformado. Es decir, que se tienen dos coordenadas: primeramente el punto original que no pierde sus valores y el punto transformado por la rotación que finalmente se visualiza en el espacio:

```
glPushMatrix();
    glRotatef ((GLfloat) 45, 0, 0, 1);
    glPushMatrix();
        glVertex3f(1,0,0);
    glPopMatrix();
glPopMatrix();
```

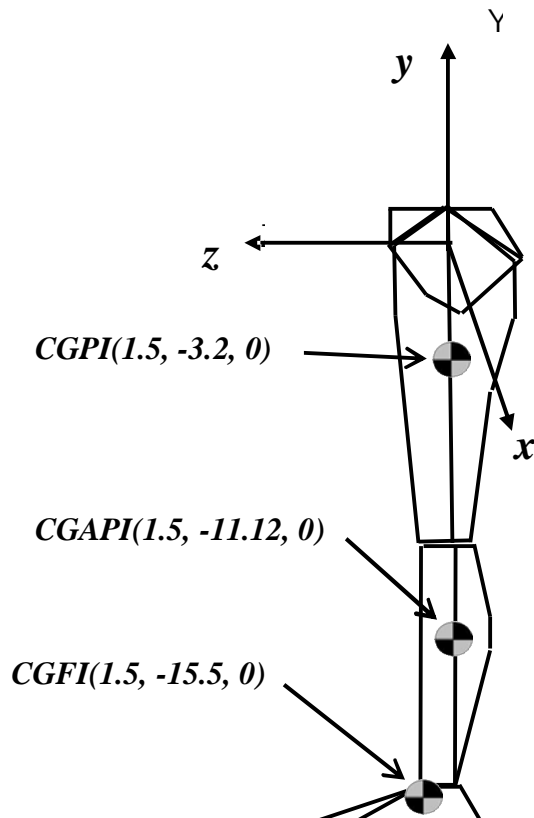


La posición exacta del centro de gravedad de cada segmento se toma en base a las tablas de medidas y pesos corporales ya citadas. En la graficación se toma el tamaño del segmento del cuerpo del modelo y se calcula el porcentaje de distancia al centro de gravedad desde la articulación proximal.

En proporción, la antepierna del modelo mide 7 puntos gráficos. Si el CG está a 44.59% de su articulación proximal \Rightarrow la distancia al CG son 3.12 puntos gráficos porque $7 \cdot (.4459) = 3.12$ desde el proximal y 3.88 desde el distal. Si el proximal está a $(x, -8, z) \Rightarrow -8 - 3.12 = -11.12$
 \therefore La coordenada del CG está en $(x, -11.12, z)$



Ahora bien, para obtener las coordenadas exactas de los centros de gravedad después de cada transformación, es necesario tomar apoyo de la multiplicación de matrices en el mismo código de Visual C++ y OpenGL. Se toma como un ejemplo la posición de los centros de gravedad de las articulaciones de la pierna izquierda en cadera, rodilla y tobillo:



Para la articulación de la pierna izquierda, inicialmente el centro de gravedad se encuentra en el punto: $CGPI(1.5, -3.2, 0)$, el que después de rotarse a θ grados de inclinación ($RotPierIzq$) con respecto al eje de las X 's, se transformará en otro punto en el espacio para ser dibujado. Usando la notación matricial se tiene que:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\text{sen}(\theta) & 0 \\ 0 & \text{sen}(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

Los valores correspondientes son:

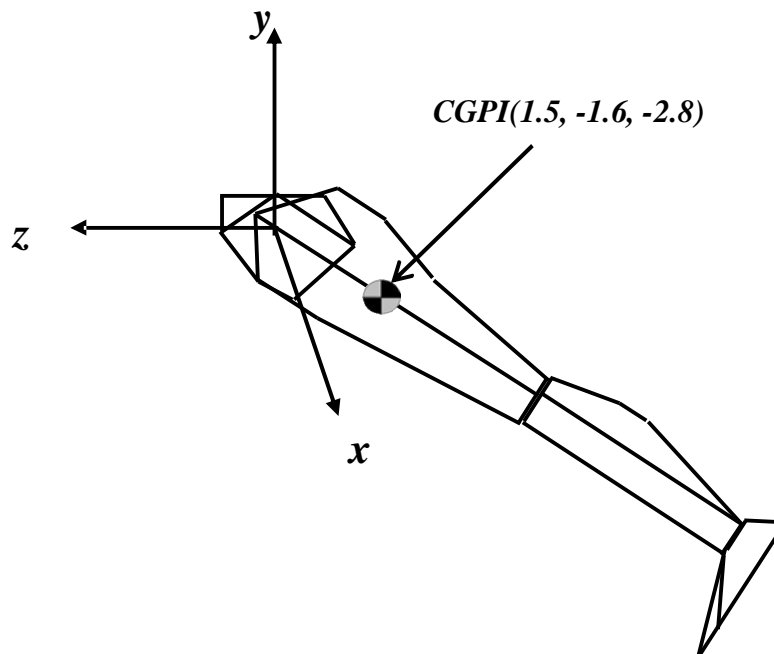
$$RxArtic(RotPiernaIzq) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(RotPiernaIzq) & -\sin(RotPiernaIzq) & 0 \\ 0 & \sin(RotPiernaIzq) & \cos(RotPiernaIzq) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P = \begin{bmatrix} 1.5 \\ -3.2 \\ 0 \\ 1 \end{bmatrix}$$

Si se hace que la rotación $RotPiernaIzq$ se incline a 60° :

$$RxArtic(60) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(60) & -\sin(60) & 0 \\ 0 & \sin(60) & \cos(60) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} P = \begin{bmatrix} 1.5 \\ -3.2 \\ 0 \\ 1 \end{bmatrix}$$

Para obtener la posición del CG de la pierna izquierda una vez rotado en el espacio, se hace un producto de matrices en el siguiente orden: $RxArtic \cdot P = CGPI$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & -0.866 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ -3.2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ -1.6 \\ -2.8 \\ 1 \end{bmatrix}$$



5.5.3 Codificación de matrices en Visual C++ /OpenGL

Desarrollando las matrices en Visual C++, en donde los grados están en radianes, se hace la conversión multiplicando por el valor de $\pi/180$. Luego el CG de la pierna izquierda queda de la siguiente forma:

```
//CG PIERNA IZQUIERDA
void setCGPI(){
    double P[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CADERA IZQ)
    P[0][0]=1.5;
    P[1][0]= -3.27;
    P[2][0]= 0;
    P[3][0]= 1;
    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);
    RxArtic[1][2]=-sin(3.1416*RotPiernalzq/180);
    RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);
    RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);
    RxArtic[3][3]= 1;
    for (i=0;i<=3; i++)
        CGPI[i][0]=0;

    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION//
    //OBTIENE POSICION CG AL ROTAR LA PIERNA
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                CGPI[i][j]= CGPI[i][j]+(RxArtic[i][k]*P[k][j]);

    glVertex3f(CGPI[0][0],CGPI[1][0],CGPI[2][0]);
}
}
```

Para la rotación de la rodilla primero se hace una traslación de ejes hacia abajo de la pierna para hacer rotar la rodilla en el origen a 54° . Los valores correspondientes son:

$$RxArtic(RotPiernalzq) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(RotPiernalzq) & -\sin(RotPiernalzq) & 0 \\ 0 & \sin(RotPiernalzq) & \cos(RotPiernalzq) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$RxSubArtic(RotAntePiernalzq) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(RotAntePiernalzq) & -\sin(RotAntePiernalzq) & 0 \\ 0 & \sin(RotAntePiernalzq) & \cos(RotAntePiernalzq) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$RotAntePiernalzq = 54^\circ$$

$$RxSubArtic(54) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.587 & -0.809 & 0 \\ 0 & 0.809 & 0.587 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 1.5 \\ -3.12 \\ 0 \\ 1 \end{bmatrix}$$

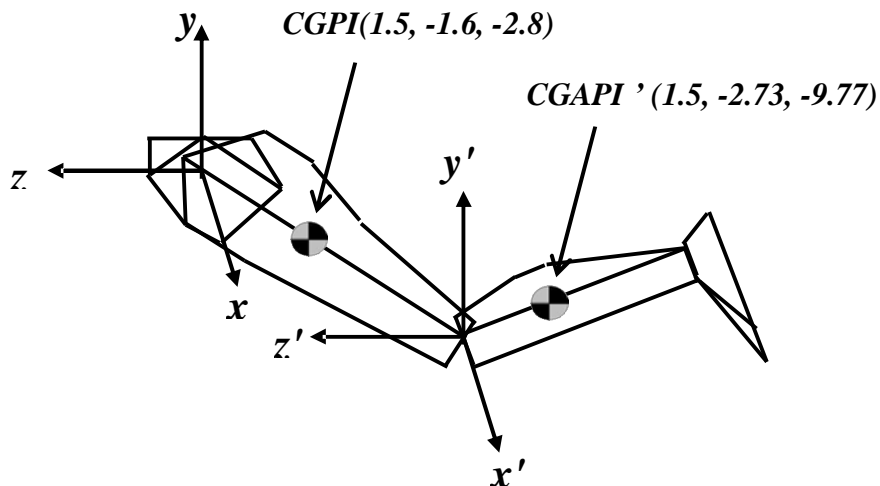
Para obtener la posición del CG de la mencionada antepierna izquierda se hace el siguiente producto matricial: $RxArtic \cdot T_1 \cdot RxSubArtic \cdot P = CGAPI'$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & -0.866 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -8 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.587 & -0.809 & 0 \\ 0 & 0.809 & 0.587 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ -3.12 \\ 0 \\ 1 \end{bmatrix}$$

Expresando la matriz traslación como matriz homogénea se tiene que:

$$RxArtic \cdot T_1 \cdot RxSubArtic \cdot P = CGAPI'$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & -0.866 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -8 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.587 & -0.809 & 0 \\ 0 & 0.809 & 0.587 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1.5 \\ -3.12 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ -2.73 \\ -9.77 \\ 1 \end{bmatrix}$$



El orden del producto matricial en la fórmula matemática, se hace desde la primera articulación (proximal) hasta la última (distal) multiplicada por el punto $P(x, y, z, w)$ donde se encuentra inicialmente el CG. El orden del producto matricial en código Visual C++ se hace multiplicando el punto $P(x, y, z, w)$ por la última articulación (distal) hasta la primera articulación (proximal).

Orden del producto matricial matemático:

$$RxArtic \cdot T_1 \cdot RxSubArtic \cdot P = CGAPI'$$

Orden del producto matricial en Visual C++:

$$A = RxSubArtic \cdot P$$

$$A = A + T_1$$

$$CGAPI' = RxArtic \cdot A$$

Desarrollando las matrices como en el caso anterior en Visual C++, el CG de la antepierna izquierda se calcula y se dibuja con el código:

```
//CG ANTEPIERNA IZQUIERDA
void setCGAPI(){
    double P[MAX][MAX],T1[MAX][MAX];

    //POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)
    P[0][0]= 1.5;
    P[1][0]= -3.12;
    P[2][0]= 0;
    P[3][0]= 1;

    RxSubArtic[0][0]= 1;
    RxSubArtic[1][1]= cos(3.1416*RotAntePiernalzq/180);
    RxSubArtic[1][2]=-sin(3.1416*RotAntePiernalzq/180);
    RxSubArtic[2][1]= sin(3.1416*RotAntePiernalzq/180);
    RxSubArtic[2][2]= cos(3.1416*RotAntePiernalzq/180);
    RxSubArtic[3][3]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);
    RxArtic[1][2]=-sin(3.1416*RotPiernalzq/180);
    RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);
    RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);
    RxArtic[3][3]= 1;

    //TRASLACION
    T1[0][0]= 0;
    T1[1][0]= -8;
    T1[2][0]= 0;
    T1[3][0]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        CGAPI[i][0]=0;
    }

    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                A[i][j]= A[i][j]+(RxSubArtic[i][k]*P[k][j]);

    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)
    for (i=0;i<=3;i++)
        A[i][0]=A[i][0]+T1[i][0];

    //OBTIENE POSICION CG AL ROTAR LA PIERNA
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                CGAPI[i][j]= CGAPI[i][j]+(RxArtic[i][k]*A[k][j]);

    glVertex3f(CGAPI[0][0],CGAPI[1][0],CGAPI[2][0]);
}
```

Para la rotación del tobillo como en el caso de la rodilla se hace una segunda traslación de ejes hacia abajo de la pierna para hacer rotar el tobillo en el origen a 27° . Los valores correspondientes son:

$$RxArtic(RotPiernaIzq) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(RotPiernaIzq) & -\sin(RotPiernaIzq) & 0 \\ 0 & \sin(RotPiernaIzq) & \cos(RotPiernaIzq) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$RxSubArtic(RotAntePiernaIzq) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(RotAntePiernaIzq) & -\sin(RotAntePiernaIzq) & 0 \\ 0 & \sin(RotAntePiernaIzq) & \cos(RotAntePiernaIzq) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$RotAntePiernaIzq = 54^\circ$$

$$RxSubArtic(54) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.587 & -0.809 & 0 \\ 0 & 0.809 & 0.587 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_2 = \begin{bmatrix} 1.5 \\ -7 \\ .5 \\ 1 \end{bmatrix}$$

$$RxSubSubArtic(RotPieIzq) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(RotPieIzq) & -\sin(RotPieIzq) & 0 \\ 0 & \sin(RotPieIzq) & \cos(RotPieIzq) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$RotPieIzq = 27^\circ$$

$$RxSubSubArtic(27) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.891 & -0.453 & 0 \\ 0 & 0.453 & 0.891 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad P = \begin{bmatrix} 0 \\ -0.5 \\ 0.54 \\ 1 \end{bmatrix}$$

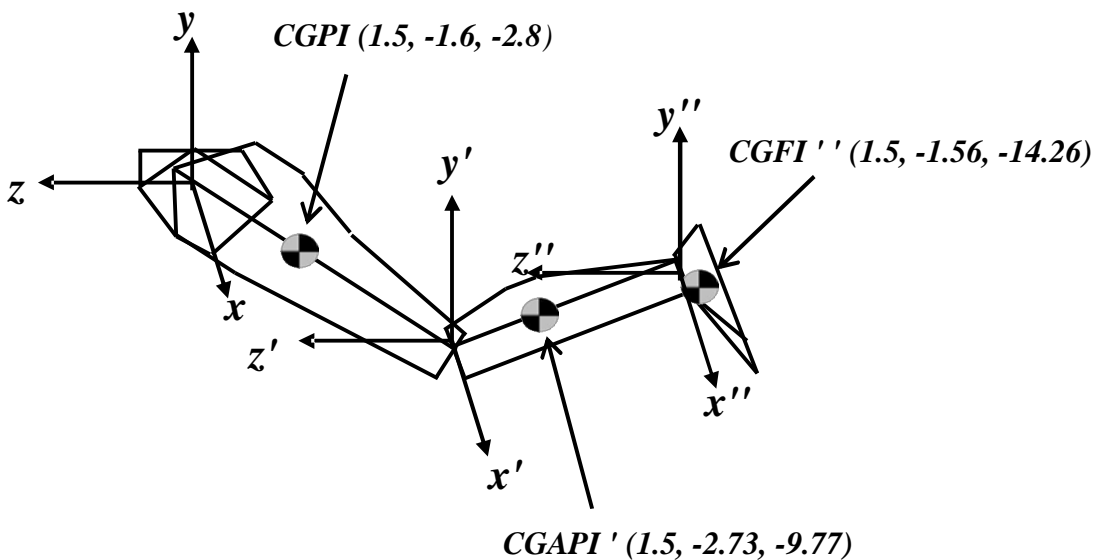
Para obtener la posición del CG del pie izquierdo se hace el siguiente producto matricial: $R_{xArtic} \cdot T_1 \cdot R_{xSubArtic} \cdot T_2 \cdot R_{xSubSubArtic} \cdot P = CGFI''$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.5 & -0.866 & 0 \\ 0 & 0.866 & 0.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -8 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \dots$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.587 & -0.809 & 0 \\ 0 & 0.809 & 0.587 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 1.5 \\ 0 & 1 & 0 & -7 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \dots$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.891 & -0.453 & 0 \\ 0 & 0.453 & 0.891 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ -0.5 \\ 0.54 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.5 \\ -1.561 \\ -14.26 \\ 1 \end{bmatrix}$$

El centro de gravedad del pie izquierdo $CGFI''$ se encuentra ahora doblemente trasladado:



Desarrollando las matrices para CG del pie izquierdo se calcula y se dibuja con el código:

```
//CG PIE IZQUIERDO
void setCGF(){
    double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];

    //POSICION CG A PARTIR DEL PROXIMAL (TOBILLO IZQ)
    P[0][0]= 0;P[1][0]= -0.5;P[2][0]= 0.54;P[3][0]= 1;RxSubSubArtic[0][0]= 1;

    RxSubSubArtic[1][1]= cos(3.1416*RotPielzq/180);
    RxSubSubArtic[1][2]=-sin(3.1416*RotPielzq/180);
    RxSubSubArtic[2][1]= sin(3.1416*RotPielzq/180);
    RxSubSubArtic[2][2]= cos(3.1416*RotPielzq/180);
    RxSubSubArtic[3][3]= 1;

    RxSubArtic[0][0]= 1;
    RxSubArtic[1][1]= cos(3.1416*RotAntePiernalzq/180);
    RxSubArtic[1][2]=-sin(3.1416*RotAntePiernalzq/180);
    RxSubArtic[2][1]= sin(3.1416*RotAntePiernalzq/180);
    RxSubArtic[2][2]= cos(3.1416*RotAntePiernalzq/180);
    RxSubArtic[3][3]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);
    RxArtic[1][2]=-sin(3.1416*RotPiernalzq/180);
    RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);
    RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);
    RxArtic[3][3]= 1;

    //TRASLACION
    T1[0][0]= 0;T1[1][0]= -8;T1[2][0]= 0;T1[3][0]= 1;
    T2[0][0]= 1.5;T2[1][0]= -7;T2[2][0]= .5;T2[3][0]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
        CGF[i][0]=0;
    }

    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL PIE
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                B[i][j]= B[i][j]+(RxSubSubArtic[i][k]*P[k][j]);

    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (TOBILLO IZQ)
    for (i=0;i<=3;i++)
        B[i][0]=B[i][0]+T2[i][0];

    //OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                A[i][j]= A[i][j]+(RxSubArtic[i][k]*B[k][j]);

    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)
    for (i=0;i<=3;i++)
        A[i][0]=A[i][0]+T1[i][0];

    //OBTIENE POSICION CG AL ROTAR LA PIERNA
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                CGF[i][j]= CGF[i][j]+(RxArtic[i][k]*A[k][j]);

    glVertex3f(CGF[0][0],CGF[1][0],CGF[2][0]);
}
}
```

5.5.4 Codificación de Interfaz de teclas de dirección

Al modelo se le agrega la interacción con el usuario para que se puedan manipular desde el teclado las rotaciones de cada segmento, incluyendo los brazos y torzo. Sin embargo por la gran cantidad de código que se ha escrito, se muestra en esta sección sólo una parte del código de interfaz. El código completo se ha puesto en una sección apartada en el Anexo B.

```
void keyboard (unsigned char key, int x, int y){
    //TECLAS DEL MOVIMIENTO DEL CUERPO
    switch (key){
        //Flexión del tronco
        case 'C':
            if (RotCabeza<90) RotCabeza = (RotCabeza + 3) % 360;
            else RotCabeza = 90;

            break;
        //Extensión del tronco
        case 'c':
            if (RotCabeza>-90) RotCabeza = (RotCabeza - 3) % 360;
            else RotCabeza = -90;

            break;
        //Flexión del tronco
        case 'H':
            if (RotTorzo<135) RotTorzo = (RotTorzo + 3) % 360;
            else RotTorzo = 135;

            break;
        //Extensión del tronco
        case 'h':
            if (RotTorzo>-45) RotTorzo = (RotTorzo - 3) % 360;
            else RotTorzo = -45;

            break;
        //Flexion Brazo Derecho
        case 'P':
            if (RotBrazoDer>-180) RotBrazoDer = (RotBrazoDer - 3) % 360;
            else RotBrazoDer = -180;

            break;
        //Extension Brazo Derecho
        case 'p':
            if (RotBrazoDer<50) RotBrazoDer = (RotBrazoDer + 3) % 360;
            else RotBrazoDer = 50;

            break;
        //Flexion AnteBrazo Derecho
        case 'O':
            if(RotAnteBrazoDer>-150) RotAnteBrazoDer = (RotAnteBrazoDer - 3) % 360;
            else RotAnteBrazoDer = -150;

            break;
        //Extension AnteBrazo Derecho
        case 'o':
            if (RotAnteBrazoDer<0) RotAnteBrazoDer = (RotAnteBrazoDer + 3) % 360;
            else RotAnteBrazoDer = 0;

            break;
        //Flexion Brazo Izquierdo
        case 'Q':
            if (RotBrazolzq>-180) RotBrazolzq = (RotBrazolzq - 3) % 360;
            else RotBrazolzq = -180;

            break;
        //Extension Brazo Izquierdo
        case 'q':
            if (RotBrazolzq<50) RotBrazolzq = (RotBrazolzq + 3) % 360;
            else RotBrazolzq = 50;

            break;
        //Flexion AnteBrazo Izquierdo
        case 'W':
            if(RotAnteBrazolzq>-150) RotAnteBrazolzq = (RotAnteBrazolzq - 3) % 360;
            else RotAnteBrazolzq = -150;

            break;
        //Extension AnteBrazo Izquierdo
        case 'w':
            if (RotAnteBrazolzq<0) RotAnteBrazolzq = (RotAnteBrazolzq + 3) % 360;
```

5.5.5 Teclas de dirección

C.....	Rotación de Cabeza del lado izquierdo (Lado positivo eje Y)
c.....	Rotación de Cabeza del lado derecho (Lado negativo eje Y)
H.....	Flexión del torzo
h.....	Extensión del torzo
Q.....	Extensión del brazo izquierdo
q.....	Flexión del brazo izquierdo
P.....	Extensión del brazo derecho
p.....	Flexión del brazo derecho
W.....	Flexión del antebrazo izquierdo
w.....	Extensión del antebrazo izquierdo
O.....	Flexión del antebrazo derecho
o.....	Extensión del antebrazo derecho
E.....	Flexión de la pierna izquierda
e.....	Extensión de la pierna izquierda
I.....	Flexión de la pierna derecha
i.....	Extensión de la pierna derecha
R.....	Flexión de la antepierna izquierda
r.....	Extensión de la antepierna izquierda
U.....	Flexión de la antepierna derecha
u.....	Extensión de la antepierna derecha
A.....	Flexión del pie izquierdo
a.....	Extensión del pie izquierdo
L.....	Flexión del pie derecho
l.....	Extensión del pie derecho
X.....	Rotación positiva de la pélvis en el eje X
x.....	Rotación negativa de la pélvis en el eje X
Y.....	Rotación positiva de la pélvis en el eje Y
y.....	Rotación negativa de la pélvis en el eje Y
Z.....	Rotación positiva de la pélvis en el eje Z
z.....	Rotación negativa de la pélvis en el eje z
+.....	Acercamiento de la escena
-.....	Alejamiento de la escena
Clic Izquierdo.....	Guardar los datos actuales
Clic Derecho.....	Leer los datos guardados
←.....	Rotación negativa de la escena en el eje Y
→.....	Rotación positiva de la escena en el eje Y
↑.....	Rotación negativa de la escena en el eje X
↓.....	Rotación positiva de la escena en el eje X

5.5.6 Ejecución de la escena "Calculo de Centro de Gravedad"

CALCULO DE CENTRO DE GRAVEDAD

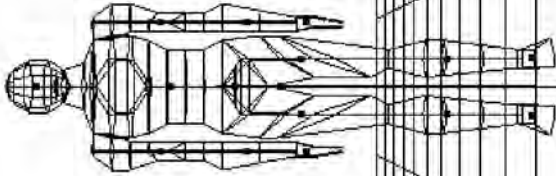
UNAM.FES.ACATLAN.MAC
 Salto Vertical y Parabólico
 Calculo de Centro de Gravedad
 Testista: Aguilar Guillen Ariel

Cabeza: C: 0
 Torzo: H: 0
 Izq Der
 Brazo: O: 0 P: 0
 AntiBraz: W: 0 O: 0
 Pierna: E: 0 I: 0
 AntiPier: R: 0 U: 0
 Pie: A: 0 L: 0
 Pelvis[X]: 0
 Pelvis[Y]: 0
 Pelvis[Z]: 0

CG Principal
 [X]: 0.000000
 [Y]: 0.250222
 [Z]: 0.000000

Z=0m +/- 0

Guardar: Click Izquierdo
 Leer: Click Derecho
 Escen X: UP/DOWN 0
 Escen Y: LEFT/RIGHT 0



CGCabeza	[X]: 0.000000	CGTorzo	[X]: 0.000000
[Y]: 11.030000	[Y]: 3.810000	[Y]: 3.810000	[Y]: 3.810000
[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000
CGBrazIzq	[X]: 3.500000	CGBrazDer	[X]: -3.500000
[Y]: 4.330000	[Y]: 4.330000	[Y]: 4.330000	[Y]: 4.330000
[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000
CGAnteBrazIzq	[X]: 3.500000	CGAnteBrazDer	[X]: -3.500000
[Y]: -0.280000	[Y]: -0.280000	[Y]: -0.280000	[Y]: -0.280000
[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000
CGManoIzq	[X]: 3.500000	CGManoDer	[X]: -3.500000
[Y]: -3.520000	[Y]: -3.520000	[Y]: -3.520000	[Y]: -3.520000
[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000
CGPierIzq	[X]: 1.500000	CGPierDer	[X]: -1.500000
[Y]: -3.270000	[Y]: -3.270000	[Y]: -3.270000	[Y]: -3.270000
[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000
CGAntiPierIzq	[X]: 1.500000	CGAntiPierDer	[X]: -1.500000
[Y]: -1.120000	[Y]: -1.120000	[Y]: -1.120000	[Y]: -1.120000
[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000	[Z]: 0.000000
CGPieIzq	[X]: 1.500000	CGPieDer	[X]: -1.500000
[Y]: 1.500000	[Y]: 1.500000	[Y]: 1.500000	[Y]: 1.500000
[Z]: 1.000000	[Z]: 1.000000	[Z]: 1.000000	[Z]: 1.000000

CALCULO DE CENTRO DE GRAVEDAD

UNAM.FES.ACATLAN.MAC
 Salto Vertical y Parabólico
 Calculo de Centro de Gravedad
 Testista: Aguilan Guillen Ariel

Cabeza: C: 0
 Torzo: H: -18
 Izq Der
 Brazo: Q: -138 P: -128
 AntBraz: W: -63 O: -66
 Pierna: E: 33 I: 57
 AntPier: R: 60 U: 57
 Pie: A: 0 L: 0

Pelvis[X]: 0
 Pelvis[Y]: 0
 Pelvis[Z]: 0

CG Principal
 [X]: 0.000000
 [Y]: 2.293450
 [Z]: -2.582189

CGCabeza
 [X]: 0.000000
 [Y]: 10.490161
 [Z]: -3.408485

CGBrazIzq
 [X]: 3.500000
 [Y]: 10.028869
 [Z]: -1.026296

CGAnteBrazIzq
 [X]: 3.500000
 [Y]: 13.928316
 [Z]: -1.515479

CGManoIzq
 [X]: 3.500000
 [Y]: 16.447251
 [Z]: -3.554500

CGPierIzq
 [X]: 1.500000
 [Y]: -2.742450
 [Z]: -1.780973

CGAntPierIzq
 [X]: 1.500000
 [Y]: -6.548059
 [Z]: -7.472845

CGPieIzq
 [X]: 1.500000
 [Y]: -7.355385
 [Z]: -11.901275

CGTorzo
 [X]: 0.000000
 [Y]: 3.623524
 [Z]: -1.177357

CGBrazDer
 [X]: -3.500000
 [Y]: 9.697517
 [Z]: -0.454369

CGAnteBrazDer
 [X]: -3.500000
 [Y]: 13.557063
 [Z]: -0.224857

CGManoDer
 [X]: -3.500000
 [Y]: 16.362971
 [Z]: -1.844381

CGPierDer
 [X]: -1.500000
 [Y]: -1.780963
 [Z]: -2.742457

CGAntPierDer
 [X]: -1.500000
 [Y]: -3.088065
 [Z]: -8.559631

CGPieDer
 [X]: -1.500000
 [Y]: -2.258625
 [Z]: -13.983962

CENTRO DE GRAVEDAD
 SE HAN GUARDADO VALORES
 ACEPTAR

Guardar: Click Izquierdo
 Legit: Click Derecho
 Escan.X: UP/DOWN -9
 Escan.Y: LEFT/RIGHT -63

5.6 Programación en OpenGL y Visual C++ de Salto Vertical/Parabólico

Para la visualización gráfica del salto parabólico se toma como apoyo el desarrollo que se mostró en el capítulo de tipos de salto. De acuerdo a la fórmula de la parábola del proyectil.

$$y = (\tan \theta_0)x - \left(\frac{g}{2v_0^2 \cos^2 \theta_0} \right) x^2 + a$$

v_0 = Velocidad inicial

θ_0 = Grados de inclinación

g = Constante de gravedad

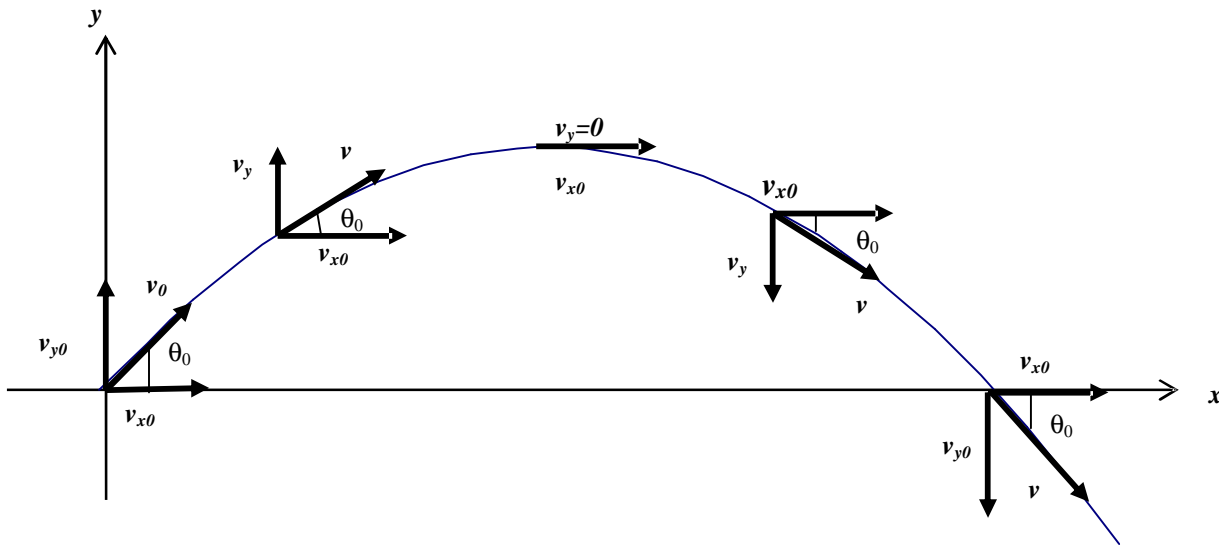
a = Distancia del suelo a la posición del centro de gravedad en el eje vertical.

5.6.1 Codificación del salto vertical/parabólico

Se utiliza en este caso Visual C++ y OpenGL para desarrollar la graficación de un salto parabólico. El código se escribe como sigue:

```
void getValParabol(){
    getParamIni();
    if ((floVel>0)&&(floGrad>0)){
        for (xi=0;yi>=0;xi+=.01){
            yi=tan(theta)*xi-(g/(2*pow(v0,2)*pow(cos(theta),2)))*pow(xi,2)+a;
            if (maxY<yi){
                maxY=yi;
            }
        }
        maxX = xi;
        floIncr=maxX/200;
    }
}
```

En donde	douVel:	La velocidad introducida
	douGrad:	La inclinación introducida
	v0:	Velocidad inicial convertida de km/h a m/s
	theta:	Grados de inclinación convertidos de radianes a grados
	xi:	El vector X
	yi:	El vector Y (sobre el suelo=a)
	g:	Constante de gravedad 9.8
	a:	Posición inicial del centro de gravedad en el eje Y.



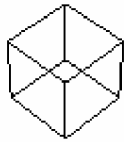
Al diseño ya desarrollado en el capítulo anterior es necesario hacerle unos ajustes en el código de Visual C++ / OpenGL para poder guiar el salto del modelo humano en el resultado de la parábola a partir de lo que sería su centro de gravedad. Por otra parte hay que dibujar la trayectoria y el movimiento del modelo de manera que en el punto mas alto se aproxime a la canasta del tablero de básquetbol que se encuentra colocado desde el extremo de la canasta a 20 puntos del origen.

```

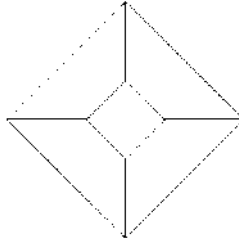
void getParamIni(){
    a = 16; //Posición inicial del CG en el eje Y
    floEscalaMts = 15.972; //1mts = 15.972 puntos gráficos en la escena
    v0 = (floVel/3600*1000)*(floEscalaMts)*(0.249); //Convierte km/hr a mts/seg
    theta = floGrad*3.1416/180; //Convierte radianes a grados
    xx=0;yy=0; //Inicializa los componentes de la parábola
}

void setDibParabol(){
    getParamIni();
    if ((floVel>0)&&(floGrad>0)){
        glBegin(GL_LINE_STRIP);
        for (xx=0;yy>=0;xx+=floIncr){
            yy=tan(theta)*xx-(g/(2*pow(v0,2)*pow(cos(theta),2)))*pow(xx,2)+a;
            glVertex3f(xx+(20-(maxX/2)),yy,0);
        }
        glEnd();
    }
}

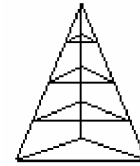
```



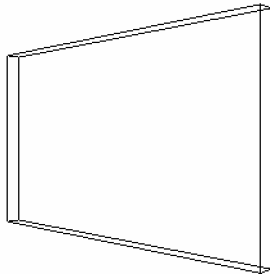
```
glPushMatrix();
    glutWireCube(1);
glPopMatrix();
```



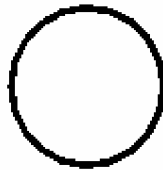
```
glPushMatrix();
    glutWireTorus(1, 2, 2, 4);
glPopMatrix();
```



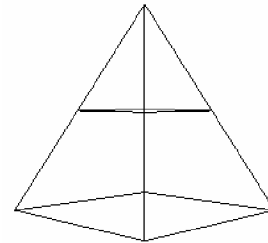
```
glPushMatrix();
    glutWireCone(1,2,3,4);
glPopMatrix();
```



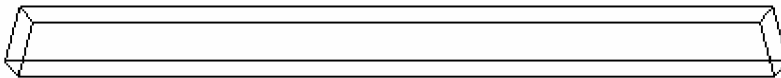
```
glPushMatrix();
    glScalef(1, 12, 20);
    glutWireCube(1);
glPopMatrix();
```



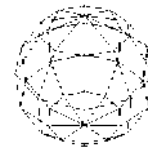
```
glPushMatrix();
    glutWireTorus(.1, 3, 10, 20);
glPopMatrix();
```



```
glPushMatrix();
    glutWireCone(8,12,4,2);
glPopMatrix();
```



```
glPushMatrix();
    glScalef(4,48,2);
    glutWireCube(1);
glPopMatrix();
```



```
glPushMatrix();
    glColor3f(.65,.25,0);
    glScalef(2.5,2.5,2.5);
    glutWireSphere(1,10,5);
glPopMatrix();
```

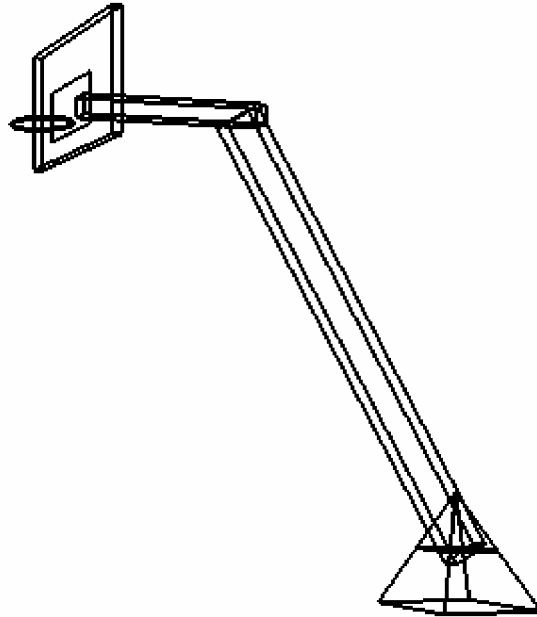
5.6.2 Transformación de objetos predefinidos

Como ya se ha comentado, dentro de OpenGL existen figuras geométricas predefinidas como pueden ser cubos, esferas, aros, poliedros y otros. Usando parte del código del desarrollo anterior sólo se agregan a la escena del salto parabólico un aro, tres cubos y un cono, los que al transformarse por traslación, escalamiento y rotación se pueden alargar, aplastar, cambiar sus radios -como en el caso- del aro para darles la forma y posición deseada.

5.6.3 Codificación del tablero de básquetbol

Una vez hechas las transformaciones a las figuras geométricas predeterminadas, sólo se hacen rotaciones y traslaciones para acomodarlas de la forma en que está hecho un tablero de básquetbol, este tablero es usado como punto de referencia en cuanto a dimensiones al momento de saltar, y tener idea de la magnitud de los resultados en la parábola que se forma.

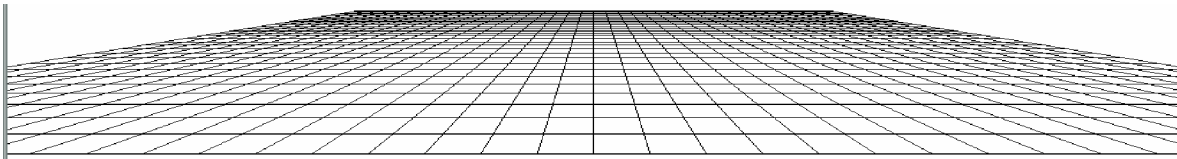
```
void setTablero(){
    glPushMatrix();
        glTranslatef(20,-3,0);
        glPushMatrix();
            glTranslatef(14, 50.75, 0);
            glScalef(20,2,2);
            glColor3f(0,0,0);
            glutWireCube(1);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(34,25, 28, 0);
            glRotatef (30,0,0,1);
            glScalef(4,52,2);
            glColor3f(0,0,0);
            glutWireCube(1);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(46,25, 0, 0);
            glRotatef (45,0,1,0);
            glRotatef (-90,1,0,0);
            glColor3f(0,0,0);
            glutWireCone(8,12,4,2);
        glPopMatrix();
        glPushMatrix();
            glTranslatef(3.5, 52, 0);
            glScalef(1, 12, 20);
            glColor3f(0,0,0);
            glutWireCube(1);
        glPopMatrix();
        glPushMatrix();
            glColor3f(0,0,0);
            glTranslatef(0,48,0);
            glBegin(GL_LINE_LOOP);
            glVertex3f(3,0,-5);
            glVertex3f(3,5,-5);
            glVertex3f(3,5,5);
            glVertex3f(3,0,5);
            glEnd();
        glPopMatrix();
        glPushMatrix();
            glTranslatef(0, 48.75, 0);
            glRotatef(90,1,0,0);
            glColor3f(0,0,0);
            glutWireTorus(.1, 3, 10, 20);
        glPopMatrix();
    glPopMatrix();
}
```



5.6.4 Codificación de la malla

Para dibujar una perspectiva que haga las veces del suelo, se usan los ciclos para unir los puntos horizontal y verticalmente en el plano XZ;

```
void setMalla(void){
    glColor3f(0,5,0);
    glBegin(GL_LINES);
        for (malla=-100;malla<=100;malla=malla+10)
        {
            //MALLA PLANO XZ HORIZONTAL
            glVertex3f(-100,0,malla);
            glVertex3f(100,0,malla);
        }
        for (malla=-100;malla<=100;malla=malla+10)
        {
            //MALLA PLANO XZ VERTICAL
            glVertex3f(malla,0,-100);
            glVertex3f(malla,0,100);
        }
    glEnd();
}
```



5.6.5 Codificación de datos iniciales

El código de la función inicial que pide los datos para graficar se escribe como sigue a continuación, el resto del código como en el caso de los desarrollos anteriores está contenido en el Anexo C:

```
void setDatos(){
    char cad1[5],cad2[2];
    int ch,ch2,j=0,pto=0;
    //PESOS DE LOS SEGMENTOS CORPORALES RELATIVOS AL PESO TOTAL DEL CUERPO [% DE 1]
    douPCAB=0.0694;
    douPTOR=0.4346;
    douPB=0.0271;
    douPAB=0.0162;
    douPM=0.0061;
    douPP=0.1416;
    douPAP=0.0433;
    douPF=0.0137;
    cout<<"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO\n";
    cout<<"FACULTAD DE ESTUDIOS SUPERIORES ACATLAN\n";
    cout<<"MATEMATICAS APLICADAS Y COMPUTACION\n";
    cout<<"CALCULO DE SALTO VERTICAL/PARABOLICO\n";
    cout<<"TESISTA: ARIEL AGUILAR GUILLEN\n\n\n";
    cout<<"VELOCIDAD EN KM/HORA [0.00 - 40.00]: ";
    while ((ch=getche())!='\n'){
        cad1[j]=ch;j++;
        if (j==5) break;
        if (cad1[j]=='.' ) pto++;
        if (pto==2) break;
    }
    cout<<"\nINCLINACION EN GRADOS [0 - 90]: ";j=0;
    while ((ch2=getche())!='\n'){
        cad2[j]=ch2;j++;
        if (j==2) break;
    }
    floVel=atof(cad1);
    floGrad=atof(cad2);
    if ((floVel<0)|| (floVel>40.00)) floVel=0.0;
    if ((floGrad<0)|| (floGrad>90)) floGrad=0;
    if ((floGrad==90)) floGrad=.;//EL LIMITE TIENDE A 90 GRADOS
    getValParabol();
}
}
```

```
C:\ARIEL\TESIS\PROGRAMAS\DEBUG\VERTPARAB18.EXE
UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO
FACULTAD DE ESTUDIOS SUPERIORES ACATLAN
MATEMATICAS APLICADAS Y COMPUTACION
CALCULO DE SALTO VERTICAL/PARABOLICO
TESISTA: ARIEL AGUILAR GUILLEN

VELOCIDAD EN KM/HORA [0.00 - 40.00]: 20.00
INCLINACION EN GRADOS [0 - 90]: 45_
```

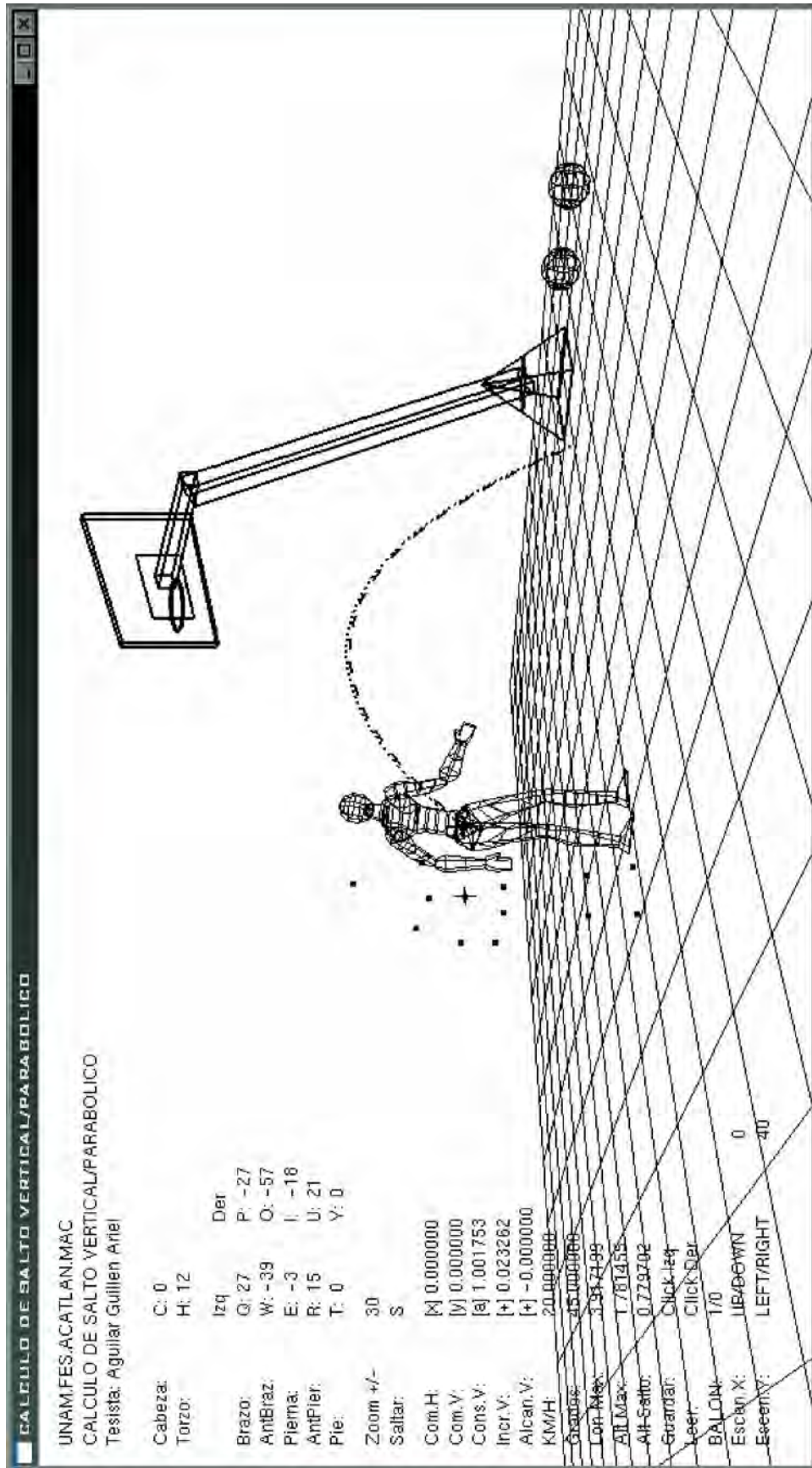
5.6.6 Teclas de dirección

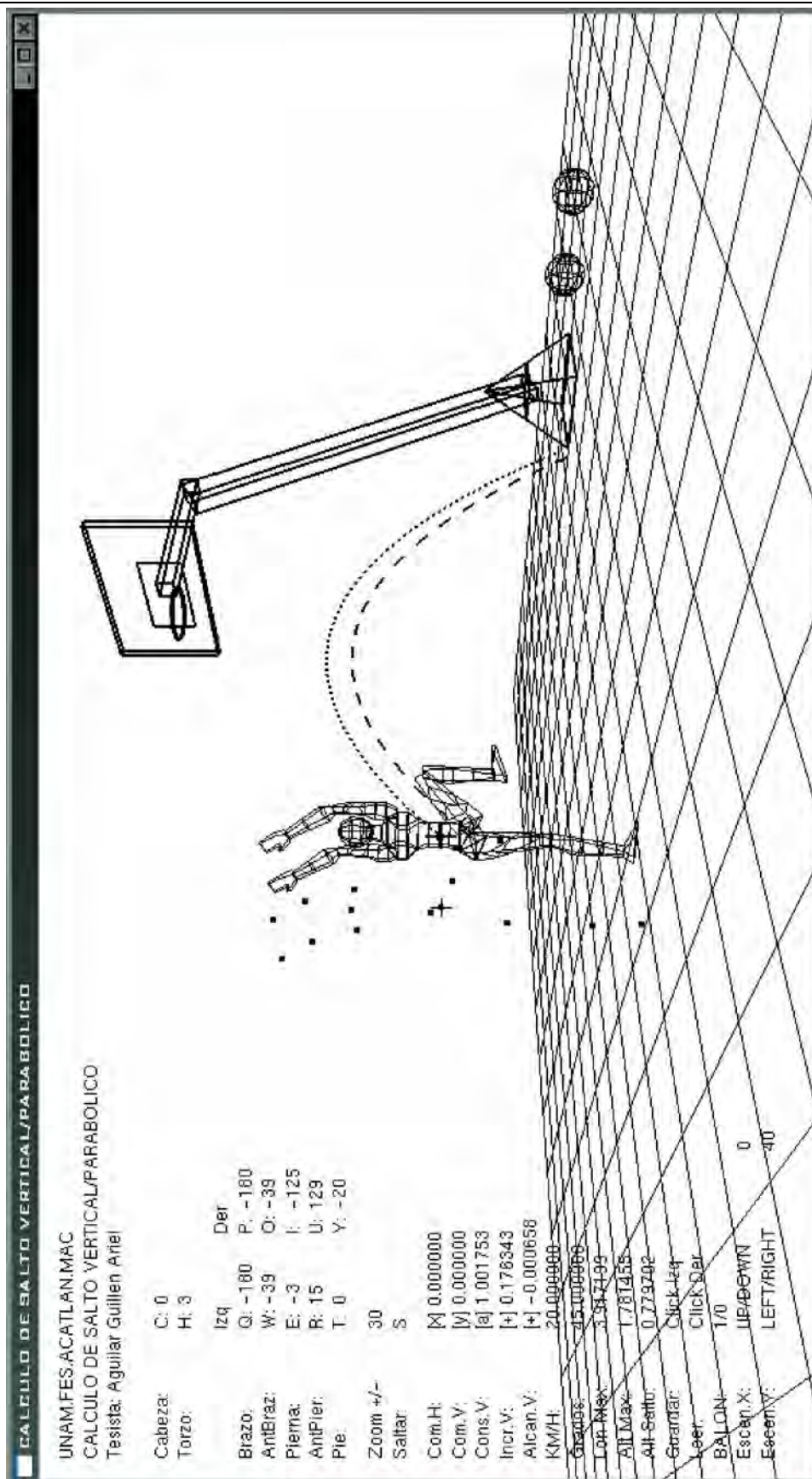
C.....	Rotación de Cabeza del lado izquierdo (Lado positivo eje Y)
c.....	Rotación de Cabeza del lado derecho (Lado negativo eje Y)
H.....	Flexión del torzo
h.....	Extensión del torzo
Q.....	Extensión del brazo izquierdo
q.....	Flexión del brazo izquierdo
P.....	Extensión del brazo derecho
p.....	Flexión del brazo derecho
W.....	Flexión del antebrazo izquierdo
w.....	Extensión del antebrazo izquierdo
O.....	Flexión del antebrazo derecho
o.....	Extensión del antebrazo derecho
E.....	Flexión de la pierna izquierda
e.....	Extensión de la pierna izquierda
l.....	Flexión de la pierna derecha
i.....	Extensión de la pierna derecha
R.....	Flexión de la antepierna izquierda
r.....	Extensión de la antepierna izquierda
U.....	Flexión de la antepierna derecha
u.....	Extensión de la antepierna derecha
T.....	Flexión del pie izquierdo
t.....	Extensión del pie izquierdo
Y.....	Flexión del pie derecho
y.....	Extensión del pie derecho
X.....	Rotación positiva de la pélvis en el eje X
x.....	Rotación negativa de la pélvis en el eje X
Y.....	Rotación positiva de la pélvis en el eje Y
y.....	Rotación negativa de la pélvis en el eje Y
Z.....	Rotación positiva de la pélvis en el eje Z
z.....	Rotación negativa de la pélvis en el eje z
S.....	Saltar hacia el lado positivo X
s.....	Saltar hacia el lado negativo X
+.....	Acercamiento de la escena
-.....	Alejamiento de la escena
Clic Izquierdo.....	Guardar los datos actuales
Clic Derecho.....	Leer los datos guardados
←.....	Rotación negativa de la escena en el eje Y
→.....	Rotación positiva de la escena en el eje Y
↑.....	Rotación negativa de la escena en el eje X
↓.....	Rotación positiva de la escena en el eje X
1.....	Poner balón
0.....	Quitar balón

5.6.7 Lecturas del salto vertical/parabólico

Com.H	Componente de la posición horizontal del CG en la parábola, son los valores que va tomando X a lo largo de la parábola. Los valores van cambiando al recorrer la parábola desde el teclado.
Com.V	Componente de la posición vertical del CG en la parábola, son los valores que va tomando Y a lo largo de la parábola. Los valores van cambiando al recorrer la parábola desde el teclado.
Cons.V	Constante vertical de la parábola, es la distancia del suelo al CG antes de iniciar el despegue, cuando el cuerpo completo está en reposo. Se ha fijado esta distancia.
Incr.V	Cantidad que se incrementa verticalmente la parábola al elevar el CG con cualquiera de los segmentos del modelo antes del despegue
Alcan. V.	Incremento vertical del cuerpo al descender el CG después del despegue. Al no poder descender el CG una vez en el aire con cualquiera de los segmentos, dicho CG reaccionará con la elevación del resto del cuerpo. Dicha distancia, es el alcance vertical de la suspensión.
KM/H	Velocidad en kilómetros por hora introducidas por el usuario.
Grados	Inclinación en grados del proyectil respecto al eje X de la parábola introducidas por el usuario.
Lon Max	Longitud máxima de la parábola en el eje X. Es la distancia similar a la del salto de longitud olímpico. En este caso la distancia es considerada hasta donde el CG se aproxime al suelo.
Alt Max	Altura máxima de la parábola en el eje Y con respecto al suelo. Es la distancia similar a la del salto de altura olímpico, sólo que el CG pasa justo a la altura de la barra horizontal.
Alt Salto	Altura que ha saltado el atleta, (altura máxima) – (constante vertical). Es lo que se denominaría el salto de altura basquetbolístico.

5.6.8 Ejecución de la escena "Calculo de Salto Vertical/Parabólico"





CALCULO DE SALTO VERTICAL/PARABOLICO

UNAM.FES.ACATLAN.MAC
 CALCULO DE SALTO VERTICAL/PARABOLICO
 Testista: Aguilar Guillen Ariel

Cabeza:	C: 0	Izq:	Der:
Torzo:	H: 3	Q: 24	P: -180
Brazo:		W: -33	O: -48
AntiBraz:		E: 6	I: -8
Pierna:		R: 15	U: 30
AntiPier:		T: 45	Y: 45
Pie:			

Zoom +/- 30
 Saltar: S

Com.H: [X] 1.606052
 Com.V: [Y] 0.779004
 Cons.V: [e] 1.001753
 Incr.V: [+] 0.177001
 Alcan.V: [x] 0.121769
 KM/H: 20.000000
 Grados: 45.000000

Alt Max: 3.417415
 Alt Max: 1.781415
 Alt Salto: 0.779004
 Gelerdar: Click Izq
 Lerel: Click Der
 BALON: 1/0
 Escel X: UP/DOWN -8
 Escel V: LEFT/RIGHT -34

SALTO PARABOLICO
 SE HAN GUARDADO VALORES
 ACEPTAR

CALCULO DE SALTO VERTICAL/PARABOLICO

UNAM.FES.ACATLAN.MAC
 CALCULO DE SALTO VERTICAL/PARABOLICO
 Tesisista: Aguillar Guillen Ariel

Cabeza:	C: 0	Izq:	Der:
Torzo:	H: -8	Q: 35	P: -180
Brazo:		W: -30	O: -42
AntiBraz:		E: 18	I: -23
Pierna:		R: 15	U: 18
AntiPier:		T: 45	Y: 45
Pie:			
Zoom +/-:	28		
Saltar:	S		
Com.H:	[X] 0.002673		
Com.V:	[Y] 0.863454		
Cons.V:	[e] 1.001753		
Incr.V:	[+] 0.173155		
Alcan.V:	[+] 0.111737		
KM/H:	15.000000		
Grados:	90.000000		

Left Mouse: 0.000000
 Alt: 0.000000
 Left-Santo: 0.000000
 Guataster: Click Izq
 Leer: Click Der
 BALON: 1/0
 Escen.X: UP/DOWN -7
 Escen.Y: LEFT/RIGHT 4

Resultados

Hay forma de entender el orden que existe en el cuerpo humano para ejecutar un salto basquetbolístico. El cuerpo permite ser entrenado en velocidad y potencia para mejorar el salto vertical y parabólico al entender que el CG puede ser controlado para obtener una mayor altura al momento de saltar, al entender el nivel de dificultad que hay en los grados de inclinación de las articulaciones del cuerpo y mejorar la posición de los brazos de palanca, por otro lado mantener la continuidad en la transmisión de fuerzas de empuje de un segmento a otro y la contracción simultánea entre los músculos que rodean al centro de gravedad.

Se puede calcular la trayectoria del salto parabólico como la trayectoria de un proyectil al comportarse el cuerpo humano como un sólo centro de gravedad, esto a partir de la velocidad de la carrera y los grados de inclinación en el despegue. Algunos atletas en definitiva tienen facultades deportivas ya sea por razón racial en la potencia de sus músculos o morfológica en los brazos de palanca ósea. Factores externos pueden alterar el resultado obtenido de un salto vertical y parabólico como es la región geográfica en donde se lleve a cabo un entrenamiento. La altitud y latitud de la tierra en los entrenamientos deportivos son determinantes en los resultados obtenidos de un salto vertical basquetbolístico debido a la presión atmosférica y la fuerza de gravedad más fuerte cerca de los polos magnéticos y menos en las regiones próximas al ecuador.

Otros factores difieren en la rama varonil y femenil como es la posición del CG, más abajo en las mujeres por encontrarse mayor masa corporal en las caderas y muslos y más arriba en los hombres por encontrarse mayor masa corporal en la espalda y en los hombros.

El cuerpo humano al ser sometido a entrenamientos para saltar no responde de manera exponencial, porque comienza a haber un desgaste al aumentar cada vez más la frecuencia de cada entrenamiento. Por lo tanto la curva de regresión no se ajusta a una curva exponencial. En el caso del ajuste logarítmico, algunos entrenamientos pueden presentar un comportamiento en este último pero generalmente al excederse y llevar una rutina diaria el cuerpo no tiende a responder de manera logarítmica alcanzando una estabilización con dicha rutina. Sin embargo el cuerpo muestra un comportamiento a los entrenamientos más cercanos en su ajuste a la curva cuadrática, puesto que si un entrenamiento es escasamente frecuente, la curva muestra los puntos en el nivel más bajo de la altura de salto vertical, no obstante si se ejecuta el entrenamiento con mayor frecuencia, la curva asciende, finalmente cuando el entrenamiento es excesivamente frecuente, los músculos comienzan a sufrir un deterioro por falta de descanso y sobreentrenamiento, aquí la curva comienza a descender. Por lo tanto la curva mejor ajustada es la cuadrática.

Se pueden analizar gráficamente los resultados de los entrenamientos deportivos por cada rama, además de especificar el músculo trabajado para analizar su regresión y su ajuste. De tal manera que es la observación del propio entrenador la que decide cuáles ejercicios son los más aptos para mejorar el salto en el básquetbol universitario y con qué frecuencia deberán llevarse a cabo al ejercitarse.

En caso de que se llegara a necesitar una muestra más grande que la que ya se llevó a cabo para su análisis, el sistema encuestador/reporteador permite seguir almacenando la información encuestada en su base de datos, para luego volver a generar los gráficos de regresión y ajuste para poder imprimirlos si se desea.

Fué necesario estudiar los 14 segmentos del cuerpo humano y principalmente su centro de gravedad total, puesto que es así como podrá entenderse el cuerpo humano completo, como un sólo punto que se mueve en el espacio y donde actúa la fuerza gravitacional resultante.

En el desarrollo tridimensional efectuado, se puede observar que con cada movimiento de cada segmento del cuerpo, el centro de gravedad cambia de posición. Esto es importante cuando de idear entrenamientos para salto se trata, además de otros que el entrenador llegue a necesitar.

El centro de gravedad es el punto de aplicación de la resultante de todas las fuerzas paralelas que actúan en el sistema humano. En el modelo en 3D se puede observar la posición de dicho centro de gravedad total de todo el cuerpo, además de los centros de cada uno de sus miembros sin ser necesario segmentar cuerpos sin vida o el uso de la báscula triangular y la disposición del atleta para su estudio en todo el análisis. Las posiciones y pesos de los centros de gravedad de cada segmento del modelo fueron tomados en base a pruebas con cuerpos reales, las que al introducirse en este modelo tridimensional nos ayudan a determinar por medio de los cálculos en el desarrollo la posición exacta del CG total del modelo humano en cada movimiento. En el desarrollo de "Salto Vertical/Parabólico" se puede analizar la trayectoria del salto parabólico de un atleta en 3D en base a velocidades y grados de inclinación distintos introducidos por el usuario desde el teclado, mientras que al mismo tiempo se puede observar el efecto de suspensión del resto del cuerpo al rotar los segmentos de este una vez en el aire, puesto que al intentar cambiar la posición del centro de gravedad y no poder alterar su trayectoria parabólica, el CG traslada en el aire al resto del cuerpo elevándolo o descendiendo ligeramente.

El entrenador deportivo puede idear técnicas de salto y entrenamientos a partir de la visualización tridimensional del modelo humano.

Conclusiones

Se debe recordar que una vez que se encuentra un campo de aplicación donde se requiere de una solución práctica, es necesario sumergirse en dicha área. Si se trata de medicina, habrá que aprender algo de medicina para entender y sentir cual es el problema que se nos plantea y dar una buena solución, de la misma manera que si se tratase de economía u otra área. No hay que olvidar que ha habido personalidades de otras áreas como el derecho, que han hecho aportaciones a las matemáticas. Se ha dicho que la matemática es reina de las ciencias, por lo tanto es de esperarse que se hagan muchísimas más aportaciones a las otras áreas del conocimiento, pero siempre respetando la formación adquirida por los profesionales de dichas áreas.

Las diferentes disciplinas vistas en la carrera de matemáticas aplicadas y computación son necesarias para entender cada una de las otras ciencias, la matemática como medio de comunicación y de solución entre ellas y la computación como herramienta tecnológica para manejar y procesar gran cantidad de información. Lo que se usó en este proyecto fue una serie de materias como: Métodos y Técnicas de Investigación, Álgebra Lineal I, Cálculo I, Programación y Lenguajes de Programación, Álgebra Lineal II, Cálculo II, Geometría Analítica, Cálculo III, Estadística, Bases de Datos, Graficación por Computadora, Programación de Sistemas, Seminario de Investigación.

La idea de combinar el área matemática y la computación es de gran ayuda para poder entender también lo que ha pasado a lo largo del tiempo en donde la ciencia y la tecnología han avanzado cada vez más a pasos agigantados. Todo lo que se vio ayuda a mantener la esencia de toda ciencia, para no perder el objetivo principal de cada problema planteado por el mundo moderno. En medicina, economía, tecnología y el espacio hay medios de comunicación y solución en cada una, como pueden ser, matemática del lado de la ciencia y computación del lado tecnológico.

Por otro lado, es de esperarse que exista un amplio campo laboral para dicha área, sin embargo es importante mantener una actualización de conocimientos constante una vez terminada la carrera, puesto que las exigencias son cada vez mayores. Dicho proyecto sólo toca una mínima parte de lo que se podría hacer con todo lo que se ha visto. Toca al egresado encontrar el campo de la aplicación a diversos problemas socio-económicos, administrativos, tecnológicos y científicos.

Bibliografía

- [001] Alvarencia, Beatriz. Máximo, Arturo. Física General con experimentos sencillos 5ta edición. Harla o México.
- [002] Anthony C.P. Thibodeau, G.A. Anatomía y Fisiología. México 1983
- [003] Baraldi B, Armando. Magazine deportivo, La excelencia editorial, edición especial. Atlanta 1996. México 1996.
- [004] Beltrán, Virgilio. Braun, Eliécer. Principios de Física México 1970
- [005] Butler, Neville y otros, Enciclopedia de la vida, El cuerpo vivo, El hombre y la fuerza física, edit. Bruguera tomo II, }, México 1978
- [006] Canavos, George C. Probabilidad y Estadística, Aplicaciones y Métodos. México 1988
- [007] Carlyle, Thomas. Grandes Biografías. Tomo 4. Pág 618.. México 1995
- [008] Castilla Serna, Luis. Cravioto, Joaquin. Estadística Simplificada. México 1991
- [009] Ceffa K, Ana María. A. Dominguez Hector y otros. El mundo de la física. México 1997
- [010] Couper B.Sc, Heather y Henbest, Nigel. M.Sc., Como se viaja al espacio. México 1979
- [011] Chou, Ya-Lun. Análisis Estadístico. México 1977
- [012] Frans, Frederick W. Revista ¡Despertad! 22 de enero 1975, Campeones de salto del mundo de los animales. México 1975
- [013] García, Humberto. Muscle Power. La máquina humana, Franja abdominal, tomo III. Feb 6 México 1988
- [014] Gonzalez Pérez, Alfonso. Visual Basic Programación Cliente Servidor. México 1999
- [015] Gutierrez Dávila, Marcos. Biomecánica deportiva. Bases para el análisis, Madrid 1998
- [016] Haaser, Norman B. LaSalle, Joseph P. y Sullivan, Joseph A. Análisis Matemático 1 Curso de introducción. México 1975
- [017] Henschel, Milton G. Revista ¡Despertad! 8 de abril 2000. Un marsupial con resortes en los pies. México 2000
- [018] Hewitt, Paul G. Física Conceptual 3a Edición. México 1999
- [019] Hochmuth, Gerhard. Biomecánica de los movimientos deportivos. Madrid 1973
- [020] Hoel, Paul G. Estadística Elemental. México 1979
- [021] Iriarte V. Valderrama, Rafael. Métodos Numéricos. México 1999
- [022] Jiménez Redondo, Manuel. Física Madrid 1976
- [023] López Navarro, Joan. Larousse Diccionario Enciclopédico 2000 Colombia 1999

- [024] Luthe, Rodolfo. Olivera, Antonio. Schutz, Fernando. Métodos Numéricos. México 1980
- [025] Moston, Muska. Gimnasia Dinámica. México 1968
- [026] Myers, Walpole. Probabilidad y Estadística 4ta edición. México 1992
- [027] Ortega, Aniceto y E. Fisiología del ejercicio. México 1948
- [028] Peck, George. Seagate Crystal Reports 8. Madrid 2002
- [029] Serway, Raymond A. Madison, James University Física I. México 1996
- [030] Swokowski, Earl W. Cálculo con Geometría Analítica. Estados Unidos de América 1988.
- [031] Tavard, Anne. Demay, Francois. Enciclopedia Científica 1. México 1996
- [032] Wirz, Daniel. Enciclopedia Mundial del deporte tomo 1, abdominales. México 1982.
- [033] Woo, Mason. Neider, Jackie. Davis, Tom. Shreiner, Dave . OpenGL Programming Guide Third Edition. The official Guide to Learning OpenGL, Version 1.2. United States Of America 2002
- [034] Yamane, Taro. Cortado de Kohan, Nuria. Estadística. México 1974
- [035] <http://verticaljump.com/html/trial.html>
- [036] <http://www.astromia.com/glosario/parabola.htm>
- [037] <http://www.csd.mec.es/CSD/Deporte/MedicinaDeportiva/Biomecanica/introduccion.htm>
- [038] <http://www.enterate.unam.mx/Articulos/2005/mayo/biomecanica.htm>
- [039] <http://www.enterate.unam.mx/Articulos/2005/mayo/biomecanica.htm>
- [040] <http://www.harunyahya.com/es/naturaleza08.php>, La Pulga : un Diseño Ideal para Saltos Elevados

Anexo A. Código Visual Basic Código Fuente del Desarrollo Encuestador Reporteador

FrmEncuesta.frm

Formulario de llenado de encuestas

```
'
' Universidad Nacional autonoma de México
' Tesista      Ariel Aguilar Guillen

' Proyecto:    Sistema Encuestador y Reporteador
' Fecha:       Octubre 2005
'
' Nombre:      frmEncuesta.frm
' Versión:     2.0 Octubre 2005
' Lenguaje:    Visual Basic 6.0
' Descripción: Formulario de captura de encuestas y
               reporteador estadístico de regresiones
' Precondición: Debe de existir una base de datos en Access con las tablas requeridas
' Base de datos: SALTO
' Tabla(s):    SALCESCUELA  Tabla catálogo de escuelas
               SALMENCUESTA  Tabla maestra de encuestas realizadas
'
' Actualización: A.A.G Octubre 2005
'
```

Public Con As New ADODB.Connection

Public Rs As ADODB.Recordset

Public Sql As String

Public Drv As String

Public BD As String

Public intDecPeso, intDecEst, intDecCuan As Integer

Public intSex, intRaz, intCom, intEda As Integer

Public douPes, douEst As Double

Public intMej, intAro As Integer

Public douCua As Double

Public intCon, intCal, intDel, intGem, intTab, intAbdCor, intAbdCom, intTra, intDor, intEsc, intPol, intEscPol, intGlu As Integer

Public intCua, intBic, intArr, intVal, intPun, intEla, intIso As Integer

Public Function getValidar() As Boolean

 If txtPes.Text <> "" Then

 If txtEst.Text <> "" Then

 If txtCuan.Text <> "" Then

 getValidar = True

 Else

 MsgBox (" ¡Falta Medición de Salto! ")

 End If

 Else

 MsgBox (" ¡Falta Medición de Estatura! ")

 End If

 Else

 MsgBox (" ¡Falta Medición de Peso! ")

 End If

End Function

Public Function setFolio()

 Drv = "DRIVER={Microsoft Access Driver (*.mdb)}; "

 BD = "DBQ=C:\Ariel\TESIS\BaseDatos\SALTO.mdb; "

 Sql = "SELECT MAX(PKIN_IDENCUEST)+1 FROM SALMENCUESTA "

 Con.ConnectionString = Drv + BD

 Con.Open

 Set Rs = Con.Execute(Sql)

 txtFolio.Text = (Rs.Fields(0).Value)

 Rs.Close

 Con.Close

End Function

```
Public Function setInicializa()  
    intDecPeso = 0  
    intDecEst = 0  
    intDecCuan = 0  
    txtPes.Text = Clear  
    txtEst.Text = Clear  
    txtCuan.Text = Clear  
    optDel1 = True  
    optGem1 = True  
    optTab1 = True  
    optAbdCor1 = True  
    optAbdCom1 = True  
    optTra1 = True  
    optDor1 = True  
    optEsc1 = True  
    optPol1 = True  
    optEscPol1 = True  
    optGlu1 = True  
    optCua1 = True  
    optBic1 = True  
    optArr1 = True  
    optVal1 = True  
    optPun1 = True  
    optEla1 = True  
    optIso1 = True  
End Function  
  
Private Sub setValidar()  
    If optSex1 = True Then intSex = 1  
    If optSex0 = True Then intSex = 0  
  
    If optRaz3 = True Then intRaz = 3  
    If optRaz2 = True Then intRaz = 2  
    If optRaz1 = True Then intRaz = 1  
  
    If optCom3 = True Then intCom = 2  
    If optCom2 = True Then intCom = 3  
    If optCom1 = True Then intCom = 1  
  
    If optEda3 = True Then intEda = 3  
    If optEda2 = True Then intEda = 2  
    If optEda1 = True Then intEda = 1  
  
    douPes = txtPes.Text  
    douEst = txtEst.Text  
  
    If optMej1 = True Then intMej = 1  
    If optMej0 = True Then intMej = 0  
  
    If optAro1 = True Then intAro = 1  
    If optAro0 = True Then intAro = 0  
  
    douCua = txtCuan.Text  
  
    If optCon3 = True Then intCon = 3  
    If optCon2 = True Then intCon = 2  
    If optCon1 = True Then intCon = 1  
  
    If optCal3 = True Then intCal = 3  
    If optCal2 = True Then intCal = 1  
    If optCal1 = True Then intCal = 2  
  
    If optDel4 = True Then intDel = 4  
    If optDel3 = True Then intDel = 3  
    If optDel2 = True Then intDel = 2  
    If optDel1 = True Then intDel = 1  
  
    If optGem4 = True Then intGem = 4  
    If optGem3 = True Then intGem = 3  
    If optGem2 = True Then intGem = 2  
    If optGem1 = True Then intGem = 1
```

If optTab4 = True Then intTab = 4
If optTab3 = True Then intTab = 3
If optTab2 = True Then intTab = 2
If optTab1 = True Then intTab = 1

If optAbdCor4 = True Then intAbdCor = 4
If optAbdCor3 = True Then intAbdCor = 3
If optAbdCor2 = True Then intAbdCor = 2
If optAbdCor1 = True Then intAbdCor = 1

If optAbdCom4 = True Then intAbdCom = 4
If optAbdCom3 = True Then intAbdCom = 3
If optAbdCom2 = True Then intAbdCom = 2
If optAbdCom1 = True Then intAbdCom = 1

If optTra4 = True Then intTra = 4
If optTra3 = True Then intTra = 3
If optTra2 = True Then intTra = 2
If optTra1 = True Then intTra = 1

If optDor4 = True Then intDor = 4
If optDor3 = True Then intDor = 3
If optDor2 = True Then intDor = 2
If optDor1 = True Then intDor = 1

If optEsc4 = True Then intEsc = 4
If optEsc3 = True Then intEsc = 3
If optEsc2 = True Then intEsc = 2
If optEsc1 = True Then intEsc = 1

If optPol4 = True Then intPol = 4
If optPol3 = True Then intPol = 3
If optPol2 = True Then intPol = 2
If optPol1 = True Then intPol = 1

If optEscPol4 = True Then intEscPol = 4
If optEscPol3 = True Then intEscPol = 3
If optEscPol2 = True Then intEscPol = 2
If optEscPol1 = True Then intEscPol = 1

If optGlu4 = True Then intGlu = 4
If optGlu3 = True Then intGlu = 3
If optGlu2 = True Then intGlu = 2
If optGlu1 = True Then intGlu = 1

If optCua4 = True Then intCua = 4
If optCua3 = True Then intCua = 3
If optCua2 = True Then intCua = 2
If optCua1 = True Then intCua = 1

If optBic4 = True Then intBic = 4
If optBic3 = True Then intBic = 3
If optBic2 = True Then intBic = 2
If optBic1 = True Then intBic = 1

If optArr4 = True Then intArr = 4
If optArr3 = True Then intArr = 3
If optArr2 = True Then intArr = 2
If optArr1 = True Then intArr = 1

If optVal4 = True Then intVal = 4
If optVal3 = True Then intVal = 3
If optVal2 = True Then intVal = 2
If optVal1 = True Then intVal = 1

If optPun4 = True Then intPun = 4
If optPun3 = True Then intPun = 3
If optPun2 = True Then intPun = 2
If optPun1 = True Then intPun = 1

A[4] Sal to vertical y parabólico

```
If optEla4 = True Then intEla = 4
If optEla3 = True Then intEla = 3
If optEla2 = True Then intEla = 2
If optEla1 = True Then intEla = 1

If optIso4 = True Then intIso = 4
If optIso3 = True Then intIso = 3
If optIso2 = True Then intIso = 2
If optIso1 = True Then intIso = 1
End Sub

Private Sub btnEliminar_Click()
    End
End Sub

Private Sub btnGuardar_Click()
    If getValidar() = True Then
        Call setValidar
        Sql = ""
        Drv = "DRIVER={Microsoft Access Driver (*.mdb)};"
        BD = "DBQ=C:\Ariel\TESIS\BaseDatos\SALTO.mdb;"
        Sql = Sql + "INSERT INTO SALMENCUESTA "
        Sql = Sql + "("
        Sql = Sql + "FKIN_IDESCUELA, "
        Sql = Sql + "NNIN_SEXO, "
        Sql = Sql + "NNIN_RAZA, "
        Sql = Sql + "NNIN_COMPLEX, "
        Sql = Sql + "NNIN_EDAD, "
        Sql = Sql + "NNNU_PESO, "
        Sql = Sql + "NNNU_ESTAT, "
        Sql = Sql + "NNIN_MEJORAM, "
        Sql = Sql + "NNIN_ALCANCE, "
        Sql = Sql + "NNNU_SALTOAPROX, "
        Sql = Sql + "NNIN_SALTADOR, "
        Sql = Sql + "NNIN_CALZADO, "
        Sql = Sql + "NNIN_DELTOIDES, "
        Sql = Sql + "NNIN_GEMELOS, "
        Sql = Sql + "NNIN_TABLERO, "
        Sql = Sql + "NNIN_ABDMCORT, "
        Sql = Sql + "NNIN_ABDMCOMP, "
        Sql = Sql + "NNIN_TRAPECIO, "
        Sql = Sql + "NNIN_DORSAL, "
        Sql = Sql + "NNIN_ESCALERAS, "
        Sql = Sql + "NNIN_POLAINAS, "
        Sql = Sql + "NNIN_ESCPOL, "
        Sql = Sql + "NNIN_GLUTEOS, "
        Sql = Sql + "NNIN_CUADRICEPS, "
        Sql = Sql + "NNIN_BICEPS, "
        Sql = Sql + "NNIN_VELOCIDAD, "
        Sql = Sql + "NNIN_VALLAS, "
        Sql = Sql + "NNIN_PUNTAS, "
        Sql = Sql + "NNIN_ELASTICIDAD, "
        Sql = Sql + "NNIN_ISOMETRICOS"
        Sql = Sql + ")"
        Sql = Sql + "VALUES "
        Sql = Sql + "("
        Sql = Sql & cmbEscuela.ListIndex & ", "
        Sql = Sql & intSex & ", "
        Sql = Sql & intRaz & ", "
        Sql = Sql & intCom & ", "
        Sql = Sql & intEda & ", "
        Sql = Sql & douPes & ", "
        Sql = Sql & douEst & ", "
        Sql = Sql & intMej & ", "
        Sql = Sql & intAro & ", "
        Sql = Sql & douCua & ", "
        Sql = Sql & intCon & ", "
        Sql = Sql & intCal & ", "
        Sql = Sql & intDel & ", "
        Sql = Sql & intGem & ", "
```

```

    Sql = Sql & intTab & ","
    Sql = Sql & intAbdCor & ","
    Sql = Sql & intAbdCom & ","
    Sql = Sql & intTra & ","
    Sql = Sql & intDor & ","
    Sql = Sql & intEsc & ","
    Sql = Sql & intPol & ","
    Sql = Sql & intEscPol & ","
    Sql = Sql & intGlu & ","
    Sql = Sql & intCua & ","
    Sql = Sql & intBic & ","
    Sql = Sql & intArr & ","
    Sql = Sql & intVal & ","
    Sql = Sql & intPun & ","
    Sql = Sql & intEla & ","
    Sql = Sql & intIso & " "
    Sql = Sql & ")"
    If Rs.State = 1 Then Rs.Close
    Con.ConnectionString = Drv + BD
    Con.Open
    Set Rs = Con.Execute(Sql)
    If Rs.State = 1 Then Rs.Close
    Con.Close
    MsgBox ("¡SE HA ALMACENADO EL ESTADISTICO!")
    Call setFolio
    Call setInicializa
End If
End Sub

Private Sub btnReport_Click()
    frmGraficos.Show
End Sub

Private Sub cmbEscuela_KeyPress(KeyAscii As Integer)
    KeyAscii = 0
End Sub

Private Sub Command1_Click()
    frmRegresAjuste.Show
End Sub

Private Sub Form_Load()
    setCenter Me
    Drv = "DRIVER={Microsoft Access Driver (*.mdb)};"
    BD = "DBQ=C:\Ariel\TESIS\BaseDatos\SALTO.mdb;"
    Sql = "SELECT * FROM SALCESCUOLA "
    Con.ConnectionString = Drv + BD
    Con.Open
    Set Rs = Con.Execute(Sql)
    While Not Rs.EOF
        cmbEscuela.AddItem (Rs.Fields(1).Value)
        Rs.MoveNext
    Wend
    Rs.Close
    Con.Close
    cmbEscuela.ListIndex = 0
    Call setFolio
End Sub

Private Sub txtCuan_KeyPress(KeyAscii As Integer)
    If KeyAscii < 48 Then
        If KeyAscii = 46 Then
            intDecCuan = intDecCuan + 1
            If intDecCuan > 1 Then
                KeyAscii = 0
            End If
        Else: KeyAscii = 0
        End If
    ElseIf KeyAscii > 57 Then
        KeyAscii = 0
    End If
End Sub

```

```
Private Sub txtEst_KeyPress(KeyAscii As Integer)
    If KeyAscii < 48 Then
        If KeyAscii = 46 Then
            intDecEst = intDecEst + 1
            If intDecEst > 1 Then
                KeyAscii = 0
            End If
        Else: KeyAscii = 0
        End If
    ElseIf KeyAscii > 57 Then
        KeyAscii = 0
    End If
End Sub
```

```
Private Sub txtPes_KeyPress(KeyAscii As Integer)
    If KeyAscii < 48 Then
        If KeyAscii = 46 Then
            intDecPeso = intDecPeso + 1
            If intDecPeso > 1 Then
                KeyAscii = 0
            End If
        Else: KeyAscii = 0
        End If
    ElseIf KeyAscii > 57 Then
        KeyAscii = 0
    End If
End Sub
```

FrmGráficos.frm

Formulario de selección de gráfica

```
,
' Universidad Nacional autonoma de México
' Tesista    Ariel Aguilar Guillen
' Proyecto:  Sistema Encuestador y Reporteador
' Fecha:     Octubre 2005
',
' Nombre:    frmGraficos.frm
' Versión:   2.0 Octubre 2005
' Lenguaje:  Visual Basic 6.0
' Descripción: Formulario de captura de encuestas y
'              reporteador estadístico de regresiones
' Precondición: Debe de existir una base de datos en Access con las tablas requeridas
' Base de datos: SALTO
' Tabla(s):   SALCESCUELA  Tabla catálogo de escuelas
'             SALMENCUESTA  Tabla maestra de encuestas realizadas
',
' Actualización: A.A.G Octubre 2005
,
```

```
Private Sub btnCancelar_Click()
    End
End Sub
```

```
Private Sub btnImprimir_Click()
    frmRegresDisper.Show
End Sub
```

```
Private Sub btnCerrar_Click()
    Unload Me
End Sub
```

```
Private Sub btnGraficar_Click()
    If optDisReg.Value = True Then
        frmRegresDisperLin.Show
    ElseIf optAjuExp.Value = True Then
        frmRegresAjusteExp.Show
    ElseIf optAjuLog.Value = True Then
```

```
    frmRegresAjusteLog.Show
    Elseif optAjuSqr.Value = True Then
        frmRegresAjusteSqr.Show
    End If
End Sub
```

```
Private Sub Form_Load()
    setCenter Me
End Sub
```

FrmRegresAjusteExp.frm
Formulario del visor Crystal Reports para la gráfica de ajuste exponencial.

```
'
' Universidad Nacional autonoma de México
' Tesista    Ariel Aguilar Guillen

' Proyecto:   Sistema Encuestador y Reporteador
' Fecha:      Octubre 2005
'
' Nombre:     frmRegresAjusteExp.rpt
' Versión:    2.0 Octubre 2005
' Lenguaje:   Visual Basic 6.0
' Descripción: Reporteador Estadístico de Dispersión
' Precondición: Debe ser llamado desde el formulario frmEncuesta
' Archivo Crystal rptRegresAjusteExp.rpt
' Base de datos: SALTO
' Tabla(s):   SALCESCUELA  Tabla catálogo de escuelas
'             SALMENCUESTA  Tabla maestra de encuestas realizadas
'
' Actualización: A.A.G Octubre 2005
```

Dim Report As New rptRegresAjusteExp

```
Private Sub Form_Load()
    setMaxim Me
    setCenter Me
    Screen.MousePointer = vbHourglass
    CRViewer91.ReportSource = Report
    CRViewer91.ViewReport
    Screen.MousePointer = vbDefault
End Sub
```

```
Private Sub Form_Resize()
    CRViewer91.Top = 0
    CRViewer91.Left = 0
    CRViewer91.Height = ScaleHeight
    CRViewer91.Width = ScaleWidth
End Sub
```

FrmRegresAjusteLog.frm
Formulario del visor Crystal Reports para la gráfica de ajuste logarítmica.

```
'
' Universidad Nacional autonoma de México
' Tesista    Ariel Aguilar Guillen

' Proyecto:   Sistema Encuestador y Reporteador
' Fecha:      Octubre 2005
'
' Nombre:     frmRegresAjusteLog.frm
' Versión:    2.0 Octubre 2005
' Lenguaje:   Visual Basic 6.0
' Descripción: Reporteador Estadístico de Ajuste Logaritmico
' Precondición: Debe ser llamado desde el formulario frmGraficos
' Archivo Crystal rptRegresAjusteLog.frm
```


A[8] Sal to vertical y parabólico

```
' Base de datos: SALTO
' Tabla(s): SALCESCUELA Tabla catálogo de escuelas
' SALMENCUESTA Tabla maestra de encuestas realizadas
'
' Actualización: A.A.G Octubre 2005
```

Dim Report As New rptRegresAjusteLog

```
Private Sub Form_Load()
    setMaxim Me
    setCenter Me
    Screen.MousePointer = vbHourglass
    CRViewer91.ReportSource = Report
    CRViewer91.ViewReport
    Screen.MousePointer = vbDefault
End Sub
```

```
Private Sub Form_Resize()
    CRViewer91.Top = 0
    CRViewer91.Left = 0
    CRViewer91.Height = ScaleHeight
    CRViewer91.Width = ScaleWidth
End Sub
```

FrmRegresAjusteSqr.frm
Formulario del visor Crystal Reports para la gráfica
de ajuste cuadrática

```
'
' Universidad Nacional autonoma de México
' Tesista Ariel Aguilar Guillen
'
' Proyecto: Sistema Encuestador y Reporteador
' Fecha: Octubre 2005
'
' Nombre: frmRegresAjusteSqr.frm
' Versión: 2.0 Octubre 2005
' Lenguaje: Visual Basic 6.0
' Descripción: Reporteador Estadístico de Ajuste Cuadrático
' Precondición: Debe ser llamado desde el formulario frmGráficos
' Archivo Crystal rptRegresAjusteSqr.rpt
' Base de datos: SALTO
' Tabla(s): SALCESCUELA Tabla catálogo de escuelas
' SALMENCUESTA Tabla maestra de encuestas realizadas
'
' Actualización: A.A.G Octubre 2005
```

Dim Report As New rptRegresAjusteSqr

```
Private Sub Form_Load()
    setMaxim Me
    setCenter Me
    Screen.MousePointer = vbHourglass
    CRViewer91.ReportSource = Report
    CRViewer91.ViewReport
    Screen.MousePointer = vbDefault
End Sub
```

```
Private Sub Form_Resize()
    CRViewer91.Top = 0
    CRViewer91.Left = 0
    CRViewer91.Height = ScaleHeight
    CRViewer91.Width = ScaleWidth
End Sub
```

frmRegresDisperLin.frm

Formulario del visor Crystal Reports para la gráfica de dispersión y regresión lineal.

```
'  
' Universidad Nacional autonoma de México  
' Tesista      Ariel Aguilar Guillen  
'  
' Proyecto:    Sistema Encuestador y Reporteador  
' Fecha:      Octubre 2005  
'  
' Nombre:     frmRegresDisperLin.frm  
' Versión:    2.0 Octubre 2005  
' Lenguaje:   Visual Basic 6.0  
' Descripción: Reporteador Estadístico de Dispersión y regresión  
' Precondición: Debe ser llamado desde el formulario frmGraficos  
' Archivo Crystal rptRegresDisperLin.rpt  
' Base de datos: SALTO  
' Tabla(s):   SALCESCUELA  Tabla catálogo de escuelas  
'            SALMENCUESTA  Tabla maestra de encuestas realizadas  
'  
' Actualización: A.A.G Octubre 2005  
'
```

Dim Report As New rptRegresDisperLin

```
Private Sub Form_Load()  
    setMaxim Me  
    setCenter Me  
    Screen.MousePointer = vbHourglass  
    CRViewer91.ReportSource = Report  
    CRViewer91.ViewReport  
    Screen.MousePointer = vbDefault  
End Sub
```

```
Private Sub Form_Resize()  
    CRViewer91.Top = 0  
    CRViewer91.Left = 0  
    CRViewer91.Height = ScaleHeight  
    CRViewer91.Width = ScaleWidth  
End Sub
```

Anexo B. Código Visual C++/OpenGL Código Fuente del Desarrollo "Cálculo de Centro de Gravedad"

```

/*
Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Acatlán
Salto Vertical y Parabólico Basquetbolístico
Cálculo de Centro de Gravedad
Tesisista: Aguilar Guillen Ariel
Programa para calcular el Centro de Gravedad
Fecha de creación: Octubre 2003
Fecha de termino: Mayo 2006
*/

#include <GL/glut.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <math.h>
#include <windows.h>
#include <conio.h>
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cstdlib>
#include <cstring>
using namespace std;

#define MAX 4

int intXPantalla = GetSystemMetrics(SM_CXFULLSCREEN);
int intYPantalla = GetSystemMetrics(SM_CYFULLSCREEN);

static char opc;
int intGrad,i,j,k,n,intXLeyendas,intCoordX,intCoordY,intCoordZ,intVZoom;

static int RotCabeza = 0, RotTorzo = 0;
static int RotBrazoDer = 0, RotAnteBrazoDer = 0, RotPiernaDer = 0, RotAntePiernaDer = 0, RotPieDer = 0;
static int RotBrazolzq = 0, RotAnteBrazolzq = 0, RotPiernalzq = 0, RotAntePiernalzq = 0, RotPielzq = 0;

static int RotPelvisX = 0;
static int RotPelvisY = 0;
static int RotPelvisZ = 0;
static int RotEscenaX = 0;
static int RotEscenaY = 0;

static char etiqueta[100];
int lin= 20;
int col=40;
int col2=22;
int col3=32;
int malla;

double Rx[MAX][MAX], Ry[MAX][MAX], Rz[MAX][MAX], RxArtic[MAX][MAX], RxSubArtic[MAX][MAX], RxSubSubArtic[MAX][MAX];
double A[MAX][MAX], B[MAX][MAX], C[MAX][MAX], D[MAX][MAX], E[MAX][MAX], P[MAX][MAX], CGTOT[MAX][MAX];
double CGCAB[MAX][MAX], CGTOR[MAX][MAX];
double CGBD[MAX][MAX], CGBI[MAX][MAX], CGABD[MAX][MAX], CGABI[MAX][MAX], CGMD[MAX][MAX], CGMI[MAX][MAX];
double CGPD[MAX][MAX], CGPI[MAX][MAX], CGAPD[MAX][MAX], CGAPI[MAX][MAX], CGFD[MAX][MAX], CGFI[MAX][MAX];
double douPCAB, douPTOR, douPB, douPAB, douPM, douPP, douPAP, douPF;
double douVCAB, douVTOR, douVB, douVAB, douVM, douVP, douVAP, douVF;

int test;

////////////////////////////////////
// Clase registro.
class Registro {
public: Registro(){}
void Guardar();
void Leer();
private: int dato[20];
};

// GUARDA VALORES ACTUALES
void Registro::Guardar() {
dato[0]=RotCabeza;
dato[1]=RotTorzo;
dato[2]=RotBrazoDer;
dato[3]=RotBrazolzq;
dato[4]=RotAnteBrazoDer;
dato[5]=RotAnteBrazolzq;
dato[6]=RotPiernaDer;

```

B[2] Sal to vertical y parabólico

```
    dato[7]=RotPiernalzq;
    dato[8]=RotAntePiernaDer;
    dato[9]=RotAntePiernalzq;
    dato[10]=RotPieDer;
    dato[11]=RotPielzq;
    dato[12]=intVZoom;
    dato[13]=RotPelvisX;
    dato[14]=RotPelvisY;
    dato[15]=RotPelvisZ;
    dato[16]=RotEscenaX;
    dato[17]=RotEscenaY;
}

// LEE VALORES GUARDADOS
void Registro::Leer()
{
    RotCabeza          = dato[0];
    RotTorzo           = dato[1];
    RotBrazoDer        = dato[2];
    RotBrazoIzq        = dato[3];
    RotAnteBrazoDer= dato[4];
    RotAnteBrazoIzq= dato[5];
    RotPiernaDer       = dato[6];
    RotPiernalzq       = dato[7];
    RotAntePiernaDer=dato[8];
    RotAntePiernalzq=dato[9];
    RotPieDer          = dato[10];
    RotPielzq          = dato[11];
    intVZoom           = dato[12];
    RotPelvisX         = dato[13];
    RotPelvisY         = dato[14];
    RotPelvisZ         = dato[15];
    RotEscenaX         = dato[16];
    RotEscenaY         = dato[17];
}

// CLASE DATOS, ALMACENA Y TRATA LOS DATOS
class Datos :public fstream {
public:
    Datos() : fstream("CGravedad.dat", ios::in | ios::out | ios::binary) {
        if(!good()) {
            open("CGravedad.dat", ios::in | ios::out | ios::trunc | ios::binary);
        }
    }
    void Guardar(Registro &reg);
    bool Recupera(long n, Registro &reg);
};

void Datos::Guardar(Registro &reg) {
    clear();
    seekg(0, ios::beg);
    write(reinterpret_cast<char *> (&reg), sizeof(Registro));
}

bool Datos::Recupera(long n, Registro &reg) {
    clear();
    seekg(n*sizeof(Registro), ios::beg);
    read(reinterpret_cast<char *> (&reg), sizeof(Registro));
    return gcount() > 0;
}

//////////FUNCION DE ETIQUETAS//////////
void inline dibujarCadena (char *s)
{
    unsigned int i;
    for (i = 0; i < strlen (s); i++)
        glutBitmapCharacter (GLUT_BITMAP_HELVETICA_12, s[i]);
}

void init(void)
{
    douPCAB=0.0694;
    douPTOR=0.4346;
    douPB=0.0271;
    douPAB=0.0162;
    douPM=0.0061;
    douPP=0.1416;
    douPAP=0.0433;
    douPF=0.0137;
}

double GetWindowHeight(void);
```

```

void setLeyendas (void)
{
    glColor3f(1,1,1);
    printf (etiqueta, "UNAM.FES.ACATLAN.MAC");glRasterPos2f (col, lin+2);dibujarCadena (etiqueta);
    printf (etiqueta, "Salto Vertical y Parabolico");glRasterPos2f (col, lin);dibujarCadena (etiqueta);
    printf (etiqueta, "Calculo de Centro de Gravedad");glRasterPos2f (col, lin-2);dibujarCadena (etiqueta);
    printf (etiqueta, "Tesista: Aguilar Guillen Ariel");glRasterPos2f (col, lin-4);dibujarCadena (etiqueta);

    glColor3f(0,1,0);

    printf (etiqueta, "Cabeza.");glRasterPos2f (col, lin-6);dibujarCadena (etiqueta);
    printf (etiqueta, "C.");glRasterPos2f (col+7, lin-6);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotCabeza);glRasterPos2f (col+9, lin-6);dibujarCadena (etiqueta);

    printf (etiqueta, "Torzo.");glRasterPos2f (col, lin-7.5);dibujarCadena (etiqueta);
    printf (etiqueta, "H.");glRasterPos2f (col+7,lin-7.5);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotTorzo);glRasterPos2f (col+9,lin-7.5);dibujarCadena (etiqueta);

    printf (etiqueta, "Izq");glRasterPos2f (col+7, lin-10);dibujarCadena (etiqueta);
    printf (etiqueta, "Der");glRasterPos2f (col+12, lin-10);dibujarCadena (etiqueta);
    printf (etiqueta, "Brazo.");glRasterPos2f (col, lin-12);dibujarCadena (etiqueta);
    printf (etiqueta, "Q.");glRasterPos2f (col+7, lin-12);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotBrazolzq);glRasterPos2f (col+9, lin-12);dibujarCadena (etiqueta);
    printf (etiqueta, "P.");glRasterPos2f (col+12, lin-12);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotBrazoDer);glRasterPos2f (col+14, lin-12);dibujarCadena (etiqueta);

    printf (etiqueta, "AntBraz.");glRasterPos2f (col, lin-13.5);dibujarCadena (etiqueta);
    printf (etiqueta, "W.");glRasterPos2f (col+7, lin-13.5);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotAnteBrazolzq);glRasterPos2f (col+9, lin-13.5);dibujarCadena (etiqueta);
    printf (etiqueta, "O.");glRasterPos2f (col+12, lin-13.5);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotAnteBrazoDer);glRasterPos2f (col+14, lin-13.5);dibujarCadena (etiqueta);

    printf (etiqueta, "Pierna.");glRasterPos2f (col, lin-15);dibujarCadena (etiqueta);
    printf (etiqueta, "E.");glRasterPos2f (col+7, lin-15);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotPiernalzq);glRasterPos2f (col+9, lin-15);dibujarCadena (etiqueta);
    printf (etiqueta, "I.");glRasterPos2f (col+12, lin-15);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotPiernaDer);glRasterPos2f (col+14, lin-15);dibujarCadena (etiqueta);

    printf (etiqueta, "AntPier.");glRasterPos2f (col, lin-16.5);dibujarCadena (etiqueta);
    printf (etiqueta, "R.");glRasterPos2f (col+7, lin-16.5);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotAntePiernalzq);glRasterPos2f (col+9, lin-16.5);dibujarCadena (etiqueta);
    printf (etiqueta, "U.");glRasterPos2f (col+12, lin-16.5);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotAntePiernaDer);glRasterPos2f (col+14, lin-16.5);dibujarCadena (etiqueta);

    printf (etiqueta, "Pie.");glRasterPos2f (col, lin-18);dibujarCadena (etiqueta);
    printf (etiqueta, "A.");glRasterPos2f (col+7, lin-18);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotPielzq);glRasterPos2f (col+9, lin-18);dibujarCadena (etiqueta);
    printf (etiqueta, "L.");glRasterPos2f (col+12, lin-18);dibujarCadena (etiqueta);
    printf (etiqueta, "%d",RotPieDer);glRasterPos2f (col+14, lin-18);dibujarCadena (etiqueta);

    glColor3f(0.25,1,0);

    printf (etiqueta, "Pelvis[X]: %d",RotPelvisX);glRasterPos2f (col, lin-21);dibujarCadena (etiqueta);
    printf (etiqueta, "Pelvis[Y]: %d",RotPelvisY);glRasterPos2f (col, lin-22.5);dibujarCadena (etiqueta);
    printf (etiqueta, "Pelvis[Z]: %d",RotPelvisZ);glRasterPos2f (col, lin-24);dibujarCadena (etiqueta);

    printf (etiqueta, "Zoom +/-");glRasterPos2f (col, lin-35);dibujarCadena (etiqueta);
    printf (etiqueta, "%i",intVZoom);glRasterPos2f (col+9, lin-35);dibujarCadena (etiqueta);

    printf (etiqueta, "Guardar: ");glRasterPos2f (col,lin-36.5);dibujarCadena (etiqueta);
    printf (etiqueta, "Click Izquierdo");glRasterPos2f (col+9,lin-36.5);dibujarCadena (etiqueta);
    printf (etiqueta, "Leer.");glRasterPos2f (col,lin-38);dibujarCadena (etiqueta);
    printf (etiqueta, "Click Derecho");glRasterPos2f (col+9,lin-38);dibujarCadena (etiqueta);

    printf (etiqueta, "Escen.X: ");glRasterPos2f (col,lin-39.5);dibujarCadena (etiqueta);
    printf (etiqueta, "UP/DOWN");glRasterPos2f (col+9,lin-39.5);dibujarCadena (etiqueta);
    printf (etiqueta, "%i",RotEscenaX);glRasterPos2f (col+24,lin-39.5);dibujarCadena (etiqueta);

    printf (etiqueta, "Escen.Y: ");glRasterPos2f (col,lin-41);dibujarCadena (etiqueta);
    printf (etiqueta, "LEFT/RIGHT");glRasterPos2f (col+9,lin-41);dibujarCadena (etiqueta);
    printf (etiqueta, "%i",RotEscenaY);glRasterPos2f (col+24,lin-41);dibujarCadena (etiqueta);

    //CENTRO DE GRAVEDAD TOTAL
    glColor3f(1,1,1);
    printf (etiqueta, "CG Principal");
    glRasterPos2f (col,lin-27);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGTOT[0][0]);
    glRasterPos2f (col,lin-28.5);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGTOT[1][0]);
    glRasterPos2f (col,lin-30);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGTOT[2][0]);
    glRasterPos2f (col,lin-31.5);dibujarCadena (etiqueta);

```

```

}

void setCoordenadas(){
    glColor3f(0,1,0);
    //CG BRAZO DERECHO
    printf (etiqueta, "CGBrazDer");glRasterPos2f (col3,17);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGBD[0][0]);glRasterPos2f (col3,16);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGBD[1][0]);glRasterPos2f (col3,15);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGBD[2][0]);glRasterPos2f (col3,14);dibujarCadena (etiqueta);
    //CG BRAZO IZQUIERDO
    printf (etiqueta, "CGBrazIzq");glRasterPos2f (col2,17);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGBI[0][0]);glRasterPos2f (col2,16);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGBI[1][0]);glRasterPos2f (col2,15);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGBI[2][0]);glRasterPos2f (col2,14);dibujarCadena (etiqueta);
    //CG ANTEBRAZO DERECHO
    printf (etiqueta, "CGAnteBrazDer");glRasterPos2f (col3,12);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGABD[0][0]);glRasterPos2f (col3,11);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGABD[1][0]);glRasterPos2f (col3,10);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGABD[2][0]);glRasterPos2f (col3,9);dibujarCadena (etiqueta);
    //CG ANTEBRAZO IZQUIERDO
    printf (etiqueta, "CGAnteBrazIzq");glRasterPos2f (col2,12);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGABI[0][0]);glRasterPos2f (col2,11);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGABI[1][0]);glRasterPos2f (col2,10);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGABI[2][0]);glRasterPos2f (col2,9);dibujarCadena (etiqueta);
    //CG MANO DERECHA
    printf (etiqueta, "CGManoDer");glRasterPos2f (col3,7);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGMD[0][0]);glRasterPos2f (col3,6);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGMD[1][0]);glRasterPos2f (col3,5);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGMD[2][0]);glRasterPos2f (col3,4);dibujarCadena (etiqueta);
    //CG MANO IZQUIERDA
    printf (etiqueta, "CGManoIzq");glRasterPos2f (col2,7);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGMI[0][0]);glRasterPos2f (col2,6);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGMI[1][0]);glRasterPos2f (col2,5);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGMI[2][0]);glRasterPos2f (col2,4);dibujarCadena (etiqueta);
    //CG PIERNA DERECHA
    printf (etiqueta, "CGPierDer");glRasterPos2f (col3,2);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGPD[0][0]);glRasterPos2f (col3,1);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGPD[1][0]);glRasterPos2f (col3,0);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGPD[2][0]);glRasterPos2f (col3,-1);dibujarCadena (etiqueta);
    //CG PIERNA IZQUIERDA
    printf (etiqueta, "CGPierIzq");glRasterPos2f (col2,2);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGPI[0][0]);glRasterPos2f (col2,1);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGPI[1][0]);glRasterPos2f (col2,0);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGPI[2][0]);glRasterPos2f (col2,-1);dibujarCadena (etiqueta);
    //CG ANTEPIERNA DERECHA
    printf (etiqueta, "CGAntPierDer");glRasterPos2f (col3,-3);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGAPD[0][0]);glRasterPos2f (col3,-4);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGAPD[1][0]);glRasterPos2f (col3,-5);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGAPD[2][0]);glRasterPos2f (col3,-6);dibujarCadena (etiqueta);
    //CG ANTEPIERNA IZQUIERDA
    printf (etiqueta, "CGAntPierIzq");glRasterPos2f (col2,-3);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGAPI[0][0]);glRasterPos2f (col2,-4);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGAPI[1][0]);glRasterPos2f (col2,-5);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGAPI[2][0]);glRasterPos2f (col2,-6);dibujarCadena (etiqueta);
    //CG PIE DERECHO
    printf (etiqueta, "CGPieDer");glRasterPos2f (col3,-8);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGFD[0][0]);glRasterPos2f (col3,-9);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGFD[1][0]);glRasterPos2f (col3,-10);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGFD[2][0]);glRasterPos2f (col3,-11);dibujarCadena (etiqueta);
    //CG PIE IZQUIERDO
    printf (etiqueta, "CGPieIzq");glRasterPos2f (col2,-8);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGFI[0][0]);glRasterPos2f (col2,-9);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGFI[1][0]);glRasterPos2f (col2,-10);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGFI[2][0]);glRasterPos2f (col2,-11);dibujarCadena (etiqueta);
    //CG TORZO
    printf (etiqueta, "CGTorzo");glRasterPos2f (col3,22);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGTOR[0][0]);glRasterPos2f (col3,21);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGTOR[1][0]);glRasterPos2f (col3,20);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGTOR[2][0]);glRasterPos2f (col3,19);dibujarCadena (etiqueta);
    //CG CABEZA
    printf (etiqueta, "CGCabeza");glRasterPos2f (col2,22);dibujarCadena (etiqueta);
    printf (etiqueta, "[X]: %f",CGCAB[0][0]);glRasterPos2f (col2,21);dibujarCadena (etiqueta);
    printf (etiqueta, "[Y]: %f",CGCAB[1][0]);glRasterPos2f (col2,20);dibujarCadena (etiqueta);
    printf (etiqueta, "[Z]: %f",CGCAB[2][0]);glRasterPos2f (col2,19);dibujarCadena (etiqueta);
}

void setMalla(void){
    glColor3f(0,.5,0);
    glBegin(GL_LINES);
    for (malla=-80;malla<=80;malla=malla+8)
    {
        //MALLA PLANO XZ HORIZONTAL

```

```

        glVertex3f(-80,-16,malla);
        glVertex3f(80,-16,malla);
    }
    for (malla=-80;malla<=80;malla=malla+8)
    {
        //MALLA PLANO XZ VERTICAL
        glVertex3f(malla,-16,-80);
        glVertex3f(malla,-16,80);
    }
    glEnd();
}

void setRxyz(){
    //ROTACIONES DEL MODELO COMPLETO DESDE LA PELVIS
    Rx[0][0]= 1;
    Rx[1][1]= cos(3.1416*RotPelvisX/180);
    Rx[1][2]=-sin(3.1416*RotPelvisX/180);
    Rx[2][1]= sin(3.1416*RotPelvisX/180);
    Rx[2][2]= cos(3.1416*RotPelvisX/180);
    Rx[3][3]= 1;

    Ry[0][0]= cos(3.1416*RotPelvisY/180);
    Ry[0][2]= sin(3.1416*RotPelvisY/180);
    Ry[1][1]= 1;
    Ry[2][0]= -sin(3.1416*RotPelvisY/180);
    Ry[2][2]= cos(3.1416*RotPelvisY/180);
    Ry[3][3]= 1;

    Rz[0][0]= cos(3.1416*RotPelvisZ/180);
    Rz[0][1]=-sin(3.1416*RotPelvisZ/180);
    Rz[1][0]= sin(3.1416*RotPelvisZ/180);
    Rz[1][1]= cos(3.1416*RotPelvisZ/180);
    Rz[2][2]= 1;
    Rz[3][3]= 1;
}

//CG BRAZO DERECHO
void setCGBD(void){
    double P[MAX][MAX], T1[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (HOMBRO DER)
    P[0][0]=-3.5;
    P[1][0]= -3.17;
    P[2][0]= 0;
    P[3][0]= 1;

    RxSubArtic[0][0]= 1;
    RxSubArtic[1][1]= cos(3.1416*RotBrazoDer/180);
    RxSubArtic[1][2]=-sin(3.1416*RotBrazoDer/180);
    RxSubArtic[2][1]= sin(3.1416*RotBrazoDer/180);
    RxSubArtic[2][2]= cos(3.1416*RotBrazoDer/180);
    RxSubArtic[3][3]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotTorzo/180);
    RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
    RxArtic[2][1]= sin(3.1416*RotTorzo/180);
    RxArtic[2][2]= cos(3.1416*RotTorzo/180);
    RxArtic[3][3]= 1;

    //TRASLACION
    T1[0][0]= 0;
    T1[1][0]= 7.5;
    T1[2][0]= 0;
    T1[3][0]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGBD[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL BRAZO
    for (i=0;i<=3;i++){
        for (j=0;j<=0;j++){
            for (k=0;k<=3;k++){
                D[i][j]= D[i][j]+(RxSubArtic[i][k]*P[k][j]);
            }
        }
        //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO DER)
        for (i=0;i<=3;i++){
            D[i][0]=D[i][0]+T1[i][0];
        }
        //OBTIENE POSICION CG AL ROTAR EL TORZO

```

B[6] Sal to vertical y parabólico

```
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      CGBD[i][j]= CGBD[i][j]+(Rx[i][k]*A[k][j]);

glVertex3f(CGBD[0][0],CGBD[1][0],CGBD[2][0]);
}

//CG BRAZO IZQUIERDO
void setCGBI(void){
  double P[MAX][MAX], T1[MAX][MAX];
  //POSICION CG A PARTIR DEL PROXIMAL (HOMBRO IZQ)
  P[0][0]=3.5;
  P[1][0]= -3.17;
  P[2][0]= 0;
  P[3][0]= 1;

  RxSubArtic[0][0]= 1;
  RxSubArtic[1][1]= cos(3.1416*RotBrazolzq/180);
  RxSubArtic[1][2]=-sin(3.1416*RotBrazolzq/180);
  RxSubArtic[2][1]= sin(3.1416*RotBrazolzq/180);
  RxSubArtic[2][2]= cos(3.1416*RotBrazolzq/180);
  RxSubArtic[3][3]= 1;

  RxArtic[0][0]= 1;
  RxArtic[1][1]= cos(3.1416*RotTorzo/180);
  RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
  RxArtic[2][1]= sin(3.1416*RotTorzo/180);
  RxArtic[2][2]= cos(3.1416*RotTorzo/180);
  RxArtic[3][3]= 1;

  //TRASLACION
  T1[0][0]= 0;
  T1[1][0]= 7.5;
  T1[2][0]= 0;
  T1[3][0]= 1;

  for (i=0;i<=3; i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGBI[i][0]=0;
  }
  //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
  //OBTIENE POSICION CG AL ROTAR EL BRAZO
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        D[i][j]= D[i][j]+(RxSubArtic[i][k]*P[k][j]);
  //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO IZQ)
  for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
  //OBTIENE POSICION CG AL ROTAR EL TORZO
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
```



```

    A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
    for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
        CGBI[i][j]= CGBI[i][j]+(Rx[i][k]*A[k][j]);

    glVertex3f(CGBI[0][0],CGBI[1][0],CGBI[2][0]);
}

//CG ANTEBRAZO DERECHO
void setCGABD(){
    double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CODO DER)
    P[0][0]=-3.5;
    P[1][0]=-2.28;
    P[2][0]=0;
    P[3][0]=1;

    RxSubSubArtic[0][0]=1;
    RxSubSubArtic[1][1]=cos(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[1][2]=-sin(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[2][1]=sin(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[2][2]=cos(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[3][3]=1;

    RxSubArtic[0][0]=1;
    RxSubArtic[1][1]=cos(3.1416*RotBrazoDer/180);
    RxSubArtic[1][2]=-sin(3.1416*RotBrazoDer/180);
    RxSubArtic[2][1]=sin(3.1416*RotBrazoDer/180);
    RxSubArtic[2][2]=cos(3.1416*RotBrazoDer/180);
    RxSubArtic[3][3]=1;

    RxArtic[0][0]=1;
    RxArtic[1][1]=cos(3.1416*RotTorzo/180);
    RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
    RxArtic[2][1]=sin(3.1416*RotTorzo/180);
    RxArtic[2][2]=cos(3.1416*RotTorzo/180);
    RxArtic[3][3]=1;

//TRASLACION
    T1[0][0]=0;
    T1[1][0]=7.5;
    T1[2][0]=0;
    T1[3][0]=1;

    T2[0][0]=0;
    T2[1][0]=-5.5;
    T2[2][0]=0;
    T2[3][0]=1;

    for (i=0;i<=3;i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGABD[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL ANTEBRAZO
    for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
        E[i][j]=E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (CODO DER)
    for (i=0;i<=3;i++)
        E[i][0]=E[i][0]+T2[i][0];
//OBTIENE POSICION CG AL ROTAR EL BRAZO
    for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
        D[i][j]=D[i][j]+(RxSubArtic[i][k]*E[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO DER)
    for (i=0;i<=3;i++)
        D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR EL TORZO
    for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
        C[i][j]=C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
    for (i=0;i<=3;i++)

```

B[8] Salto vertical y parabólico

```
for (j=0;j<=0;j++)
  for (k=0;k<=3;k++)
    B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      CGABD[i][j]= CGABD[i][j]+(Rx[i][k]*A[k][j]);

  glVertex3f(CGABD[0][0],CGABD[1][0],CGABD[2][0]);
}

//CG ANTEBRAZO IZQUIERDO
void setCGABI(){
  double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
  //POSICION CG A PARTIR DEL PROXIMAL (CODO IZQ)
  P[0][0]= 3.5;
  P[1][0]= -2.28;
  P[2][0]= 0;
  P[3][0]= 1;

  RxSubSubArtic[0][0]= 1;
  RxSubSubArtic[1][1]= cos(3.1416*RotAnteBrazolzq/180);
  RxSubSubArtic[1][2]=sin(3.1416*RotAnteBrazolzq/180);
  RxSubSubArtic[2][1]= sin(3.1416*RotAnteBrazolzq/180);
  RxSubSubArtic[2][2]= cos(3.1416*RotAnteBrazolzq/180);
  RxSubSubArtic[3][3]= 1;

  RxSubArtic[0][0]= 1;
  RxSubArtic[1][1]= cos(3.1416*RotBrazolzq/180);
  RxSubArtic[1][2]=sin(3.1416*RotBrazolzq/180);
  RxSubArtic[2][1]= sin(3.1416*RotBrazolzq/180);
  RxSubArtic[2][2]= cos(3.1416*RotBrazolzq/180);
  RxSubArtic[3][3]= 1;

  RxArtic[0][0]= 1;
  RxArtic[1][1]= cos(3.1416*RotTorzo/180);
  RxArtic[1][2]=sin(3.1416*RotTorzo/180);
  RxArtic[2][1]= sin(3.1416*RotTorzo/180);
  RxArtic[2][2]= cos(3.1416*RotTorzo/180);
  RxArtic[3][3]= 1;

  //TRASLACION
  T1[0][0]= 0;
  T1[1][0]= 7.5;
  T1[2][0]= 0;
  T1[3][0]= 1;

  T2[0][0]= 0;
  T2[1][0]= -5.5;
  T2[2][0]= 0;
  T2[3][0]= 1;

  for (i=0;i<=3; i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGABI[i][0]=0;
  }
  //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
  //OBTIENE POSICION CG AL ROTAR EL ANTEBRAZO
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (CODO IZQ)
  for (i=0;i<=3;i++)
    E[i][0]=E[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR EL BRAZO
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO IZQ)
  for (i=0;i<=3;i++)
```

```

D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR EL TORZO
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      CGABI[i][j]= CGABI[i][j]+(Rx[i][k]*A[k][j]);

  glVertex3f(CGABI[0][0],CGABI[1][0],CGABI[2][0]);
}

//CG MANO DERECHA
void setCGMD(){
  double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
  //POSICION CG A PARTIR DEL PROXIMAL (CODO DER)
  P[0][0]=-3.5;
  P[1][0]=-5.52;
  P[2][0]=0;
  P[3][0]=1;

  RxSubSubArtic[0][0]=1;
  RxSubSubArtic[1][1]=cos(3.1416*RotAnteBrazoDer/180);
  RxSubSubArtic[1][2]=-sin(3.1416*RotAnteBrazoDer/180);
  RxSubSubArtic[2][1]=sin(3.1416*RotAnteBrazoDer/180);
  RxSubSubArtic[2][2]=cos(3.1416*RotAnteBrazoDer/180);
  RxSubSubArtic[3][3]=1;

  RxSubArtic[0][0]=1;
  RxSubArtic[1][1]=cos(3.1416*RotBrazoDer/180);
  RxSubArtic[1][2]=-sin(3.1416*RotBrazoDer/180);
  RxSubArtic[2][1]=sin(3.1416*RotBrazoDer/180);
  RxSubArtic[2][2]=cos(3.1416*RotBrazoDer/180);
  RxSubArtic[3][3]=1;

  RxArtic[0][0]=1;
  RxArtic[1][1]=cos(3.1416*RotTorzo/180);
  RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
  RxArtic[2][1]=sin(3.1416*RotTorzo/180);
  RxArtic[2][2]=cos(3.1416*RotTorzo/180);
  RxArtic[3][3]=1;

  //TRASLACION
  T1[0][0]=0;
  T1[1][0]=7.5;
  T1[2][0]=0;
  T1[3][0]=1;

  T2[0][0]=0;
  T2[1][0]=-5.5;
  T2[2][0]=0;
  T2[3][0]=1;

  for (i=0;i<=3;i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGMD[i][0]=0;
  }
  //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
  //OBTIENE POSICION CG AL ROTAR EL ANTEBRAZO
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
  //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (CODO DER)
  for (i=0;i<=3;i++)
    E[i][0]=E[i][0]+T2[i][0];

```

B[10] Sal to vertical y parabólico

```
//OBTIENE POSICION CG AL ROTAR EL BRAZO
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO DER)
for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR EL TORZO
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            CGMD[i][j]= CGMD[i][j]+(Rx[i][k]*A[k][j]);

glVertex3f(CGMD[0][0],CGMD[1][0],CGMD[2][0]);
}

//CG MANO IZQUIERDA
void setCGMI(){
double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
//POSICION CG A PARTIR DEL PROXIMAL (CODO IZQ)
P[0][0]= 3.5;
P[1][0]= -5.52;
P[2][0]= 0;
P[3][0]= 1;

RxSubSubArtic[0][0]= 1;
RxSubSubArtic[1][1]= cos(3.1416*RotAnteBrazolzq/180);
RxSubSubArtic[1][2]=-sin(3.1416*RotAnteBrazolzq/180);
RxSubSubArtic[2][1]= sin(3.1416*RotAnteBrazolzq/180);
RxSubSubArtic[2][2]= cos(3.1416*RotAnteBrazolzq/180);
RxSubSubArtic[3][3]= 1;

RxSubArtic[0][0]= 1;
RxSubArtic[1][1]= cos(3.1416*RotBrazolzq/180);
RxSubArtic[1][2]=-sin(3.1416*RotBrazolzq/180);
RxSubArtic[2][1]= sin(3.1416*RotBrazolzq/180);
RxSubArtic[2][2]= cos(3.1416*RotBrazolzq/180);
RxSubArtic[3][3]= 1;

RxArtic[0][0]= 1;
RxArtic[1][1]= cos(3.1416*RotTorzo/180);
RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
RxArtic[2][1]= sin(3.1416*RotTorzo/180);
RxArtic[2][2]= cos(3.1416*RotTorzo/180);
RxArtic[3][3]= 1;

//TRASLACION
T1[0][0]= 0;
T1[1][0]= 7.5;
T1[2][0]= 0;
T1[3][0]= 1;

T2[0][0]= 0;
T2[1][0]= -5.5;
T2[2][0]= 0;
T2[3][0]= 1;

for (i=0;i<=3;i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGMI[i][0]=0;
}
//-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
//OBTIENE POSICION CG AL ROTAR EL ANTEBRAZO
```

```

for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
      //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (CODO IZQ)
for (i=0;i<=3;i++)
  E[i][0]=E[i][0]+T2[i][0];
  //OBTIENE POSICION CG AL ROTAR EL BRAZO
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
      //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO IZQ)
    for (i=0;i<=3;i++)
      D[i][0]=D[i][0]+T1[i][0];
  //OBTIENE POSICION CG AL ROTAR EL TORZO
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      CGMI[i][j]= CGMI[i][j]+(Rx[i][k]*A[k][j]);

  glVertex3f(CGMI[0][0],CGMI[1][0],CGMI[2][0]);
}

//CG PIERNA DERECHA
void setCGPD(){
  double P[MAX][MAX];
  //POSICION CG A PARTIR DEL PROXIMAL (CADERA DER)
  P[0][0]= -1.5;
  P[1][0]= -3.27;
  P[2][0]= 0;
  P[3][0]= 1;

  RxArtic[0][0]= 1;
  RxArtic[1][1]= cos(3.1416*RotPiernaDer/180);
  RxArtic[1][2]=sin(3.1416*RotPiernaDer/180);
  RxArtic[2][1]= sin(3.1416*RotPiernaDer/180);
  RxArtic[2][2]= cos(3.1416*RotPiernaDer/180);
  RxArtic[3][3]= 1;

  for (i=0;i<=3; i++){
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGPD[i][0]=0;
  }
  //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
  //OBTIENE POSICION CG AL ROTAR LA PIERNA
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        B[i][j]= B[i][j]+(RxArtic[i][k]*P[k][j]);
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        C[i][j]= C[i][j]+(Rz[i][k]*B[k][j]);
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        D[i][j]= D[i][j]+(Ry[i][k]*C[k][j]);
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        CGPD[i][j]= CGPD[i][j]+(Rx[i][k]*D[k][j]);

```

B[12] Salto vertical y parabólico

```
    glVertex3f(CGPD[0][0],CGPD[1][0],CGPD[2][0]);
}

//CG PIERNA IZQUIERDA
void setCGPI(){
    double P[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CADERA IZQ)
    P[0][0]=1.5;
    P[1][0]= -3.27;
    P[2][0]= 0;
    P[3][0]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);
    RxArtic[1][2]=-sin(3.1416*RotPiernalzq/180);
    RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);
    RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);
    RxArtic[3][3]= 1;

    for (i=0;i<=3; i++){
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGPI[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR LA PIERNA
    for (i=0;i<=3;i++){
        for (j=0;j<=0;j++){
            for (k=0;k<=3;k++){
                B[i][j]= B[i][j]+(RxArtic[i][k]*P[k][j]);
                //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
            }
            for (i=0;i<=3;i++){
                for (j=0;j<=0;j++){
                    for (k=0;k<=3;k++){
                        C[i][j]= C[i][j]+(Rz[i][k]*B[k][j]);
                        //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
                    }
                    for (i=0;i<=3;i++){
                        for (j=0;j<=0;j++){
                            for (k=0;k<=3;k++){
                                D[i][j]= D[i][j]+(Ry[i][k]*C[k][j]);
                                //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
                            }
                            for (i=0;i<=3;i++){
                                for (j=0;j<=0;j++){
                                    for (k=0;k<=3;k++){
                                        CGPI[i][j]= CGPI[i][j]+(Rx[i][k]*D[k][j]);
                                    }
                                }
                            }
                        }
                    }
                }
            }
            glVertex3f(CGPI[0][0],CGPI[1][0],CGPI[2][0]);
        }
    }

//CG ANTEPIERNA DERECHA
void setCGAPD(){
    double P[MAX][MAX],T1[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (RODILLA DER)
    P[0][0]= -1.5;
    P[1][0]= -3.12;
    P[2][0]= 0;
    P[3][0]= 1;

    RxSubArtic[0][0]= 1;
    RxSubArtic[1][1]= cos(3.1416*RotAntePiernaDer/180);
    RxSubArtic[1][2]=-sin(3.1416*RotAntePiernaDer/180);
    RxSubArtic[2][1]= sin(3.1416*RotAntePiernaDer/180);
    RxSubArtic[2][2]= cos(3.1416*RotAntePiernaDer/180);
    RxSubArtic[3][3]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernaDer/180);
    RxArtic[1][2]=-sin(3.1416*RotPiernaDer/180);
    RxArtic[2][1]= sin(3.1416*RotPiernaDer/180);
    RxArtic[2][2]= cos(3.1416*RotPiernaDer/180);
    RxArtic[3][3]= 1;

    //TRASLACION
    T1[0][0]= 0;
    T1[1][0]= -8;
    T1[2][0]= 0;
    T1[3][0]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
    }
}
```

```

C[i][0]=0;
D[i][0]=0;
CGAPD[i][0]=0;
}
//-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
//OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      D[i][j]= D[i][j]+(RxSubArtic[i][k]*P[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA DER)
for (i=0;i<=3;i++)
  D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR LA PIERNA
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      CGAPD[i][j]= CGAPD[i][j]+(Rx[i][k]*A[k][j]);

glVertex3f(CGAPD[0][0],CGAPD[1][0],CGAPD[2][0]);
}

//CG ANTEPIERNA IZQUIERDA
void setCGAPI(){
  double P[MAX][MAX],T1[MAX][MAX];
  //POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)
  P[0][0]= 1.5;
  P[1][0]= -3.12;
  P[2][0]= 0;
  P[3][0]= 1;

  RxSubArtic[0][0]= 1;
  RxSubArtic[1][1]= cos(3.1416*RotAntePiernalzq/180);
  RxSubArtic[1][2]= -sin(3.1416*RotAntePiernalzq/180);
  RxSubArtic[2][1]= sin(3.1416*RotAntePiernalzq/180);
  RxSubArtic[2][2]= cos(3.1416*RotAntePiernalzq/180);
  RxSubArtic[3][3]= 1;

  RxArtic[0][0]= 1;
  RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);
  RxArtic[1][2]= -sin(3.1416*RotPiernalzq/180);
  RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);
  RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);
  RxArtic[3][3]= 1;

  //TRASLACION
  T1[0][0]= 0;
  T1[1][0]= -8;
  T1[2][0]= 0;
  T1[3][0]= 1;

  for (i=0;i<=3; i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    CGAPI[i][0]=0;
  }
  //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
  //OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        D[i][j]= D[i][j]+(RxSubArtic[i][k]*P[k][j]);
  //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)
  for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
  //OBTIENE POSICION CG AL ROTAR LA PIERNA

```

B[14] Sal to vertical y parabólico

```
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
  for (j=0;j<=0;j++)
    for (k=0;k<=3;k++)
      CGAPI[i][j]= CGAPI[i][j]+(Rx[i][k]*A[k][j]);

glVertex3f(CGAPI[0][0],CGAPI[1][0],CGAPI[2][0]);
}

//CG PIE DERECHO
void setCGFD(){
  double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
  //POSICION CG A PARTIR DEL PROXIMAL (TOBILLO DER)
  P[0][0]= 0;
  P[1][0]= -0.5;
  P[2][0]= 0.54;
  P[3][0]= 1;

  RxSubSubArtic[0][0]= 1;
  RxSubSubArtic[1][1]= cos(3.1416*RotPieDer/180);
  RxSubSubArtic[1][2]=-sin(3.1416*RotPieDer/180);
  RxSubSubArtic[2][1]= sin(3.1416*RotPieDer/180);
  RxSubSubArtic[2][2]= cos(3.1416*RotPieDer/180);
  RxSubSubArtic[3][3]= 1;

  RxSubArtic[0][0]= 1;
  RxSubArtic[1][1]= cos(3.1416*RotAntePiernaDer/180);
  RxSubArtic[1][2]=-sin(3.1416*RotAntePiernaDer/180);
  RxSubArtic[2][1]= sin(3.1416*RotAntePiernaDer/180);
  RxSubArtic[2][2]= cos(3.1416*RotAntePiernaDer/180);
  RxSubArtic[3][3]= 1;

  RxArtic[0][0]= 1;
  RxArtic[1][1]= cos(3.1416*RotPiernaDer/180);
  RxArtic[1][2]=-sin(3.1416*RotPiernaDer/180);
  RxArtic[2][1]= sin(3.1416*RotPiernaDer/180);
  RxArtic[2][2]= cos(3.1416*RotPiernaDer/180);
  RxArtic[3][3]= 1;

  //TRASLACION
  T1[0][0]= 0;
  T1[1][0]= -8;
  T1[2][0]= 0;
  T1[3][0]= 1;

  T2[0][0]= -1.5;
  T2[1][0]= -7;
  T2[2][0]= .5;
  T2[3][0]= 1;

  for (i=0;i<=3; i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGFD[i][0]=0;
  }
  //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
  //OBTIENE POSICION CG AL ROTAR EL PIE
  for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
      for (k=0;k<=3;k++)
        E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
  //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (TOBILLO DER)
  for (i=0;i<=3;i++)
    E[i][0]=E[i][0]+T2[i][0];
  //OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
  for (i=0;i<=3;i++)
```



```

    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
            //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA DER)
for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
    //OBTIENE POSICION CG AL ROTAR LA PIERNA
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;j<=0;j++)
    for (j=0;j<=0;j++)
        for (k=0;k<=3;k++)
            CGFD[i][j]= CGFD[i][j]+(Rx[i][k]*A[k][j]);

    glVertex3f(CGFD[0][0],CGFD[1][0],CGFD[2][0]);
}

//CG PIE IZQUIERDO
void setCGFI(){
    double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (TOBILLO IZQ)
    P[0][0]= 0;
    P[1][0]= -0.5;
    P[2][0]= 0.54;
    P[3][0]= 1;

    RxSubSubArtic[0][0]= 1;
    RxSubSubArtic[1][1]= cos(3.1416*RotPielzq/180);
    RxSubSubArtic[1][2]=sin(3.1416*RotPielzq/180);
    RxSubSubArtic[2][1]= sin(3.1416*RotPielzq/180);
    RxSubSubArtic[2][2]= cos(3.1416*RotPielzq/180);
    RxSubSubArtic[3][3]= 1;

    RxSubArtic[0][0]= 1;
    RxSubArtic[1][1]= cos(3.1416*RotAntePiernalzq/180);
    RxSubArtic[1][2]=sin(3.1416*RotAntePiernalzq/180);
    RxSubArtic[2][1]= sin(3.1416*RotAntePiernalzq/180);
    RxSubArtic[2][2]= cos(3.1416*RotAntePiernalzq/180);
    RxSubArtic[3][3]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);
    RxArtic[1][2]=sin(3.1416*RotPiernalzq/180);
    RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);
    RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);
    RxArtic[3][3]= 1;

    //TRASLACION
    T1[0][0]= 0;
    T1[1][0]= -8;
    T1[2][0]= 0;
    T1[3][0]= 1;

    T2[0][0]= 1.5;
    T2[1][0]= -7;
    T2[2][0]= .5;
    T2[3][0]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGFI[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL PIE
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)

```

```

    for (k=0;k<=3;k++)
        E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
        //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (TOBILLO IZQ)
        for (i=0;i<=3;i++)
            E[i][0]=E[i][0]+T2[i][0];
        //OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
        for (i=0;i<=3;i++)
            for (j=0;j<=0;j++)
                for (k=0;k<=3;k++)
                    D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
                    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)
        for (i=0;i<=3;i++)
            D[i][0]=D[i][0]+T1[i][0];
            //OBTIENE POSICION CG AL ROTAR LA PIERNA
        for (i=0;i<=3;i++)
            for (j=0;j<=0;j++)
                for (k=0;k<=3;k++)
                    C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
                    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
        for (i=0;i<=3;i++)
            for (j=0;j<=0;j++)
                for (k=0;k<=3;k++)
                    B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
                    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
        for (i=0;i<=3;i++)
            for (j=0;j<=0;j++)
                for (k=0;k<=3;k++)
                    A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
                    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
        for (i=0;i<=3;i++)
            for (j=0;j<=0;j++)
                for (k=0;k<=3;k++)
                    CGFI[i][j]= CGFI[i][j]+(Rx[i][k]*A[k][j]);

        glVertex3f(CGFI[0][0],CGFI[1][0],CGFI[2][0]);
}

//CG TORZO
void setCGTOR(void){
    double P[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL ( ARTICULACION LUMBAR)
    P[0][0]= 0;
    P[1][0]= 3.81;
    P[2][0]= 0;
    P[3][0]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotTorzo/180);
    RxArtic[1][2]= -sin(3.1416*RotTorzo/180);
    RxArtic[2][1]= sin(3.1416*RotTorzo/180);
    RxArtic[2][2]= cos(3.1416*RotTorzo/180);
    RxArtic[3][3]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        CGTOR[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL TORZO
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                C[i][j]= C[i][j]+(RxArtic[i][k]*P[k][j]);
                //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
                //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
                //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                CGTOR[i][j]= CGTOR[i][j]+(Rx[i][k]*A[k][j]);

    glVertex3f(CGTOR[0][0],CGTOR[1][0],CGTOR[2][0]);
}

```

```

//CG CABEZA
void setCGCAB(void){
  double P[MAX][MAX];
  //POSICION CG A PARTIR DEL PROXIMAL (ARTICULACION LUMBAR)
  P[0][0]= 0;
  P[1][0]= 11.03;
  P[2][0]= 0;
  P[3][0]= 1;

  RxArtic[0][0]= 1;
  RxArtic[1][1]= cos(3.1416*RotTorzo/180);
  RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
  RxArtic[2][1]= sin(3.1416*RotTorzo/180);
  RxArtic[2][2]= cos(3.1416*RotTorzo/180);
  RxArtic[3][3]= 1;

  for (i=0;i<=3;i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    CGCAB[i][0]=0;
  }
  //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
  //OBTIENE POSICION CG AL ROTAR EL TORZO
  for (i=0;i<=3;i++){
    for (j=0;j<=0;j++){
      for (k=0;k<=3;k++){
        C[i][j]= C[i][j]+(RxArtic[i][k]*P[k][j]);
      }
    }
  }
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
  for (i=0;i<=3;i++){
    for (j=0;j<=0;j++){
      for (k=0;k<=3;k++){
        B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
      }
    }
  }
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
  for (i=0;i<=3;i++){
    for (j=0;j<=0;j++){
      for (k=0;k<=3;k++){
        A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
      }
    }
  }
  //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
  for (i=0;i<=3;i++){
    for (j=0;j<=0;j++){
      for (k=0;k<=3;k++){
        CGCAB[i][j]= CGCAB[i][j]+(Rx[i][k]*A[k][j]);
      }
    }
  }

  glVertex3f(CGCAB[0][0],CGCAB[1][0],CGCAB[2][0]);
}

//CG TOTAL
void setCGTOT(void){
  glColor3f(1,0,0);
  glLineWidth(2);
  for (i=0;i<=3;i++){
    CGTOT[i][0] =(CGCAB[i][0]*douPCAB)
    +(CGTOR[i][0]*douPTOR)
    +(CGBD[i][0]*douPB)
    +(CGBI[i][0]*douPB)
    +(CGABD[i][0]*douPAB)
    +(CGABI[i][0]*douPAB)
    +(CGMD[i][0]*douPM)
    +(CGMI[i][0]*douPM)
    +(CGPD[i][0]*douPP)
    +(CGPI[i][0]*douPP)
    +(CGAPD[i][0]*douPAP)
    +(CGAPI[i][0]*douPAP)
    +(CGFD[i][0]*douPF)
    +(CGFI[i][0]*douPF);
  }

  //CG PRINCIPAL
  glBegin(GL_POINTS);
  glVertex3f(CGTOT[0][0],CGTOT[1][0],CGTOT[2][0]);
  glEnd();

  //EJES DEL CG PRINCIPAL
  glBegin(GL_LINES);
  //EJE X
  glVertex3f(CGTOT[0][0]-1,CGTOT[1][0],CGTOT[2][0]);
  glVertex3f(CGTOT[0][0]+1,CGTOT[1][0],CGTOT[2][0]);
  //EJE Y
  glVertex3f(CGTOT[0][0],CGTOT[1][0]-1,CGTOT[2][0]);
  glVertex3f(CGTOT[0][0],CGTOT[1][0]+1,CGTOT[2][0]);
  //EJE Z
  glVertex3f(CGTOT[0][0],CGTOT[1][0],CGTOT[2][0]-1);
  glVertex3f(CGTOT[0][0],CGTOT[1][0],CGTOT[2][0]+1);
}

```

B[18] Salto vertical y parabólico

```
        //LINEA DE GRAVEDAD
        glVertex3f(CGTOT[0][0],CGTOT[1][0],CGTOT[2][0]);
        glVertex3f(CGTOT[0][0],-16,CGTOT[2][0]);
    glEnd();
    glLineWidth(1);
}

void setCuello(){
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 24,1.5);
        glVertex3f(2.5, 24,0);
        glVertex3f(0, 24,-1.5);
        glVertex3f(-2.5, 24,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 24.5, 1);
        glVertex3f(1.25, 24.5, 0);
        glVertex3f(0, 24.5, -1);
        glVertex3f(-1.25, 24.5, 0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 26, .75);
        glVertex3f(.75, 26,0);
        glVertex3f(0, 26, -.75);
        glVertex3f(-.75, 26, 0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 24, 1.5);
        glVertex3f(0, 24.5, 1);
        glVertex3f(0, 26, .75);
        glVertex3f(0, 26, -.75);
        glVertex3f(0, 24.5, -1);
        glVertex3f(0, 24,-1.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(2.5, 24,0);
        glVertex3f(1.25, 24.5,0);
        glVertex3f(.75, 26,0);
        glVertex3f(-.75, 26,0);
        glVertex3f(-1.25, 24.5,0);
        glVertex3f(-2.5, 24,0);
    glEnd();
}

void setCabeza(){
    glTranslatef(0,27, 0);
    glScalef(1.5,1.75,1.5);
    glRotatef(90,1,0,0);
    glutWireSphere(1,7,5);
}

void setTorzo(){
    glPushMatrix();
        glScalef(1,1,-1);
        glBegin(GL_LINE_LOOP);
            glVertex3f(0, 21,1.5);
            glVertex3f(2.5, 21,0);
            glVertex3f(0, 21,-1.5);
            glVertex3f(-2.5, 21,0);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(-1.5, 22,2);
            glVertex3f(1.5, 22,2);
            glVertex3f(2.75, 22,1);
            glVertex3f(2.75, 22,-.5);
            glVertex3f(1.5, 22,-1.5);
            glVertex3f(-1.5, 22,-1.5);
            glVertex3f(-2.75, 22,-.5);
            glVertex3f(-2.75, 22,1);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(-1.5, 23,2);
            glVertex3f(1.5, 23,2);
            glVertex3f(2.75, 23,1);
            glVertex3f(2.75, 23,-.5);
            glVertex3f(1.5, 23,-1.5);
            glVertex3f(-1.5, 23,-1.5);
            glVertex3f(-2.75, 23,-.5);
            glVertex3f(-2.75, 23,1);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(0, 24,1.5);
            glVertex3f(2.5, 24,0);
        glEnd();
    glPopMatrix();
}
```

```

        glVertex3f(0, 24,-1.5);
        glVertex3f(-2.5, 24,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 21, 1.5);
        glVertex3f(1.5, 22,2);
        glVertex3f(1.5, 23,2);
        glVertex3f(0, 24,1.5);
        glVertex3f(-1.5, 23,2);
        glVertex3f(-1.5, 22,2);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(2.5, 21,0);
        glVertex3f(2.75, 22,1);
        glVertex3f(2.75, 23,1);
        glVertex3f(2.5, 24,0);
        glVertex3f(2.75, 23,-.5);
        glVertex3f(2.75, 22,-.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 21,-1.5);
        glVertex3f(1.5, 22,-1.5);
        glVertex3f(1.5, 23,-1.5);
        glVertex3f(0, 24,-1.5);
        glVertex3f(-1.5, 23,-1.5);
        glVertex3f(-1.5, 22,-1.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(-2.5, 21,0);
        glVertex3f(-2.75, 22,1);
        glVertex3f(-2.75, 23,1);
        glVertex3f(-2.5, 24,0);
        glVertex3f(-2.75, 23,-.5);
        glVertex3f(-2.75, 22,-.5);
    glEnd();
    glPopMatrix();
}

void setAbdomen(){
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 17,-1.5);
        glVertex3f(-2.5, 17,0);
        glVertex3f(0, 17,1.5);
        glVertex3f(2.5, 17,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 18,-1);
        glVertex3f(-2, 18,0);
        glVertex3f(0, 18,1.5);
        glVertex3f(2, 18,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 19,-1);
        glVertex3f(-2, 19,0);
        glVertex3f(0, 19,1.5);
        glVertex3f(2, 19,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 20,-1);
        glVertex3f(-2, 20,0);
        glVertex3f(0, 20,1.5);
        glVertex3f(2, 20,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 21,-1.5);
        glVertex3f(-2.5, 21,0);
        glVertex3f(0, 21,1.5);
        glVertex3f(2.5, 21,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 17,1.5);
        glVertex3f(0, 18,1.5);
        glVertex3f(0, 20,1.5);
        glVertex3f(0, 21,1.5);
        glVertex3f(0, 21,-1.5);
        glVertex3f(0, 20,-1);
        glVertex3f(0, 18,-1);
        glVertex3f(0, 17,-1.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(-2.5,17,0);
        glVertex3f(-2,18,0);
        glVertex3f(-2,20,0);

```

```
        glVertex3f(-2.5,21,0);
        glVertex3f(2.5,21,0);
        glVertex3f(2,20,0);
        glVertex3f(2,18,0);
        glVertex3f(2.5,17,0);
    glEnd();
}

void setPelvis(){
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5, 16,1.5);
        glVertex3f(0, 14.5,0);
        glVertex3f(-1.5, 16,1.5);
        glVertex3f(-2.5, 17,0);
        glVertex3f(-1.5, 15.5,-1.8);
        glVertex3f(0, 14.5,0);
        glVertex3f(1.5, 15.5,-1.8);
        glVertex3f(2.5, 17,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5, 16,1.5);
        glVertex3f(-1.5, 16,1.5);
        glVertex3f(-1.5, 15.5,-1.8);
        glVertex3f(1.5, 15.5,-1.8);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5, 16,1.5);
        glVertex3f(0, 17,1.5);
        glVertex3f(-1.5, 16,1.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(-1.5, 15.5,-1.8);
        glVertex3f(0, 17,-1.5);
        glVertex3f(1.5, 15.5,-1.8);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 17,-1.5);
        glVertex3f(-2.5, 17,0);
        glVertex3f(0, 17,1.5);
        glVertex3f(2.5, 17,0);
    glEnd();
}

void setBrazo(void){
    glBegin(GL_LINE_LOOP);
        glVertex3f(3,18,0);
        glVertex3f(3,22,0);
        glVertex3f(2.5,23,0);
        glVertex3f(2.5,24,0);
        glVertex3f(3.75,24,0);
        glVertex3f(4.5,23,0);
        glVertex3f(4.5,22,0);
        glVertex3f(4,20.5,0);
        glVertex3f(4,18,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,18,.5);
        glVertex3f(3.5,21,.5);
        glVertex3f(3.5,22,1);
        glVertex3f(3.5,23,1);
        glVertex3f(3.5,24,.5);
        glVertex3f(3.5,24,-.5);
        glVertex3f(3.5,23,-1);
        glVertex3f(3.5,20,-1);
        glVertex3f(3.5,18,-.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3,19,0);
        glVertex3f(3.5,20,-1);
        glVertex3f(4,19,0);
        glVertex3f(3.5,19,.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3,21,0);
        glVertex3f(3.5,21,-1);
        glVertex3f(4,20.5,0);
        glVertex3f(3.5,21,.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3,22,0);
        glVertex3f(3.5,22,-1);
        glVertex3f(4,22,0);
        glVertex3f(3.5,22,1);
    glEnd();
}
```

```

    glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(2.5,23,0);
            glVertex3f(3.5,23,-1);
            glVertex3f(4.5,23,0);
            glVertex3f(3.5,23,1);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(2.5,24,0);
            glVertex3f(3.5,24,.5);
            glVertex3f(3.75,24,0);
            glVertex3f(3.5,24,-.5);
        glEnd();
    }

void setAntebrazo(){
    glBegin(GL_LINE_LOOP);
        glVertex3f(3,13,0);
        glVertex3f(3,18,0);
        glVertex3f(4,18,0);
        glVertex3f(4.5,17,0);
        glVertex3f(3.75,13,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,13,.5);
        glVertex3f(3.5,17,1);
        glVertex3f(3.5,18,.5);
        glVertex3f(3.5,18,-.5);
        glVertex3f(3.5,17,-1);
        glVertex3f(3.5,13,-.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,13,.5);
        glVertex3f(3,13,0);
        glVertex3f(3.75,13,-.5);
        glVertex3f(3.75,13,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,17,1);
        glVertex3f(3,17,0);
        glVertex3f(3.5,17,-1);
        glVertex3f(4.5,17,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,15,.75);
        glVertex3f(3,15,0);
        glVertex3f(3.5,15,-.75);
        glVertex3f(4,15,0);
    glEnd();
}

void setMano(void){
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,10.5,.5);
        glVertex3f(3.75,12,.5);
        glVertex3f(3.5,11.5,1);
        glVertex3f(3.75,12,1);
        glVertex3f(3.75,13,.5);
        glVertex3f(3.75,13,-.5);
        glVertex3f(3.75,12,-.75);
        glVertex3f(3.5,10.5,-.75);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.25,10.5,.5);
        glVertex3f(3.25,12,.5);
        glVertex3f(3.25,11.5,1);
        glVertex3f(3.25,12,1);
        glVertex3f(3,13,.5);
        glVertex3f(3,13,-.5);
        glVertex3f(3.25,12,-.75);
        glVertex3f(3.25,10.5,-.75);
    glEnd();
}

void setPierna(void)
{
    glBegin(GL_LINE_LOOP);
        glVertex3f(1,8,0);
        glVertex3f(1.5,9,0);
        glVertex3f(1.5,12,0);
        glVertex3f(0,14.5,0);
        glVertex3f(2.5,17,0);
        glVertex3f(2.75,15,0);
    glEnd();
}

```

```
        glVertex3f(2.75,14,0);
        glVertex3f(2,8,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5,8,1);
        glVertex3f(1.5,14,1.5);
        glVertex3f(1.5,16,1.5);
        glVertex3f(2.5,17,0);
        glVertex3f(1.5, 15.5, -1.8);
        glVertex3f(1.5,13.5,-1.5);
        glVertex3f(1.5,12,-1);
        glVertex3f(1.5,8,-.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0,14.5,0);
        glVertex3f(1.5,16,1.5);
        glVertex3f(1.5,15.5,-1.8);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(.5,12,0);
        glVertex3f(1.5,14,1.5);
        glVertex3f(2.75,15,0);
        glVertex3f(1.5,13.5,-1.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(.5,9,0);
        glVertex3f(1.5,12,-1);
        glVertex3f(2.75,14,0);
        glVertex3f(1.5,11,1.25);
    glEnd();
}

void setAntepierna(void){
    glBegin(GL_LINE_LOOP);
        glVertex3f(1,1,0);
        glVertex3f(.5,6,0);
        glVertex3f(1,8,0);
        glVertex3f(2,8,0);
        glVertex3f(2.5,6,0);
        glVertex3f(2.5,5,0);
        glVertex3f(2,1,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5,1,1);
        glVertex3f(1.5,8,1);
        glVertex3f(1.5,8,-.5);
        glVertex3f(1.5,6,-1);
        glVertex3f(1.5,5,-1);
        glVertex3f(1.5,1,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(.5,6,0);
        glVertex3f(1.5,6,-1);
        glVertex3f(2.5,6,0);
        glVertex3f(1.5,6,1);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(.6,5,0);
        glVertex3f(1.5,5,-1);
        glVertex3f(2.5,5,0);
        glVertex3f(1.5,5,1);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(.8,3,0);
        glVertex3f(1.5,3,-.5);
        glVertex3f(2.25,3,0);
        glVertex3f(1.5,3,1);
    glEnd();
}

void setPie(void){
    glBegin(GL_LINE_LOOP);
        glVertex3f(1,0,4);
        glVertex3f(1.5,0,4);
        glVertex3f(2.5,0,3);
        glVertex3f(2,1,1);
        glVertex3f(1,1,1);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(2,1,1);
        glVertex3f(2.5,0,3);
        glVertex3f(2,0,-.5);
        glVertex3f(2,1,0);
    glEnd();
}
```



```

glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(1,0,-.5);
    glVertex3f(2,0,-.5);
    glVertex3f(2,1,0);
    glVertex3f(1,1,0);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(1,1,0);
    glVertex3f(1,1,1);
    glVertex3f(1,0,4);
    glVertex3f(1,0,-.5);
glEnd();
}

void display(void)
{
    glClearColor (0.0, 0.20, 0.05, 0.0);
    glPointSize(4.0);
    glShadeModel (GL_FLAT);
    glClear (GL_COLOR_BUFFER_BIT);
    setRxyz();
    //MATRIZ PRINCIPAL
    glPushMatrix();
        //ESPACIO VECTORIAL PRINCIPAL
        glTranslatef(0,0,intVZoom);
        glRotatef ((GLfloat) RotEscenaX, 1.0, 0.0, 0.0);
        glRotatef ((GLfloat) RotEscenaY, 0.0, 1.0, 0.0);
        glPushMatrix();
            setMalla();
            glBegin(GL_POINTS);
                glColor3f(1,0,0);
                setCGBD();setCGABD();setCGMD();setCGPD();setCGAPD();setCGFD();
                setCGBI();setCGMI();setCGABI();setCGPI();setCGAPI();setCGFI();
                setCGTOR();setCGCAB();

            glEnd();

            setCGTOT();
            glColor3f(0.8, 0.8, 0.8);
            glTranslatef(0,0,0);
            glRotatef ((GLfloat) RotPelvisX, 1.0, 0.0, 0.0);
            glRotatef ((GLfloat) RotPelvisY, 0.0, 1.0, 0.0);
            glRotatef ((GLfloat) RotPelvisZ, 0.0, 0.0, 1.0);
            //CUERPO ENTERO 28.75 = 1.80 mts
            glPushMatrix();
                glTranslatef(0,-16,0);
                //CABEZA-CUELLO-TORZO-ABDOMEN
                glPushMatrix();
                    glTranslatef(0,16,0);
                    glRotatef ((GLfloat) RotTorzo, 1.0, 0.0, 0.0);
                    glTranslatef(0,-16,0);
                    //CABEZA
                    glPushMatrix();
                        glRotatef ((GLfloat) RotCabeza, 0.0, 1.0, 0.0);
                        setCabeza();
                    glPopMatrix();
                glPushMatrix();
                    setCuello();
                    setTorzo();
                    setAbdomen();
                glPopMatrix();//FIN DE CABEZA
                //BRAZO DERECHO
                glPushMatrix();
                    glTranslatef(0,23.5,0);
                    glRotatef ((GLfloat) RotBrazoDer, 1.0, 0.0, 0.0);
                    glTranslatef(0,-23.5,0);
                    glPushMatrix();
                        glScalef(-1,1,1);
                        setBrazo();
                    glPopMatrix();
                //ANTEBRAZO DERECHO
                glPushMatrix();
                    glTranslatef(0,18,0);
                    glRotatef ((GLfloat) RotAnteBrazoDer, 1.0, 0.0, 0.0);
                    glTranslatef(0,-18,0);
                    glPushMatrix();
                        glScalef(-1,1,1);
                        setAntebrazo();
                    glPopMatrix();
                //MANO DERECHA
                glPushMatrix();
                    glScalef(-1,1,1);
                    setMano();

```

```
        glPopMatrix();
        //FIN MANO DERECHA
    glPopMatrix();
    //FIN ANTEBRAZO DERECHO
glPopMatrix();
//FIN BRAZO DERECHO
//BRAZO IZQUIERDO
glPushMatrix();
    glTranslatef(0,23.5,0);
    glRotatef ((GLfloat) RotBrazolzq, 1.0, 0.0, 0.0);
    glTranslatef(0,-23.5,0);
    glPushMatrix();
        setBrazo();
    glPopMatrix();
    //ANTEBRAZO IZQUIERDO
glPushMatrix();
    glTranslatef(0,18,0);
    glRotatef ((GLfloat) RotAnteBrazolzq, 1.0, 0.0, 0.0);
    glTranslatef(0,-18,0);
    glPushMatrix();
        setAntebrazo();
    glPopMatrix();
    //MANO IZQUIERDA
glPushMatrix();
    setMano();
glPopMatrix();
    //FIN MANO IZQUIERDA
glPopMatrix();
//FIN ANTEBRAZO IZQUIERDO
glPopMatrix();
//FIN BRAZO IZQUIERDO
glPopMatrix();
//FIN CABEZA-CUELLO-TORZO-ABDOMEN
//PELVIS
glPushMatrix();
    setPelvis();
glPopMatrix();
//FIN PELVIS
//PIERNA DERECHA
glPushMatrix();
    glTranslatef(0,16, 0);
    glRotatef ((GLfloat) RotPiernaDer, 1.0, 0.0, 0.0);
    glTranslatef(0,-16, 0);
    glPushMatrix();
        glScalef(-1,1,1);
        setPierna();
    glPopMatrix();
    //ANTEPIERNA DERECHA
glPushMatrix();
    glTranslatef(0,8, 0);
    glRotatef ((GLfloat) RotAntePiernaDer, 1.0, 0.0, 0.0);
    glTranslatef(0,-8, 0);
    glPushMatrix();
        glScalef(-1,1,1);
        setAntepierna();
    glPopMatrix();
    //PIE DERECHO
glPushMatrix();
    glTranslatef(0,1,-.5);
    glRotatef ((GLfloat) RotPieDer, 1.0, 0.0, 0.0);
    glTranslatef(0,-1,-.5);
    glPushMatrix();
        glScalef(-1,1,1);
        setPie();
    glPopMatrix();
glPopMatrix();
    //FIN PIE DERECHO
glPopMatrix();
//FIN ANTEPIERNA DERECHA
glPopMatrix();
//FIN PIERNA DERECHA
//PIERNA IZQUIERDA
glPushMatrix();
    glTranslatef(0,16, 0);
    glRotatef ((GLfloat) RotPiernalzq, 1.0, 0.0, 0.0);
    glTranslatef(0,-16, 0);
    glPushMatrix();
        setPierna();
    glPopMatrix();
    //ANTEPIERNA IZQUIERDA
glPushMatrix();
    glTranslatef(0,8, 0);
    glRotatef ((GLfloat) RotAntePiernalzq, 1.0, 0.0, 0.0);
```

```

        glTranslatef(0,-8, 0);
        glPushMatrix();
            setAntepierna();
        glPopMatrix();
        //PIE IZQUIERDO
        glPushMatrix();
            glTranslatef(0,1,.5);
            glRotatef ((GLfloat) RotPieIzq, 1.0, 0.0, 0.0);
            glTranslatef(0,-1,-.5);
            glPushMatrix();
                setPie();
            glPopMatrix();
        glPopMatrix();
        //FIN PIE IZQUIERDO
    glPopMatrix();
    //FIN ANTEPIERNA IZQUIERDA
glPopMatrix();
//FIN PIERNA IZQUIERDA
    glPopMatrix();
    //FIN CUERPO ENTERO
glPopMatrix();
//FIN ESPACIO VECTORIAL PRINCIPAL
glPopMatrix();
//FIN MATRIZ PRINCIPAL
switch (intXPantalla){
    case 640: intXLeyendas=-7;intCoordX=-6;intCoordY=-5;intCoordZ=8;break;
    case 800: intXLeyendas=-6;intCoordX=-4;intCoordY=-5;intCoordZ=4;break;
    case 1024: intXLeyendas=-5;intCoordX=0;intCoordY=0;intCoordZ=0;break;
    case 1152: intXLeyendas=-4;intCoordX=4;intCoordY=0;intCoordZ=0;break;
    case 1280: intXLeyendas=0;intCoordX=0;intCoordY=0;intCoordZ=0;break;
    case 1400: intXLeyendas=4;intCoordX=-4;intCoordY=0;intCoordZ=0;break;
    case 1600: intXLeyendas=-3;intCoordX=3;intCoordY=0;intCoordZ=0;break;
    default: intXLeyendas=0;intCoordX=0;intCoordY=0;intCoordZ=0;break;
}
glPushMatrix();
    glTranslatef(intXLeyendas,0,0);
    setLeyendas();
glPopMatrix();
glPushMatrix();
    glTranslatef(intCoordX,intCoordY,intCoordZ);
    setCoordenadas();
glPopMatrix();
//FIN MATRIZ PRINCIPAL
glutSwapBuffers();
glFlush();
}

void reshape (int w, int h)
{
    w=intXPantalla-8;
    h=intYPantalla*.75;
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluPerspective(40.0, (GLfloat) w/(GLfloat) h, 1, 200);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslatef (0.0, 0.0, -68);
}

void keyboard (unsigned char key, int x, int y)
{
    //TECLAS DEL MOVIMIENTO DEL CUERPO
    switch (key)
    {
        //Flexión del tronco
        case 'C':
            if (RotCabeza<90) RotCabeza = (RotCabeza + 3) % 360;
            else RotCabeza = 90;

            break;
        //Extensión del tronco
        case 'c':
            if (RotCabeza>-90) RotCabeza = (RotCabeza - 3) % 360;
            else RotCabeza = -90;

            break;
        //Flexión del tronco
        case 'H':
            if (RotTorzo<135) RotTorzo = (RotTorzo + 3) % 360;
            else RotTorzo = 135;

            break;
        //Extensión del tronco
        case 'h':
            if (RotTorzo>-45) RotTorzo = (RotTorzo - 3) % 360;
    }
}

```

```
        else RotTorzo = -45;
break;
//Flexion Brazo Derecho
case 'P':
    if (RotBrazoDer>-180) RotBrazoDer = (RotBrazoDer - 3) % 360;
    else RotBrazoDer = -180;
break;
//Extension Brazo Derecho
case 'p':
    if (RotBrazoDer<50) RotBrazoDer = (RotBrazoDer + 3) % 360;
    else RotBrazoDer = 50;
break;
//Flexion AnteBrazo Derecho
case 'O':
    if(RotAnteBrazoDer>-150) RotAnteBrazoDer = (RotAnteBrazoDer - 3) % 360;
    else RotAnteBrazoDer = -150;
break;
//Extension AnteBrazo Derecho
case 'o':
    if (RotAnteBrazoDer<0) RotAnteBrazoDer = (RotAnteBrazoDer + 3) % 360;
    else RotAnteBrazoDer = 0;
break;
//Flexion Brazo Izquierdo
case 'Q':
    if (RotBrazolzq>-180) RotBrazolzq = (RotBrazolzq - 3) % 360;
    else RotBrazolzq = -180;
break;
//Extension Brazo Izquierdo
case 'q':
    if (RotBrazolzq<50) RotBrazolzq = (RotBrazolzq + 3) % 360;
    else RotBrazolzq = 50;
break;
//Flexion AnteBrazo Izquierdo
case 'W':
    if(RotAnteBrazolzq>-150) RotAnteBrazolzq = (RotAnteBrazolzq - 3) % 360;
    else RotAnteBrazolzq = -150;
break;
//Extension AnteBrazo Izquierdo
case 'w':
    if (RotAnteBrazolzq<0) RotAnteBrazolzq = (RotAnteBrazolzq + 3) % 360;
    else RotAnteBrazolzq = 0;
break;
//Flexion Pierna Derecha
case 'I':
    if (RotPiernaDer>-125) RotPiernaDer = (RotPiernaDer - 3) % 360;
    else RotPiernaDer = -125;
break;
//Extension Pierna Derecha
case 'i':
    if (RotPiernaDer<60) RotPiernaDer = (RotPiernaDer + 3) % 360;
    else RotPiernaDer = 60;
break;
//Flexion AntePierna Derecha
case 'U':
    if (RotAntePiernaDer<135) RotAntePiernaDer = (RotAntePiernaDer + 3) % 360;
    else RotAntePiernaDer = 135;
break;
//Extension AntePierna Derecha
case 'u':
    if (RotAntePiernaDer>0) RotAntePiernaDer = (RotAntePiernaDer - 3) % 360;
    else RotAntePiernaDer = 0;
break;
//Flexion Pierna Izquierda
case 'E':
    if (RotPiernalzq>-125) RotPiernalzq = (RotPiernalzq - 3) % 360;
    else RotPiernalzq = -125;
break;
//Extension Pierna Izquierda
case 'e':
    if (RotPiernalzq<60) RotPiernalzq = (RotPiernalzq + 3) % 360;
    else RotPiernalzq = 60;
break;
//Flexion AntePierna Izquierda
case 'R':
    if (RotAntePiernalzq<135) RotAntePiernalzq = (RotAntePiernalzq + 3) % 360;
    else RotAntePiernalzq = 135;
break;
//Extension AntePierna Izquierda
case 'r':
    if (RotAntePiernalzq>0) RotAntePiernalzq = (RotAntePiernalzq - 3) % 360;
    else RotAntePiernalzq = 0;
break;
//Flexion Pie Izquierdo
```

```

    case 'a':
        if (RotPieIzq<45) RotPieIzq = (RotPieIzq + 3) % 360;
        else RotPieIzq = 45;
    break;
    //Extension Pie Izquierdo
    case 'A':
        if (RotPieIzq>-20) RotPieIzq = (RotPieIzq - 3) % 360;
        else RotPieIzq = -20;
    break;
    //Flexion Pie Derecho
    case 'l':
        if (RotPieDer<45) RotPieDer = (RotPieDer + 3) % 360;
        else RotPieDer = 45;
    break;
    //Extension Pie Derecho
    case 'L':
        if (RotPieDer>-20) RotPieDer = (RotPieDer - 3) % 360;
        else RotPieDer = -20;
    break;
    ///////////////////////////////////////////////////ROTACIONES DE LA PELVIS EJES X,Y,Z////////////////////////////////////
    case 'X':
        RotPelvisX = (RotPelvisX + 3) % 360;
    break;
    case 'x':
        RotPelvisX = (RotPelvisX - 3) % 360;
    break;
    case 'Y':
        RotPelvisY = (RotPelvisY + 3) % 360;
    break;
    case 'y':
        RotPelvisY = (RotPelvisY - 3) % 360;
    break;
    case 'Z':
        RotPelvisZ = (RotPelvisZ + 3) % 360;
        break;
    case 'z':
        RotPelvisZ = (RotPelvisZ - 3) % 360;
    break;
    ///////////////////////////////////////////////////ZOOM////////////////////////////////////
    case '+':
        if (intVZoom<30) intVZoom=intVZoom+1;
        else intVZoom=30;
    break;
    case '-':
        if (intVZoom>-60) intVZoom=intVZoom-1;
        else intVZoom=-60;
    break;
    case 27:
        exit(0);
}
    glutPostRedisplay();
}

void keyboard2(int key, int x, int y)
{
    //MOVIMIENTO DEL ESPACIO VECTORIAL PRINCIPAL
    switch (key) {
        case GLUT_KEY_LEFT:
            RotEscenaY = (RotEscenaY - 1) % 360;
            break;
        case GLUT_KEY_RIGHT:
            RotEscenaY = (RotEscenaY + 1) % 360;
            break;
        case GLUT_KEY_UP:
            RotEscenaX = (RotEscenaX - 1) % 360;
            break;
        case GLUT_KEY_DOWN:
            RotEscenaX = (RotEscenaX + 1) % 360;
            break;
    }
    glutPostRedisplay();
}

//LEER-GUARDAR DATOS////////////////////////////////////
void clicks(int button, int state, int x, int y)
{
    Registro reg;
    Datos datos;
    //CLICK IZQUIERDO
    if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
    {
        reg.Guardar();
    }
}

```

```
        datos.Guardar(reg);

        MessageBox(NULL,"Se han guardado valores","Centro de Gravedad",MB_OK);
    }
    //CLICK DERECHO
    else if(button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
    {
        if(datos.Recupera(0, reg)) reg.Leer();
        MessageBox(NULL,"Se han leído valores","Centro de Gravedad",MB_OK);
    }
    glutPostRedisplay();
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(intXPantalla-8, (intYPantalla*.75));
    glutInitWindowPosition (0, 0);
    glutCreateWindow ("CALCULO DE CENTRO DE GRAVEDAD");
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMouseFunc(clicks);
    glutKeyboardFunc(keyboard);
    glutSpecialFunc(keyboard2);
    glutMainLoop();
    return 0;
}
//FIN DEL PROGRAMA CENTRO DE GRAVEDAD
```

Anexo C. Código Visual C++/OpenGL Código Fuente del Desarrollo “Cálculo de Salto Vertical/Parabólico”

```

/*
Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Acatlán
Salto Vertical y Parabólico Basquetbolístico
Cálculo de Salto Vertical/Parabólico
Tesisista:Aguilar Guillen Ariel
Fecha de creación:      Octubre 2003
Fecha de término:      Mayo 2006
*/

#include <GL/glut.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <math.h>
#include <windows.h>
#include <conio.h>
#include <iostream>
#include <fstream>
#include <iomanip>
#include <cstdlib>
#include <cstring>
using namespace std;

#define MAX 4
#define MIN 2

int intXPantalla = GetSystemMetrics(SM_CXFULLSCREEN);
int intYPantalla = GetSystemMetrics(SM_CYFULLSCREEN);
int i,j,k,n;
int lin=24, col=-40, col2=22, col3=32, col4=48, malla;
static int RotCabeza = 0, RotTorzo = 12;
static int RotBrazolzq = 27, RotAnteBrazolzq = -39, RotPiernalzq = -3, RotAntePiernalzq = 15, RotPielzq = 0;
static int RotBrazoDer = -27, RotAnteBrazoDer = -57, RotPiernaDer = -18, RotAntePiernaDer = 21, RotPieDer = 0;
static int globalx = 0, globaly = 90, globalz = 0, RotEscenaX = 0, RotEscenaY = 40;
static char etiqueta[100];
static char opc;
int intZoom=30, intXLeyendas, a, intBalon;
float floCGTmp, floCGTmp2, floEscalaMts;
float floVel, floGrad, v0, theta, g=9.8, floIncr, xx, yy, xi, yi, xii, yiii, maxX, maxY;
double Rx[MAX][MAX], Ry[MAX][MAX], Rz[MAX][MAX], RxArtic[MAX][MAX], RxSubArtic[MAX][MAX], RxSubSubArtic[MAX][MAX];
double A[MAX][MAX], B[MAX][MAX], C[MAX][MAX], D[MAX][MAX], E[MAX][MAX], P[MAX][MAX], CGTOT[MAX][MAX];
double CGCAB[MAX][MAX], CGTOR[MAX][MAX];
double CGBD[MAX][MAX], CGBI[MAX][MAX], CGABD[MAX][MAX], CGABI[MAX][MAX], CGMD[MAX][MAX], CGMI[MAX][MAX];
double CGPD[MAX][MAX], CGPI[MAX][MAX], CGAPD[MAX][MAX], CGAPI[MAX][MAX], CGFD[MAX][MAX], CGFI[MAX][MAX];
double douPCAB, douPTOR, douPB, douPAB, douPM, douPP, douPAP, douPF;
double douVCAB, douVTOR, douVB, douVAB, douVM, douVP, douVAP, douVF;

////////////////////
int test;

// Clase registro.
class Registro {
public:Registro(){}
void Guardar();
void Leer();
private: float dato[25];
};

// GUARDA VALORES ACTUALES
void Registro::Guardar() {
dato[0]=RotCabeza;
dato[1]=RotTorzo;
dato[2]=RotBrazoDer;
dato[3]=RotBrazolzq;
dato[4]=RotAnteBrazoDer;
dato[5]=RotAnteBrazolzq;
dato[6]=RotPiernaDer;
dato[7]=RotPiernalzq;
dato[8]=RotAntePiernaDer;
dato[9]=RotAntePiernalzq;
dato[10]=RotPieDer;
dato[11]=RotPielzq;
dato[12]=intZoom;
dato[13]=xi;
dato[14]=xii;
dato[15]=xiii;
dato[16]=maxX;

```

C[2] Sal to vertical y parabólico

```
    dato[17]=maxY;
    dato[18]=floVel;
    dato[19]=floGrad;
    dato[20]=RotEscenaX;
    dato[21]=RotEscenaY;
    dato[22]=intBalon;
    dato[23]=floCGTmp;
}

// LEE VALORES GUARDADOS
void Registro::Leer()
{
    RotCabeza      = dato[0];
    RotTorzo       = dato[1];
    RotBrazoDer    = dato[2];
    RotBrazoIzq    = dato[3];
    RotAnteBrazoDer= dato[4];
    RotAnteBrazoIzq= dato[5];
    RotPiernaDer   = dato[6];
    RotPiernaIzq   = dato[7];
    RotAntePiernaDer=dato[8];
    RotAntePiernaIzq=dato[9];
    RotPieDer      = dato[10];
    RotPielzq      = dato[11];
    intZoom        = dato[12];
    xi             = dato[13];
    xii            = dato[14];
    xiii           = dato[15];
    maxX          = dato[16];
    maxY          = dato[17];
    floVel         = dato[18];
    floGrad        = dato[19];
    RotEscenaX     = dato[20];
    RotEscenaY     = dato[21];
    intBalon       = dato[22];
    floCGTmp       = dato[23];
}

// CLASE DATOS, ALMACENA Y TRATA LOS DATOS
class Datos :public fstream {
public:
    Datos() : fstream("SVertParab.dat", ios::in | ios::out | ios::binary) {
        if(!good()) {
            open("SVertParab.dat", ios::in | ios::out | ios::trunc | ios::binary);
        }
        void Guardar(Registro &reg);
        bool Recupera(long n, Registro &reg);
};

void Datos::Guardar(Registro &reg) {
    clear();
    seekg(0, ios::beg);
    write(reinterpret_cast<char *> (&reg), sizeof(Registro));
}

bool Datos::Recupera(long n, Registro &reg) {
    clear();
    seekg(n*sizeof(Registro), ios::beg);
    read(reinterpret_cast<char *> (&reg), sizeof(Registro));
    return gcount() > 0;
}

////////////////////FUNCION DE ETIQUETAS////////////////////
void inline dibujarCadena (char *s)
{
    unsigned int i;
    for (i = 0; i < strlen (s); i++)
        glutBitmapCharacter (GLUT_BITMAP_HELVETICA_12, s[i]);
}

void getParamIni(){
    a = 16;
    floEscalaMts = 15.972;
    v0 = (floVel/3600*1000)*(floEscalaMts)*(0.249);
    theta = floGrad*3.1416/180;
    xx=0;yy=0;
}

void getValParabol(){
    getParamIni();
    if ((floVel>0)&&(floGrad>0)){
        for (xi=0;yi>=0;xi+=.01){
            yi=tan(theta)*xi-(g/(2*pow(v0,2)*pow(cos(theta),2)))*pow(xi,2)+a;

```



```

        if (maxY<yi){
            maxY=yi;
        }
    }
    maxX = xi;
    floIncr=maxX/200;
}

void setDibParabol(){
    getParamIni();
    if ((floVel>0)&&(floGrad>0)){
        glBegin(GL_LINE_STRIP);
        for (xx=0;yy>=0;xx+=floIncr){
            yy=tan(theta)*xx-(g/(2*pow(v0,2)*pow(cos(theta),2)))*pow(xx,2)+a;
            glVertex3f(xx+(20-(maxX/2)),yy,0);
        }
        glEnd();
    }
}

void getMovParabol(){
    getParamIni();
    if ((floVel>0.0)&&(floGrad>0)){
        yii=(tan(theta)*xii-(g/(2*pow(v0,2)*pow(cos(theta),2)))*pow(xii,2))+a;
        if (yii<a){
            yii=a;
            if(xii<0) xii=0;
            else xii-=floIncr;
        }
        else yii=a;
    }
}

void setDatos(){
    char cad1[5],cad2[2];
    int ch,ch2,j=0,pto=0;
    //PESOS DE LOS SEGMENTOS CORPORALES RELATIVOS AL PESO TOTAL DEL CUERPO [% DE 1]
    douPCAB=0.0694;
    douPTOR=0.4346;
    douPB=0.0271;
    douPAB=0.0162;
    douPM=0.0061;
    douPP=0.1416;
    douPAP=0.0433;
    douPF=0.0137;
    cout<<"UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO\n";
    cout<<"FACULTAD DE ESTUDIOS SUPERIORES ACATLAN\n";
    cout<<"MATEMATICAS APLICADAS Y COMPUTACION\n";
    cout<<"CALCULO DE SALTO VERTICAL/PARABOLICO\n";
    cout<<"TESISTA: ARIEL AGUILAR GUILLEN\n\n\n";
    cout<<"VELOCIDAD EN KM/HORA [0.00 - 40.00]: ";
    while ((ch=getche())!='\n')
    {
        cad1[j]=ch;j++;
        if (j==5) break;
        if (cad1[j]=='.') pto++;
        if (pto==2) break;
    }
    cout<<"\nINCLINACION EN GRADOS [0 - 90]: ";j=0;
    while ((ch2=getche())!='\n')
    {
        cad2[j]=ch2;j++;
        if (j==2) break;
    }
    floVel=atof(cad1);
    floGrad=atof(cad2);
    if ((floVel<0)|| (floVel>40.00)) floVel=0.0;
    if ((floGrad<0)|| (floGrad>90)) floGrad=0;
    if ((floGrad==90)) floGrad=.;//EL LIMITE TIENDE A 90 GRADOS
    getValParabol();
}

void init(void)
{
    glClearColor (0.0, 0.20, 0.05, 0.0);
    glShadeModel (GL_FLAT);
}

double GetWindowHeight(void);

void setLeyendas(void)
{

```

```

glColor3f(1,1,1);
printf (etiqueta, "UNAM.FES.ACATLAN.MAC");glRasterPos2f (col, lin-1);dibujarCadena (etiqueta);
printf (etiqueta, "CALCULO DE SALTO VERTICAL/PARABOLICO");glRasterPos2f (col, lin-3);dibujarCadena (etiqueta);
printf (etiqueta, "Tesis:ta: Aguilar Guillen Ariel");glRasterPos2f (col, lin-5);dibujarCadena (etiqueta);
glColor3f(0,1,0);
printf (etiqueta, "Cabeza:");glRasterPos2f (col, lin-9);dibujarCadena (etiqueta);
printf (etiqueta, "C:");glRasterPos2f (col+11, lin-9);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotCabeza);glRasterPos2f (col+13, lin-9);dibujarCadena (etiqueta);
printf (etiqueta, "Torzo:");glRasterPos2f (col, lin-11);dibujarCadena (etiqueta);
printf (etiqueta, "H:");glRasterPos2f (col+11,lin-11);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotTorzo);glRasterPos2f (col+13,lin-11);dibujarCadena (etiqueta);
printf (etiqueta, "Izq");glRasterPos2f (col+11, lin-14);dibujarCadena (etiqueta);
printf (etiqueta, "Der");glRasterPos2f (col+19, lin-14);dibujarCadena (etiqueta);
printf (etiqueta, "Brazo:");glRasterPos2f (col, lin-16);dibujarCadena (etiqueta);
printf (etiqueta, "Q:");glRasterPos2f (col+11, lin-16);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotBrazoIzq);glRasterPos2f (col+13, lin-16);dibujarCadena (etiqueta);
printf (etiqueta, "P:");glRasterPos2f (col+19, lin-16);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotBrazoDer);glRasterPos2f (col+21, lin-16);dibujarCadena (etiqueta);
printf (etiqueta, "AntBraz:");glRasterPos2f (col, lin-18);dibujarCadena (etiqueta);
printf (etiqueta, "W:");glRasterPos2f (col+11, lin-18);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotAnteBrazoIzq);glRasterPos2f (col+13, lin-18);dibujarCadena (etiqueta);
printf (etiqueta, "O:");glRasterPos2f (col+19, lin-18);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotAnteBrazoDer);glRasterPos2f (col+21, lin-18);dibujarCadena (etiqueta);
printf (etiqueta, "Pierna:");glRasterPos2f (col, lin-20);dibujarCadena (etiqueta);
printf (etiqueta, "E:");glRasterPos2f (col+11, lin-20);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotPiernaIzq);glRasterPos2f (col+13, lin-20);dibujarCadena (etiqueta);
printf (etiqueta, "I:");glRasterPos2f (col+19, lin-20);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotPiernaDer);glRasterPos2f (col+21, lin-20);dibujarCadena (etiqueta);
printf (etiqueta, "AntPier:");glRasterPos2f (col, lin-22);dibujarCadena (etiqueta);
printf (etiqueta, "R:");glRasterPos2f (col+11, lin-22);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotAntePiernaIzq);glRasterPos2f (col+13, lin-22);dibujarCadena (etiqueta);
printf (etiqueta, "U:");glRasterPos2f (col+19, lin-22);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotAntePiernaDer);glRasterPos2f (col+21, lin-22);dibujarCadena (etiqueta);
printf (etiqueta, "Pie:");glRasterPos2f (col, lin-24);dibujarCadena (etiqueta);
printf (etiqueta, "T:");glRasterPos2f (col+11, lin-24);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotPieIzq);glRasterPos2f (col+13, lin-24);dibujarCadena (etiqueta);
printf (etiqueta, "Y:");glRasterPos2f (col+19, lin-24);dibujarCadena (etiqueta);
printf (etiqueta, "%d",RotPieDer);glRasterPos2f (col+21, lin-24);dibujarCadena (etiqueta);
printf (etiqueta, "Zoom +/-");glRasterPos2f (col, lin-27);dibujarCadena (etiqueta);
printf (etiqueta, "%i",intZoom);glRasterPos2f (col+11, lin-27);dibujarCadena (etiqueta);
getMovParabol();
printf (etiqueta, "Saltar:");glRasterPos2f (col, lin-29);dibujarCadena (etiqueta);
printf (etiqueta, "S");glRasterPos2f (col+11, lin-29);dibujarCadena (etiqueta);
glColor3f(1,1,1);
printf (etiqueta, "Com.H:");glRasterPos2f (col, lin-32);dibujarCadena (etiqueta);
printf (etiqueta, "[x] %f",xii/floEscalaMts);glRasterPos2f (col+11, lin-32);dibujarCadena (etiqueta);
printf (etiqueta, "Com.V:");glRasterPos2f (col, lin-34);dibujarCadena (etiqueta);
printf (etiqueta, "[y] %f", (yii-16)/floEscalaMts);glRasterPos2f (col+11, lin-34);dibujarCadena (etiqueta);
printf (etiqueta, "Cons.V:");glRasterPos2f (col, lin-36);dibujarCadena (etiqueta);
printf (etiqueta, "[a] %f",a/floEscalaMts);glRasterPos2f (col+11, lin-36);dibujarCadena (etiqueta);
printf (etiqueta, "Incr.V:");glRasterPos2f (col, lin-38);dibujarCadena (etiqueta);
printf (etiqueta, "[+] %f",floCGTmp/floEscalaMts);glRasterPos2f (col+11, lin-38);dibujarCadena (etiqueta);
printf (etiqueta, "Alcan.V:");glRasterPos2f (col, lin-40);dibujarCadena (etiqueta);
printf (etiqueta, "[-] %f", (floCGTmp-CGTOT[1][0])/floEscalaMts);glRasterPos2f (col+11, lin-40);dibujarCadena (etiqueta);
printf (etiqueta, "KM/H:");glRasterPos2f (col, lin-42);dibujarCadena (etiqueta);
printf (etiqueta, "%f",floVel);glRasterPos2f (col+11, lin-42);dibujarCadena (etiqueta);
printf (etiqueta, "Grados:");glRasterPos2f (col, lin-44);dibujarCadena (etiqueta);
printf (etiqueta, "%f",ceil(floGrad));glRasterPos2f (col+11, lin-44);dibujarCadena (etiqueta);
printf (etiqueta, "Lon Max:");glRasterPos2f (col, lin-46);dibujarCadena (etiqueta);
printf (etiqueta, "%f",maxX/floEscalaMts);glRasterPos2f (col+11, lin-46);dibujarCadena (etiqueta);
printf (etiqueta, "Alt Max:");glRasterPos2f (col, lin-48);dibujarCadena (etiqueta);
printf (etiqueta, "%f",maxY/floEscalaMts);glRasterPos2f (col+11, lin-48);dibujarCadena (etiqueta);
printf (etiqueta, "Alt Salto:");glRasterPos2f (col, lin-50);dibujarCadena (etiqueta);
printf (etiqueta, "%f", (maxY/floEscalaMts)-(16/floEscalaMts));glRasterPos2f (col+11, lin-50);dibujarCadena (etiqueta);
glColor3f(0,1,0);
printf (etiqueta, "Guardar: ");glRasterPos2f (col, lin-52);dibujarCadena (etiqueta);
printf (etiqueta, "Click Izq");glRasterPos2f (col+11, lin-52);dibujarCadena (etiqueta);
printf (etiqueta, "Leer:");glRasterPos2f (col, lin-54);dibujarCadena (etiqueta);
printf (etiqueta, "Click Der");glRasterPos2f (col+11, lin-54);dibujarCadena (etiqueta);
printf (etiqueta, "BALON: ");glRasterPos2f (col, lin-56);dibujarCadena (etiqueta);
printf (etiqueta, "1/0");glRasterPos2f (col+11, lin-56);dibujarCadena (etiqueta);
printf (etiqueta, "Escen.X: ");glRasterPos2f (col, lin-58);dibujarCadena (etiqueta);
printf (etiqueta, "UP/DOWN");glRasterPos2f (col+11, lin-58);dibujarCadena (etiqueta);
printf (etiqueta, "%i",RotEscenaX);glRasterPos2f (col+26, lin-58);dibujarCadena (etiqueta);
printf (etiqueta, "Escen.Y: ");glRasterPos2f (col, lin-60);dibujarCadena (etiqueta);
printf (etiqueta, "LEFT/RIGHT");glRasterPos2f (col+11, lin-60);dibujarCadena (etiqueta);
printf (etiqueta, "%i",RotEscenaY);glRasterPos2f (col+26, lin-60);dibujarCadena (etiqueta);
glColor3f(0,1,1);
}

void setMalla(void){
glColor3f(0.,5,0);
glBegin(GL_LINES);

```

```

    for (malla=-100;malla<=100;malla=malla+10)
    {
        //MALLA PLANO XZ HORIZONTAL
        glVertex3f(-100,0,malla);
        glVertex3f(100,0,malla);
    }
    for (malla=-100;malla<=100;malla=malla+10)
    {
        //MALLA PLANO XZ VERTICAL
        glVertex3f(malla,0,-100);
        glVertex3f(malla,0,100);
    }
    glEnd();
}

```

```

void setBalon(){
    glPushMatrix();
    glColor3f(.65,.25,0);
    glScalef(2.5,2.5,2.5);
    glutWireSphere(1,10,5);
    glPopMatrix();
}

```

```

void setTablero(){
    glPushMatrix();
    glTranslatef(20,0,0);
    glPushMatrix();
    glTranslatef(14, 50.75, 0);
    glScalef(20,2,2);
    glColor3f(.5, .5, .5);
    glutWireCube(1);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(34.25, 28, 0);
    glRotatef(30,0,0,1);
    glScalef(4,52,2);
    glColor3f(.5, .5, .5);
    glutWireCube(1);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(46.25, 0, 0);
    glRotatef(45,0,1,0);
    glRotatef(-90,1,0,0);
    glColor3f(.5, .5, .5);
    glutWireCone(8,12,4,2);
    glPopMatrix();
    glPushMatrix();
    glTranslatef(3.5, 52, 0);
    glScalef(1, 12, 20);
    glColor3f(0,0,1);
    glutWireCube(1);
    glPopMatrix();
    glPushMatrix();
    glColor3f(.8,.8,.8);
    glTranslatef(0,48,0);
    glBegin(GL_LINE_LOOP);
        glVertex3f(3,0,-5);
        glVertex3f(3,5,-5);
        glVertex3f(3,5,5);
        glVertex3f(3,0,5);
    glEnd();
    glPopMatrix();
    glPushMatrix();
    glTranslatef(0, 48.75, 0);
    glRotatef(90,1,0,0);
    glColor3f(.5, .25, 0);
    glutWireTorus(.1, 3, 10, 20);
    glPopMatrix();
}

```

```
//ROTACIONES DEL CUERPO COMPLETO
```

```

void setRxyz(){
    Rx[0][0]= 1;
    Rx[1][1]= cos(3.1416*globalx/180);
    Rx[1][2]=sin(3.1416*globalx/180);
    Rx[2][1]= sin(3.1416*globalx/180);
    Rx[2][2]= cos(3.1416*globalx/180);
    Rx[3][3]= 1;

    Ry[0][0]= cos(3.1416*globaly/180);
    Ry[0][2]= sin(3.1416*globaly/180);
}

```

C[6] Sal to vertical y parabólico

```
Ry[1][1]= 1;
Ry[2][0]= -sin(3.1416*globaly/180);
Ry[2][2]= cos(3.1416*globaly/180);
Ry[3][3]= 1;

Rz[0][0]= cos(3.1416*globalz/180);
Rz[0][1]= -sin(3.1416*globalz/180);
Rz[1][0]= sin(3.1416*globalz/180);
Rz[1][1]= cos(3.1416*globalz/180);
Rz[2][2]= 1;
Rz[3][3]= 1;
}

//CG BRAZO DERECHO
void setCGBD(void){
    double P[MAX][MAX], T1[MAX][MAX];
    //POSICION CG A PARTIR DEL PRÓXIMAL (HOMBRO DER)
    P[0][0]=-3.5;
    P[1][0]= -3.17;
    P[2][0]= 0;
    P[3][0]= 1;

    RxSubArtic[0][0]= 1;
    RxSubArtic[1][1]= cos(3.1416*RotBrazoDer/180);
    RxSubArtic[1][2]= -sin(3.1416*RotBrazoDer/180);
    RxSubArtic[2][1]= sin(3.1416*RotBrazoDer/180);
    RxSubArtic[2][2]= cos(3.1416*RotBrazoDer/180);
    RxSubArtic[3][3]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotTorzo/180);
    RxArtic[1][2]= -sin(3.1416*RotTorzo/180);
    RxArtic[2][1]= sin(3.1416*RotTorzo/180);
    RxArtic[2][2]= cos(3.1416*RotTorzo/180);
    RxArtic[3][3]= 1;

    //TRASLACION
    T1[0][0]= 0;
    T1[1][0]= 7.5;
    T1[2][0]= 0;
    T1[3][0]= 1;

    for (i=0;i<=3;i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGBD[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL BRAZO
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                D[i][j]= D[i][j]+(RxSubArtic[i][k]*P[k][j]);
    //OBTIENE POSICION CG A PARTIR DEL PRÓXIMAL (HOMBRO DER)
    for (i=0;i<=3;i++)
        D[i][0]=D[i][0]+T1[i][0];
    //OBTIENE POSICION CG AL ROTAR EL TORZO
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                CGBD[i][j]= CGBD[i][j]+(Rx[i][k]*A[k][j]);

    glVertex3f(CGBD[0][0],CGBD[1][0],CGBD[2][0]);
}
```

```

//CG BRAZO IZQUIERDO
void setCGBI(void){
    double P[MAX][MAX], T1[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (HOMBRO IZQ)
    P[0][0]=3.5;
    P[1][0]=-3.17;
    P[2][0]=0;
    P[3][0]=1;

    RxSubArtic[0][0]=1;
    RxSubArtic[1][1]=cos(3.1416*RotBrazolzq/180);
    RxSubArtic[1][2]=-sin(3.1416*RotBrazolzq/180);
    RxSubArtic[2][1]=sin(3.1416*RotBrazolzq/180);
    RxSubArtic[2][2]=cos(3.1416*RotBrazolzq/180);
    RxSubArtic[3][3]=1;

    RxArtic[0][0]=1;
    RxArtic[1][1]=cos(3.1416*RotTorzo/180);
    RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
    RxArtic[2][1]=sin(3.1416*RotTorzo/180);
    RxArtic[2][2]=cos(3.1416*RotTorzo/180);
    RxArtic[3][3]=1;

    //TRASLACION
    T1[0][0]=0;
    T1[1][0]=7.5;
    T1[2][0]=0;
    T1[3][0]=1;

    for (i=0;i<=3;i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGBI[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL BRAZO
    for (i=0;i<=3;i++){
        for (j=0;j<=0;j++){
            for(k=0;k<=3;k++){
                D[i][j]=D[i][j]+(RxSubArtic[i][k]*P[k][j]);
            }
        }
        //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO IZQ)
        D[i][0]=D[i][0]+T1[i][0];
        //OBTIENE POSICION CG AL ROTAR EL TORZO
        for (i=0;i<=3;i++){
            for (j=0;j<=0;j++){
                for(k=0;k<=3;k++){
                    C[i][j]=C[i][j]+(RxArtic[i][k]*D[k][j]);
                }
            }
        }
        //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
        for (i=0;i<=3;i++){
            for (j=0;j<=0;j++){
                for(k=0;k<=3;k++){
                    B[i][j]=B[i][j]+(Rz[i][k]*C[k][j]);
                }
            }
        }
        //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
        for (i=0;i<=3;i++){
            for (j=0;j<=0;j++){
                for(k=0;k<=3;k++){
                    A[i][j]=A[i][j]+(Ry[i][k]*B[k][j]);
                }
            }
        }
        //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
        for (i=0;i<=3;i++){
            for (j=0;j<=0;j++){
                for(k=0;k<=3;k++){
                    CGBI[i][j]=CGBI[i][j]+(Rx[i][k]*A[k][j]);
                }
            }
        }

        glVertex3f(CGBI[0][0],CGBI[1][0],CGBI[2][0]);
    }
}

//CG ANTEBRAZO DERECHO
void setCGABD(){
    double P[MAX][MAX], T1[MAX][MAX], T2[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CODER)
    P[0][0]=-3.5;
    P[1][0]=-2.28;
    P[2][0]=0;
    P[3][0]=1;

    RxSubSubArtic[0][0]=1;
    RxSubSubArtic[1][1]=cos(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[1][2]=-sin(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[2][1]=sin(3.1416*RotAnteBrazoDer/180);

```

C[8] Salto vertical y parabólico

```

RxSubSubArtic[2][2]= cos(3.1416*RotAnteBrazoDer/180);
RxSubSubArtic[3][3]= 1;

RxSubArtic[0][0]= 1;
RxSubArtic[1][1]= cos(3.1416*RotBrazoDer/180);
RxSubArtic[1][2]=-sin(3.1416*RotBrazoDer/180);
RxSubArtic[2][1]= sin(3.1416*RotBrazoDer/180);
RxSubArtic[2][2]= cos(3.1416*RotBrazoDer/180);
RxSubArtic[3][3]= 1;

RxArtic[0][0]= 1;
RxArtic[1][1]= cos(3.1416*RotTorzo/180);
RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
RxArtic[2][1]= sin(3.1416*RotTorzo/180);
RxArtic[2][2]= cos(3.1416*RotTorzo/180);
RxArtic[3][3]= 1;

//TRASLACION
T1[0][0]= 0;
T1[1][0]= 7.5;
T1[2][0]= 0;
T1[3][0]= 1;

T2[0][0]= 0;
T2[1][0]= -5.5;
T2[2][0]= 0;
T2[3][0]= 1;

for (i=0;i<=3;i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGABD[i][0]=0;
}
//-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
//OBTIENE POSICION CG AL ROTAR EL ANTEBRAZO
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (CODO DER)
for (i=0;i<=3;i++)
    E[i][0]=E[i][0]+T2[i][0];
//OBTIENE POSICION CG AL ROTAR EL BRAZO
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO DER)
for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR EL TORZO
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGABD[i][j]= CGABD[i][j]+(Rx[i][k]*A[k][j]);
glVertex3f(CGABD[0][0],CGABD[1][0],CGABD[2][0]);
}

//CG ANTEBRAZO IZQUIERDO
void setCGABI(){
    double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CODO IZQ)
    P[0][0]= 3.5;
    P[1][0]= -2.28;
    P[2][0]= 0;

```

```

P[3][0]= 1;

RxSubSubArtic[0][0]= 1;
RxSubSubArtic[1][1]= cos(3.1416*RotAnteBrazolzq/180);
RxSubSubArtic[1][2]=-sin(3.1416*RotAnteBrazolzq/180);
RxSubSubArtic[2][1]= sin(3.1416*RotAnteBrazolzq/180);
RxSubSubArtic[2][2]= cos(3.1416*RotAnteBrazolzq/180);
RxSubSubArtic[3][3]= 1;

RxSubArtic[0][0]= 1;
RxSubArtic[1][1]= cos(3.1416*RotBrazolzq/180);
RxSubArtic[1][2]=-sin(3.1416*RotBrazolzq/180);
RxSubArtic[2][1]= sin(3.1416*RotBrazolzq/180);
RxSubArtic[2][2]= cos(3.1416*RotBrazolzq/180);
RxSubArtic[3][3]= 1;

RxArtic[0][0]= 1;
RxArtic[1][1]= cos(3.1416*RotTorzo/180);
RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
RxArtic[2][1]= sin(3.1416*RotTorzo/180);
RxArtic[2][2]= cos(3.1416*RotTorzo/180);
RxArtic[3][3]= 1;

//TRASLACION
T1[0][0]= 0;
T1[1][0]= 7.5;
T1[2][0]= 0;
T1[3][0]= 1;

T2[0][0]= 0;
T2[1][0]= -5.5;
T2[2][0]= 0;
T2[3][0]= 1;

for (i=0;i<=3; i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGABI[i][0]=0;
}
//-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
//OBTIENE POSICION CG AL ROTAR EL ANTEBRAZO
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (CODO IZQ)
for (i=0;i<=3;i++)
    E[i][0]=E[i][0]+T2[i][0];
//OBTIENE POSICION CG AL ROTAR EL BRAZO
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO IZQ)
for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR EL TORZO
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGABI[i][j]= CGABI[i][j]+(Rx[i][k]*A[k][j]);

glVertex3f(CGABI[0][0],CGABI[1][0],CGABI[2][0]);
}

```

C[10] Sal to vertical y parabólico

```
//CG MANO DERECHA
void setCGMD(){
    double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CODO DER)
    P[0][0]=-3.5;
    P[1][0]=-5.52;
    P[2][0]=0;
    P[3][0]=1;

    RxSubSubArtic[0][0]=1;
    RxSubSubArtic[1][1]=cos(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[1][2]=-sin(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[2][1]=sin(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[2][2]=cos(3.1416*RotAnteBrazoDer/180);
    RxSubSubArtic[3][3]=1;

    RxSubArtic[0][0]=1;
    RxSubArtic[1][1]=cos(3.1416*RotBrazoDer/180);
    RxSubArtic[1][2]=-sin(3.1416*RotBrazoDer/180);
    RxSubArtic[2][1]=sin(3.1416*RotBrazoDer/180);
    RxSubArtic[2][2]=cos(3.1416*RotBrazoDer/180);
    RxSubArtic[3][3]=1;

    RxArtic[0][0]=1;
    RxArtic[1][1]=cos(3.1416*RotTorzo/180);
    RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
    RxArtic[2][1]=sin(3.1416*RotTorzo/180);
    RxArtic[2][2]=cos(3.1416*RotTorzo/180);
    RxArtic[3][3]=1;

    //TRASLACION
    T1[0][0]=0;
    T1[1][0]=7.5;
    T1[2][0]=0;
    T1[3][0]=1;

    T2[0][0]=0;
    T2[1][0]=-5.5;
    T2[2][0]=0;
    T2[3][0]=1;

    for (i=0;i<=3;i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGMD[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL ANTEBRAZO
    for (i=0;i<=3;i++){
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                E[i][j]=E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
    }
    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (CODO DER)
    for (i=0;i<=3;i++){
        E[i][0]=E[i][0]+T2[i][0];
    }
    //OBTIENE POSICION CG AL ROTAR EL BRAZO
    for (i=0;i<=3;i++){
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                D[i][j]=D[i][j]+(RxSubArtic[i][k]*E[k][j]);
    }
    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO DER)
    for (i=0;i<=3;i++){
        D[i][0]=D[i][0]+T1[i][0];
    }
    //OBTIENE POSICION CG AL ROTAR EL TORZO
    for (i=0;i<=3;i++){
        for (j=0;j<=0;j++)
            for (k=0;k<=3;k++)
                C[i][j]=C[i][j]+(RxArtic[i][k]*D[k][j]);
    }
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
    for (i=0;i<=3;i++){
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                B[i][j]=B[i][j]+(Rz[i][k]*C[k][j]);
    }
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
    for (i=0;i<=3;i++){
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                A[i][j]=A[i][j]+(Ry[i][k]*B[k][j]);
    }
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)

```



```

for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGMD[i][j]= CGMD[i][j]+(Rx[i][k]*A[k][j]);

glVertex3f(CGMD[0][0],CGMD[1][0],CGMD[2][0]);
}

//CG MANO IZQUIERDA
void setCGMI(){
    double P[MAX][MAX], T1[MAX][MAX], T2[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CODO IZQ)
    P[0][0]= 3.5;
    P[1][0]= -5.52;
    P[2][0]= 0;
    P[3][0]= 1;

    RxSubSubArtic[0][0]= 1;
    RxSubSubArtic[1][1]= cos(3.1416*RotAnteBrazolzq/180);
    RxSubSubArtic[1][2]=-sin(3.1416*RotAnteBrazolzq/180);
    RxSubSubArtic[2][1]= sin(3.1416*RotAnteBrazolzq/180);
    RxSubSubArtic[2][2]= cos(3.1416*RotAnteBrazolzq/180);
    RxSubSubArtic[3][3]= 1;

    RxSubArtic[0][0]= 1;
    RxSubArtic[1][1]= cos(3.1416*RotBrazolzq/180);
    RxSubArtic[1][2]=-sin(3.1416*RotBrazolzq/180);
    RxSubArtic[2][1]= sin(3.1416*RotBrazolzq/180);
    RxSubArtic[2][2]= cos(3.1416*RotBrazolzq/180);
    RxSubArtic[3][3]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotTorzo/180);
    RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
    RxArtic[2][1]= sin(3.1416*RotTorzo/180);
    RxArtic[2][2]= cos(3.1416*RotTorzo/180);
    RxArtic[3][3]= 1;

    //TRASLACION
    T1[0][0]= 0;
    T1[1][0]= 7.5;
    T1[2][0]= 0;
    T1[3][0]= 1;

    T2[0][0]= 0;
    T2[1][0]= -5.5;
    T2[2][0]= 0;
    T2[3][0]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGMI[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL ANTEBRAZO
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (CODO IZQ)
    for (i=0;i<=3;i++)
        E[i][0]=E[i][0]+T2[i][0];
    //OBTIENE POSICION CG AL ROTAR EL BRAZO
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (HOMBRO IZQ)
    for (i=0;i<=3;i++)
        D[i][0]=D[i][0]+T1[i][0];
    //OBTIENE POSICION CG AL ROTAR EL TORZO
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)

```

C[12] Sal to vertical y parabólico

```

        B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGM[i][j]= CGM[i][j]+(Rx[i][k]*A[k][j]);
glVertex3f(CGM[0][0],CGM[1][0],CGM[2][0]);
}

//CG PIERNA DERECHA
void setCGPD(){
    double P[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CADERA DER)
    P[0][0]= -1.5;
    P[1][0]= -3.27;
    P[2][0]= 0;
    P[3][0]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernaDer/180);
    RxArtic[1][2]=-sin(3.1416*RotPiernaDer/180);
    RxArtic[2][1]= sin(3.1416*RotPiernaDer/180);
    RxArtic[2][2]= cos(3.1416*RotPiernaDer/180);
    RxArtic[3][3]= 1;

    for (i=0;i<=3; i++){
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGPD[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR LA PIERNA
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                B[i][j]= B[i][j]+(RxArtic[i][k]*P[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                C[i][j]= C[i][j]+(Rz[i][k]*B[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                D[i][j]= D[i][j]+(Ry[i][k]*C[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                CGPD[i][j]= CGPD[i][j]+(Rx[i][k]*D[k][j]);
    glVertex3f(CGPD[0][0],CGPD[1][0],CGPD[2][0]);
}

//CG PIERNA IZQUIERDA
void setCGPI(){
    double P[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (CADERA IZQ)
    P[0][0]=1.5;
    P[1][0]= -3.27;
    P[2][0]= 0;
    P[3][0]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);
    RxArtic[1][2]=-sin(3.1416*RotPiernalzq/180);
    RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);
    RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);
    RxArtic[3][3]= 1;

    for (i=0;i<=3; i++){
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        E[i][0]=0;
        CGPI[i][0]=0;
    }

```

```

}
//-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
//OBTIENE POSICION CG AL ROTAR LA PIERNA
for (i=0;i<=3;i++)
    or (j=0;j<=0;j++)
        r(k=0;k<=3;k++)
            [j]= B[i][j]+(RxArtic[i][k]*P[k][j]);
/OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    or (j=0;j<=0;j++)
        r(k=0;k<=3;k++)
            [j]= C[i][j]+(Rz[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    or (j=0;j<=0;j++)
        r(k=0;k<=3;k++)
            [j]= D[i][j]+(Ry[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    or (j=0;j<=0;j++)
        r(k=0;k<=3;k++)
            [i][j]= CGPI[i][j]+(Rx[i][k]*D[k][j]);
glVertex3f(CGPI[0][0],CGPI[1][0],CGPI[2][0]);
}

//CG ANTEPIERNA DERECHA
void setCGAPD(){
    double P[MAX][MAX],T1[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (RODILLA DER)
    P[0][0]= -1.5;
    P[1][0]= -3.12;
    P[2][0]= 0;
    P[3][0]= 1;

    RxSubArtic[0][0]= 1;
    RxSubArtic[1][1]= cos(3.1416*RotAntePiernaDer/180);
    RxSubArtic[1][2]=-sin(3.1416*RotAntePiernaDer/180);
    RxSubArtic[2][1]= sin(3.1416*RotAntePiernaDer/180);
    RxSubArtic[2][2]= cos(3.1416*RotAntePiernaDer/180);
    RxSubArtic[3][3]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotPiernaDer/180);
    RxArtic[1][2]=-sin(3.1416*RotPiernaDer/180);
    RxArtic[2][1]= sin(3.1416*RotPiernaDer/180);
    RxArtic[2][2]= cos(3.1416*RotPiernaDer/180);
    RxArtic[3][3]= 1;

    //TRASLACION
    T1[0][0]= 0;
    T1[1][0]= -8;
    T1[2][0]= 0;
    T1[3][0]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        D[i][0]=0;
        CGAPD[i][0]=0;
    }
}
//-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
//OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            [i][j]= D[i][j]+(RxSubArtic[i][k]*P[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA DER)
for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR LA PIERNA
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)

```

C[14] Sal to vertical y parabólico

```

        for(k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGAPD[i][j]= CGAPD[i][j]+(Rx[i][k]*A[k][j]);
glVertex3f(CGAPD[0][0],CGAPD[1][0],CGAPD[2][0]);
}

//CG ANTEPIERNA IZQUIERDA
void setCGAPL(){
    double P[MAX][MAX],T1[MAX][MAX];
        //POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)
P[0][0]= 1.5;
P[1][0]= -3.12;
P[2][0]= 0;
P[3][0]= 1;

RxSubArtic[0][0]= 1;
RxSubArtic[1][1]= cos(3.1416*RotAntePiernalzq/180);
RxSubArtic[1][2]=-sin(3.1416*RotAntePiernalzq/180);
RxSubArtic[2][1]= sin(3.1416*RotAntePiernalzq/180);
RxSubArtic[2][2]= cos(3.1416*RotAntePiernalzq/180);
RxSubArtic[3][3]= 1;

RxArtic[0][0]= 1;
RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);
RxArtic[1][2]=-sin(3.1416*RotPiernalzq/180);
RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);
RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);
RxArtic[3][3]= 1;

//TRASLACION
T1[0][0]= 0;
T1[1][0]= -8;
T1[2][0]= 0;
T1[3][0]= 1;

for (i=0;i<=3; i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    CGAPL[i][0]=0;
}
//-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
//OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            D[i][j]= D[i][j]+(RxSubArtic[i][k]*P[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)
for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR LA PIERNA
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGAPL[i][j]= CGAPL[i][j]+(Rx[i][k]*A[k][j]);
glVertex3f(CGAPL[0][0],CGAPL[1][0],CGAPL[2][0]);
}

//CG PIE DERECHO
void setCGFD(){
    double P[MAX][MAX],T1[MAX][MAX],T2[MAX][MAX];
        //POSICION CG A PARTIR DEL PROXIMAL (TOBILLO DER)
P[0][0]= 0;

```

```

P[1][0]= -0.5;
P[2][0]= 0.54;
P[3][0]= 1;

RxSubSubArtic[0][0]= 1;
RxSubSubArtic[1][1]= cos(3.1416*RotPieDer/180);
RxSubSubArtic[1][2]=-sin(3.1416*RotPieDer/180);
RxSubSubArtic[2][1]= sin(3.1416*RotPieDer/180);
RxSubSubArtic[2][2]= cos(3.1416*RotPieDer/180);
RxSubSubArtic[3][3]= 1;

RxSubArtic[0][0]= 1;
RxSubArtic[1][1]= cos(3.1416*RotAntePiernaDer/180);
RxSubArtic[1][2]=-sin(3.1416*RotAntePiernaDer/180);
RxSubArtic[2][1]= sin(3.1416*RotAntePiernaDer/180);
RxSubArtic[2][2]= cos(3.1416*RotAntePiernaDer/180);
RxSubArtic[3][3]= 1;

RxArtic[0][0]= 1;
RxArtic[1][1]= cos(3.1416*RotPiernaDer/180);
RxArtic[1][2]=-sin(3.1416*RotPiernaDer/180);
RxArtic[2][1]= sin(3.1416*RotPiernaDer/180);
RxArtic[2][2]= cos(3.1416*RotPiernaDer/180);
RxArtic[3][3]= 1;

//TRASLACION
T1[0][0]= 0;
T1[1][0]= -8;
T1[2][0]= 0;
T1[3][0]= 1;

T2[0][0]= -1.5;
T2[1][0]= -7;
T2[2][0]= .5;
T2[3][0]= 1;

for (i=0;i<=3;i++){
    A[i][0]=0;
    B[i][0]=0;
    C[i][0]=0;
    D[i][0]=0;
    E[i][0]=0;
    CGFD[i][0]=0;
}
//-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
//OBTIENE POSICION CG AL ROTAR EL PIE
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (TOBILLO DER)
for (i=0;i<=3;i++)
    E[i][0]=E[i][0]+T2[i][0];
//OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);
//OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA DER)
for (i=0;i<=3;i++)
    D[i][0]=D[i][0]+T1[i][0];
//OBTIENE POSICION CG AL ROTAR LA PIERNA
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGFD[i][j]= CGFD[i][j]+(Rx[i][k]*A[k][j]);

glVertex3f(CGFD[0][0],CGFD[1][0],CGFD[2][0]);

```

C[16] Sal to vertical y parabólico

```
}  
  
//CG PIE IZQUIERDO  
void setCGF(){  
    double P[MAX][MAX], T1[MAX][MAX], T2[MAX][MAX];  
    //POSICION CG A PARTIR DEL PROXIMAL (TOBILLO IZQ)  
    P[0][0]= 0;  
    P[1][0]= -0.5;  
    P[2][0]= 0.54;  
    P[3][0]= 1;  
  
    RxSubSubArtic[0][0]= 1;  
    RxSubSubArtic[1][1]= cos(3.1416*RotPielzq/180);  
    RxSubSubArtic[1][2]=-sin(3.1416*RotPielzq/180);  
    RxSubSubArtic[2][1]= sin(3.1416*RotPielzq/180);  
    RxSubSubArtic[2][2]= cos(3.1416*RotPielzq/180);  
    RxSubSubArtic[3][3]= 1;  
  
    RxSubArtic[0][0]= 1;  
    RxSubArtic[1][1]= cos(3.1416*RotAntePiernalzq/180);  
    RxSubArtic[1][2]=-sin(3.1416*RotAntePiernalzq/180);  
    RxSubArtic[2][1]= sin(3.1416*RotAntePiernalzq/180);  
    RxSubArtic[2][2]= cos(3.1416*RotAntePiernalzq/180);  
    RxSubArtic[3][3]= 1;  
  
    RxArtic[0][0]= 1;  
    RxArtic[1][1]= cos(3.1416*RotPiernalzq/180);  
    RxArtic[1][2]=-sin(3.1416*RotPiernalzq/180);  
    RxArtic[2][1]= sin(3.1416*RotPiernalzq/180);  
    RxArtic[2][2]= cos(3.1416*RotPiernalzq/180);  
    RxArtic[3][3]= 1;  
  
    //TRASLACION  
    T1[0][0]= 0;  
    T1[1][0]= -8;  
    T1[2][0]= 0;  
    T1[3][0]= 1;  
  
    T2[0][0]= 1.5;  
    T2[1][0]= -7;  
    T2[2][0]= .5;  
    T2[3][0]= 1;  
  
    for (i=0;i<=3;i++){  
        A[i][0]=0;  
        B[i][0]=0;  
        C[i][0]=0;  
        D[i][0]=0;  
        E[i][0]=0;  
        CGF[i][0]=0;  
    }  
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//  
    //OBTIENE POSICION CG AL ROTAR EL PIE  
    for (i=0;i<=3;i++)  
        for (j=0;j<=0;j++)  
            for(k=0;k<=3;k++)  
                E[i][j]= E[i][j]+(RxSubSubArtic[i][k]*P[k][j]);  
    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (TOBILLO IZQ)  
    for (i=0;i<=3;i++)  
        E[i][0]=E[i][0]+T2[i][0];  
    //OBTIENE POSICION CG AL ROTAR LA ANTEPIERNA  
    for (i=0;i<=3;i++)  
        for (j=0;j<=0;j++)  
            for(k=0;k<=3;k++)  
                D[i][j]= D[i][j]+(RxSubArtic[i][k]*E[k][j]);  
    //OBTIENE POSICION CG A PARTIR DEL PROXIMAL (RODILLA IZQ)  
    for (i=0;i<=3;i++)  
        D[i][0]=D[i][0]+T1[i][0];  
    //OBTIENE POSICION CG AL ROTAR LA PIERNA  
    for (i=0;i<=3;i++)  
        for (j=0;j<=0;j++)  
            for(k=0;k<=3;k++)  
                C[i][j]= C[i][j]+(RxArtic[i][k]*D[k][j]);  
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)  
    for (i=0;i<=3;i++)  
        for (j=0;j<=0;j++)  
            for(k=0;k<=3;k++)  
                B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);  
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)  
    for (i=0;i<=3;i++)  
        for (j=0;j<=0;j++)  
            for(k=0;k<=3;k++)  
                A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);  
}
```

```

//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGF[i][j]= CGF[i][j]+(Rx[i][k]*A[k][j]);

glVertex3f(CGF[0][0],CGF[1][0],CGF[2][0]);
}

//CG TORZO
void setCGTOR(void){
    double P[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (ARTICULACION LUMBAR)
    P[0][0]= 0;
    P[1][0]= 3.81;
    P[2][0]= 0;
    P[3][0]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotTorzo/180);
    RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
    RxArtic[2][1]= sin(3.1416*RotTorzo/180);
    RxArtic[2][2]= cos(3.1416*RotTorzo/180);
    RxArtic[3][3]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        CGTOR[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL TORZO
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                C[i][j]= C[i][j]+(RxArtic[i][k]*P[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
    //OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                CGTOR[i][j]= CGTOR[i][j]+(Rx[i][k]*A[k][j]);
    glVertex3f(CGTOR[0][0],CGTOR[1][0],CGTOR[2][0]);
}

//CG CABEZA
void setCGCAB(void){
    double P[MAX][MAX];
    //POSICION CG A PARTIR DEL PROXIMAL (ARTICULACION LUMBAR)
    P[0][0]= 0;
    P[1][0]= 11.03;
    P[2][0]= 0;
    P[3][0]= 1;

    RxArtic[0][0]= 1;
    RxArtic[1][1]= cos(3.1416*RotTorzo/180);
    RxArtic[1][2]=-sin(3.1416*RotTorzo/180);
    RxArtic[2][1]= sin(3.1416*RotTorzo/180);
    RxArtic[2][2]= cos(3.1416*RotTorzo/180);
    RxArtic[3][3]= 1;

    for (i=0;i<=3; i++){
        A[i][0]=0;
        B[i][0]=0;
        C[i][0]=0;
        CGCAB[i][0]=0;
    }
    //-----MULTIPLICACION DE MATRICES DE ROTACION Y TRASLACION-----//
    //OBTIENE POSICION CG AL ROTAR EL TORZO
    for (i=0;i<=3;i++)
        for (j=0;j<=0;j++)
            for(k=0;k<=3;k++)
                C[i][j]= C[i][j]+(RxArtic[i][k]*P[k][j]);

```

C[18] Sal to vertical y parabólico

```
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Z)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            B[i][j]= B[i][j]+(Rz[i][k]*C[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE Y)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            A[i][j]= A[i][j]+(Ry[i][k]*B[k][j]);
//OBTIENE POSICION CG AL ROTAR LA PELVIS (EJE X)
for (i=0;i<=3;i++)
    for (j=0;j<=0;j++)
        for(k=0;k<=3;k++)
            CGCAB[i][j]= CGCAB[i][j]+(Rx[i][k]*A[k][j]);
glVertex3f(CGCAB[0][0],CGCAB[1][0],CGCAB[2][0]);
}

//CG TOTAL
void setCGTOT(void){
    glColor3f(1,0,0);
    for (i=0;i<=3;i++){
        CGTOT[i][0] =(CGCAB[i][0]*douPCAB)+(CGTOR[i][0]*douPTOR)
        +(CGBD[i][0]*douPB)+(CGBI[i][0]*douPB)
        +(CGABD[i][0]*douPAB)+(CGABI[i][0]*douPAB)
        +(CGMD[i][0]*douPM)+(CGMI[i][0]*douPM)
        +(CGPD[i][0]*douPP)+(CGPI[i][0]*douPP)
        +(CGAPD[i][0]*douPAP)+(CGAPI[i][0]*douPAP)
        +(CGFD[i][0]*douPF)+(CGFI[i][0]*douPF);
    }
    glBegin(GL_POINTS);
    glVertex3f(CGTOT[0][0],CGTOT[1][0],CGTOT[2][0]);
    glEnd();
    glBegin(GL_LINES);
    //EJE X
    glVertex3f(CGTOT[0][0]-1,CGTOT[1][0],CGTOT[2][0]);
    glVertex3f(CGTOT[0][0]+1,CGTOT[1][0],CGTOT[2][0]);
    //EJE Y
    glVertex3f(CGTOT[0][0],CGTOT[1][0]-1,CGTOT[2][0]);
    glVertex3f(CGTOT[0][0],CGTOT[1][0]+1,CGTOT[2][0]);
    //EJE Z
    glVertex3f(CGTOT[0][0],CGTOT[1][0],CGTOT[2][0]-1);
    glVertex3f(CGTOT[0][0],CGTOT[1][0],CGTOT[2][0]+1);
    glEnd();
}

void setCuello(){
    glBegin(GL_LINE_LOOP);
    glVertex3f(0, 24,1.5);
    glVertex3f(2.5, 24,0);
    glVertex3f(0, 24,-1.5);
    glVertex3f(-2.5, 24,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex3f(0, 24.5, 1);
    glVertex3f(1.25, 24.5, 0);
    glVertex3f(0, 24.5, -1);
    glVertex3f(-1.25, 24.5, 0);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex3f(0, 26, .75);
    glVertex3f(.75, 26,0);
    glVertex3f(0, 26, -.75);
    glVertex3f(-.75, 26, 0);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex3f(0, 24, 1.5);
    glVertex3f(0, 24.5, 1);
    glVertex3f(0, 26, .75);
    glVertex3f(0, 26, -.75);
    glVertex3f(0, 24.5, -1);
    glVertex3f(0, 24,-1.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex3f(2.5, 24,0);
    glVertex3f(1.25, 24.5,0);
    glVertex3f(.75, 26,0);
    glVertex3f(-.75, 26,0);
    glVertex3f(-1.25, 24.5,0);
    glVertex3f(-2.5, 24,0);
    glEnd();
}
```



```

void setCabeza(){
    glTranslatef(0,27, 0);
    glScalef(1.5,1.75,1.5);
    glRotatef(90,1,0,0);
    glutWireSphere(1,7,5);
}

void setTorzo(){
    glPushMatrix();
        glScalef(1,1,-1);
        glBegin(GL_LINE_LOOP);
            glVertex3f(0, 21,1.5);
            glVertex3f(2.5, 21,0);
            glVertex3f(0, 21,-1.5);
            glVertex3f(-2.5, 21,0);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(-1.5, 22,2);
            glVertex3f(1.5, 22,2);
            glVertex3f(2.75, 22,1);
            glVertex3f(2.75, 22,-.5);
            glVertex3f(1.5, 22,-1.5);
            glVertex3f(-1.5, 22,-1.5);
            glVertex3f(-2.75, 22,-.5);
            glVertex3f(-2.75, 22,1);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(-1.5, 23,2);
            glVertex3f(1.5, 23,2);
            glVertex3f(2.75, 23,1);
            glVertex3f(2.75, 23,-.5);
            glVertex3f(1.5, 23,-1.5);
            glVertex3f(-1.5, 23,-1.5);
            glVertex3f(-2.75, 23,-.5);
            glVertex3f(-2.75, 23,1);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(0, 24,1.5);
            glVertex3f(2.5, 24,0);
            glVertex3f(0, 24,-1.5);
            glVertex3f(-2.5, 24,0);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(0, 21, 1.5);
            glVertex3f(1.5, 22,2);
            glVertex3f(1.5, 23,2);
            glVertex3f(0, 24,1.5);
            glVertex3f(-1.5, 23,2);
            glVertex3f(-1.5, 22,2);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(2.5, 21,0);
            glVertex3f(2.75, 22,1);
            glVertex3f(2.75, 23,1);
            glVertex3f(2.5, 24,0);
            glVertex3f(2.75, 23,-.5);
            glVertex3f(2.75, 22,-.5);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(0, 21,-1.5);
            glVertex3f(1.5, 22,-1.5);
            glVertex3f(1.5, 23,-1.5);
            glVertex3f(0, 24,-1.5);
            glVertex3f(-1.5, 23,-1.5);
            glVertex3f(-1.5, 22,-1.5);
        glEnd();
        glBegin(GL_LINE_LOOP);
            glVertex3f(-2.5, 21,0);
            glVertex3f(-2.75, 22,1);
            glVertex3f(-2.75, 23,1);
            glVertex3f(-2.5, 24,0);
            glVertex3f(-2.75, 23,-.5);
            glVertex3f(-2.75, 22,-.5);
        glEnd();
    glEnd();
    glPopMatrix();
}

void setAbdomen(){
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 17,-1.5);
        glVertex3f(-2.5, 17,0);
        glVertex3f(0, 17,1.5);
        glVertex3f(2.5, 17,0);
    glEnd();
}

```

```
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(0, 18,-1);
    glVertex3f(-2, 18,0);
    glVertex3f(0, 18,1.5);
    glVertex3f(2, 18,0);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(0, 19,-1);
    glVertex3f(-2, 19,0);
    glVertex3f(0, 19,1.5);
    glVertex3f(2, 19,0);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(0, 20,-1);
    glVertex3f(-2, 20,0);
    glVertex3f(0, 20,1.5);
    glVertex3f(2, 20,0);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(0, 21,-1.5);
    glVertex3f(-2.5, 21,0);
    glVertex3f(0, 21,1.5);
    glVertex3f(2.5, 21,0);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(0, 17,1.5);
    glVertex3f(0, 18,1.5);
    glVertex3f(0, 20,1.5);
    glVertex3f(0, 21,1.5);
    glVertex3f(0, 21,-1.5);
    glVertex3f(0, 20,-1);
    glVertex3f(0, 18,-1);
    glVertex3f(0, 17,-1.5);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(-2.5,17,0);
    glVertex3f(-2,18,0);
    glVertex3f(-2,20,0);
    glVertex3f(-2.5,21,0);
    glVertex3f(2.5,21,0);
    glVertex3f(2,20,0);
    glVertex3f(2,18,0);
    glVertex3f(2.5,17,0);
glEnd();
}

void setPelvis(){
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5, 16,1.5);
        glVertex3f(0, 14.5,0);
        glVertex3f(-1.5, 16,1.5);
        glVertex3f(-2.5, 17,0);
        glVertex3f(-1.5, 15.5,-1.8);
        glVertex3f(0, 14.5,0);
        glVertex3f(1.5, 15.5,-1.8);
        glVertex3f(2.5, 17,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5, 16,1.5);
        glVertex3f(-1.5, 16,1.5);
        glVertex3f(-1.5, 15.5,-1.8);
        glVertex3f(1.5, 15.5,-1.8);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5, 16,1.5);
        glVertex3f(0, 17,1.5);
        glVertex3f(-1.5, 16,1.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(-1.5, 15.5,-1.8);
        glVertex3f(0, 17,-1.5);
        glVertex3f(1.5, 15.5,-1.8);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(0, 17,-1.5);
        glVertex3f(-2.5, 17,0);
        glVertex3f(0, 17,1.5);
        glVertex3f(2.5, 17,0);
    glEnd();
}
```

```
void setBrazo(void){
```

```

glBegin(GL_LINE_LOOP);
    glVertex3f(3,18,0);
    glVertex3f(3,22,0);
    glVertex3f(2.5,23,0);
    glVertex3f(2.5,24,0);
    glVertex3f(3.75,24,0);
    glVertex3f(4.5,23,0);
    glVertex3f(4.5,22,0);
    glVertex3f(4.20,5,0);
    glVertex3f(4,18,0);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(3.5,18,.5);
    glVertex3f(3.5,21,.5);
    glVertex3f(3.5,22,1);
    glVertex3f(3.5,23,1);
    glVertex3f(3.5,24,.5);
    glVertex3f(3.5,24,-.5);
    glVertex3f(3.5,23,-1);
    glVertex3f(3.5,20,-1);
    glVertex3f(3.5,18,-.5);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(3,19,0);
    glVertex3f(3.5,20,-1);
    glVertex3f(4,19,0);
    glVertex3f(3.5,19,.5);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(3,21,0);
    glVertex3f(3.5,21,-1);
    glVertex3f(4,20,5,0);
    glVertex3f(3.5,21,.5);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(3,22,0);
    glVertex3f(3.5,22,-1);
    glVertex3f(4.5,22,0);
    glVertex3f(3.5,22,1);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(2.5,23,0);
    glVertex3f(3.5,23,-1);
    glVertex3f(4.5,23,0);
    glVertex3f(3.5,23,1);
glEnd();
glBegin(GL_LINE_LOOP);
    glVertex3f(2.5,24,0);
    glVertex3f(3.5,24,.5);
    glVertex3f(3.75,24,0);
    glVertex3f(3.5,24,-.5);
glEnd();
}

void setAntebrazo(){
    glBegin(GL_LINE_LOOP);
        glVertex3f(3,13,0);
        glVertex3f(3,18,0);
        glVertex3f(4,18,0);
        glVertex3f(4.5,17,0);
        glVertex3f(3.75,13,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,13,.5);
        glVertex3f(3.5,17,1);
        glVertex3f(3.5,18,.5);
        glVertex3f(3.5,18,-.5);
        glVertex3f(3.5,17,-1);
        glVertex3f(3.5,13,-.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,13,.5);
        glVertex3f(3,13,0);
        glVertex3f(3.75,13,-.5);
        glVertex3f(3.75,13,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5,17,1);
        glVertex3f(3,17,0);
        glVertex3f(3.5,17,-1);
        glVertex3f(4.5,17,0);
    glEnd();
    glBegin(GL_LINE_LOOP);

```

```
        glVertex3f(3.5, 15, .75);
        glVertex3f(3,15,0);
        glVertex3f(3.5, 15, -.75);
        glVertex3f(4,15,0);
    glEnd();
}

void setMano(void){
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.5, 10.5, .5);
        glVertex3f(3.75, 12, .5);
        glVertex3f(3.5, 11.5, 1);
        glVertex3f(3.75, 12, 1);
        glVertex3f(3.75, 13, .5);
        glVertex3f(3.75, 13, -.5);
        glVertex3f(3.75, 12, -.75);
        glVertex3f(3.5, 10.5, -.75);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(3.25, 10.5, .5);
        glVertex3f(3.25, 12, .5);
        glVertex3f(3.25, 11.5, 1);
        glVertex3f(3.25, 12, 1);
        glVertex3f(3,13, .5);
        glVertex3f(3,13, -.5);
        glVertex3f(3.25, 12, -.75);
        glVertex3f(3.25, 10.5, -.75);
    glEnd();
}

void setPierna(void)
{
    glBegin(GL_LINE_LOOP);
    glVertex3f(1,8,0);
    glVertex3f(.5,9,0);
    glVertex3f(.5,12,0);
    glVertex3f(0,14.5,0);
    glVertex3f(2.5,17,0);
    glVertex3f(2.75,15,0);
    glVertex3f(2.75,14,0);
    glVertex3f(2,8,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex3f(1.5,8,1);
    glVertex3f(1.5,14,1.5);
    glVertex3f(1.5,16,1.5);
    glVertex3f(2.5,17,0);
    glVertex3f(1.5, 15.5, -1.8);
    glVertex3f(1.5,13.5,-1.5);
    glVertex3f(1.5,12,-1);
    glVertex3f(1.5,8,-.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex3f(0,14.5,0);
    glVertex3f(1.5,16,1.5);
    glVertex3f(1.5,15.5,-1.8);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex3f(.5,12,0);
    glVertex3f(1.5,14,1.5);
    glVertex3f(2.75,15,0);
    glVertex3f(1.5,13.5,-1.5);
    glEnd();
    glBegin(GL_LINE_LOOP);
    glVertex3f(.5,9,0);
    glVertex3f(1.5,12,-1);
    glVertex3f(2.75,14,0);
    glVertex3f(1.5,11,1.25);
    glEnd();
}

void setAntepierna(void){
    glBegin(GL_LINE_LOOP);
        glVertex3f(1,1,0);
        glVertex3f(.5,6,0);
        glVertex3f(1,8,0);
        glVertex3f(2,8,0);
        glVertex3f(2.5,6,0);
        glVertex3f(2.5,5,0);
        glVertex3f(2,1,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1.5,1,1);
```

```

        glVertex3f(1.5,8,1);
        glVertex3f(1.5,8,-.5);
        glVertex3f(1.5,6,-1);
        glVertex3f(1.5,5,-1);
        glVertex3f(1.5,1,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(.5,6,0);
        glVertex3f(1.5,6,-1);
        glVertex3f(2.5,6,0);
        glVertex3f(1.5,6,1);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(.6,5,0);
        glVertex3f(1.5,5,-1);
        glVertex3f(2.5,5,0);
        glVertex3f(1.5,5,1);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(.8,3,0);
        glVertex3f(1.5,3,-.5);
        glVertex3f(2.25,3,0);
        glVertex3f(1.5,3,1);
    glEnd();
}

void setPie(void){
    glBegin(GL_LINE_LOOP);
        glVertex3f(1,0,4);
        glVertex3f(1.5,0,4);
        glVertex3f(2.5,0,3);
        glVertex3f(2,1,1);
        glVertex3f(1,1,1);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(2,1,1);
        glVertex3f(2.5,0,3);
        glVertex3f(2,0,-.5);
        glVertex3f(2,1,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1,0,-.5);
        glVertex3f(2,0,-.5);
        glVertex3f(2,1,0);
        glVertex3f(1,1,0);
    glEnd();
    glBegin(GL_LINE_LOOP);
        glVertex3f(1,1,0);
        glVertex3f(1,1,1);
        glVertex3f(1,0,4);
        glVertex3f(1,0,-.5);
    glEnd();
}

void display(void)
{
    glPointSize(4.0);
    glClear (GL_COLOR_BUFFER_BIT);
    setRxyz();
    //MATRIZ PRINCIPAL
    glPushMatrix();
    //ESPACIO VECTORIAL PRINCIPAL
    glTranslatef(0,0,intZoom);
    glRotatf ((GLfloat) RotEscenaX, 1.0, 0.0, 0.0);
    glRotatf ((GLfloat) RotEscenaY, 0.0, 1.0, 0.0);
    ///////////////////////////////////////////////////////////////////
    glLineWidth(1);
    setMalla();
    glLineWidth(2);
    setTablero();
    glPushMatrix();
        glEnable(GL_LINE_STIPPLE);
        glPushMatrix();
            glColor3f(1,1,1);
            glLineStipple(1,0x00FF);
            setDibParabol();
        glPopMatrix();
        glPushMatrix();
            if (xii==0) {
                glTranslatef(0,CGTOT[1][0],0);
                floCGTmp=CGTOT[1][0];
            }
    }
}

```

```

        else glTranslatef(0,floCGTmp,0);
        glColor3f(0,1,0);
        glLineStipple(2,0xAAAA);
        setDibParabol();
    glPopMatrix();
    glDisable(GL_LINE_STIPPLE);
glPopMatrix();
glLineWidth(1);
glColor3f(.9,.9,.9);
glPushMatrix();
    if ((floVel>0)&&(floGrad>0)) xiii = xii+(20-(maxX/2));
    getMovParabol();
    glPushMatrix();
        glTranslatef(-20,16,0);
        //Dibuja centro de gravedad de cada segmento
        glBegin(GL_POINTS);
            setCGBD();setCGABD();setCGMD();setCGPD();setCGAPD();setCGFD();
            setCGBI();setCGMI();setCGABI();setCGPI();setCGAPI();setCGFI();
            setCGTOR();setCGCAB();
        glEnd();
        //Dibuja centro de gravedad principal
        setCGTOT();
    glPopMatrix();

    if (xii==0) glTranslatef(xiii,yii,0);
    else glTranslatef(xiii,yii-CGTOT[1][0]+floCGTmp,0);

    //////////////////////////////////////
    glPointSize(5.0);
    glLineWidth(2);
    glBegin(GL_POINTS);
        glVertex3f(0,CGTOT[1][0],0);
    glEnd();
    glBegin(GL_LINES);
        //EJE X
        glVertex3f(-1.5,CGTOT[1][0],0);
        glVertex3f(1.5,CGTOT[1][0],0);
        //EJE Y
        glVertex3f(0,CGTOT[1][0]-1.5,0);
        glVertex3f(0,CGTOT[1][0]+1.5,0);
        //EJE Z
        glVertex3f(0,CGTOT[1][0],-1.5);
        glVertex3f(0,CGTOT[1][0],1.5);
    glEnd();
    glPointSize(4.0);
    glLineWidth(1);
    //////////////////////////////////////
    glColor3f(0.5, 0.5, 0.5);
    glRotatef ((GLfloat) 90, 0.0, 1.0, 0.0);
    //CUERPO ENTERO 28.75 = 1.80 mts
    glPushMatrix();
        //glTranslatef(0,-16,0);
        glTranslatef(0,-16,0);
        //CABEZA-CUELLO-TORZO-ABDOMEN
        glPushMatrix();
            glTranslatef(0,16,0);
            glRotatef ((GLfloat) RotTorzo, 1.0, 0.0, 0.0);
            glTranslatef(0,-16,0);
            glPushMatrix();
                glRotatef ((GLfloat) RotCabeza, 0.0, 1.0, 0.0);
                setCabeza();
            glPopMatrix();
            glPushMatrix();
                setCuello();
                setTorzo();
                setAbdomen();
            glPopMatrix();
            //BRAZO DERECHO
            glPushMatrix();
                glTranslatef(0,23.5,0);
                glRotatef ((GLfloat) RotBrazoDer, 1.0, 0.0, 0.0);
                glTranslatef(0,-23.5,0);
                glPushMatrix();
                    glScalef(-1,1,1);
                    setBrazo();
                glPopMatrix();
                glPushMatrix();
                    //ANTEBRAZO DERECHO
                    glPushMatrix();
                        glTranslatef(0,18,0);
                        glRotatef ((GLfloat) RotAnteBrazoDer, 1.0, 0.0,0.0);
                        glTranslatef(0,-18,0);
                        glPushMatrix();
                            glScalef(-1,1,1);

```

```

        setAntebrazo();
        glPopMatrix();
        //MANO DERECHA
        glPushMatrix();
        glScalef(-1,1,1);
        setMano();
        if (intBalon==1){
            glPushMatrix();
            glTranslatef(1,10,0);
            setBalon();
            glColor3f(0.5, 0.5, 0.5);
            glPopMatrix();
        }
        glPopMatrix();
        //FIN MANO DERECHA
        glPopMatrix();
        //FIN ANTEBRAZO DERECHO
        glPopMatrix();
        //FIN BRAZO DERECHO
        //BRAZO IZQUIERDO
        glPushMatrix();
        glTranslatef(0,23.5,0);
        glRotatef ((GLfloat) RotBrazolzq, 1.0, 0.0, 0.0);
        glTranslatef(0,-23.5,0);
        glPushMatrix();
        setBrazo();
        glPopMatrix();
        //ANTEBRAZO IZQUIERDO
        glPushMatrix();
        glTranslatef(0,18,0);
        glRotatef ((GLfloat) RotAnteBrazolzq, 1.0, 0.0,0.0);
        glTranslatef(0,-18,0);
        glPushMatrix();
        setAntebrazo();
        glPopMatrix();
        //MANO IZQUIERDA
        glPushMatrix();
        setMano();
        glPopMatrix();
        //FIN MANO IZQUIERDA
        glPopMatrix();
        //FIN ANTEBRAZO IZQUIERDO
        glPopMatrix();
        //FIN BRAZO IZQUIERDO
        glPopMatrix();
        //FIN CABEZA-CUELLO-TORZO-ABDOMEN
        //PELVIS
        glPushMatrix();
        setPelvis();
        glPopMatrix();
        //FIN PELVIS
        //PIERNA DERECHA
        glPushMatrix();
        glTranslatef(0,16, 0);
        glRotatef ((GLfloat) RotPiernaDer, 1.0, 0.0, 0.0);
        glTranslatef(0,-16, 0);
        glPushMatrix();
        glScalef(-1,1,1);
        setPierna();
        glPopMatrix();
        //ANTEPIERNA DERECHA
        glPushMatrix();
        glTranslatef(0,8, 0);
        glRotatef ((GLfloat) RotAntePiernaDer, 1.0, 0.0, 0.0);
        glTranslatef(0,-8, 0);
        glPushMatrix();
        glScalef(-1,1,1);
        setAntepierna();
        glPopMatrix();
        //PIE DERECHO
        glPushMatrix();
        glTranslatef(0,1,-.5);
        glRotatef ((GLfloat) RotPieDer, 1.0, 0.0, 0.0);
        glTranslatef(0,-1,-.5);
        glPushMatrix();
        glScalef(-1,1,1);
        setPie();
        glPopMatrix();
        glPopMatrix();
        //FIN PIE DERECHO
        glPopMatrix();
        //FIN ANTEPIERNA DERECHA
        glPopMatrix();

```

```

//FIN PIERNA DERECHA
//PIERNA IZQUIERDA
glPushMatrix();
    glTranslatef(0,16, 0);
    glRotatef ((GLfloat) RotPiernalzq, 1.0, 0.0, 0.0);
    glTranslatef(0,-16, 0);
    glPushMatrix();
        setPierna();
    glPopMatrix();
//ANTEPIERNA IZQUIERDA
glPushMatrix();
    glTranslatef(0,8, 0);
        glRotatef ((GLfloat) RotAntePiernalzq, 1.0, 0.0, 0.0);
        glTranslatef(0,-8, 0);
        glPushMatrix();
            setAntepierna();
        glPopMatrix();
//PIE IZQUIERDO
glPushMatrix();
    glTranslatef(0,1,-.5);
    glRotatef ((GLfloat) RotPieIzq, 1.0, 0.0, 0.0);
    glTranslatef(0,-1,-.5);
    glPushMatrix();
        setPie();
    glPopMatrix();
    glPopMatrix();
//FIN PIE IZQUIERDO
glPopMatrix();
//FIN ANTEPIERNA IZQUIERDA
glPopMatrix();
//FIN PIERNA IZQUIERDA
glPopMatrix();
//FIN CUERPO ENTERO
glPopMatrix();
//FIN ESPACIO VECTORIAL PRINCIPAL
glPushMatrix();
    glTranslatef(60,3,20);
    setBalon();
glPopMatrix();
if (intBalon==0){
    glPushMatrix();
        glTranslatef(60,3,30);
        setBalon();
    glPopMatrix();
}
glPopMatrix();
//ADAPTACIONES AL TIPO DE RESOLUCION
switch (intXPantalla){
    case 640: intXLeyendas=-22;          break;
    case 800: intXLeyendas=-21;          break;
    case 1024: intXLeyendas=-20;         break;
    case 1152: intXLeyendas=-19;         break;
    case 1280: intXLeyendas=-15;         break;
    case 1400: intXLeyendas=-19;         break;
    case 1600: intXLeyendas=-18;         break;
    default: intXLeyendas=-15;          break;
}
glPushMatrix();
    glTranslatef(intXLeyendas,27,50);
    setLeyendas();
glPopMatrix();
//FIN MATRIZ PRINCIPAL
glutSwapBuffers();
glFlush();
}

void reshape (int w, int h)
{
    w=intXPantalla-8;
    h=intYPantalla*.75;
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluPerspective(40.0, (GLfloat) w/(GLfloat) h, 1, 300);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    glTranslatef (0, -20, -140);
}

void keyboard (unsigned char key, int x, int y)
{
    //TECLAS DEL MOVIMIENTO DEL CUERPO
    switch (key)

```



```

{
//Flexión del tronco
case 'C':
    if (RotCabeza<90) RotCabeza = (RotCabeza + 3) % 360;
    else RotCabeza = 90;
break;
//Extensión del tronco
case 'c':
    if (RotCabeza>90) RotCabeza = (RotCabeza - 3) % 360;
    else RotCabeza = -90;
break;
//Flexión del tronco
case 'H':
    if (RotTorzo<135) RotTorzo = (RotTorzo + 3) % 360;
    else RotTorzo = 135;
break;
//Extensión del tronco
case 'h':
    if (RotTorzo>-45) RotTorzo = (RotTorzo - 3) % 360;
    else RotTorzo = -45;
break;
//Flexion Brazo Derecho
case 'P':
    if (RotBrazoDer>-180) RotBrazoDer = (RotBrazoDer - 3) % 360;
    else RotBrazoDer = -180;
break;
//Extension Brazo Derecho
case 'p':
    if (RotBrazoDer<50) RotBrazoDer = (RotBrazoDer + 3) % 360;
    else RotBrazoDer = 50;
break;
//Flexion AnteBrazo Derecho
case 'O':
    if (RotAnteBrazoDer>-150) RotAnteBrazoDer = (RotAnteBrazoDer - 3) % 360;
    else RotAnteBrazoDer = -150;
break;
//Extension AnteBrazo Derecho
case 'o':
    if (RotAnteBrazoDer<0) RotAnteBrazoDer = (RotAnteBrazoDer + 3) % 360;
    else RotAnteBrazoDer = 0;
break;
//Flexion Brazo Izquierdo
case 'Q':
    if (RotBrazolzq>-180) RotBrazolzq = (RotBrazolzq - 3) % 360;
    else RotBrazolzq = -180;
break;
//Extension Brazo Izquierdo
case 'q':
    if (RotBrazolzq<50) RotBrazolzq = (RotBrazolzq + 3) % 360;
    else RotBrazolzq = 50;
break;
//Flexion AnteBrazo Izquierdo
case 'W':
    if (RotAnteBrazolzq>-150) RotAnteBrazolzq = (RotAnteBrazolzq - 3) % 360;
    else RotAnteBrazolzq = -150;
break;
//Extension AnteBrazo Izquierdo
case 'w':
    if (RotAnteBrazolzq<0) RotAnteBrazolzq = (RotAnteBrazolzq + 3) % 360;
    else RotAnteBrazolzq = 0;
break;
//Flexion Pierna Derecha
case 'I':
    if (RotPiernaDer>-125) RotPiernaDer = (RotPiernaDer - 3) % 360;
    else RotPiernaDer = -125;
break;
//Extension Pierna Derecha
case 'i':
    if (RotPiernaDer<60) RotPiernaDer = (RotPiernaDer + 3) % 360;
    else RotPiernaDer = 60;
break;
//Flexion AntePierna Derecha
case 'U':
    if (RotAntePiernaDer<135) RotAntePiernaDer = (RotAntePiernaDer + 3) % 360;
    else RotAntePiernaDer = 135;
break;
//Extension AntePierna Derecha
case 'u':
    if (RotAntePiernaDer>0) RotAntePiernaDer = (RotAntePiernaDer - 3) % 360;
    else RotAntePiernaDer = 0;
break;
//Flexion Pierna Izquierda
case 'E':

```

```

        if (RotPiernalzq>-125) RotPiernalzq = (RotPiernalzq - 3) % 360;
        else RotPiernalzq = -125;
    break;
    //Extension Pierna Izquierda
    case 'e':
        if (RotPiernalzq<60) RotPiernalzq = (RotPiernalzq + 3) % 360;
        else RotPiernalzq = 60;
    break;
    //Flexion AntePierna Izquierda
    case 'R':
        if (RotAntePiernalzq<135) RotAntePiernalzq = (RotAntePiernalzq + 3) % 360;
        else RotAntePiernalzq = 135;
    break;
    //Extension AntePierna Izquierda
    case 'r':
        if (RotAntePiernalzq>0) RotAntePiernalzq = (RotAntePiernalzq - 3) % 360;
        else RotAntePiernalzq = 0;
    break;
    //Flexion Pie Izquierdo
    case 't':
        if (RotPielzq<45) RotPielzq = (RotPielzq + 3) % 360;
        else RotPielzq = 45;
    break;
    //Extension Pie Izquierdo
    case 'T':
        if (RotPielzq>-20) RotPielzq = (RotPielzq - 3) % 360;
        else RotPielzq = -20;
    break;
    //Flexion Pie Derecho
    case 'y':
        if (RotPieDer<45) RotPieDer = (RotPieDer + 3) % 360;
        else RotPieDer = 45;
    break;
    //Extension Pie Derecho
    case 'Y':
        if (RotPieDer>-20) RotPieDer = (RotPieDer - 3) % 360;
        else RotPieDer = -20;
    break;
    //////////////////////////////////////MOVIMIENTO PARABOLICO////////////////////////////////////
    case 'S':
        if ((floVel>0)&&(floGrad>0)){
            if ((xii<maxX)&&(yii>=16)) xii += floIncr;
            else xii=maxX;
        }
    break;
    case 's':
        if ((floVel>0)&&(floGrad>0)){
            if ((xii>0)&&(yii>=16)) xii -= floIncr;
            else xii=0;
        }
    break;
    //////////////////////////////////////ZOOM////////////////////////////////////
    case '+'://80 -80
        if (intZoom<80) intZoom+=1;
        else intZoom=80;
    break;
    case '-':
        if (intZoom>80) intZoom-=1;
        else intZoom=-80;
    break;
    //////////////////////////////////////BALON////////////////////////////////////
    case '1':
        intBalon=1;
    break;
    case '0':
        intBalon=0;
    break;
    case 27:
        exit(0);
    }
    glutPostRedisplay();
}

void keyboard2(int key, int x, int y)
{
    //MOVIMIENTO DEL ESPACIO VECTORIAL PRINCIPAL
    switch (key) {
    case GLUT_KEY_LEFT:
        RotEscenaY = (RotEscenaY - 1) % 360;
        break;
    case GLUT_KEY_RIGHT:
        RotEscenaY = (RotEscenaY + 1) % 360;
        break;
    }
}

```

```
case GLUT_KEY_UP:
RotEscenaX = (RotEscenaX - 1) % 360;
break;
case GLUT_KEY_DOWN:
RotEscenaX = (RotEscenaX + 1) % 360;
break;
}
glutPostRedisplay();
}

//LEER-GUARDAR DATOS////////////////////////////////////
void clicks(int button, int state, int x, int y)
{
    Registro reg;
    Datos datos;
    //CLICK IZQUIERDO
    if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
    {
        reg.Guardar();
        datos.Guardar(reg);
        MessageBox(NULL,"Se han guardado valores","Salto Parabólico",MB_OK);
    }
    //CLICK DERECHO
    else if(button == GLUT_RIGHT_BUTTON && state == GLUT_DOWN)
    {
        if(datos.Recupera(0, reg)) reg.Leer();
        getParamIni();
        MessageBox(NULL,"Se han leído valores","Salto Parabólico",MB_OK);
    }
    glutPostRedisplay();
}

////////////////////////////////////

int main(int argc, char** argv)
{
    setDatos();
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize (intXPantalla-8, (intYPantalla*.75));
    glutInitWindowPosition (0, 0);
    glutCreateWindow ("CALCULO DE SALTO VERTICAL/PARABOLICO");
    init();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMouseFunc(clicks);
    glutKeyboardFunc(keyboard);
    glutSpecialFunc(keyboard2);
    glutMainLoop();
    return 0;
}
//FIN DEL PROGRAMA SALTO VERTICAL/PARABOLICO
```

Anexo D. Sistema coordinado del cuerpo humano

Para la extracción del modelo del cuerpo humano, se usa un DVD (Digital Video Disc) o un scanner para extraer las imágenes de los saltadores en basketbol, luego se procede a imitar



1. Obtención de imagen escaneada o capturada del DVD



2. Recorte de imagen, agregando fondo negro con Paint Brush 5.1



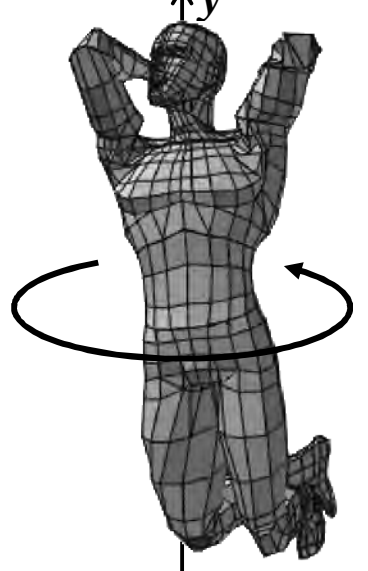
3. Sustitución de fondo negro a transparente con Photo Editor 3.01.



4. Diseño en Poser 5, de la posición de la imagen en 3D.



5. Cambio de textura con luz y sombra en Poser 5.



6. Rotación y traslación del sistema coordinado para su análisis.