



UNIVERSIDAD DE
SOTAVENTO, A.C.



ESTUDIOS INCORPORADOS A LA UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

FACULTAD DE INFORMATICA

“RECREACION DEL MODULO DE PAGOS DEL SISTEMA DE CONTROL
ESCOLAR DE LA UINIVERSIDAD ISTMO AMERICANA, A.C”

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN INFORMATICA

PRESENTA:

MARIO EMMANUEL COBOS HERNÁNDEZ

ASESOR DE TESIS

LIC. JUAN JOSÉ GUTIERREZ QUIROZ

COATZACOALCOS, VER.

ENERO 2008



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.

Con cariño:

Para mis maestros por abrirme el camino del conocimiento y darme las bases para ser un buen profesionalista

A mi padre por motivarme siempre a superarme y darme su apoyo a lo largo de toda mi carrera.

A mi madre por siempre creer en mi e inculcarme valores bien cimentados y útiles en la sociedad.

A mi hermana Karina por sus consejos y su apoyo en todos los momentos que lo he necesitado.

A mi hermana Nayeli que a lo largo de toda mi vida me ha motivado a ser ejemplo de superación.

A Dios por haberme concedido estas humildes alas para acariciar un triunfo más y alcanzar las alturas con serenidad ecuánime.

Y a todos aquellos que me acompañaron a lo largo de todo este proceso, mis mas infinitos agradecimientos por hacer de mi una persona ávida de conocimientos.

INDICE

ESTRUCTURA DE LA TESIS

CAPITULO 1 INTRODUCCION

CAPITULO 2 PLANTEAMIENTO DEL PROBLEMA

- 2,1 JUSTIFICACION
- 2,2 HIPOTESIS
- 2,3 OBJETIVO GENERAL
- 2,4 OBJETIVOS ESPECIFICOS
- 2,5 ALCANCES Y LIMITACIONES

CAPITULO 3 MARCO TEORICO

- 3,1 REGISTRO ALUMNOS
 - 3.1.1 ALUMNOS DE REINGRESO
 - 3.1.2 PROCEDIMIENTO DE INSCRIPCION
 - 3.1.3 OTORGAMIENTO DE BECAS
 - 3.1.4 PAGO DE SERVICIOS EDUCATIVOS
- 3,2 PROGRAMACION EN JAVA
 - 3.2.1 ¿QUE ES JAVA?
 - 3.2.2 ORIGENES DE JAVA
 - 3.2.3 CARACTERISTICAS DE JAVA
 - 3.2.4 JAVA Y BASES DE DATOS
- 3,3 ¿QUÉ ES MySQL?
 - 3.3.1 HISTORIA DE MySQL
 - 3.3.2 CARACTERISTICAS DE MySQL

CAPITULO 4 DISEÑO DEL PROGRAMA PROPUESTO

- DEFINICION DE LAS TABLAS DE LA BASES DE
- 4,1 DATOS
 - 4.1.1 TABLA DE PAGOS
 - 4.1.2 TABLA DE

RECIBOS
TABLA
CONCEPTOS

4.1.3

4,2 CAPTURA DE PANTALLAS DEL SIAU

4,3 ADMINISTRACION DEL SISTEMA

CONCLUSIONES

BIBLIOGRAFIA

ESTRUCTURA DE LA TESIS

Nos toca vivir una época en la que cada vez la informática invade más campos de nuestra vida, estando el ciudadano medio cada vez más familiarizado con términos del mundo informático, entre ellos, como no, los lenguajes de programación. Los cuales nos ayudan a la creación de sistemas de información que faciliten el uso y almacenamiento de información dentro de las organizaciones.

Día con día la informática contribuye mas al alcance de logros que permiten a las organizaciones la automatización de muchos de los procesos que llevan a cabo cotidianamente, logrando con esto la reducción de tiempo en la realización de todos estos.

A lo largo de la presente, se muestra como ha presentado un avance significativo la informática en el campo de la educación. A continuación se expone una de las necesidades que se crean de manera especial para llevar a cabo las operaciones de la Universidad Istmo Americana.

El Capitulo 1 corresponde la introducción del Tema, donde se describe la historia de la Universidad Istmo Americana, sus objetivos, la misión, la visión y su oferta académica.

En el Capítulo 2 se realiza el Planteamiento del Problema, comprendiendo la justificación, objetivos tanto generales como específicos, así como los alcances y limitaciones de la presente investigación.

En el Capitulo 3 se presentan los lineamientos y requisitos necesarios para llevar a cabo las operaciones en la propia Universidad, así como los aspectos más relevantes del lenguaje de programación empleado para dar solución al problema.

El capítulo 4 expone el diseño del proceso que deberá seguirse para la realización de las operaciones y solución del problema, por otro lado se especifica la forma en que deberá operar el sistema, mediante una demostración de las pantallas, así como la función de los usuarios del sistema.

Al finalizar se presentan las conclusiones a las que se llegaron después de haber terminado el presente trabajo de investigación.

En general, lo que se presenta a continuación es una alternativa para el mejoramiento de las funciones administrativas de la Universidad Istmo Americana, valiéndose de las herramientas informáticas, específicamente de los lenguajes de programación.

|

CAPITULO 1

INTRODUCCIÓN

La Universidad Istmo Americana, A.C. tiene como funciones sustantivas la docencia, la investigación, la difusión de la cultura y la extensión de los servicios, mismos que están en permanente vinculación con los diferentes sectores sociales.

Bajo la concepción dinámica que articula congruentemente el objetivo y la visión de la institución esta se orienta a las acciones a seguir en el proceso educativo, para asegurar el cumplimiento de la misión en un afán de mejorar y medir las interacciones con fines descriptivos y predictivos.

El Modelo educativo esta articulado como un sistema que interactúan con un propósito común a partir de un proceso central denominado “proceso educativo” y que se apoyará con cinco procesos estratégicos que son: el académico, planeación, administración de recursos, vinculación y difusión de la cultura y el de innovación y calidad, en el que de manera fundamental gira entorno del ser humano y de su aprendizaje, desde la perspectiva de la construcción del conocimiento en sus diversas formas.

1.1 Principios filosóficos.

Los principios que norman la vida de la Universidad Istmo Americana, A.C., emanan de conformidad a la reglamentación interna y sustentada en el Artículo 3º de la Constitución Política de los Estados Unidos Mexicanos, y de la Ley General de Educación en su Artículo Séptimo, en la que de acuerdo con éstos, la educación en todos sus tipos y modalidades, debe basarse en los resultados de progreso científico, ser ajena a los doctrinas religiosas, ser democrática y contribuir al logro de un sistema de vida que busque el constante mejoramiento económico, social y cultural, promoviendo contenidos ecológicos que permitan la generación de conocimientos estratégicos acerca de la naturaleza, su aprovechamiento y conservación.

La Ley General de Educación en su artículo séptimo señala que los esfuerzos educativos deben promover el máximo desarrollo de las capacidades humanas, la observación, análisis y pensamiento crítico, fortalecer la conciencia de nacionalidad y soberanía así como el uso de un idioma común a todos los mexicanos; propiciar el conocimiento y práctica de la democracia y enfatizar el valor de la justicia, fomentar la

investigación y la innovación científica y tecnológica, la difusión de los bienes y valores culturales, la solidaridad, el respeto a la libertad del individuo, el rechazo a los vicios y la protección del ambiente.

En consecuencia el Modelo Educativo de la Universidad Istmo Americana, A.C., se enfoca en el ser humano y todas sus estrategias educativas se centran en el aprendizaje. Además de formar profesionistas competentes, el Modelo propicia la orientación de los proyectos de vida hacia la búsqueda de la autorrealización, en un escenario de cultivo permanente del humanismo, con fundamento en las consideraciones expuestas la Universidad establece los siguientes como principios filosóficos de su Modelo Educativo:

- a) El ser humano como persona es capaz de trascender; con identidad propia, pero al mismo tiempo interdependiente; un ser histórico, creador de su cultura, su lenguaje, consciente y por ende responsable de sus acciones, es un ser constante transformador y siempre en la búsqueda del conocimiento, la justicia, la felicidad y de su autorrealización.
- b) La educación integral como un proceso continuo de desarrollo de todas las potencialidades del ser humano, lo orienta hacia la búsqueda de su realización, en el aprender ser, aprender a hacer, aprender a aprender, aprender a emprender y aprender a convivir, para que se convierta en un ciudadano consciente, responsable y solidario.
- c) La formación del ser humano comprende el desarrollo del espíritu, a través de la cultura; del intelecto, mediante la vida académica; de los sentimientos y emociones, por la convivencia y la vida artística, de la integridad física, por medio del deporte y la orientación para la salud y de la vida social mediante actividades cívicas.
- d) El ser humano como actor fundamental del proceso educativo. En su formación se promueve el aprendizaje significativo mediante la labor del docente como facilitador, a través de la reflexión y la participación apoyados con los instrumentos didácticos y tecnología de vanguardia, y

asegurando la calidad, actualización y pertinencia del aprendizaje. Su compromiso será con el desarrollo sostenido e integral del país y como miembro de la comunidad de la Universidad Istmo Americana, A.C., adoptará la filosofía y deberá estar consciente del privilegio y la responsabilidad que significa su ingreso, permanencia y egreso de la institución.

- e) La Universidad Istmo Americana, A.C., promueve, reconoce, respeta y gestiona las aportaciones intelectuales originales, producto de los diversos ámbitos del trabajo profesional.
- f) Todos los recursos de la Universidad Istmo Americana, A.C., se organizan para su ejercicio eficiente y eficaz con una clara pertinencia para el Proceso Educativo, bajo un esquema de mejora continua, todo ello, en un ambiente sociolaboral responsable, respetuosa y solidario.
- g) Los cuerpos colegiados y el trabajo en equipo favorecen la sinergia y potencian la obtención de resultados de gran alcance y calidad, por lo que se fomentan en el Proceso Educativo, y se promueve el establecimiento de redes de colaboración entre departamentos.
- h) La Universidad Istmo Americana, A.C., reconoce en sus funciones sustantivas vías importantes para el crecimiento humano y el desarrollo social; de manera que su personal, estudiantes y egresados conformen un valioso capital humano, con plena capacidad de comprender críticamente, con tolerancia, apertura y flexibilidad a las expresiones actuales del pensamiento universal, además de generar y aplicar conocimientos.
- i) La Universidad Istmo Americana, A.C., concibe al liderazgo como la capacidad para integrarse en la conducción visionaria, participativa y comprometida con los procesos de innovación, calidad y desarrollo, por lo que promueve y lo impulsa.

1.2 Código de Ética y Valores.

Considerando que la actualidad se caracteriza por un nuevo renacimiento de la humanidad, en el que se dan las condiciones para que el ser humano se reconstituya en su integridad histórica y social, bajo esta perspectiva basamos la convicción de que es posible construir esquemas y espacios de convivencia humana, en los que coincidan las bondades del avance científico y tecnológico, y del pensamiento social.

Los valores que guían el actual renacimiento son los que históricamente han sido base de los grandes logros de la sociedad, por lo que la Universidad Istmo Americana, A.C., los hace suyos y los incorpora a su modelo educativo, estos valores son:

a) Respeto a la persona

La persona, en toda su dignidad es el fin y razón de la Universidad Istmo Americana, A.C., por lo tanto el proceso educativo se orienta a que los seres humanos, con el ejercicio de su libertad, responsabilidad e inteligencia, y con la fuerza de sus sentimientos y actitudes que los lleven a la plenitud de sus potencialidad en beneficio propio y de la sociedad a la que pertenecen.

b) Respeto a la diversidad

La Universidad Istmo Americana, A.C., ofrece con equidad sus servicios educativos a la sociedad en general sin distinción de ninguna índole y desarrolla, entre otros, programas y estrategias para estudiantes que se encuentran en situaciones especiales

c) Desarrollo sostenible

El desarrollo social que se promueve desde los diversos espacios de la Universidad Istmo Americana, A.C., se dirige al respeto y cuidado a los recursos naturales además de propiciar acciones que responsabilicen del cuidado que corresponda a su generación y a generaciones futuras.

d) Formación integral

La Universidad Istmo Americana, A.C., promueve el desarrollo armónico de todas las potencialidades del ser humano, incluidas todas sus formas de pensar.

e) Responsabilidad y colaboración

Los retos que enfrentan las instituciones y las personas exigen un replanteamiento de la práctica educativa donde la participación responsable de todos los actores y el equilibrio dinámico entre el trabajo individual y el colaborativo son propósitos explícitos en los procesos formativos.

f) Creatividad e innovación

Los problemas inéditos, su complejidad y los escenarios cambiantes, que no se agotan en un campo disciplinario ni con una sola metodología, así como la urgencia de encontrar soluciones, exigiendo promover la actitud proactiva, la innovación y el pensamiento lateral; por consiguiente para la Universidad Istmo Americana, A.C., la creatividad, la iniciativa y el desarrollo de competencias para plantear y resolver problemas son estrategias formativas prioritarias.

g) Calidad de vida

El sentido humano de la ciencia y la tecnología estriba en el por qué, para qué y para quién se usan y dirigen; como consecuencia, la Universidad Istmo Americana, A.C., asume la responsabilidad y el compromiso de promover la generación y aplicación del conocimiento en beneficio de la humanidad.

h) Identidad nacional y cultura universal

La interdependencia entre los pueblos es cada vez más necesaria, por esta razón y con el fin de fortalecer nuestra identidad nacional y, simultáneamente, fortalecer el sentido de la convivencia, es esencial promover, incrementar, transmitir y difundir los mejores logros de nuestra cultura regional, nacional y de la cultura universal.

i) Ética profesional

La Universidad Istmo Americana, A.C., se rige por códigos de conducta que nos comprometen con los integrantes de la institución a ejercer su función enalteciendo los valores universales en beneficio de ellos mismos, de los educandos y de la sociedad en general.

j) Conocimiento y comunidad

Los procesos formativos de la Universidad Istmo Americana, A. C., dan respuesta a las necesidades individuales y colectivas de la realidad institucional y comunitaria, por lo que la actualización, capacitación, extensión y la formación permanentes constituyen pilares de la sociedad del conocimiento.

1.2.1 Misión

Facilitar el proceso enseñanza-aprendizaje en la formación y creatividad de los estudiantes, para que sean personas integrales con una capacidad de preparación profesional y cultural, con profundos valores morales y humanos, con una capacidad de liderazgo de acción positiva para fomentar el auténtico desarrollo del ser humano y de la sociedad.

1.2.2 Visión

Ser una institución líder en la formación de personas que transformen su trabajo con calidad en beneficio de la sociedad global.

1.3 Filosofía Institucional

- I. La enseñanza se basará en los resultados obtenidos por la ciencia y el conocimiento, aplicando nuevos métodos y técnicas pedagógicas;
- II. Los egresados serán capaces de aplicar los conocimientos adquiridos con acierto a los problemas inherentes al ámbito propio del ejercicio de su profesión;
- III. Crear espacios que posibiliten el desarrollo de las personas y de las comunidades sociales;
- IV. Promover la capacitación de su personal como actitudes básicas;
- V. Fomentar, organizar y coordinar la investigación científica, tecnológica y humanística como fuente del desarrollo;

- VI. Organizar y desarrollar las actividades de docencia a nivel superior en sus diversas modalidades siendo congruentes a los requerimientos de la sociedad;
- VII. Promover, organizar y coadyuvar en todas aquellas actividades encaminadas a la actualización de los profesionales en ejercicio y vigilar el correcto desempeño de su función profesional y social;

1.4 Objetivos Institucionales

- 1) Conservar, crear y transmitir la cultura en beneficio de la sociedad con el más alto nivel de calidad académica a través de las funciones sustantivas de docencia, investigación, difusión de la cultura y extensión de los servicios.
- 2) Estar vinculada permanentemente con la sociedad, para incidir en la solución de sus problemas y en el planteamiento de alternativas para el desarrollo, sustentadas en el avance de la ciencia y la tecnología, proporcionándole los beneficios de la cultura y obteniendo de ella en reciprocidad, los apoyos necesarios para su fortalecimiento.
- 3) Formar profesionistas de una excelencia académica, capaces de asumir el compromiso que tienen con la sociedad, de conservar, generar y transmitir el conocimiento científico, humanístico, artístico y tecnológico, mediante la docencia, la investigación, las actividades deportivas y la difusión de la cultura.
- 4) Fomentar en los integrantes de la comunidad universitaria la actitud crítica, humanística, científica y el espíritu emprendedor y de participación.
- 5) Promover que los programas educativos y acciones de extensionismos institucional, sean congruentes a los requerimientos de la sociedad veracruzana y mexicana, e impliquen soluciones a los mismos.
- 6) Apoyar, estimular y generar en el alumno la capacidad de crítica reflexiva y realista, como expresión básica del desarrollo y de su forma de pensar.

- 7) Orientar al alumno en la estructuración consciente y responsable de su jerarquía de valores ético-morales que le permitan modelar y moderar su relación y comportamiento personal.
- 8) Transformar el aula en laboratorio de búsqueda constante, de estrategia y metodología, que propicien el mejor aprovechamiento académico.
- 9) Proporcionar a los alumnos un ambiente de estabilidad, tranquilidad y reflexión, que le permita el óptimo y continuo desarrollo educacional, así como el análisis de los acontecimientos económicos, políticos y culturales de la sociedad que lo rodea.
- 10) Estructurar una planta docente con elementos de amplia experiencia, reconocida y solvencia moral, que fomente el libre desarrollo de la personalidad humana, alentando la búsqueda de la verdad, mediante la formación de un espíritu científico y una conciencia social y liberal.
- 11) Mantener la capacitación continua del personal docente y la implantación de nuevas técnicas educativas.
- 12) Impulsar en sus programas académicos, los principios, valores y práctica de la democracia, la justicia, la libertad, la igualdad, la solidaridad y el respeto a la dignidad humana.
- 13) Realizar acciones conjuntas con asociaciones y organizaciones de docencia, investigación, difusión de la cultura y extensión de los servicios, en el ámbito nacional e internacional. (Reglamento Interno de la UIA. 2003)

1.5 La Dimensión Académica de la Universidad Istmo Americana, A. C.

La dimensión académica es otro de los ejes fundamentales del Modelo Educativo de la Universidad Istmo Americana, A.C., en ella se engloban los parámetros de referencia para la formación profesional, la concepción del aprendizaje y sus condiciones, así como los estándares de la práctica educativa.

1.6 Formación Profesional

La formación profesional de que es objeto el ser humano que participa en el Proceso Educativo de la Universidad Istmo Americana, A.C., busca desarrollarse como un actor ético, asertivo y exitoso en su campo de acción con una clara identidad organizacional y compromiso social, así como en una persona íntegra en constante búsqueda de su autorrealización.

En consecuencia, una de las finalidades fundamentales del Proceso Educativo de la Universidad Istmo Americana, A.C., busca desarrollarse en un actor ético, asertivo y exitoso en su campo de acción con una clara identidad organizacional y compromiso social, así como una persona íntegra en constante búsqueda de su autorrealización a través de un proceso continuo de desarrollo en todas sus potencialidades del ser humano, que lo orienta a la búsqueda de su plenitud en el aprender a ser, a hacer, a aprender a emprender y a convivir, lo cual lo convierte en un ciudadano y un profesionalista consciente, responsable y solidario.

La formación profesional que impulsa y fomenta el Modelo propicia lo siguiente:

- a) El humanismo, como el entendimiento respetuoso tolerante y crítico de todas las manifestaciones de la cultura universal, sin menoscabo alguno ni exclusivismos, con el propósito de ampliar los horizontes de la educación integral del ser humano.
- b) El desarrollo de un conocimiento amplio y objetivo de la problemática contemporánea del ámbito profesional, desde las perspectivas regional, nacional y mundial.
- c) El desarrollo de las diferentes formas de competencias que permitan la certificación y el reconocimiento de clase mundial.
- d) La investigación como una forma de generar conocimientos que por su pertinencia y actualidad enriquezcan el acervo, así como una estrategia de

fortalecimiento de la vinculación de la Universidad Istmo Americana, A.C., con su entorno regional y nacional e internacional, y que tiene como propósito mejorar las condiciones de vida del ser humano.

- e) Un sentido de identidad profesional y pertenencia institucional.
- f) El desarrollo de las capacidades y habilidades para obtener, analizar, interpretar y aplicar información, generar conocimientos, así como para identificar, plantear, resolver problemas y tomar decisiones.
- g) El cultivo de habilidades y la capacidad de liderazgo para gestionar, emprender, negociar y dirigir.
- h) El crecimiento cualitativo de las capacidades y habilidades para el trabajo colegiado, en equipo, en situaciones cambiantes y en ambientes multiculturales.
- i) La formación de un profesionista que se actualiza constantemente y es competente en la comunicación oral y escrita, así como en otro idioma.
- j) La búsqueda permanente del conocimiento y la corresponsabilidad en el proceso de formación, como características inherentes al modo de vida del ser humano.

1.7 La Práctica Educativa

A través de la práctica educativa la Universidad Istmo Americana, A.C., busca asegurar la formación integral del ser humano. Por ello se considera que, en todo proceso de aprendizaje, son esenciales las acciones del facilitador y de la persona que construye su conocimiento, desde la perspectiva que el proceso es un escenario de aprendizaje significativos para ambos, bajo la concepción de este nuevo paradigma, en el sentido que exige actitudes nuevas y propicias para la construcción del conocimiento, su facilitación, el trabajo colaborativo y la comunicación asertiva.

La práctica educativa está orientada por la convicción de que el trabajo educativo en la Universidad Istmo Americana, A.C., asegura el desarrollo de la región así como de la entidad y la cual se caracteriza por lo siguiente:

- a) Impulsa la formación integral del ser humano.
- b) Busca el aprendizaje significativo de los actores involucrados en el Proceso Educativo.
- c) Se funda en el interés intelectual de los actores del Proceso Educativo.
- d) Fomenta el trabajo colegiado y colaborativo, y el desarrollo de la comunicación asertiva.
- e) Se alimenta de conocimientos actuales, vigentes y pertinentes para el desarrollo sostenible.
- f) Se orienta hacia el desarrollo de habilidades para el planteamiento y la solución de problemas.
- g) Concibe a la evaluación como una estrategia para asegurar e impulsar la construcción del conocimiento.
- h) Estimula la continua actualización profesional.
- i) Se nutre con la filosofía humanista y cultiva el espíritu de colaboración y compromiso social.
- j) Crea un ambiente propicio para el aprendizaje y la generación de conocimientos a los que se dedican los recursos necesarios para asegurar las condiciones que permitan el éxito del Proceso Educativo.

1.8 Planes de Estudios

Las licenciaturas de la Universidad Istmo Americana, A. C., conciben procesos integrados de generación de conocimientos y aprendizajes. Las habilidades profesionales que se buscan crear en los estudiantes comprenden:

1.8.1 Licenciatura en Administración de Empresas.

Formar profesionales que propicien la satisfacción de necesidades y el logro de objetivos de las instituciones públicas, privadas y sociales a través del aprovechamiento adecuado de sus recursos, mediante la planeación, organización, dirección y control, en las distintas áreas funcionales: Personal, finanzas, mercadotecnia, producción e informática.

1.8.2 Licenciatura en Ciencias de la Comunicación

Estudiar los orígenes, la estructura, el desarrollo y los cambios de los procesos de comunicación humana en sus distintos niveles y facetas, tanto interpersonales como grupales, organizacionales, colectivos o masivos, para contribuir de manera crítica a la difusión y análisis de la problemática social a través de la elaboración de mensajes y contenidos informativos escritos y audiovisuales.

1.8.3 Licenciatura en Derecho

Promover la Justicia Social e Individual por medio de la educación de los ordenamientos legales a la realidad. Procurar la defensa y aplicación de los valores nacionales surgidos y consagrados a lo largo de nuestra historia y plasmados en la Constitución Política de los Estados Unidos Mexicanos. Aplicar con honestidad y lealtad los principios éticos del derecho en beneficio de la comunidad.

1.8.4 Licenciatura en Informática

Formar profesionales capaces de diseñar e implantar soluciones basadas en sistemas de información, que faciliten la toma de decisiones y agilicen las operaciones propias de una organización. Esta labor debe adaptarse a los avances tecnológicos y cambios socioeconómicos en el ámbito nacional e internacional.

1.8.5 Licenciatura en Ingeniería en Sistemas Computacionales

- Formar Ingenieros en Sistemas Computacionales con amplios conocimientos tanto teóricos como prácticos, que le sean útiles para el desarrollo y diseño de sistemas de información, con la capacidad de formular proyectos de inteligencia artificial, sistemas de comunicación y automatización de sistemas.
- Que sean capaces de construir, configurar, evaluar y seleccionar obras y entornos de servicios computacionales y telecomunicaciones a partir del análisis de situaciones y problemas propios de ser tratados mediante sistemas de cómputo.
- Distinguir y diseñar nuevas formas de expresión e interacción entre el hombre y la computadora.

1.8.6 Licenciatura en Mercadotecnia

Formar profesionales capacitados para diseñar y aplicar estrategias en cuanto al desarrollo de productos, publicidad, promoción, relaciones públicas, ventas, distribución, precio, servicio e investigación de mercados en empresas comerciales y de servicio que compitan en un contexto local, regional y global. Así como también tengan conocimientos elementales en campos de Administración, Economía, Contabilidad, Comunicaciones, etc., que son variables fundamentales en la evaluación e implantación de una estrategia

1.8.7 Licenciatura en Pedagogía

Formar profesionales capaces de analizar problemas, proponer soluciones relacionadas con el campo educativo y atender a diversos sectores de la población que requieren ayuda pedagógica. El pedagogo emplea técnicas como la psicometría, la estadística descriptiva, programas de computación, dinámica de grupos y otras, de acuerdo con el método de enseñanza o investigación adoptado.

1.8.8 Licenciatura en Psicología Organizacional

Preparar especialistas en el conocimiento humano en las organizaciones y por ende debe conocer a la organización misma con todos sus procesos, aplicando siempre la visión de sistemas. Está enfocada a desarrollar habilidades ínter-conductuales y de observación, para que el especialista asegure la efectividad organizacional a través del factor humano.

1.9 La reconstrucción histórica de la Universidad Istmo Americana, A. C.

La Universidad Istmo Americana, A. C. esta situada en la ciudad y puerto de Coatzacoalcos en la calle de Román Marín # 1107, Colonia Manuel Ávila Camacho. Constituyéndose la Sociedad Civil en el año 1999, iniciando actividades el 18 de agosto del 2002, con los programas de licenciatura de Mercadotecnia y Derecho incorporados a la Secretaría de Educación Pública, contándose durante ese período escolar con un total de 115 alumnos y una plantilla de 18 docentes. En el 2003 la Universidad Istmo Americana, A. C. integró las licenciaturas en Administración de Empresas, Pedagogía, Informática y Ciencias de la Comunicación incorporadas a la Secretaría de Educación y Cultura del Estado de Veracruz; integrando en el 2004 las licenciaturas de Psicología Organizacional e Ingeniería en Sistemas Computacionales.

CAPITULO 2

PLANTEAMIENTO DEL PROBLEMA

2.1 JUSTIFICACION

2.2. HIPOTESIS

2.3 OBJETIVO GENERAL

2.3 OBJETIVOS ESPECIFICOS

2.5 ALCANCES Y LIMITACIONES

La Universidad Istmo Americana es un organismo dedicado a la oferta de servicios educativos a la sociedad en general sin distinción de ninguna índole y desarrolla, entre otros, programas y estrategias para estudiantes que se encuentran en situaciones especiales

Actualmente la Universidad cuenta con 8 Licenciaturas y más de 600 alumnos, a los cuales, además de su formación educativa también ofrecen cursos y congresos así como apoyos económicos a los alumnos mediante becas como parte de su responsabilidad social.

Debido al crecimiento tan importante que ha presentado desde el inicio de sus operaciones en el 2002, y apoyándose las tecnologías de información existentes, se ha recurrido a la implementación de un sistema de control escolar “ESCOLA”, mismo que hasta la fecha se encuentra operando en la institución.

2.1 JUSTIFICACION.

El sistema actual “ESCOLA”, se encarga entre otras cosas de llevar a cabo un registro de altas y/o inscripciones de toda la comunidad de alumnos, así como del cobro de los arenceles por conceptos de colegiaturas e inscripciones y otros pagos. Todo esto con la finalidad de tener un mejor control de los mismos, logrando con esto garantizar la oferta de un servicio eficiente.

El sistema de control administrativo actual esta escrito en QBASIC y su base de datos se almacena mediante el manejo de archivos indexados. Su interfaz poco agradable en modo de texto no permite la utilización del mouse, lo cual no solo entorpece la interacción con el sistema , sino que además si se consideran los avances en tecnologías de información actuales dicho sistema podría considerarse primitivo.

La gran desventaja de la manera en la que opera el ESCOLA es que la Base de Datos tiene únicamente capacidad para registrar 16 pagos durante un año fiscal (Enero-Diciembre) 12 Correspondientes a colegiaturas, 2 mas para el pago de inscripciones (Enero y Agosto) y únicamente dos extras para el registro de otros tipo de pagos, no

permitiendo con esto el correcto registro del resto de la información. Aunado a esto por el proceso de registro que realiza no permite saber el monto total de la deuda de los alumnos, ya que simplemente se registran los pagos que se realizan no reflejando así el adeudo que pudieran presentar los alumnos con la institución.

Otra desventaja que presenta el seguir operando estas actividades con el sistema “ESCOLA”, es que la Universidad Istmo Americana, A.C. no tiene el control sobre el mismo, sino mas bien dependen de una empresa externa de desarrollo de software, la cual mantiene dicho sistema bajo arrendamiento a la Universidad, lo que genera gastos que podrían evitarse sobre todo si se parte del punto que la propia institución cuenta con las carreras de Licenciatura en Informática y Licenciatura en Ingeniería en Sistemas Computacionales.

El desarrollo de un Sistema de control administrativo propio de la Universidad permitirá además de un ahorro económico, lograr tener un mejor control de la información, así como la flexibilidad de realizar cambios operativos contando con acceso libre a información a la que antes no se tenía acceso.

2.2 HIPOTESIS

“La implementación de un sistema con una interfaz mas agradable al usuario, adiciona a un mejor funcionamiento, contribuirá a un ahorro de tiempo en los procedimientos de la caja así como a tener información mas precisa y detallada de los movimientos administrativos que realicen los alumnos, logrando con esto mejorar el servicio que se ofrece en la actualidad”

2.3 OBJETIVO GENERAL.

En un esfuerzo por solucionar los problemas ya mencionados, así como la parte Administrativa del Sistema de Control Escolar, el objetivo general del presente trabajo de tesis consiste en la recreación del sistema de control escolar utilizado actualmente, así como la ordenación lógica de la Base de Datos que será utilizada en todo el proceso de inscripciones y cobros de colegiaturas en la Universidad de Sotavento

2.4 OBJETIVOS ESPECIFICOS

- ❖ Describir las características operativas de la inscripción y pagos de colegiaturas y exámenes de los alumnos
- ❖ Efectuar el análisis correspondiente para la generación del programa de control escolar nuevo (SIAU) de tal forma que el usuario pueda realizar sus labores con mayor facilidad.
- ❖ Participar en el rediseño de un nuevo sistema de control Escolar
- ❖ Diseñar la nueva interfaz del sistema de control escolar

2.5 ALCANCES Y LIMITACIONES

Concretando la información que se presenta en este proyecto, se abarcan los siguientes puntos:

1. Definición del esquema de la base de datos para el sistema de control escolar (SIAU), concretamente pagos usando el manejador de Base de Datos MySQL 5.0.26
2. Diseño funcional de la interfaz del proceso de inscripciones y cobros de colegiaturas y otros pagos.

El diseño y la implementación del sistema de Alumnos, Calificaciones y Horarios, quedan fuera del alcance del presente.

Respecto a las limitantes presentadas la principal, es la falta de documentación del sistema actual (ESCOLA) , así como la de su Base de Datos, la pérdida de tiempo en el ordenamiento de los datos, así como la dificultad de tener acceso a estos.

CAPITULO 3

MARCO TEORICO

3.1 REGISTRO DE ALUMNOS

3.1.1 ALUMNOS DE REINGRESO

3.1.2 PROCEDIMIENTO DE INSCRIPCION

3.1.3 OTORGAMIENTO DE BECAS

3.1.4 PAGO DE SERVICIOS EDUCATIVOS

3.2 PROGRAMACION EN JAVA

3.2.1 ¿QUÉ ES JAVA?

3.2.2 ORIGENES DE JAVA

3.2.3 CARACTERISTICAS DE JAVA

3.2.4 JAVA Y BASES DE DATOS

3.3 ¿QUÉ ES MySQL?

3.3.1 HISTORIA DE MySQL

3.3.2 CARACTERISTICAS DEL MySQL

En este capítulo, se analiza de una manera práctica y general el procedimiento que se lleva a cabo en la Universidad Istmo Americana para el registro de alumnos de nuevo ingreso y reingreso, documentos que se deben presentar, así como la forma en como se realizan los pagos de las inscripciones semestrales y las colegiaturas mensuales.

Así mismo, se redactan los orígenes del lenguaje de programación a utilizar (java) además de sus principales ventajas sobre otros lenguajes de programación existentes. De igual manera se detalla la información más relevante del gestor de Base de Datos MySQL, como lo son las estructuras que todas las tablas deben contener.

3.1 REGISTRO DE NUEVO INGRESO

Para efectos de este apartado se consideran alumnos de primer ingreso a:

Aquellos que se registran en el sistema incorporado y que cumplan con los requisitos académicos así como administrativos, que rijan a la Universidad Istmo Americana, procedentes de cualquier escuela y/o Institución Educativa de: SEP, Estatal, del extranjero o de la UNAM.

Antes del inicio del curso escolar para lograr la correcta ubicación dentro de la carrera que el aspirante o los aspirantes a cursar el nivel superior deseen realizar y que imparta la Universidad Istmo Americana, deberán de presentar al Departamento de Registro y Control Escolar los siguientes documentos:

1. Acta de Nacimiento Original en perfectas condiciones y dos fotocopias.
2. Certificado de Secundaria original y dos fotocopias.
3. Certificado de Bachillerato original y dos fotocopias.

4. Carta de Buena Conducta original y dos fotocopias.
5. Certificado Médico expedido por una dependencia pública (Centro de Salud.)
6. 6 (seis) Fotografías tamaño infantil (Deberá anotar al reverso de las mismas el nombre completo, carrera y grupo al que será asignado con lápiz)
7. Llenar hoja de solicitud de inscripción (Formato I-01) que proporcionará el Departamento de Registro y Control Escolar.

El Departamento de Registro y Control Escolar deberá expedirle al interesado acuse de recibo de lo anterior.(Anexo)

Si el alumno procede de alguna institución extranjera, deberá de presentar además, Dictamen de Revalidación o equivalencia de los estudios realizados en el extranjero y deberán estar legalizados por las siguientes dependencias del país donde se expidieron y contar con el “Apostille” correspondiente:

- ❖ Ministerio de Educación.
- ❖ Ministerio de relaciones exteriores; y del
- ❖ Consulado Mexicano

El alumno deberá cubrir el costo de inscripción al inicio de cada semestre. El cual será fijado por el consejo de Administración de la Universidad Istmo Americana y se dará a conocer en los tiempos establecidos por la misma institución.

Si el alumno cubre los requisitos antes mencionados el Departamento de Registro y Control Escolar, deberá llenar la **Hoja de Registro** anexando a esta el original del Acta de Nacimiento, Certificado de Secundaria y Certificado de Bachillerato, y pegar la fotografía con pegamento blanco y cancelar la misma con el sello correspondiente a la licenciatura a cursar.

El Departamento de Registro y Control Escolar abrirá expedientes en forma individual para cada uno de los aspirantes a ingresar al sistema incorporado y anotará en la hoja de inscripción cualquier observación con respecto a la documentación del interesado, no deberá de perforar los originales de lo contrario perderá validez oficial dicha documentación.

El Departamento de Registro y Control Escolar deberá checar minuciosamente la documentación presentada por el aspirante corroborando que coincidan nombres y apellidos del acta de nacimiento con los certificados de secundaria y bachillerato, no se aceptaran documentos rayados, manchados, corregidos, enmendados o rotos.

Por ningún motivo deberá poner cinta adhesiva a los documentos originales que se encuentren en resguardo del Centro de Estudios Universitario.

El Departamento de Registro y Control Escolar dará a conocer por escrito al aspirante que no presente la documentación completa marcada como requisitos de inscripción, que cuenta con 30 días hábiles a partir de la fecha de inicio del ciclo escolar para la entrega de la documentación faltante y necesaria para su registro correspondiente ante la Secretaría de Educación Pública o la Secretaria de Educación de Veracruz, según sea el caso, de lo contrario se procederá a la baja definitiva y sin perjuicio alguno para la Universidad Istmo Americana debiendo firmar de conformidad en dicho comunicado el aspirante a la Secretaría de Educación Pública o Secretaria de Educación de Veracruz, en caso de entregar la documentación de forma diferida, deberá hacer la anotación en el expediente, anotando claramente la fecha en que estos serán entregados.

Deberá de hacer entrega inmediata al aspirante después de la inscripción al sistema incorporado:

1. Reglamento de Alumnos
2. Circular de los costos a cubrir durante el semestre.

3. Normatividad Interna
4. Calendario Escolar
5. Reglamento de Biblioteca
6. Reglamento de Becas (En caso de que se vea beneficiado con beca)
7. Circular de requisitos necesarios para justificar faltas

Nota:

En caso de que aspirante llegará a presentar documentación falsa, total o parcialmente, se anulara el registro respectivo, quedando sin efecto todos los actos derivados del mismo.

3.1.1 Registro de alumnos de Reingreso.

Se considera alumno de reingreso al Sistema SEP o SEV el registro reconocido por la Secretaría de Educación Pública o la Secretaria del Estado de Veracruz de aquellos alumnos que ya tienen número de matrícula y que cuenten con los requisitos necesarios tanto académicos como administrativos para conservar la misma.

El Departamento de Registro y Control Escolar deberá verificar que el alumno de reingreso cuente con los requisitos necesarios para hacer efectivo dicho trámite.

1. Deberá ser alumno regular en su semestre antecedente y no adeudar más de dos asignaturas del semestre anterior.
2. No podrá ser promovido en una asignatura seriada, quien adeude la asignatura antecedente.
3. No Tener Adeudo en los pagos correspondientes al semestre anterior.

4. No adeudar material bibliográfico, material didáctico y/o equipo de apoyo.
5. Cubrir los pagos correspondientes a inscripción del semestre a cursar así como incorporación del mismo. (Este pago se realizará al inicio de cada semestre)
6. Si proviene de otra escuela, deberá presentar kardex emitido por la escuela de procedencia, cambio de plantel debidamente sellado y requisitado por el plantel del que se hace el cambio, así como una baja por parte de la otra institución, y seis fotografías tamaño infantil.
7. Para los alumnos que cursen el séptimo semestre de cualquiera que fuere la carrera deberá de presentar 06 fotografías tamaño infantil para la tramitación de su Servicio Social.
8. Certificado Médico expedido por el Centro de Salud actualizado.
9. Llenar la solicitud de reinscripción que proporcione la Institución al momento de Reinscribirse.

Si el alumno cuenta con los requisitos antes mencionados el Departamento de Registro y Control Escolar procederá a dar de alta en el sistema al alumno solicitante, en el semestre que académicamente corresponda.

Si el alumno que solicita su reinscripción al sistema gozó de una baja temporal el Departamento de Registro y Control Escolar, deberá Verificar en el Kardex ó del solicitante en que quedo oficialmente registrado y darle de alta en el semestre que académicamente corresponda de acuerdo al mismo.

En caso de que el alumno de reingreso solicite cambiar de carrera deberá hacerlo registrándose desde el primer semestre de la carrera que desee cursar, presentando un oficio en donde manifieste que será la única ocasión en que realizará dicho cambio y otro en donde solicite la autorización y aceptación del Director General, dicho alumno deberá ser registrado con el número de matrícula antes asignado, así como el Formato de cambio de plantel.

En caso de que el alumno de reingreso haya interrumpido sus estudios, podrá continuarlos, teniendo que sujetarse al plan de estudios vigente en la fecha de su reingreso.

Los alumnos de reingreso tendrán derecho a:

- ✓ Conocer su número de matrícula.
- ✓ Contar con una credencial que lo acredite como alumno.

Nota: siempre y cuando cumplan con lo dispuesto para poder ser alumno regular en el semestre a iniciar.

3.1.2 PROCEDIMIENTO DE INSCRIPCIÓN.

1. El Departamento de Registro y Control Escolar, recibirá la documentación completa del aspirante para su cotejo.
2. Una vez revisada la documentación y autorizada la solicitud de inscripción, pasará al departamento de cobranza para su pago respectivo.
3. Deberá de regresar al Departamento de Registro y Control Escolar para entregar el comprobante de pago (Orinal para el interesado y copia para el Archivo) así como la solicitud de Inscripción.
4. El Departamento de Registro y Control Escolar, deberá de abrir un expediente de manera individual para el alumno en donde se anexará la copia del folio correspondiente al archivo.
5. Una vez realizado el pago de inscripción el Departamento de Registro y Control Escolar hará entrega del paquete informativo a los alumnos inscritos.

3.1.3 Otorgamiento de Becas.

Para que un alumno de la Universidad Istmo Americana pueda gozar del derecho de gozar con algún tipo de beca, deberá sujetarse a las normas establecidas por la propia institución así como cubrir los siguientes requisitos:

- ❖ Haber cursado la escolaridad dentro de la institución
- ❖ Tener un promedio mínimo de 9.0 (Nueve)
- ❖ Haber observado una excelente conducta y actitud cooperadora
- ❖ Llenar la solicitud que le será entregada previamente por la institución.
- ❖ No adeudar materias ni presentación de las mismas con carácter de 2da Oportunidad o Extraordinario.
- ❖ No ausentarse en horario de clases, únicamente en las ocasiones en que las causas sean justificadas.

El alumno becario deberá presentar un óptimo rendimiento académico durante el semestre, así mismo deberá asistir con puntualidad a todas sus clases entregando trabajos que le fueron previamente encargados por los diversos maestros del semestre que cursa, debiendo también observar buena conducta dentro y fuera del plantel educativo.

Son de igual manera sus obligaciones presentarse y desempeñar satisfactoriamente las comisiones y trabajos que le sean asignados por la Universidad Istmo Americana, así como asistir a todas las juntas de becarios que se convoquen por parte de la coordinación de extensión universitaria.

No deberá presentar atraso en el pago de sus colegiaturas y otros servicios educativos, además de participar en los eventos que sean organizados por la Universidad, apoyarla realizando actividades dentro del plantel como son:

- A. Servicio de auxiliares en la biblioteca

- B. Servicio de auxiliares en el centro de cómputo para supervisión de tiempo y uso apropiado de las computadoras.
- C. Participar en el orden u otro tipo de información que la dirección técnica tenga interés en dar a conocer
- D. Organización e integración de grupo del cual forme parte para que todos participen en la realización de actividades deportivas, sociales, culturales y de servicios a la comunidad; así como las requeridas para el beneficio del plantel del cual forma parte.
- E. Auxiliar del catedrático en donde obtuvo la mejor calificación del semestre anterior

El alumno podrá gozar de su beca mientras esta sea renovada y autorizada semestralmente por la Universidad Istmo Americana, así mismo son causas de reducción o cancelación definitiva cuando se presenten las siguientes causas entre otras:

- A. La comprobación de información falsa proporcionada por el becario sobre la situación económica.
- B. No cumplir con el promedio establecido.
- C. Haber reprobado una o más materias.
- D. Salirse de clases.
- E. Participar en movimientos en contra del personal docente o administrativo
- F. Alguna otra falta establecida en el Reglamento Interno de Alumnos.

El becario y su tutor deberán firmar el reglamento de becas para su conocimiento y la vez como un compromiso con la institución denominada Universidad Istmo Americana, A.C., para el buen cumplimiento o desempeño de todo lo antes mencionado.

3.1.4 Disposiciones Administrativas

Pagos de Servicios Educativos.

Los pagos por servicios educativos (colegiaturas) deberán pagarse los primeros diez días de cada mes y podrán pagarse de contado o en 6 Pagos. Si se cubre de contado el pago del importe total del semestre, este deberá realizarse el primer día de clases y se otorgara un descuento del 10%.

Respecto a la inscripción, esta se realizara durante los primeros diez días de clases, quedando el alumno registrado en la institución con todos los derechos y obligaciones establecidos dentro del reglamento interno.

A lo largo de la carrera el alumno además del pago de colegiaturas e inscripciones deberá realizar otros tipos de pagos mismos que cubrirán servicios adicionales a los cuales tiene derecho. Entre estos conceptos se encuentran los siguientes:

A. Durante el Semestre.

- a) Pago de constancias (y/o reposición)
- b) Pago de historial académico (y/o reposición)
- c) Pago de exámenes extraordinarios y/o II Vuelta (que tendrá el carácter de Ordinario)

B. Otros conceptos de pagos.

- a) Al finalizar la carrera:
 - 1. Pago por registro y liberación del servicio social.
 - 2. Pago por concepto de Examen profesional.
 - 3. Pago por concepto de expedición de Título profesional.
 - 4. Pago por concepto de expedición de carta de pasante, que expide la institución.
 - 5. Pago por certificado completo de estudios profesionales.

- b) Otros Servicios educativos:
 - 1. Pago por duplicados de documentos
 - 2. Pago por expedición de Certificados parciales.
 - 3. Pagos por baja y/o cambios de plantel.
 - 4. Pago por equivalencia de asignaturas.
 - 5. Pago por trámite de cambio de carrera.
 - 6. Pago por trámite de segunda carrera.

C. Intereses Moratorios.

- a) La colegiatura deberá cubrirse los primeros diez días de cada mes, en caso de incumplimiento cubrirán adicionalmente los intereses bancarios que correspondan.

D. Medidas por incumplimiento de pagos.

- a) El incumplimiento del pago de tres o más colegiaturas, tendrán como consecuencia la baja definitiva del alumno.
- b) La institución se reservara el derecho de
- c) Pago de intereses moratorios correspondientes por atrasos en los mismos.

E. Pagos por exámenes extraordinarios, trámites y otros pagos.

- a) Los pagos por exámenes Extraordinarios serán cubiertos a tiempo por los alumnos que adeuden materias, en caso contrario perderán el derecho a examen sin perjuicio alguno para la institución.
- b) Los alumnos que no entreguen a tiempo la documentación que requiere la SEP y/o SEV, cubrirán los pagos impuestos por las mismas como multas, calificando la entrega de documentación como extemporánea sin perjuicio para la institución.

3.2 Programación en Java

3.2.1 ¿Qué es Java?

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los 90. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en un bytecode que es interpretado (usando normalmente un compilador JIT), por una máquina virtual Java.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos mucho más simple y elimina herramientas de bajo nivel como punteros.

La plataforma Java y el lenguaje Java empezaron como un proyecto interno de Sun Microsystems en diciembre de 1990. Patrick Naughton, ingeniero de Sun, estaba decepcionado con el estado de C++ y la API de C y sus herramientas. Mientras consideraba migrar a NeXT, Naughton recibió la oferta de trabajar en una nueva tecnología, y así comenzó el proyecto *Stealth*.

3.2.2 Orígenes de Java

El Proyecto Stealth fue rebautizado como *Green Project* (o *Proyecto Verde*) cuando James Gosling y Mike Sheridan se unieron a Naughton. Con la ayuda de otros ingenieros, empezaron a trabajar en una pequeña oficina en Sand Hill Road en Menlo Park, California. Intentaban desarrollar una nueva tecnología para programar la siguiente generación de *dispositivos inteligentes*, en los que Sun veía un campo nuevo a explotar.

El equipo pensó al principio usar C++, pero se descartó por varias razones. Al estar desarrollando un sistema empotrado con recursos limitados, C++ no es adecuado por necesitar mayor potencia además de que su complejidad conduce a errores de desarrollo. La ausencia de un *recolector de basura* (*garbage collector*) obligaba a los programadores a manejar manualmente el sistema de memoria, una tarea peligrosa y proclive a fallos. El equipo también se encontró con problemas por la falta de herramientas portables en cuanto a seguridad, programación distribuida, y programación concurrente. Finalmente abogaban por una plataforma que fuese fácilmente portable a todo tipo de dispositivo.

Bill Joy había concebido un nuevo lenguaje que combinase lo mejor de *Mesa* y C. En un escrito titulado *Further* (más lejos), proponía a Sun que sus ingenieros crearan un entorno *orientado a objetos* basado en C++. Al principio Gosling intentó modificar y ampliar C++, a lo que llamó C++ ++ --, pero pronto descartó la idea para crear un lenguaje completamente nuevo, al que llamó *Oak*, en referencia al roble que tenía junto a su oficina.

El equipo dedicó largas horas de trabajo y en el verano de 1992 tuvieron listas algunas partes de la plataforma, incluyendo el Sistema Operativo Verde, el lenguaje Oak, las librerías y el hardware. La primera prueba, llevada a cabo el 3 de Septiembre de 1992, se centró en construir una PDA (*Personal Digital Assistant* o Asistente Digital Personal) llamada *Star7*, que contaba con una interfaz gráfica y un asistente apodado "Duke" para guiar al usuario.

En noviembre de ese mismo año, el Proyecto Verde se convirtió en **FirstPerson, Inc**, una división propiedad de Sun Microsystems, y el equipo se trasladó a Palo Alto (California). El interés se centró entonces en construir dispositivos interactivos, hasta que Time Warner publicó una solicitud de oferta para un adaptador de televisión. Es decir, un aparato que se sitúa entre la televisión y una fuente de señal externa y que adapta el contenido de ésta (video, audio, páginas Web, etc.) para verse en la pantalla. Entonces, Firstperson cambió de idea y envió a Warner una propuesta para el dispositivo que deseaban. Sin embargo, la industria del cable consideró que esa propuesta daba demasiado control al usuario, con lo que FirstPerson perdió la puja a favor de Silicon Graphics Incorporated. Un trato con la empresa 3DO para el mismo tipo de dispositivo tampoco llegó a buen puerto. Viendo que no había muchas posibilidades en la industria de la televisión, la compañía volvió al seno de Sun..

3.2.3 Características de Java

A continuación se hace una pequeña redacción de las características del lenguaje, que ayudarán a ver para que tipo de problemas está pensado Java:

Simple

Es un lenguaje sencillo de aprender. Su sintaxis es la de C++ “simplificada”. Los creadores de Java partieron de la sintaxis de C++ y trataron de eliminar de este todo lo que resultase complicado o fuente de errores en este lenguaje.

Orientado a Objetos

Posiblemente sea el lenguaje más orientado a objetos de todos los existentes; en Java todo, a excepción de los tipos fundamentales de variables (int, char, long...) es un objeto.

Distribuido

Java está muy orientado al trabajo en red, soportando protocolos como TCP/IP, UDP, HTTP y FTP. Por otro lado el uso de estos protocolos es bastante sencillo comparandolo con otros lenguajes que los soportan.

Robusto

El compilador Java detecta muchos errores que otros compiladores solo detectarían en tiempo de ejecución o incluso nunca. (ej: `if(a=b) then ...` el compilador Java no nos dejaría compilar este código.

Seguro

Sobre todo un tipo de desarrollo: los Applet. Estos son programas diseñados para ser ejecutados en una página web. Java garantiza que ningún Applet puede escribir o leer de nuestro disco o mandar información del usuario que accede a la página a través de la red (como, por ejemplo, la dirección de correo electrónico). En general no permite realizar cualquier acción que pudiera dañar la máquina o violar la intimidad del que visita la página web.

Portable

En Java no hay aspectos dependientes de la implementación, todas las implementaciones de Java siguen los mismos estándares en cuanto a tamaño y almacenamiento de los datos.

Esto no ocurre así en C++, por ejemplo. En éste un entero, por ejemplo, puede tener un

tamaño de 16, 32 o más bits, siendo lo única limitación que el entero sea mayor que un short y menor que un long int. Así mismo C++ bajo UNIX almacena los datos en formato little endian, mientras que bajo Windows lo hace en big endian. Java lo hace siempre en little edian para evitar confusiones.

Arquitectura Neutral

El código generado por el compilador Java es independiente de la arquitectura: podría

ejecutarse en un entorno UNIX, Mac o Windows. El motivo de esto es que el que realmente ejecuta el código generado por el compilador no es el procesador del ordenador directamente, sino que este se ejecuta mediante una máquina virtual. Esto permite que los Applets de una web pueda ejecutarlos cualquier máquina que

se conecte a ella independientemente de que sistema operativo emplee (siempre y cuando el ordenador en cuestión tenga instalada una máquina virtual de Java).

Rendimiento medio

Actualmente la velocidad de procesamiento del código Java es semejante a la de C++, hay ciertas pruebas estándares de comparación (benchmarks) en las que Java gana a C++ y viceversa.

Esto es así gracias al uso de compiladores *just in time*, compiladores que traducen los bytecodes de Java en código para una determinada CPU, que no precisa de la máquina virtual para ser ejecutado, y guardan el resultado de dicha conversión, volviéndolo a llamar en caso de volverlo a necesitar, con lo que se evita la sobrecarga de trabajo asociada a la interpretación del bytecode.

No obstante por norma general el programa Java consume bastante más memoria que el programa C++, ya que no sólo ha de cargar en memoria los recursos necesarios para la ejecución del programa, sino que además debe simular un sistema operativo y hardware virtuales (la máquina virtual). Por otro lado la programación gráfica empleando las librerías

Swing es más lenta que el uso de componentes nativos en las interfaces de usuario.

En general en Java se ha sacrificado el rendimiento para facilitar la programación y sobre todo para conseguir la característica de neutralidad arquitectural, si bien es cierto que los avances en las máquinas virtuales remedian cada vez más estas decisiones de diseño.

Multithread

Soporta de modo nativo los threads, sin necesidad del uso de librerías específicas (como es el caso de C++). Esto le permite además que cada Thread de una aplicación Java pueda correr en una CPU distinta, si la aplicación se ejecuta en una máquina que posee varias CPUs. Las aplicaciones de C++ no son capaces de distribuir, de modo transparente para el programador, la carga entre varias CPUs.

3.2.4 Java y las Bases de Datos

Muchos programadores quizás tengan mayor interés en realizar programación basada conjunto a Bases de Datos, pues Java no se queda atrás, Java no implementa Bases de Datos, ya que solo es un lenguaje de programación, pero implementa funciones que permiten al programador realizar conexiones entre la interfaz de usuario y el Gestor de Base de Datos.

Java permite conectarse por medio de puentes JDBC o a través de Driver's a programas gestores de bases de datos, su independencia entre ambos permite al usuario mantener siempre un enfoque, separando el diseño de la Base de Datos y el de la interfaz en dos mundos de pensamientos diferentes el mundo de los datos y el mundo de las interfaces.

Java es orientado a objetos por ende da solidez a la aplicación evitando cortes bruscos del programa y permitiendo continuar de esta manera con la aplicación, lo que permite montar cualquier aplicación con Bases de Datos a través de la red de forma segura y sólida.

Java En el PC de escritorio

Aunque cada vez la tecnología Java se acerca más y más al PC de sobremesa, las aplicaciones Java han sido relativamente raras para uso doméstico, por varias razones.

- ❖ Las aplicaciones Java pueden necesitar gran cantidad de memoria física.
- ❖ La Interfaz Gráfica de Usuario (GUI) no sigue de forma estricta la *Guía para Interfaces Humana* (Human Interface Guidelines), así como tampoco aquella a la que estamos habitualmente acostumbrados. La apariencia de las fuentes no tiene las opciones de optimización activadas por defecto, lo que hace aparecer al texto como si fuera de baja calidad.
- ❖ Las herramientas con que cuenta el JDK no son suficientemente potentes para construir de forma simple aplicaciones potentes. Aunque el uso de herramientas como Eclipse, un IDE con licencia libre de alta calidad, facilita enormemente las tareas de desarrollo.

- ❖ Hay varias versiones del Entorno en Tiempo de Ejecución de Java, el JRE. Es necesario tener instalada la versión adecuada.
- ❖ Las aplicaciones basadas en la Web están tomando la delantera frente a aquellas que funcionan como entidades independientes. Las nuevas técnicas de programación producen aplicaciones basadas en un modelo en red cada vez más potentes.

Sin embargo hay aplicaciones Java cuyo uso está ampliamente extendido, como los NetBeans, el entorno de desarrollo (IDE) Eclipse, y otros programas como LimeWire y Azureus para intercambio de archivos. Java también es el motor que usa MATLAB para el renderizado de la interfaz gráfica y para parte del motor de cálculo. Las aplicaciones de escritorio basadas en la tecnología Swing y SWT (Standard Widget Toolkit) suponen una alternativa a la plataforma .Net de Microsoft.

Disponibilidad del JRE de Java

Una versión del JRE (Java Runtime Environment) está disponible en la mayoría de equipos de escritorio. Sin embargo, Microsoft no lo ha incluido por defecto en su sistema operativo, ya que Windows XP fue lanzado en 2001. En el caso de Apple, éste incluye una versión propia del JRE en su sistema operativo, el Mac OS. También es un producto que por defecto aparece en la mayoría de las distribuciones de Linux. Debido a incompatibilidades entre distintas versiones del JRE, muchas aplicaciones prefieren instalar su propia copia del JRE antes que confiar su suerte a la aplicación instalada por defecto. Los desarrolladores de applets de Java o bien deben insistir a los usuarios en la actualización del JRE, o bien desarrollar bajo una versión antigua de Java y verificar el correcto funcionamiento en las versiones posteriores.

Historial de versiones

Java ha experimentado numerosos cambios desde la versión primigenia, JDK 1.0, así como un enorme incremento en el número de clases y paquetes que componen la librería estándar. Desde J2SE 1.4, la evolución del lenguaje ha sido regulada por el JCP (Java Community Process), que usa *Java Specification Requests* (JSRs)

para proponer y especificar cambios en la plataforma Java. El lenguaje en sí mismo está especificado en la *Java Language Specification* (JLS), o Especificación del Lenguaje Java. Los cambios en los JLS son gestionados en JSR 901.

- **JDK 1.0** (23 de enero de 1996) — Primer lanzamiento. comunicado de prensa
- **JDK 1.1** (19 de febrero de 1997) — Principales adiciones incluidas: comunicado de prensa
 - una reestructuración intensiva del modelo de eventos AWT (Abstract Windowing Toolkit)
 - clases internas (inner classes)
 - JavaBeans
 - JDBC (Java Database Connectivity), para la integración de bases de datos
 - RMI (Remote Method Invocation)
- **J2SE 1.2** (8 de diciembre de 1998) — Nombre clave *Playground*. Esta y las siguientes versiones fueron recogidas bajo la denominación **Java 2** y el nombre "J2SE" (Java 2 Platform, Standard Edition), reemplazó a JDK para distinguir la plataforma base de J2EE (Java 2 Platform, Enterprise Edition) y J2ME (Java 2 Platform, Micro Edition). Otras mejoras añadidas incluían: comunicado de prensa
 - la palabra reservada (keyword) `strictfp`
 - reflexión en la programación
 - la API gráfica (Swing) fue integrada en las clases básicas
 - la máquina virtual (JVM) de Sun fue equipada con un compilador JIT (Just in Time) por primera vez
 - Java Plug-in
 - Java IDL, una implementación de IDL (Interfaz para Descripción de Lenguaje) para la interoperabilidad con CORBA
 - Colecciones (Collections)

- **J2SE 1.3** (8 de mayo de 2000) — Nombre clave *Kestrel*.

Los cambios más notables fueron: comunicado de prensa lista completa de cambios

- - la inclusión de la máquina virtual de HotSpot JVM (la JVM de HotSpot fue lanzada inicialmente en abril de 1999, para la JVM de J2SE 1.2)
 - RMI fue cambiado para que se basara en CORBA
 - JavaSound
 - se incluyó el Java Naming and Directory Interface (JNDI) en el paquete de librerías principales (anteriormente disponible como una extensión)
 - Java Platform Debugger Architecture (JPDA)
- **J2SE 1.4** (6 de febrero de 2002) — Nombre Clave *Merlin*. Este fue el primer lanzamiento de la plataforma Java desarrollado bajo el Proceso de la Comunidad Java como JSR 59. Los cambios más notables fueron: comunicado de prensalista completa de cambios
 - Palabra reservada `assert` (Especificado en JSR 41.)
 - Expresiones regulares modeladas al estilo de las expresiones regulares Perl
 - Encadenación de excepciones Permite a una excepción encapsular la excepción de bajo nivel original.
 - non-blocking NIO (New Input/Output) (Especificado en JSR 51.)
 - Logging API (Specified in JSR 47.)
 - API I/O para la lectura y escritura de imágenes en formatos como JPEG o PNG
 - Parser XML integrado y procesador XSLT (JAXP) (Especificado en JSR 5 y JSR 63.)
 - Seguridad integrada y extensiones criptográficas (JCE, JSSE, JAAS)

- Java Web Start incluido (El primer lanzamiento ocurrió en Marzo de 2001 para J2SE 1.3) (Especificado en JSR 56.)
- **J2SE 5.0** (30 de septiembre de 2004) — Nombre clave: *Tiger*. (Originalmente numerado 1.5, esta notación aún es usada internamente.[4]) Desarrollado bajo JSR 176, Tiger añadió un número significativo de nuevas características comunicado de prensa
 - Plantillas (genéricos) — provee conversión de tipos (type safety) en tiempo de compilación para colecciones y elimina la necesidad de la mayoría de conversión de tipos (type casting). (Especificado por JSR 14.)
 - Metadatos — también llamados anotaciones, permite a estructuras del lenguaje como las clases o los métodos, ser etiquetados con datos adicionales, que puedan ser procesados posteriormente por utilidades de proceso de metadatos. (Especificado por JSR 175.)
 - Autoboxing/unboxing — Conversiones automáticas entre tipos primitivos (Como los int) y clases de envoltura primitivas (Como Integer). (Especificado por JSR 201.)
 - Enumeraciones — la palabra reservada enum crea una typesafe, lista ordenada de valores (como Dia.LUNES, Dia.MARTES, etc.). Anteriormente, esto solo podía ser llevado a cabo por constantes enteras o clases construidas manualmente (enum pattern). (Especificado por JSR 201.)
 - Varargs (número de argumentos variable) — El último parámetro de un método puede ser declarado con el nombre del tipo seguido por tres puntos (e.g. void drawtext(String... lines)). En la llamada al método, puede usarse cualquier número de parámetros de ese tipo, que serán almacenados en un array para pasarlos al metodo.
 - Bucle for mejorado — La sintaxis para el bucle for se ha extendido con una sintaxis especial para iterar sobre cada miembro de un array o sobre cualquier clase que implemente Iterable, como la clase estándar Collection, de la siguiente forma:


```
void displayWidgets (Iterable<Widget> widgets) {  
    for (Widget w : widgets) {  
        w.display();  
    }  
}
```

Este ejemplo itera sobre el objeto Iterable widgets, asignando, en orden, cada uno de los elementos a la variable w, y llamando al método display() de cada uno de ellos. (Especificado por JSR 201.)

- **Java SE 6** (11 de diciembre de 2006) — Nombre clave *Mustang*. Estuvo en desarrollo bajo la JSR 270. En esta versión, Sun cambió el nombre "J2SE" por **Java SE** y eliminó el ".0" del número de versión.[5]. Está disponible en <http://java.sun.com/javase/6/>. Los cambios más importantes introducidos en esta versión son:
 - Incluye un nuevo marco de trabajo y APIs que hacen posible la combinación de Java con lenguajes dinámicos como PHP, Python, Ruby y JavaScript.
 - Incluye el motor Rhino, de Mozilla, una implementación de Javascript en Java.
 - Incluye un cliente completo de Servicios Web y soporta las últimas especificaciones para Servicios Web, como JAX-WS 2.0, JAXB 2.0, STAX y JAXP.
 - Mejoras en la interfaz gráfica y en el rendimiento.
- **Java SE 7** — Nombre clave *Dolphin*. En el año 2006 aún se encontraba en las primeras etapas de planificación. Se espera que su desarrollo dé comienzo en la primavera de 2006, y se estima su lanzamiento para 2008.
 - Soporte para XML dentro del propio lenguaje
 - Un nuevo concepto de superpaquete
 - Soporte para closures
 - Introducción de anotaciones estándar para detectar fallos en el software

Además de los cambios en el lenguaje, con el paso de los años se han efectuado muchos más cambios dramáticos en la librería de clases de Java (*Java class library*) que ha crecido de unos pocos cientos de clases en JDK 1.0 hasta más de tres mil en J2SE 5.0. APIs completamente nuevas, como Swing y Java2D, han sido introducidas y muchos de los métodos y clases originales de JDK 1.0 están desaprobados.

Filosofía

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Para conseguir la ejecución de código remoto y el soporte de red, los programadores de Java a veces recurren a extensiones como CORBA (Common Object Request Broker Architecture), Internet Communications Engine o OSGi respectivamente.

Orientado a Objetos

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que use estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio

en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y librerías de objetos.

Independencia de la plataforma

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Es lo que significa ser capaz de escribir un programa una vez y

que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “write once, run everywhere”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode)—instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (VM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran librerías adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCJ. Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

La licencia sobre Java de Sun insiste que todas las implementaciones sean “compatibles”. Esto dio lugar a una disputa legal entre Microsoft y Sun, cuando éste último alegó que la implementación de Microsoft no daba soporte a las interfaces RMI y JNI además de haber añadido características “dependientes” de su plataforma. Sun demandó a Microsoft y ganó por daños y perjuicios (unos 20 millones de dólares) así como una orden judicial forzando la acatación de la licencia de Sun. Como respuesta, Microsoft no ofrece Java con su versión de sistema operativo, y en recientes versiones de Windows, su navegador Internet Explorer no admite la ejecución de applets sin un conector (o plugin) aparte. Sin embargo, Sun y otras fuentes ofrecen versiones gratuitas para distintas versiones de Windows.

Las primeras implementaciones del lenguaje usaba una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad. Otra técnica, conocida como compilación JIT (Just In Time, o “compilación al vuelo”), convierte el bytecode a código nativo cuando se ejecuta la aplicación. Otras máquinas virtuales más sofisticadas usan una “recompilación dinámica” en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura, el gran número de estas con pequeños errores o inconsistencias llevan a que a veces se parodie el eslogan de Sun, "Write once, run anywhere" como "Write once, debug everywhere" (o “Escríbelo una vez, ejecútalo en todas partes” por “Escríbelo una vez, depúralo en todas partes”)

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios

Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

El recolector de basura

Un argumento en contra de lenguajes como C++ es que los programadores se encuentran con la carga añadida de tener que administrar la memoria de forma manual. En C++, el desarrollador debe asignar memoria en una zona conocida como *heap* (montículo) para crear cualquier objeto, y posteriormente desalojar el espacio asignado cuando desea borrarlo. Un olvido a la hora de desalojar memoria previamente solicitada, o si no lo hace en el instante oportuno, puede llevar a una *fuga de memoria*, ya que el sistema operativo piensa que esa zona de memoria está siendo usada por una aplicación cuando en realidad no es así. Así, un programa mal diseñado podría consumir una cantidad desproporcionada de memoria. Además, si una misma región de memoria es desalojada dos veces el programa puede volverse inestable y llevar a un eventual *cuelgue*.

En Java, este problema potencial es evitado en gran medida por el recolector automático de basura (o *automatic garbage collector*). El programador determina cuándo se crean los objetos y el entorno en tiempo de ejecución de Java (Java runtime) es el responsable de gestionar el ciclo de vida de los objetos. El programa, u otros objetos pueden tener localizado un objeto mediante una referencia a éste (que, desde un punto de vista de bajo nivel es una dirección de memoria). Cuando no quedan referencias a un objeto, el recolector de basura de Java borra el objeto, liberando así la memoria que ocupaba previniendo posibles fugas (ejemplo: un objeto creado y únicamente usado dentro de un método sólo tiene entidad dentro de éste; al salir del método el objeto es eliminado). Aún así, es posible que se produzcan fugas de memoria si el código almacena referencias a objetos que ya no son necesarios—es decir, pueden aún ocurrir, pero en un nivel conceptual superior. En definitiva, el recolector de basura de Java permite una fácil creación y eliminación de objetos, mayor seguridad y frecuentemente más rápida que en C++.

La recolección de basura de Java es un proceso prácticamente invisible al desarrollador. Es decir, el programador no tiene conciencia de cuándo la recolección de basura tendrá lugar, ya que ésta no tiene necesariamente que guardar relación con las acciones que realiza el código fuente.

Debe tenerse en cuenta que la memoria es sólo uno de los muchos recursos que deben ser gestionados.

Críticas

En 1995 se dijo que Java fue creado para abrir una nueva vía en la gestión de software complejo, y es por regla general aceptado que se ha comportado bien en ese aspecto. Sin embargo no puede decirse que Java no tenga grietas, ni que se adapta completamente a todos los estilos de programación, todos los entornos, o todas las necesidades.

General

- Java no ha aportado capacidades estándares para aritmética en punto flotante. El estándar IEEE 754 para “Estándar para Aritmética Binaria en Punto Flotante” apareció en 1985, y desde entonces es el estándar para la industria. Y aunque la aritmética flotante de Java (*cosa que cambió desde el 13 de Noviembre de 2006, cuando se abrió el código fuente y se adoptó la licencia GPL, aparte de la ya existente*) se basa en gran medida en la norma del IEEE, no soporta aún algunas características. Más información al respecto puede encontrarse en la sección final de enlaces externos.
- A raíz de la naturaleza propietaria de Java, la supuesta inflexibilidad para cambiar, y un creciente estrechamiento de líneas alrededor del sector corporativo, se dice que Java es “el nuevo COBOL”. Aunque puede ser una afirmación exagerada, hace referencia a una preocupación real sobre el panorama de futuro de Java.
- El recolector de basura de Java sólo gestiona la memoria, pero el instante en que tiene lugar su tarea no puede controlarse manualmente. Por tanto,

aquellos objetos que reservan recursos externos deben ser desalojados de memoria a mano (usando el mecanismo del lenguaje “finally”), o deben mantenerse hasta que acabe la ejecución del programa.

El lenguaje

- En un sentido estricto, Java no es un lenguaje absolutamente orientado a objetos, a diferencia de, por ejemplo, Ruby o Smalltalk. Por motivos de eficiencia, Java ha relajado en cierta medida el paradigma de orientación a objetos, y así por ejemplo, no todos los valores son objetos.
- Por el contrario, los programadores de C++ pueden caer en la confusión con Java, porque en éste los tipos primitivos son siempre variables automáticas, y los objetos siempre residen en el montículo (heap), mientras que en C++ ambos casos están en manos del programador, usando el operador new.
- El código Java puede ser a veces redundante en comparación con otros lenguajes. Esto es en parte debido a las frecuentes declaraciones de tipos y conversiones de tipo manual (casting). También se debe a que no se dispone de operadores sobrecargados, y a una sintaxis relativamente simple. Sin embargo, J2SE 5.0 introduce elementos para tratar de reducir la redundancia, como una nueva construcción para los bucles “foreach”.
- A diferencia de C++, Java no dispone de operadores de sobrecarga definidos por el usuario. Sin embargo esta fue una decisión de diseño que puede verse como una ventaja, ya que esta característica puede hacer los programas difíciles de leer y mantener.
- Java es un lenguaje basado en un solo paradigma.
- Java no permite herencia múltiple como otros lenguajes. Sin embargo el mecanismo de los interfaces de Java permite herencia múltiple de tipos y métodos abstractos.
- El soporte de Java para patrones de texto y la manipulación de éste no es tan potente como en lenguajes como Perl, Ruby o PHP, aunque J2SE 1.4 introdujo las expresiones regulares.

Apariencia

La apariencia externa (el “look and feel”) de las aplicaciones GUI (Graphical User Interface) escritas en Java usando la plataforma Swing difiere a menudo de la que muestran aplicaciones nativas. Aunque el programador puede usar el juego de herramientas AWT (Abstract Windowing Toolkit) que genera objetos gráficos de la plataforma nativa, el AWT no es capaz de funciones gráficas avanzadas sin sacrificar la portabilidad entre plataformas; ya que cada una tiene un conjunto de APIs distinto, especialmente para objetos gráficos de alto nivel. Las herramientas de Swing, escritas completamente en Java, evitan este problema construyendo los objetos gráficos a partir de los mecanismos de dibujo básicos que deben estar disponibles en todas las plataformas. El inconveniente es el trabajo extra requerido para conseguir la misma apariencia de la plataforma destino. Aunque esto es posible (usando GTK+ y el Look-and-Feel de Windows), la mayoría de los usuarios no saben cómo cambiar la apariencia que se proporciona por defecto por aquella que se adapta a la de la plataforma. Mención merece la versión optimizada del Java Runtime que ha desarrollado Apple y que incluye en su sistema operativo, el Mac OS X. Por defecto implementa su propio look-and-feel (llamado Aqua), dando a las aplicaciones Swing ejecutadas en un Macintosh una apariencia similar a la que tendría si se hubiese escrito en código nativo.

Rendimiento

El rendimiento de una aplicación está determinado por multitud de factores, por lo que no es fácil hacer una comparación que resulte totalmente objetiva. En tiempo de ejecución, el rendimiento de una aplicación Java depende más de la eficiencia del compilador, o la JVM, que de las propiedades intrínsecas del lenguaje. El bytecode de Java puede ser interpretado en tiempo de ejecución por la máquina virtual, o bien compilado al cargarse el programa, o durante la propia ejecución, para generar código nativo que se ejecuta directamente sobre el hardware. Si es interpretado, será más lento que usando el código máquina intrínseco de la plataforma destino. Si es compilado, durante la carga inicial o la ejecución, la penalización está en el tiempo necesario para llevar a cabo la compilación.

Algunas características del propio lenguaje conllevan una penalización en tiempo, aunque no son únicas de Java. Algunas de ellas son el chequeo de los límites de arrays, chequeo en tiempo de ejecución de tipos, y la indirección de funciones virtuales.

El uso de un recolector de basura para eliminar de forma automática aquellos objetos no requeridos, añade una sobrecarga que puede afectar al rendimiento, o ser apenas apreciable, dependiendo de la tecnología del recolector y de la aplicación en concreto. Las JVM modernas usan recolectores de basura que gracias a rápidos algoritmos de manejo de memoria, consiguen que algunas aplicaciones puedan ejecutarse más eficientemente.

El rendimiento entre un compilador JIT y los compiladores nativos puede ser parecido, aunque la distinción no está clara en este punto. La compilación mediante el JIT puede consumir un tiempo apreciable, un inconveniente principalmente para aplicaciones de corta duración o con gran cantidad de código. Sin embargo, una vez compilado, el rendimiento del programa puede ser comparable al que consiguen compiladores nativos de la plataforma destino, inclusive en tareas numéricas. Aunque Java no permite la expansión manual de llamadas a métodos, muchos compiladores JIT realizan esta optimización durante la carga de la aplicación y pueden aprovechar información del entorno en tiempo de ejecución para llevar a cabo transformaciones eficientes durante la propia ejecución de la aplicación. Esta recompilación dinámica, como la que proporciona la máquina virtual HotSpot de Sun, puede llegar a mejorar el resultado de compiladores estáticos tradicionales, gracias a los datos que sólo están disponibles durante el tiempo de ejecución.

Java fue diseñado para ofrecer seguridad y portabilidad, y no ofrece acceso directo al hardware de la arquitectura ni al espacio de direcciones. Java no soporta expansión de código ensamblador, aunque las aplicaciones pueden acceder a características de bajo nivel usando librerías nativas (JNI, Java Native Interfaces).

Sintaxis

La sintaxis de Java se deriva en gran medida de C++. Pero a diferencia de éste, que combina la sintaxis para programación genérica, estructurada y orientada a objetos, Java fue construido desde el principio para ser completamente orientado a objetos. Todo en Java es un objeto (salvo algunas excepciones), y todo en Java reside en alguna clase.

Para que esto quede explicado detalladamente, presentare el siguiente ejemplo, el famoso “**Hola Mundo**” que no es mas que un programa que imprime el texto «¡Hola, mundo!» en un dispositivo de visualización (generalmente una pantalla de monitor). Se suele usar como introducción al estudio de un lenguaje de programación, siendo un primer ejercicio típico.

El *Hola Mundo* se caracteriza por su sencillez, especialmente cuando se utiliza en programas de línea de comandos. En interfaces gráficas este programa suele tener una complejidad mayor.

Aplicaciones autónomas

```
// Hola.java
public class Hola
{
    public static void main(String[] args) {
        System.out.println("Hola, mundo!");
    }
}
```

Este ejemplo necesita una pequeña explicación.

- Todo en Java está dentro de una clase, incluyendo programas autónomos.
- El código fuente se guarda en archivos con el mismo nombre que la clase que contienen y con extensión “.java”. Una clase (class) declarada pública (public) debe seguir este convenio. En el ejemplo anterior, la clase es *Hola*, por lo que el código fuente debe guardarse en el fichero “*Hola.java*”

- El compilador genera un archivo de clase (con extensión “.class”) por cada una de las clases definidas en el archivo fuente. Una clase anónima se trata como si su nombre fuera la concatenación del nombre de la clase que la encierra, el símbolo “\$”, y un número entero.
- Los programas que se ejecutan de forma independiente y autónoma, deben contener el método ”main()”.
- La palabra reservada ”void” indica que el método main no devuelve nada.
- El método main debe aceptar un array de objetos tipo String. Por acuerdo se referencia como ”args”, aunque puede emplearse cualquier otro identificador.
- La palabra reservada ”static” indica que el método es un método de clase, asociado a la clase en vez de a instancias de la misma. El método main debe ser estático o ”de clase”.
- La palabra reservada **public** significa que un método puede ser llamado desde otras clases, o que la clase puede ser usada por clases fuera de la jerarquía de la propia clase. Otros tipos de acceso son ”private” o ”protected”.
- La utilidad de impresión (en pantalla por ejemplo) forma parte de la librería estándar de Java: la clase ”System” define un campo público estático llamado ”out”. El objeto out es una instancia de ”PrintStream”, que ofrece el método ”println(String)” para volcar datos en la pantalla (la salida estándar).
- Las aplicaciones autónomas se ejecutan dando al entorno de ejecución de Java el nombre de la clase cuyo método main debe invocarse. Por ejemplo, una línea de comando (en Unix o Windows) de la forma java -cp . Hola ejecutará el programa del ejemplo (previamente compilado y generado “Hola.class”) . El nombre de la clase cuyo método main se llama puede especificarse también en el fichero “MANIFEST” del archivo de empaquetamiento de Java (.jar).

Aplicaciones con ventanas

Swing es la librería para la interfaz gráfica de usuario avanzada de la plataforma Java SE.

```
// Hola.java
import javax.swing.*;

public class Hola extends JFrame {
    Hola() {
        setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
        add(new JLabel("Hola, mundo!"));
        pack();
    }

    public static void main(String[] args) {
        new Hola().setVisible(true);
    }
}
```

Las sentencias **import** indican al compilador de Java que las clases e interfaces del paquete **javax.swing** se incluyan en la compilación.

La clase **Hola** extiende (**extends**) la clase **javax.swing.JFrame**, que implementa una ventana con una barra de título y un control para cerrarla.

El constructor **Hola()** inicializa el marco o frame llamando al método **setDefaultCloseOperation(int)** heredado de **JFrame** para establecer las operaciones por defecto cuando el control de cierre en la barra de título es seleccionado al valor **WindowConstants.DISPOSE_ON_CLOSE**. Esto hace que se liberen los recursos tomados por la ventana cuando es cerrada, y no simplemente ocultada, lo que permite a la máquina virtual y al programa acabar su ejecución. A continuación se crea un objeto de tipo **JLabel** con el texto "Hola, mundo!", y se añade al marco mediante el método **add(Component)**, heredado de la clase **Container**. El método **pack()**, heredado de la clase **Window**, es invocado para dimensionar la ventana y distribuir su contenido.

El método **main()** es llamado por la JVM al comienzo del programa. Crea una instancia de la clase **Hola** y hace la ventana sea mostrada invocando al método

setVisible(boolean) de la superclase (clase de la que hereda) con el parámetro a true. Véase que, una vez el marco es dibujado, el programa no termina cuando se sale del método **main()**, ya que el código del que depende se encuentra en un hilo de ejecución independiente ya lanzado, y que permanecerá activo hasta que todas las ventanas hayan sido destruidas.

Recursos

JRE

El **JRE** (Java Runtime Environment, o Entorno en Tiempo de Ejecución de Java) es el software necesario para ejecutar cualquier aplicación desarrollada para la plataforma Java. El usuario final usa el JRE como parte de paquetes software o plugins (o conectores) en un navegador Web. Sun ofrece también el SDK de Java 2, o JDK (Java Development Kit) en cuyo seno reside el JRE, e incluye herramientas como el compilador de Java, Javadoc para generar documentación o el depurador. Puede también obtenerse como un paquete independiente, y puede considerarse como el entorno necesario para ejecutar una aplicación Java, mientras que un desarrollador debe además contar con otras facilidades que ofrece el JDK.

Componentes

- Bibliotecas de Java, que son el resultado de compilar el código fuente desarrollado por quien implementa la JRE, y que ofrecen apoyo para el desarrollo en Java. Algunos ejemplos de estas librerías son:
 - Las bibliotecas centrales, que incluyen:
 - Una colección de bibliotecas para implementar estructuras de datos como listas, arrays, árboles y conjuntos.
 - Bibliotecas para análisis de XML.
 - Seguridad.
 - Bibliotecas de internacionalización y localización.
 - Bibliotecas de integración, que permiten la comunicación con sistemas externos. Estas librerías incluyen:

- La API para acceso a bases de datos JDBC (Java DataBase Connectivity).
- La interfaz JNDI (Java Naming and Directory Interface) para servicios de directorio.
- RMI (Remote Method Invocation) y CORBA para el desarrollo de aplicaciones distribuidas.
- Bibliotecas para la interfaz de usuario, que incluyen:
 - El conjunto de herramientas nativas AWT (Abstract Windowing Toolkit), que ofrece componentes GUI (Graphical User Interface), mecanismos para usarlos y manejar sus eventos asociados.
 - Las Bibliotecas de Swing, construidas sobre AWT pero ofrecen implementaciones no nativas de los componentes de AWT.
 - APIs para la captura, procesamiento y reproducción de audio.
- Una implementación dependiente de la plataforma en que se ejecuta de la máquina virtual de Java (JVM), que es la encargada de la ejecución del código de las librerías y las aplicaciones externas.
- Plugins o conectores que permiten ejecutar applets en los navegadores Web.
- Java Web Start, para la distribución de aplicaciones Java a través de Internet.
- Documentación y licencia.

APIs

Sun define tres plataformas en un intento por cubrir distintos entornos de aplicación. Así, ha distribuido muchas de sus APIs (Application Program Interface) de forma que pertenezcan a cada una de las plataformas:

- Java ME (Java Platform, Micro Edition) o J2ME — orientada a entornos de limitados recursos, como teléfonos móviles, PDAs (Personal Digital Assistant), etc.

- Java SE (Java Platform, Standard Edition) o J2SE — para entornos de gama media y estaciones de trabajo. Aquí se sitúa al usuario medio en un PC de escritorio.
- Java EE (Java Platform, Enterprise Edition) o J2EE — orientada a entornos distribuidos empresariales o de Internet.

Las clases en las APIs de Java se organizan en grupos disjuntos llamados **paquetes**. Cada paquete contiene un conjunto de interfaces, clases y excepciones relacionadas. La información sobre los paquetes que ofrece cada plataforma puede encontrarse en la documentación de ésta.

El conjunto de las APIs es controlado por Sun Microsystems junto con otras entidades o personas a través del programa JCP (Java Community Process). Las compañías o individuos participantes del JCP pueden influir de forma activa en el diseño y desarrollo de las APIs, algo que ha sido motivo de controversia.

En 2004, IBM y BEA apoyaron públicamente la idea de crear una implementación de código abierto (open source) de Java, algo a lo que Sun, a fecha de 2006, se ha negado.

Extensiones y arquitecturas relacionadas

Las extensiones de Java están en paquetes que cuelgan de la raíz javax: javax.*. No se incluyen en la JDK o el JRE. Algunas de las extensiones y arquitecturas ligadas estrechamente al lenguaje Java son:

- Java EE (Java Platform, Enterprise Edition; antes J2EE) —para aplicaciones distribuidas orientadas al entorno empresarial
- Java ME (Java Platform, Micro Edition; antes J2ME)—para dispositivos de recursos limitados como teléfonos móviles y PDAs
- JMF (Java Media Framework)
- JNDI (Java Naming and Directory Interface)
- JSML (Java Speech API Markup Language)
- JDBC (Java Database Connectivity)
- JDO (Java Data Objects)

- JAI (Java Advanced Imaging)
- JAIN (Java API for Integrated Networks)
- JDMK (Java Dynamic Management Kit)
- Jini (una arquitectura de red para la construcción de sistemas distribuidos)
- Jiro
- Java Card
- JavaSpaces
- JML (Java Modeling Language)
- JMI (Java Metadata Interface)
- JMX (Java Management Extensions)
- JSP (JavaServer Pages)
- JSF (JavaServer Faces)
- JNI (Java Native Interface)
- JXTA (Protocolos abiertos para redes virtuales Peer-to-Peer o P2P)
- Java 3D (Una API de alto nivel para programación gráfica en 3D)
- JOGL (Java OpenGL—Una API de bajo nivel para programación gráfica usando OpenGL)
- LWJGL (Light Weight Java Game Library—Una API de bajo nivel para acceso a OpenGL, OpenAL y varios dispositivos de entrada)
- OSGi (Dynamic Service Management and Remote Maintenance)

Java en código abierto

¿Hasta donde Java es Software libre?

En diciembre de 2006, Sun está en pleno relanzamiento de su plataforma Java bajo la licencia GPL de GNU. Cuando este cambio de licencia haya terminado, esperamos que Java siga siendo un recurso amigable para todo tipo de usuarios.

Compromiso de Sun Microsystems con el código abierto

La importancia del código abierto en relación con Java puede verse entre otras cosas, en que el presidente y CEO de Sun, Jonathan Schwartz, ha retado a la compañía a que ofrezca código abierto para todo el software que produce Sun

(fuente), Sun ya hace mucho tiempo que empezó a apostar por el código abierto cuando liberó StarOffice (Llamado a ahora OpenOffice).

Sun ha aportado más líneas de código abierto que cualquier otra organización (fuente) También Richard Stallman opina eso mismo (fuente)

- El éxito del código abierto - Artículo puesto en el sito oficial en Español de Sun Microsystems, en el que habla de la apuesta de Sun Microsystems por el código abierto.

3.3 ¿Qué es MySQL?

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Aunque MySQL es software libre, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

3.3.1 Historia de MySQL

MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por

lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

Se comenta en el manual [MySQL_Manual] que no se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

La versión estable de este gestor a días de hoy es la 3.23.49. Se puede encontrar más información sobre este gestor en el manual [MySQL_Manual].

3.3.2 Características de MySQL

Las principales características de este gestor de bases de datos son las siguientes:

1. Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.
 2. Soporta gran cantidad de tipos de datos para las columnas.
 3. Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
 4. Gran portabilidad entre sistemas.
 5. Soporta hasta 32 índices por tabla.
 6. Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- MySQL es un sistema de gestión de bases de datos

Una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compra a una galería de pintura o las más vastas cantidades de información en una red corporativa. Para añadir, acceder, y procesar los datos almacenados en una base de datos, necesita un sistema de gestión de base de datos como MySQL Server. Al ser los computadores muy buenos en tratar grandes cantidades de datos, los sistemas de gestión de bases de datos juegan un papel central en

computación, como aplicaciones autónomas o como parte de otras aplicaciones.

- MySQL es un sistema de gestión de bases de datos relacionales

Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad. La parte SQL de "MySQL" se refiere a "Structured Query Language". SQL es el lenguaje estandarizado más común para acceder a bases de datos y está definido por el estándar ANSI/ISO SQL. El estándar SQL ha evolucionado desde 1986 y existen varias versiones. En este manual, "SQL-92" se refiere al estándar del 1992, "SQL:1999" se refiere a la versión del 1999, y "SQL:2003" se refiere a la versión actual del estándar. Usamos la frase "el estándar SQL" para referirnos a la versión actual de SQL.

- MySQL software es Open Source.

Open Source significa que es posible para cualquiera usar y modificar el software. Cualquiera puede bajar el software MySQL desde internet y usarlo sin pagar nada. Si lo desea, puede estudiar el código fuente y cambiarlo para adaptarlo a sus necesidades. El software MySQL usa la licencia GPL (GNU General Public License), <http://www.fsf.org/licenses/>, para definir lo que puede y no puede hacer con el software en diferentes situaciones. Si no se encuentra cómodo con la GPL o necesita añadir código MySQL en una aplicación comercial, puede comprarnos una licencia comercial.

El servidor de base de datos MySQL es muy rápido, fiable y fácil de usar.

- Seguridad
 - Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las

contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

- Escalabilidad y límites
 - Soporte a grandes bases de datos. Usamos MySQL Server con bases de datos que contienen 50 millones de registros. También conocemos usuarios que usan MySQL Server con 60.000 tablas y acerca de 5.000.000 de registros.
 - Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna **CHAR**, **VARCHAR**, **BLOB**, o **TEXT**.

- Conectividad
 - Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT,2000,XP, o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros socket Unix.
 - En MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción **--shared-memory**. Los clientes pueden conectar a través de memoria compartida usando la opción **--protocol=memory**.
 - La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity). Por ejemplo, puede usar MS Access para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows o Unix. El código fuente de MyODBC está disponible. Todas las funciones para ODBC 2.5 están soportadas,

así como muchas otras. Consulte Sección 25.1, “MySQL Connector/ODBC”.

- La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible. Consulte Sección 25.4, “MySQL Connector/J”.

- Localización
 - El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas. Consulte Sección 5.9.2, “Escoger el idioma de los mensajes de error”.
 - Soporte completo para distintos conjuntos de caracteres, incluyendo **latin1** (ISO-8859-1), **german**, **big5**, **ujis**, y más. Por ejemplo, los caracteres escandinavos 'â', 'ä' y 'ö' están permitidos en nombres de tablas y columnas. El soporte para Unicode está disponible
 - Todos los datos se guardan en el conjunto de caracteres elegido. Todas las comparaciones para columnas normales de cadenas de caracteres son case-insensitive.
 - La ordenación se realiza acorde al conjunto de caracteres elegido (usando colación Sueca por defecto). Es posible cambiarla cuando arranca el servidor MySQL. Para ver un ejemplo de ordenación muy avanzada, consulte el código Checo de ordenación. MySQL Server soporta diferentes conjuntos de caracteres que deben ser especificados en tiempo de compilación y de ejecución.

A partir de MySQL 5.0, se usa el motor de almacenamiento **MyISAM**, el máximo tamaño de las tablas es de 65536 terabytes ($256^7 - 1$ bytes). Por lo tanto, el tamaño efectivo máximo para las bases de datos en MySQL usualmente

los determinan los límites de tamaño de ficheros del sistema operativo, y no por límites internos de MySQL.

El motor de almacenamiento **InnoDB** mantiene las tablas en un espacio que puede ser creado a partir de varios ficheros. Esto permite que una tabla supere el tamaño máximo individual de un fichero. Este espacio puede incluir particiones de disco, lo que permite tablas extremadamente grandes. El tamaño máximo del espacio de tablas es 64TB.

La siguiente tabla lista algunos ejemplos de límites de tamaño de ficheros de sistemas operativos. Esto es sólo una burda guía y no pretende ser definitiva. Para la información más actual, asegúrese de consultar la documentación específica de su sistema operativo.

Sistema operativo	Tamaño máximo de fichero
Linux 2.2-Intel 32-bit	2GB (LFS: 4GB)
Linux 2.4	(usando sistema de ficheros ext3) 4TB
Solaris 9/10	16TB
Sistema de ficheros NetWare w/NSS	8TB
win32 w/ FAT/FAT32	2GB/4GB
win32 w/ NTFS	2TB (posiblemente mayor)
MacOS X w/ HFS+	2TB

Por defecto, MySQL crea tablas **MyISAM** con una estructura interna que permite un tamaño máximo de unas 4GB.

CAPITULO 4

DISEÑO DEL PROGRAMA PROPUESTO

4.1 DEFINICION DE LAS TABLAS DE LAS SABADOS

4.1.1 TABLA DE PAGOS

4.1.2 TABLA DE RECIBOS

4.1.3 TABLA DE CONCEPTOS

4.2 CAPTURA DE PANTALLAS DEL SIAU

4.3 ADMINISTRACION DEL SISTEMA

DISEÑO DEL PROGRAMA PROPUESTO

Como ya se menciona en el capítulo correspondiente al planteamiento del problema, La Universidad Istmo Americana A.C, opera sus actividades académicas y administrativas hacia los alumnos mediante un sistema de control escolar llamado ESCOLA, por medio del cual se lleva a cabo el registro de cobros por los diferentes conceptos de los servicios que la misma ofrece.

Para esto se registra al alumno desde el momento en que pasa a inscribirse y, posteriormente se habilitan 12 columnas para el pago de colegiaturas y 4 extras para otros conceptos de pagos que vayan a realizarse durante el ciclo escolar.

4.1 DEFINICION DE LAS TABLAS DE LA BASE DE DATOS

La definición de la Base de Datos del módulo de pagos del Sistema Integral de Administración es a través de tres tablas que muestran la relación de los alumnos con los pagos de colegiaturas, exámenes y/o otros conceptos, así como las facturas que amparan lo anterior. A continuación se presenta la estructura de las tablas utilizadas por el SIAU:

4.1.1

1) Tabla de Pagos:

<i>Nombre del campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
id_pagos	Int	Clave del registro, autoincrementable y llave primaria.
Pagos_matricula	Varchar	Matricula del alumno.
Pagos_tipo_pago	Varchar	Tipo de pago (Colegiatura, inscripción, exámenes, etc.).
Pagos_mes_cargo	Varchar	Mes correspondiente al pago (Colegiaturas).
Pagos_cargo	Double	Monto del cargo de acuerdo al concepto.
Pago_abono	Double	Monto del pago realizado.
Pago_saldo	Double	Total del cargo menos el abono.

Pago_recargo	Double	Recargos por atraso en pagos.
Pago_carrera	Varchar	Carrera que cursa el alumno.
Pago_semestre	Varchar	Semestre en que esta inscrito el alumno.
Pago_curso	Varchar	Periodo en el que el alumno esta inscrito.
Pago_fecha_pago	Date	Fecha en que se realiza el pago.
Pago_nip	Varchar	NIP de control del alumno.
Pago_folio	Varchar	Folio del recibo (Factura)

4.1.2

2) Tabla de Recibos:

<i>Nombre del campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
Id_recibo	Int	Clave del registro, autoincrementable y llave primaria.
Folio_recibo	Varchar	Folio del recibo (Factura)
Matricula_recibo	Varchar	Matricula del alumno
Fecha_recibo	Varchar	Fecha en que se realiza el pago.
Importe_recibo	Double	Importe del los conceptos que se operen.
Recargo_recibo	Double	Monto del recargo.
Total_recibo	Double	Suma del importe más el recargo.
Mes_recibo	Varchar	Mes correspondiente al pago (Colegiaturas).
Tipo_cargo_recibo	Varchar	Clave del concepto de cargo.
Concepto_recibo	Varchar	Concepto del cargo.
Semestre_recibo	Varchar	Semestre en que esta inscrito el alumno.
Obs_recibos	Varchar	Observaciones
Nombre_recibo	Varchar	Nombre del alumno.
Estado_recibo	Varchar	Estado del recibo.
Ciclo_recibo	Varchar	Periodo en el que el alumno esta inscrito.

4.1.3

3) Conceptos:

<i>Nombre del campo</i>	<i>Tipo de dato</i>	<i>Descripción</i>
Id_concepto	Int	Clave del registro, autoincrementable y llave primaria.
Nombre_concepto	Varchar	Concepto del cargo.
Costo_concepto	Double	Monto del concepto.

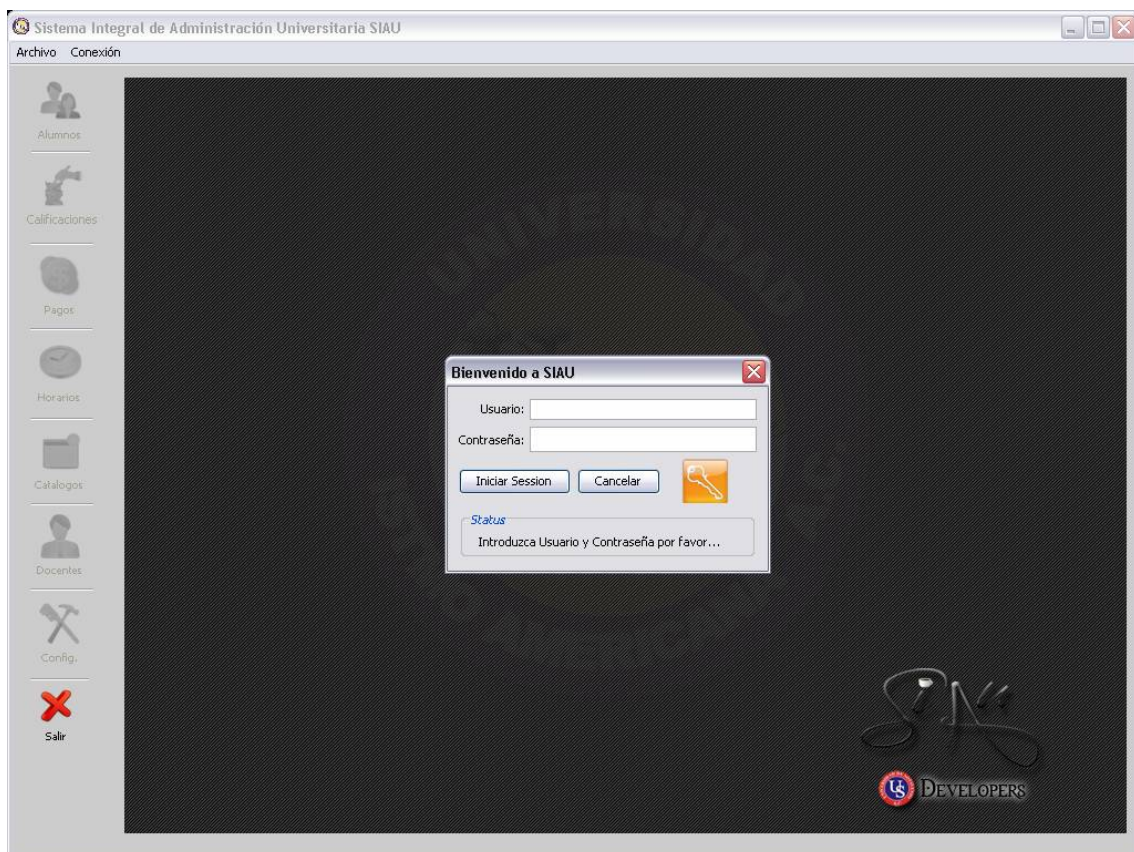
Debido al rediseño del sistema de control escolar, realizado en este trabajo, surge la necesidad de crear tablas diferentes a las que se encontraban operando, todo esto con la finalidad de mejorar el funcionamiento y facilitar la generación de reportes a los usuarios del sistema propuesto (SIAU).

Cabe mencionar que el problema mas presente en este sentido es la inexactitud de la información que se proporciona al querer obtener el monto total de los saldos (adeudos) de los alumnos.

4.2 CAPTURA DE PANTALLAS DEL SIAU

En este apartado se presentan las pantallas con las cuales se lleva a cabo la operación del SIAU, mismas que fueron creadas en JAVA, y están ordenadas de acuerdo a la importancia y el uso que siguen durante la operación del sistema.

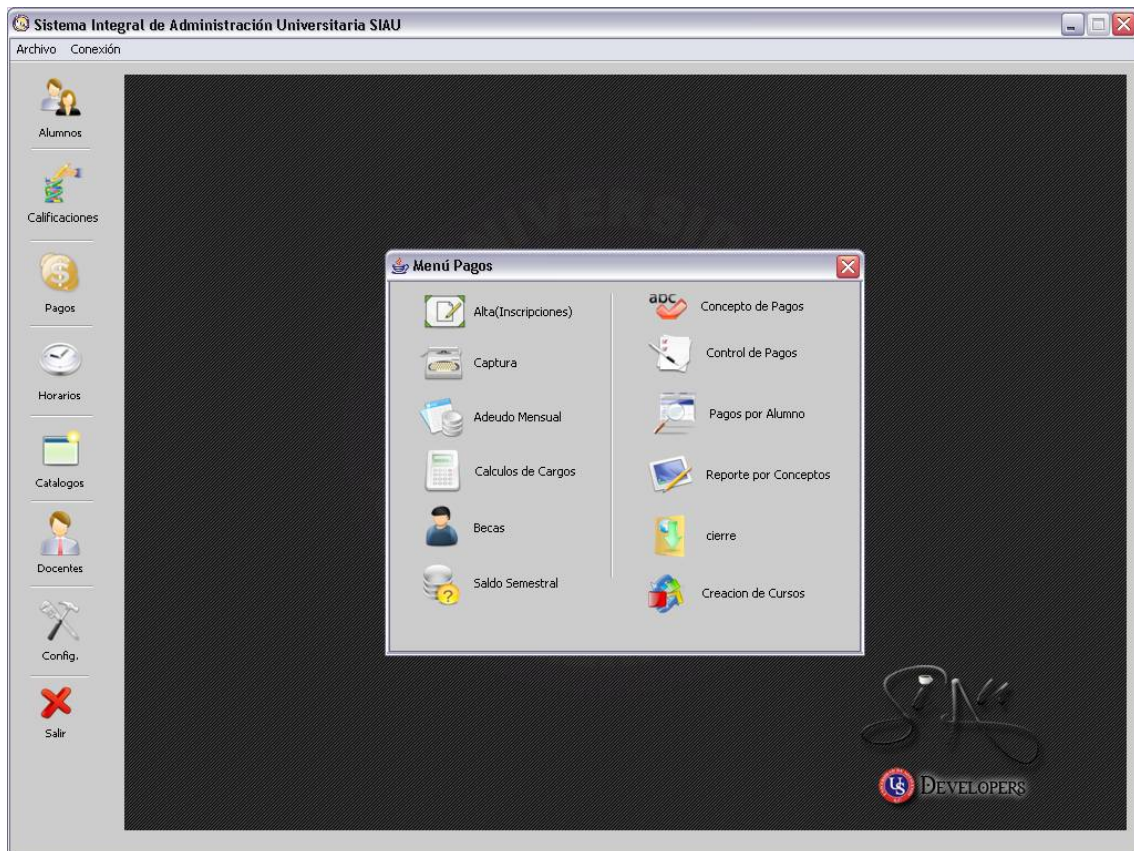
Pantalla de Inicio del SIAU.



Ventana principal del SIAU .El usuario (previamente registrado) ingresara su nombre de usuario y contraseña al sistema, una vez hecho esto, este mostrara el menú de junto con los submenús a los cuales tendrá acceso según la contraseña indicada (Siendo el presente caso el del menú de pagos)

Menú Pagos.

Ventana principal del menú de pagos. Desde las múltiples opciones que se presenta se tiene acceso para realizar todo tipo de actividades relacionadas con el cobro a alumnos, así como los distintos reportes que se generan utilizando dicha información.



Registro de Alumnos.

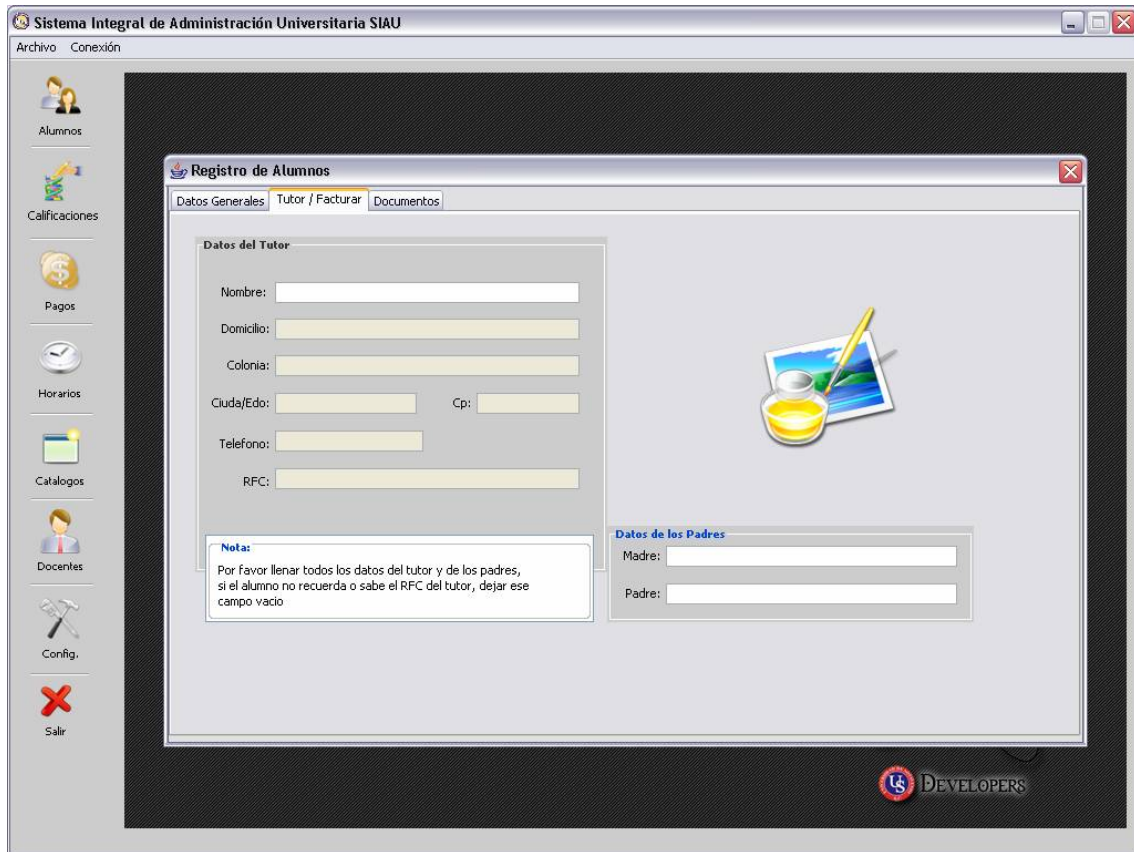
CAPÍTULO 4

En la siguiente ventana se realiza el registro de alumnos nuevos (inscripción), es aquí donde se solicita toda su información personal, así como su documentación, y los datos de la persona física o moral a la que se va a facturar (si es el caso)

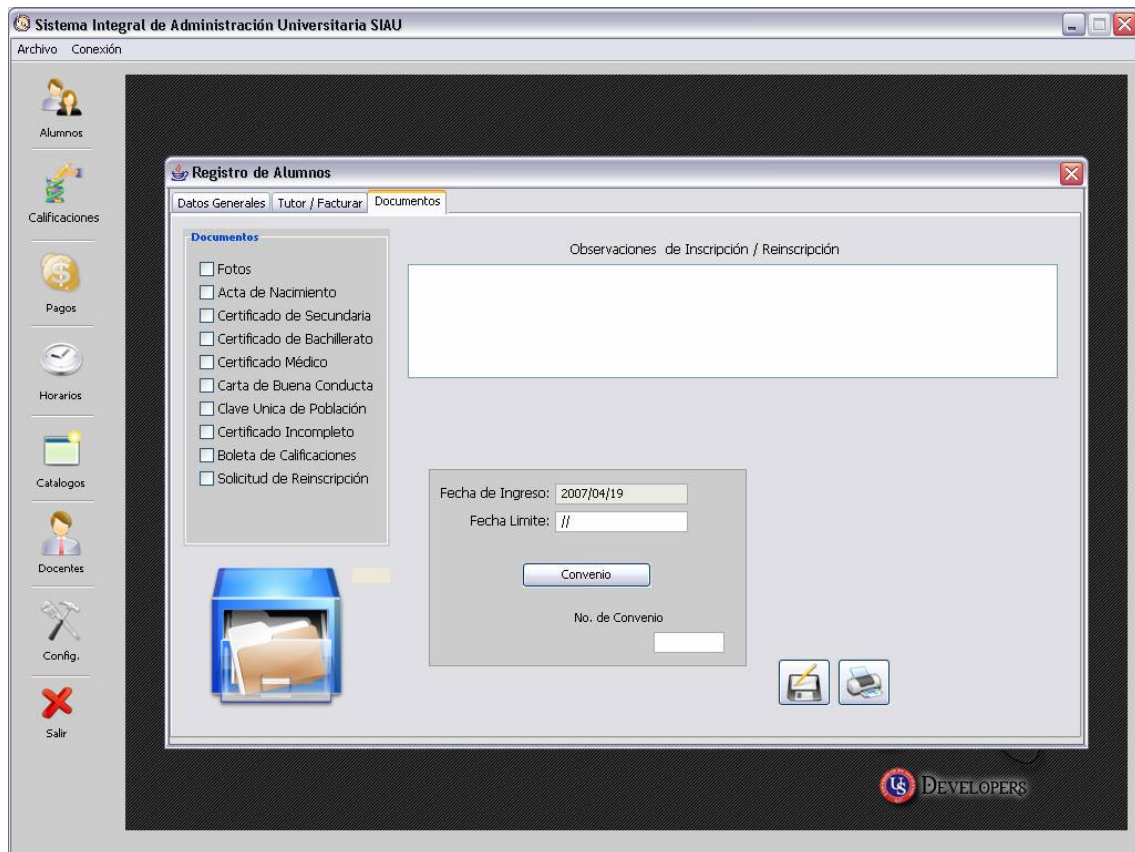
The screenshot shows the 'Registro de Alumnos' window in the SIAU system. The window is titled 'Registro de Alumnos' and has three tabs: 'Datos Generales', 'Tutor / Facturar', and 'Documentos'. The 'Datos Generales' tab is active. It contains several input fields: 'Matricula: 409415', 'NIP: 1', 'Fecha de Inscripción: 2007/04/19', 'Alumno:', 'Domicilio:', 'Telefono:', 'Nacimiento: //', 'Colonias:', 'Ciudad/Edo: Coatzacoalcos, Ver', 'Cp:', 'Edo. Civil' (with radio buttons for Soltero, Casado, Divorciado, Viudo, Unión Libre), 'Sexo: Femenino (checked), Masculino', 'Folio de la Escuela: [input] [checkmark]', 'Edad: [input]', 'Nombre de la Preparatoria: [input]', 'Nacionalidad: Mexicana', 'Año de Egreso: [input]', 'E-mail: [input]', 'Carrera: LICENCIATURA EN MERCADOTECNIA', 'Semestre: 1', 'Grupo: 1010', 'Sistema: SEP (checked), SEC', 'Turno: Matutino (checked), Vespertino, Especial', 'Carp: [input]', 'Generación: [input]'. A 'Salir' button is at the bottom left. A 'DEVELOPERS' logo is at the bottom right.

En esta sección del sistema, el alumno proporcionara sus datos generales como son: domicilio, edad, sexo, escuela de procedencia, nacionalidad, año de egreso, así como al carrera en la que se inscribe, el grupo y turno que le sea asignado.

Asi mismo se le genera una matricula y un NIP, ambos son de manera consecutiva y no dependen el uno del otro, la matricula sirve para el registro académico del alumno, mientras que el nip es utilizado para el control de los pagos. Todo esto obedece a que el registro con las matriculas oficiales otorgadas por la Secretaria de Educación presentan un desfase en su tiempo de entrega.



Posteriormente proporcionara los datos se su tutor o nombre de la persona a la cual se facturará. Aunque en este apartado se proporcionan los datos que deben imprimirse en la factura, al momento de realizar el pago, se puede escoger un registro distinto al cual se le realizara la factura.



En caso de no contar con la documentación correspondiente, el nuevo alumno deberá fijar una fecha, en la cual se compromete a entregar el o los documentos que resulten faltantes. Así mismo, el usuario del SIAU deberá anotar todo esto en el recuadro de observaciones, para posteriormente imprimir desde el mismo SIAU el convenio que deberá firmar el alumno, liberando así a la Universidad Istmo Americana de cualquier responsabilidad educativa en caso no entregar su documentación completa en la fecha pactada.

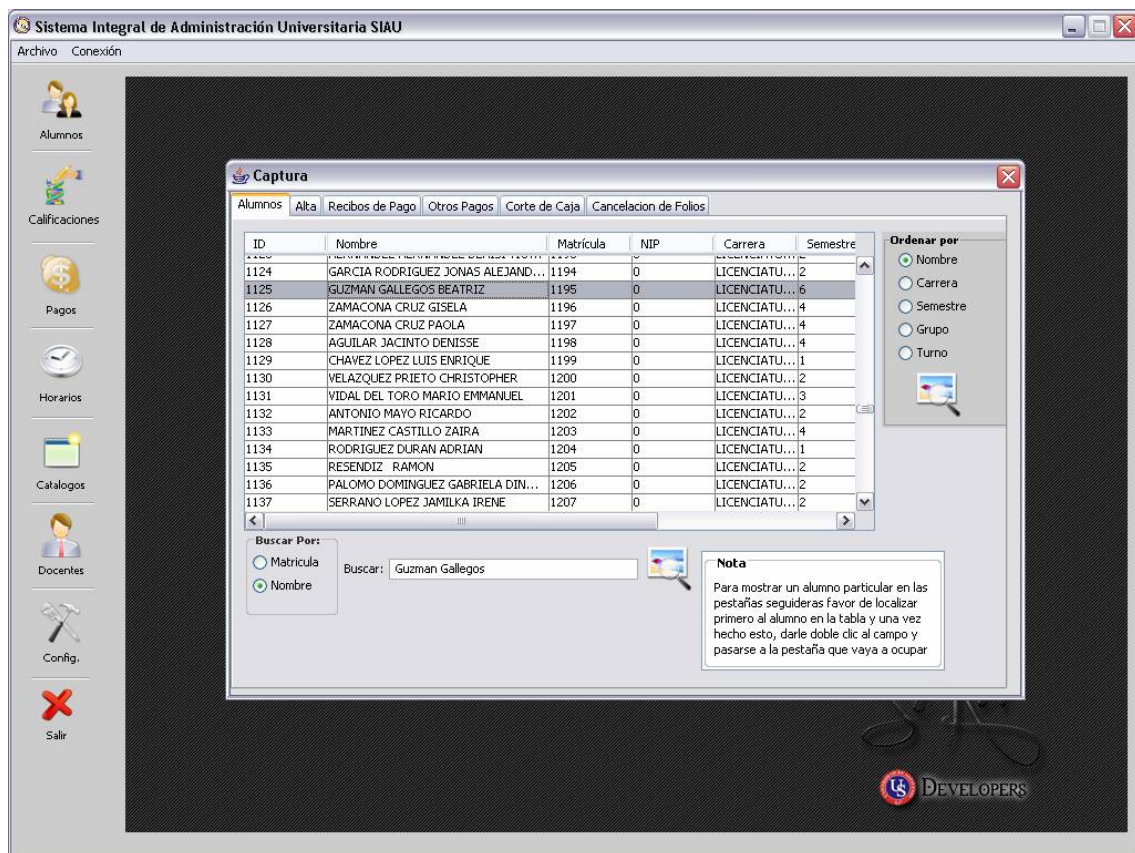
Para finalizar el proceso de la nueva inscripción, la persona encargada de dicho registro se encarga de verificar que el nuevo alumno cumpla con toda la documentación solicitada por la Universidad Istmo Americana y procede a guardar todos los datos e imprimir su hoja de registro.

Cabe mencionar que aunque este procedimiento es mas académico que administrativo, se lleva a cabo en la opción de pagos, ya que el alumno inmediatamente después de haber quedado inscrito tiene que pagar por el concepto de inscripción y/o

colegiaturas, por lo tanto una vez sus datos han sido guardados en el SIUA, de manera automática le son generados los cargos correspondientes al periodo (semestre) que va a cursar.

Captura.

Dentro de esta ventana, se procede al cobro de los diversos servicios que ofrece la Universidad Istmo Americana, A.C. En el caso de los alumnos de nuevo ingreso, tendrán generados los cargos desde el momento en que termina el proceso de inscripción (7 uno por cada de las 6 colegiaturas y 1 mas por el concepto de inscripción). Para los alumnos de reingreso este proceso se realiza al finalizar el periodo escolar y únicamente se corre la rutina una vez cada semestre, generando así los cargos del nuevo semestre para todos los alumnos vigentes.



Una vez realizado el proceso de generación de cargos por cualquiera de los dos métodos (inscripción o reinscripción) se procede a realizar los cobros por los servicios solicitados por los alumnos, en la ventana se despliega una lista con todos los alumnos existentes en la base de datos ordenados de manera alfabética.

El SIAU cuenta con cinco formas de ordenar a los alumnos: por Nombre, Carrera, Semestre, Grupo o Turno, así mismo cuenta con dos formas de hacer búsquedas específicas de un alumno, estas pueden ser por matricula o por nombre (apellidos).

Sistema Integral de Administración Universitaria SIAU

Archivo Conexión

Alumnos

Calificaciones

Pagos

Horarios

Catalogos

Docentes

Config.

Salir

Captura

Alumnos Alta Recibos de Pago Otros Pagos Corte de Caja Cancelacion de Folios

Matricula: 1195 N.I.P. 0

Carrera: LICENCIATURA EN PEDAGOGIA Semestre: 6 Grupo: 6520

Alumno: GUZMAN GALLEGOS BEATRIZ Estado: Alumno

Edad: * Sexo:

Turno

Matutino

Vespertino

Especial

Sistema

SEP

SEC

Domicilio: AV. VERACRUZ NO. 300.

Colonia: TIERRA Y LIBERTAD C.p. 96580

Ciudad: COATZACOALCOS, VER.

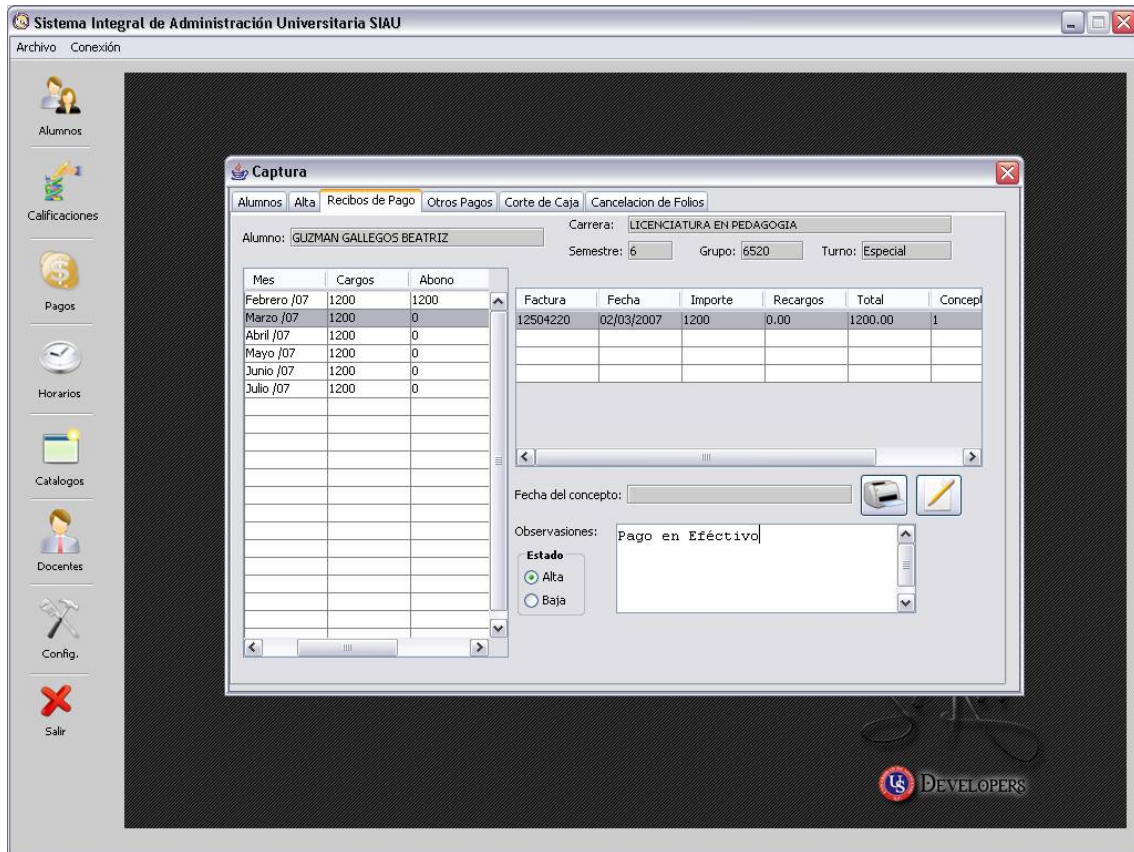
Telefono: 2137253

DEVELOPERS

Toda vez que se ha buscado un alumno de la tabla anterior, se procede a seleccionarlo y pasar a la pestaña de “Alta” en esta opción es donde se procede a la reinscripción de los alumnos o algún cambio en alguno de sus datos generales.

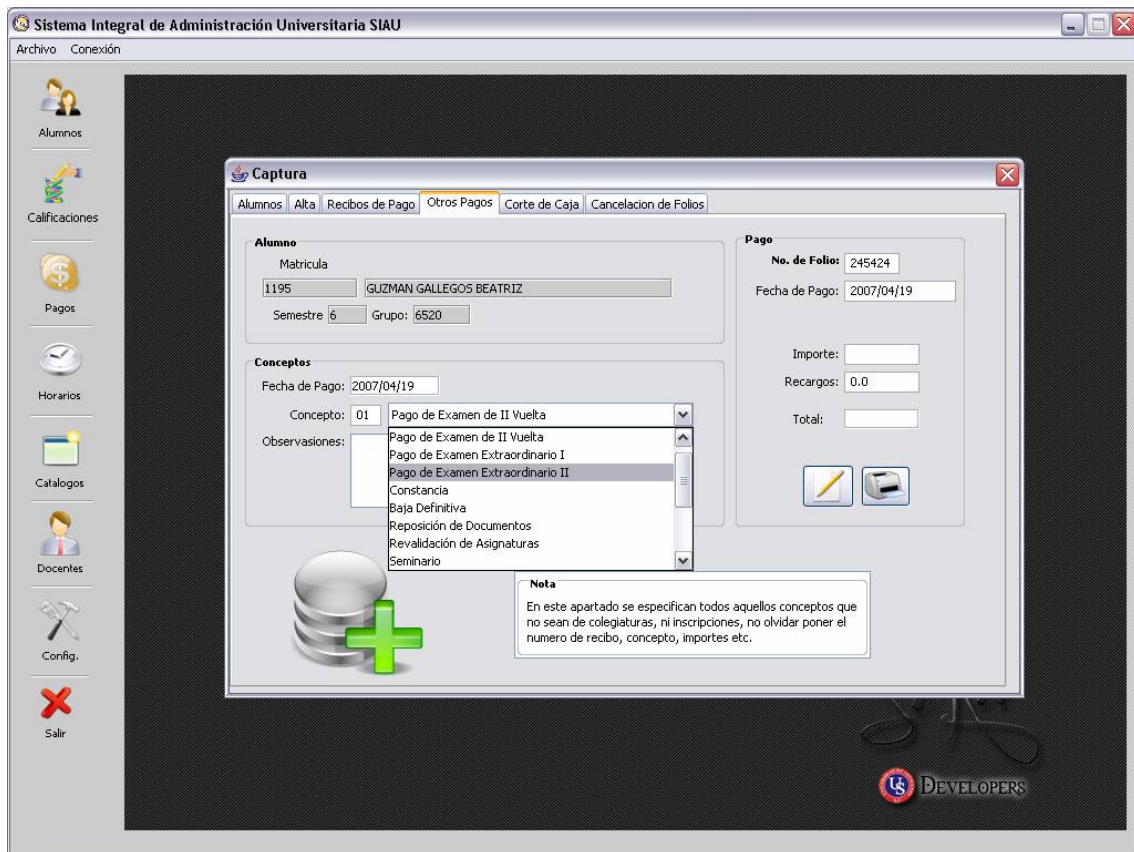
Generalmente el alumno cambia de semestre mediante la opción de Cierre Semestral, pero debido a circunstancias ajenas al usuario es importante el poder realizar

algún cambio de semestre, grupo o turno del alumno, así como sus datos personales más importantes.



En esta ventana (Recibos de pago) dentro del menú CAPTURA, se realizan los cobros de colegiaturas a los alumnos, como ya se mencionó, al inscribirse o al generar el ciclo escolar siguiente, se generan los cargos por concepto de inscripción y de incorporación, los cuales se despliegan en la tabla de la izquierda debiéndose seleccionar todos aquellos pagos que el alumno vaya a realizar para proceder al registro del monto a pagar e imprimir el recibo de pago.

El apartado observaciones funciona para detallar cualquier aclaración que deba hacerse en el recibo tales como si fue pago en efectivo, fecha de depósito en banco, si son recargos, si es pago parcial de un cargo, etc.

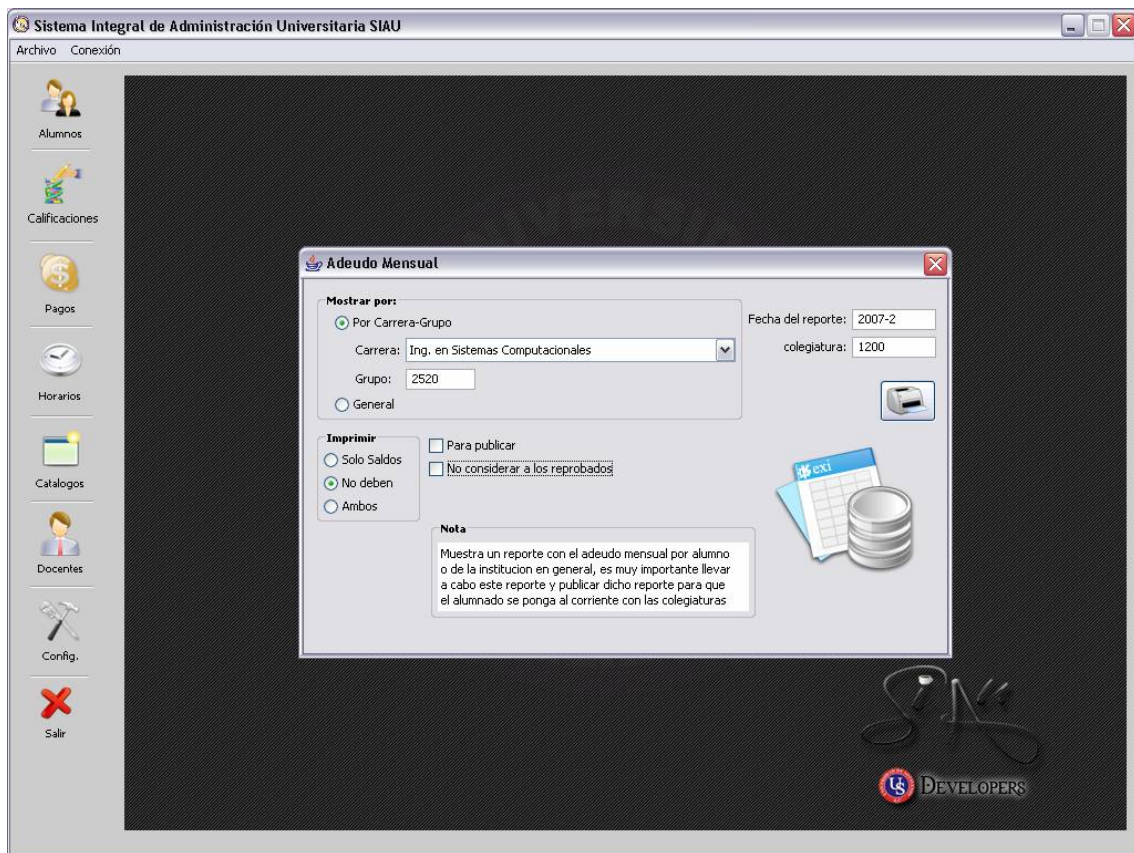


Otros Pagos.

Cuando se desea realizar un cobro por servicios diferentes a las colegiaturas como pagos de exámenes extraordinarios o de ingles, cursos de idiomas, conceptos por titulación, etc. Estos deben realizar en este apartado, el cual ya cuenta con una lista de todos los conceptos por los cuales el alumnos puede realizar un pago, todos ellos con sus respectivas cuotas previamente registradas en el SIAU en el apartado Conceptos dentro del menú pagos.

En caso que el alumno desee dar solo un pago parcial de un concepto o en su defecto requiera realizar mas de un pago (por ejemplo dos exámenes) esta situación deberá ser especificada en el campo de observaciones, que al igual que en la opción de pago de colegiaturas esta será impresa en recibo o factura.

El sistema calcula el total del pago de acuerdo a los pagos que se realizan y este se muestra en el campo total en pantalla y equivale al monto total que es impreso en el recibo o factura.

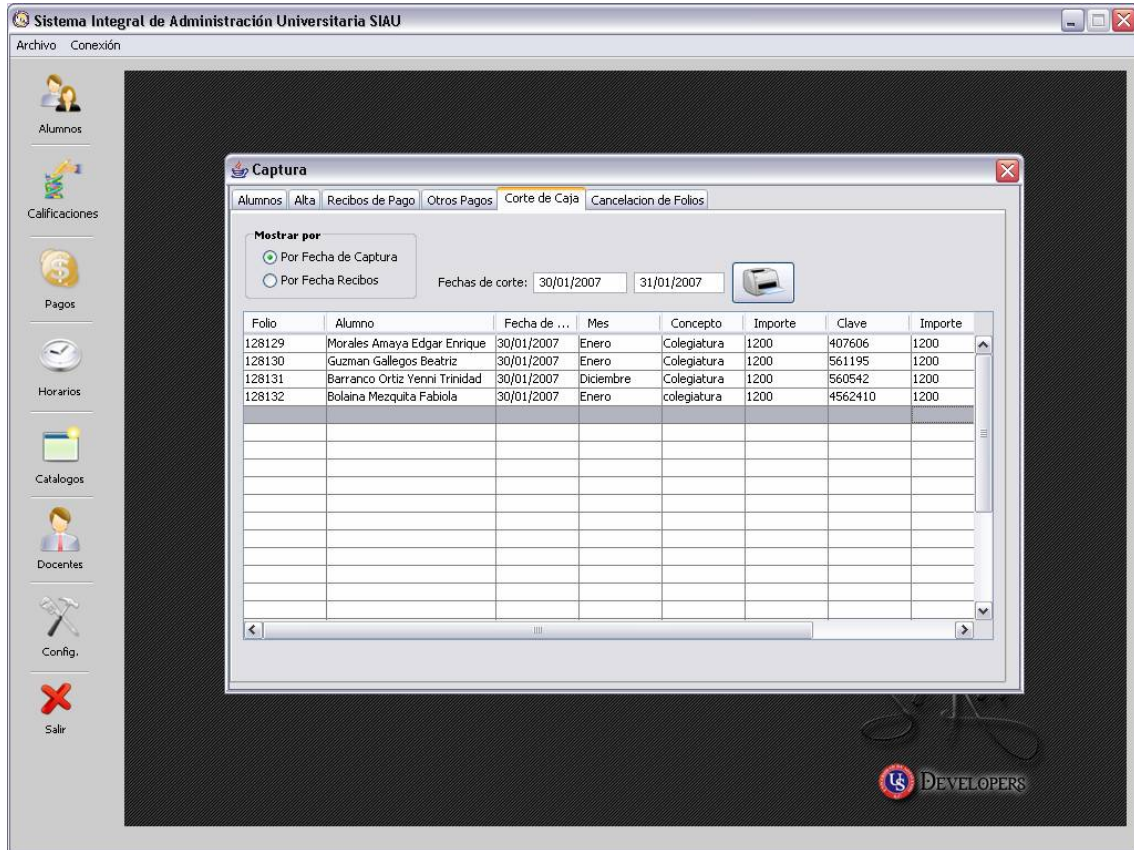


Adeudo Mensual.

En este sub-menú del menú pagos, se generan los saldos pendientes que presenten los alumnos. Estos reportes pueden ser por un grupo determinado o por toda una facultad.

Estos reportes son de gran utilidad para la dirección administrativa, ya que mediante estos obtiene la información sobre el estado financiero que presentan los alumnos y determinar así si el alumno tiene o no derecho a la continuidad de los servicios que ofrece la Universidad.

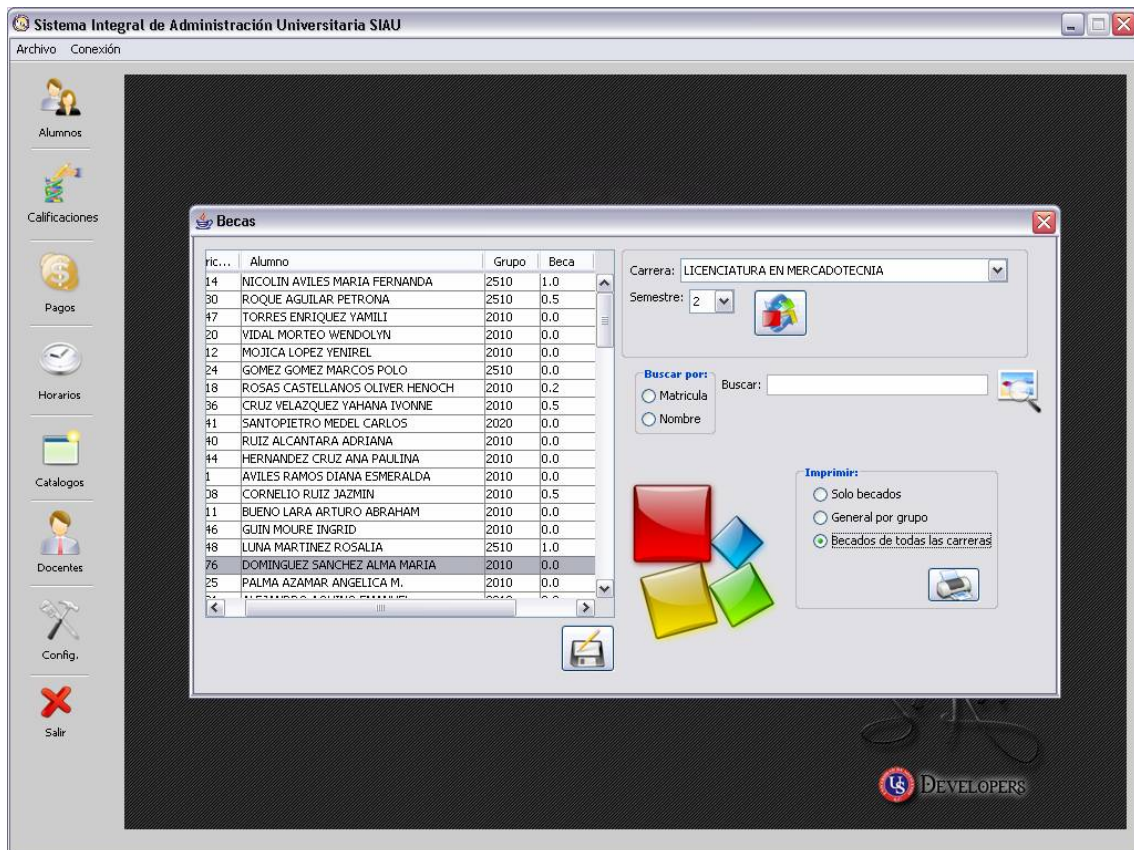
Cabe mencionar que aunque el alumno este registrado en un periodo o semestre diferente al que se este en el momento de solicitar el reporte, este muestra los saldos que presenten los alumnos a lo largo de todos los semestres que tienen cursados.



Corte de caja.

Al finalizar el día (o en cualquier momento) se realiza un corte de caja con la finalidad de ver el monto total que ha ingresado a la Universidad, y verificar que este corresponda al efectivo que se tenga en caja.

Este reporte se realiza por fecha en que haya sido ingresado al sistema, por lo que simplemente se debe proporcionar un rango de fechas para que el sistema despliegue todos los conceptos que fueron cobrados en ese periodo, los datos que este reporte muestra son: Folio, Nombre del alumno, fecha en que realizo el pago, concepto del pago, matricula y el importe tal del recibo.

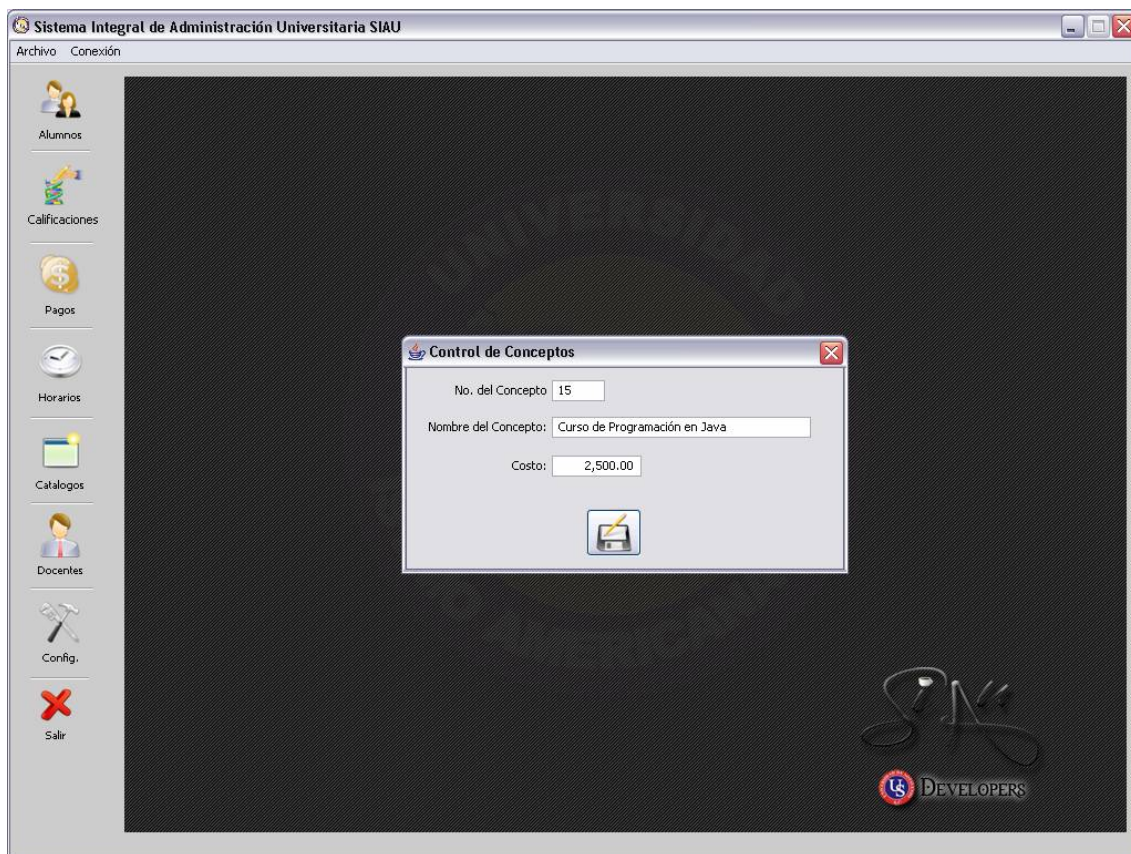


Becas.

Cada inicio de semestre los alumnos deben realizar su tramite de renovación de beca (si cuentan con ella o solicitud en caso contrario). Después de estudiar las solicitudes la Universidad asigna un porcentaje que aplica únicamente en el cobro de las seis colegiaturas correspondientes al semestre.

En este apartado después de seleccionar la carrera y semestre al que los alumnos pertenecen se muestra la lista de todos los alumnos inscritos para posteriormente asignar el porcentaje que tengan asignado .

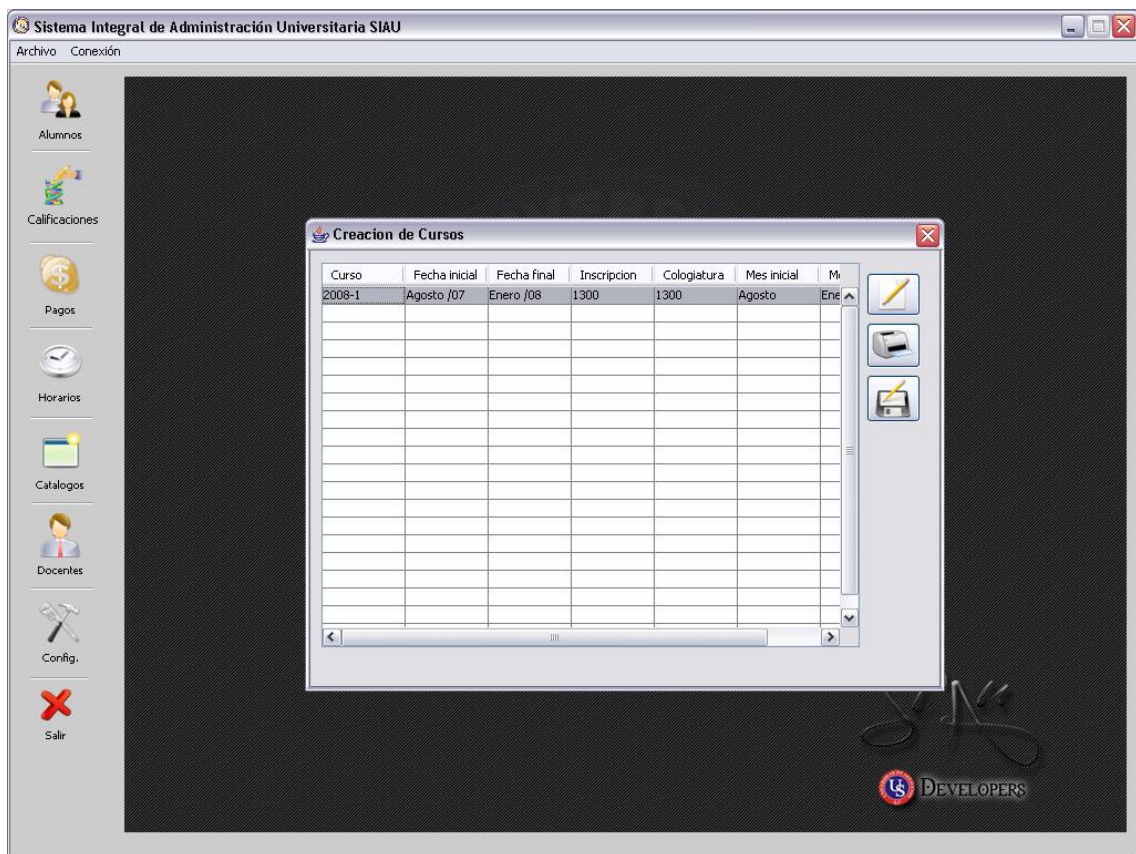
Con este procedimiento la base de datos de alumnos cambiara el porcentaje de beca para que el usuario (especialmente el encargado de caja) tenga esa información visible y pueda realizar los ajustes respecto a los cargos que el alumno tenga generado.



Control de conceptos.

Para dar de alta un nuevo concepto de servicios, como un curso, congreso, o cualquier otro que se presente en esta ventana se debe registrar un número de concepto (el cual es consecutivo y el SIAU muestra como sugerencia el número siguiente al último registro capturado), nombre del concepto que se desea agregar y el costo que tendrá dicho servicio.

Una vez realizado este procedimiento, el concepto que se acaba de agregar a la tabla, se desplegará en la pantalla de otros pagos desde donde el alumno podrá realizar el pago correspondiente.



Creación de cursos.

Desde esta opción del SIAU, el usuario puede crear el nuevo curso (semestre) que este por iniciar, deberá llenar los campos correspondientes como el ciclo, fecha en que este inicia y termina, el costo de las colegiaturas y de la inscripción y el mes en que inicia y termina el semestre (para efectos de la constancia).

Una vez llenado todos estos datos, se procede a guardarlo (y con ello crearlo) generando los siete cargos correspondientes a todos los alumnos que estén vigentes en la Base de datos (un cargo de inscripción y seis mas de colegiaturas).

Es importante que los montos de colegiatura e inscripción esten bien definidos en el momento de crear el nuevo curso, ya que estos valores son los que se tomaran a la hora de generar los cargos.

4.3 Administración del Sistema.

El Sistema Integral de Administración Universitaria (SIAU), se encuentra a disponibilidad de la Dirección Administrativa de la Universidad Istmo Americana, A.C. ya que es donde se lleva a cabo el proceso de cobro de colegiatura de los alumnos, desde el momento de inscripción hasta el pago de los derechos por titulación.

El acceso a éste está controlado por el Director General de la Universidad, es el quien tiene la responsabilidad de otorgar los permisos necesarios para el acceso a los usuarios con la finalidad de darle el uso correcto al sistema, a manera de evitar problemas de diversos tipos con la información que el propio departamento maneja.

Los que realmente interactúan con el sistema, son los auxiliares del departamento del de cobranza, ya que ellos son los encargados de llevar a cabo el correcto registro respecto a los pagos de los alumnos.

Por ello aunque la responsabilidad respecto a la administración del SIAU recaerá sobre el director General de la institución, depende también en gran parte de los usuarios, ya que ellos son la fuente principal de la información que el mismo maneja.

Por lo tanto, para la administración de este programa, se propone continuar con la misma política de utilización.

CONCLUSIONES

CONCLUSIÓN

CONCLUSIONES

La Universidad Istmo Americana se encarga de prestar servicios de educación y cuenta con un número importante de alumnos matriculados, con los cuales tiene una gran responsabilidad.

Con la finalidad de mejorar la calidad en su servicio, surge la necesidad de apoyarse en las tecnologías de información implementando sistemas que faciliten sus operaciones, llevando registros de las operaciones administrativas que realizan los alumnos.

El hecho de utilizar estos sistemas trae muchos beneficios, ya que al tener la información disponible, esta puede emplearla de maneras que ayuden a realizar su trabajo de una manera más eficiente, ya que cuenta con la posibilidad de presentarla de diversas formas lo cual facilitara su interpretación.

Es por esto que surge la necesidad de proponer la mejora de estos sistemas con la finalidad de seguir aprovechando los recursos ya que con la propuesta de mejora que el presente hace se lograra garantizar el mejoramiento de las operaciones administrativas dentro de la Universidad lo cual lleva implícito el mejoramiento en la ofera de los servicios que esta ofrece.

BIBLIOGRAFIA

<http://java.sun.com>

<http://www.mysql.com>

Manual del ESCOLA

Manual de procedimiento de inscripción y reinscripción UIA

Manual de procedimiento de cobro de colegiaturas

JAVA como programar

Deitel Harvey M.

Prentice Hall 2004

Análisis y Diseño de Sistemas de información

Senn James A.

McGraw Hill 1992

Java

Deitel Harvey M.

Prentice Hall 1998