



UNIVERSIDAD AMERICANA DE ACAPULCO

“EXCELENCIA PARA EL DESARROLLO”

FACULTAD DE INGENIERÍA EN COMPUTACIÓN

INCORPORADA A LA UNIVERSIDAD NACIONAL

AUTÓNOMA DE MÉXICO

CLAVE DE INCORPORACIÓN 8852-16

**“SISTEMA GENERADOR DE HORARIOS PARA NIVEL
LICENCIATURA, CASO DE ESTUDIO: INGENIERÍA EN
COMPUTACIÓN, DE LA UNIVERSIDAD AMERICANA
DE ACAPULCO, A.C.”**

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTA

MIGUEL ANGEL SILVA SIERRA

DIRECTOR DE TESIS

MC. JOSÉ MARIO MARTÍNEZ CASTRO



ACAPULCO, GUERRERO, ABRIL DEL 2016.



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.

Agradecimientos:

Mi agradecimiento a la Universidad Americana de Acapulco, A.C., institución que me brindó la oportunidad de realizar y culminar mis estudios de Ingeniería en Computación.

Al personal administrativo de la carrera durante el ciclo de mi generación 1997-2012, ya que me brindaron todo su apoyo.

A mis profesores, quienes me formaron como profesional, compartiéndome sus conocimientos.

A quienes guiaron mi tesis, que dedicaron su tiempo y atención a leer, comentar y ayudarme corregir éste trabajo, Ing. Eloísa Mercedes Vivas Villasana, quien me facilitó toda la información para poder realizar este proyecto; Maribel Sánchez Gregorio, sin tu gran apoyo y compromiso conmigo hubiese sido más difícil lograr ésta meta; ISC. Carlos Ernesto Islas García, gracias por tu paciencia y apoyo, sin importar la distancia y hora; y a mi Director de Tesis, M.C. José Mario Martínez Castro, gracias por su gran apoyo.

Dedicatoria:

A Dios, Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mis padres, con la mayor gratitud por los esfuerzos realizados para que yo lograra terminar mi carrera profesional, siendo para mí la mejor herencia, por su amor, esfuerzo, dedicación y entera confianza.

Para ustedes que me dieron la gracia de la vida; su apoyo y comprensión lograron hacerme llegar al final de este camino.

Papá, desde pequeño has sido para mí un hombre grande y con mucha fortaleza, siempre te he admirado, gracias por todo tu apoyo a lo largo de mi vida y sobre todo en mi etapa de estudiante, por la orientación que me has dado y por darme la pauta para realizar mis estudios y mi vida personal. Agradezco los consejos sabios y esas pláticas largas que hemos tenido, ya que han sido en el momento exacto y preciso para no dejarme caer y enfrentar los momentos difíciles, para ayudarme a tomar decisiones que me ayudan a balancear mi vida.

Mamá, gracias por todo el apoyo, cariño, comprensión y amor incondicional que desde pequeño me diste, por guiar mi camino y estar siempre junto a mí en los momentos difíciles, tú fuiste la persona que siempre me levanto los ánimos tanto en mi vida estudiantil como en mi vida personal. Mami, en ti conocí a un ser maravilloso, gracias por la paciencia y las palabras sabias que siempre tuviste para mis enojos, tristezas y momentos felices; tu forma alegre de ser arreglaba siempre todo, gracias por haber sido mi amiga y aconsejarme para que lograra mis metas y sueños, donde quiera que te encuentres: Muchas Gracias.
(QEPD)

Agradezco a mis amigos y excompañeros de generación; que juntos escribimos grandes historias y anécdotas que siempre recordaremos.

ÍNDICE GENERAL

	PÁGINA
INTRODUCCIÓN.....	6
PLANTEAMIENTO DEL PROBLEMA.....	9
JUSTIFICACIÓN.....	10
HIPÓTESIS.....	10
OBJETIVO GENERAL.....	11
OBJETIVOS ESPECÍFICOS.....	11
CAPÍTULO 1. ESTADO DEL ARTE.....	14
1. UNIVERSIDAD AMERICANA DE ACAPULCO, A.C.....	14
1.1 ANTECEDENTES.....	15
1.1.2 JOSÉ FRANCISCO RUIZ MASSIEU.....	15
1.1.3 CORRELACIÓN DE LA UNIVERSIDAD CON OTRAS INSTITUCIONES.....	16
1.1.4 UBICACIÓN Y ARQUITECTURA.....	17
1.1.5 FILOSOFÍA INSTITUCIONAL.....	18
1.1.5.1 MISIÓN.....	18
1.1.5.2 VISIÓN.....	18
1.1.5.3 PRINCIPIOS INSTITUCIONALES.....	19
1.1.5.4 PROPÓSITOS.....	19
1.1.6 OFERTA ACADÉMICA.....	20
1.2 INGENIERÍA EN COMPUTACIÓN.....	20
1.2.1 PLAN DE ESTUDIOS 2009.....	22
1.2.1.1 OBJETIVO DEL PLAN DE ESTUDIOS 2009.....	22
1.2.1.2 PERFIL DEL EGRESADO.....	22
1.2.1.3 MAPA CURRICULAR.....	23
1.3 ANTECEDENTES DE SISTEMAS GENERADORES DE HORARIOS PARA INSTITUCIONES EDUCATIVAS.....	25
1.3.1 UNTIS.....	25
CAPÍTULO 2. CASO DE ESTUDIO.....	30
2.1 PROCESO ACTUAL PARA ASIGNACIÓN DE HORARIOS.....	31
2.2 FORMATO ACTUAL DE HORARIOS.....	33
2.3 GENERACIÓN DE HORARIOS DE LA CARRERA DE INGENIERÍA EN COMPUTACIÓN.....	36

2.3.1 REGISTRO DE PROFESORES Y GRUPOS ANTE UNAM.....	41
2.3.1.1 PROFESORES DE NUEVO INGRESO Y REINGRESO.....	42
2.4 ANÁLISIS DE FORMATO DE RELACIÓN DE CÁTEDRAS AUTORIZADAS POR LA UNAM.	44
2.4.1 MANUAL DE DISPOSICIONES Y PROCEDIMIENTOS PARA EL SISTEMA INCORPORADO DE LA UNAM. CAPITULO VII. DEL PERSONAL DOCENTE.....	45
CAPÍTULO 3. CONCEPTOS Y DEFINICIONES BÁSICAS.	51
3.1 BASES DE DATOS.....	52
3.1.1 ¿QUÉ SON LAS BASES DE DATOS?.....	52
3.1.2 ESTRUCTURA DE ALMACENAMIENTO DE DATOS.....	56
3.1.2.1 ÍNDICES.	58
3.1.2.2 VISTAS.	59
3.1.2.3 ENTIDAD – RELACIÓN.....	59
3.1.2.4 RELACIONES ENTRE TABLAS.....	60
3.2 INTRODUCCIÓN A ACCESS.	62
3.2.1 CONOCIENDO EL PROGRAMA.....	62
3.2.2. LA INTERFAZ.	64
3.3 INTRODUCCIÓN A VISUAL BASIC.	65
3.3.1. RESEÑA DEL LENGUAJE.	67
3.3.2 BASES DE DATOS Y VISUAL BASIC.	71
3.3.3 HERRAMIENTAS COMPLEMENTARIAS.	71
3.4 LENGUAJES DE PROGRAMACIÓN.....	72
3.4.1 DIFERENCIAS ENTRE LOS LENGUAJES DE PROGRAMACIÓN DE ALTO Y BAJO NIVEL.	73
3.4.2 EL CÓDIGO MÁQUINA.	75
3.4.3 EL LENGUAJE ENSAMBLADOR.....	76
3.4.4 LOS LENGUAJES DE ALTO NIVEL.....	76
3.4.4.1 COMPARACIONES.....	77
3.6 JUSTIFICACIÓN DEL LENGUAJE UTILIZADO EN EL PROGRAMA DE GENERACIÓN DE HORARIOS.....	78
MS ACCESS.....	79
ORACLE.	79
SQL SERVER.....	80
MICROSOFT VISUAL BASIC.....	82
JAVA Y C#.	83

PHP Y JAVASCRIPT.....	84
3.7 REQUERIMIENTOS MÍNIMOS DE HARDWARE, NECESARIOS PARA LA INSTALACIÓN Y UTILIZACIÓN DEL SISTEMA.....	85
CAPÍTULO 4. DESARROLLO E IMPLEMENTACIÓN DEL SISTEMA.....	87
4.1. ANÁLISIS.....	91
4.1.1. RECOPIACIÓN DE LA INFORMACIÓN.....	91
4.1.2. ANÁLISIS DEL PROCESO PARA LA ELABORACIÓN DE LOS HORARIOS.....	91
4.1.3. ANÁLISIS DE LA INFORMACIÓN.....	109
4.2. DISEÑO.....	115
4.2.1. MAPA CONCEPTUAL DEL PROYECTO.....	115
4.2.2. DIAGRAMA DE CONTEXTO DEL PROYECTO.....	120
4.2.3. MODELO DE DATOS.....	122
4.2.4. DICCIONARIO DE DATOS.....	123
4.2.5. SOLUCIÓN DEFINIDA.....	128
4.3. DESARROLLO.....	130
4.3.1. MÓDULOS DE EVENTOS.....	130
4.3.2. ARQUITECTURA DEL SISTEMA.....	134
4.4. IMPLEMENTACIÓN.....	139
4.4.1. INSTALACIÓN.....	139
4.4.2 USO DEL SISTEMA Y PRUEBAS.....	143
CAPÍTULO 5. RESULTADOS Y TRABAJO A FUTURO.....	166
5.1. PRUEBAS Y RESULTADOS.....	166
5.1.1 PRUEBAS DEL SISTEMA.....	167
5.1.1.1 PRUEBA 1. ÁREAS DE MENÚS.....	167
5.1.1.2 PRUEBA 2. VALIDACIÓN DE ENTRADAS.....	168
5.1.1.3 PRUEBA 3: DESPLEGADO DE RESULTADOS.....	171
5.1.1.4 PRUEBA 4: GENERAR HORARIOS.....	175
5.2. TRABAJO A FUTURO.....	177
CONCLUSIONES.....	177
BIBLIOGRAFÍA.....	181

ÍNDICE DE FIGURAS

PÁGINA

FIGURA 1. LIC. JOSÉ FRANCISCO RUIZ MASSIEU.	15
FIGURA 2. FACHADA DE LA UNIVERSIDAD AMERICANA DE ACAPULCO A.C.	17
FIGURA 3. LOGO Y LEMA DE LA UNIVERSIDAD AMERICANA DE ACAPULCO A.C.	20
FIGURA 4. LOGOTIPO DE FACULTAD DE INGENIERÍA EN COMPUTACIÓN DE LA UNIVERSIDAD AMERICANA DE ACAPULCO A.C.	21
FIGURA 5. MAPA CURRICULAR 2009.	24
FIGURA 6. ICONO DE UNTIS.	26
FIGURA 7. DIAGRAMA ACTUAL DE PROCESOS PARA GENERAR HORARIOS.	32
FIGURA 8. EJEMPLO DE FORMATO DE HORARIO ESCOLAR.	33
FIGURA 9. LOGOS INSTITUCIONALES EN EL FORMATO DE HORARIO ACTUAL.	34
FIGURA 10. NOMBRES DE IDENTIFICACIÓN DEL FORMATO DE HORARIO.	34
FIGURA 11. DIAGRAMA PARA DEFINIR CICLO ESCOLAR.	36
FIGURA 12. FORMATO DE DISPONIBILIDAD.	38
FIGURA 13. DIAGRAMA DE FLUJO PARA ASIGNACIÓN DE HORARIOS.	40
FIGURA 14. FORMATO DE RELACIÓN DE CÁTEDRAS AUTORIZADAS DE LA UNAM DE UN PROFESOR ACTUAL.	43
FIGURA 15. ICONO DE INICIO DE ACCESS.	63
FIGURA 16. PANTALLA DE BIENVENIDA.	64
FIGURA 17. DISEÑO DE ENTORNO DINÁMICO DE ACCESS 2007.	65
FIGURA 17. PANEL DE EXPLORACIÓN DE VISUAL BASIC.	65
FIGURA 18. ESQUEMA DE LA METODOLOGÍA RUP (RATIONAL UNIFIED PROCESS), BASADA EN SU CICLO DE VIDA.	90
FIGURA 19. DIAGRAMA DE FLUJO DEL ANÁLISIS DEL PROCESO PARA LA ELABORACIÓN DE LOS HORARIOS.	92
FIGURA 20. DIAGRAMA DE CASOS DE USO.	97
FIGURA 21. DIAGRAMA DE SECUENCIA.	99
FIGURA 22. DIAGRAMA DE COLABORACIÓN.	101
FIGURA 23. DIAGRAMA DE TRANSICIÓN DE ESTADOS DE LA DISPONIBILIDAD HORARIA DEL PROFESOR.	104
FIGURA 24. DIAGRAMA DE TRANSICIÓN DE ESTADOS DEL HORARIO.	105
FIGURA 25. DIAGRAMA DE ACTIVIDAD.	108
FIGURA 26. MAPA CURRICULAR DE LA CARRERA INGENIERÍA EN COMPUTACIÓN, PLAN 2009 UNAM.	113

FIGURA 27. MÓDULOS DE SALIDA DE LA CARRERA INGENIERÍA EN COMPUTACIÓN, PLAN2009 UNAM.	114
FIGURA 28. MAPA CONCEPTUAL DEL PROYECTO.	119
FIGURA 29. DIAGRAMA DE CONTEXTO DEL PROYECTO.	121
FIGURA 30. MODELO ENTIDAD RELACIÓN DEL SISTEMA GENERADOR DE HORARIOS.	122
FIGURA 31. DIAGRAMA DE ARQUITECTURA DE MENÚS DEL SISTEMA GENERADOR DE HORARIOS.	138
FIGURA 32. ICONO DE ACCESO DIRECTO PARA INGRESAR AL SISTEMA GENERADOR DE HORARIOS.	142
FIGURA. 33. PANTALLA DE ACCESO AL SISTEMA.	143
FIGURA 34. PANTALLA PRINCIPAL DEL SISTEMA GENERADOR DE HORARIOS.	145
FIGURA 35. BARRA DE MENÚS.	168
FIGURA 36. MENÚS CON SUBMENÚS Y VENTANA DE ACCIÓN.	168
FIGURA 37. VENTANA DE CAPTURA DE PROFESOR CON AVISO DE ERROR POR FALTA DE INFORMACIÓN.	169
FIGURA 38. VENTANA DE CAPTURA DE GRUPO CON DATOS FALTANTES Y AVISO DE ERROR POR PARTE DEL SISTEMA.	170
FIGURA 39 VENTANA DE LA ASIGNACIÓN DE CÁTEDRAS AUTORIZADAS POR LA UNAM A UN PROFESOR.	172
FIGURA 40. SELECCIÓN DE LA PREFERENCIA DEL PROFESOR BASÁNDONOS EN EL LISTADO DE CÁTEDRAS AUTORIZADAS PARA ESE PROFESOR ESPECIFICO.	173
FIGURA 41. FORMATO DE CARGA HORARIA QUE GENERA EL SISTEMA.	174
FIGURA 42. VENTANA CON LA VISTA PRELIMINAR DEL HORARIO DE CLASES.	176

ÍNDICE DE TABLAS

	PÁGINA
TABLA 1. TIPOS DE BASES DE DATOS.....	54
TABLA 2. MODELOS DE BASES DE DATOS, MÁS USADOS EN LAS ÚLTIMAS DÉCADAS.....	56
TABLA 3. TIPOS DE DATOS QUE PUEDE ALMACENAR UNA TABLA.	57
TABLA 4. UN DETALLE DE LOS TIPOS DE CARDINALIDAD EXISTENTE EN UNA BASE DE DATOS.	61
TABLA 5. COMPARATIVA DE DBMS (SISTEMAS ADMINISTRADORES DE BASES DE BOTOS, POR SUS SIGLAS EN INGLÉS DATABASE MANAGEMENT SYSTEM).	78
TABLA 6. COMPARATIVA DE LENGUAJES DE PROGRAMACIÓN MÁS POPULARES.....	81

Introducción.

Los horarios de clases son un elemento fundamental en la organización de cualquier institución académica, dado que constituyen el elemento organizativo de las actividades escolares entre docentes y alumnos y permiten una visión general de la distribución de los recursos de la institución.

El proceso de elaboración de horarios es considerado una tarea muy complicada, debido a la gran cantidad de información y condiciones que deben tenerse en cuenta, en cada inicio de un período escolar se presenta el problema de ubicar a maestros y alumnos de la manera más eficiente, en los espacios adecuados, siguiendo las directrices que los planes de estudios marcan.

Actualmente en la carrera de Ingeniería en Computo de la Universidad Americana de Acapulco, A.C., la gestión y confección de los horarios de clases se realiza de forma manual, debiendo cumplirse una gran cantidad de restricciones, por lo que la confección de la carga horaria es una tarea sumamente difícil que consume mucho tiempo y generalmente requiere de varias revisiones para obtener un horario de clases congruente y lo más óptimo posible.

El presente trabajo de tesis, consiste en la creación de un sistema automatizado para la generación de los horarios de clase en la carrera de Ingeniería en Computo de la Universidad Americana de

Acapulco, A.C., en respuesta a la necesidad de mejorar y perfeccionar la manera en que actualmente se lleva a cabo esta tarea, la utilización de este sistema significaría una mejora considerable en este proceso en cuanto a eficiencia, tiempo y organización, contribuyendo favorablemente al desarrollo del proceso docente y educativo de la universidad.

El contenido de este trabajo de Tesis, comprende un total de seis capítulos referentes a diversas áreas temáticas, y un apartado de anexos, estructurados de la siguiente manera:

- Capítulo 1. Marco Teórico, en éste apartado se abordan la historia y antecedentes de la Universidad Americana de Acapulco como institución educativa, la carrera de Ingeniería en Computación dentro de la Universidad Americana de Acapulco incorporada a la UNAM y el antecedente de los sistemas de Información como apoyo en la organización escolar.
- Capítulo 2. Caso de Estudio, en éste segmento se analizarán algunas de las metodologías que sirven como base para el desarrollo de los sistemas de Bases de Datos, a fin de especificar un modelo de trabajo para el Sistema que se propone utilizar en la asignación de Horarios y Grupos de la carrera de Ingeniería en Computación de la Universidad Americana de Acapulco.

- Capítulo 3. Conceptos y Definiciones Básicas, en éste capítulo se definirán los conceptos básicos de las herramientas y elementos que conforman nuestro sistema de Base de Datos y que nos sirva como sustento a lo largo de nuestra investigación.
- Capítulo 4. Desarrollo e Implementación del sistema, en ésta parte de la tesis se ejecutarán las fases de nuestro modelo de trabajo, aplicándolas al caso práctico de asignación de Horarios y Grupos de la carrera en Ingeniería en Computación de la Universidad Americana de Acapulco.
- Capítulo 5. Resultados y trabajo a futuro, en ésta sección se cerrará el proceso de desarrollo del sistema de base de datos para la asignación de horarios y grupos con las propuestas para el mantenimiento y subsecuentes actualizaciones que se requieran.
- Capítulo 6. Conclusiones, en este apartado se expondrán las impresiones derivadas del trabajo realizado durante este proyecto de tesis.

Planteamiento del Problema.

En cualquier institución educativa se manejan diversos grados y grupos de alumnos, los cuales requieren de cierta organización para ubicar a maestros y alumnos de la manera más eficiente; la generación de horarios siempre muestra dificultades con la organización de diversos elementos, tales como: profesores, alumnos, asignaturas, salones, etc., a este problema se le conoce como timetabling y su complejidad está directamente relacionada con las restricciones asociadas a estos recursos.

Actualmente el personal administrativo de la carrera de Ingeniería en Computación de la Universidad Americana de Acapulco, A.C. realiza la asignación de horarios de forma manual, causando un desperdicio de tiempo, salario y esfuerzo; lo que provoca atrasos en los procedimientos administrativos con la Dirección de Servicios Escolares, así como con los mismos profesores y alumnos.

Justificación.

En todas las Instituciones académicas existe la necesidad de asignar y coordinar los recursos económicos, materiales y humanos, en beneficio de los estudiantes, es por esto que en esta investigación de tesis se realizará un software capaz de encontrar la asignación óptima de clases, profesores, grupos y horarios, utilizando nuevas tecnologías.

Con el diseño de este sistema se agilizará el proceso de asignación de horario cada vez que lo llegue a requerir la carrera de Ingeniería en Computación y se realizaran los trámites más eficazmente con profesores y alumnos y permitirá realizarlo en el menor tiempo posible, eliminando en su totalidad los posibles errores en empalmes de horas, y aumentando la productividad del personal administrativo.

Hipótesis.

Se realizará una asignación óptima de clases, profesores, grupos y horarios que permitirá agilizar los trámites requeridos por profesores, alumnos y personal administrativo, a través de un sistema generador de horarios para la carrera de Ingeniería en Computación de la Universidad Americana de Acapulco A.C., que permitirá reducir el tiempo y la ocurrencia de errores en la actual asignación manual de los horarios de clases.

Objetivo General.

Se desarrollará un sistema generador de horarios, que permitirá realizar la gestión de estos en el menor tiempo posible y con la mayor exactitud, para la carrera de Ingeniería en Computación en la Universidad Americana de Acapulco, A.C.

Objetivos Específicos.

Lo que se pretende alcanzar con esta propuesta de solución es desarrollar un sistema que realice lo siguiente:

- Se analizará y valorará el desempeño de la asignación de horarios actual.
- Se realizarán las bases de datos de los maestros y materias.
- Se analizará y diseñará un algoritmo para resolver el problema de asignación de horarios.
- Se propondrá un modelo prototipo para la asignación de horarios.
- Se establecerá una metodología para la realización del software.
- Se analizarán los diversos programas de software para su futura implementación.
- Se realizarán la instalación del software para ponerlo en práctica.
- Se analizará su funcionamiento para corregir errores si existieran.

1

Capítulo

Estado del Arte

Capítulo 1. Estado del Arte.

1. Universidad Americana de Acapulco, A.C¹.

La Universidad Americana de Acapulco se funda en 1991, siendo la primera opción de educación superior privada en el Estado de Guerrero; surgió como una propuesta de José Francisco Ruiz Massieu, en ese entonces Gobernador del Estado de Guerrero, quien junto a distinguidas personalidades buscaban cimentar un centro de excelencia académica para los jóvenes del estado.

Desde sus inicios La Universidad Americana de Acapulco asumió el compromiso de adoptar los principios que rigen a una comunidad académica digna y honorable, como son: las libertades de cátedra, enseñanza e investigación, la calidad académica, el respeto y la tolerancia.

El lema “Excelencia para el Desarrollo” ha guiado su actuación desde entonces.

¹ <http://www.uaa.edu.mx>

1.1 Antecedentes.

En octubre de 1991, se inauguró la Facultad Libre de Derecho que en marzo de 1992 se transformaría en la Universidad Americana de Acapulco consolidándose como una institución de enseñanza superior privada el día 31 de agosto de 1992 con una ceremonia inaugural. El primer Rector Alfredo Álvarez recibió como invitados a esa ceremonia al Gobernador del Estados José Francisco Ruiz Massieu, al Rector de la Universidad Nacional Autónoma de México, al Abogado General de la Universidad Tatoro Gallardo, al Secretario de Educación Pública Ernesto Zedillo Ponce de León, al Secretario de Turismo, Pedro Joaquín Coldwell, al Embajador de México en la OEA, Alejandro Carrillo Castro, entre otras distinguidas personalidades.

1.1.2 José Francisco Ruiz Massieu.

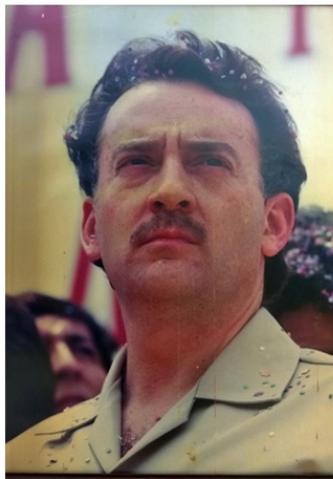


Figura 1. Lic. José Francisco Ruiz Massieu.

José Francisco Ruiz Massieu, (figura 1) fundó la Universidad Americana de Acapulco, fue un destacado hijo de Guerrero: brillante como abogado, catedrático, historiador, investigador jurídico, autor de más de un centenar de libros, artículos, prólogos y presentaciones, promotor y defensor ferviente de la cultura en todas sus facetas, fue un hombre de ideas progresistas reflejadas en los planes y programas que realizó durante su gestión como Gobernador del estado de Guerrero, durante el periodo entre 1987 y 1993.

1.1.3 Correlación de la Universidad con otras Instituciones.

La Universidad Americana de Acapulco ha mantenido desde sus inicios una estrecha relación con la Universidad Nacional Autónoma de México (UNAM) desde la incorporación oficial de estudios de los programas académicos, como con el apoyo y participación de maestros y directivos, hasta la celebración de numerosos convenios generales y específicos de colaboración académica que le han permitido estar a la vanguardia del conocimiento.

1.1.4 Ubicación y Arquitectura.



Figura 2. Fachada de la Universidad americana de Acapulco A.C.

El campus central de la Universidad Americana de Acapulco se encuentra en la Avenida Costera Miguel Alemán Valdés, avenida principal del puerto, El edificio central es una construcción de cinco niveles, donde se encuentran las principales oficinas de la Universidad como la rectoría, la vicerrectoría y la oficina del Abogado General y las áreas administrativas no obstante el mayor espacio lo ocupan salones, auditorios y la Biblioteca Central.

En la entrada principal existen dos fuentes monumentales, la del lado izquierdo, cuyo fondo es de color azul verdoso, un homenaje a los ríos y lagunas de Guerrero, y la del lado derecho, de tono azul turquesa como homenaje al océano pacífico que baña las costas del Estado de Guerrero, tal como se aprecia en la figura 2.

1.1.5 Filosofía Institucional.

1.1.5.1 Misión.

La Universidad Americana de Acapulco tiene como misión formar ciudadanos comprometidos con la sociedad mexicana. Su objeto es preparar a sus estudiantes para la vida profesional con responsabilidad social. Sus ejes son la enseñanza media superior, superior y de posgrado, la apreciación del conocimiento científico, la difusión de los valores culturales y humanísticos, el sentido de la solidaridad nacional, la conciencia de comunidad universitaria y la responsabilidad social.

1.1.5.2 Visión.

Formar profesionales, maestros e investigadores comprometidos con el desarrollo económico, político, social y cultural de Guerrero, conscientes de su responsabilidad como mexicanos integrantes de una sociedad plural, democrática y abierta al pensamiento universal.

1.1.5.3 Principios Institucionales.

- Libertad académica
- Apertura al diálogo
- Libertad de cátedra
- Humanismo y sus valores fundamentales
- Formación de profesionistas, maestros e investigadores
- Formación integral
- Ética profesional
- Conciencia de la problemática social contemporánea
- Búsqueda de la verdad científica como principio y fin del trabajo académico
- Excelencia en el ejercicio profesional
- Integración comunitaria

1.1.5.4 Propósitos.

La Universidad Americana de Acapulco, para cumplir con su misión institucional, establece los propósitos siguientes:

- La Docencia
- La Investigación
- La Difusión y Extensión de la Cultura, y
- Sentido de Comunidad y Distribución Social del Conocimiento.

1.1.6 Oferta Académica.

La UAA tiene distintas opciones incorporadas a la Universidad Nacional Autónoma de México, la máxima casa de estudios del país. Conforme a la incorporación al Sistema UNAM, nuestro modelo educativo ha sido diseñado para preparar profesionales comprometidos con los más altos valores de la sociedad mexicana, conscientes de su papel como mexicanos responsables, atentos al desarrollo social, cultural y económico de México y cuyo lema compartido es “Excelencia para el Desarrollo” como lo muestra la figura 3.



Figura 3. Logo y lema de la Universidad Americana de Acapulco A.C.

1.2 Ingeniería en Computación.

En el mes de agosto de 1994, inicia formalmente actividades académicas la licenciatura de Ingeniería en Computación, con planes y programas de estudios incorporados a la Universidad Nacional Autónoma de México, con el plan de estudios 1990, en la figura 4 se muestra el logo oficial de esta Facultad.



Figura 4. Logotipo de Facultad de Ingeniería en Computación de la Universidad Americana de Acapulco A.C.

Durante los años que han transcurrido desde su inicio de actividades, la Licenciatura de Ingeniería en Computación, ha contemplado 4 planes de estudios incorporados a la UNAM:²

- Plan 1990.- conformado por 10 Semestres (5 años)
- Plan 1993.- estructurado en 10 semestres (5 años)
- Plan 2005 .- conformado por 9 semestres (4.5 años)
- Plan 2009.- estructurado en 9 semestres (4.5 años) Actual

² Dirección de Servicios Escolares, Universidad Americana de Acapulco, A.C.

1.2.1 Plan de Estudios 2009.

1.2.1.1 Objetivo del Plan de Estudios 2009.

El objetivo general de la carrera de El Ingeniero en Computación será el de formar un profesionalista de alto nivel científico y tecnológico, con conocimientos sólidos y generales que le permitan ser capaz de identificar, analizar, planear, diseñar, organizar, producir, operar y dar soporte a los sistemas electrónicos (Ingeniería de Hardware) para el procesamiento digital de datos y control de procesos, a los sistemas de programación tanto de base como de aplicación (Ingeniería de Software); al desarrollo e investigación en las ciencias de la computación ; a los sistemas de comunicación y seguridad (Redes de datos), a los sistemas de bases de datos, a los sistemas inteligentes y sistemas de cómputo gráfico; que le permitan responder a las necesidades que se presentan en el campo de trabajo de la ingeniería en computación.

1.2.1.2 Perfil del Egresado.

El egresado de la Facultad de Ingeniería contará con conocimientos sólidos en Matemáticas y Física, y con conocimientos generales de Química así como de las áreas de sistemas de programación (software), sistemas electrónicos digitales (hardware), ciencias de la computación, control y comunicaciones, que le permiten

responder a las diversas necesidades que se presentan en el campo de trabajo de la Ingeniería en Computación.

Además sabrá diseñar e instalar redes de teleinformática; planear, diseñar construir sistemas de interface máquina-máquina y hombre-máquina, así como sistemas automáticos de control digital para la industria; desarrollar nuevos lenguajes de computadora; resolver problemas con orientación teórica, tales como: diseños autómatas, modelado de estructuras de datos, desarrollo de sistemas operativos, desarrollo de manejadores de sistemas de bases de datos, compiladores, etc.

De igual forma, estará capacitado para trabajar conjuntamente con otros especialistas en la solución de problemas en otros campos de acción; sin olvidar que deberá estar a la altura del avance tecnológico, a fin de permanecer actualizado en el estado del arte de la computación y dominar, por lo menos una lengua extranjera

1.2.1.3 Mapa Curricular.

Para hacer una correcta asignación de horarios se debe tomar en cuenta que además de las materias curriculares incorporadas al plan de estudios 2009 UNAM, es necesario integrar los talleres que imparten de forma adicional la carrera de Ingeniería en Cómputo de la Universidad Americana de Acapulco, AC., como se muestra en la siguiente figura las materias hasta el noveno semestre, si desea

adentrar más en ésta información puede revisar el apéndice A o en la siguiente dirección www.dgire.unam.mx/normatividad/planes_de_estudio/ingenieria_en_computacion/año_plan_2009

**Planes de Estudio
Ingeniería en Computación**

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
SECRETARÍA GENERAL
DIRECCIÓN GENERAL DE INCORPORACIÓN Y REVALIDACIÓN DE ESTUDIOS**

PLAN: 16		Área: 00	
AÑO PLAN: 2009 INGENIERÍA EN COMPUTACIÓN		GRADO QUE OBTIENE: LICENCIATURA	
FACULTAD O ESCUELA : INGENIERÍA		SERIACION: SÍ	
CRÉDITOS OBLIGATORIOS: 360		CRÉDITOS OPTATIVOS: 48	
TOTAL DE CRÉDITOS: 408		NÚMERO DE SEMESTRES: 09	
ASIG.	CRÉDITOS	NOMBRE DE LAS ASIGNATURAS	TIPO DE MATERIA SERIACIÓN
PRIMER SEMESTRE			
1100	9	Álgebra	Obligatoria General
1102	9	Geometría Analítica	Obligatoria General
1107	6	Cultura y Comunicación	Obligatoria General
1108	9	Cálculo Diferencial	Obligatoria General
1109	10	Química y Estructura de Materiales	Obligatoria General
SEGUNDO SEMESTRE			
0062	9	Álgebra Lineal	Obligatoria General Asig. 1100
0065	9	Estática	Obligatoria General Asig. 1102
1112	8	Computación para Ingenieros	Obligatoria General
1207	9	Cálculo Integral	Obligatoria General Asig. 1108
1211	9	Introducción a la Economía	Obligatoria General
TERCER SEMESTRE			
0063	9	Cálculo Vectorial	Obligatoria General Asig. 1207
0066	9	Cinemática y Dinámica	Obligatoria General Asig. 0065
1306	9	Ecuaciones Diferenciales	Obligatoria General
1312	8	Programación Avanzada y Métodos Numéricos	Obligatoria General Asig. 1112
1314	11	Principios de Termodinámica y Electromagnetismo	Obligatoria General
CUARTO SEMESTRE			
0712	9	Probabilidad y Estadística	Obligatoria General
1418	9	Análisis de Sistemas y Señales	Obligatoria General
1420	6	Literatura Hispanoamericana Contemporánea	Obligatoria General
1422	9	Algoritmos y Estructuras de Datos	Obligatoria General
1429	9	Estructura y Programación de Computadoras	Obligatoria General
OPTATIVA (HUMANIDADES)			
QUINTO SEMESTRE			
1551	11	Diseño de Sistemas Digitales	Obligatoria General Ciclo 1
1552	9	Estructuras Discretas	Obligatoria General Ciclo 1 y Asig. 1422
1553	9	Ingeniería de Software	Obligatoria General Ciclo 1
1554	9	Sistemas Operativos	Obligatoria General Ciclo 1 y Asig. 1429
1562	8	Circuitos Eléctricos	Obligatoria General Ciclo 1 y Asig. 1418
SEXTO SEMESTRE			
1654	11	Dispositivos y Circuitos Electrónicos	Obligatoria General Ciclo 1 y 2
1670	9	Lenguajes Formales y Automatas	Obligatoria General Ciclo 1 y 2

Figura 5. Mapa curricular 2009.

1.3 Antecedentes de Sistemas generadores de horarios para Instituciones educativas.

Actualmente en México son muchas las Instituciones educativas que utilizan generadores de horarios debido a que el uso de estos sistemas permite optimizar el tiempo del personal y está relacionado directamente con la correcta organización y planificación académica de cada Institución.

En internet se pueden encontrar varias propuestas de sistemas generadores de horarios que se ofertan al público en general, especializados en instituciones educativas, de entre los que haremos una remembranza general analizando el más popular de ellos: UNTIS.

1.3.1 UNTIS.

UNTIS es una herramienta que permite administrar de forma sencilla horarios, aulas y recursos de los centros educativos, que trabaja con un algoritmo de optimización que confecciona automáticamente los horarios del centro académico e incorpora otras funcionalidades como la identificación de clases irregulares en el proceso de planificación, valores contractuales de horas en que los profesores deben impartir clases, horario óptimo de guardias, planificación de sustituciones, estadísticas, etc. El generador de horarios UNTIS está completamente integrado en la plataforma de

gestión Alexia. Ambos servicios los proporciona Cospa&Agilmic y ha sido desarrollado por la empresa UNTIS Grubers & Petters.³

El programa se inicia mediante el icono de UNTIS que normalmente se instala en el escritorio de la maquina donde se instala tal como podemos observar en la figura 6.



Figura 6. Icono de UNTIS.

Para familiarizarse con este programa se recomienda empezar con la instalación, continuar con la entrada de los datos del centro, los datos básicos y las clases, posteriormente Pasar por la optimización, que es el cálculo de los horarios, y el diagnóstico, que evalúa los resultados obtenidos. Para finalmente acabar con el diálogo interactivo, que permite modificar los horarios, y la impresión de los horarios de profesores, grupos, aulas y materias.

Los datos básicos son los profesores, grupos, aulas y materias. Las secciones, vacaciones, alumnos y espacios, que figuran en el

³ http://www.grupet.at/home_es.php

mismo menú, los complementan. Entre los datos básicos los de los grupos, profesores y aulas son los más importantes ya que definen las clases y materias. Como reciben un trato igual en su gestión, los llamaremos, de forma generalizada, “elementos” o titulares”. Todos estos datos se administran en ventanas, en las cuales se definen y gestionan, estas ventanas tienen la misma estructura y el mismo manejo, y constan de tres partes: la Barra de Herramientas, la Matriz de datos y la Caja de Diálogo Tarjeta.

Las clases son las horas lectivas de las Materias que imparten los Profesores a los alumnos de un grupo en un aula. Pueden ser de 30, 45 ó 60 minutos o de cualquier otra duración y tener otras características adicionales. Se recomienda utilizar sólo los parámetros estándares que propone el menú <Clases>, porque un uso equívoco puede invalidar los horarios obtenidos.

Hay que distinguir entre clases previstas o de finidas y clases planificadas o colocadas.

Al termino de capturar toda la información de clases / grupos, se continua con el proceso de optimización, que es la planificación y colocación automática de las horas lectivas por parte del programa. En esta tarea usa un algoritmo sofisticado. La planificación acabará con el Diagnóstico, en el cual el programa compara los resultados obtenidos con los criterios que el usuario había establecido en la ponderación pedagógica y que un buen horario debería cumplir. El algoritmo de la generación de horarios consta de dos niveles:

- 1) La colocación inicial de clases y
- 2) Los posteriores intercambios múltiples de clases.

Cuando el programa ya ha realizado los cálculos y nos brinda los horarios óptimos, se puede proceder con el ajuste manual, es decir, nos permite mover o arrastrar horas, profesores, aulas, etc., a fin de que el horario final sea totalmente el adecuado.

Una vez realizada la optimización y aprobados los horarios, llega el momento de recoger los frutos de todo el trabajo anterior y presentarlos del modo más adecuado posible, de acuerdo con varias plantillas de diseño con los que ya cuenta el programa, además de brindar la oportunidad de análisis y facilitar la generación de gráficos especializados de cada rubro que se pondera dentro del mismo programa.

2

Capítulo



Caso de Estudio

Capítulo 2. Caso de Estudio.

Los sistemas de información han brindado soporte a las instituciones de educación superior en sus procesos académico-administrativos, tales como la administración de los registros de los estudiantes, administración financiera, administración de los currículos y la programación de cursos.

El uso de estos sistemas de información permite a las instituciones adquirir eficiencia operativa y organizacional. Dentro de los procesos académico-administrativos más complejos que las universidades deben afrontar se encuentra el problema de asignación de horarios y consiste a grandes rasgos en la asignación de cada curso a ofrecer en un periodo académico (semestre, trimestre, cuatrimestre, etc.) a una franja de tiempo, un salón de clase, un profesor, sujeto a un conjunto de restricciones y requerimientos.

Las restricciones de este tipo de problemas de asignación de recursos obedecen a tamaño y tipo de salón requerido, disponibilidad del profesor, currículo de los programas, prerrequisitos de cursos, etc.

Debido a su compleja solución el problema de programación de horarios se ha convertido en un reto intelectual interesante y ha sido estudiado por muchos investigadores desde la década de los 90s y como resultado se han planteado diferentes alternativas de solución.

El presente trabajo de tesis en este capítulo 2 describe la problemática asociada para el problema de programación de horarios en el caso específico de estudio: Ingeniería en Computación, nivel licenciatura de la Universidad Americana de Acapulco, AC.

2.1 Proceso actual para asignación de horarios.

Como inicio de esta investigación se le pidió al personal encargado que nos desglosará el proceso actual que se realiza para lograr asignar los horarios en cada ciclo escolar, de lo cual definimos lo siguiente:

- 1) Se envía el formato de Disponibilidad de Materias de manera electrónica o impresa a los maestros interesados en impartir cátedra en la UAA, dependiendo del ciclo escolar que sea él que se está programando.
- 2) Este formato es llenado con base a las indicaciones contenidas en el.
- 3) Después de ser llenado es enviado electrónicamente o entregado impreso de manera personal en la Facultad.
- 4) Se inicia la descarga de materias por maestro, basándonos en la repetición, es decir, si la materia la dio el maestro con anterioridad.

- 5) Seleccionando al maestro titular de cada materia, organizamos su horario disponible, respetando Inglés y certificación.
- 6) Colocamos los horarios y se cotejan con el listado general para ver si las horas no se empatan y si el profesor tiene disponibilidad.
- 7) Ya que están organizados los horarios, se envían de manera personal a cada maestro, vía electrónicamente en el formato de Horario.

Todos estos pasos se pueden apreciar mejor en el diagrama de procesos de la figura 7.

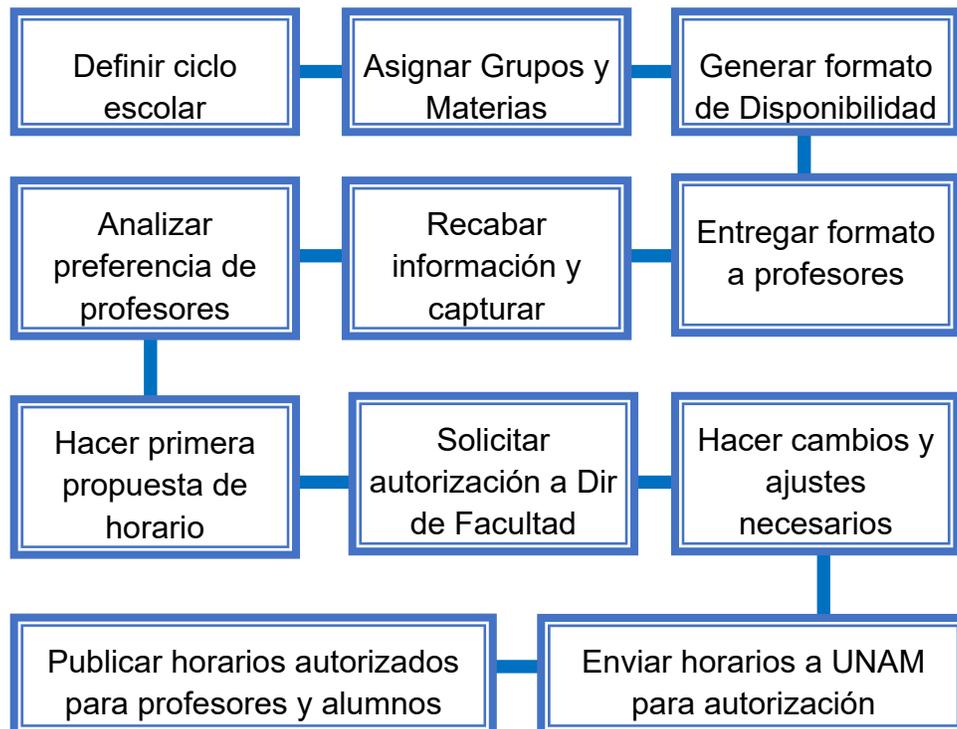


Figura 7. Diagrama actual de procesos para generar horarios.

2.2 Formato actual de horarios.

El formato que se genera para publicar a la comunidad universitaria la carga horario de cada semestre, específicamente a los involucrados en nuestro caso de estudio: el nivel licenciatura de la Ingeniería en Computación, es el que se puede apreciar en la figura siguiente:



UNIVERSIDAD AMERICANA DE ACAPULCO
FACULTAD DE ARQUITECTURA E INGENIERÍA
INGENIERÍA EN COMPUTACIÓN
HORARIO DEL SEGUNDO SEMESTRE
GRUPO 2010 IC



AULA:	A-304
-------	-------

PLAN DE ESTUDIOS 2009CICLO ESCOLAR 2014/2015-2

	ASIGNATURA	HRS. T	HRS. P	PROFESOR (ES)	HORARIO					
					LUNES	MARTES	MIÉ	JUEVES	VIÉ	SABADO
	INGLÉS	5	0	CENTRO DE LENGUAS	13 A 14	13 A 14	13 A 14	13 A 14	13 A 14	
0062	ALGEBRA LINEAL	4.5	0	ING. GIBRAN HERRERA GARCIA					11 - 13	11 - 13:30
0065	ESTÁTICA	4.5	0	ING. AGUSTIN VIVAS GARCIA		10:30 - 13		11:00 - 13		
1112	COMPUTACIÓN PARA INGENIEROS	3	2	ING. EDUARDO PERALTA MARTIÑON	7:30 - 10:00		LAB. DE REDES 7 - 9:30			
1207	CÁLCULO INTEGRAL	4.5	0	M.C. NOÉ CHAVEZ AGUILERA			9:30 - 11			8 - 11:00
1211	INTRODUCCIÓN A LA ECONOMIA	4.5	0	LIC. DESIDERIO GUILLERMO GARZA HERNÁNDEZ	10:30 - 13		11 - 13:00			
TR01	TALLER DE ROBÓTICA I - UAA	0	3	DR. DAVID JAIME GONZALEZ MAXINEZ					LAB. DE REDES 8 - 11:00	
TP01	TALLER DE PROGRAMACIÓN II - PROGRAMACION DE LENGUAJE ORIENTADO A OBJETOS EN JAVA- UAA	0	5	ING. CHRISTIAN ALBERTO RAMÍREZ HERNÁNDEZ		8 - 10:30		8 - 10:30		

Figura 8. Ejemplo de formato de horario escolar.

El formato se compone de varias partes, que se describen a continuación:

- 1) Logos.- En ambas esquinas superiores se encuentran los logos institucionales, de lado superior izquierdo encontramos el logo de la Universidad Americana de Acapulco, A.C. y en la esquina superior derecha se encuentra el logo de la Facultad de Ingeniería en Computación tal como vemos en la figura 9.



Figura 9. Logos institucionales en el formato de horario actual.

- 2) Información de identificación del documento.- Como parte del encabezado de cada hoja de horarios, se encuentran: el nombre de la institución educativa, el nombre de la Facultad a la cual pertenece la carrera, el nombre de la carrera, la descripción de lo que se trata el documento (p.e. horario del segundo semestre) y finalmente el grupo al que corresponde, ver figura 10.



Figura 10. Nombres de identificación del formato de horario.

- 3) Plan de estudios.- Especifica el plan de estudios en el que se basa la carga horaria de ese semestre y grupo en particular.
- 4) Ciclo Escolar.- Se refiere al ciclo escolar en el cual este horario es válido.
- 5) Aula.- Se especifica el aula de clases donde se impartirán las asignaturas que se describen en ese grupo específico.
- 6) Clave de Materia.- Es el número que se le asigna a las asignaturas por parte del plan de estudios autorizado por la UNAM.
- 7) Nombre de Asignatura.- Se refiere al nombre completo de la materia, tal cual está relacionado en el plan de estudios autorizado por la UNAM.
- 8) Horas.- Se refiere a las horas que de manera teórica o práctica se deben cumplir para el estudio de esta asignatura según el plan de estudios incorporado a la UNAM.
- 9) Nombre de los Profesores.- Aquí se escribe el nombre completo de los profesores titulares de cada materia.
- 10) Horario.- En esta sección, se distribuye la carga horario dentro de los días de la semana marcados.

Derivado del análisis del formato final que debe generar nuestro sistema como Horario definitivo en cada ciclo escolar, y después de entrevistas con el personal encargado de esta actividad, éste personal trabajó en definir un paso a paso para obtener el algoritmo secuencial para nuestro sistema, mismo que se ha revisado y corregido para ser aprobado.

2.3 Generación de horarios de la carrera de Ingeniería en Computación.

1. Definir ciclo escolar (I ó II par o non) como se muestra en el diagrama que sigue:

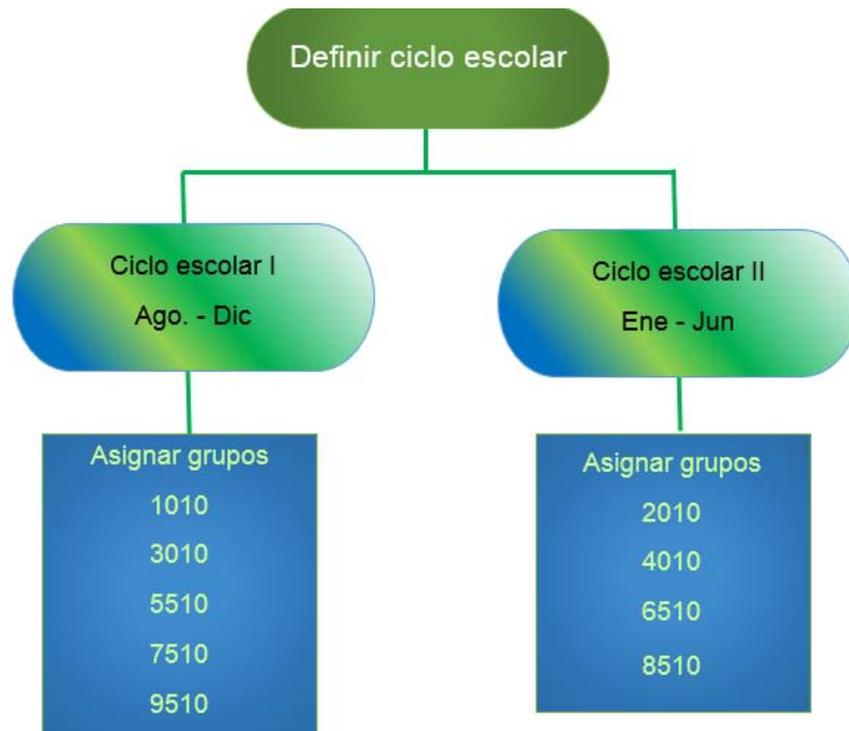


Figura 11. Diagrama para definir ciclo escolar.

2. Crear grupos.
 - 2.1. Los nombres de grupos son conformados por 4 dígitos:
 - 2.1.1. El primer dígito es el semestre del grupo.
 - 2.1.2. El segundo dígito significa el turno del grupo.
 - 2.1.3. El tercer y cuarto dígito corresponden al número de grupo (Campo alfanumérico).
 - 2.2. Asignar materias al grupo creado.
 - 2.2.1 Asignar las materias del semestre de acuerdo al plan de estudios UNAM.
 - 2.2.2 En caso de haber materias optativas, estas se seleccionarán de acuerdo al listado del plan de estudios UNAM.
 - 2.2.3 En caso de haber materias adicionales (UAA), estas se elegirán de acuerdo al listado autorizado de la UAA.
 - 2.2.3.1 En caso de ser materia adicional de nueva creación, se tendrá que dar de alta en el listado de materias adicionales de la UAA.
3. Generar listado de disponibilidad de horarios, basado en la lista de materias acreditadas por la UNAM y las que el profesor ya haya impartido el ciclo anterior inmediato.
4. Se envía este formato de disponibilidad a los profesores ya sea impreso o electrónicamente y se establece fecha última de entrega. Figura 12.



INGENIERÍA EN COMPUTACIÓN
DISPONIBILIDAD DE MATERIAS
 SEMESTRE AGOSTO - DICIEMBRE 2014



PLAN 2009			
TURNO MATUTINO			
PRIMER SEMESTRE	PRIORIDAD	REPETICIÓN	HRS/SEM
ALGEBRA			4.5
GEOMETRÍA ANALÍTICA			4.5
CULTURA Y COMUNICACIÓN			3
CALCULO DIFERENCIAL			4.5
QUIMICA Y ESTRUCTURA DE MATERIALES (L+)			6
TALLER DE MATEMATICAS I			3
TALLER DE PROGRAMACIÓN I - ALGORITMOS Y PROGRAMACIÓN			5
TERCER SEMESTRE	PRIORIDAD	REPETICIÓN	HRS/SEM
CALCULO VECTORIAL			4.5
CINEMÁTICA Y DINÁMICA			4.5
ECUACIONES DIFERENCIALES			4.5
PROGRAMACIÓN AVANZADA Y MÉTODOS NUMÉRICOS			5
PRINCIPIOS DE TERMODINÁMICA Y ELECTROMAGNETISMO			6.5
TALLER DE PROGRAMACIÓN III - ESTRUCTURA DE DATOS I			5
TALLER DE ROBOTICA II - UAA			3
TURNO VESPERTINO			
QUINTO SEMESTRE	PRIORIDAD	REPETICIÓN	HRS/SEM
DISEÑO DE SISTEMAS DIGITALES (L+)			6.5
ESTRUCTURAS DISCRETAS			4.5
INGENIERIA DE SOFTWARE			4.5
SISTEMAS OPERATIVOS			4.5
CIRCUITOS ELECTRICOS (L+)			5
TALLER DE PROGRAMACION IV - ESTRUCTURA DE DATOS II			5
SEPTIMO SEMESTRE	PRIORIDAD	REPETICIÓN	HRS/SEM
COMPUTACIÓN GRÁFICA (L+)			5
ARQUITECTURA DE COMPUTADORAS			3
BASE DE DATOS			4.5
COMPILADORES			4.5
REDES DE DATOS (L+)			6.5
ADMINISTRACIÓN DE PROYECTOS DE SOFTWARE			3
TALLER DE PROGRAMACIÓN V - BD Y DESARROLLO DE PW			3
NOVENO SEMESTRE	PRIORIDAD	REPETICIÓN	HRS/SEM
RECURSOS Y NECESIDADES DE MÉXICO			3
OPTATIVA - TEMAS SELECTOS DE INGENIERIA DE HARDWARE			3
OPTATIVA - DISEÑO DE INTERFACES PARA COMPUTADORAS			3
OPTATIVA - SISTEMAS DIFUSOS			3
OPTATIVA - ROBOTS MOVILES Y AGENTES INTELIGENTES			3
OPTATIVA - PROCESAMIENTO DIGITAL DE SEÑALES			3
OPTATIVA - SEMINARIO DE TITULACIÓN			3
OPTATIVA DE COMPETENCIAS PROFESIONALES - DESARROLLO EMPRESARIAL			3
TALLER DE GENEXUS			3

DISPONIBILIDAD DE HORARIO

Favor de indicar las horas disponibles relleniendo los espacios.

	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
TURNO MATUTINO						
1	7 A 8					
2	8 A 9					
3	9 A 10					
4	10 A 11					
5	11 A 12					
6	12 A 13					
7	INGLÉS MATUTINO					
8	13 A 14					
	14 A 15					
TURNO VESPERTINO						
INGLÉS VESPERTINO						
1	15 A 16					
2	16 A 17					
3	17 A 18					
4	18 A 19					
5	19 A 20					
6	20 A 21					
7	21 A 22					

NOMBRE DEL MAESTRO: _____

COMENTARIOS ADICIONALES: _____

Figura 12. Formato de disponibilidad.

5. Se recupera los formatos de disponibilidad de horario ya lleno con la información de cada profesor.
 - 5.1 En caso de que los profesores entreguen el formato fuera de la fecha límite, no se dará prioridad en la elección de materias, y se asignarán únicamente las disponibles si existiesen éstas.
6. Se analiza la información recabada y se define que materias se le asignarán a cada maestro de acuerdo a su preferencia.
7. Se asigna materia – profesor en los horarios y días disponibles.
8. Se verifica que no se empalmen los horarios de grupo y de profesor en caso de dar más de una asignatura, en caso de que si se empalme algún horario se corrige.
9. Se genera una propuesta de horario que debe ser autorizada por el director, se envía copia a la dirección de Servicios Escolares para que a su vez se envíe a UNAM para autorización, en caso de que UNAM pida cambio de profesores, se repite el proceso hasta obtener la autorización UNAM.
10. Se publican horarios para uso de profesores y alumnos.

Podemos observar mejor este proceso en el diagrama de flujo que sigue:

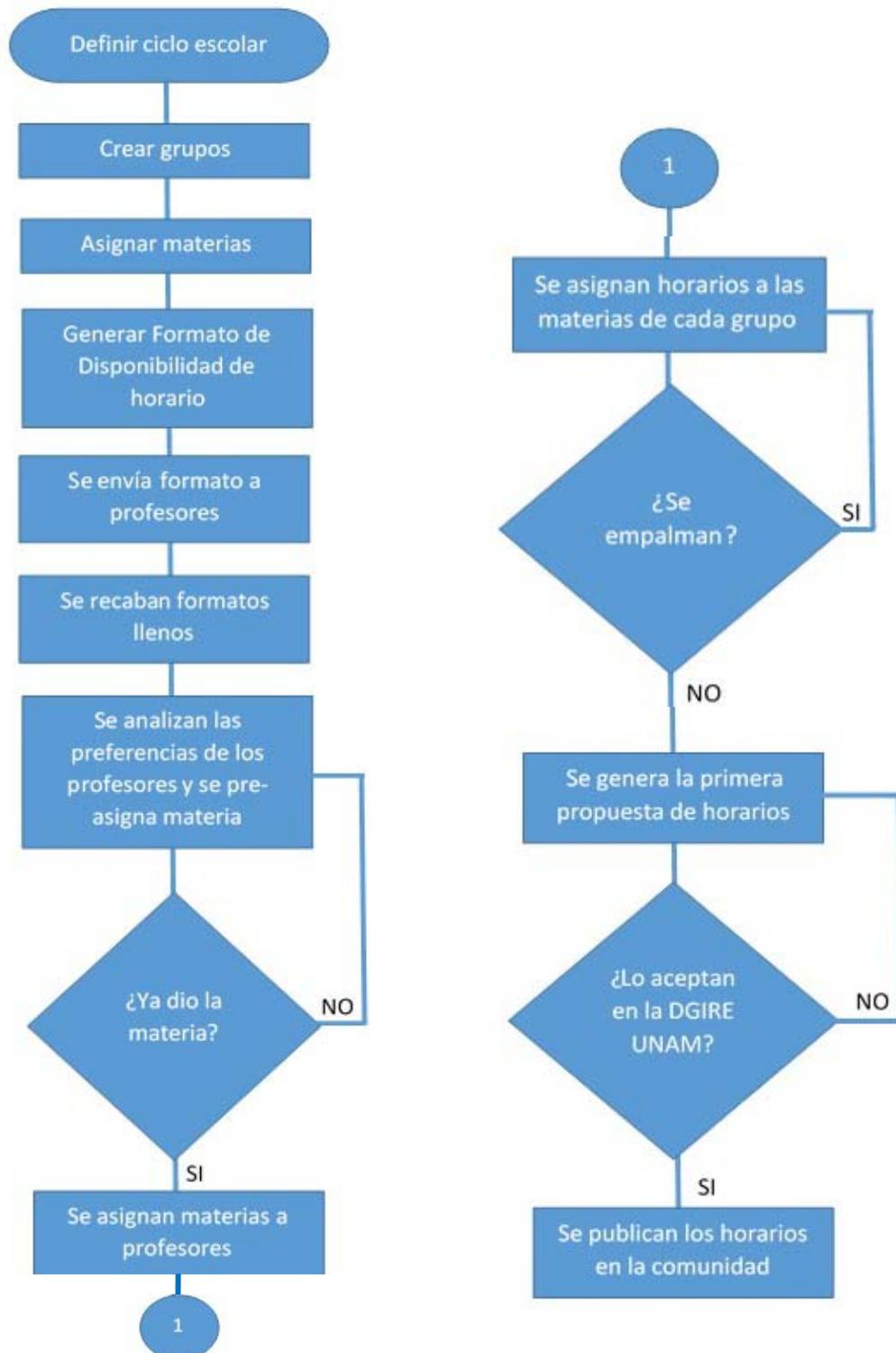


Figura 13. Diagrama de flujo para asignación de horarios.

2.3.1 Registro de Profesores y Grupos ante UNAM.

La Universidad Americana de Acapulco, A.C. pone mucha atención a su Planta Docente, tener maestros capacitados que formen a las mentes jóvenes de nuestro estado no es una tarea fácil, debido a esto la selección de los maestros que impartirán las materias de cada semestre se decide con una serie de procesos y trámites administrativos antes de iniciar cada semestre.

En este caso específico de estudio del nivel licenciatura de la Ingeniería en Computación, quien nos dicta los formalismos que se deben cumplir es la UNAM.

Para que la planta docente que propone la facultad al inicio de cada semestre sea autorizada por la DGIRE-UNAM, los profesores deben cumplir dos requisitos bases: tener título y cedula correspondiente al perfil de la materia que se le desea asignar, y tener la experiencia comprobable en esa área en particular; a la vez que se entregan formatos específicos a la Dirección de Servicios Escolares de nuestra Institución para que a través de su conducto se realicen las gestiones necesarias y se autorice la carga horaria del ciclo escolar correspondiente.

2.3.1.1 Profesores de Nuevo Ingreso y Reingreso.

Durante el proceso de registro de Planta Docente y carga horaria al inicio de cada ciclo escolar que se lleva a cabo ante la UNAM, es importante destacar que existen dos clasificaciones importantes para nuestros profesores: los profesores de Nuevo Ingreso y los profesores de Reingreso.

Los profesores de Nuevo Ingreso al sistema Incorporado UNAM, deben registrarse adecuadamente, con la presentación de documentos oficiales que validen todas las áreas que exige la DGIRE UNAM, tales como título y cedula profesional, acta de nacimiento, CURP, reconocimientos que avalen su experiencia en el área de la materia que se propone para impartir.

Los profesores de Reingreso, son aquellos que ya están debidamente registrados ante el consejo técnico de la DGIRE UNAM, y se consideran aptos para impartir una determinada carga de materias afines a su perfil profesional. Estos profesores cuentan con un número de expediente UNAM y la facultad cuenta con una relación de estas materias en un formato que se describe a continuación y podemos observar en la figura siguiente:

FECHA : 18/11/14

PAGINA : 89

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

DIRECCIÓN GENERAL DE INCORPORACIÓN Y REVALIDACIÓN DE ESTUDIOS

RELACIÓN DE CÁTEDRAS - DOCUMENTO DE TRABAJO -

PROFESOR : DAVILA ZURITA FRANCISCO NARCES EXPEDIENTE : 98001226
R.F.C. : DAZF750705 NACIONALIDAD : MEXICANA GÉNERO : MASCULINO
CRÉDITOS : 0 %
PLANTEL : 8852 UNIV AMERICANA DE ACAPULCO, A.C. PLAN : 16 INGENIERO EN COMPUTACIÓN
ÁREA(S) DE CONOCIMIENTO : 0664 ELECTRICO Y ELECTRONICO, ING. TITULADO

FACULTAD	MATERIA	NOMBRE DE LA MATERIA	F.INICIO	DICTAMEN	F. MODIFICACIÓN
14	0062	ALGEBRA LINEAL	19/02/2013	10	15/03/2013
14	0112	CONTROL ANALÓGICO (L+)	24/09/1998	10	
14	0114	CONTROL DIGITAL (L)	03/02/1998	10	
14	0480	METODOS NUMÉRICOS	08/04/2002	10	
14	0593	PROCESAMIENTO DIGITAL DE SEÑALES	27/03/2001	10	
14	0712	PROBABILIDAD Y ESTADÍSTICA	19/02/2013	10	15/03/2013
14	0755	PROYECTO DE INVESTIGACION	19/02/2013	10	15/03/2013
14	1013	OPTOELECTRÓNICA	23/02/2010	10	17/03/2010
06	1101	MATEMÁTICAS I (ÁLGEBRA Y GEOMETRÍA)	03/02/1998	10	
14	1200	ÁLGEBRA LINEAL	10/12/1999	10	
06	1201	MATEMÁTICAS II (ÁLGEBRA Y GEOMETRÍA)	03/02/1998	10	
06	1301	MATEMÁTICAS III (ÁLGEBRA GEOM ANALÍTICA)	03/02/1998	10	
06	1302	FÍSICA I	03/02/1998	10	
14	1302	ECUACIONES DIFERENCIALES	13/06/2000	10	
14	1306	ECUACIONES DIFERENCIALES	13/04/1999	10	
14	1314	PRINCIPIOS TERMODINAMICA ELECTRO MA	17/09/2013	10	06/11/2013
20	1400	MATEMÁTICAS IV	03/02/1998	10	
06	1401	MATEMÁTICAS IV (ÁLGEBRA GEOM ANALÍTICA)	03/02/1998	10	

Figura 14. Formato de Relación de Cátedras Autorizadas de la UNAM de un profesor actual.

2.4 Análisis de Formato de Relación de Cátedras Autorizadas por la UNAM.

La relación de cátedras de un profesor que ya está registrado en el sistema incorporado a la UNAM, consta de varias partes que nos ayudan a establecer la base de datos de profesores de la carrera de Ingeniería en Computación.

Ésta base de datos se actualiza cada vez que se integra un nuevo profesor a la planta docente y es una parte fundamental en nuestro sistema generador de horarios.

A continuación, se desglosan las partes que conforman este formato:

- Fecha: Es referente a la fecha en que se expide este formato, este rubro es importante porque al ser un proceso cíclico, debemos tener siempre la información más actualizada, nos afecta de manera directa por que una de las condiciones que pide la DGIRE es que el profesor que sea propuesto si es de reingreso, debe haber dado la cátedra en el ciclo anterior donde se impartió esta asignatura, o se debe hacer la petición de aumentar esta asignatura a la tira de cátedras de este profesor.

- Datos de información personal. En este espacio encontramos el nombre completo del profesor con apellidos tal cual se describe en su acta de nacimiento, así como su RFC, y la clave de registro de su título, como el nombre correcto de su área de conocimiento.
- Plan: Se refiere al plan de estudios al cual pertenece el profesor como parte de su planta docente registrada a DGIRE UNAM
- Lista de cátedras autorizadas. Es el listado de todas las cátedras que la DGIRE UNAM autoriza a este profesor para impartir en el sistema incorporado, contiene claves y nombre de asignaturas, así como la facultad a la que pertenecen.

Una vez que el profesor haya sido aceptado dentro del Sistema UNAM, deberá acatar la normatividad aplicada al personal docente, misma que se desglosa en el capítulo VII del Manual de Disposiciones y Procedimientos para el Sistema Incorporado de la UNAM.

2.4.1 Manual de Disposiciones y Procedimientos para el Sistema Incorporado de la UNAM. Capítulo VII. Del Personal Docente.

Art. 118. Para impartir cátedra en una institución del SI, el personal docente deberá obtener, a través de la propia Institución, la autorización correspondiente de la DGIRE; ésta se otorgará conforme a los respectivos perfiles profesiográficos y acuerdos emitidos por las Comisiones de Incorporación y Revalidación de Estudios, y de Títulos y

Grados del H. Consejo Universitario. Sólo tendrán validez los cursos impartidos por personal docente autorizado.

Art. 119. El profesor deberá registrar la información de su expediente digital para que, de ser autorizado, el sistema de cómputo de la DGIRE le asigne un número de expediente y le proporcione los elementos necesarios para que obtenga su firma electrónica, la cual será personal e intransferible.

Art. 120. La autorización podrá ser definitiva o provisional, se concederá por asignatura y permitirá, al docente, impartir en cualquier ISI cátedra de la(s) asignatura(s) del plan de estudios para la(s) cual(es) fue(ron) otorgada(s).

Art. 121. En el nivel licenciatura, todos los profesores deberán tener autorización definitiva.

Art. 122. En el nivel bachillerato, sólo se autorizará a la institución, un máximo del 20% de profesores provisionales. El personal docente que pretenda impartir cátedra y que aún no cuente con el título de licenciatura o con el perfil profesiográfico requerido, podrá impartir clases con autorización provisional hasta por un máximo de tres años escolares en el transcurso de los cuales, deberá acreditar su suficiencia académica, según los requisitos y/o mecanismos de evaluación que establezca la DGIRE, para su autorización definitiva.

Art. 123. Los candidatos para impartir cátedra de Lenguas Extranjeras y de Informática en el bachillerato del SI que no tengan el título de licenciatura requerido en el respectivo perfil profesiográfico, podrán impartir clases con autorización provisional hasta por un máximo de tres años escolares, en el transcurso de los cuales, en su caso, deberán presentar y aprobar, en el lugar que señale la DGIRE, el examen que les permita obtener la autorización definitiva.

Art. 124. Para obtener la autorización provisional a que se refiere la disposición anterior, se requerirá que el docente tenga como mínimo:

- a. Su certificado de bachillerato completo.
- b. Una constancia de estudios que avale los conocimientos para impartir la cátedra correspondiente.

Art. 125. Los profesores que pretendan obtener autorización para impartir cátedra en el SI y hayan obtenido su título en el extranjero, deberán solicitar, ante la DGIRE, la equivalencia de sus estudios, a fin de que la Comisión de Títulos y Grados del H. Consejo Universitario emita el dictamen correspondiente.

Art. 126. La baja de algún profesor en el transcurso del año escolar deberá notificarse a la DGIRE, durante los cinco días hábiles siguientes al hecho. La propuesta de profesor sustituto deberá presentarse en un lapso no mayor de 10 días hábiles, a partir de la fecha en que fue notificada la baja respectiva. Los cambios anuales, en el total de la planta de profesores, en un plan de estudios, no deberán exceder el 20%.

Art. 127. Son obligaciones de los profesores de las ISI:

- a) Realizar sus actividades de acuerdo con los principios de igualdad, equidad y libertad de cátedra.
- b) Cumplir íntegramente con el contenido del programa de la asignatura que imparten, así como con sus objetivos generales y específicos.
- c) Iniciar y concluir sus clases puntualmente.
- d) Elaborar, aplicar y portar en clase el programa operativo para la planeación didáctica y/o el programa de trabajo de laboratorio de su asignatura, según corresponda (Modelo en www.dgire.unam.mx / Profesores/Programa Operativo /Lineamientos).
- e) Entregar y revisar con sus alumnos, al inicio del año o semestre escolares, la síntesis del programa operativo de su asignatura.
- f) Impartir las horas-clase que establece el programa de su asignatura, según el horario convenido con la institución y reportado a la DGIRE.
- g) Firmar y registrar en el kárdex la asistencia y el tema por desarrollar en cada clase.
- h) Llevar un control interno de asistencia y evaluación de sus alumnos por cada grupo. La información respectiva deberá ser entregada al Director Técnico después de cada examen parcial y/o final, y concentrarse en las respectivas actas económicas que deberá firmar al finalizar el año o semestre escolares.

- i) Realizar los exámenes ordinarios, extraordinarios y, en su caso, los de titulación en la hora, fecha y lugar que les señalen las autoridades de la Institución.
- j) Entregar, al Director Técnico, los exámenes calificados en las fechas en que se les indique.
- k) Llenar y firmar las actas de examen ordinario y/o extraordinario que le correspondan.
- l) Autoevaluar el desarrollo del programa operativo, de su asignatura, a la conclusión de cada ciclo escolar. m. Utilizar, con absoluta confidencialidad y responsabilidad, su firma electrónica.
- m) En su caso, dirigir y asesorar tesis; formar parte de los jurados de examen profesional; llenar y firmar las actas correspondientes, y remitirlas al Director Técnico de la Institución dentro del periodo establecido para ello.
- n) Abstenerse de impartir clases particulares, remuneradas o no, a sus propios alumnos.
- o) Enriquecer y actualizar sus conocimientos, tanto en los contenidos temáticos de las asignaturas que imparta, como en la metodología didáctica, participando en, al menos, 20 hrs. anuales de cursos de formación y actualización docentes.
- p) Promover la participación de sus alumnos en actividades extracurriculares organizadas por la UNAM o por cualquier otra instancia académica reconocida ([www.dgire.unam.mx/Extensión y Vinculación](http://www.dgire.unam.mx/Extensión_y_Vinculación)).
- q) Identificarse, a requerimiento de la DGIRE, con su credencial UNAMSI.

Art. 128. Cuando, por causa de fuerza mayor no se cubra el 100% del programa correspondiente, el profesor deberá presentar, oportunamente, al Director Técnico, un plan de recuperación académica. La DGIRE podrá solicitar los exámenes finales ya calificados, para su revisión.

Art. 129. Cuando la DGIRE, detecte deficiencias en la práctica docente, será obligatorio que el profesor participe y acredite las actividades de formación y/o actualización que ella indique.

Si se desea abundar en esta información, puede hacerlo a través del portal, <http://www.dgire.unam.mx> ⁴

⁴ <http://www.dgire.unam.mx> / normatividad / Manual de Disposiciones / índices

3

Capítulo

Conceptos y
Definiciones

Capítulo 3. Conceptos y Definiciones Básicas.

Aún cuando un sistema generador de horarios puede ser desarrollado en diferentes lenguajes de programación, en este caso en específico de estudio para la carrera de Ingeniería en Computación de la Universidad Americana de Acapulco, A.C. se decidió por ocupar el lenguaje de programación Visual Basic para realizar los cálculos y el manejo de datos; junto con el sistema gestor de base de datos de Access para elaborar ventanas y el manejo de información.

Esta decisión se tomó puesto que son lenguajes ligeros que no ocupan mucha memoria y de fácil manipulación para el usuario, además de que facilitará el trabajo de incorporación de bases de datos con los que ya cuenta el personal administrativo actualmente, las cuales se encuentran en Excel y la impresión de su horario es en el mismo formato, causando así que la implementación de éste sistema propuesto no conlleve mucho tiempo e interfiera con otras actividades del personal.

3.1 Bases de Datos.

3.1.1 ¿Qué son las bases de datos?⁵

Las actividades productivas y de servicios requieren de un volumen de datos considerable para desarrollarse de manera efectiva, y necesitan que estos se encuentren organizados con relación a la actividad que se desea desarrollar. Por lo tanto, disponer de información relevante, en forma rápida y sencilla, es imprescindible para tomar decisiones acertadas. Para lograr su correcta y dinámica organización disponemos de una de las herramientas más potentes para su manejo: Access.

Esta aplicación permite gestionar contenido mediante bases de datos (estructura donde se guarda la información de forma organizada), de una manera ágil y completa.

Una base de datos se define como una colección de datos organizados sistemáticamente en formato de tablas.

⁵ Access / coordinado por Daniel Benchimol. - 1a ed. -, Buenos Aires: Fox Andina; Banfield - Lomas de Zamora: Gradi, 2011. v. 14, 192 p. ; 19x15 cm. - (Desde cero), ISBN 978-987-1773-11-4, 1. Informática. I. Benchimol, Daniel, coord. CDD 004.1

Estas nos permiten generar nueva información a partir de dichos datos y a través de diversos procesos, haciendo uso de campos (columnas) y registros (filas).

Si bien una base puede contener infinitas tablas, es conveniente que definamos previamente su estructura, para no generar datos duplicados. Lo primero que debemos hacer es tomar lápiz y papel y realizar un esquema de las actividades que queremos registrar, teniendo en cuenta la forma de distribución de la información en tablas.

Debido a que estas son los elementos más importantes de una base de datos, gran parte de nuestro trabajo se centrará en su apropiada construcción y posterior mantenimiento.

Existen distintos tipos de bases de datos. En la Tabla 1 se presenta cada uno de ellos, con la información que puede albergar.

TIPO DE BASE DE DATOS	GESTIÓN BRINDADA
Bases de datos estáticas	Son bases de solo lectura, usadas para guardar información histórica. Son consultadas para estudiar el comportamiento de un conjunto de datos través del tiempo y, luego, poder tomar decisiones.
Bases de datos dinámicas	Son bases donde la información almacenada sufre modificaciones con el tiempo. Permiten realizar operaciones de consulta, borrado o modificación de registros.
Bases de datos bibliográficas	Contienen información de la fuente primaria para poder localizarla. Un ejemplo es una base de datos bibliográfica, con autores, fechas de publicaciones, editoriales y ediciones de determinadas publicaciones, junto con un probable resumen del contenido de cada una de ellas.
Bases de datos de texto completo	Almacenan una fuente primaria, como el contenido de todas las ediciones de una colección de revistas científicas o de saber popular.
Bases de datos de directorio	Almacenan, por ejemplo, un directorio de teléfonos de una ciudad.

Tabla 1. Tipos de bases de datos.

Por otro lado, las bases de datos también se dividen en modelos, que indican la administración de información. Los modelos son descripciones de la abstracción de su contenido de datos, como

también el tipo de método utilizado para almacenar y recuperar los registros.

En la Tabla 2 encontramos los distintos modelos de bases de datos y sus usos más comunes.

MODELO DE BASE DE DATOS	APLICACIONES MÁS COMUNES
Jerárquicas	Almacenan la información en una estructura jerárquica o de árbol, donde un nodo denominado padre puede tener varios nodos hijos. Son útiles para aplicaciones que deben manejar un gran volumen de información y datos compartidos
De red	Difieren muy poco del modelo jerárquico, dado que cambia su concepto de nodo: un mismo nodo puede tener varios nodos padres. Este modelo mejoró la posibilidad de redundancia de datos, que el jerárquico no podía manejar de manera óptima.
Transaccionales	Permiten enviar y recibir datos a altas velocidades. No son muy comunes, y se utilizan más en el ámbito del análisis de la calidad y para datos de producción industrial.
Relacionales	Este modelo es el más utilizado en la actualidad para administrar los datos de forma dinámica. Se creó en 1970 en los laboratorios de IBM y se consolidó con un nuevo paradigma de los modelos de base de datos ideal para relacionar la información contenida. Fue el primer modelo en utilizar lo que se conoce como normalización de datos. En la década del 80, Dbase fue la primera base de datos en utilizar este modelo.

Multidimensionales	Este modelo fue ideado para desarrollar aplicaciones que permiten, por ejemplo, la creación de cubos OLAP. No son muy distintas de las bases de datos relacionales.
Orientadas a objetos	Este modelo es muy reciente, y se aplica mucho en los tiempos que corren. Almacenan los objetos completos, con su estado y comportamiento. Utilizan encapsulación, herencia y polimorfismo.

Tabla 2. Modelos de bases de datos, más usados en las últimas décadas.

3.1.2 Estructura de almacenamiento de datos.

Las bases de datos deben estar estructuradas de tal manera que toda la información que contengan sea de fácil acceso. Para esto, su estructura debe dividirse correctamente en tablas, campos y registros.

1. Tablas. En una base de datos, una tabla es un medio donde se concentra la información de los campos y registros de manera lógica. Cada base puede contar con una o miles de tablas, dependiendo de la complejidad de la información que almacene.
2. Campos. Los campos se almacenan dentro de las tablas, y contienen la información sobre el tipo de datos que se guardará en ellos: numéricos, alfanuméricos o binarios; a su vez, de cada uno de ellos se pueden desprender nuevos tipos.

Vemos en la Tabla 3 qué tipos de datos pueden contener los campos.

TIPO DE DATOS	USO COMÚN
Numérico	Pueden ser enteros o reales. Los primeros son números sin decimales, y los segundos pueden contener decimales.
Booleanos	Solo poseen dos estados: verdadero o falso.
Memo	Es un campo alfanumérico de longitud ilimitada.
Fechas	Permiten almacenar fechas para ordenar de manera cronológica o calcular períodos de tiempo.
Alfanuméricos	Admiten números y letras. Se limitan a 255 caracteres como máximo.
Autoincrementables	Comúnmente son campos numéricos que incrementan su unidad de valor por cada nuevo registro agregado. Sirven como identificador único de cada registro en una tabla.

Tabla 3. Tipos de datos que puede almacenar una tabla.

Existen convenciones determinadas para los tipos de datos que se almacenarán en cada campo. Si bien un campo de tipo alfanumérico puede contener solo números, estos no podrían ser nunca procesados por el lenguaje de la base de datos para entregar, por ejemplo, una suma total, porque al no ser un tipo de campo numérico, no podrá ser sumado.

3. Registros. Se denomina registro al conjunto de información que se almacena en una tabla, indicando cada dato en su campo correspondiente.

3.1.2.1 Índices.

Los índices son campos que permiten asociar una tabla con una o varias columnas de otras tablas. Determinan una relación entre el contenido y el número de fila donde está ubicado el registro. Estos tipos de campos permiten agilizar las consultas a las tablas, y evitar así que el motor de la base de datos deba revisar uno a uno los registros hasta dar con el que estamos buscando para devolver un conjunto de resultados.

Cuando creamos una tabla, lo ideal es que en ese momento establezcamos el campo índice. Un índice puede contener una o más columnas por indexar.

Para los motores de bases de datos, un índice ágil generalmente es el denominado por un dato numérico.

3.1.2.2 Vistas.

Las bases de datos suelen estar compuestas por muchas tablas. Por lo general, debemos combinar información de distintas tablas para mostrar en pantalla o, simplemente, necesitamos mostrar algunos campos de una tabla. Para llevar a cabo estas acciones, siempre es conveniente establecer una Vista de datos que nos permita acceder de manera rápida a estos registros.

En los motores de bases de datos más populares esto se conoce como Vistas. Las vistas agrupan los campos que necesitamos ver de una o más tablas en un conjunto de datos de solo lectura.

3.1.2.3 Entidad – Relación.

Las bases de datos conllevan un modelo de Entidad-Relación, a través del cual se logra normalizar la información contenida para que sea lo menos redundante posible. Esto permite establecer una dinámica en la respuesta sobre las consultas de datos.

Este modelo se sustenta en diferentes conceptos que permiten representar como resultado un modelo de la vida real.

- Entidad. La entidad representa un objeto del mundo real independiente que se diferencia de otros similares, aunque sea del mismo tipo de sus pares.

- **Atributos.** Los atributos, son propiedades que describen a una entidad en un conjunto de entidades. Cada entidad dispone de valores específicos asignados a sí misma para lograr una identificación unívoca.
- **Relación.** Una relación equivale a una cierta dependencia entre las entidades. Un conjunto de personas, cada una con atributos distintos, pueden estar relacionadas entre sí, por ejemplo, por su nacionalidad o porque trabajan en una misma compañía.

3.1.2.4 Relaciones entre tablas.

Todas las tablas de una misma base de datos pueden relacionarse entre sí a través de uno o más campos que permitan representar una relación de la misma naturaleza.

Las reglas de una relación pueden mantener restricciones, que no deben quebrantarse a menos que haya una relación de una tabla de un solo registro con muchos registros de otra tabla. El conjunto de relaciones del que participan dos o más entidades se establece con la correspondencia de cardinalidad, la cual indica el número de entidades con la que puede estar relacionada una sola entidad.

Suponiendo que tenemos un conjunto de entidades denominadas A y B, entre ellas puede haber más de un tipo de cardinalidad. Veamos en detalle la cardinalidad en la tabla 4.

TIPO DE RELACIÓN	DESCRIPCIÓN
Uno a uno	Una entidad de A se relaciona de manera directa con una entidad de B (y viceversa).
Uno a varios	Una entidad de A se relaciona con ninguna o muchas entidades de B, pero una entidad de B se relaciona con una única entidad de A.
Varios a uno	Una entidad de A se relaciona de manera exclusiva con una entidad de B, y una entidad de B puede relacionarse con ninguna o muchas entidades de A.
Varios a varios	Una entidad de A se relaciona con ninguna o muchas entidades de B (y viceversa).

Tabla 4. Un detalle de los tipos de cardinalidad existente en una base de datos.

3.2 Introducción a Access.

Existen muchas maneras de organizar la información relevante de una determinada gestión por medio de una PC. Sin embargo, la forma más eficiente de lograrlo es a través de la herramienta más potente llamada Microsoft Access 2007. Ésta permite almacenar y organizar los datos de tal forma que el usuario pueda controlarlos en todo momento y realizar diversas tareas según necesite.

Lo que realicemos en Access podemos aplicarlo en las tareas de nuestra vida cotidiana, ya sea para registrar información o para administrar la totalidad de los datos de una empresa: tablas, consultas, informes, formularios y macros.

3.2.1 Conociendo el programa.

Microsoft Access 2007 funciona bajo el sistema operativo Windows y viene incluido dentro del paquete Microsoft Office, de manera que debemos tener este sistema instalado en nuestro equipo.

Para acceder al programa, podemos desplegar las opciones del menú Iniciar o utilizar el icono identificativo del programa que suele ubicarse en el Escritorio de Windows.

Para iniciar el programa, podemos desplegar las opciones del menú Inicio o también utilizar el icono identificativo del programa situado en el Escritorio de Windows, tal como se muestra en la figura 15.



Figura 15. Icono de inicio de Access.

A continuación, nos encontramos con la pantalla de bienvenida, que nos permite elegir la forma de empezar a trabajar: desde una plantilla o con la creación de una base de datos en blanco

En este punto, el panel izquierdo de la pantalla nos permite seleccionar una determinada categoría de plantillas, que se mostrarán en el centro de la ventana representadas por un icono ilustrativo correspondiente con el tema de cada categoría, ver figura 16.

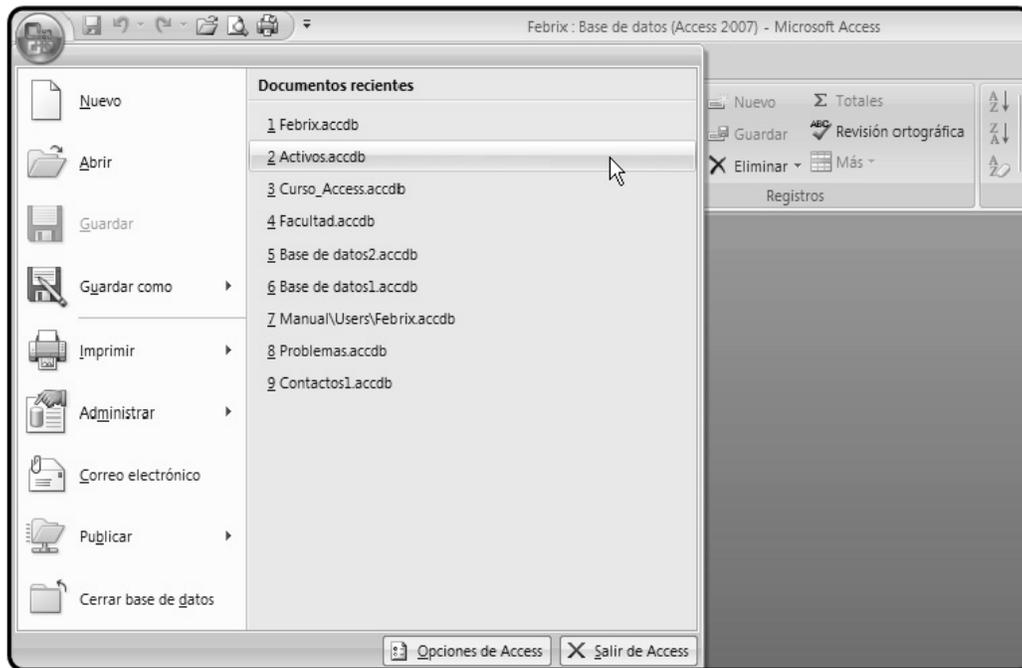


Figura 16. Pantalla de Bienvenida.

3.2.2. La interfaz.

La interfaz de Access posee elementos para manipular la ventana del programa, independientemente del contenido de sus bases de datos. La versión 2007 incorpora un diseño de entorno dinámico para la realización de todas las tareas.

Para comenzar, veamos la barra de título, que muestra el nombre de la base de datos en la cual estamos trabajando y el nombre del programa. Además, posee los botones Minimizar, Maximizar y Cerrar, así como también la barra de acceso rápido y el botón de Office.

En la figura 17 se muestra más a detalle los elementos que conforman el panel de exploración de este programa:

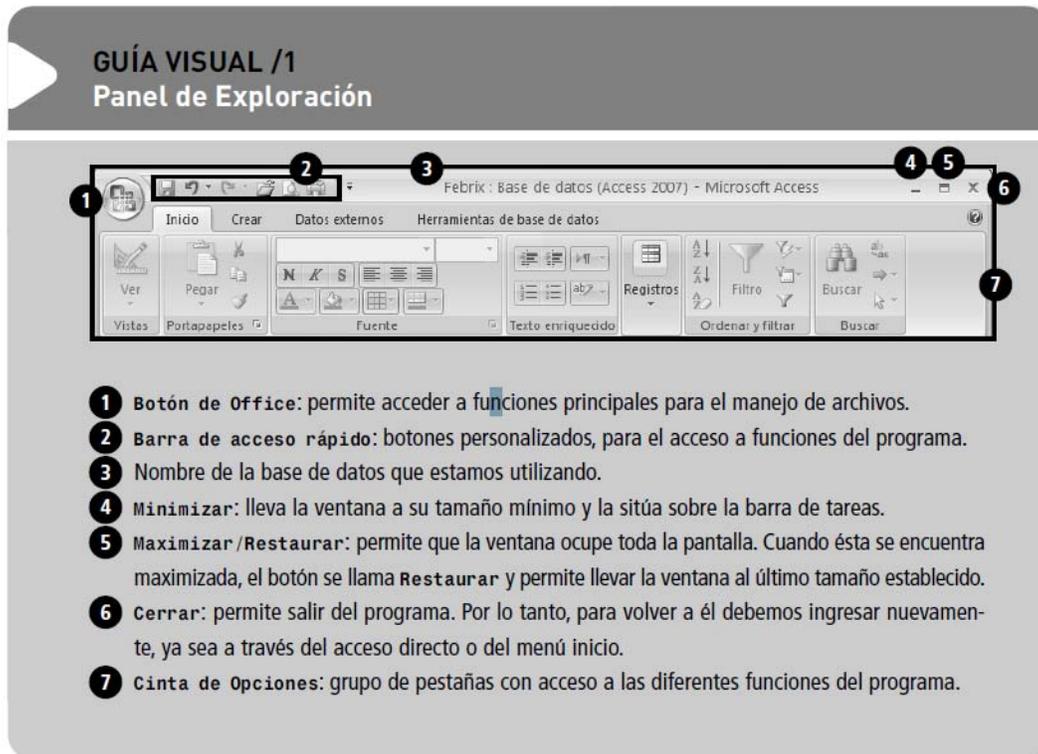


Figura 17. Panel de exploración de Visual Basic.

3.3 Introducción a Visual Basic.

El mundo informático es una pasión que cada día atrapa a más personas, quienes usan programas destinados a cubrir diversas necesidades. Si bien el software no se palpa, es real; y por más sencillo que sea su cometido, requirió tiempo, ingeniería e idealización de una o más personas para implementarlo.

Año a año se incrementa la demanda de expertos en el mundo binario y son contadas las empresas que no dependen de una computadora o de un sistema a medida.

También el desarrollo de software está disfrutando de un crecimiento ininterrumpido en cualquier país del mundo. Convertirse en desarrollador requiere de algunos conocimientos puntuales y de mentes bien abiertas, pero gracias a la facilidad de las herramientas de que disponemos en la era de la computación, más la biblioteca universal de información abierta las 24 horas del día, como lo es Internet, el aprendizaje de esta profesión puede llevarse a cabo en tiempo récord.

En poco más de 30 años, el paradigma de la programación ha dado muchas vueltas de tuerca a beneficio de las personas, al dejar de ser un mundo estrecho y cerrado a unos pocos ingenieros, para convertirse en la pasión de millones de personas interesadas en este campo.

Microsoft, la empresa desarrolladora de software que contribuyó mucho a cambiar el panorama de la informática, ha jugado un papel importante en este terreno, por tener una amplia visión de futuro y prever las necesidades del usuario final, quien no solo se interesa por la computación, sino que también quiere avanzar en el fascinante mundo de la programación de aplicaciones para computadoras.

Visual Basic ha evolucionado tanto, que hoy en día permite llegar, de manera fácil y práctica, al desarrollo de aplicaciones de escritorio, web y otros campos que hoy nos son cada vez más cotidianos, como la telefonía celular, poniendo a disposición de los interesados la potencia y la flexibilidad de una herramienta que ya tiene más de cuarenta años.

3.3.1. Reseña del lenguaje.

En 1964, John Kemeny y Thomas Kurtz se propusieron crear una variante del lenguaje de programación denominado BASIC (*Beginners All-Purpose Symbolic Instruction Code*, Código de instrucciones simbólicas para principiantes orientado a todo propósito).

Si bien ya había en el mercado varias versiones de BASIC, la generada en Dartmouth College por Kemeny y Kurtz fue la que más se popularizó.

BASIC llegó para reducir de una manera notable los tiempos de aprendizaje y de escritura de un programa para computadoras. Así, gracias a la visión de futuro de sus propulsores, muchos estudiantes pudieron desarrollar aplicaciones en tiempo récord, algo muy valioso para esa época; ésta versión fue el puente entre la versión BASIC para DOS y la versión Visual Basic para Windows.

Al momento de diseñar el lenguaje, sus creadores tomaron como filosofía ocho principios que debían destacar a BASIC por sobre los demás:

1. Ser fácil de usar.
2. Ser un lenguaje de propósito general.
3. Permitir la incorporación de características avanzadas por expertos, priorizando su facilidad para principiantes.
4. Gozar de interactividad.
5. Ofrecer claros mensajes de error.
6. Brindar rápida respuesta en programas pequeños.
7. No requerir que los usuarios tengan conocimientos sobre hardware.
8. Alejar al usuario de la complejidad del sistema operativo

Hacia 1975, la empresa fundada por Bill Gates y Paul Allen lanzó su propia versión de BASIC, inspirada en una de las alternativas del lenguaje, creada por Alan Cooper, denominada Altair BASIC. El mercado siguió inundándose con más versiones opcionales, y para fines de la década del 70, apareció la primera adaptada a la plataforma Apple II. En 1979, Microsoft negoció vender su licencia de BASIC a varias empresas que comercializaban microcomputadoras, incluyendo a IBM, creadora en ese entonces de la computadora personal. BASIC se incorporó en los chips ROM de las IBM PC, con lo cual se puso una versión innovadora en equipos que no contaban con disco rígido, pero que sí disponían ya de una unidad de disquete.

Años más tarde, Microsoft siguió distribuyendo una versión reducida de BASIC junto a su popular sistema operativo **MS DOS**, la cual permitía a los programadores diseñar aplicaciones que sólo podían ejecutarse a través del entorno de desarrollo utilizando una serie de instrucciones **BATCH**, para que el programa se ejecutara casi sin intervención de los usuarios poco expertos. En la entrada era donde DOS pasaba a un segundo plano. Luego la firma lanzó al mercado **Visual Basic 1.0**, un entorno de desarrollo que facilitaba la creación de aplicaciones con menús, ventanas y botones, pero que aún corría bajo DOS. Recién en la versión 2.0, desarrollada para **Windows 3.0/3.1**, Microsoft dejó de lado el entorno gráfico construido mediante caracteres **ASCII**, para dar inicio a una era distinta: la era **RAD** de desarrollo de aplicaciones para Windows.

Desde **Visual Basic 3.0**, Microsoft fortaleció el desarrollo de aplicaciones RAD orientado a bases de datos y dio un gran soporte al lenguaje para conectarse a cualquier base entre las más populares del mercado (**Dbase, Paradox, Fox Pro**), al utilizar las librerías de enlace dinámico a través de **ODBC** (*Open Data Base Connectivity*). Para las bases de datos que no eran tan difundidas en ese momento, solo restaba que la empresa que las soportaba creara una librería DLL para que Visual Basic pudiera conectarse e interactuar con ellas y, así, leer, mostrar y escribir información en sus archivos.

La versión 4.0 llegó al mercado casi al mismo tiempo que **Windows 95**, con lo cual se lanzó una edición doble, para 16 y 32 bits, que podía instalarse en **Windows 3.1x** o **Windows 95**, con la diferencia

de que todos los proyectos creados en 16 bits podían ser portados a 32 bits, pero no a la inversa.

Visual Basic 5.0 contó con una versión lite denominada CCE (Control Creation Edition), en la que no solo era posible crear librerías DLL y archivos ejecutables, sino que también se habilitaba a los programadores a generar controles personalizados, combinando dos o más controles ActiveX existentes. También se facilitaba la incorporación de nuevos eventos y propiedades a los controles predefinidos que se incluían con el entorno de desarrollo.

En 1998 Microsoft introdujo **Visual Basic 6.0**. En la era donde Internet se devoraba cualquier mercado, esta versión tuvo que aportar flexibilidad para dar paso al desarrollo de aplicaciones web y no solo de escritorio. Con Visual Basic 6 se podían crear controles personalizados, programas ejecutables bajo la plataforma Windows, librerías DLL y aplicaciones web, incluyendo soporte para el lenguaje ASP (Active Server Pages), que permitía generar páginas web dinámicas que se compilaban al momento de ser solicitadas en el servidor.

El nuevo milenio trajo consigo un giro total de sus lenguajes de programación, que se orientaron al framework .NET. Los **ActiveX** pasaron a segundo plano, y con esto, también varios problemas de seguridad que sufría Microsoft Windows.

3.3.2 Bases de Datos y Visual Basic.

Visual Basic, en todas sus versiones, está preparado para trabajar con bases de datos. El entorno de desarrollo está listo para conectarse y realizar operaciones con dos bases de datos populares propias de **Microsoft: Access y SQL Server**. SQL Server es un motor de base de datos profesional, que debe adquirirse por separado, aunque desde su versión 2005, Microsoft decidió lanzar al mercado una versión Express gratuita del motor de base de datos, orientada más a la práctica de los usuarios.

3.3.3 Herramientas Complementarias.

Crystal Reports fue el primer software para la generación de reportes visualmente atractivos, que pertenecía a una empresa ajena a los intereses de Microsoft, pero que supo complementarse bien con este lenguaje para triunfar.

Microsoft Access, base de datos que pertenece aún hoy al paquete ofimático Microsoft Office, es otro complemento ideal para quienes buscan crear aplicaciones pequeñas que no requieran de un potente y costoso motor de base de datos.

Microsoft SQL Server, desde la versión 4.x, comenzó a ser un aliado para Visual Basic en el desarrollo de aplicaciones empresariales

que requirieran un robusto motor de base de datos y necesitaran manejar grandes volúmenes de información.

3.4 Lenguajes de programación.

Un lenguaje de programación es cualquier lenguaje con el que los humanos pueden dar instrucciones a las computadoras.

Un programa es una lista de instrucciones preparadas para ser entregadas a una computadora y ser ejecutadas. Esencialmente, las instrucciones almacenan, mueven y cambian valores en la memoria de la computadora.

Estos valores pueden ser interpretados por varios dispositivos, como monitores, altavoces, teclados, touch pads o ratones, que proporcionan formas de comunicación con el usuario.

3.4.1 Diferencias entre los lenguajes de programación de alto y bajo nivel.

Un lenguaje de programación de alto nivel se caracteriza por expresar los algoritmos de una manera adecuada a la capacidad cognitiva humana, en lugar de a la capacidad ejecutora de las máquinas.⁶

En los primeros lenguajes de alto nivel la limitación era que se orientaban a un área específica y sus instrucciones requerían de una sintaxis predefinida. Otra limitación de los lenguajes de alto nivel es que se requiere de ciertos conocimientos de programación para realizar las secuencias de instrucciones lógicas. Los lenguajes de alto nivel se crearon para que el usuario común pudiese solucionar un problema de procesamiento de datos de una manera más fácil y rápida.

Por esta razón, a finales de los años 1950 surgió un nuevo tipo de lenguajes de programación que evitaba estos inconvenientes, a costa de ceder un poco en las ventajas. Estos lenguajes se llaman de tercera generación o de alto nivel, en contraposición a los de bajo nivel o de nivel próximo a la máquina.

En otras palabras, un lenguaje de programación de alto nivel es el que se asemeja al lenguaje humano.

⁶ Wikipedia

Características:

- Genera un código más sencillo y comprensible.
- Escribir un código válido para diversas máquinas y, posiblemente, sistemas operativos.
- Reducción de velocidad al ceder el trabajo de bajo nivel a la máquina.
- Algunos requieren que la máquina cliente posea una determinada plataforma.

Un lenguaje de programación de bajo nivel es el que proporciona un set de instrucciones aritméticas y lógicas sin la capacidad de encapsular dichas instrucciones en funciones que no estén ya contempladas en la arquitectura del hardware. Esto es, lenguaje de máquina puro y duro. El ensamblador es un buen ejemplo de ello.

Características:

- Adaptación máxima entre programación y aprovechamiento del recurso de la máquina.
- Velocidad máxima al contar con un acceso directo a los recursos, sin capas intermedias.
- Portabilidad mínima por estar restringido a las especificaciones del fabricante.
- Abstracción mínima por depender completamente de la técnica del hardware.

Los lenguajes de programación de computadoras de alto y de bajo nivel están diseñados para permitir la comunicación entre un humano y una computadora a distintos niveles de abstracción. Un lenguaje de muy bajo nivel requiere que un humano proporcione instrucciones directamente al hardware de la computadora, usando el lenguaje y la estructura de dicho hardware; en el caso de los lenguajes de alto nivel, los humanos trabajan con herramientas lógicas complejas y abstractas para escribir instrucciones que un programa determinado debe traducir para que la computadora comprenda. Generalmente es más fácil para los humanos usar estas herramientas.

3.4.2 El Código Máquina.

Un lenguaje de bajo nivel es aquel que se encuentra cerca de los conceptos básicos del hardware de la computadora. El lenguaje de más bajo nivel es el código máquina, que el hardware entiende directamente y que no requiere interpretación o traducción.

El código máquina está compuesto completamente por cadenas de números binarios: los famosos ceros y unos. Si bien puede hacer todo lo que cualquier otro lenguaje puede lograr (de hecho el resto de los lenguajes deben ser traducidos a código máquina por el intérprete), no está diseñado para cumplir con las necesidades del programador y requiere un conocimiento profundo del procesador y de los dispositivos de la computadora, además de que es casi imposible que un humano pueda leerlo.

3.4.3 El Lenguaje Ensamblador.

El siguiente lenguaje de programación de "más alto" nivel es el lenguaje ensamblador, que se trata de código máquina cuyos códigos de instrucciones han sido reemplazados por comandos más intuitivos. Por ejemplo, el comando para colocar el valor 97 en un registro de la memoria es llamado AL (en código máquina sería "10110000 01100001"), y en lenguaje ensamblador sería "MOV AL, 97", que sigue siendo un comando críptico pero es mucho más sencillo de leer. Aunque sigue siendo ineficiente escribirlo y requiere que el programador trate directamente con el hardware de la computadora, es un lenguaje de nivel más alto que el de código máquina porque es más abstracto, es decir, está más cerca del programador.

3.4.4 Los Lenguajes de Alto Nivel.

Los programadores modernos muy rara vez escriben en lenguaje ensamblador. En su lugar, usan uno de los muchos lenguajes de alto nivel como C, Java o Python, cuyos programas llamados compiladores o intérpretes pueden traducir al código máquina.

Estos lenguajes evitan que el programador tenga que involucrarse con el mundo físico del hardware y lo llevan a la abstracción lógica: en lugar de mover valores hexadecimales por los registros de la memoria, el programador trabaja con variables cuyos contenidos pueden ser modificados, con bucles que pueden repetirse

hasta que se cumpla una condición, con sentencias lógicas como IF, AND, THEN, OR y ELSE, entre otras herramientas. Estos lenguajes están diseñados para el programador, atendiendo a las formas en las que se puede lograr la máxima potencia con la menor dificultad.

3.4.4.1 Comparaciones.

Los lenguajes de alto nivel no reciben ese nombre por ser "mejores" que los lenguajes de bajo nivel.

Un lenguaje de muy alto nivel puede tratar sólo con un sistema operativo determinado, como Microsoft Visual Basic, o con un programa determinado, como las "macros" de un procesador de textos.

Estos programas son muy útiles para cualquiera que quiera manipular Windows o Word sin tener que saber cómo funciona, pero un lenguaje así no sería útil para alguien que intente escribir un programa propio y que necesita un lenguaje de nivel más bajo. Por lo tanto, un programador elige un lenguaje dependiendo del trabajo que necesite llevar a cabo.⁷

⁷ http://www.ehowenespanol.com/diferencias-lenguajes-programacion-alto-nivel-nivel-info_191818/

3.6 Justificación del lenguaje utilizado en el Programa de Generación de Horarios.

La siguiente comparativa es una justificación del porqué se decidió usar Microsoft Visual Basic y Microsoft Access como tecnologías a implementar para el desarrollo de éste proyecto. En la tabla a continuación se compararán algunos DBMS (sistemas administradores de bases de datos, por sus siglas en inglés Database Management System) más populares, tanto software propietario (software no libre) como open source (software libre).

DBMS	Base de Datos Relacional	Gratuita	Instalable	Requiere Administrador	Requiere Servidor Dedicado
MS Access	Sí	No	Sí	No	No
Oracle	Sí	No	Sí	Sí	Sí
SQL Server	Sí	No	Sí	Sí	Sí
MySQL	Sí	Sí	Sí	Sí	Sí
PostgreSQL	Sí	Sí	Sí	Sí	Sí

Tabla 5. Comparativa de DBMS (Sistemas Administradores de Bases de Batos, por sus siglas en inglés Database Management System).

MS Access.

Microsoft Access es la base de datos elegida para el desarrollo de este proyecto. Su propietario es el mismo que el propietario de Visual Basic, Microsoft, lo que hace que estas dos tecnologías sean altamente compatibles proveyendo muchas herramientas para la comunicación entre ellas. Otra razón aún más importante por la cual se eligió ésta tecnología es que la institución ya cuenta con las licencias necesarias para el uso de la base de datos MS Access, por lo tanto no se requiere de una inversión para adquirir nuevas licencias de uso.

Como se muestra en la tabla comparativa, MS Access no requiere de un servidor dedicado para su instalación e implementación. Puede ser instalado en una o más computadoras en las que se desee tener acceso a la base de datos. Además, la migración, exportación e importación de datos es relativamente fácil de realizar en comparación con la migración de datos de los otros sistemas en donde se requieren conocimientos especializados en administración de bases de datos.

Oracle.

En resumen, Oracle es una base de datos de alto rendimiento implementada principalmente por grandes empresas las cuales manejan grandes cantidades de información. Los costos de licencia de uso de Oracle Enterprise van desde los 50 mil dólares por licencia. Los

precios pueden variar de acuerdo a diferentes conceptos como el número de usuarios/empleados que tendrán acceso al sistema, etc.

SQL Server.

A pesar de que SQL Server es una base de datos desarrollada por Microsoft, no se eligió como sistema de base de datos para implementar en el actual proyecto ya que su uso requiere de la adquisición de una licencia (independiente de la licencia para el uso de MS Access). Además, dado que SQL Server es una base de datos más especializada, su manejo requiere de un personal profesional con conocimientos y experiencia en administración de bases de datos que den mantenimiento frecuente tanto al sistema.

Otro punto importante para mencionar es que el DBSM SQL Server requiere ser instalado en un servidor dedicado; el servidor debe tener la suficiente capacidad para que la base de datos trabaje con buen rendimiento.

Por último, los sistemas administradores de bases de datos MySQL y PostgreSQL a pesar de ser bases de datos gratuitas (no requieren de la compra de ningún tipo de licencia para su uso), sí requieren de conocimientos en administración de bases de datos, un mantenimiento constante, además de un servidor dedicado en donde puedan ser instaladas y asegurarse de su funcionamiento y rendimiento.

En la siguiente tabla, se hace una comparación entre los lenguajes de programación más populares y se proveerá una justificación del porqué se eligió el uso de Visual Basic para el desarrollo del proyecto. La siguiente tabla muestra algunas características de los lenguajes de programación descritos anteriormente en éste documento.

Lenguaje de programación	Fácil de aprender	Orientado a Web	Aplicaciones Instalables	Requiere Administrador	Requiere Servidor Dedicado
Visual Basic	Sí	No	Sí	No	No
Java	No	Sí	Sí	No	No
C#	No	No	Sí	No	No
PHP	No	Sí	No	Sí	Sí
JavaScript	Sí	Sí	No	No	No

Tabla 6. Comparativa de Lenguajes de Programación más populares.

Microsoft Visual Basic.

Una vez más, Microsoft es el propietario de Visual Basic.

Realmente, Visual Basic no es un lenguaje de programación como tal sino más bien un ambiente de desarrollo integrado (llamado también IDE, por sus siglas en inglés Integrated Development Environment), el cual facilita el desarrollo de aplicaciones a través de un ambiente gráfico altamente intuitivo y fácil de usar.

Los programas generados por Visual Basic son archivos ejecutables con extensión .exe, los cuales pueden ser ejecutados en cualquier versión del sistema operativo Windows que cumpla con las especificaciones del programa generado.

A pesar de que internamente Visual Basic hace uso de los lenguajes de programación C y C++, que son lenguajes relativamente difíciles de dominar, la línea de aprendizaje es corta ya que no se está trabajando directamente con tales lenguajes sino con una capa más arriba, por lo que el desarrollador no se preocupará por el código generado en tanto no se requiera la creación de tareas más especializadas.

Los programas creados con Visual Basic pueden ser transportados de una computadora a otra con sistema operativo Windows sin ningún tipo de configuración o requerimiento extra, ya que

dichos programas son compilados en archivos ejecutables que Windows puede entender y ejecutar sin ningún problema.

A pesar de que Visual Basic no está orientado al desarrollo de aplicaciones web, con él se pueden desarrollar aplicaciones que pueden trabajar dentro de una red de computadoras y comunicarse con otros sistemas diferentes como sistemas administradores de bases de datos o servicios web.

Java y C#.

Java y C# (pronunciado en inglés see sharp) son lenguajes de programación relativamente nuevos creados para resolver problemas específicos pero que han ido evolucionando incluyendo cada vez nuevas características y librerías. Ambos requieren de una línea de aprendizaje mayor además del entendimiento de herramientas especializadas para el desarrollo de aplicaciones haciendo uso de éstas tecnologías.

Si se deseará crear aplicaciones web con estas herramientas, en el caso de Java por ejemplo, se requeriría de un servidor dedicado además de aprender características del lenguaje que no están incluidas dentro de sus estándares.

PHP y JavaScript.

PHP y JavaScript son lenguajes orientados cien por ciento a ambientes web. En el caso de PHP, para la implementación de una aplicación desarrollada con éste lenguaje requiere la instalación de un servidor de aplicaciones el cuál actúa tanto como intérprete y proveedor de tal aplicación. Además, para que la aplicación pueda funcionar en red, el servidor debe estar conectado a una red de computadoras o a internet para que los equipos cliente puedan tener acceso a la aplicación.

JavaScript es un lenguaje script que sólo puede ser ejecutado por navegadores web (browsers). JavaScript es más bien un lenguaje secundario que es usado para manipular documentos (páginas web) dentro de un navegador. Cada navegador tiene diferentes estándares en cuanto a la implementación de JavaScript, por lo que el aprendizaje de éste lenguaje requiere un vasto conocimiento de la compatibilidad de características con los diferentes navegadores que existen en el mercado.

3.7 Requerimientos Mínimos de Hardware, necesarios para la instalación y utilización del Sistema.

- Procesador Celeron o Pentium en adelante
- Tarjeta de video VGA
- 128 Mb de RAM en adelante
- 25 MB de espacio libre mínimo en disco duro solo para la instalación del sistema
- Para un mejor desempeño utilizar Windows XP en adelante.

4

Capítulo

Desarrollo e Implementación del Sistema



Capítulo 4. Desarrollo e Implementación del Sistema.

Habiendo analizado y valorado el desempeño de la asignación de horarios actual se desarrolló el prototipo del sistema generador de horarios, el cual tiene como objetivo realizar la gestión de estos en el menor tiempo posible y con la mayor exactitud, para la carrera de Ingeniería en Computación en la Universidad Americana de Acapulco, A.C.

Para ello se elaboraron las bases de datos de los maestros y materias, con la información de las tiras de cátedras UNAM actualizada que brinda la dirección de Servicios Escolares de la UAA, el plan de estudios UNAM vigente y el formato de disponibilidad de horarios, en el que se utilizó la siguiente metodología para la realización del software programado en Visual Basic y su Bases de Datos en Access.

METODOLOGÍA RUP (Rational Unified Process)

Es una metodología cuyo fin es entregar un producto de software. Se estructuran todos los procesos y se mide la eficiencia de la organización. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP (Rational Unified Process), es un conjunto de metodologías adaptables al contexto y necesidades de cada organización, describe cómo aplicar enfoques para el desarrollo del software, llevando a cabo unos pasos para su realización, además se centra en la producción y mantenimiento de modelos del sistema.

Principales características.

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)
- Pretende implementar las mejores prácticas en Ingeniería de Software
- Desarrollo iterativo
- Administración de requisitos
- Uso de arquitectura basada en componentes
- Control de cambios
- Modelado visual del software
- Verificación de la calidad del software

El RUP (Rational Unified Process), Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

En la figura siguiente se puede apreciar el esquema de la metodología RUP basada en su ciclo de vida.

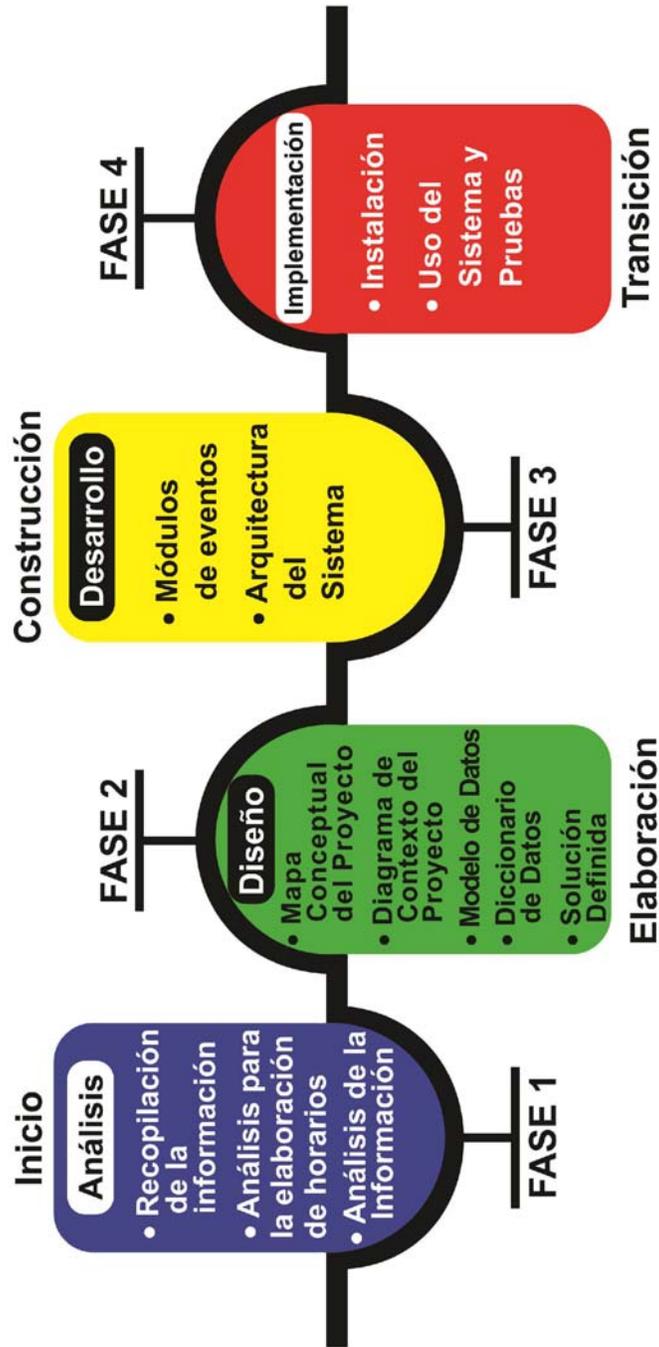


Figura 18. Esquema de la metodología RUP (Rational Unified Process), basada en su ciclo de vida.

4.1. Análisis

4.1.1. Recopilación de la información

- Plan de estudios vigente (Capítulo 1. Estado del Arte, pág.22)
- Mapa Curricular del plan de estudios vigente (Capítulo 4. Desarrollo e Implementación del Sistema, pág.113)
- Tiras de cátedras autorizadas por la UNAM actualizadas (Capítulo 2. Caso de Estudio, pág.42)
- Formato de disponibilidad de horario llenado por el profesor (Capítulo 2. Caso de Estudio, pág.38)

4.1.2. Análisis del proceso para la elaboración de los horarios

- Se define el ciclo escolar non o par
- Se estipulan las materias que se van a dar de acuerdo al plan de estudios (Normal, Optativas) en cada semestre o grupo
- Se consigna un profesor a cada materia
- Se fija un horario a cada profesor en cada materia de cada semestre
- Se destina un aula al grupo y en caso de que la materia así lo requiera se asigna aula especializada para materias optativas o adicionales (p.e. Talleres, Laboratorio)
- Se verifica si es correcta la información y se autoriza
- Se generan los horarios del ciclo escolar que corresponde
- Se envía a Excel para su manipulación e impresión

En la figura 19 se muestra el diagrama de flujo de este procedimiento.

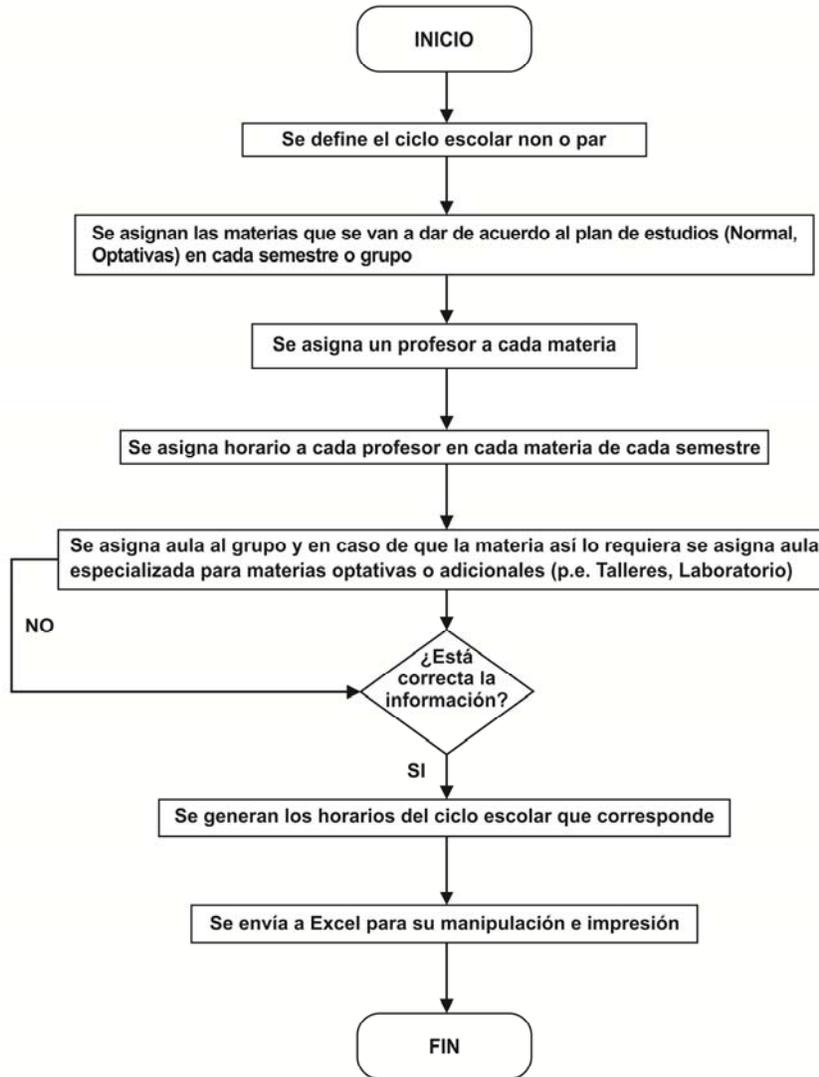


Figura 19. Diagrama de flujo del Análisis del proceso para la elaboración de los horarios.

Para elaborar los horarios de un semestre determinado se tienen en consideración los siguientes elementos:

- La tira de cátedras UNAM actualizada que brinda la dirección de Servicios Escolares de la UAA. En este documento se encuentra el listado de materias que la UNAM autoriza impartir a cada maestro, dado su perfil y experiencia profesional.
- El plan de estudios UNAM. Éste se descarga de la página <http://www.dgire.unam.mx> en el apartado de normatividad.
- El formato de disponibilidad de horarios. En este formato el profesor selecciona las materias que quiere impartir, la preferencia, la repetición y el horario que dispone para impartirla.

Se envía el formato de Disponibilidad de Materias de manera electrónica o impresa a los maestros interesados en impartir cátedra en la UAA.

Este formato es llenado con base a las indicaciones contenidas en él.

Después de ser llenado es enviado electrónicamente o entregado impreso de manera personal en la Facultad.

Se descargan las materias por maestro, ya que están organizados los horarios, se envían de manera personal a cada maestro, vía electrónicamente en el formato de Horario.

Diagrama de casos de Uso.

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen qué es lo que debe hacer el sistema, pero no cómo.

Caso de uso.

Un caso de uso describe, —desde el punto de vista de los actores—, un grupo de actividades de un sistema que produce un resultado concreto y tangible.

Los casos de uso son descriptores de las interacciones típicas entre los usuarios de un sistema y ese mismo sistema. Representan el interfaz externo del sistema y especifican qué requisitos de funcionamiento debe tener este (recuerde, únicamente el qué, nunca el cómo).

Cuando se trabaja con casos de uso, es importante tener presentes algunas sencillas reglas:

Cada caso de uso está relacionado como mínimo con un actor.

Cada caso de uso es un iniciador (es decir, un actor).

Cada caso de uso lleva a un resultado relevante (un resultado con «valor intrínseco»).

Los casos de uso pueden tener relaciones con otros casos de uso. Los tres tipos de relaciones más comunes entre casos de uso son:

<<include>> que especifica una situación en la que un caso de uso tiene lugar dentro de otro caso de uso.

<<extends>> que especifica que en ciertas situaciones, o en algún punto (llamado punto de extensión) un caso de uso será extendido por otro.

Generalización que especifica que un caso de uso hereda las características del «super» caso de uso, y puede volver a especificar algunas o todas ellas de una forma muy similar a las herencias entre clases.

Actor.

Un actor es una entidad externa (de fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otros ordenadores o eventos externos.

Los actores no representan a personas físicas o a sistemas, sino su rol. Esto significa que cuando una persona interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores. Por ejemplo, una persona que proporciona servicios de atención telefónica a clientes y realiza pedidos para los clientes estaría representada por un actor «equipo de soporte» y por otro actor «representante de ventas».

Descripción de casos de uso.

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso.

En la siguiente figura se puede observar el diagrama de casos de uso del proceso de llenado del formato de disponibilidad de horario que se envía a los profesores:



Figura 20. Diagrama de Casos de Uso.

Diagrama de secuencia:

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

Los mensajes pueden ser o bien síncronos, el tipo normal de llamada del mensaje donde se pasa el control a objeto llamado hasta que el método finalice, o asíncronos donde se devuelve el control directamente al objeto que realiza la llamada. Los mensajes síncronos tienen una caja vertical en un lateral del objeto invocante que muestra el flujo del control del programa.

En la siguiente figura se describe el diagrama de secuencia del proceso de llenado de formato de disponibilidad de horario para obtención de datos:

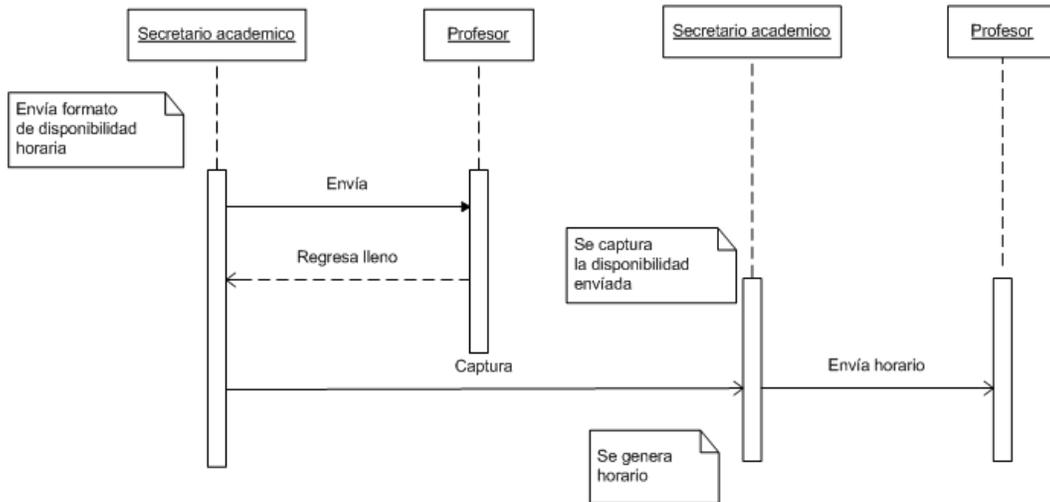


Figura 21. Diagrama de Secuencia.

Diagrama de colaboración.

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

En la siguiente figura se puede apreciar el diagrama de colaboración del proceso de llenado de formato de disponibilidad de horario de los profesores.

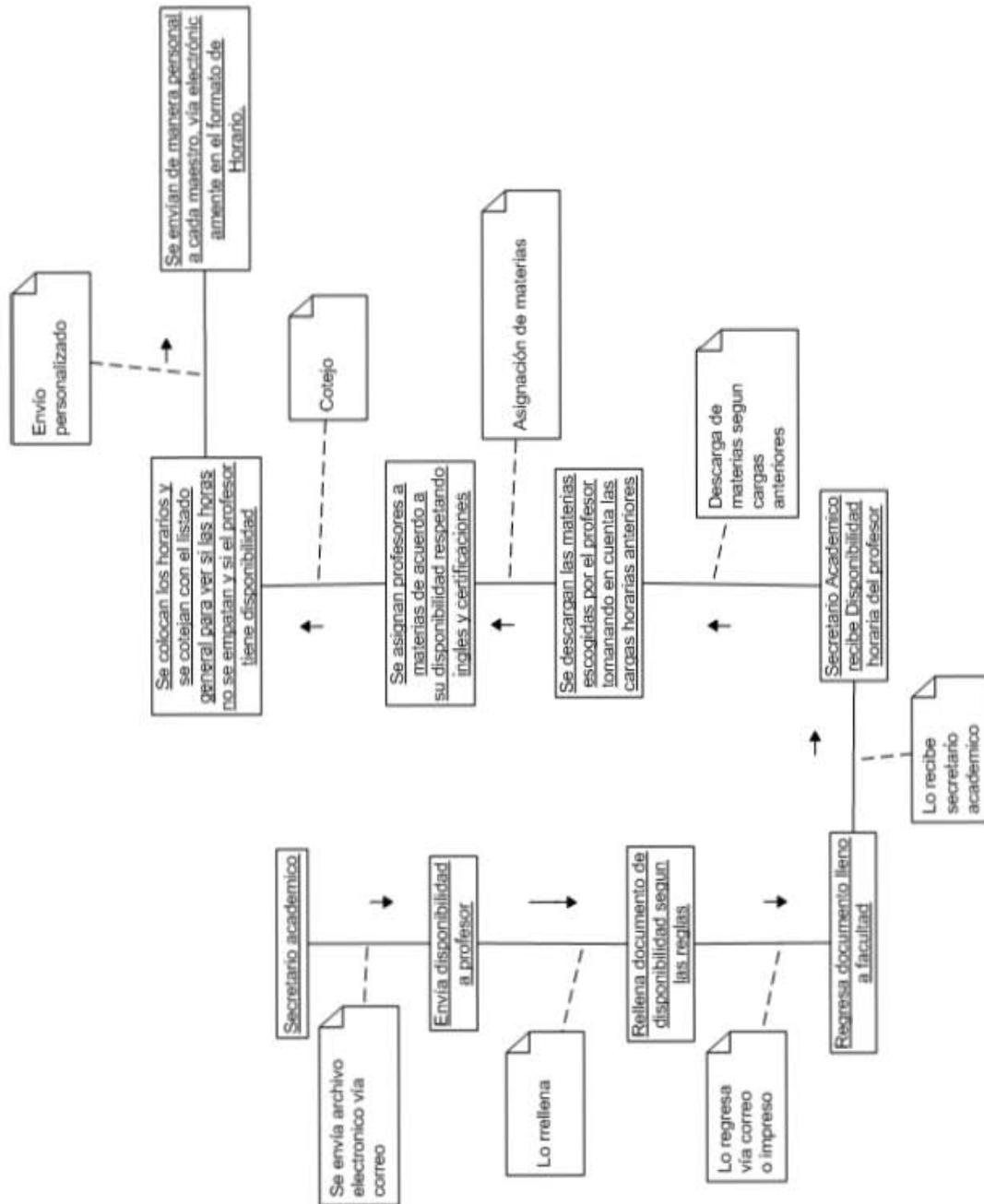


Figura 22. Diagrama de Colaboración.

Diagrama de transición de estados.

Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.

Los diagramas de estado ven a los objetos como máquinas de estado o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito. Por ejemplo, un objeto de tipo NetServer puede tener durante su vida uno de los siguientes estados:

- Listo
- Escuchando
- Trabajando
- Detenido
- Y los eventos que pueden producir que el objeto cambie de estado son
- Se crea el objeto
- El objeto recibe un mensaje de escucha
- Un cliente solicita una conexión a través de la red
- Un cliente finaliza una solicitud
- La solicitud se ejecuta y se termina
- El objeto recibe un mensaje de detención

Estado.

Los estados son los ladrillos de los diagramas de estado. Un estado pertenece a exactamente una clase y representa un resumen de los valores y atributos que puede tener la clase. Un estado UML describe el estado interno de un objeto de una clase particular.

Tenga en cuenta que no todos los cambios en los atributos de un objeto deben estar representados por estados, sino únicamente aquellos cambios que pueden afectar significativamente a la forma de funcionamiento del objeto.

Hay dos tipos especiales de estados: inicio y fin. Son especiales en el sentido de que no hay ningún evento que pueda devolver a un objeto a su estado de inicio, y de la misma forma no hay ningún evento que pueda sacar a un objeto de su estado de fin.

En las siguientes figuras se puede ver el diagrama de transición de estados de la disponibilidad horaria del profesor y el diagrama de transición de estados del horario.

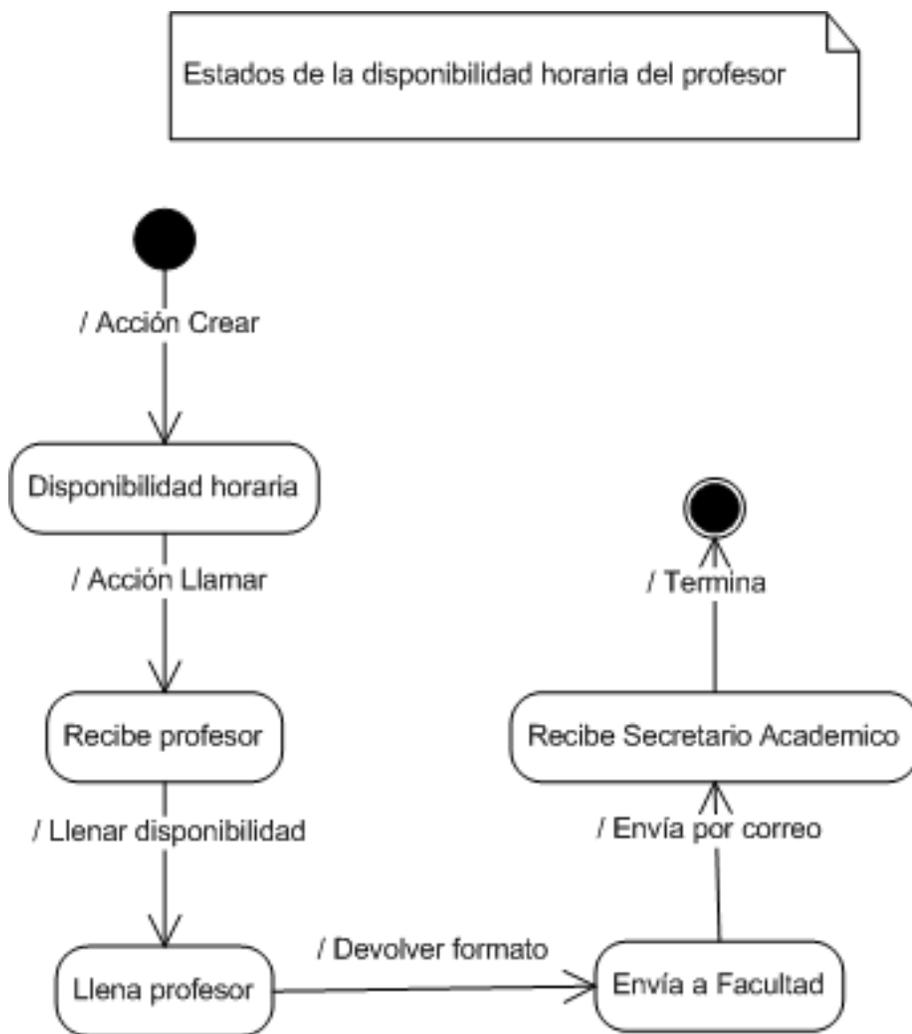


Figura 23. Diagrama de transición de estados de la disponibilidad horaria del profesor.

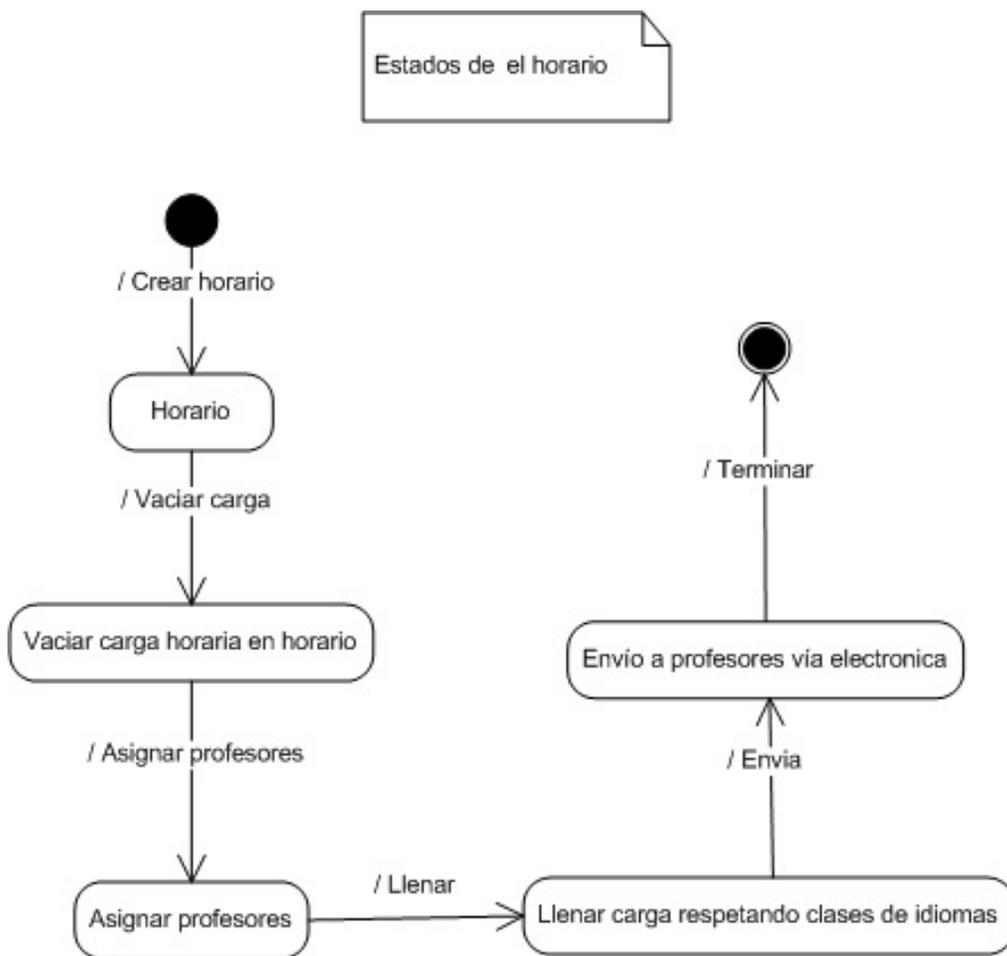


Figura 24. Diagrama de transición de estados del horario.

Diagrama de actividad.

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades.

Los diagramas de actividad son similares a los diagramas de flujo procesales, con la diferencia de que todas las actividades están claramente unidas a objetos.

Los diagramas de actividad siempre están asociados a una clase, a una operación o a un caso de uso.

Los diagramas de actividad soportan actividades tanto secuenciales como paralelas. La ejecución paralela se representa por medio de iconos de fork/espera, y en el caso de las actividades paralelas, no importa en qué orden sean invocadas (pueden ser ejecutadas simultáneamente o una detrás de otra).

Actividad.

Una actividad es un único paso de un proceso. Una actividad es un estado del sistema que actividad interna y, al menos, una transición saliente. Las actividades también pueden tener más de una transición saliente, si tienen diferentes condiciones.

Las actividades pueden formar jerarquías, lo que significa que una actividad puede estar formada de varias actividades «de detalle», en cuyo caso las transiciones entrantes y salientes deberían coincidir con las del diagrama de detalle.

Elementos de ayuda.

Existen unos pocos elementos en UML que no tiene un valor semántico real en la maqueta, pero que ayudan a clarificar partes del programa. Estos elementos son:

- Línea de texto
- Notas de texto y enlaces
- Cajas

Las líneas de texto son útiles para añadir información textual a un diagrama. Es texto es libre y no tiene ningún significado para la maqueta.

Las notas son útiles para añadir información más detallada de un objeto o una situación específica. Tienen la gran ventaja de que se pueden anclar a los elementos UML para mostrar que una nota «pertenece» a un objeto o situación específicos.

Las cajas son rectángulos repartidos libremente que pueden usarse para juntar objetos haciendo los diagramas más legibles. No tienen significado lógico en la maqueta.

En la figura 25 podemos ver el diagrama de la actividad de asignación de horarios.

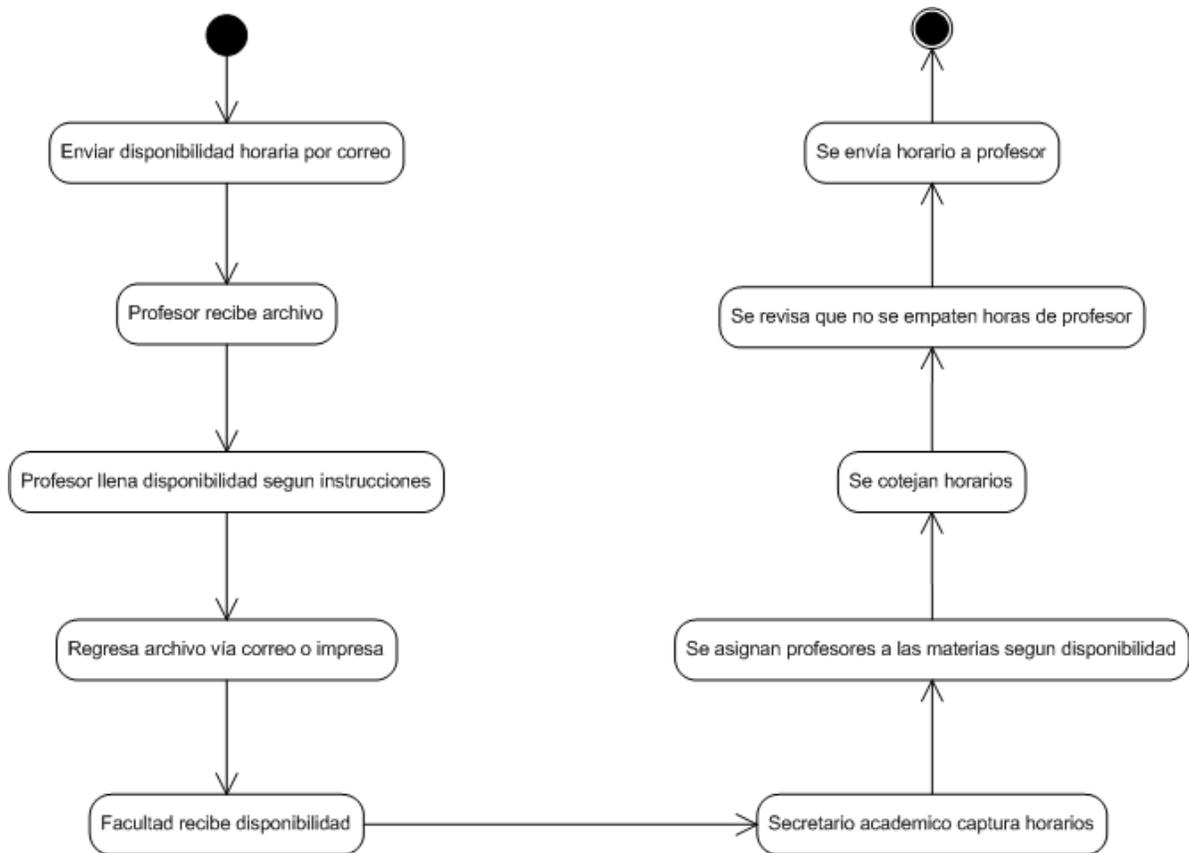


Figura 25. Diagrama de Actividad.

4.1.3. Análisis de la Información.

- **Catálogo Ciclo.**

Función: Dar a conocer el periodo de inicio y de término del semestre y si es Non o Par (1 ó 2).

Origen: Periodo anual (Conocimiento).

Campos: ID, Nombre Ciclo y ciclo.

- **Catálogo Plan de Estudios.**

Función: Dar a conocer el o los planes de estudios vigentes para el o los semestres que inician.

Origen: Autorizado por la UNAM y publicado en la página <http://www.dgire.unam.mx> para su descarga y aplicación.

Campos: ID y Nombre Plan Estudios.

- **Catálogo Grupo.**

Función: Dar a conocer su nomenclatura formada por el nombre del semestre y el turno.

Origen: Departamento de Servicios Escolares.

Campos: ID y Nombre Grupo.

- **Catálogo Aula.**

Función: Dar a conocer la ubicación exacta del salón de clases ósea el edificio en el que está ubicada el aula, el nivel de piso en el que se encuentra y el numero con el que esta ordenado el salón de clases, también sí es laboratorio y taller.

Origen: Servicios Escolares.

Campos: ID y Nombre Aula.

- **Catálogo Semestre.**

Función: Dar a conocer el nombre del semestre para el ciclo escolar vigente (Non o par) y la cantidad de semestres.

Origen: Plan de Estudios otorgado por UNAM.

Campos: ID, Nombre Semestre y Ciclo.

- **Catálogo Tipo de Cátedra.**

Función: Dar a conocer el tipo de cátedra asignado en el plan de estudios por la UNAM (Obligatoria General, Optativa).

Origen: Plan de Estudios otorgado por UNAM.

Campos: ID y Tipo de Cátedra.

- **Profesor.**

Función: Dar a conocer los datos personales, número de expediente UNAM, perfil académico, especialidad, materias que puede impartir pre-autorizadas por la UNAM.

Origen: Tiras de Cátedras.

Campos: ID, Nombre, Apellido paterno, Apellido Materno, Dirección, Ciudad, R.F.C., Género, Teléfono, Área de conocimiento, Fecha de nacimiento y Cátedras.

- **Horarios.**

Función: Asignar la o las materias que impartirá el profesor y asignar el horario a las materias.

Origen: Personal Administrativo de la Carrera.

Campos: Catedrático, Selección Catedra (Cat1, Cat2, Cat3), Materia, Horas por semana, Clave de la materia, Semestre, Tipo de materia (Obligatoria General, Optativa), Horas Profesor.

- **Mapa Curricular de la Carrera.**

Función: Muestra gráficamente las asignaturas por semestres, sus requisitos, formula crediticia, materias optativas, etc.

Origen: Plan de Estudios otorgado por UNAM.

Campos: Nombre y Clave de la Carrera, Nombre de la Facultad, Plan de Estudios, Semestres, Nombre de las materias, créditos obligatorios, créditos optativos (mínimos), total de créditos, hora teóricas, horas prácticas, materias obligatorias, optativas, materias seriadas y un formato con los módulos de salida.

En las figuras siguientes se muestran el mapa curricular de la carrera de Ingeniería en Computación y los módulos de salida de dicha carrera.

Capítulo 4. Desarrollo e Implementación del Sistema

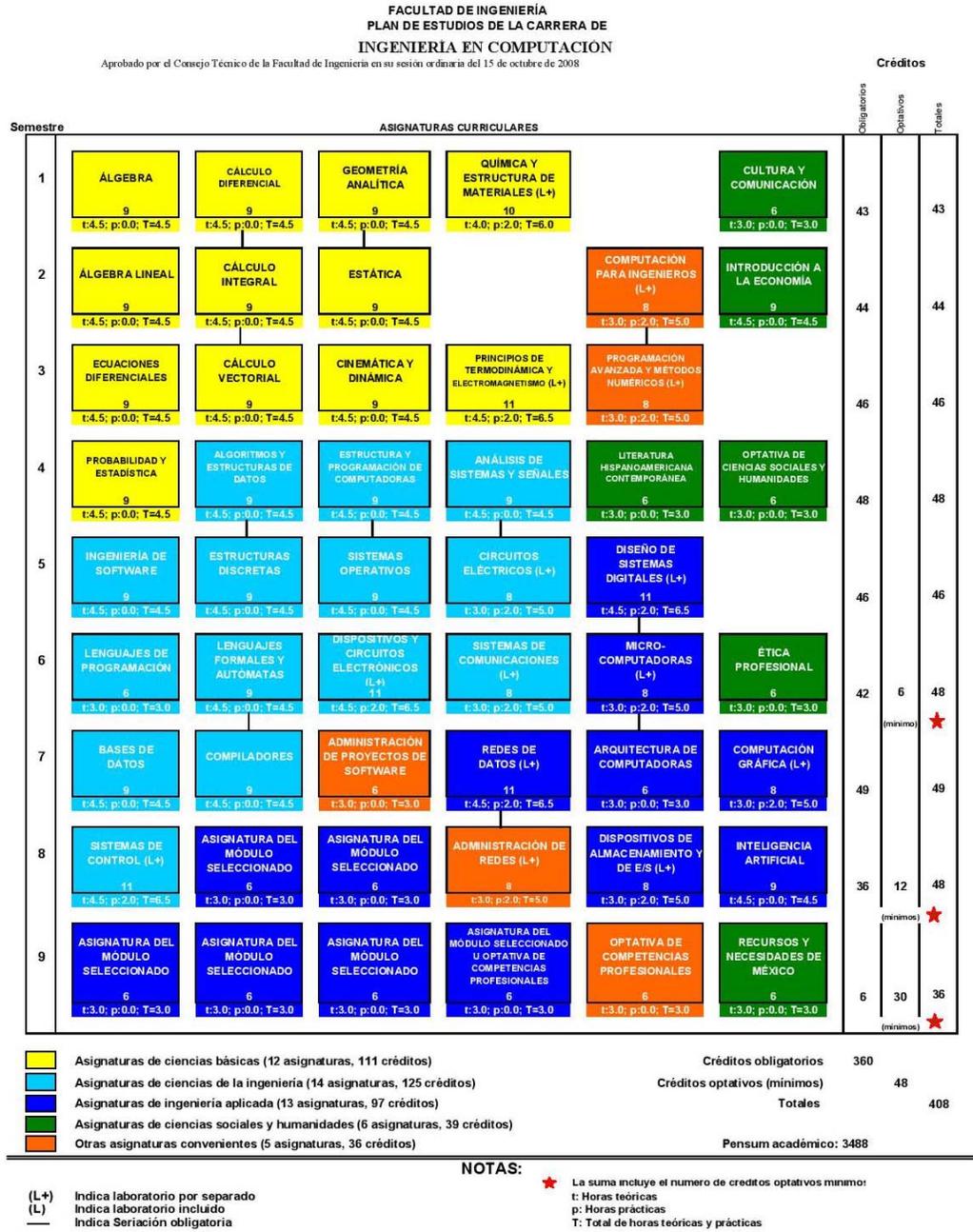


Figura 26. Mapa curricular de la Carrera Ingeniería en Computación, Plan 2009 UNAM.

Capítulo 4. Desarrollo e Implementación del Sistema

FACULTAD DE INGENIERÍA PLAN DE ESTUDIOS DE LA CARRERA DE INGENIERÍA EN COMPUTACIÓN MÓDULOS DE SALIDA

INGENIERÍA DE HARDWARE

OBLIGATORIAS
DISEÑO DE INTERFACES PARA COMPUTADORAS (6)
COMPUTO MÓVIL (6)
SISTEMAS EMBEBIDOS (6)
OPTATIVAS
ROBÓTICA (L) (10)
ROBOTS MÓVILES Y AGENTES INTELIGENTES (6)
PROCESAMIENTO DIGITAL DE SEÑALES (9)
SISTEMAS DIFUSOS (6)
INSTRUMENTACIÓN VIRTUAL (L+) (8)
CONTROL AUTOMÁTICO INDUSTRIAL (L+) (8)
FÍSICA MODERNA (L) (6)
TEMAS SELECTOS DE INGENIERÍA DE HARDWARE (6)
SEMINARIO DE TITULACIÓN (6)*
PROYECTO DE INVESTIGACIÓN (6)**

REDES Y SEGURIDAD

OBLIGATORIAS
CRIPTOGRAFÍA (6)
SEGURIDAD INFORMÁTICA I (6)
SEGURIDAD INFORMÁTICA II (6)
ARQUITECTURAS CLIENTE/SERVIDOR (6)
OPTATIVAS
DESARROLLO DE SOFTWARE SEGURO (6)
ANÁLISIS Y DISEÑO DE REDES DE DATOS (6)
REDES INALÁMBRICAS AVANZADAS (6)
TEMAS SELECTOS DE NORMALIZACIÓN (6)
COMPRESIÓN DE DATOS (6)
CODIFICACIÓN DE AUDIO Y VIDEO (6)
TEMAS SELECTOS DE REDES Y SEGURIDAD (6)
SEMINARIO DE TITULACIÓN (6)*
PROYECTO DE INVESTIGACIÓN (6)**

BASES DE DATOS

OBLIGATORIAS
BASES DE DATOS AVANZADAS (6)
BASES DE DATOS DISTRIBUIDAS (6)
BASES DE DATOS ESPACIALES (6)
OPTATIVAS
DEPOSITOS DE DATOS (6)
MINERÍA DE DATOS (6)
TEMAS SELECTOS DE BASES DE DATOS (6)
SEMINARIO DE TITULACIÓN (6)*
PROYECTO DE INVESTIGACIÓN (6)**

INGENIERÍA DE SOFTWARE

OBLIGATORIAS
NEGOCIOS ELECTRÓNICOS (6)
VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE (6)
COMPUTO DE ALTO DESEMPEÑO (6)
COMPUTO MÓVIL (6)
OPTATIVAS
SISTEMAS EN TIEMPO REAL (6)
DESARROLLO DE SOFTWARE SEGURO (6)
ARQUITECTURAS CLIENTE/SERVIDOR (6)
MINERÍA DE DATOS (6)
APRENDIZAJE (6)
DISEÑO DE INTERFACES, MULTIMEDIA Y REALIDAD VIRTUAL (6)
FÍSICA MODERNA (L) (6)
TEMAS SELECTOS DE INGENIERÍA DE SOFTWARE (6)
SEMINARIO DE TITULACIÓN (6)*
PROYECTO DE INVESTIGACIÓN (6)**

OPTATIVAS DE CIENCIAS SOCIALES Y HUMANIDADES

REDACCIÓN Y EXPOSICIÓN DE TEMAS DE INGENIERÍA (6)
TEMAS SELECTOS DE FILOSOFÍA DE LA CIENCIA Y DE LA TECNOLOGÍA: CIENCIA, TECNOLOGÍA Y SOCIEDAD (6)
ASIGNATURA SOCIOHUMANÍSTICA EN OTRAS ENTIDADES ACADÉMICAS (6)

NOTAS

* La asignatura de Seminario de titulación únicamente podrá ser seleccionada por los alumnos que elijan la opción de titulación por "Seminario de tesis o tesina"

** La asignatura de Proyecto de investigación únicamente podrá ser seleccionada por los alumnos que elijan la opción de titulación mediante "Tesis o tesina y examen profesional" o titulación por "Actividad de investigación"

SISTEMAS INTELIGENTES Y COMPUTACIÓN GRÁFICA

SISTEMAS INTELIGENTES
OBLIGATORIAS
SISTEMAS EXPERTOS (6)
ROBOTS MÓVILES Y AGENTES INTELIGENTES (6)
APRENDIZAJE (6)
RECONOCIMIENTO DE PATRONES (6)
TEMAS SELECTOS DE SISTEMAS INTELIGENTES (6)
OPTATIVAS
SEMINARIO DE TITULACIÓN (6)*
PROYECTO DE INVESTIGACIÓN (6)**

TECNOLOGÍA DEL LENGUAJE

OBLIGATORIAS
PROCESAMIENTO DEL LENGUAJE NATURAL (6)
PROCESAMIENTO DIGITAL DE VOZ (6)
PROCESAMIENTO DE CORPUS TEXTUALES Y ORALES (6)
OPTATIVAS
RECONOCIMIENTO DE PATRONES (6)
ANÁLISIS Y PROCESAMIENTO INTELIGENTE DE TEXTOS (6)
APRENDIZAJE (6)
TEMAS SELECTOS DE TECNOLOGÍAS DEL LENGUAJE (6)
SEMINARIO DE TITULACIÓN (6)*
PROYECTO DE INVESTIGACIÓN (6)**

COMPUTACIÓN GRÁFICA

OBLIGATORIAS
COMPUTACIÓN GRÁFICA AVANZADA (6)
PROCESAMIENTO DIGITAL DE IMÁGENES (6)
DISEÑO DE INTERFACES, MULTIMEDIA Y REALIDAD VIRTUAL (6)
DISEÑO ASISTIDO POR COMPUTADORA (6)
TEMAS SELECTOS DE GRAFICACIÓN (6)
OPTATIVAS
SEMINARIO DE TITULACIÓN (6)*
PROYECTO DE INVESTIGACIÓN (6)**

INGENIERÍA BIOMÉDICA

OBLIGATORIAS
CIRCUITOS INTEGRADOS ANALÓGICOS (L+) (11)
INTRODUCCIÓN A LA FISIOLÓGIA (L+) (8)
FUNDAMENTOS DE INSTRUMENTACIÓN BIOMÉDICA (L+) (8)
PROCESAMIENTO DIGITAL DE IMÁGENES MÉDICAS: Imagenología (L+) (8)
OPTATIVAS
APLICACIONES DE OPTOELECTRÓNICA EN MEDICINA (L+) (8)
AUDIOMETRÍA (6)
TELESAUD (6)
SISTEMAS Y EQUIPOS BIOMÉDICOS ELECTRÓNICOS (6)
TEMAS SELECTOS DE INGENIERÍA BIOMÉDICA (8)
TRANSDUCTORES BIOMÉDICOS (6)
INTRODUCCIÓN A LA BIOFÍSICA (6)
SEMINARIO DE TITULACIÓN (6)*
PROYECTO DE INVESTIGACIÓN (6)**

OPTATIVAS DE COMPETENCIAS PROFESIONALES

CREATIVIDAD (6)
RELACIONES LABORALES Y ORGANIZACIONALES (6)
CONTABILIDAD FINANCIERA Y COSTOS (6)
SISTEMAS DE PLANEACIÓN (8)
INTRODUCCIÓN AL ANÁLISIS ECONÓMICO EMPRESARIAL (6)
ADMINISTRACIÓN DE CENTROS DE TECNOLOGÍA DE INFORMACIÓN (8)
DESARROLLO EMPRESARIAL (6)
COSTOS Y EVALUACIÓN DE PROYECTOS (6)
CALIDAD (6)

Figura 27. Módulos de Salida de la Carrera Ingeniería en Computación, Plan2009

UNAM.

4.2. Diseño.

El diseño de sistemas es el arte de definir la arquitectura de hardware y software, componentes, módulos y datos de un sistema de cómputo, a efectos de satisfacer ciertos requerimientos. Es la etapa posterior al análisis de sistemas.

El diseño de sistemas tiene un rol más respetado y crucial en la industria de procesamiento de datos. La importancia del software multiplataforma ha incrementado la ingeniería de software a costa de los diseños de sistemas.

Los métodos de análisis y diseño orientado a objetos están siendo los métodos más ampliamente utilizados para el diseño de sistemas. El UML se ha vuelto un estándar en el Análisis y diseño orientado a objetos. Es ampliamente utilizado para el modelado de sistemas de software y se ha incrementado su uso para el diseño de sistemas que no son software así como organizaciones

4.2.1. Mapa Conceptual del Proyecto

En la siguiente figura se muestra el mapa conceptual del proyecto. La burbuja roja representa el sistema generador de horarios y las 6 burbujas verdes junto con la burbuja azul representan cada uno de los 7 módulos que lo componen, estas a su vez contienen varias

ramificaciones en las que se muestran de manera gráfica los componentes más representativos de los procesos utilizados, la información que manejan, el conocimiento que utilizan para realizar el manejo de datos que les corresponda.

El módulo Catálogo es en el que se hace el almacenamiento de datos necesarios para desplegar las listas que contienen la información requerida para el ordenamiento de los procesos de los demás módulos, las 6 burbujas amarillas que están conectadas al módulo Catálogo, contienen la información que personalizará cada horario generado como son:

- Nombre del Ciclo (formado por el año en el que inicia y el año en que termina, conformado por 2 semestres),
- Ciclo (1 ó 2 / Non o par),
- Nombre del plan de estudios (Contiene el o los planes de estudios vigentes autorizados por la UNAM),
- Grupo (define el nombre del grupo, su nomenclatura está formada por el número del semestre y el turno),
- Aula (contiene el nombre del aula y describe la ubicación exacta del salón de clases es decir el edificio en el que está ubicada el aula, el nivel de piso en el que se encuentra y el número con el

que esta ordenado el salón de clases, también sí es laboratorio y taller),

- Semestre (formado por Nombre del Semestre y Ciclo que corresponde al semestre),
- Tipo de Cátedra (Puede ser Obligatoria General, Optativa Requerida, Optativa de Humanidades, Optativa requerida (Número según el módulo requerido), Optativa de Competencias Profesionales; además de que cada módulos contienen un ID (número consecutivo asignado por el sistema automáticamente esto con la finalidad de saber cuántos datos se tienen almacenados en cada módulo).

El modulo Cátedra es enriquecido con información obtenida del Plan de Estudios vigente y se ingresan todas las cátedras que están marcadas en el dicho plan con todas sus características (Nombre, Clave, Semestre, Tipo (UNAM/UAA), Tipo (Obligatoria General/Optativa), Horas teóricas, Horas prácticas, Créditos, etc.

El módulo Profesor almacena datos personales obtenidos del mismo catedrático y académicos obtenidos de la tira de cátedras otorgadas por la UNAM, estos datos ayudan a generar su carga académica y la disponibilidad horaria para impartir las materias autorizadas que tiene cada uno, a su vez son utilizados para generar un historial sobre las cátedras que ha impartido.

El módulo Grupos selecciona el Grupo, el Semestre, el Turno y el Aula, cabe mencionar que estos datos son seleccionados de la lista que despliega cada uno, mismas que fueron capturadas y almacenadas en el módulo Catálogo y sirven para definir esos componentes dentro del Horario Final.

Módulo Cambiar Contraseña de Acceso, éste módulo realiza la acción de cambiar la contraseña de acceso al Sistema Generador de Horarios y puede ser manipulada por el usuario introduciendo la contraseña actual y posteriormente poniendo una nueva y confirmándola, dicho acto es para proteger la información concentrada en el sistema.

El módulo Salir del Sistema ejecuta la orden de cerrar el sistema por completo al momento de presionar el botón de ese nombre.

Veamos la siguiente figura en la que podemos ver a detalle el mapa conceptual de nuestro proyecto.

Mapa Conceptual del Proyecto

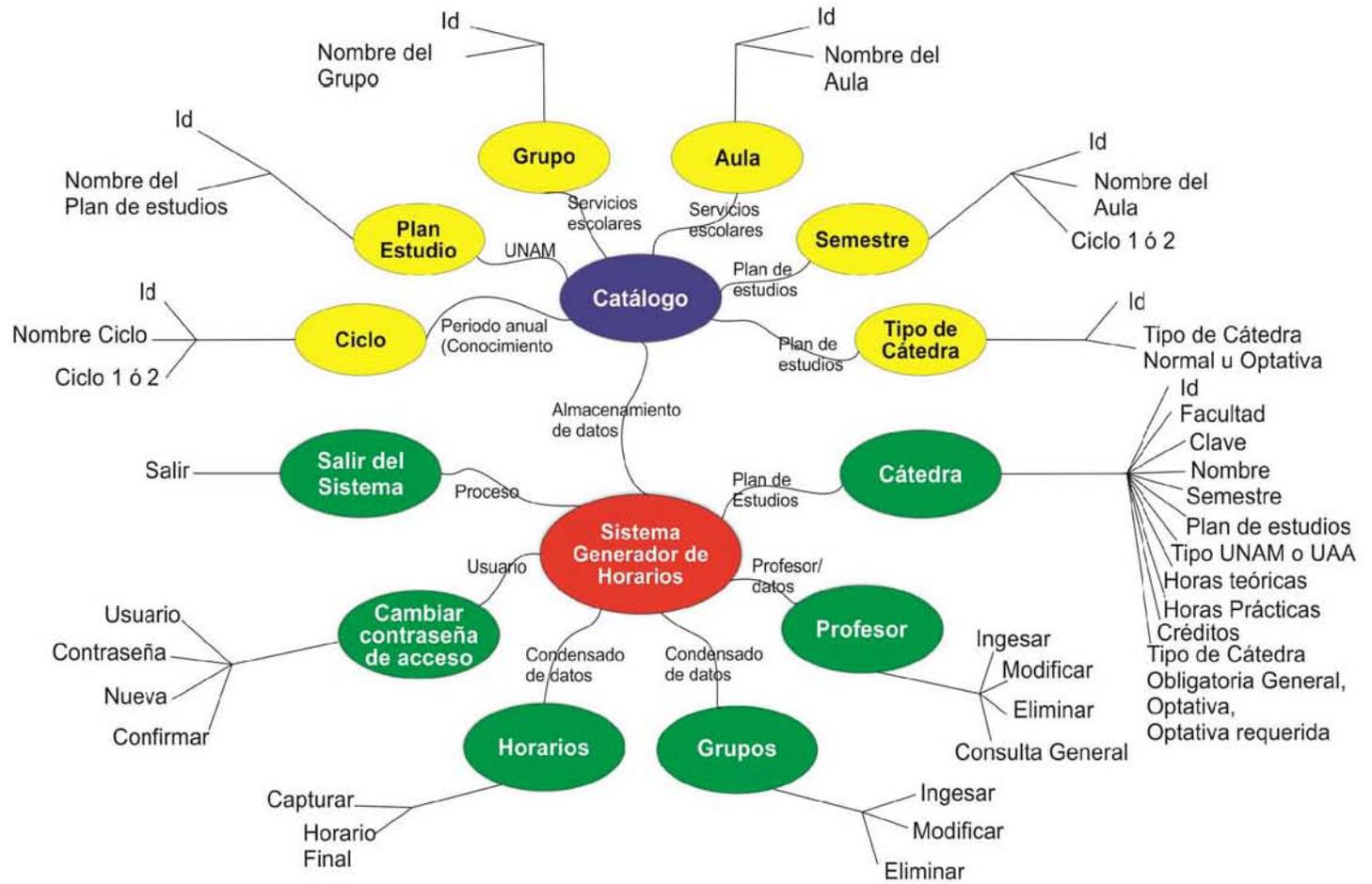


Figura 28. Mapa Conceptual del Proyecto.

4.2.2. Diagrama de Contexto del Proyecto.

El sistema esta enriquecido por una tira de cátedras otorgadas por la UNAM, en la cual se incluyen los datos personales de los profesores, nombre, carrera, número de expediente, R.F.C., plantel, plan, áreas de conocimiento, facultad, clave de materia, nombre de materia, asignaturas registradas para el profesor, etc. El personal administrativo de la carrera de Ingeniería en Computación de la Universidad Americana A.C. es el responsable de la elaboración de los horarios, por tanto éste es pieza fundamental de este proceso porque aporta sus conocimientos.

Los profesores son otro de los recursos valiosos, ya que éstos son los responsables de transmitir sus conocimientos a los alumnos, cada catedrático posee conocimientos de diversas áreas, tópicos y ramas del saber humano, el perfil del catedrático está en relación directa con el dominio que tiene de cada una de ellas y varía desde un conocimiento superficial hasta el dominio total de un área.

A los profesores y alumnos se les proporciona un horario, así los alumnos conocen su grupo, horas, aulas, asignaturas y profesor que las impartirá el semestre que van a cursar.

Como se puede apreciar en la figura siguiente.

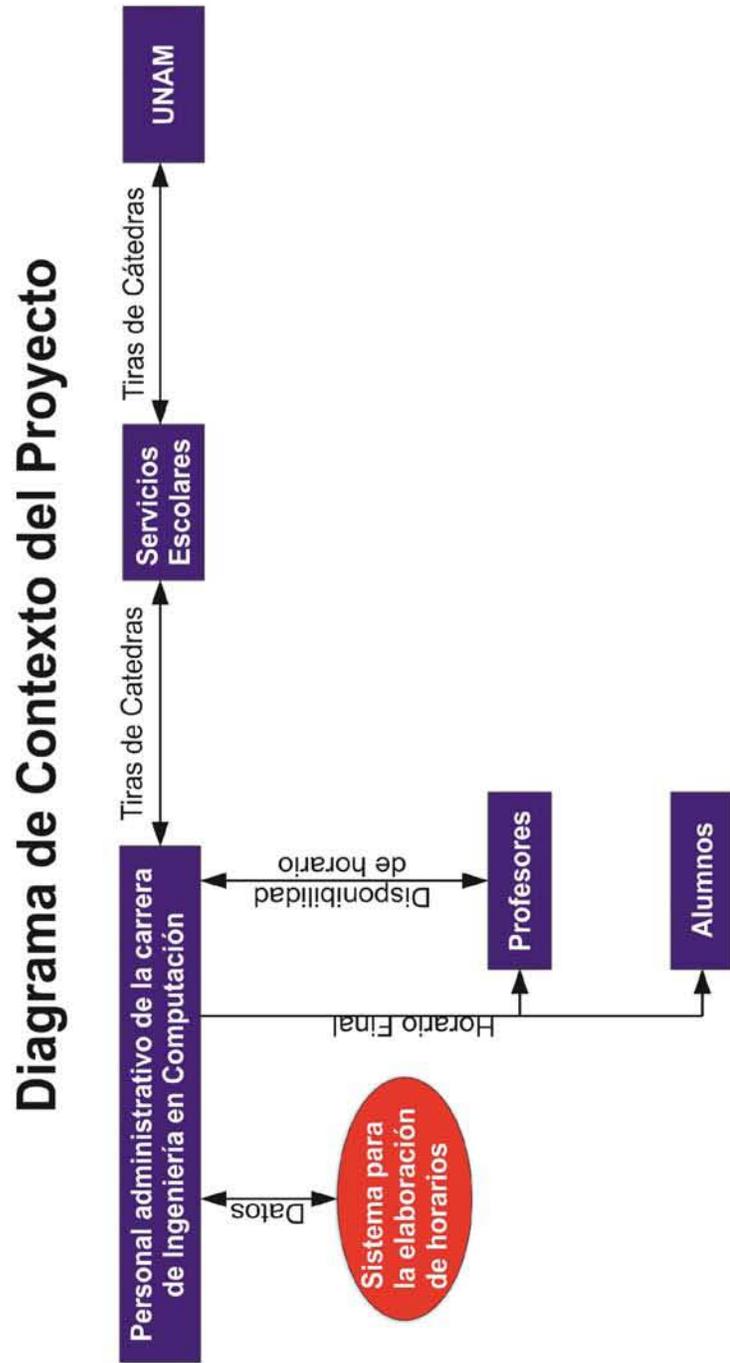


Figura 29. Diagrama de Contexto del proyecto.

4.2.3. Modelo de datos.

A continuación se muestra en la siguiente figura el modelo de datos del sistema generador de horarios para la carrera de Ingeniería en Computación de la Universidad Americana de Acapulco.

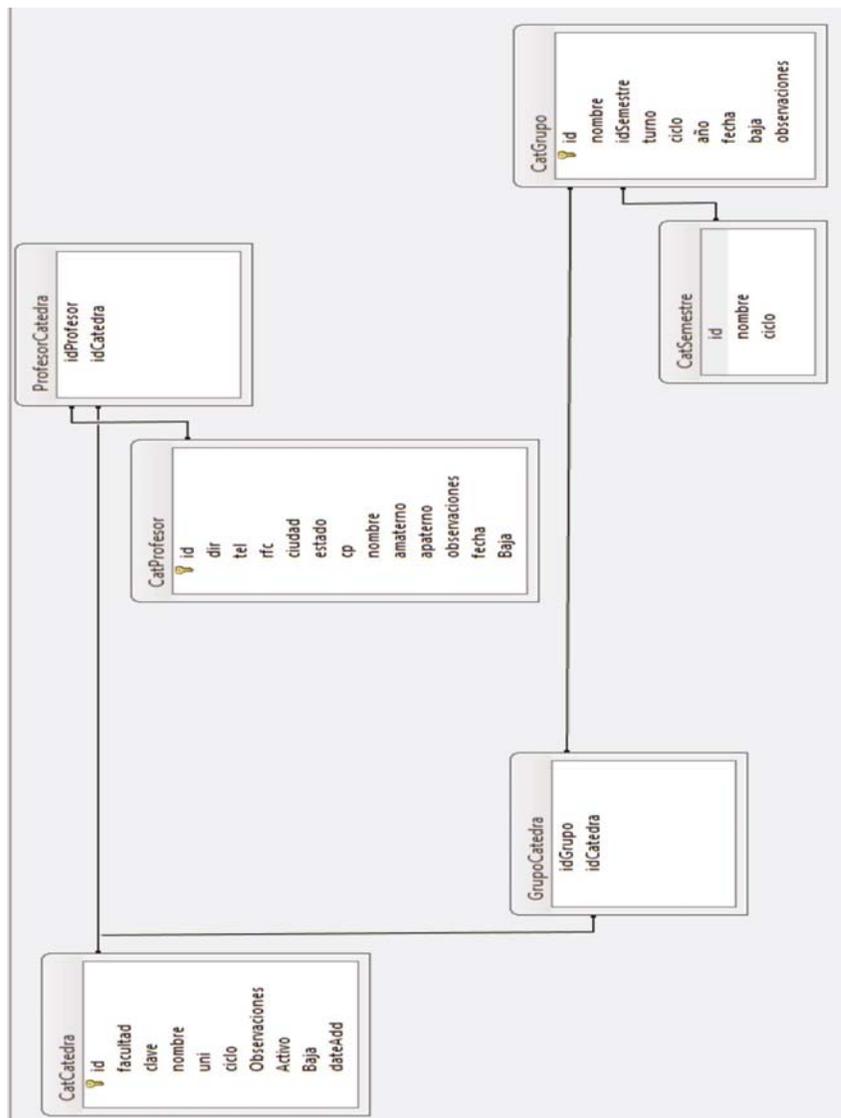


Figura 30. Modelo Entidad Relación del sistema generador de horarios.

4.2.4. Diccionario de Datos.

Un Diccionario de Datos es un listado organizado de todos los datos que pertenecen al sistema con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento en común de todas las entradas, salidas, componentes y cálculos.

Contiene las características lógicas de los datos que se van a utilizar en un sistema, incluyendo nombre, tipo de dato, restricción, longitud, descripción.

El objetivo de un diccionario de datos es dar precisión sobre los datos que se manejan en un sistema, evitando así malas interpretaciones o ambigüedades. A continuación se enlista el Diccionario de Datos del Sistema Generador de Horarios de la Carrera de Ingeniería en Computación de la Universidad Americana de Acapulco A.C.

- **Diccionario de Datos.**

CatAula					
#	Nombre Campo	Tipo Dato	Restricción	Long	Descripción
1	ID	Auto Núm	Llave primaria	255	Clave única que identifica el dato
2	Nombre	Texto/ Núm	Texto y números	20	Guarda el nombre del aula
3	addDate	Fecha/ Hora	Sólo fechas	64	Fecha de ingreso de la información
CatCatedras					
1	ID	Núm	Llave primaria	Entero Largo	Clave única que identifica el dato
2	facultad	Texto	Solo Texto	255	Campo para guardar facultad
3	clave	Texto/ Núm	Texto y números	255	Campo para guardar facultad
4	nombre	Texto	Solo Texto	100	Guarda nombre de la cátedra
5	uni	Texto	Solo Texto	100	Campo para guardar universidad
6	idCiclo	Núm	Sólo números	Simple	Vinculación cátedras con el ciclo
7	idSemestre	Núm	Sólo números	Simple	Vinculación cátedras con semestre
8	idPlan	Texto/ Núm	Texto y números	Simple	Vinculación cátedras con plan de estudios
9	hosasT	Núm	Sólo números	5	Horas Teóricas
10	horasP	Núm	Sólo números	5	Horas Practicas
11	creditos	Núm	Sólo números	20	Créditos de la materia
12	idTipoCat	Núm	Sólo números	Simple	Vinculación con el Tipo de catedra
13	observaciones	Texto	Solo texto	250	
14	activo	Si/No	Selección		Si la cátedra está activa

Capítulo 4. Desarrollo e Implementación del Sistema

15	addDate	Fecha/ Hora	Solo fechas	64	Fecha de ingreso de la información
CatCiclo					
1	ID	Auto Núm	Llave primaria	255	Clave única que identifica el dato
2	NombreCiclo	Texto/ Núm	Texto y números	20	Nombre del Ciclo
3	numCiclo	Numérico	Solo números	simple	Número de ciclo (1 ó 2)
4	addDate	Fecha/ Hora	Solo fechas	64	Fecha de ingreso de la información
CatGrupo					
1	ID	Auto Núm	Llave primaria	255	Clave única que identifica el dato
2	Nombre	Texto/ Núm	Texto y números	20	Guarda el nombre del grupo
3	addDate	Fecha/ Hora	Sólo fechas	64	Fecha de ingreso de la información
CatPlanEstudio					
1	ID	Auto Núm	Llave primaria	255	Clave única que identifica el dato
2	Nombre	Texto/ Núm	Texto y números	20	Nombre del plan de estudio vigente
3	addDate	Fecha/ Hora	Sólo fechas	64	Fecha de ingreso de la información
CatProfesor					
1	ID	Auto Núm	Llave primaria	Entero Largo	Clave única que identifica el dato
2	noExpediente	Texto/ Núm	Texto y números	20	Núm de Expediente del Profesor
3	dir	Texto	Sólo texto	100	Dirección del Profesor
4	tel	Texto	Sólo texto	20	Teléfono del Profesor
5	rfc	Texto	Sólo texto	30	R.F.C. del Profesor
6	ciudad	Texto	Sólo texto	20	Ciudad del Profesor
7	genero	Núm	Sólo números	Entero	Género del Profesor

Capítulo 4. Desarrollo e Implementación del Sistema

8	areaConocimiento	Memo	Sólo texto	64	Área de Conocimiento
9	nombre	Texto	Sólo texto	20	Nombre del Profesor
10	amaterno	Texto	Sólo texto	20	Apellido materno
11	apaterno	Texto	Sólo texto	20	Apellido paterno
12	cumpleaños	Fecha/ Hora	Sólo fechas	65	Fecha de Cumpleaños
13	observaciones	Texto/ Núm	Texto y números	100	Agrega alguna nota u observación
14	activo	Si/No	Selección		Si la cátedra está activa
15	fecha	Fecha/ Hora	Sólo fechas	64	Fecha de ingreso de la información
CatSemestre					
1	ID	Auto Núm	Llave primaria	255	Clave única que identifica el dato
2	Nombre	Texto	Solo Texto	30	Guarda el nombre del semestre
3	ciclo	Texto	Solo texto	1	Ciclo que corresponde el semestre
CatTipoCatedra					
1	ID	Auto Núm	Llave primaria	255	Clave única que identifica el dato
2	Nombre	Texto	Solo texto	64	Nombre del tipo catedra
3	addDate	Fecha/ Hora	Sólo fechas	64	Fecha de ingreso de la información
Grupo					
1	ID	Auto Núm	Llave primaria	Entero Largo	Clave única que identifica el dato
2	idGrupo	Núm	Solo números	simple	Vinculación con el grupo
3	idSemestre	Núm	Solo números	simple	Vinculación con el Semestre
4	idTurno	Núm	Solo números	simple	Vinculación con el Turno
5	idAula	Núm	Solo números	simple	Vinculación con el Aula

Capítulo 4. Desarrollo e Implementación del Sistema

6	idCiclo	Núm	Solo números	simple	Vinculación con el Ciclo
7	idPlanEstudio	Núm	Solo números	simple	Vinculación con el plan de estudios
8	año	Núm	Solo números	simple	Agrega el año
9	fecha	Fecha/ Hora	Solo fechas	64	Fecha de ingreso de la información
10	baja	Si/No	Selección		Baja
11	observaciones	Texto/ Núm	Texto y números	100	Agrega alguna nota u observación
GrupoDetalle					
1	IDGrupoDetalle	Auto Núm	Llave primaria	Entero	Clave única que identifica el dato
2	idGrupo	Núm	Solo Número	simple	Vinculación con el Grupo
3	idCatedra	Núm	Solo Número	simple	Vinculación con la catedra
4	idProfesor	Número	Solo Número	simple	Vinculación con el Profesor
GrupoDetalleHoras					
1	idGrupoDetalle	Número	Sólo números	Entero	Clave única que identifica el dato
2	idDia	Número	Sólo números	simple	Día de clase
3	Horainicial	Fecha/ Hora	Sólo Horas	simple	Hora inicial de clase
4	HoraFinal	Fecha/ Hora	Sólo Horas	simple	Hora final de clase
5	comentario	Texto/ Núm	Texto y números	50	Notas al pie de materia (taller, laboratorio, etc.)
ProfesorCatedra					
1	idProfesor	Número	Sólo números	simple	Vinculación con el Profesor
2	idCatedra	Número	Sólo números	simple	Vinculación con la Catedra

4.2.5. Solución definida

La solución del problema la podemos analizar dividiendo en 8 bloques fundamentales el sistema generador de horarios:

- Selección de Ciclo Escolar.
- Restricción en la preferencia del profesor por repetición de materia en año anterior.
- Asignación de materia a profesor.
- Restricción de horas a cubrir, que marca el plan de estudios vigente.
- Asignación de horario preferente.
- Restricción de empalme de horas.
- Asignación de Aula.
- Restricción de empalme de aula.

Debido a la naturaleza y características de este sistema generador de horarios se iniciara desde cero la recopilación de información, el análisis, el diseño y la programación de cada módulo del sistema, programado en Visual Basic y Bases de Datos en Access.

La principal ventaja de ésta opción es que el sistema está hecho a la medida de las necesidades de la carrera de Ingeniería en Computación de la Universidad Americana de Acapulco, poniendo como restricción el uso del formato que se tiene actualmente.

Interfaz de Usuario: Ésta recibe y entrega información interactuando con el usuario, es decir, el usuario puede:

- Alimentar las bases de datos y retroalimentarlas o actualizarlas con forme se necesite.
- Generar formato de disponibilidad de horario de acuerdo al ciclo escolar.
- Capturar información del formato de disponibilidad.
- Generar formato de horario final.
- Exportarlo a Excel.

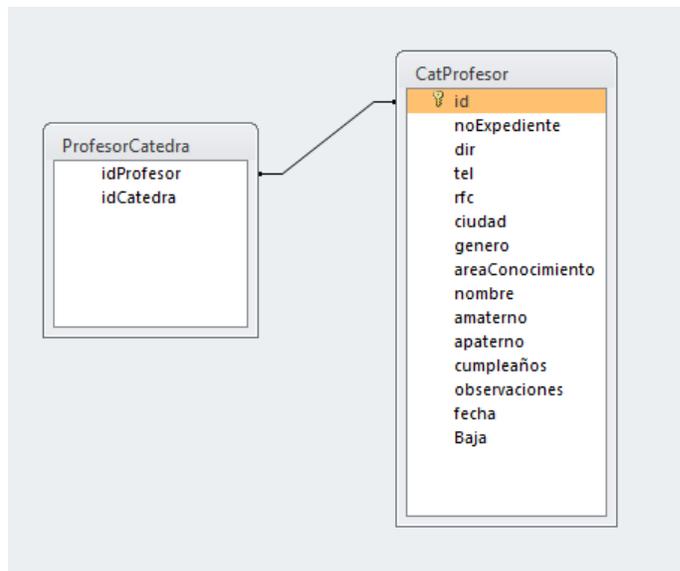
Características de la Interfaz de Usuario

- Comunicarse en el lenguaje del usuario.
- Recibir datos que introduzca al sistema el usuario y almacenarlos.
- Dar resultados por escrito
- Proporcionar al usuario la facilidad de alimentar la base de conocimientos.
- Dar al usuario la facultad de corregir las inferencias.

4.3. Desarrollo

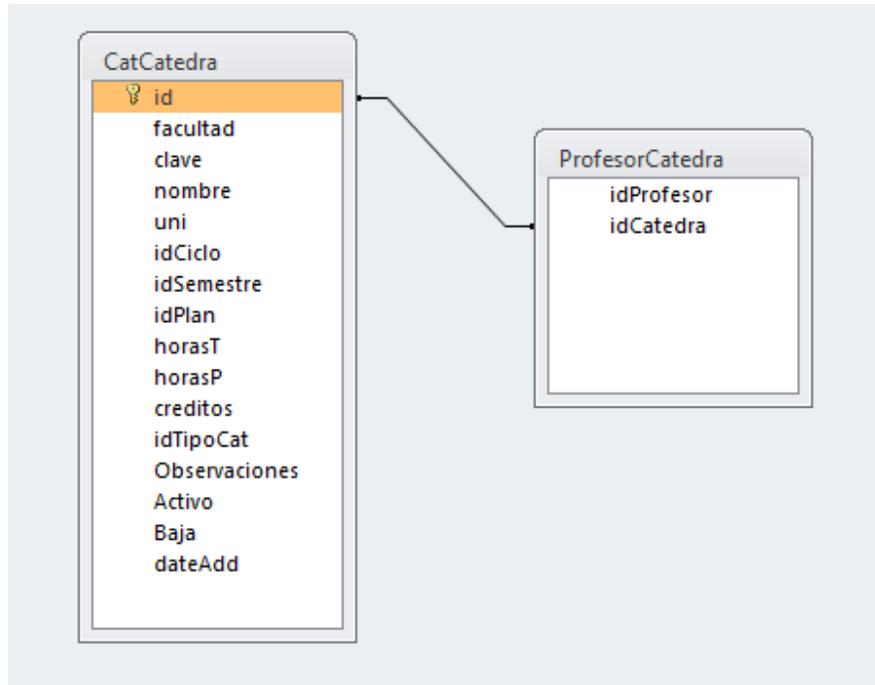
4.3.1. Módulos de eventos

ASIGNACIÓN DE CATEDRÁTICO



```
Private Sub almacenar_Click()
    guardar
End Sub
Private Sub guardar()
    Dim num As Integer
    num = 0
    For i = 1 To nombre.Rows - 1
        If IsNumeric(nombre.TextMatrix(i, 7)) Then
            inst = "UpDate GrupoDetalle set idProfesor=" & nombre.TextMatrix(i, 7) & "
Where idGrupoDetalle=" & nombre.TextMatrix(i, 8)
            db.Execute inst, num
        End If
    Next i
    If num > 0 Then
        MsgBox "Información actualiada correctamente!!"
    End If
End Sub
```

ASIGNACIÓN DE MATERIA

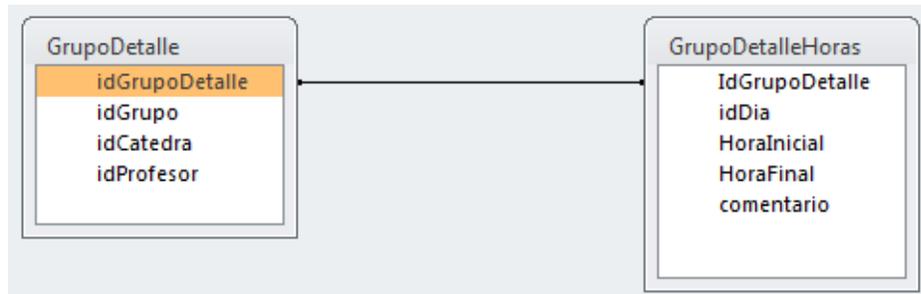


```
Private Sub almacenar_Click()
    If ComGrupo.ListIndex = -1 Then
        MsgBox "Falta seleccionar GRUPO", 0 + 48, "Error"
        ComGrupo.SetFocus
        Exit Sub
    End If
    If ComSemestre.ListIndex = -1 Then
        MsgBox "Falta seleccionar SEMESTRE", 0 + 48, "Error"
        ComSemestre.SetFocus
        Exit Sub
    End If
    If ComTurno.ListIndex = -1 Then
        MsgBox "Falta seleccionar TURNO", 0 + 48, "Error"
        ComTurno.SetFocus
        Exit Sub
    End If
    If ComAula.ListIndex = -1 Then
        MsgBox "Falta seleccionae AULA", 0 + 48, "Error"
        ComAula.SetFocus
        Exit Sub
    End If
End Sub
```

```
End If
guardar
End Sub

Private Sub guardar()
obtenerIDGrupo
Dim i As Integer, idCatedra As Integer
idCatedra = 0
Set tb = db.OpenRecordset("Grupo")
tb.AddNew
tb!ID = CDbI(TxtNoGrupo)
tb!idGrupo = ComGrupoID.List(ComGrupo.ListIndex)
tb!idSemestre = ComSemestreID.List(ComSemestre.ListIndex)
tb!idTurno = ComTurno.ListIndex + 1
tb!idCiclo = idCiclo
tb!idAula = ComAulaID.List(ComAula.ListIndex)
tb!año = Year(Now)
tb!fecha = Date
tb.Update
For i = 0 To ListBox.ListCount - 1
' If (ListBox.Selected(i)) Then
idCatedra = ComCatedrasID.List(i)
Set tb1 = db.OpenRecordset("GrupoDetalle")
tb1.AddNew
tb1!idGrupo = CDbI(TxtNoGrupo)
tb1!idCatedra = idCatedra
tb1.Update
' End If
Next i
MsgBox "La información se ha guardado correctamente", 0
tb.Close
borrar
obtenerIDGrupo
ComGrupo.SetFocus
End Sub
```

ASIGNAR HORARIOS



```

Private Sub ComGuardarCatedra_Click()
    For i = 1 To Rejilla.Rows - 1
        If IsDate(Rejilla.TextMatrix(i, 2)) Then
            If Not IsDate(Rejilla.TextMatrix(i, 3)) Then
                MsgBox "Ingrese hora salida!!", vbCritical
                Rejilla.Row = i: Rejilla.Col = 3
                mover
                Exit Sub
            End If
        End If
    Next i
    GuardarInfo
    Unload Me
End Sub
Private Sub GuardarInfo()
    inst = "Delete From GrupoDetalleHoras Where idGrupoDetalle=" &
txtIdGrupoDetalle
    db.Execute inst
    For i = 1 To Rejilla.Rows - 1
        If IsDate(Rejilla.TextMatrix(i, 2)) Then
            inst = "Insert Into
GrupoDetalleHoras(idGrupoDetalle,idDia,HoraInicial,HoraFinal,Comentario) Values(" &
-
                txtIdGrupoDetalle.Text & "," & Rejilla.TextMatrix(i, 0) & "," &
Rejilla.TextMatrix(i, 2) & "," & _
                Rejilla.TextMatrix(i, 3) & "," & Rejilla.TextMatrix(i, 4) & """)"
            db.Execute inst
        End If
    Next i
End Sub
  
```

4.3.2. Arquitectura del Sistema.

Arquitectura del Software: La Arquitectura del Software es el diseño de más alto nivel de la estructura de un sistema. Una Arquitectura de Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco.

Una arquitectura de software se selecciona y diseña con base en objetivos (requerimientos) y restricciones. Los objetivos son aquellos prefijados para el sistema de información, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de información. Las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar sistemas de información. Unas arquitecturas son más recomendables de implementar con ciertas tecnologías mientras que otras tecnologías no son aptas para determinadas arquitecturas. Por ejemplo, no es viable emplear una arquitectura de software de tres capas para implementar sistemas en tiempo real.

La arquitectura de software define, de manera abstracta, los componentes que llevan a cabo alguna tarea de computación, sus interfaces y la comunicación entre ellos. Toda arquitectura debe ser implementable en una arquitectura física, que consiste simplemente en determinar qué computadora tendrá asignada cada tarea.

La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad.

El objetivo del software desarrollado es el de apoyar a la facultad a la elaboración de los horarios de clases de la Carrera de Ingeniería en Computación de la Universidad Americana de Acapulco A.C., en base a una lista de disponibilidad de los profesores asignarles las materias que ellos deseen de acuerdo a su plantilla asignada de la UNAM, y que no se sobre pongan los horarios de las materias asignadas en otros grupos, de forma tal que sea más rápido el crear estos horarios, ahorrando tiempo en su elaboración, y evitando que se generen retrasos en su entrega a los profesores.

Arquitectura de Hardware: De primera instancia el sistema se utilizará en un solo equipo, de forma que no se requerirá ninguna instalación especial, las características de este equipo para obtener una mejor funcionabilidad del sistema se recomienda que sean:

- Procesador Celeron o Pentium en adelante.
- Tarjeta de video VGA.
- 128 Mb de RAM en adelante.
- 25 MB de espacio libre mín en disco duro solo para la instalación.
- Windows XP en adelante.

Arquitectura de Software: La plataforma utilizada para desarrollar el cliente y la interfaz de usuario del mismo es Visual Basic, una plataforma ampliamente disponible en dispositivos de escritorio y que puede manejar un gran número de características de las terminales a través de varias especificaciones opcionales.

Arquitectura de Datos: Las disponibilidades de los profesores son capturadas dentro del sistema, este se encargará de revisar que los horarios asignados a cada profesor no se empalmen con otras materias del mismo profesor y de que estas se encuentren dentro de los horarios que el profesor solicitó en la lista de disponibilidad que le envió a la facultad, en caso de estar empalmada mostrará el error para que este se corrija, y en caso de que no cumpla la disponibilidad también mandará un error mostrando la materia que incumple este horario.

Restricciones del Sistema: El sistema fue diseñado considerando las siguientes restricciones para su correcto funcionamiento:

- El horario que el profesor puede ofrecer de acuerdo a su disponibilidad.
- Las materias que el profesor puede dar de acuerdo a las que tiene asignadas o permitidas por la UNAM.

- Un profesor no puede dar dos materias a dos grupos a la misma hora.
- El sistema no genera los horarios de manera automática, los realiza el usuario y de acuerdo a la captura de disponibilidad se van ocupando los campos de tal manera que el sistema detectará los empalmes capturados, por lo tanto puede haber errores de captura.

Requerimientos de la entrada.

- Disponibilidad del profesor.
- Materias a impartir en la carrera.
- Lista de materias por profesor que tiene autorizadas por la UNAM en el semestre.
- Grupos.
- Salones.
- Horarios de clases.

Arquitectura de Menús.

En el siguiente diagrama se muestra la Arquitectura de Menús del sistema Generador de Horarios de la Carrera de Ingeniería en Computación de la Universidad Americana de Acapulco A.C., en el cual se observan los componentes que lo conforman para su funcionamiento, además la elaboración de la Arquitectura de Menús es pieza clave para diseñar, proyectar y construir el espacio principal de un sistema de Información. Ver figura 31.

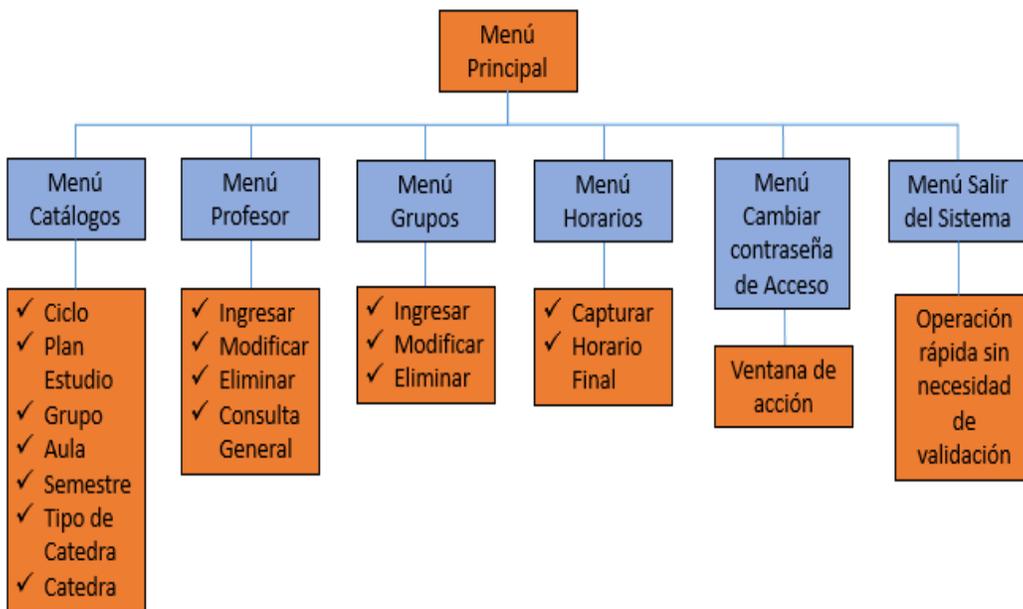


Figura 31. Diagrama de Arquitectura de Menús del Sistema Generador de Horarios.

4.4. Implementación.

4.4.1. Instalación.

Después de haber cubierto las anteriores etapas esquematizadas en la metodología que ocupamos y habiendo elaborado las bases de datos de horarios, aulas y materias con la información que nos proporcionó muy minuciosamente el personal administrativo de la carrera de Ingeniería en Computación de la Universidad Americana de Acapulco AC, procedimos con la implementación del Sistema Generador de Horarios en un ordenador que se encuentra dentro de las oficinas de la Facultad de Arquitectura e Ingenierías, siendo este prototipo para uso exclusivo para la carrera de ingeniería.

Para ello como primer paso se generan la carpeta de instalación del Sistema Generador de Horarios y guardaremos esta carpeta en un dispositivo USB o CD para llevarlo a la Facultad de Arquitectura e Ingenierías de la Universidad Americana de Acapulco AC.

Habiendo seleccionado el ordenador en el que se instalara el sistema Generador de Horarios, se inserta el dispositivo USB o el CD con la carpeta de instalación, abrimos la carpeta y buscamos el archivo setup.exe, al cual se le da doble click para que inicie la instalación.

Podremos ver que aparece una ventana flotante con el texto: “copiando archivos”, lo que indica que se copiarán un total de 8 archivos, y una vez terminada la copia, inicia la instalación del sistema.

Cuando se inicia la instalación nos aparece una pantalla azul con una ventana flotante que da la bienvenida al programa de instalación del Sistema Generador de Horarios y menciona una restricción: el programa de instalación no puede instalar los archivos del sistema o actualizar los archivos compartidos si están en uso, recomienda que se cierre cualquier aplicación que se está ejecutando, si se desea seguir con la instalación se da click en el botón aceptar, en caso contrario se oprime el botón salir para cancelar la instalación.

Si se da aceptar, el sistema de instalación hace un cambio de pantalla en donde se escoge el directorio donde se va a instalar el sistema generador de horarios.

El sistema de instalación escoge un directorio predeterminado que es el siguiente: C:\\proframfiles(x86)\\GeneradorHorarios\\

Si se desea instalar el programa en otra ubicación del ordenador, se pulsa el botón cambiar directorio y nos despliega una ventana que solicita que se especifique un directorio destino, escribiendo la ruta o seleccionado desde el menú directorios.

También especifica en que unidad de la computadora deseas instalarlo, en caso de tener un disco duro o un disco particionado. Una vez seleccionado el directorio requerido y la unidad de disco duro o partición donde se desea instalar se da aceptar.

El sistema de instalación nos regresa a la ventana anterior y se le da click en el botón que tiene el icono de un ordenador para iniciar la instalación del software del sistema generador de horarios en el directorio de destino seleccionado.

Dando click automáticamente comienza la instalación misma que podemos ver el avance de carga, al terminar la instalación aparece una ventana que dice la instalación del Sistema Generador de Horarios fue exitosa, le das aceptar y se termina la instalación.

Para comodidad del usuario, crearemos un icono de acceso directo para ingresar al sistema de una forma más sencilla, seguimos el siguiente procedimiento:

- Abrimos el explorador de Windows.
- Damos doble click en la unidad C.
- Escogemos la ruta donde fue instalado C:\\archivosdeprograma (x86)\\GeneradorHorarios
- Se abre esta carpeta generador horarios

- Dentro de esta carpeta encontraremos el archivo GeneradorHorarios.exe con el logo de la Universidad Americana de Acapulco, A.C.
- Se selecciona dicho archivo ejecutable y con botón izquierdo del mouse despliega un submenú donde podemos seleccionar la orden: crear acceso directo
- Damos clic y aparece una alerta que dice: Windows no puede crear un acceso directo aquí ¿desea colocar el icono de acceso directo en el escritorio?
- Se da click en el botón si y automáticamente Windows genera el acceso directo en el escritorio. Ver figura 32.

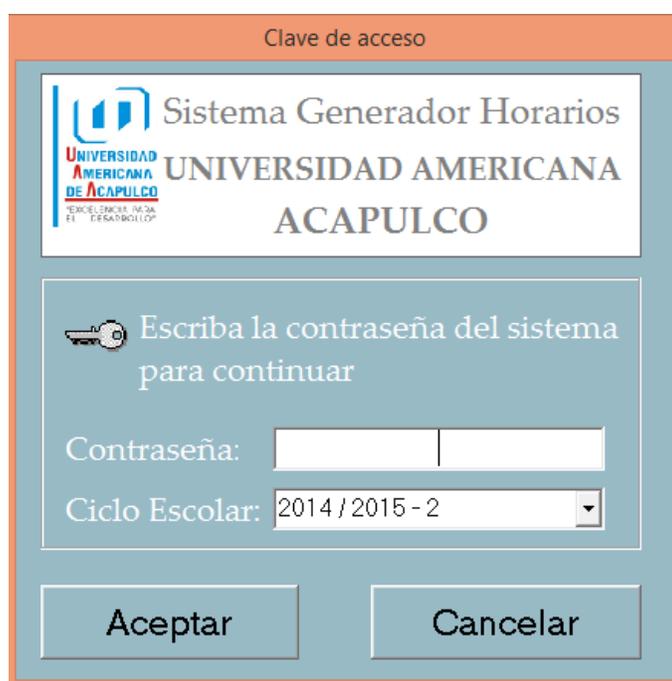


Figura 32. Icono de acceso directo para ingresar al Sistema Generador de Horarios.

4.4.2 Uso del Sistema y Pruebas.

Módulo de acceso al Sistema.

Este formulario se mostrara al inicio de la aplicación del Sistema Generador de Horarios y funcionara para validar el ciclo escolar con el que se desea acceder al sistema, siendo nuestro primer filtro de información. Ver figura 33.



Clave de acceso

 Sistema Generador Horarios
UNIVERSIDAD AMERICANA
DE ACAPULCO
EXCELENCIA EN SU
DESARROLLO

UNIVERSIDAD AMERICANA
ACAPULCO

 Escriba la contraseña del sistema
para continuar

Contraseña:

Ciclo Escolar:

Aceptar Cancelar

Figura. 33. Pantalla de acceso al sistema.

En esta ventana aparece el logo de la Universidad Americana de Acapulco, AC, un campo activado para que el usuario escriba la contraseña que autorice el acceso al sistema, y una lista desplegable donde deberá seleccionar el ciclo escolar en el que se va a trabajar la información.

Finalmente se encuentran los botones de Aceptar, que sirve para validar como correcta la información proporcionada por el usuario que desea ingresar al Sistema y el botón de cancelar que termina con el proceso y cierra el formulario de acceso al Sistema.

Modulo Principal.

El Modulo principal del Sistema está formado por varias partes entre las que encontramos, ver figura 34:

- Nombre y Logo de la Universidad Americana de Acapulco A.C.
- Nombre y logo de la Facultad de Ingeniería en Computación.
- Nombre del Sistema.
- Nombre del ciclo en el que se va a crear, editar o modificar la información.
- Barra de menús principales: Catálogos, Profesor, Grupos, Horarios, Cambiar Contraseña de Acceso y salir del sistema.
- 6 botones de acceso directo a ventanas específicas: Capturar Catedra, Capturar Profesor, Capturar Grupo, Capturar Horario, Horario Final y Salir del Sistema.

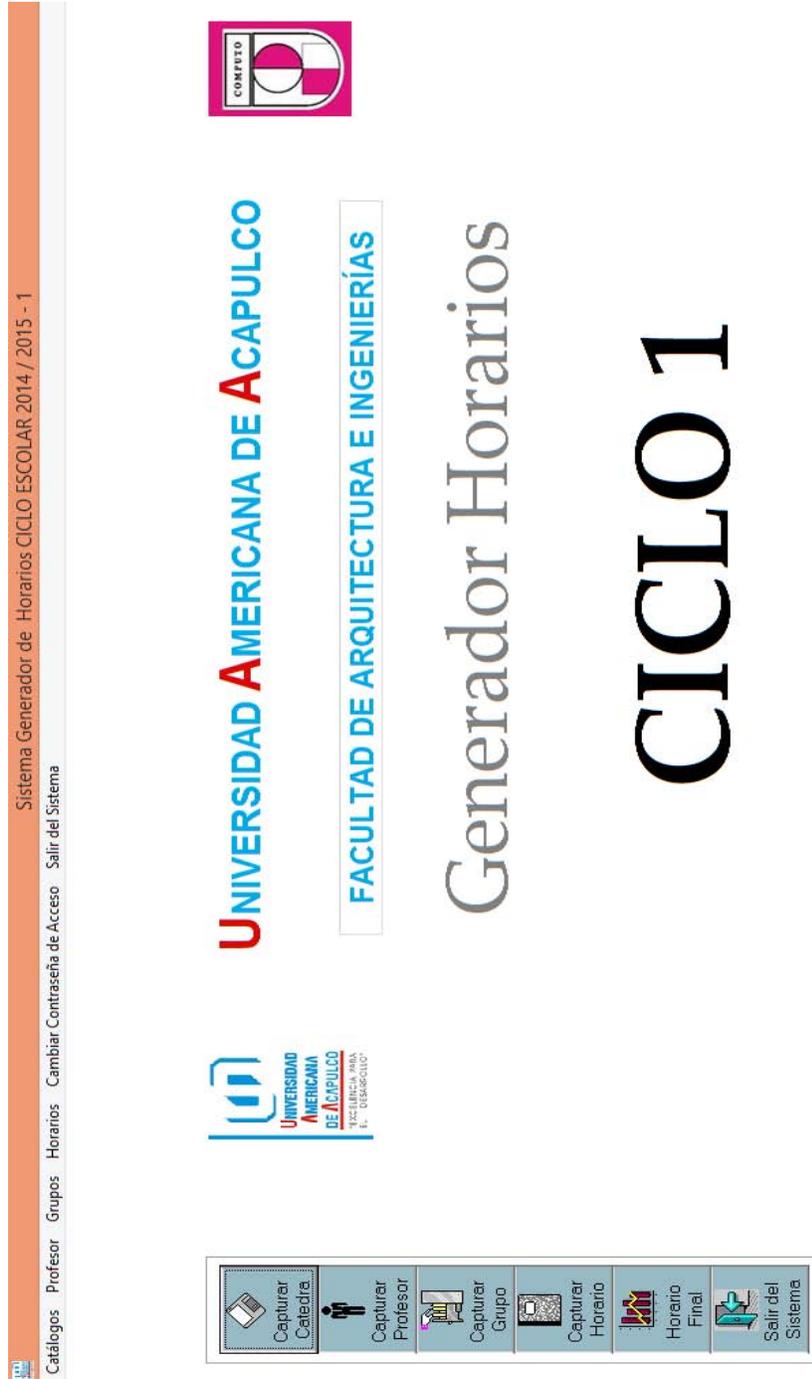


Figura 34. Pantalla principal del Sistema Generador de Horarios.

Menú Catálogos.

Recordemos que un catálogo (del latín *catalögus*, y este del griego *κατάλογος*, lista, registro) es la relación ordenada de un número generalmente extenso de elementos pertenecientes al mismo conjunto, para facilitar su localización; por ejemplo, en un archivo o una biblioteca. En nuestro Sistema el menú catálogos despliega las diferentes secciones para capturar la información y crear los registros pertinentes en nuestra base de datos, estas secciones son:

- Ciclo.
- Plan de Estudio.
- Grupo.
- Aula.
- Semestre.
- Tipo de Catedra.
- Catedra.

Catalogo Ciclos. Es una ventana emergente donde se ingresan los ciclos escolares válidos para operar el Sistema Generador de Horarios. Está conformada por la siguiente información:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro de esta tabla.

- Nombre del Ciclo. Campo donde el usuario debe capturar el nombre correcto del ciclo escolar referente al año.
- Ciclo. Campo numérico donde se registra si el ciclo escolar es NON o PAR, es decir si es 1 o 2.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Catalogo Plan de Estudios. En esta ventana se captura el o los nombres del o los Planes de Estudios que el Sistema Generador de Horarios reconocerá como válidos para operar. Está conformado por los siguientes elementos:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro de esta tabla.
- Nombre del Plan Estudio. Campo donde el usuario debe capturar el o los nombres del Plan de Estudios autorizados por la UNAM; este nombre regularmente es el año en que la DGIRE aprueba dicho plan.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Catalogo Grupo. En esta ventana se define el nombre oficial con que se distinguen los grupos dentro del Sistema Generador de

Horarios. (Para ver más información sobre elementos del nombre de grupo, ver Capítulo 2, caso de estudio.)

Esta Ventana contiene los siguientes elementos:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro de esta tabla.
- Nombre Grupo. Campo donde el usuario debe capturar el nombre correcto de los grupos que están activos en el ciclo escolar que estamos trabajando.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Catalogo Aula. En esta ventana se captura el nombre de las aulas asignadas a la carrera de Ingeniería en Computación.

Estos nombres son fijos y están compuestos por la letra inicial que indica en cual edificio está ubicado el salón, el siguiente digito nos indica el nivel (1 para primer piso, 2 para segundo, etc.), las aulas que se asignan a cada carrera son rotativas, es decir no se mantienen siempre fijas, y cada inicio de ciclo escolar la Dirección Académica les informa a las Facultades cuales son las aulas a las que pueden asignar sus distintos grupos dependiendo de sus necesidades.

Los elementos que encontramos dentro de esta ventana son:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro de esta tabla.
- Nombre Aula. Campo donde el usuario debe capturar el nombre correcto de las diferentes Aulas asignadas a la carrera de Ingeniería en Computación.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Catalogo Semestre. En esta ventana se capturan los nombres de los semestres activos y el ciclo escolar al que pertenecen.

Este catálogo está formado por los siguientes elementos:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro de esta tabla.
- Nombre Semestre. Campo donde el usuario debe capturar el nombre correcto del semestre.
- Ciclo. Campo numérico donde se registra si el ciclo escolar es NON o PAR, es decir si es 1 o 2.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Catalogo Tipo de Cátedras. En esta ventana se especifican los diferentes tipos en que se clasifican las asignaturas.

Dentro de esta Ventana nos encontramos con los siguientes elementos:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro de esta tabla.
- Tipo de Catedra. Campo donde el usuario debe capturar el nombre correcto con que se clasifica el tipo de catedra. Dentro de estas clasificaciones encontramos:
 - Obligatoria. Catedra regular dentro del plan de estudios, que se mantiene fija siempre.
 - Optativa. Asignaturas enfocadas a determinadas áreas específicas, que la Facultad asigna al semestre que así lo requiere, estas cátedras se encuentran en el plan de estudios autorizado por la UNAM. (ver Capitulo2, Caso de Estudio).
 - Adicional. Cátedras que la Universidad elige para complementar un área de conocimiento específico.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Catalogo Cátedras. En esta ventana se captura toda la información que el plan de estudios marca para cada asignatura. Esta ventana es de suma importancia para operar nuestro sistema dado que la información que aquí se capture será nuestra biblioteca principal de datos.

Los elementos que conforman esta tabla de información son:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro de esta tabla.
- Facultad. Campo donde el usuario debe capturar la clave que la UNAM asigna a la Facultad de Ingeniería en Computación.
- Clave. Se debe capturar la clave de la asignatura que el plan de estudios nos proporciona.
- Nombre. El usuario debe capturar de forma exacta el nombre de la materia.
- Semestre. Se debe especificar en qué semestre es obligatorio cursar la asignatura.
- Plan Est. En este campo se captura el plan de estudios al que corresponde la materia.
- Tipo. Campo para especificar si es materia UNAM o UAA.
- H.T. Se refiere a la cantidad de Horas Teóricas que el plan de estudios nos marca.
- H.P. Se refiere a la cantidad de Horas Practicas que el plan de estudio marca para esta asignatura.
- Créditos. Se debe capturar la cantidad de créditos que tiene asignada la materia.
- Tipo Cat. Campo para escribir el tipo de materia que es, de acuerdo a la relación vista anteriormente en el catálogo tipo de catedra.
- Botón Guardar F5. Guarda la información capturada.

- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Menú Profesor.

Nos despliega 4 submenús con las opciones que tenemos para crear, editar y consultar la base de datos de profesores del Sistema Generador de Horarios.

Menú Profesor / Ingresar. Ventana que se encuentra dividida en dos grandes partes:

- 1.- Datos del Profesor: contiene los campos donde se capturan los datos generales del profesor.
- 2.- Cátedras: en esta sección se agregan las asignaturas autorizadas por la UNAM para el profesor.

En esta ventana encontraremos 4 botones:

- Botón Add. Nos sirve para agregar las cátedras autorizadas por la UNAM al profesor.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

- Limpiar Datos F11. Borra toda la información que este capturada en las dos secciones de la ventana.

Dentro de esta ventana encontramos el botón Add, que nos lleva a una ventana emergente, en la cual encontramos los siguientes elementos del Plan de Estudios, directamente enviados del catálogo catedra capturado con anterioridad:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro.
- Facultad. Clave que da la UNAM a la Ingeniería en Computación.
- Ciclo. Campo numérico donde se registra si el ciclo escolar es NON o PAR, es decir si es 1 o 2.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Dando click en el botón ADD que nos despliega el listado de materias del plan de estudios que seleccionemos; en esta lista marcaremos las cátedras autorizadas por la UNAM para que el profesor imparta; en esta ventana encontramos los siguientes elementos:

- Plan de Estudios. Opción desplegable donde el usuario elige el plan de estudios.

- OP. Columna para seleccionar materia, al dar doble click sobre la fila de la materia se marca un símbolo (*) que nos indica que esta materia esta seleccionada.
- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro.
- Facultad. Clave que la UNAM asigna a la Ingeniería en Computación.
- Clave. Numero único asignado por la UNAM a la materia específica.
- Nombre. Titulo con el que la UNAM reconoce a la asignatura.
- Tipo. Clasificación de la materia, (UNAM o UAA).
- H.T. indica la cantidad de Horas Teóricas que la materia requiere.
- H.P. indica la cantidad de Horas Practicas que la materia requiere.
- Créditos. Campo que indica la cantidad de créditos que la UNAM asigna a dicha asignatura.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Una vez asignadas las materias al profesor, estas aparecen relacionadas en la sección cátedras de la ficha de Ingreso.

Menú Profesor / Modificar.

En esta ventana se pueden editar y/o modificar los datos generales y las cátedras asignadas a cada profesor anteriormente capturado en la base de datos del Sistema Generador de Horarios.

Esta ventana contiene los mismos elementos y secciones que la ficha de ingreso de profesor anteriormente descrita, con la sección agregada de buscar, donde contamos con un campo para ingresar el nombre o el teléfono del profesor que buscamos para editar la información sobre él.

Esta opción de modificar es muy importante en el caso de tener que aumentar cátedras a las anteriormente agregadas.

Menú Profesor / Eliminar.

En la opción Eliminar del menú Profesor podemos suprimir la ficha de información de cualquier profesor que este capturado en la base de datos del Sistema Generador de Horarios.

Su estructura es idéntica a la ficha de modificación anteriormente descrita, solo se agrega el botón Eliminar Datos F5, al cual le daremos click en caso de ser afirmativo.

El sistema despliega entonces una ventana de verificación de la acción eliminar, en la cual el usuario debe afirmar o negar esta acción.

Menú Profesor / Consulta General.

En esta ventana se nos muestra el listado de los profesores que se encuentran capturados dentro del Sistema Generador de Horarios.

Cuenta con los siguientes elementos:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro.
- Nombre. Campo que muestra nombre completo del profesor.
- Fecha de Nacimiento. Muestra la fecha de cumpleaños del profesor.
- Dirección. Indica el domicilio del profesor.
- Ciudad. Muestra la Ciudad en que radica.
- RFC. Muestra el Registro Federal de Contribuyentes del profesor.
- Teléfono. Es el número de contacto telefónico.
- Botón ordenar. Ordena alfabéticamente de la A a la Z la relación de profesores basándose en el campo de nombre.
- Botón Cerrar. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Menú Grupos.

Dentro de este menú encontramos tres submenús:

- Ingresar.
- Modificar.
- Eliminar.

Menú Grupos / Ingresar. Ventana que se encuentra dividida en dos grandes partes:

1.- Datos del Grupo: contiene los campos donde se seleccionan los datos generales del Grupo.

2.- Materias o Cátedras: en esta sección se agregan las asignaturas autorizadas por la UNAM para el grupo.

En esta ventana encontraremos los siguientes elementos:

- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro.
- Nombre. El usuario debe seleccionar el nombre del grupo del listado capturado anteriormente en el catálogo grupo.
- Semestre. Se selecciona el nombre del semestre que corresponde a ese grupo, el listado de semestres fue capturado anteriormente en el catálogo catedra.

- Turno. Se debe seleccionar el turno del grupo, Matutino o Vespertino, estas dos opciones las trae por default el sistema.
- Aula. Se selecciona el aula asignada a ese grupo, el listado de aulas fue capturado anteriormente en el catálogo aulas.
- Botón Add. Nos sirve para agregar las cátedras autorizadas por la UNAM al grupo.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.
- Limpiar Datos F11. Borra toda la información que este capturada en las dos secciones de la ventana del Menú Grupo / ingresar.

Para el uso correcto de esta ventana se debe respetar el siguiente procedimiento:

- Seleccionar Grupo.
- Seleccionar Semestre.
- Seleccionar Turno.
- Seleccionar Aula.
- Dar click en el botón ADD y asignar las materias al grupo.
- Guardar datos usando el botón adecuado.

Cuando presionamos el botón Add, nos despliega una ventana emergente con el listado de materias del plan de estudios que seleccionemos; en esta lista marcaremos las materias asignadas por la

UNAM para este grupo; dentro de esta ventana encontramos los siguientes elementos:

- Plan de Estudios. Opción desplegable donde el usuario elige el plan de estudios.
- OP. Columna para seleccionar materia, al dar doble click sobre la fila de la materia se marca un símbolo (*) que nos indica que esta materia esta seleccionada.
- ID. Numero serial consecutivo que el sistema por default asigna a cada nuevo registro.
- Facultad. Clave que la UNAM asigna a la Ingeniería en Computación.
- Clave. Numero único asignado por la UNAM a la materia específica.
- Nombre. Titulo con el que la UNAM reconoce a la asignatura.
- Tipo. Clasificación de la materia, (UNAM o UAA).
- H.T. indica la cantidad de Horas Teóricas que la materia requiere.
- H.P. indica la cantidad de Horas Practicas que la materia requiere.
- Créditos. Campo que indica la cantidad de créditos que la UNAM asigna a dicha asignatura.
- Botón Guardar F5. Guarda la información capturada.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Menú Grupos / modificar.

Esta opción nos permite editar, agregar, o eliminar las materias asignadas a un grupo o semestre en particular.

En esta ventana encontramos las mismas secciones de Menú Grupo / ingresar, con la sección agregada de búsqueda, donde podemos buscar el grupo en específico al que queremos hacerle modificaciones.

Menú Grupo / Eliminar.

En la opción Eliminar del menú Grupos, podemos borrar los grupos que deseemos con toda la información que le hayamos asignado. Su estructura es idéntica al menú Grupo / modificar, con el botón agregado de Eliminar Datos. Al dar click sobre este botón nos muestra una ventana emergente que requiere confirmar la acción de borrar, el usuario deberá aceptar o declinar para que se lleve a cabo esta acción.

Menú Horario.

Encontramos dos submenús con las opciones:

- Capturar.
- Horario Final.

Menú Horarios / Capturar.

Dentro la ventana Capturar, nos encontramos con los siguientes elementos:

- Catedrático. Es el listado de la planta docente activa durante el ciclo escolar. Esta lista se captura en la Base de datos del menú profesores/ingresar.
- Tres opciones para seleccionar materias que desee impartir en orden de preferencia (CAT1, CAT2 y CAT3), cada materia tiene al lado derecho varios campos de información que se encuentran capturados en las bases de datos del Sistema.
- CATEDRA 1. El usuario debe seleccionar de un listado desplegable de materias la preferencia 1 del profesor para impartir catedra. Este listado se captura previamente en Menú profesores/ingresar/Add. Al seleccionar la materia el campo de H.SEM se debe llenar automáticamente con la información capturada en catalogo cátedras.
- CATEDRA 2. El usuario debe seleccionar de un listado desplegable de materias la preferencia 2 del profesor para impartir catedra. Este listado se captura previamente en Menú profesores/ingresar/Add. Al seleccionar la materia el campo de H.SEM se debe llenar automáticamente con la información capturada en catalogo cátedras.
- CATEDRA 3. El usuario debe seleccionar de un listado desplegable de materias la preferencia 3 del profesor para impartir catedra. Este listado se captura previamente en Menú

profesores/ingresar/Add. Al seleccionar la materia el campo de H.SEM se debe llenar automáticamente con la información capturada en catalogo cátedras.

- Botón Guardar F5. Guarda la información capturada.
- Botón Ingresar Horas. Donde se captura la disponibilidad de horarios de cada profesor para cada catedra.
- Botón Cerrar F12. Cierra la ventana y nos regresa al Menú Principal del Sistema.

Para capturar la disponibilidad de horarios de los profesores debemos seguir el siguiente procedimiento:

- Posicionándose en el Nombre del profesor, se selecciona su catedra preferente 1.
- Una vez seleccionada la materia 1, dar click al botón Ingresar Horas.
- Nos despliega una ventana en la que encontramos un listado de días de la semana, en la que debemos ingresar Horas de Inicio y Fin de la materia en el día que tenga la disponibilidad, esta información la consultamos del formato llenado por el profesor de selección de materias y disponibilidad de horarios.
- Se cuenta con una columna de comentarios, donde se debe ingresar solo en caso de que haya cambios de aula para esa materia en ese día y hora en particular como es el caso de talleres y laboratorios.
- Dar click en el botón Guardar F5.

- Dar click en Cerrar F12.

Dentro de la ventana de captura de horario encontramos un botón llamado “Enviar a Excel” que nos permite exportar la información de carga horaria de cada profesor. Esta información es de suma importancia para la Facultad y para la Dirección de Servicios Escolares, puesto que cada profesor tiene límite de cantidad de horas permitidas a la semana por la UNAM.

Derivada de esta información en la carga horaria se pueden generar los reportes correspondientes que el personal administrativo requiere.

Menú Horarios/Horario Final.

Es una ventana con el horario de clases con toda la información capturada. Este formato cuenta con todos los elementos anteriormente descritos en el capítulo 2. Caso de Estudio, agregándole la columna créditos por petición expresa del personal administrativo que lo usa.

Dentro de esta ventana contamos con el botón Exportar a Excel, que nos permite enviar nuestro formato llenado a una hoja de Excel, para facilitar el manejo de la información ahí capturada por el personal administrativo, y generar así estadísticas y reportes que requiera.

Menú Cambiar contraseña de usuario.

Nos genera una ventana donde podemos modificar la clave de acceso para ingresar al sistema.

Menú Salir del Sistema.

Para cerrar nuestro sistema solo tenemos que seleccionar el menú salir del sistema y aceptar.

5

Capítulo



Resultados y Trabajo a Futuro

Capítulo 5. Resultados y Trabajo a Futuro.

5.1. Pruebas y Resultados.

Se pueden detectar, dentro del desarrollo de sistemas, las actividades para realizar pruebas: las producidas durante la etapa de requerimientos, de diseño preliminar y las de diseño detallado. Clasificadas de otra manera estas puede ser: funcionales, de integración y de implementación.

La finalidad de realizar Pruebas funcionales es detectar errores que pudieran haberse dado durante las fases de análisis y de especificación. Por otro lado, ellas pueden ayudar a probar si el diseño es correcto, ya que esta fase en sus dos niveles: preliminar y detallado, está basada en el análisis y especificación de requerimientos del sistema. Esta actividad es guiada por los objetivos y requerimientos plasmados. Se busca en esta prueba la posibilidad de que se haya incurrido en errores de conceptualización y/o de especificación. Las pruebas se organizan tomando como guías las diferentes opciones del sistema final.

Las Pruebas de integración se realizan cuando se procede a interconectar los diferentes subsistemas que constituyen el futuro sistema. Como es conocido, un sistema puede ser desarrollado por diferentes personas. Por esta razón, es de suma importancia probar la

interconexión de los diferentes subsistemas y así verificar que a ha sido realizada de manera correcta.

Las Pruebas de implementación o codificación son las que se realizan directamente sobre los programas a su vez constituyen subsistemas. Por esta razón, la integración de los diferentes módulos que forman un subsistema deben ser probados. Estas pruebas deben estar basadas en la especificación de diseño preliminar y detallado. (Gerez. 1985).

Los siguientes rubros, describen las pruebas más importantes que se consideraron para validar el sistema, las pruebas funcionales. Las pruebas de implementación se realizaron durante el desarrollo de la codificación de los diferentes módulos.

5.1.1 Pruebas del Sistema.

5.1.1.1 Prueba 1. Áreas de Menús.

- **Objetivo:** Guiar al usuario en el desarrollo del proceso. Esta prueba verifica que el sistema tenga como opciones principales y submenús los señalados en el Diagrama de Arquitectura (Menús).
- **Criterios de aceptación:** Cada prueba verifica que el sistema produzca los resultados óptimos de operación: que los submenús se desplieguen correctamente y se generen las ventanas correctas o se realice la acción que se solicita.

- **Procedimiento de la prueba:** A continuación, se despliegan las pantallas correspondientes a los menús principales y submenús, en las figuras 35 y 36.
- **Resultado de la prueba:** Los resultados fueron satisfactorios. En todas las opciones activadas, el sistema presento los submenús definidos en la etapa de requerimientos.

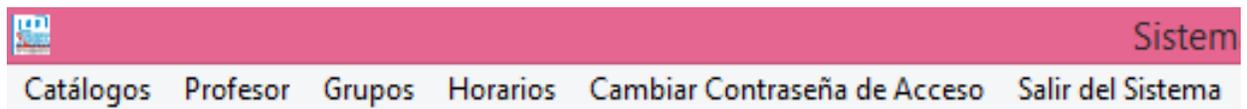


Figura 35. Barra de Menús.



Figura 36. Menús con submenús y ventana de acción.

5.1.1.2 Prueba 2. Validación de Entradas.

- **Objetivo:** Evitar pérdidas de tiempo, con entradas erróneas de datos y resultados sin validez.
- **Criterios de aceptación:** Esta prueba deberá mostrar de manera exacta la información al seleccionar la opción deseada.

- **Procedimiento de la prueba:** Se verifica que todas las cajas de texto tengan el dato que se pide y una vez dado los datos se guardan para que estos datos posteriormente sean procesados por el sistema y almacenes los datos correctos.
- **Resultado de la prueba:** Los resultados fueron satisfactorios. En todas las opciones activadas, el sistema presento los resultados correctos a cada opción. Además los desplegados de mensajes de error se presentaron de manera correcta en todas las pruebas.

En la sig. figura se muestra el registro de un profesor incompleto a propósito, el sistema de forma automática avisa que tengo un campo necesario sin llenar.

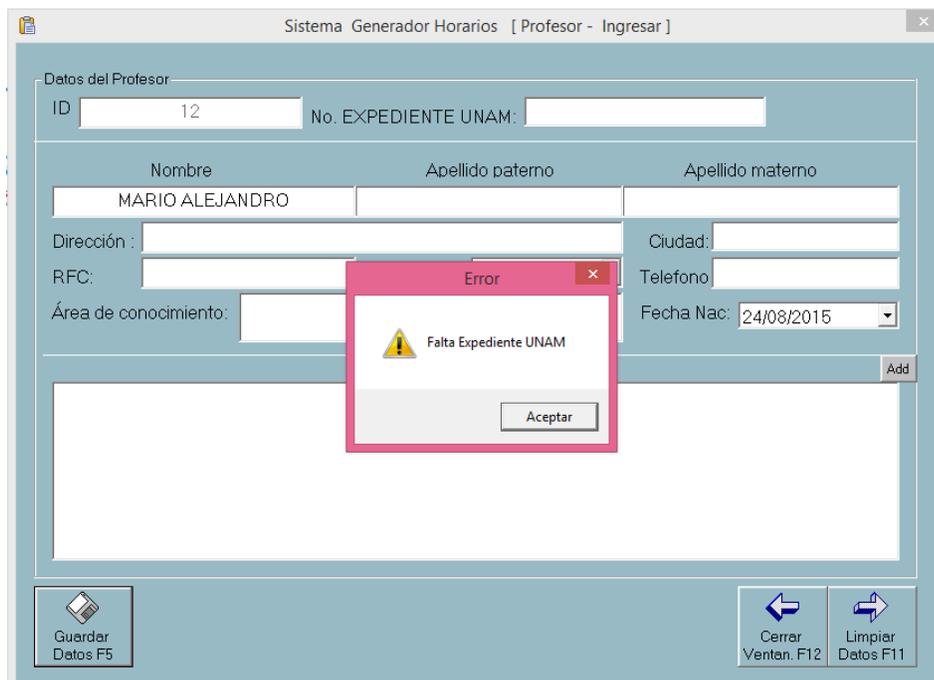


Figura 37. Ventana de captura de profesor con aviso de error por falta de información.

En la siguiente figura observamos el ingreso de un grupo, en el que a propósito dejamos de seleccionar el semestre, dimos la orden de guardar y el sistema nos avisa que nos falta la información necesaria.

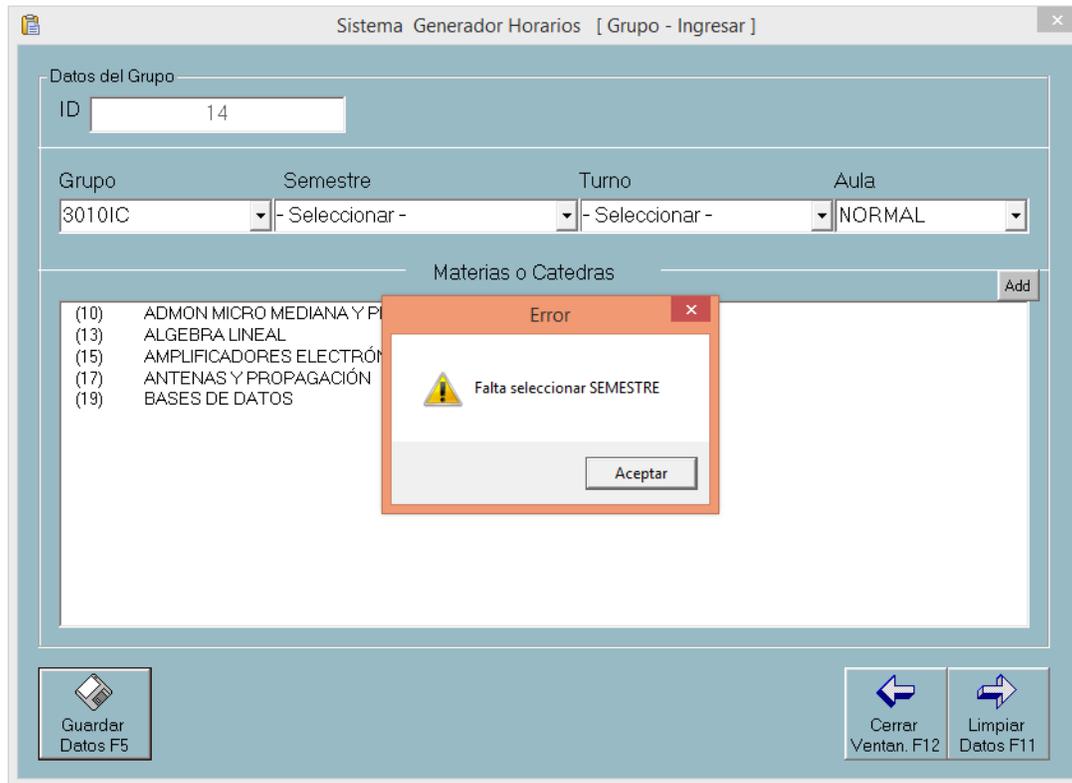


Figura 38. Ventana de captura de grupo con datos faltantes y aviso de error por parte del sistema.

5.1.1.3 Prueba 3: Despliegado de Resultados.

Menú Horarios / Submenú Captura.

- **Objetivo:** Capturar la preferencia de cátedras de cada profesor, verificando dos acciones del sistema: 1) que el listado de cátedras que se despliega sea solo el capturado en el módulo profesor / Ingresar y 2) una vez seleccionada la catedra, verificar que los datos faltantes se llenen de forma automática con la información capturada en el catálogo cátedras.
- **Criterios de aceptación:** Esta prueba deberá mostrar la información correcta en la ventana de captura y después debe generar el formato de carga horaria exportado a Excel.
- **Procedimiento de la prueba:** En el momento que el usuario selecciona la catedra de la lista de cátedras autorizadas, la información restante de esta catedra y profesor se muestra en sus respectivos campos; así mismo al dar click en el botón Exportar a Excel, el sistema generara el formato de carga horaria.
- **Resultado de la prueba:** Los resultados fueron satisfactorios. En todas las opciones activadas, el sistema presento la lista de cátedras autorizadas solamente y al exportar a Excel el sistema genero el formato solicitado con la información adecuada y correcta.

En la siguiente figura se observa la captura que se realizó en el menú Profesor, donde se agregaron cátedras autorizadas al profesor.

The screenshot shows a software window titled "Sistema Generador Horarios [Modificar]". It contains a form for entering a professor's details and a list of authorized courses.

Datos del Profesor:

- ID: 12
- No. EXPEDIENTE UNAM: 40032567
- Nombre: MARIBEL
- Apellido paterno: GREGORIO
- Apellido materno: SANCHEZ
- Dirección: FDO ROSAS 11 COL HOGAS MODERNO
- Ciudad: ACAPULCO
- RFC: AGM810214
- Género: FEMENINO
- Telefono: 4832591
- Área de conocimiento: SISTEMAS DE INFORMACION, LIC
- Fecha Nac: 14/02/1981

Catedras:

NUM	NOMBRE	TEL
2	JOSE MARIO MARTINEZ	4335678
9	Maribel Sanche Greg	213456789
12	MARIBEL SANCHEZ GRI	4832591

Buttons: Guardar Datos F5, Cerrar Ventan. F12, Limpiar Datos F11

Figura 39 Ventana de la asignación de cátedras autorizadas por la UNAM a un profesor.

En la siguiente figura se muestra la lista de las asignaturas autorizadas por la UNAM al mismo profesor en el momento de la selección de materias, para después capturar horario.

The screenshot shows a software window titled 'Sistema Generador Horarios [Horarios]'. It contains a table with columns: CATEDRATICO, MATERIA, H. SEM, CLAVE, SEMESTRE, TIPO, and HRS PROF. A dropdown menu is open over the 'MATERIA' column for the first professor, showing a list of subjects with checkboxes. The table lists several professors and their assigned subjects, hours, and semesters.

CATEDRATICO	MATERIA	H. SEM	CLAVE	SEMESTRE	TIPO	HRS PROF	
SANCHEZ GREGORIO MARIBEL, L.S.I.	CAT1		6.5	1100	9	OBLIGATORIA	11
	CAT2	<input checked="" type="checkbox"/> ADMON MIVRO MED Y PEQUEÑA EMPRESA				OBLIGATORIA	
	CAT3	<input checked="" type="checkbox"/> ADMON Y REG DE LAS TELECOMUNICACIONES					
CHAVEZ AGUILERA, NOE M.C.	CAT1	<input checked="" type="checkbox"/> ALGEBRA LINEAL				OBLIGATORIA	7
	CAT2	<input checked="" type="checkbox"/> ALGORTIMOS Y ESTRUCTURAS DE DATOS					
	CAT3	<input checked="" type="checkbox"/> ESTRUCTURA Y PROGRAMACION DE COMPUTADORAS					
CUEVAS VALENCIA, RENE EDMUNDO M.	CAT1	CINEMÁTICA Y DINÁMICA	4.5	0066	9	OBLIGATORIA	11
	CAT2	ECUACIONES DIFERENCIALES	6.5	1306	9	OBLIGATORIA	
	CAT3						
DAVILA ZURITA, NARCES DR.	CAT1	PRINCIPIOS DE TERMODINÁMICA	6.5	1314	11	OBLIGATORIA	17.5
	CAT2	DISEÑO DE SISTEMAS DIGITALES	7	1551	11	OBLIGATORIA	
	CAT3	ESTRUCTURAS DISCRETAS	4	1552	9	OBLIGATORIA	
DOMINGUEZ GONZALEZ, SANTIAGO M.C.	CAT1	INGENIERÍA DE SOFTWARE	6	1553	9	OBLIGATORIA	6
	CAT2						
	CAT3						
FLORES GODINEZ, RENE ING.	CAT1	COMPUTACIÓN GRÁFICA	4.5	1669	8	OBLIGATORIA	18
	CAT2	ARQUITECTURA DE COMPUTADORES	6.5	1762	6	OBLIGATORIA	
	CAT3	BASES DE DATOS	7	1763	9	OBLIGATORIA	
GONZALEZ MAXINES, DAVID JAIME DR.	CAT1	COMPILADORES	4	1764	9	OBLIGATORIA	4
	CAT2						
	CAT3						

Figura 40. Selección de la preferencia del profesor basándose en el listado de cátedras autorizadas para ese profesor específico.

Quando Exportamos a Excel la carga horaria, el sistema nos genera un formato específico en el que se conjuntan la selección de materias, y el horario asignado que se capturo, tal como lo muestra la siguiente imagen.

	A	B	C	D	E	F	G	H	I	J	K
	UNIVERSIDAD AMERICANA DE ACAPULCO										
											
	FACULTAD DE ARQUITECTURA E INGENIERIAS CARGA HORARIA POR MAESTRO - CARRERA : INGENIERIA EN COMPUTACIÓN SEMESTRE ENERO - JUNIO 2013										
	COMPUTACIÓN										
	PROFESOR										
3	CHAVEZ AGUILERA, NOE M.C.	LUN	MAR	MIE	JUE	VIE	SAB				SEMESTRE
	CÁLCULO INTEGRAL			9:30 - 11			8 - 11:00				4.5
	PROBABILIDAD Y ESTADÍSTICA	14:00 - 16		7 - 9:30							4.5
											SUMA
											9
6	DAVILA ZURITA, MARCES DR.	LUN	MAR	MIE	JUE	VIE	SAB				SEMESTRE
	ANÁLISIS DE SISTEMAS Y SEÑALES	16:30 - 19	16:30 - 18:30								HRS
											4.5
											SUMA
											4.5
7	HERRERA GARCÍA, GIBRAN ING.	LUN	MAR	MIE	JUE	VIE	SAB				SEMESTRE
	ALGEBRA LINEAL					11 - 13	11 - 13:30				HRS
											4.5
											SUMA
											3
	TALLER DE ELECTRONICA I - UAA			LAB. DE REDES 15 - 18:00							4to
8	GONZALEZ MAXINEZ, DAVID JAIME DR.	LUN	MAR	MIE	JUE	VIE	SAB				SEMESTRE
											HRS
											SUMA
											7.5

Figura 41. Formato de carga horaria que genera el sistema.

5.1.1.4 Prueba 4: Generar Horarios.

- **Objetivo:** Verificar que el formato que el sistema genera este completo con toda la información que nos solicita el personal administrativo de la carrera de ingeniería de la Universidad Americana de Acapulco AC, y que la columna de créditos se haya agregado satisfactoriamente.
- **Criterios de aceptación:** Esta prueba deberá mostrar la información correcta en la ventana de captura y después debe generar el formato exportado a Excel.
- **Procedimiento de la prueba:** Se selecciona el submenú generar horario dentro del menú horarios, revisamos la ventana que nos envía el sistema y si es correcta la información se exporta a Excel, donde verificamos que se agregó la columna créditos.
- **Resultado de la prueba:** Los resultados fueron satisfactorios. El sistema genero el horario correspondiente con la columna de créditos y presenta de forma correcta la información que ingresamos. Este formato se envía a Excel y se respeta el formato como se observa en la figura siguiente.

Sistema Generador Horarios [Horario Final]



UNIVERSIDAD AMERICANA DE ACAPULCO

FACULTAD DE ARQUITECTURA E INGENIERÍA

INGENIERÍA EN COMPUTACIÓN

HORARIO DEL SEGUNDO SEMESTRE

GRUPO 2010 IC



AULA:	A-304
-------	-------

PLAN DE ESTUDIOS 2009				CICLO ESCOLAR 2014/2015-2							
				HORARIO							
	ASIGNATURA	HRS. T	HRS. P	CRED	PROFESOR (ES)	LUNES	MARTES	MIE	JUEVES	VIE	SABADO
	INGLÉS	5	0	12	CENTRO DE LENGUAS	13 A 14	13 A 14	13 A 14	13 A 14	13 A 14	
0062	ALGEBRA LINEAL	4.5	0	8	ING. GIBRAN HERRERA GARCIA					11 - 13	11 - 13:30
0065	ESTÁTICA	4.5	0	6	ING. AGUSTIN VIVAS GARCIA		10:30 - 13		11:00 - 13		
1112	COMPUTACIÓN PARA INGENIEROS	3	2	6	ING. EDUARDO PERALTA MARTÍÑON	7:30 - 10:00		LAB. DE REDES 7 - 9:30			



Enviar Excel



Cerrar Ventan. F12

Figura 42. Ventana con la vista preliminar del horario de clases.

5.2. Trabajo a futuro.

Finalmente para éste trabajo de desarrollo, “Sistema Generador de Horarios para la Carrera de Ingeniería en Computación de la Universidad Americana de Acapulco A.C.” se sugieren las siguientes recomendaciones de trabajo a futuro:

Ampliar el proyecto, personalizándolo esto es, que en los registros de datos de cada profesor se incluya su fotografía y también su correo electrónico, éste último con la finalidad de enviar primero su formato de disponibilidad y en seguida su carga académica a través de internet, así se logrará hacer más rápido el flujo de información, sobre todo con profesores foráneos.

Mejorar éste Sistema Generador de Horarios para poder como prueba piloto en las tres carreras de la Facultad de Arquitectura e Ingenierías (Arquitectura, Ingeniería en Computación e Ingeniería en Telecomunicaciones), separando en tres módulos desde el arranque del Sistema para una mejor organización y así tener separadas las Cátedras y Materias de cada Carrera.

Otra posible mejora puede ser, elaborar un prototipo para todas las carreras de todas las Facultades de las Universidad Americana de Acapulco A.C.

Podría también desarrollarse el Sistema Generador de Horarios para web, esto centralizaría esa información de todas las facultades, además se publicarían los horarios en la página de la Universidad, la carga horaria de los profesores y de ser posible las calificaciones del alumnado publicando esta información en la página de la Universidad para su consulta en línea.

Conclusiones



Conclusiones.

Desde los inicios de la informática se han estado utilizando distintas formas de sistematizar los procesos para reducir tiempo y esfuerzo en los ambientes laborales de Empresas e Instituciones, mejorando con ello la organización y el desempeño de los trabajadores.

En la Carrera de Ingeniería en Computación de la Universidad Americana de Acapulco A.C., la generación de horarios se ha hecho hasta la fecha de forma manual, al implementarse éste sistema se generan de manera automatizada, además se reducen tiempos, se forma una concentrado con los datos personales y académicos de cada uno de los profesores, y de ésta manera se obtiene un historial de trayectoria digital por las materias que han impartido cada uno maestros, logrando con esto facilitar el proceso.

Al reducir el tiempo en la generación de los horarios el personal administrativo de dicha carrera podrá realiza más actividades y ser más productivo para la institución.

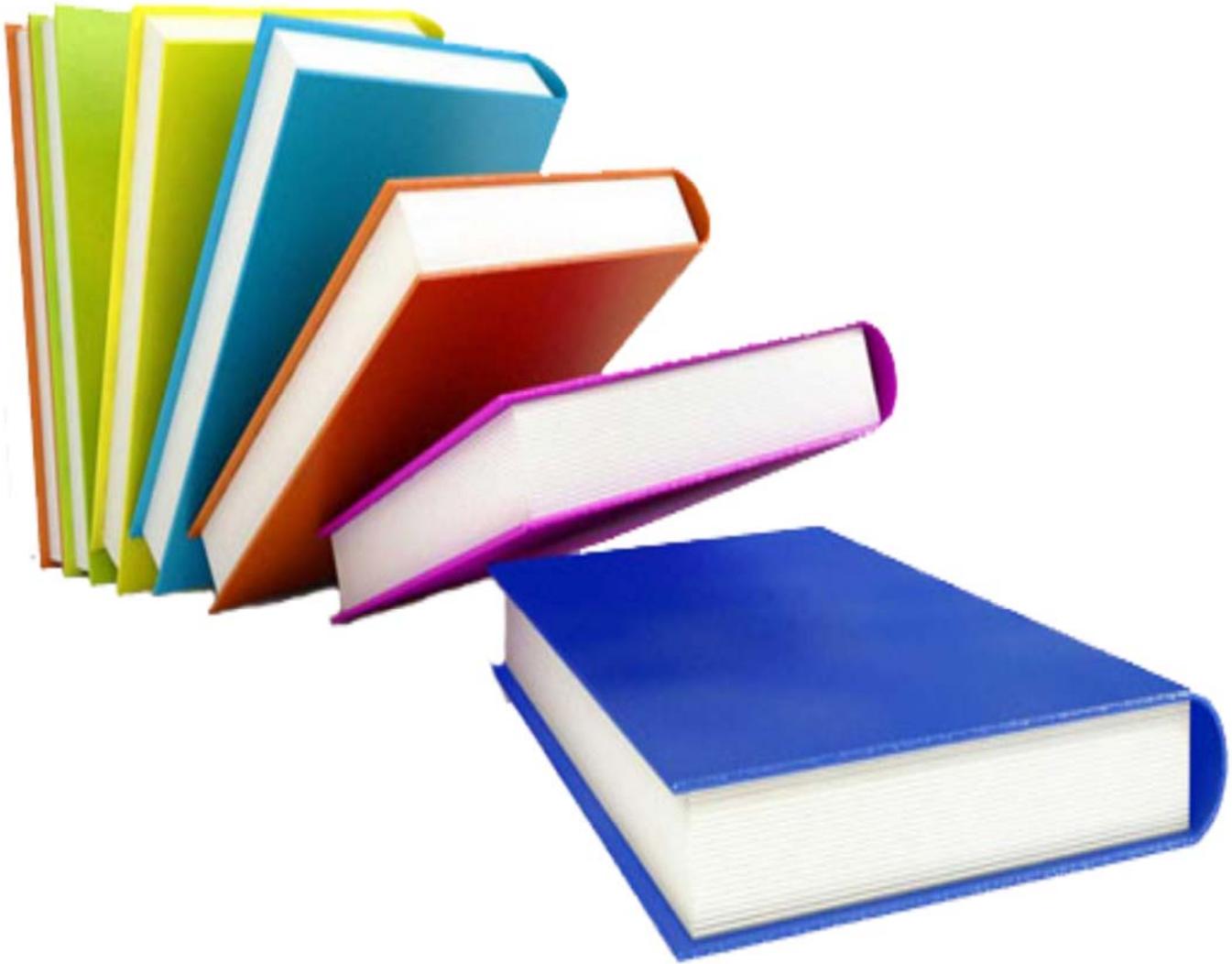
El total desarrollo de un sistema de información no se reduce solo a la construcción, es decir, su análisis, diseño, arquitectura, desarrollo, o al hacer unas pruebas y realizar sus correcciones y/o modificaciones y/o actualizaciones, tampoco concluye con su implementación; el éxito tampoco se conseguirá después de su conclusión y lanzamiento, si no se toma en cuenta una buena

capacitación para quien originalmente fue diseñado y desarrollado “EL USUARIO”, además de reconocer su experiencia en la elaboración de horarios.

Un óptimo proceso de desarrollo e implementación deberá ser iterativo e incremental, tener una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo), así como estar centrado en la arquitectura basada en componentes y guiado por los casos de uso, tal como lo plantea la Metodología RUP (Rational Unified Process), la cual es en la que ésta basado éste Sistema Generador de Horarios, para la Carrera de Ingeniería en Computación, de la Universidad Americana de Acapulco A.C.

Por último, es importante mencionar que el sistema no genera los horarios de manera automática, los realiza el usuario, por lo tanto puede haber errores de captura, por tal motivo se recomienda que el usuario ponga mucha atención para obtener un resultado libre de errores.

Bibliografía



Bibliografía.

- Pressman, R. (2006). *Ingeniería del Software: Un enfoque práctico*. México, D.F.: McGraw Hill.
- Sommerville, I. (2005). *Ingeniería del Software*. España: Prentice Hall.
- Álvarez, Sara, "Modelo entidad-relación". Internet: (<<http://www.desarrolloweb.com/articulos/modelo-entidad-relacion.html>>)
- Unified Modeling Language. UML: Resource Page. Internet: (<<http://www.uml.org/> >)
- Peñalara Software de España. Generador de Horarios para Centros de Enseñanza (GHC). Disponible en < <http://www.penalara.com/index.asp>> [última versión 2008]
- Introducción a la programación con restricciones <<http://iajournal.aepia.org/aepia/Uploads/20/72.pdf>>,[2003] Hamilton K., Miles R. Abril 2006. Learning UML 2.0. Ed. O'Reilly.
- Sebastopol. [Libro en línea] Disponible desde Internet en: <http://www.amazon.com/Learning-UML-2-0-Russ-Miles/dp/0596009828>

- Access / coordinado por Daniel Benchimol. - 1a ed. -, Buenos Aires: Fox Andina; Banfield - Lomas de Zamora: Gradi, 2011. v. 14, 192 p. ; 19x15 cm. - (Desde cero), ISBN 978-987-1773-11-4, 1. Informática. I. Benchimol, Daniel, coord. CDD 004.1
- Mendoza, M.C. (2001). La Universidad Americana de Acapulco, proyecto académico de José Francisco Ruiz Massieu. En José Francisco Ruiz Massieu: Aportaciones al Estado de Guerrero México: Universidad Americana de Acapulco.
- González, M., Pascual, A., & Lorés, J. (2006). Evaluación Heurística. España: Asociación Interacción Persona-Ordenador.
- Acid, S., Marin, N., Mrdins, J., Pons, O., & Villa, A. (2005) Introducción a las Bases de Datos. El Modelo Relacional. España: Parinfo.
- Pressman, Roger S. Ingeniería del Software: Un enfoque práctico. Quinta edición. España. Mc Graw Hill.
- Francisco Charte. (2013). VISUAL BASIC 2012 (MANUAL IMPRESCINDIBLE). España: Anaya Multimedia.
- SOLÓRZANO, J. Fernando, VILLAVICENCIO, C. J. (2003) Panorama Histórico de la Computación México Facultad de Ingeniería UNAM, Tomo I
- CAIRÓ, Osvaldo. (2003). Metodología de la Programación Algoritmos, Diagramas de Flujo y Programas 2a. edición, Tomos I y II. México Alfaomega.
- F. J. Ceballos. (2010). Enciclopedia de Microsoft Visual Basic (2da. Ed.), España. RA-MA
- aSc Time Tables. Disponible en <<http://asctimetables.com/>> [última versión 2008]

- GrupoCF Developer de Colombia. Software de Gestión Escolar (DocCF) Disponible en <<http://www.grupocfdeveloper.com/>> [última VERSIÓN 2008]
- Instituto para la Sostenibilidad Computacional <<http://computational-sustainability.org/crocs09>>, [Septiembre 20 de 2009]
- Rossi F. Van Beek P. Walsh T. Agosto 2006, Handbook of Constraint Programming. Holanda. Elsevier. [Libro en Línea] Disponible desde Internet: <http://www.elsevier.com/wps/find/bookaudience.cws_home/708863/description>
- E. Kendall, Kenneth y E. Kendall, Julie. (2005) Analisis y Diseño de Sistemas (6 ed.). México: Pearson México
- Microsoft, Corporation. (2012). The Official Microsoft IIS Site. <http://www.iis.net/>