



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería en Sistemas

***Generación de Horarios y Asignación de Aulas para las Facultades de
Administración, Filosofía y Derecho de la Universidad del Azuay***

**Tesis previa a la obtención del título de
Ingeniero en Sistema**

**Autores: Cristian Xavier Montero Mariño
María Verónica Vásquez Vítors**

Director: Ing. Pablo Esquivel

**Cuenca, Ecuador
2006**

Todo el contenido de esta obra es de completa responsabilidad de los autores de la misma.

Cristian Xavier Montero Mariño

Maria Verónica Vásquez Vítores

DEDICATORIA

A mis padres y hermanos
Las personas más importantes de mi vida
A ellos este trabajo
- Cristian Montero Mariño -

Dedico esta tesis a mis
padres quienes me brindan su
apoyo incondicional a lo largo de
toda mi vida.
Verónica Vásquez V.

AGRADECIMIENTOS

Agradecemos a la Ing. Janela Encalada, quien nos permitió la elaboración e implementación de nuestra tesis en el Centro de Computo de la Universidad del Azuay.

Expresamos nuestro sincero agradecimiento al Ing. Pablo Esquivel nuestro director de tesis quien nos brindó toda su ayuda y apoyo incondicional en la realización de la misma.

A todo el personal administrativo de las facultades involucradas en nuestro trabajo de la Universidad del Azuay quienes nos brindaron toda la información necesaria para el desarrollo de este trabajo.

ÍNDICE DE CONTENIDOS

DEDICATORIA	I
AGRADECIMIENTOS.....	II
ÍNDICE DE CONTENIDOS	III
ÍNDICE DE ILUSTRACIONES Y CUADROS	VI
RESUMEN	VIII
ABSTRACT.....	IX
INTRODUCCION.....	1
CAPITULO I: INVESTIGACION DE CAMPO	3
1.1 Universidad del Azuay.....	3
1.1.1 Elaboración de horarios de clase y asignación de aulas	3
1.1.1.1 Importancia.....	3
1.1.1.2 Principios y Normas.....	4
1.1.1.3 Proceso.....	5
1.1.1.3.1 Restricciones.....	6
1.2 conclusiones.....	8
CAPITULO II : ANALISIS	9
2.1. Lenguaje de Modelado Unificado (UML).....	9
2.1.1 Definición	9
2.1.2 Objetivos	10
2.1.3 Diagrama de casos de uso.	10
2.1.4 Diagrama de secuencia	12
2.1.5 Diagrama de clases	13
2.1.6 Diagrama de objetos	15
2.1.7 Diagrama de despliegue	16
2.2 Situación actual	16
2.2.1 Diagramas de flujo	17
2.2.2 Casos de Uso.....	18
2.2.3 Modelo entidad relación.....	21

2.2.4	Estructura de la base de datos	21
2.3	Modelo propuesto.....	23
2.3.1	Diagramas de flujo	23
2.3.2	Casos de uso	23
2.3.3	Diagrama de secuencias.....	33
2.3.4	Diagrama de clases	33
2.3.5	Diagrama de despliegue	33
2.3.6	Estructura de la base de datos	34
2.3.7	Ventajas	34
2.4	Conclusiones.....	35
CAPITULO III : DISEÑO		37
3.1	Diseño de la base de datos.....	37
3.1.1	Creación de tablas.....	38
3.1.2	Establecimiento de índices.....	40
3.1.3	Diccionario de datos.....	44
3.2	Diseño de la aplicación	45
3.2.1	Estandarización de las pantallas.....	46
3.2.2	Estandarización de procesos.....	49
3.2.3	Creación de las clases.....	50
3.3	Conclusiones.....	55
CAPITULO IV : HERRAMIENTAS DE DESARROLLO		57
4.1	Administración de base de datos en oracle.....	57
4.1.1	La instancia.....	58
4.1.2	Procesos que dan soporte a una instancia.....	61
4.1.3	Memoria utilizada por una instancia.....	62
4.1.3.1	System global area (sga).....	62
4.1.3.2	Program global area (PGA)	63
4.1.4	Organización de los datos.....	63
4.2	Conexiones a base de datos.....	65
4.3	Programación orientada a objetos.....	66
4.4	Desarrollo de aplicaciones distribuidas	70
4.5	Herramientas de desarrollo	73
4.6	Conclusiones.....	74
CAPITULO V : PROGRAMACION Y PRUEBAS DE LA APLICACION.....		75

5.1	Modelado de ciclo de vida.....	75
5.1.1	Modelo en cascada.....	76
5.1.2	Modelo construcción de prototipos	77
5.1.3	Modelo de desarrollo incremental.....	78
5.1.4	Modelo espiral.....	79
5.2	Establecimiento del modelo de ciclo de vida.....	80
5.3	Control de cambios	82
5.4	Gestión de pruebas.....	83
5.5	Conclusiones.....	84
CAPITULO VI : IMPLEMENTACION.....		85
6.1	Creación de la base de datos en el servidor de la Universidad del Azuay.....	85
6.2	Establecer la conexión a la base de datos.....	86
6.3	Instalación del software.....	87
6.4	Asignación de claves a los usuarios.....	92
6.5	Conclusiones.....	93
CONCLUSIONES		94
RECOMENDACIONES		96
GLOSARIO		97
BIBLIOGRAFIA		102
ANEXOS		105
	Anexo 1: Ficha de aspiración de horario.....	105
	Anexo 2: Modelo Entidad Relación del sistema actual	106
	Anexo 3: Estructura de la base de datos del sistema actual	107
	Anexo 4: Diagrama de Clases sistema propuesto	108
	Anexo 5: Modelo Entidad – Relación del sistema propuesto.....	109
	Anexo 6: Estructura de la base de datos sistema propuesto.....	110
	Anexo 7: Diccionario de datos	111
	Anexo 8: Descripción de las herramientas de desarrollo.....	115
	Anexo 9: Selección del software de desarrollo	117
	Anexo 10: manual del programador.....	118
	Anexo 11: Manual de Usuario.....	137

ÍNDICE DE ILUSTRACIONES Y CUADROS

Tabla 01. Nomenclatura Diagrama Casos de Uso 1.....	11
Tabla 02. Nomenclatura Diagrama Casos de Uso 2.....	11
Tabla 03. Nomenclatura Diagrama de Secuencias.....	12
Tabla 04. Estructura de una clase.....	13
Tabla 05. Nomenclatura Diagrama de Clases.....	14
Tabla 06. Tipos de Multiplicidad.....	15
Tabla 07. Nomenclatura Diagrama Despliegue.....	16
Tabla 08. Llaves Principales y Llaves foráneas.....	40
Tabla 09. Clases refinadas en el diseño, indican atributos con su tipo de dato y las operaciones que realizará.....	50
Tabla 10. Procedimiento de iniciación de una instancia.....	59
Tabla 11. Selección del Modelo de Ciclo de Vida.....	81
Tabla 12. Características computadores de cada facultad.....	87
Tabla 13. Requerimientos mínimos de hardware.....	88
Figura 01. Estructura de un objeto.....	15
Figura 02. Diagrama de Flujo Elaboración de Horarios.....	17
Figura 03. Diagrama de Flujo Asignación de Aulas.....	18
Figura 04. Diagrama de Caso de Uso Elaboración de Horarios.....	20
Figura 05. Diagrama de Caso de Uso Asignación de Aulas.....	21
Figura 06. Diagrama de Flujo Elaboración de Horarios.....	23
Figura 07. Diagrama de Caso de Uso Disponibilidad de Horario del Profesor.....	24
Figura 08. Diagrama de Caso de Uso Gestionar Distributivo.....	26
Figura 09. Diagrama de Caso de Uso Gestionar Materias Vigentes.....	28
Figura 10. Diagrama de Caso de Uso Gestionar Aulas de Clase.....	29
Figura 11. Diagrama de Uso Gestionar Horario de Clase y Asignación de Aulas...	31
Figura 12. Diagrama de Caso de Uso GEHO.....	32
Figura 13. Diagrama de Secuencia Establecer Materias Vigentes.....	33
Figura 14. Diagrama de Despliegue.....	33
Figura 14. Diferencia en pantalla con resolución de 800x600 y 1024x728.....	46
Figura 15. Pantalla de Inicio de Sesión.....	47

Figura 16. Formulario principal del sistema.....	47
Figura 17. Colocación del Título del proceso.	48
Figura 18. Solicitud de datos del docente.	48
Figura 19. Ingreso de escuela.....	48
Figura 20. Formato de pestañas en caso de realizar mantenimientos.....	49
Figura 21. Arquitectura de una Instancia en Oracle.....	60
Figura 22. Organización de los datos.....	64
Figura 23. Agrupación de los componentes de nuestro sistema.....	72
Figura 24. Modelo en Espiral.....	82
Figura 25. Pantalla del asistente de instalación	89
Figura 26. Asistente que define donde instalar el programa	89
Figura 27. Pantalla de confirmación	90
Figura 28. Pantalla de instalación	90
Figura 29. Características computadores de cada facultad	91
Figura 30. Ubicación general luego de haber instalado el sistema.	91
Figura 31. Inicio de Sesión.....	92

RESUMEN

Nuestro trabajo “Generación de horarios y Asignación de Aulas para las facultades de Administración de Empresas, Derecho y Filosofía de la Universidad del Azuay” esta orientado a automatizar este proceso de gran importancia para tan prestigiosa institución educativa. Partiendo del esquema utilizado actualmente en la universidad; esto es, conforme a la disponibilidad de cada uno de los profesores, las materias que se dicten en el periodo para el cual se elaborara el horario, número de créditos de las materias, el número de paralelos para cada materia y estandarizando la metodología con la cual se la realiza, tomando las características mas importantes de los procesos que se realizan manualmente e implementando nuevas metodologías de programación.

Con esto se logrará satisfacer las necesidades de la universidad permitiendo un ahorro significativo en tiempo, esfuerzo y de personal involucrado en esta tarea.

En el desarrollo de esta obra se va detallando cada uno de los aspectos que conforman este proyecto, lo cual permitirá a todas las personas que la lean entender claramente todo lo que involucra nuestro trabajo.

ABSTRACT

Our job, “Schedule Generation and Classroom Assignment for the Faculties of Business Administration, Law, and Philosophy in the University of Azuay” is oriented to automate this process of great importance for such a prestigious educational institution.

The project starts from the system currently used in the university. That is, based on the availability of each of the professors, the subjects to be taught in the period which the schedule is being designed for, the number of credits, and the number of classes per subject. Then it standardizes the methodology used by keeping the most important features of the manual processes and implementing new programming methodologies.

With this project, the university will be able to meet its need regarding schedules and classrooms with a significant saving of time, effort, and personnel involved in this task.

At this work is developed, every aspect that makes it up is described in detail, which will allow every reader to understand clearly all the material included in it.

INTRODUCCION

La Universidad del Azuay se propone ser una institución con calidad académica y humanística, que aporte al conocimiento y promueva el desarrollo integral de la persona y su entorno, formando personas con sólidos valores y conocimientos; y responder a las necesidades de la sociedad mediante la variada oportuna y permanente renovación de su oferta académica.

Se encuentra ubicada en la ciudad de Cuenca de la provincia del Azuay, en la Av. 24 de Mayo y Av. Hernán Malo. Cuenta con cinco facultades las mismas que comprenden varias escuelas de especialización.

Nuestro proyecto se basa en la importancia de agilizar el proceso para elaborar los horarios de clase al inicio de cada ciclo de estudio en cada una de las escuelas de la Universidad del Azuay.

Nuestro software de generación de horarios y asignación de aulas pretende facilitar la labor del personal encargado de realizar esta tarea manualmente.

El consejo de cada facultad es el autorizado de designar las personas responsables para manipular nuestro programa, a los que se les asignará un nombre de usuario y contraseña pasando a ser los únicos usuarios de nuestro software.

Toda la información será guardada dentro de una base de datos consistente que posee la universidad, la cual deberá ser actualizada por los usuarios.

Para el funcionamiento de nuestro software utilizaremos información ya existente en la base de datos de la universidad, entre estas:

- Capacidad y ubicación de cada aula.
- Las materias del pensum de cada escuela con sus respectivos créditos y descripción.
- Datos del personal docente.

Además se agregará a dicha base de datos la siguiente información:

- Las horas disponibles del personal docente de toda la semana.
- Las materias dictadas por cada profesor
- El numero de paralelos por materia de cada ciclo
- El número de paralelos en el cual el profesor esta dispuesto a dictar cada materia.
- Capacidad y ubicación de cada aula.
- Las materias del pensum de cada escuela con sus respectivos créditos y descripción.

La generación de los horarios estará basada en el esquema utilizado actualmente en la universidad; esto es, conforme a la disponibilidad de cada uno de los profesores, las materias que se dicten en el periodo para el cual se elaborara el horario, número de créditos de las materias, el número de paralelos para cada materia.

La asignación de las aulas necesitará de los créditos de cada materia, así como la escuela a la cual pertenece cada una de las aulas.

Al final del procesamiento de esta información por parte de nuestro software se obtendrá una completa descripción de los horarios para cada uno de los niveles vigentes, es decir, los ciclos que se dictarán en un periodo determinado en cada una de las escuelas pertenecientes a cada facultad. Además de las aulas en las cuales se deberán dictar cada materia en su respectivo horario, de acuerdo a la distribución de las escuelas en cada uno de los edificios de la universidad.

CAPITULO I: INVESTIGACION DE CAMPO

1.1 Universidad del Azuay

La Universidad del Azuay fue creada en 1968 por el gobierno ecuatoriano.

El Campus principal de la Universidad del Azuay se encuentra localizado en la Ciudad de Cuenca (Av.24 de Mayo 7-77 y Hernán Malo) cuenta con aproximadamente ocho hectáreas de terreno en las que se encuentran 19.557 m² de construcciones.

1.1.1 Elaboración de horarios de clase y asignación de aulas

La elaboración de horarios de clase y asignación de aulas está bajo la responsabilidad de los secretarios de las facultades que conforman la Universidad del Azuay. Esta actividad se la realiza antes de que concluya el ciclo de clases para que los alumnos cuenten con esta información al momento de escoger las cátedras que serán tomadas por dicho alumno el próximo periodo de estudio.

1.1.1.1 Importancia

La importancia de los horarios de clase y la asignación de aulas radica en que describen la estructura organizacional y distributiva con las que cuentan las materias dentro de las diferentes horas del día, así como la ubicación física del local en el que se impartirá dicha cátedra por parte del docente, dando a conocer tanto a estudiantes como al personal docente las horas designadas para realizar sus actividades estudiantiles y de enseñanza respectivamente. Para la universidad permiten cumplir una función de control de Asistencia eficiente al personal docente ya que cuenta con la información completa y personalizada de las horas en las que dicta cátedra.

1.1.1.2 Principios y Normas.

Dentro de la Universidad se han establecido normas y principios para la elaboración de los horarios las cuales describimos a continuación:

- El consejo universitario presidido por el señor Rector y conformado por el señor Vicerrector, decanos, representantes estudiantiles y representantes de empleados, es el encargado para decidir la fecha límite en la cual pueden presentar los horarios de clase cada facultad.
- El personal docente esta en la obligación de cumplir con un mínimo de ocho horas semanales para dictar clase y un máximo de veinte. En casos excepcionales se requiere de una autorización del Consejo Ejecutivo de la Universidad.
- El Consejo de Facultad define el distributivo para el nuevo ciclo, que consiste en asignar que materias van a ser dictadas por cada profesor.
- Con la información contenida en ese distributivo cada uno de los Directores de Escuela se disponen a elaborar los horarios y asignar las aulas para el próximo ciclo de estudio.
- El Consejo de cada facultad es responsable de la aprobación de los horarios de clase.
- El responsable de ingresar los horarios de clase al sistema de cómputo de la Universidad es el secretario de facultad.
- Los horarios de clase son puestos a disposición de los estudiantes en la página Web de la Universidad con la finalidad de brindar una ayuda para elegir las materias que cada estudiante esta dispuesto a recibir en el nuevo ciclo de clase.
- Una vez que los horarios han sido publicados en la pagina Web de la Universidad no podrán ser sometidos a cambios de ningún tipo, en caso de ser necesario realizar alguna modificación esta será elaborada al inicio de clases con mutuo acuerdo de los estudiantes y personal docente.

1.1.1.3 Proceso

El proceso para realizar los horarios de clase y distribuir las aulas varía dependiendo de cada facultad y su tiempo de elaboración fluctúa entre 10 y 15 días según su dificultad y grado de experiencia del encargado.

El proceso inicia con la solicitud a cada profesor de su aspiración de horario, para lo cual se le entrega una hoja en la que el docente indicará las horas en las que podrá dictar cátedra durante la semana (ver Anexo 1), existen facultades en las que no se requiere realizar esta labor como es el caso de la Facultad de Derecho en la que el encargado conoce la disponibilidad de cada profesor, ya que se ha venido manteniendo sin cambios dicho horario por muchos años. Como dato adicional en ciertas facultades se pide a los profesores ingresen un número de horas disponibles equivalente al doble de la suma de los créditos de las materias que va a dictar en el próximo ciclo, lo cual facilitará la elaboración de los horarios.

El siguiente paso consiste en que el Consejo de Facultad entre en sesión y establezca el distributivo definitivo para su facultad, previamente cada Director de Escuela elabora un distributivo tentativo de la escuela que representa en el que especifica las posibles cátedras que dictará cada uno de los profesores. En la sesión del Consejo de Facultad se evalúa esta información y se la completa con datos acerca de la cantidad de paralelos que existirán, basándose en un número estimado de estudiantes que en la mayoría de los casos es dado por la experiencia de cada encargado de facultad, en caso de que este número sea elevado, se considera la posibilidad de que más de un profesor dicte una misma cátedra.

Una vez obtenido el distributivo definitivo por facultad se procede a la entrega al responsable de elaborar los horarios, para que con esta información proceda a realizar dicha labor conjuntamente con la asignación de las aulas de las cuales cada encargado sabe cuales son las establecidas para cada escuela, incluso las que son directamente establecidas para una materia en particular si el caso lo amerita, por ejemplo, en el caso de la Escuela de Ingeniería de Sistemas perteneciente a la Facultad de Ciencias

de la Administración, posee aulas en el edificio de Administración para recibir clases en las mañanas, en cambio para las noches disponen de aulas del edificio del Colegio la Asunción y los laboratorios de computación en el Edificio de Filosofía todo el día.

Ya terminado el horario es entregado y sometido a una evaluación por parte del decano de la facultad, el cual revisará posibles fallas en el horario y las corregirá, además de realizar la aprobación del mismo. Luego es enviado al secretario de facultad para que se encargue de ingresar en el sistema computarizado de horarios que existente en la universidad para que se encuentre disponible esta información en la página Web de la Universidad y los estudiantes puedan realizar su prematricula.

1.1.1.3.1 Restricciones

Luego de haber realizado un análisis exhaustivo en el proceso de elaboración de horarios hemos determinado que existen muchas y variadas restricciones en cada facultad, a continuación exponemos dichas restricciones:

FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN

- Trata en lo posible que las materias dictadas por un profesor sean seguidas, es decir, que entre cada cambio de materia no existan horas vacías en las cuales el profesor este “desocupado”.
- Las materias son dictadas generalmente en dos horas seguidas. Dependiendo de la cantidad de créditos se establecerá el número de días en el que serán dictadas.
- Cuando existe mucho alumnado y por lo tanto, se decide crear un paralelo adicional, se conserva el horario y se busca un profesor que pueda tomar la materia en dicho horario, caso contrario, se procederá a realizar un cambio en el horario de dicho paralelo.
- Excepto la Escuela de Economía, las otras escuelas: Ingeniería de Sistemas, Administración de Empresas y Contabilidad Superior, se abren

todos los ciclos, y en el caso de Economía en septiembre los ciclos impares y marzo los ciclos pares.

- Para la Escuela de Ingeniería de Sistemas se consideran para las clases en la mañana y tarde las aulas que corresponden al edificio de Administración mientras que para las noches están designadas las aulas del edificio del Colegio La Asunción y para las clases que son dictadas en laboratorio están designadas los laboratorios del Edificio de Filosofía.

FACULTAD DE FILOSOFIA

- Se procura no dejar horas huecas en el horario de los estudiantes.
- Las materias son dictadas generalmente en una hora al día como máximo. Esto es sometido a elección del docente pudiendo variar entre dos horas seguidas y dos horas en el día. Dependiendo de la cantidad de créditos y de la preferencia del docente se establecerá el número de días en el que serán dictadas.
- Normalmente en septiembre de cada año se abren los ciclos impares y en marzo los ciclos pares.
- Las aulas que utilizan las siete escuelas de esta facultad están distribuidas en el edificio de Filosofía. Solo en casos excepcionales varían de edificio.

FACULTAD DE DERECHO

- Para los módulos que se dictan en la tarde existen conflictos en lo que se refiere a disponibilidad de aulas, ya que en esas horas otras escuelas utilizan las aulas requeridas.
- Ningún profesor posee disponibilidad de tiempo para dar clases en media mañana. La preferencia para la elaboración de horarios en este caso se debe a la antigüedad de docencia de los profesores, es decir, que los de mayor antigüedad tienen su horario inamovible.
- Las materias no se dictan en más de dos horas seguidas, depende de la preferencia del profesor.
- Existe la posibilidad de que una materia sea dictada por dos o más profesores a un mismo paralelo.

- Normalmente en septiembre de cada año se abren los ciclos impares y en marzo los ciclos pares.

1.2 Conclusiones

La metodología propuesta en base a entrevistas al personal encargado, se acopló plenamente a la investigación de campo, ya que desde el inicio pudimos obtener la información necesaria para nuestro estudio.

El conocimiento detallado y la correcta interpretación de las normas y principios fue fundamental para entender la secuencia de los procesos e identificar quienes son los responsables de esta labor, de esta manera se nos facilitó las entrevistas al personal apropiado.

A partir de dichas entrevistas, recopilamos los procesos que realizan para la elaboración de los horarios de clase, obteniendo diferentes metodologías de elaboración en cada facultad. No obstante se pueden definir claramente variables utilizadas por cada una de ellas que son similares, lo cual nos permite agruparlas e identificarlas para la fase de análisis.

CAPITULO II : ANALISIS

Una vez obtenidos los requisitos del sistema se realiza el análisis que no es más que lograr una comprensión mas precisa de los requisitos que nos ayude a dar una estructura al sistema incluyendo su arquitectura.

Para ello utilizamos el Lenguaje de Modelado Unificado (UML) que nos permite representar varios aspectos del sistema mediante diagramas visuales, con lo que obtenemos una visión mas clara de lo involucra nuestro sistema.

2.1. Lenguaje de Modelado Unificado (UML)

2.1.1 Definición

Según El Lenguaje Unificado de Modelado. Manual de referencia define a UML como *“El Lenguaje Unificado de Modelado (UML) es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas. Esta pensado para usarse con todos los métodos de desarrollo, etapas del ciclo de vida, dominios de aplicación y medios. El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar. UML incluye conceptos semánticas, notación, y principios generales. Tiene partes estáticas, dinámicas, de entorno y organizativas. Está pensado para ser utilizado en herramientas interactivas de modelado visual que tengan generadores de código así como generadores de informes. La especificación de UML no define un proceso estándar pero esta pensado para ser útil en un proceso de desarrollo iterativo. Pretende dar apoyo a la mayoría de los procesos de desarrollo orientados a objetos.”*

2.1.2 Objetivos

- El objetivo más importante de UML es que pueda ser utilizado por todos los modeladores, para conseguirlo UML incluye conceptos de los métodos líderes, como es la notación de OMT, Booch, objectory y otros métodos importantes.
- UML pretende trabajar con la mayoría de los procesos de desarrollo existentes.
- Pretende ser tan simple como fuera posible, manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir.

Como su nombre lo indica es un lenguaje, es decir provee de diagramas que cuentan con una nomenclatura para suministrar una vista estática o dinámica de la aplicación. Y no es un proceso que se lo deba seguir al pie de la letra. Simplemente se debe aplicar lo que se crea necesario para el óptimo entendimiento del sistema. Bajo este concepto hemos seleccionado los siguientes diagramas:

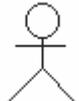
2.1.3 Diagrama de casos de uso.

Es un diagrama que muestra la funcionalidad global del sistema que permite a quienes participan en el desarrollo del software entender de manera general lo que el sistema hace.

Muestran la interacción entre los actores y los casos de uso. Los casos de uso representan la funcionalidad del sistema y los requisitos del sistema desde el punto de vista del usuario, los actores representan a usuarios externos que proporcionan o reciben información del sistema y la interacción significa asociaciones entre los actores y los casos de uso.

Nomenclatura

Tabla 01. Nomenclatura Diagrama Casos de Uso 1.

Nombre	Descripción	Notación
Actor	Persona o proceso que interactúa con el caso de uso. Se debe establecer el nombre del mismo.	 Nombre del Actor
Caso de Uso	Representa una parte de la funcionalidad del sistema. Se debe establecer el nombre del mismo.	 Nombre Caso de Uso

Pueden existir relaciones directas entre casos de uso las cuales pueden ser de varios tipos como se describe en la siguiente tabla.

Tabla 02. Nomenclatura Diagrama Casos de Uso 2.

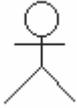
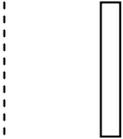
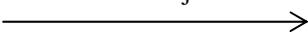
Relación	Función	Notación
Asociación	Representa la comunicación entre un actor y un caso de uso. La dirección que tome se indica mediante una flecha.	
Extensión	Indica un comportamiento adicional de un caso de uso base.	<< extend >> 
Inclusión	Utiliza la funcionalidad de un caso de uso base	<< include >> 

2.1.4 Diagrama de secuencia

Muestra la interacción de los objetos mediante mensajes en una secuencia de tiempo.

Nomenclatura

Tabla 03. Nomenclatura Diagrama de Secuencias.

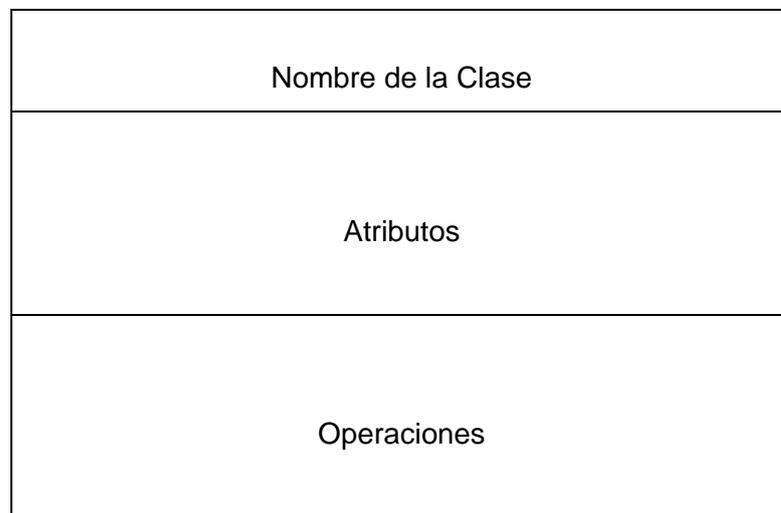
Nombre	Descripción	Notación
Actor	Persona o proceso que interactúa con los objetos que se incluyen en el diagrama.	 Nombre del Actor
Líneas de vida	Línea entrecortada que representa el tiempo de vida del objeto. Se dibuja como una línea doble mientras dura la activación de un procedimiento en el objeto.	
Objeto	Entidad sobre la que pueden aplicarse un conjunto de operaciones y que tiene un estado que almacena los efectos de las operaciones.	<div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">Nombre_objeto</div> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-bottom: 5px;">Nombre_objeto : Nombre_clase</div> <div style="border: 1px solid black; padding: 2px; width: fit-content;">: Nombre_clase</div>
Mensajes	Comunicación entre un objeto y otro. Se organizan en orden cronológico hacia abajo definiendo el orden en el que fluyen los mensajes.	 Mensaje

2.1.5 Diagrama de clases

Muestra un conjunto de clases y relaciones entre estas. Los diagramas de clase dan a conocer el diseño de un sistema desde el punto de vista estático. Proporciona una vista de algunas o de todas las clases que intervienen en el modelo.

Clase.- Una descripción de un conjunto de objetos que comparten la misma estructura, atributos, operaciones y relaciones. Las clases son la parte principal alrededor de la cual se organizan los sistemas orientados a objetos.

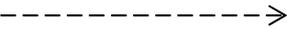
Tabla 04. Estructura de una clase.



La nomenclatura para definir una clase consta de un rectángulo dividido en tres secciones. En la parte superior contiene el nombre de la clase, la parte intermedia contiene los atributos aquellas características que identifican una clase. Y en la parte inferior se describe el comportamiento de la clase a través de operaciones.

Existen diferentes tipos de relaciones entre clases las cuales describimos en la siguiente tabla:

Tabla 05. Nomenclatura Diagrama de Clases

Relación	Función	Notación
Asociación	Una descripción de una conexión entre instancias de clases.	
Agregación	Relación entre 2 elementos, en la cual un cambio en un elemento independiente puede afectar la estructura de otro elemento dependiente.	
Generalización y Herencia	Generalización: Proporciona la capacidad para crear superclases que encapsulen una estructura y una conducta común para varias clases. Herencia: Define una relación entre clases donde una clase comparte su estructura y/o conducta con una o más clases.	

Una asociación puede tener atributos propios en cuyo caso es una asociación y una clase a la cual se la denomina “clase asociación”.

Multiplicidad: La multiplicidad define el número de objetos que participan en una relación es decir, el número de objetos que se unen entre sí. Suele especificarse mediante una expresión de texto formada por una lista de intervalo enteros separados por una coma, cada uno de los cuales tiene la forma:

Mínimo...máximo En donde mínimo y máximo son enteros o bien máximo puede ser un “*” que denota un límite superior no definido.

Tabla 06. Tipos de Multiplicidad.

EJEMPLO	DESCRIPCION
1	Exactamente uno
0..*	Ceros o más
1..*	Uno o más
0..1	Cero o uno
5..8	Rango específico (5,6,7,u 8)
4..7,9	Combinación (4,5,6,7,o9)

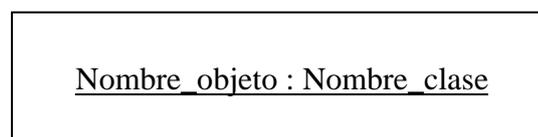
2.1.6 Diagrama de objetos

Muestra un conjunto de objetos y sus relaciones en un determinado momento, se consideran como un caso especial de las clases ya que presentan tanto las clases como las instancias.

Objeto: Es una manifestación concreta de abstracción. Una entidad real o conceptual en la que pueden aplicarse un conjunto de operaciones y que tiene un estado que almacena los efectos de las operaciones.

Nomenclatura

Figura 01. Estructura de un objeto.

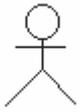


Este diagrama al igual que el diagrama de clases utiliza las mismas relaciones para poder conectarse entre sí.

2.1.7 Diagrama de despliegue

El diagrama de despliegue define la arquitectura física del sistema mediante nodos interconectados los cuales son elementos hardware sobre los cuales pueden ejecutarse los elementos software, podemos modelar los nodos y las conexiones del modelo de despliegue tan pronto como comience el flujo de trabajo de los requisitos.

Tabla 07. Nomenclatura Diagrama Despliegue.

Nombre	Descripción	Notación
Actor	Persona o proceso que interactúa con el nodo. Se debe establecer el nombre del mismo.	 Nombre del Actor
Conexión	Indica el medio de comunicación entre dos nodos.	Descripción _____
Nodo	Representa un elemento de hardware. Se debe incluir una descripción del mismo.	

2.2 Situación actual

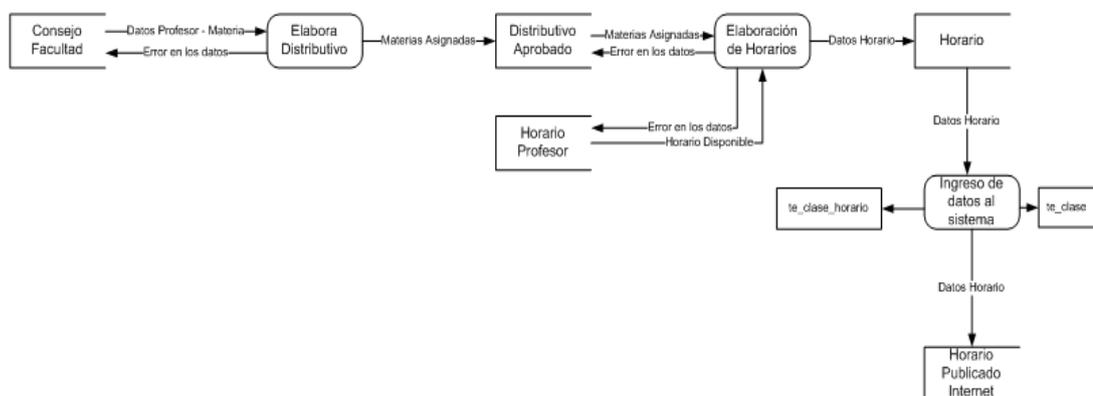
Una vez recopilada toda la información necesaria en el capítulo anterior, procedemos a realizar el análisis de la situación actual para de esta manera determinar cuales son las falencias y ventajas en el método actual, es decir definir que está bien o está mal en la elaboración actual de los horarios de clases y la asignación de aulas.

2.2.1 Diagramas de flujo

Para comenzar el análisis de la situación actual decidimos utilizar los diagramas de flujo de datos los cuales nos muestran gráficamente la secuencia de un proceso y los datos o mensajes que lleva entre proceso y proceso.

El diagrama de flujo de datos de la Elaboración de Horarios sigue los pasos según indican los reglamentos de la universidad. Comienza con la reunión del consejo de facultad, este con los datos de profesor y materias elaboran el distributivo, este contiene las materias vigentes para el próximo ciclo el cual luego de un análisis detallado se aprueba o se realizan cambios. Una vez con el distributivo aprobado tomamos las aspiraciones de horario de los profesores para el próximo ciclo y procedemos a elaborar los horarios de clase. Una vez creado el horario este es ingresado al sistema el cual a su vez se lo puede apreciar en Internet desde el momento de las pre-matrículas.

Figura 02. Diagrama de Flujo Elaboración de Horarios.



A continuación indicamos el diagrama de flujo de datos de la Asignación de Aulas. Cada facultad tiene su método de asignar aulas basados en la experiencia y en aproximados, pero logramos crear un estándar para poder graficarlo de la siguiente manera:

Figura 03. Diagrama de Flujo Asignación de Aulas



2.2.2 Casos de Uso

Los diagramas de casos de uso como explicamos con anterioridad muestran la funcionalidad global del sistema. Existen varios métodos de expresar estos diagramas ya sea gráficos o escritos. Los diagramas gráficos que utilizaremos en nuestro análisis son fáciles de entender para el usuario ya que gráficamente muestran los procesos, relación e interacción con el usuario. Aquellos diagramas escritos o también llamados narrativos, se recomiendan especificar la actividad del caso de uso y los requisitos necesarios para su funcionamiento, de este modo es más fácil entender cada uno de ellos.

A continuación indicamos el nombre del Diagrama de Casos de Usos, sus respectivos casos de uso y requerimientos. Al final de cada Diagrama se encuentra el gráfico correspondiente para tener mayor entendimiento.

1. ELABORACION DE HORARIO:

Este caso de uso indica todo los pasos que se realizan para que el horario de clases pueda se elaborado e ingresado al sistema.

1.1 Obtener Aspiración de Horario

Previamente el director de escuela solicita a cada profesor que dicta clase en su carrera que elabore un horario tentativo con las horas que dispone para dictar clases el próximo ciclo.

- R.1.1 El docente entregará su aspiración de horario indicando las horas y días en las que podrá dictar clases, aumentando un número de horas determinado por la facultad para facilitar la generación de horarios.

1.2 Elaboración Distributivo

Dentro de este caso de uso se convoca al Consejo de Facultad para que elabore el distributivo vigente para el próximo periodo. Una vez elaborado lo entregan al director de cada escuela.

R.2.1 El distributivo elaborado por el consejo de facultad, documento necesario para el director al momento de elaborar los horarios de clase.

1.3 Elaboración de Horario

Una vez que el director de escuela tenga en sus manos las aspiraciones de horario de todos los profesores y el distributivo procede a elaborar manualmente el horario de clases tomando en cuenta que no se crucen las horas entre las materias de un mismo ciclo y paralelo, no choquen las materias que dicta un mismo profesor, que exista un profesor designado para cada materia, que el profesor tenga suficientes horas disponibles para dictar un número de materias, etc.

R.3.1 La aspiración de horario, requisito explicado en el caso de uso 1.1.

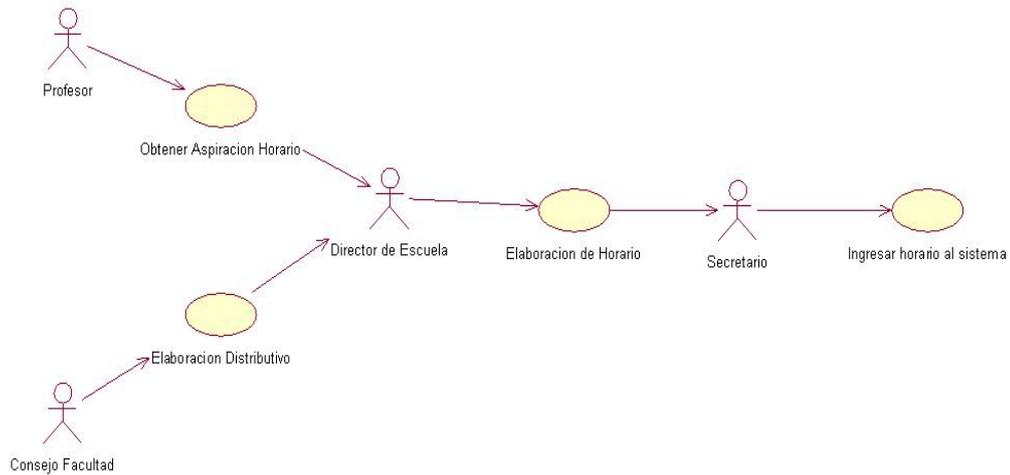
R.3.2 El distributivo, requisito detallado en el caso de uso 1.2.

1.4 Ingresar Horario al Sistema

Ya elaborado el horario de clases el Secretario de facultad es el designado para ingresar el horario al sistema de la universidad y pueda ser visualizado por los alumnos en el tiempo de las prematriculas.

R.4.1 Horario de clases.

Figura 04. Diagrama de Caso de Uso Elaboración de Horarios



2. ASIGNACION DE AULAS

2.1 Verificar número de alumnos por escuela

El director de escuela en base a la experiencia, periodo y al historial de alumnos matriculados en ciclos anteriores, calcula el número de alumnos que se matricularán el nuevo periodo.

R.1.1 Las estadísticas o historial de alumnos matriculados en ciclos anteriores.

R.1.2 El periodo correspondiente.

2.2 Asignación Aulas

Teniendo el listado de las aulas que utilizará una escuela durante el próximo ciclo y los horarios de clase se procede a determinar aulas a casa nivel; dependiendo de los horarios de clase un aula puede ser asignada a uno o más niveles y paralelos. Por tanto se necesita el requisito

R.1.3 Listado de todas las aulas disponibles para la facultad.

2.3 Listado de aulas designadas para la escuela

Antes de realizar la asignación de las aulas se debe realizar un listado de todas las aulas que serán designadas para una escuela y calcular si podrían abastecer el número de alumnos que reciban clase.

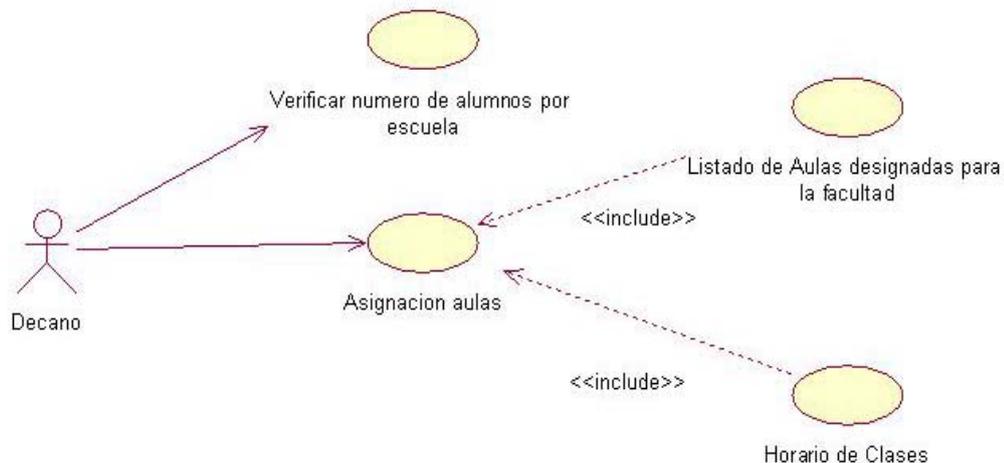
R.1.3 Listado Aulas.

2.4 Horario de Clases

Se debe primero elaborar el horario de clases para poder realizar la asignación de las aulas.

R.1.4 Horario de clases de cada nivel.

Figura 05. Diagrama de Caso de Uso Asignación de Aulas



2.2.3 Modelo entidad relación

El modelo entidad relación nos permite observar de manera gráfica de que manera se encuentran relacionadas las entidades. Anexo 2.

2.2.4 Estructura de la base de datos

El centro de cómputo de la universidad fue quien accedió luego de un análisis de requisitos a proveernos la base de datos con acceso a todas las estructuras de

tablas que necesitaba nuestro sistema, en base a esto, creamos de manera gráfica la estructura de base de datos.

Su estructura es un poco confusa ya que a nuestro criterio existen enlaces con diferentes tablas que no tienen razón de ser, se puede decir que conforme ha crecido la universidad se han ido añadiendo tablas acorde a la estructura principal es por ello que encontramos esta clase de desórdenes y lastimosamente no cubren las necesidades de manera rápida y eficaz como es el caso de los horarios de clase y asignación de aulas.

Cabe recalcar que no es nuestro objetivo reprochar ni desacreditar dicha estructura de datos, mas bien nos es de beneficioso para el desarrollo, si bien no está estructurada de la mejor manera según nuestro punto de vista, nos es de mucha utilidad conocerla ya que de este modo podemos analizar su forma de programar, ver la situación y proponer un cambio.

Debido a que muchas otras aplicaciones (entre ellas el control de asistencia a los docentes, prematriculas, horarios de clases en la página Web, etc.) utilizan esta estructura, nos es imposible cambiarla. Por ello acordamos con el centro de cómputo crear nuestra propia estructura enlazada con la antigua para el empleo de datos como son datos del docente, materias, pensums, facultades, etc.

Una vez que nuestro sistema genere el horario de clases y la asignación de horario debemos migrar estos datos a las tablas de horario de la antigua estructura para que puedan de esta manera utilizar otras aplicaciones.

Todos estos detalles son de gran relevancia para nuestra elaboración de la estructura de la base de datos ya que debemos ver tanto aspectos de cubrir las necesidades del usuario como aspectos técnicos necesarios en el momento en desarrollar el software.

Presentamos de manera gráfica la estructura de la base de datos de la Universidad del Azuay en el Anexo 3, vale recalcar que solo se encuentran las tablas que intervienen en los horarios de clase y son manipuladas con la aplicación actual.

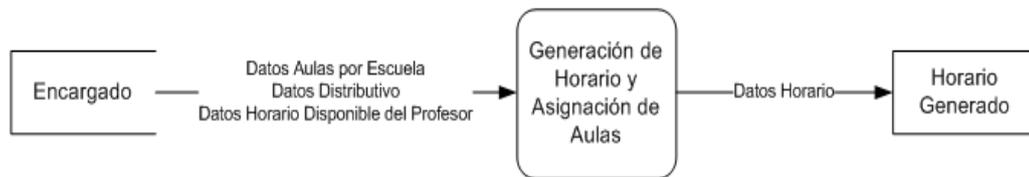
2.3 Modelo propuesto

Listo el análisis y estudio de la situación actual es posible determinar cuales son las falencias en la elaboración y que procesos se están realizando correctamente.

2.3.1 Diagramas de flujo

Con el siguiente gráfico expresamos el modo en que van a fluir los datos y el orden en que se van a realizar los procesos de todo el sistema:

Figura 06. Diagrama de Flujo Elaboración de Horarios



2.3.2 Casos de uso

Procedemos a elaborar el análisis de la situación propuesta con la utilización de diagramas de flujo de datos y el lenguaje UML

1. DISPONIBILIDAD DE HORARIO:

Este Diagrama es un subsistema que indica las actividades que se realizan para gestionar la disponibilidad de Horario de los profesores.

1.1 Obtener Horario Disponible de Profesor

Previamente el actor encargado solicita al actor profesor que labora en la universidad entreguen la aspiración de horario para el próximo ciclo, indicando las posibles materias que dictarán. Este caso de uso empieza en la recepción de dicha aspiración de horario elaborada por cada uno de los profesores y la verificación de las respectivas horas de clase.

R.1.1 El docente entregará su aspiración de horario indicando las horas y días en las que podrá dictar clases, aumentando un número de horas determinado por la facultad para facilitar la generación de horarios.

1.2 Mantenimientos de los Horarios Disponibles del Profesor

Dentro de este caso de uso el encargado gestiona el horario disponible del profesor, es decir, realiza las siguientes actividades: Ingresar el Horario disponible del profesor (obtenido del caso de uso 1.1), Modificar Horario Disponible, Eliminar Horario Disponible.

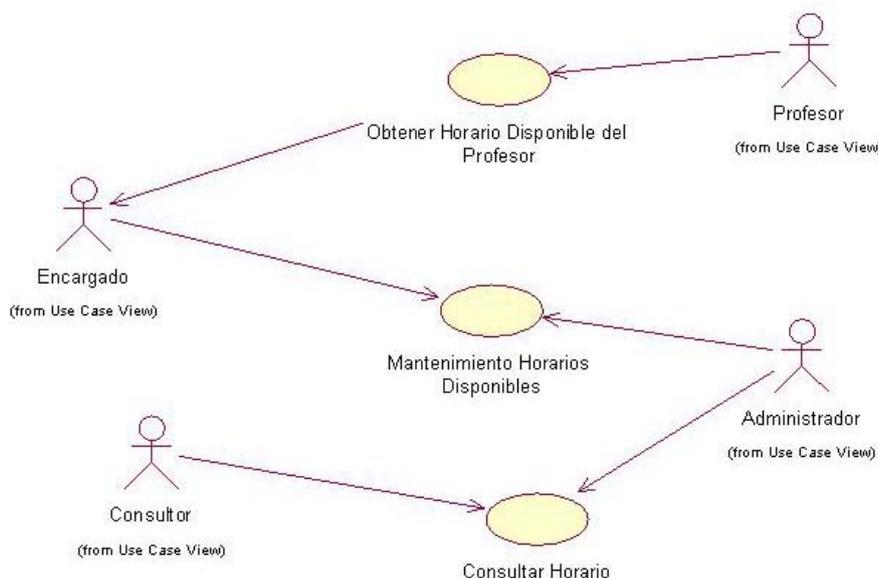
R.2.1 La aspiración de horario del profesor para el próximo ciclo (caso de uso 1.1)

R.2.2 El nombre del profesor para realizar la modificación y eliminación.

1.3 Consultas Horario Disponible del Profesor

El caso de uso Consultar Horario Disponible del Profesor muestra todos los reportes que se generen luego del ingreso de la aspiración de horario, dichos reportes visualizan la información dependiendo la necesidad del usuario. Los actores que tendrán acceso son: Encargado, consultor y administrador. Empieza este caso de uso una vez que se haya ejecutado el requisito R.2.1 contenido en el caso de uso 1.2. Mantenimiento de los Horarios Disponibles del Profesor.

Figura 07. Diagrama de Caso de Uso Disponibilidad de Horario del Profesor.



2. GESTION DEL DISTRIBUTIVO

Este subsistema muestra todas las actividades que se realizan dentro y fuera del sistema con el distributivo de facultad.

2.1 Materias Vigentes

Este paquete es el primero en ser considerado antes de elaborar el distributivo de la facultad, el cual está detallado en el punto 3.

2.2 Obtener Distributivo

Según los principios y normas de la Universidad del Azuay previamente se reúne el Consejo de Facultad para definir el Distributivo para el próximo ciclo. Al momento de definir la asignación de profesores a materias utilizan el requisito R.1.1, el cual es útil para examinar si es posible asignarle una o más materias, o número de paralelos a dar clases. Este caso de uso empieza al momento que el Consejo de Facultad entregue al encargado dicho distributivo.

R.3.1 El distributivo: Documento indispensable para nuestro sistema que contiene información acerca de:

- Ciclos que estarán vigentes para el próximo periodo.
- Asignación de Materias a Profesor.
- Paralelos que abrirán para cada ciclo.

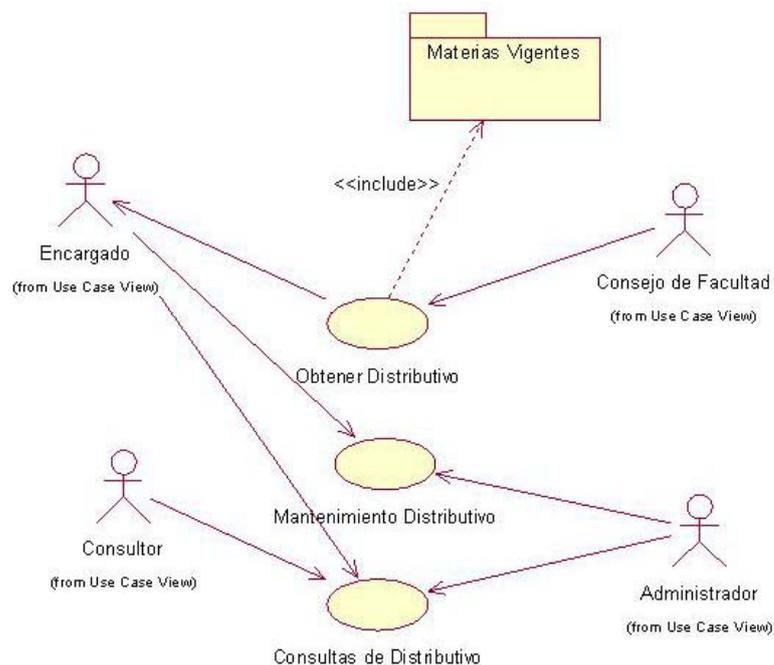
2.3 Mantenimiento Distributivo

Antes de este caso de uso primero debe ejecutarse el caso de uso 2.2 "Obtener Distributivo", por tanto, utiliza el requisito R.3.1. Gestiona el distributivo dentro del sistema realizando las siguientes actividades: Ingresar Distributivo, Modificar Distributivo, Eliminar Distributivo. Cabe recalcar que primero se debe ingresar el distributivo para luego realizar las otras actividades. Quienes podrán gestionar el distributivo son los actores encargado y administrador.

2.4 Consultas de Distributivo

Este caso de uso engloba todas las consultas y reportes que mostrará el sistema a los diferentes usuarios tales como los actores encargados, consultor y administrador.

Figura 08. Diagrama de Caso de Uso Gestionar Distributivo



3 GESTIONAR MATERIAS VIGENTES

Este subsistema indica las actividades que se realizan con las materias que serán dictadas el próximo ciclo. Dicho subsistema es utilizado en el Diagrama de Caso de Uso Gestionar Distributivo y a continuación se lo explicará con mayor detalle:

3.1 Definir Ciclos para próximo periodo

El Consejo de Facultad define los niveles que se abrirán el próximo ciclo, dependiendo la escuela y la fecha estos pueden ser pares, impares o todos.

3.2 Definir Materias Vigentes

Una vez que se haya ejecutado el caso de uso 3.1, el Consejo de Facultad procede a revisar las materias de los niveles definidos, verificando que correspondan al pensum vigente para el nuevo periodo. El listado de las materias vigentes se lo entrega al encargado.

R.4.1 Listado de materias vigentes para el nuevo periodo, indicando escuela, pensum y materia.

3.3 Agregar Características a Materias

Una vez que el encargado tenga el listado de materias vigentes, este agrega en el sistema propiedades a las materias tales como el número de paralelos, si necesita de un aula específica, etc. Por tanto se ejecuta luego del caso de uso 3.2.

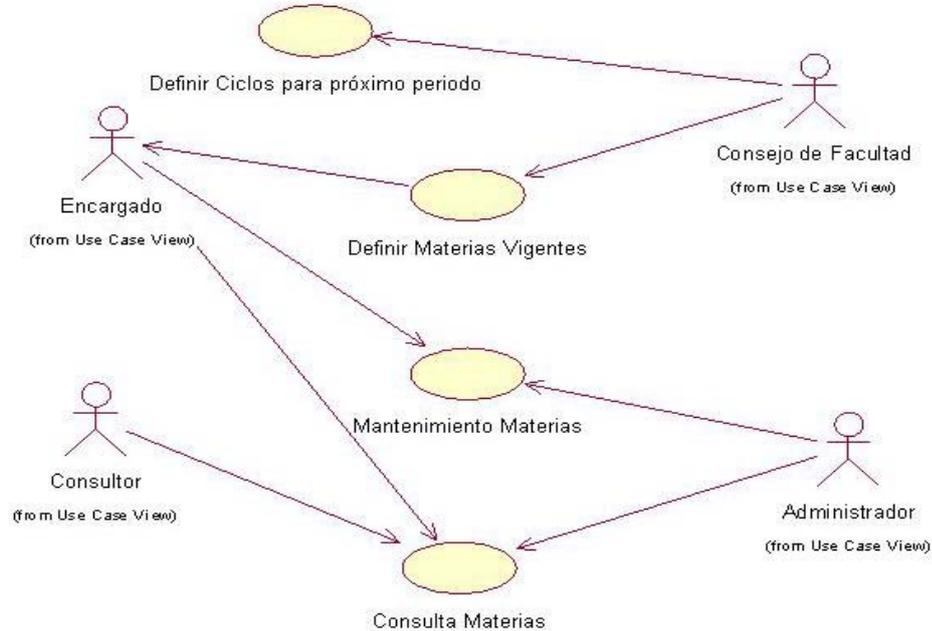
3.4 Mantenimiento de Materias

En este caso de uso se gestionan las actividades de modificar las propiedades de las materias, o eliminar dichas propiedades.

3.5 Consulta Materias

Muestra información a los actores consultor, encargado y administrador de cada materia vigente con sus respectivas propiedades, detallando la escuela y pensum a la que pertenecen.

Figura 09. Diagrama de Caso de Uso Gestionar Materias Vigentes



4. GESTIONAR AULAS DE CLASE

Paralelo a la gestión de Horario Disponible del profesor se realiza este diagrama, el cual pretende abordar la gestión que se realiza dentro y fuera del sistema respecto a las aulas de clase de toda la Universidad.

4.1 Asignar Aulas a Escuela

El encargado obtiene el listado de las aulas que utilizará su facultad para el próximo periodo, detallando las aulas que utilizará cada escuela. El encargado asignará en el sistema las aulas a las escuelas y asignando características basándose en el listado y dependiendo las horas que se dictan clase en una carrera, por ejemplo: la carrera de Ingeniería Automotriz se dicta solo en las tardes y noches, mientras que la carrera de Ingeniería de Alimentos se dicta solo en las mañanas.

R.5.1 Listado de Aulas que utilizará una escuela.

R.5.2 Horario de la Carrera.

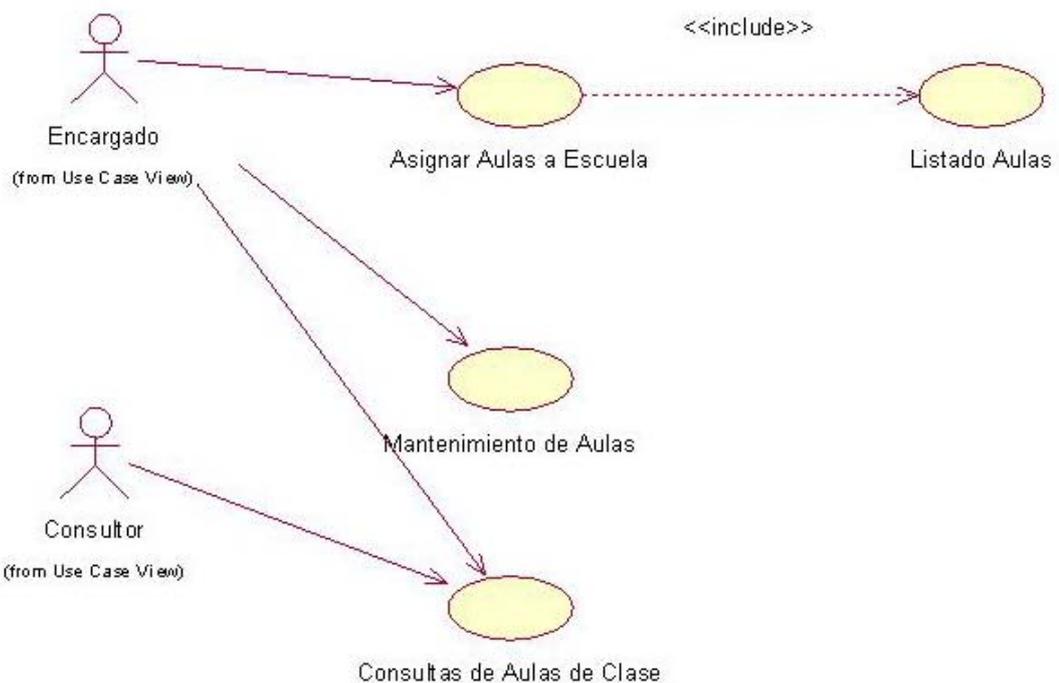
4.2 Mantenimiento de Aulas

Este caso de uso modifica o elimina un aula de aquellas registradas para una escuela.

4.3 Consultas de Aulas de Clase

En este caso de uso participan los actores encargado y consultor, quienes pueden observar los reportes que se generan de la información de las aulas.

Figura 10. Diagrama de Caso de Uso Gestionar Aulas de Clase



5 GESTIONAR HORARIO DE CLASE

Este diagrama se realiza una vez ejecutado los diagramas 1,2 y 4.

5.1 Generar Horario

Este caso de uso se lo ejecuta una vez realizados los casos de uso 1.2, 2.3 y 4.1, y realizando las respectivas verificaciones que confirmen que la información dentro del sistema es la correcta para así evitar errores en el sistema.

Genera los horarios de clase por escuela indicando el día, hora, materia, profesor que dictará la materia y aula en la que se dictará la materia.

R.6.1 Ejecutar casos de uso 1.2, 2.3, 4.1.

R.6.2 Información de los horarios disponibles de los profesores

R.6.3 Información del Distributivo

R.6.4 Información de las Aulas asignadas a las Escuelas.

5.2 Mantenimiento Horario

Una vez generado los horarios de clases este caso de uso permite modificarlos, eliminarlos o ingresar nuevos.

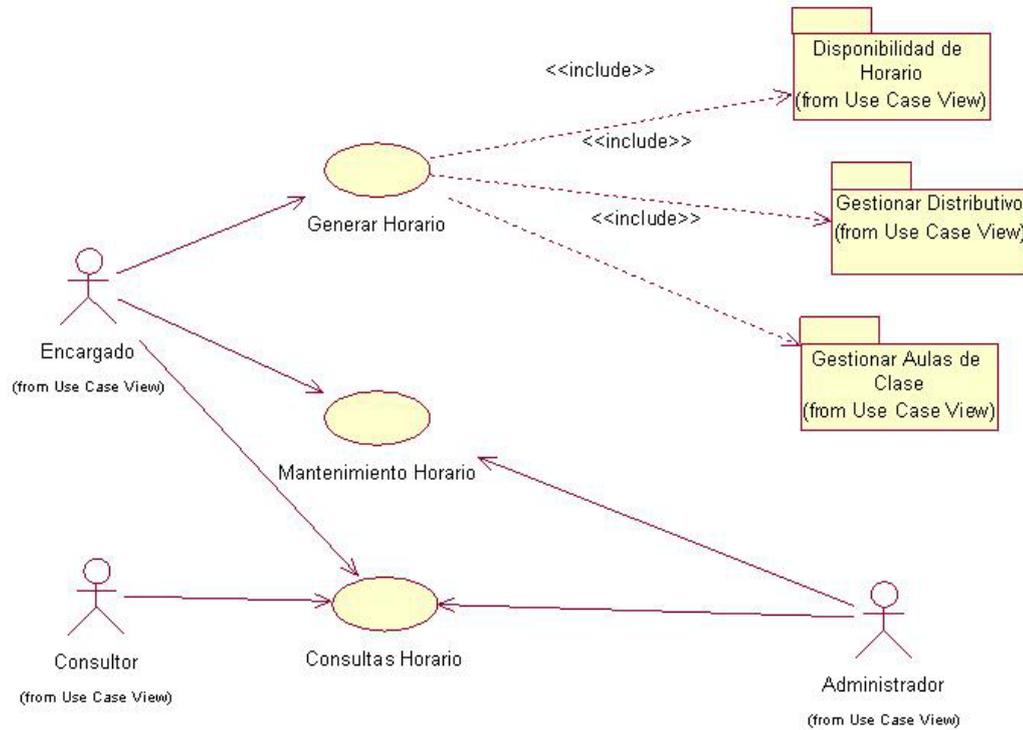
R.7.1 Ejecutar el caso de uso 5.1

5.3 Consultas Horario

El caso de uso consultas de horario incluyen todos los reportes que se desglosan de la generación de horario (caso de uso 5.1), como el horario de clases de cada profesor, horario de clase por paralelo, etc. Quienes interactúan con este caso de uso son los actores encargados, consultor y administrador.

R.8.1 Ejecutar el caso de uso 5.1.

Figura 11. Diagrama de Uso Gestionar Horario de Clase y Asignación de Aulas



6 GEHO

Este es el diagrama principal, donde engloba los paquetes que detallan los subsistemas del análisis.

6.1 Gestionar Disponibilidad de Horario

Subsistema que se explica detalladamente en el punto 1. Se debe ejecutar antes del subsistema 2 Gestión Distributivo.

6.2 Gestionar Distributivo

Paquete que se detalla en el punto 2.

R.9.1 Ejecutar el subsistema 1.

6.3 Gestión Aulas de Clase

Subsistema que se indica en el punto 4.

6.4 Gestionar Horario de Clases

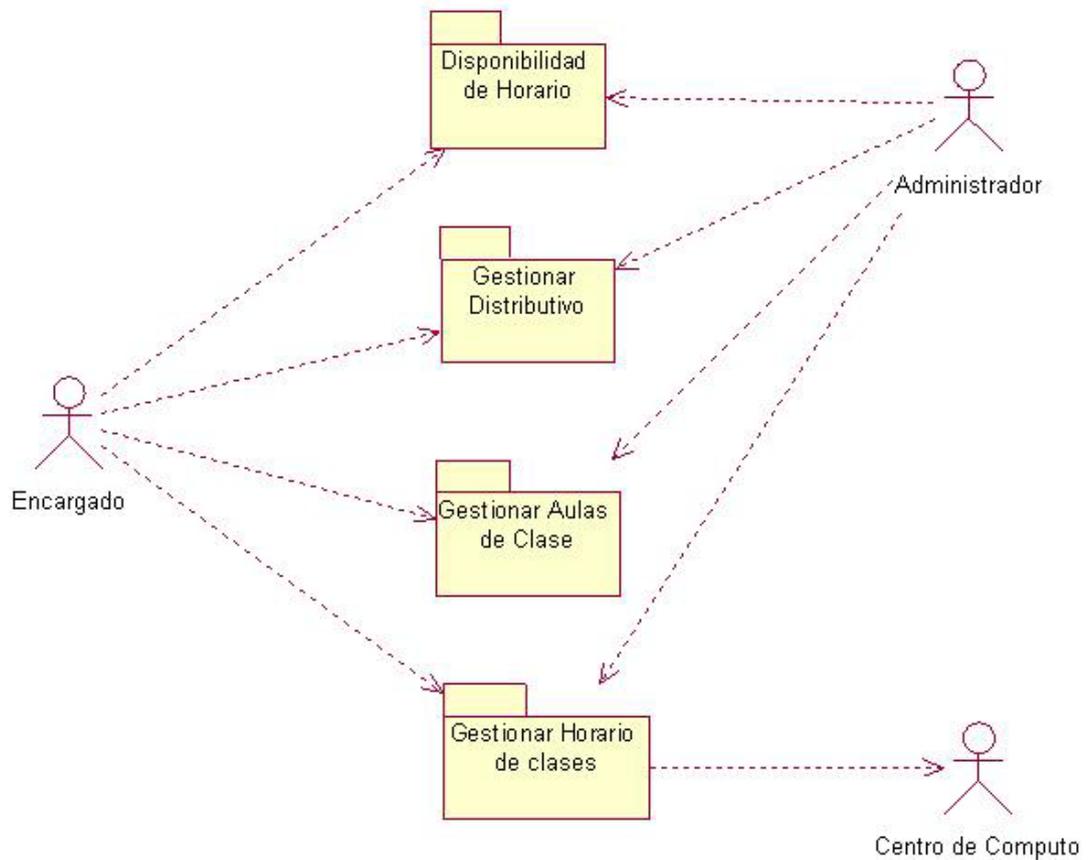
Paquete que se detalla en el punto 5. Una vez generados los horarios de clases se pasa esta información al centro de cómputo para que puedan mostrarla en la página Web de la universidad a todos los alumnos.

R.10.1 Ejecutar el subsistema 1.

R.10.2 Ejecutar el subsistema 2.

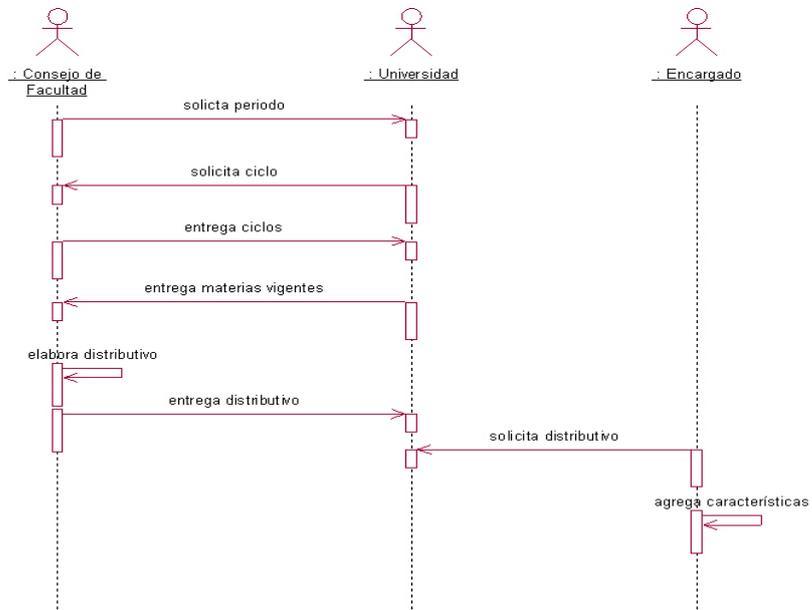
R.10.3 Ejecutar el subsistema 4.

Figura 12. Diagrama de Caso de Uso GEHO



2.3.3 Diagrama de secuencias

Figura 13. Diagrama de Secuencia Establecer Materias Vigentes



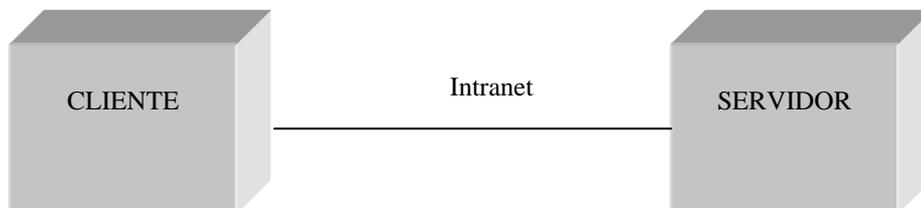
2.3.4 Diagrama de clases

El diagrama de clases se encuentra en el Anexo 4.

2.3.5 Diagrama de despliegue

Mediante el diagrama de despliegue podemos determinar como se va a disponer físicamente los nodos que componen el sistema:

Figura 14. Diagrama de Despliegue



2.3.6 Estructura de la base de datos

Utilizamos el modelo entidad – relación (Anexo 5) para graficar la relación de las entidades ya existentes en el sistema con las propuestas, lo cual nos facilita elaborar la estructura de la base de datos.

Para realizar la estructura de la base de datos de la situación propuesta tuvimos que basarnos en la estructura de la situación actual debido a que necesitamos obtener los datos de las tablas de la universidad para lograr la generación de los horarios. La organización que proponemos está analizada para poder cubrir todas las necesidades de la universidad.

La estructura de la base de datos del sistema propuesto se encuentra en el Anexo 6, en el cual detallamos nuestra estructura y sus enlaces con color rojo y adherimos la estructura de la situación actual con color negro para indicar como relacionamos todas nuestras tablas con las tablas del centro de cómputo.

2.3.7 Ventajas

Luego de haber observado los diversos diagramas que hemos puesto a consideración podemos decir que el sistema de Generación de Horarios y Asignación de Aulas presenta muchas ventajas sobre la metodología actual de desarrollo, debido a que el software será un programa interactivo que esta orientado a proporcionar facilidad en la manipulación de la información además de proveer los reportes necesarios a los usuarios del sistema.

El producto pretende reducir el tiempo que se toma en realizar esta tarea manualmente, además de guardar y mantener siempre la información actualizada, con lo cual se logrará una aceptación favorable por parte del personal docente tanto como de los estudiantes hacia la aplicación.

Además, presentará una ventaja en lo que se refiere al número de personas que deberán realizar esta labor ya que solo se requerirá del ingreso de información y automáticamente se obtendrán los horarios para todas y cada una de las escuelas de la universidad, de tal manera que una sola persona podría ser la encargada de manipular nuestro software.

El impacto social de este proyecto recaerá sobre toda la Universidad del Azuay, ya que serán muchas las personas beneficiadas con la implementación del software que vamos a desarrollar.

En el caso de las personas que se dedican a elaborar los horarios dentro de cada una de las facultades, se les reduciría de manera considerable el tiempo que les toma realizar esta labor.

Para los estudiantes los beneficios serán contar con información detallada y fiable de los horarios de clases al momento de hacer las prematriculas, además de tener conocimientos de las aulas en que recibirán clase dependiendo de las materias que elijan para el nuevo periodo. El personal administrativo tendrá control más detallado de la asistencia del personal docente, así como de las aulas en donde dictará clase.

2.4 Conclusiones

El análisis de la elaboración de los horarios se lo efectuó en base a toda la investigación de campo realizada en el primer capítulo, se clasificaron todos los datos y procesos necesarios que fueron incorporados en los diagramas de flujo y casos de uso.

Es así, que se realizó el análisis del sistema actual el cual es totalmente manual, ya que tan solo ingresan al sistema, el horario ya creado, y muy aparte manejan la asignación de aulas; causando al inicio de clases siempre confusión y cruces entre ciclos incluso de diferentes carreras. La situación actual por ser tan compleja resulta ser un “dolor de cabeza” para los responsables de su elaboración, es confusa y toma tiempo exagerado su construcción.

Por tanto, procedimos a plantear un sistema que elimine los errores en cuanto a cruces, disminuya la complejidad en el análisis de toda la información, optimice tiempo (factor muy importante). Unificamos los procesos de elaboración de horarios y asignación de aulas en un solo sistema el cual generará los horarios de clases con su respectiva aula automáticamente, siendo el responsable de esta labor el encargado en ingresar toda la información referente a horarios de disponibilidad de profesores, distributivo de trabajo y aulas asignadas a cada carrera.

La mejor manera para expresar nuestro análisis es mediante el Lenguaje de Modelado Unificado UML. La sencillez de sus diferentes diagramas nos permitió expresar nuestro análisis dependiendo la vista, ya sea esta, estática o dinámica. Como ejemplo podemos mencionar que los casos de uso nos fueron de utilidad para dar a entender de manera general la interacción del usuario con el sistema, mientras que el diagrama de clases nos permite expresar la agrupación de las variables en clases y su interacción entre si.

CAPITULO III : DISEÑO

3.1 Diseño de la base de datos

Uno de los pasos más importantes cuando se desea trabajar con base de datos es sin duda el diseño de base de datos. Si las tablas no se encuentran bien definidas no traerá muchos inconvenientes al momento de la programación y ejecutar consultas para obtener cualquier información. No es necesario saber si vamos a manejar pocos registros o miles de ellos pues es importante que esté bien diseñada así la base de datos sea eficiente y útil por largo tiempo.

Al momento de realizar el diseño de la base de datos tenemos que tomar en cuenta nuestras necesidades y decidir que información es realmente necesaria, ya que es posible generar algunos datos sin necesidad de almacenarlos en la base de datos. Dependiendo de los requerimientos del sistema que se va a diseñar se puede elegir la base de datos más adecuada, entre las que podemos mencionar SQL, MySql, Oracle, etc.

Para el diseño de la base de datos se debieron previamente determinar estándares respecto a los nombres de las tablas, campos, llaves foráneas, etc., los cuales nos son de mucha utilidad al momento de la programación.

Los nombres de tablas tendrán un tamaño máximo de 12 caracteres, en caso de tener varias palabras están serán unidas con el carácter guión bajo (_). Si dicho nombre es más largo del tamaño establecido esta se cortará hasta el carácter 12, por ejemplo la tabla Clase Horario tendrá como nombre Clase_Horari. Los campos de cada tabla tendrán un tamaño máximo de 21 caracteres, las palabras podrán ser unidas con el carácter guión bajo (_). El estándar de los índices será explicado más adelante en este capítulo.

3.1.1 Creación de tablas.

En la parte de diseño las entidades se convierten en tablas la cuales son combinadas para mejorar el desempeño de la base de datos. Por tanto estas pueden ser sustantivos que no denoten datos específicos o que no sean importantes para la empresa. Cada clase por tanto tiene atributos los cuales son propiedad descriptiva de la entidad incluyendo el tipo de dato que serán sean estos numérico, entero, carácter, fecha, etc.

En base al análisis realizado en el capítulo anterior y el modelo entidad – relación, procedemos a la creación de las tablas en el Gestor de Base de Datos ORACLE. Las tablas creadas son las que detallamos a continuación:

MATERIA: La tabla materia será aquella que guarda todas las características de las materias que serán vigentes para el próximo ciclo. Dentro de la base de datos tendrá el mismo nombre.

MATERIA LABORATORIO: En esta tabla se depositan los datos de materias que utilizan laboratorios o aulas específicas. De acuerdo con los estándares establecidos para el nombre de las tablas dicha tabla se llamará MAT_LAB.

PROFESOR MATERIA: Esta tabla guarda la asignación de las materias a los profesores que se hayan dispuesto en el distributivo. Según los estándares esta materia tendrá el nombre de PROFE_MATERI.

AULA ESCUELA: Guarda la asignación de aulas a las diferentes escuelas indicando el horario de ocupación ya sea este mañana, tarde o noche, dependiendo el horario de la carrera. Dentro de la base de datos se le llamará AULA_ESCUELA.

DIAS: Contiene los días de la semana. Según los estándares dentro de la base de datos tendrá el mismo nombre.

CLASE: Esta tabla es parte del horario de clases, esta tabla se llenará una vez que haya la información suficiente para ejecutar la generación y horarios. Se podría

decir que es la cabecera del horario pues tiene valores que no son repetitivos. Dentro de la base de datos tendrá el mismo nombre.

CLASE HORARIO: Esta tabla es considerada el detalle del horario de clases, ya que especifica las horas en que se dictará una materia, los profesores que la dictarán, los días y horas que se dará cátedra, y las aulas de clases. En base a los estándares esta tabla tendrá el nombre de CLASE_HORARI dentro de la base de datos.

CLASE TEMPORAL: En base a los requerimientos de usuario existen docentes que establecen un horario de clases para ciertas materias específicas, por ello esta tabla almacena los horarios de clases definidos antes de realizar la generación. Según los estándares se llamará CLASE_TEMP.

CLASE HORARIO TEMPORAL: Esta tabla es considerada el detalle del horario de clases temporal, ya que especifica las horas en que se dictará una materia, los profesores que la dictarán, los días y horas que se dará cátedra, y las aulas de clases, antes de que se realice la generación. En la base de datos tendrá el nombre de CLASE_H_TEMP.

PERMISOS: La tabla permisos almacena los permisos y tipos de accesos que tienen los usuarios a ciertas escuelas. Dentro de la base de datos tendrá el mismo nombre.

TIPO_ACCESO: En esta tabla se deposita la información de los tipos de accesos que tendrán los usuarios ya sean estos de ingreso, modificación o consulta. Según los estándares para definición de nombres de tablas esta conservará su mismo nombre.

USUARIO: Almacena los usuarios que podrán tener acceso al sistema.

3.1.2 Establecimiento de índices.

Para establecer los índices primero se decidió crear un estándar lo cual nos facilitaría al momento de la programación. Entre los índices tenemos las llaves primarias aquellas que nos servirán para identificar y diferenciar un registro de los demás es decir es única, corta, simple. Estas pueden ser simples (un atributo) o compuestas (varios atributos). Simples aquellas que son tan solo un campo; y compuestas son la unión de varios campos. Las llaves principales comenzaran con las letras PK_ seguido del nombre de la tabla.

Llaves foráneas, son aquellas que son primarias en una tabla que se relacione con la que estamos trabajando, para enlazar una tabla con otra. También pueden ser simples o compuestas dependiendo como se encuentren en la tabla original. Son aquellas que permite relacionar las tablas de la base de datos. Las llaves secundarias comenzarán con las letras FK_ seguido del nombre de la tabla que contiene la llave foránea y el nombre de la tabla cuya misma llave es primaria.

Tabla 08. Llaves Principales y Llaves foráneas.



A continuación detallamos los índices de cada tabla:

TABLA MATERIA

Llave primaria: PK_MATERIA. Llave compuesta por los campos MACODIGO y MACODAREA.

Llave foránea: FK_MATERIA_UDAMATERIA. Llave compuesta por los campos MACODIGO, MACODAREA. Se relaciona con la tabla TE_MATERIA que corresponde a la base de datos de la universidad.

TABLA MAT_LAB

Llave primaria: PK_MATLAB. Llave compuesta por los campos MACODIGO, MACODIAREA, MACODLABOR.

Llaves foráneas:

- FK_MATLAB_UDAAULA. Llave compuesta por los campos MACODLABOR, MACODLABOB. Llave que se enlaza con la tabla TE_AULA la cual es parte de la base de datos de la universidad.
- FK_MATLAB_UDAMAT. Llave compuesta por los campos MACODIGO, MACODIAREA. Se relaciona con la tabla TE_MATERIA perteneciente a la base de datos de la universidad.

TABLA AULA ESCUELA

Llave primaria: PK_AULA_ESCUELA. Llave compuesta por los campos AECODESCUE, AECODAULA, AECODBLOQUE, AEDISPONIB.

Llaves foráneas:

- FK_AE_AULA. Llave compuesta por los campos AECODBLOQUE, AECODAULA. Llave que se enlaza con la tabla TE_AULA la cual es parte de la base de datos de la universidad.
- FK_AE_CARRERA. Tipo simple cuyo campo es AECODESCUE. Se relaciona con la tabla TE_CARRERA perteneciente a la base de datos de la universidad.

TABLA PROFE_MATERI

Llave primaria: PK_MATLAB. Llave compuesta por los campos MACODIGO, MACODIAREA, MACODLABOR.

Llaves foráneas:

- FK_MATLAB_UDAAULA. Llave compuesta por los campos MACODLABOR, MACODLABOB. Llave que se enlaza con la tabla TE_AULA la cual es parte de la base de datos de la universidad.
- FK_MATLAB_UDAMAT. Llave compuesta por los campos MACODIGO, MACODIAREA. Se relaciona con la tabla TE_MATERIA perteneciente a la base de datos de la universidad.

TABLA PROFE_HORARI

Llave primaria: PK_PROFEHORARIO. Llave compuesta por los campos PHPROFESOR, PHDIA, PHHRAINIC, PHEMPRESA.

Llaves foráneas:

- FK_PROFEHORARIO_UDAPROFE. Llave compuesta por los campos PHPROFESOR, PHEMPRESA. Esta llave que se enlaza con la tabla TE_EMPLEADO la cual es parte de la base de datos de la universidad.

TABLA DIAS

Llave primaria: PK_DIA. Llave simple contiene el campo DIANUMERO.

TABLA CLASE

Llave primaria: PK_CLASE. Llave simple contiene el campo CLCODIGO.

Llaves foráneas:

- FK_CLASE_MAT_PENSUM. Llave compuesta por los campos CLCODESCUE, CLCODPENSU, CLCODAREA, CLCODMAT. Esta llave se enlaza con la tabla TE_MATERIA_PENSUM la cual es parte de la base de datos de la universidad.

TABLA CLASE_HORARI

Llave primaria: PK_CLASE_HORARI. Llave compuesta por los campos CHCODIGO, CHCODDOCEN, CHCODEMPRE, CHCODAULA, CHCODBLOQ, CHDIA, CHHORDESDE, CHMINDESDE, CHHORHASTA, CHMINHASTA.

Llaves foráneas:

- FK_CH_AULA. Llave compuesta por los campos CHCODBLOQ, CHCODAULA. Llave que se enlaza con la tabla TE_AULA la cual es parte de la base de datos de la universidad.
- FK_CH_EMPLEADO. Llave compuesta por los campos CHCODDOCEN, CHCODEMPRE. Se relaciona con la tabla TN_EMPLEADO perteneciente a la base de datos de la universidad.

TABLA CLASE_TEMP

Llave primaria: PK_CLASE_TEMP. Llave simple contiene el campo CLTCODIGO.

Llaves foráneas:

- FK_CT_MAT_PENSUM. Llave compuesta por los campos CLTCODESCU, CLTCODPENS, CLTCODAREA, CLTCODMAT. Esta llave se enlaza con la tabla TE_MATERIA_PENSUM la cual es parte de la base de datos de la universidad.

TABLA CLASE_H_TEMP

Llave primaria: PK_CHT. Llave compuesta por los campos CHTCODIGO, CHTCODDOCE, CHTCODEMPR, CHTCODAULA, CHTCODBLOQ, CHTDIA, CHTHORDESDE, CHTMINDESDE, CHTHORHAST, CHTMINHAST.

Llaves foráneas:

- FK_CHT_AULA. Llave compuesta por los campos CHTCODBLOQ, CHTCODAULA. Llave que se enlaza con la tabla TE_AULA la cual es parte de la base de datos de la universidad.

- FK_CHT_EMPLEADO. Llave compuesta por los campos CHTCODDOCE, CHTCODEMPR. Se relaciona con la tabla TN_EMPLEADO perteneciente a la base de datos de la universidad.

TABLA PERMISOS

Llave primaria: PK_PERMISOS. Llave compuesta por los campos PECODUSU, PECODESC.

TABLA TIPO_ACCESO

Llave primaria: PK_TIPO_ACCESO. Llave simple contiene el campo TACODIGO.

TABLA USUARIO

Llave primaria: PK_USUARIO. Llave simple contiene el campo USCODIGO, USCODEMP.

3.1.3 Diccionario de datos.

Se puede decir que el Diccionario de Datos es un catálogo que contiene información y describe la finalidad de las tablas de la base de datos. Normalmente el diccionario de datos almacena cierta información como: nombre mnemotécnico del campo, nombre de la tabla a la que pertenece, descripción del campo, tipo de dato ya sea este numérico, carácter, fecha, booleano, etc., y si es llave principal o foránea.

El diccionario de datos es de mucha utilidad ya que mediante él se puede tener control sobre los datos como recurso, es más fácil la comunicación entre los desarrolladores ya que aquí se especifica el significado exacto de cada campo, es posible identificar si hay redundancia, se puede tener un historial de los cambios que se realizan sobre la base de datos, se puede determinar el impacto que causará un cambio antes de implementarlo, ya que aquí tenemos información sobre los tipos de datos y sus relaciones con otras tablas. En otras palabras, proporciona información detallada, en el caso de auditorías. Anexo 5.

3.2 Diseño de la aplicación

En el diseño de la aplicación tomamos en cuenta aspectos de programación imperceptibles para el usuario como es la reutilización de código, y el diseño de la presentación del sistema como los formularios y pantallas que van a ser vistos por los usuarios.

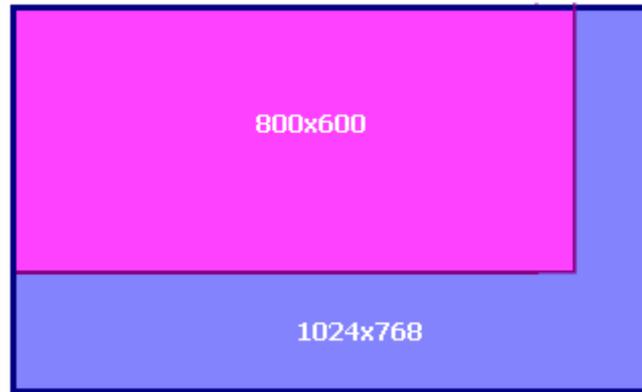
Dentro de los aspectos de programación decidimos reutilizar el código mediante los llamados User Controls, que son controles que nos permiten crear otros controles útiles para el programador, los cuales pueden ser utilizados en varios lugares de la aplicación y así crear un estándar en la presentación del sistema. Otra manera de reutilizar el código es con el uso de clases en donde depositamos todas los procedimientos que contengan peticiones de datos es decir las sentencias SQL; las cuales son llamadas al programa mediante la utilización de objetos, este punto lo tocaremos más a detalle en el siguiente capítulo.

Realizamos un estudio el cual consistía en examinar las computadoras que posiblemente tendrían nuestro sistema en cada facultad, averiguar sus características y en base a estas, elegir el lenguaje de programación más conveniente, determinar la resolución que se utilizarán, etc. La elección del lenguaje de programación y comparaciones entre ellos lo veremos en el capítulo V.

La importancia de la resolución de pantalla sobre la forma de ser visualizadas es esencial. A mayor resolución se dispone de más puntos de información para presentar los elementos en pantalla, pero estos puntos son más pequeños, con lo que los elementos de la interfaz (textos, imágenes, objetos de formulario, etc.) se ven más pequeños. Si diseñamos pantallas con una resolución dada, aquellos usuarios que visualicen a resoluciones menores no tendrán espacio en pantalla para contener toda la ventana, por lo que se verán obligados a usar las barras de desplazamiento. Por el contrario, aquellos usuarios que visualicen a resoluciones mayores tendrán demasiado espacio en pantalla, por lo que les quedará bastante espacio vacío, sin contenidos. Para evitar estas diferencias resolvimos utilizar como resolución estándar en todas las pantallas de 800 x 600, la cual es la más adecuada para el tipo de computadoras que tiene la universidad.

Esto nos evita el inconveniente que se produzca por la distorsión de las pantallas en caso de que se trabajen con computadoras cuyos monitores son antiguos y de baja resolución.

Figura 14. Diferencia en pantalla con resolución de 800x600 y 1024x728.



3.2.1 Estandarización de las pantallas.

Antiguamente el software era diseñado con poca consideración hacia el usuario: el usuario debía adaptarse al sistema. Sin embargo, esto hoy en día no puede ocurrir. La estandarización de diseño de pantallas es de mucha importancia para el diseñador con el fin de aumentar la usabilidad del producto a diseñar, es decir, que disminuya el tiempo de diseño de cada una de las pantallas de la aplicación y que tome diferentes aspectos que permita al software a desarrollar sea amigable al usuario.

Incluso al tener entrevistas con el personal de las facultades observamos sus actividades con respecto a la generación de los horarios, aspectos que tomamos muy en cuenta para estandarizarlos y permitir que sea mucho más fácil y amigable la interfaz de usuario.

A continuación detallamos las estandarizaciones que acordamos para el diseño de las pantallas:

- 5 Los colores que se utilizarán en el sistema son los estándares de Windows para permitir que cuando el usuario personalice los colores en el sistema operativo se modifique también en la aplicación.
- 6 Las ventanas utilizarán de fondo el color gris de Windows a excepción de los reportes, y contendrán la barra de título color azul con letras blancas.
- 7 Las letras dentro de los botones y etiquetas serán tipo Microsoft Sans Serif, tamaño 8, estilo de fuente negrita, estará escrito la primera letra de cada palabra en mayúscula y el resto en minúsculas.
- 8 Para ingresar a la aplicación se pedirá el nombre de usuario y la contraseña. Esta misma ventana se mostrará en caso de querer cambiar de usuario. Contendrá el título de la ventana y los botones Aceptar y Salir.

Figura 15. Pantalla de Inicio de Sesión

- 9 Se tendrá un formulario principal el cual llevará el menú en la parte superior de la ventana y en la parte central se visualizará el logo del sistema. Este formulario también contendrá a todos los otros formularios de la aplicación.

Figura 16. Formulario principal del sistema.



- 10 Colocar el título a todos los formularios dependiendo la actividad que realiza. Dicho título estará colocado en la parte superior izquierda del formulario al lado del gráfico de la aplicación. Su tipo de letra será Microsoft Sans Serif, tamaño 8, estilo de fuente normal, color blanco, la primera letra de la primera palabra estará en mayúscula.
- 11 El título principal de la actividad que realice el formulario estará situado en la parte superior central del mismo, con el fin de que el usuario reconozca la actividad que va realizar. Su tipo de letra será Microsoft Sans Serif, tamaño 14, estilo de fuente negrita, color negro, todo el título en mayúsculas.

Figura 17. Colocación del Título del proceso.



- 12 En caso de solicitar el código o nombre del docente se utilizará el siguiente esquema:

Figura 18. Solicitud de datos del docente.

Código	Apellido Paterno	Apellido Materno	Primer Nombre	Segundo Nombre	
<input type="text"/>	<input type="button" value="Limpiar"/>				

- 13 Para solicitar que el usuario ingrese la escuela o una carrera específica tendrá que digitar el nombre de la escuela:

Figura 19. Ingreso de escuela.

Escuela: <input type="text"/>

- 14 Todo formulario tendrá el botón de salir o cancelar.
- 15 Si se requiere solicitar el pensum de estudio de alguna carrera se digitará el código de dicho pensum en un cuadro de texto.

- 16 Para la selección de ciclo y paralelo se utilizará un combo para evitar que el usuario digite con errores los mismos.
- 17 Se emitirán reportes con fondo blanco, los cuales serán llamados de un formulario y podrán ser impresos.
- 18 El sistema contendrá mantenimientos en los cuales se podrá realizar ingresos, modificaciones y eliminaciones en caso de ser necesario, y van a estar disponibles en pestañas.

Figura 20. Formato de pestañas en caso de realizar mantenimientos.

The image shows a web application interface with a tabbed menu at the top. The tabs are labeled 'Identificación', 'Permisos', and 'Pestañas'. An arrow points to the 'Pestañas' tab, indicating it is the active one. Below the tabs is a form area with a dotted background. The form contains a label 'Código de Usuario:' followed by a text input field. Below this, there are five input fields for 'Profesor: Código', 'Apellido Paterno', 'Apellido Materno', 'Primer Nombre', and 'Segundo Nombre'. To the right of these fields is a button labeled 'Limpiar'.

3.2.2 Estandarización de procesos.

Antes de realizar la programación es muy importante encontrar cuales son los procesos más relevantes, por ello una vez identificados se sugiere resolverlos mediante pseudo código para que en el momento de la programación sea mucho más fácil codificarlos.

Uno de los tantos procesos que identificamos fue aquel para calcular el número de días que se dictará la materia. A continuación indicamos mediante pseudo código dicho proceso:

Horas

Créditos: Créditos de dicha materia.

Días: Número de días en los que se dictará la materia.

For i= días to 1 step -1

Horas = credito / dias

Credito= credito -1

Next i

3.2.3 Creación de las clases.

Ya teniendo las clases en la fase de análisis especificadas en el capítulo anterior podemos pulirlas para tener las clases que se utilizarán en el sistema al momento de la codificación. En este punto definimos por completo que vamos a trabajar con la programación orientada a objetos.

Cada clase le detallaremos los atributos que necesitará con su respectivo nombre y tipo de dato y las operaciones que realizará:

Tabla 09. Clases refinadas en el diseño, indican atributos con su tipo de dato y las operaciones que realizará.

Clase Aula	
<u>ATRIBUTO</u>	<u>TIPO DATO</u>
AtrAula	Carácter (4)
AtrBloque	Numérico (2)
AtrTipoAula	Carácter (3)
AtrNombre	Carácter(30)
AtrCapacidad	Numérico (3)
<u>OPERACIONES</u>	
Listar ()	
Listar por Escuela()	

Clase AulaEscuela	
ATRIBUTO	TIPO DATO
AtrEscuela	Carácter (3)
AtrAula	Carácter (4)
AtrBloque	Numérico (2)
AtrDisponible	Numérico (8)
OPERACIONES	
Guardar()	
Eliminar()	
ListarxEscuela()	
Contar_Aulas Repetidas	

Clase Bloque	
ATRIBUTO	TIPO DATO
AtrBloque	Numérico (2)
AtrNombre	Carácter (30)
AtrFacultad	Carácter (2)
<u>OPERACIONES</u>	
Listar()	
Obtener_Nombre()	

Clase Carrera	
ATRIBUTO	TIPO DATO
AtrCarrera	Carácter (3)
AtrNombre	Carácter (30)
AtrFacultad	Carácter (2)
<u>OPERACIONES</u>	
Listar()	
Obtener_Nombre()	

Clase Facultad	
ATRIBUTO	TIPO DATO
AtrFacultad	Carácter (3)
AtrNombre	Carácter (30)
<u>OPERACIONES</u>	
Listar()	
Listar_MateriasxFacultad	
Obtener_Nombre()	

Clase Generacion	
ATRIBUTO	TIPO DATO
AtrHorario	Numérico (8)
AtrProfesor	Numérico (5)
AtrMateria	Carácter (7)
AtrArea	Carácter (3)
AtrEscuela	Carácter (3)
AtrPensum	Numérico (4)
AtrNivel	Numérico (3)
Atrdias	Numérico (1)
AtrHoras	Numérico (2)
<u>OPERACIONES</u>	
Generar() Verificar_cruce_materias Verificar_aulas_disponibles Grabar_Horario Obtener_Distributivo Eliminar () Obtener_Paralelo()	

Clase Horario	
ATRIBUTO	TIPO DATO
AtrHorario	Numérico (8)
AtrEscuela	Carácter (3)
AtrPensum	Numérico (4)
AtrMateria	Carácter (7)
AtrArea	Carácter (3)
AtrParalelo	Numérico (2)
AtrGrupo	Numérico (2)
AtrDocente	Numérico (5)
AtrAula	Carácter (4)
AtrBloque	Numérico (2)
Atrdias	Numérico (1)
AtrHoraDesde	Numérico (2)
AtrHoraHasta	Numérico (2)

<u>OPERACIONES</u>	
Guardar_Horario()	
Eliminar_Horario ()	
Verificar_existencia ()	
Verificar_aulas_ocupadas ()	
Verificar_asignacion_profesor ()	
Genera_codigos ()	

Clase Materia	
ATRIBUTO	TIPO DATO
AtrMateria	Carácter (7)
AtrArea	Carácter (3)
AtrNombre	Carácter (30)
AtrNumParalelo	Numérico (1)
AtrNumAlumnos	Numérico (1)
AtrLaboratorio	Numérico (1)
AtrNumHorasLab	Numérico (1)
AtrNumHorasAula	Numérico (1)
AtrEstado	Numérico (1)
AtrPensum	Numérico (4)
<u>OPERACIONES</u>	
Ingresar()	
Modificar ()	
Eliminar ()	
Listar ()	
Verificar_Cupos ()	

Clase Materia – Laboratorio	
ATRIBUTO	TIPO DATO
AtrMateria	Carácter (7)
AtrLaboratorio	Carácter (4)
AtrArea	Carácter (3)
AtrBloque	Numérico (2)
AtrCapacidadLaboratorio	Numérico (2)
<u>OPERACIONES</u>	
Listar_Laboratorios() Ingresar () AsignarLaboratoriosAMaterias Eliminar()	

Clase Profesor – Horario	
ATRIBUTO	TIPO DATO
AtrProfesor	Numérico (5)
AtrDia	Numérico (1)
AtrHorainicio	Numérico (2)
AtrHoraFinal	Numérico (2)
<u>OPERACIONES</u>	
Guardar () Eliminar () Verificar_Disponibilidad () Listar ()	

Clase Profesor – Materia	
ATRIBUTO	TIPO DATO
AtrProfesor	Numérico (5)
AtrMateria	Carácter (7)
AtrArea	Carácter (3)
AtrEscuela	Carácter (3)
AtrPensum	Numérico (4)
AtrNumeroParalelos	Numérico (2)
AtrHorasNoSeguidas	Numérico (1)
AtrDias	Numérico (1)
AtrFechaInicio	Fecha
AtrFechaFin	Fecha
<u>OPERACIONES</u>	
Guardar_Asignacion ()	
Eliminar ()	
Listar ()	

3.3 Conclusiones

El diseño nos permite definir la arquitectura, hardware, software, componentes, es decir, “armar el esqueleto” de nuestro sistema, para que en base a este sea mucho más fácil la programación.

La base de datos fue diseñada con la arquitectura cliente – servidor, es una base de datos relacional minuciosamente elaborada para evitar futuros cambios al momento de programación y asegurar la consistencia e integridad de cada uno de los datos. La creación de las tablas fue en base al modelo Entidad – Relación elaborado en el capítulo anterior.

Los estándares definidos en los campos de tablas y procesos son de mucha importancia ya que así obligamos a llevar un orden en la programación, muy útil para momentos de cambios y modificaciones, en los cuales resulta bastante confuso si no se llevara organización. Los campos de cada tabla y su relación con otra se especifican en el diccionario de datos, es muy útil cuando manejamos campos mnemotécnicos ya que también suele indicar en una breve descripción el significado de cada campo.

El diseño de las pantallas nos permite ahorrar tiempo al momento de programar, ya que al tener definido como se mostrará la información es posible agrupar ciertas funciones en los llamados "User Controls", los cuales pueden ser utilizados en diferentes pantallas y realizan las mismas acciones.

Es claro que aplicaremos la Programación Orientada a Objetos (POO), por tanto, las clases creadas en el capítulo anterior fueron pulidas para en el momento de codificar sean las primeras en ser creadas y guarden todos los atributos y funciones que trabajan directamente con requerimientos a la base de datos.; de esta forma aplicamos los conceptos de encapsulamiento, herencia, abstracción, básico para POO.

CAPITULO IV : HERRAMIENTAS DE DESARROLLO

En este capitulo se dará a conocer detalles de las herramientas que se utilizaran para el desarrollo del software GEHO, sus beneficios, funcionamiento y un análisis del por que se escogió tal o cual herramienta.

Para el almacenamiento de la información que GEHO manipule se determino que el Sistema de Gestión de Base de Datos (DBMS Database Management System) será el de Oracle 9i debido a que la información de la Universidad del Azuay que nuestro sistema requiere esta bajo dicho gestor, además de que la Universidad cuenta con las respectivas licencias de uso, la mayoría de la información de esta herramienta presentada en esta obra fue obtenida del libro electrónico Professional Oracle Programming descrito en la bibliografía de la presente.

Se estudiara la manera de realizar un acceso seguro y confiable desde la herramienta de programación hacia la base de datos a través de los denominados ODBC, además de explicar el funcionamiento de la Arquitectura de Base de Datos a utilizarse.

4.1 Administración de base de datos en Oracle.

Oracle es un Sistema de Gestión de Base de Datos (SGBD o DBMS por sus siglas en Ingles) profesional desarrollado por Oracle Corporation, que cuenta con características que lo hacen único entre las cuales podemos mencionar:

- Soporte de transacciones.
- Estabilidad.
- Escalabilidad.
- Es multiplataforma.

Además de contar con un sistema de seguridad muy estricto que posee cuentas de usuario, passwords y definición de accesos para cada tipo de usuario.

A continuación damos a conocer la arquitectura de esta herramienta que es de vital importancia para una administración eficaz y eficiente por parte del DBA o Administrador de Base de Datos

La Base de Datos Oracle es una base de datos relacional, se divide en 2 partes importantes:

- La instancia es el servicio de software que actúa de intermediario entre los requerimientos de la aplicación y los archivos de datos. La instancia es un proceso dinámico, utiliza gran variedad de tareas y memoria para dar soporte sus operaciones.
- Los archivos de datos están almacenados en disco.

4.1.1 La instancia

La instancia de Oracle es una colección de procesos que maneja requerimientos de clientes para los datos.

Una instancia de Oracle podría iniciarse como parte de los procesos de arranque del servidor o iniciarla explícitamente a través de comandos. Si bien se puede iniciar una instancia con un comando, existen tres pasos diferentes en esta iniciación:

La siguiente tabla fue obtenida de Administración y optimización de Bases de Datos Oracle....

	Arrancar una instancia de BD	Montar una BD	Abrir una BD
Fase	Nomount	Mount	Open
3			Startup
2+1		startup mount	Alter database open;
1+1+1	Startup nomount	alter database mount;	Alter database open;
Al finalizar la fase, el significado del estado es	Decir que la instancia está arrancada significa que existen procesos y recursos que van a ser asociados a los ficheros de esa BD	Decir que la base de datos está montada significa que los procesos y recursos de la instancia se asocian a los ficheros de la BD física	Decir que la base de datos está abierta significa que está disponible a los usuarios, y que se han abierto todos los archivos log.
Pasos que se realizan	<ul style="list-style-type: none"> Se lee el archivo init<sid>.ora Se identifican los controlfiles ctl<sid>.ora Se crea e inicializa la SGA usando init<sid>.ora Se arrancan los procesos batch de background 	<ul style="list-style-type: none"> Se localizan los datafiles (*.dat, *.dbf) Se abren y leen los control files (archivos de control, ctl<sid>.ora) Se "monta" la BD (se obtiene <i>instance lock</i>) Se determina el modo de apertura de la BD (Normal o <i>crash recover</i>) 	<ul style="list-style-type: none"> Se abren y bloquean los datafiles (*.dat, *.dbf) Si es la primera instancia, se obtiene el <i>startup lock</i> Abre los online redo logs Si es la primera instancia, se realiza <i>crash recovery</i> si es necesario
Ejemplos de operaciones a realizar en este estado		<p>Cualquier sentencia <i>alter database</i> puede ser ejecutada si la base de datos está montada, y algunas de ellas, sólo si está montada. Montada y no abierta permite:</p> <ul style="list-style-type: none"> Acciones de archivo redo log Cambio del modo de archivo (Alternar entre los modos <i>archivelog</i> y <i>noarchivelog</i>) Recuperación Renombrado de archivos Con <i>media recovery</i>, recuperar archivos dañados Poner un datafile online o offline Mover datafile y redo log files 	<ul style="list-style-type: none"> Accesos normales de los usuarios
Razones por las que puede fallar cada fase	<ul style="list-style-type: none"> El ejecutable del sgbd está corrupto Init<sid>.ora es léxicamente o sintácticamente incorrecto, hay errores en la especificación de sus parámetros Las configuraciones especificadas en init<sid>.ora no son posibles (errores semánticos) 	<ul style="list-style-type: none"> No existen los control files o están corruptos (archivos de control, ctl<sid>.ora). En ese caso se deben recuperar de un backup o recrearlos mediante el comando create controlfile Los ficheros de control no están sincronizados. La solución es usar el más reciente. 	<ul style="list-style-type: none"> Alguno de los datafiles no existen como ficheros del sistema operativo Otro tipo de inconsistencias en la información contenida en el diccionario de datos No se encuentra alguno de los archivos de redo log esperados (log<sid>.ora)

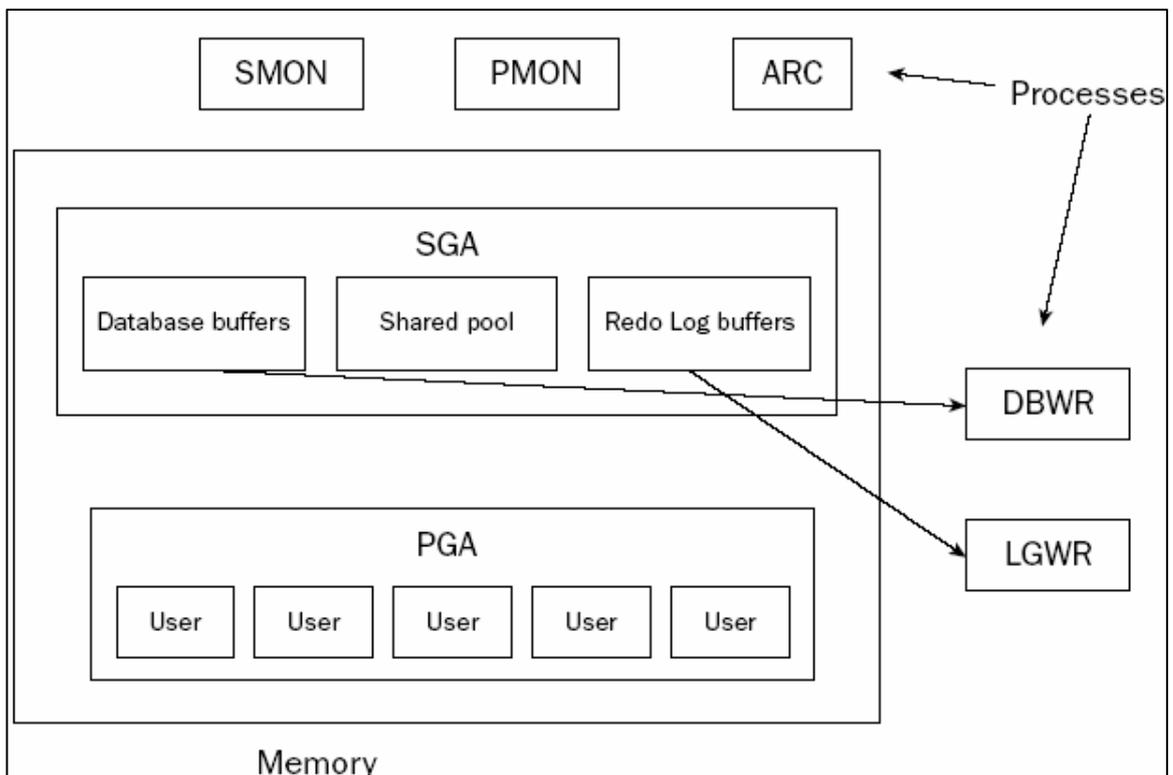
Tabla 10. Procedimiento de iniciación de una instancia

Una instancia puede ser detenida con algún comando o mediante el sistema de consola, la cual sigue la misma secuencia de eventos a la inversa. A continuación detallamos los modos de cierre de la Base de Datos.

Comando	Descripción
Shutdown normal	La Base de Datos no se parará si hay usuarios conectados
Shutdown transactional	Después de ejecutarla, los clientes no podrán comenzar nuevas transacciones, y la base de datos se parará cuando todas se hayan confirmado (commit) o anulado (rollback)
Shutdown immediate	Todas las sentencias SQL se terminan inmediatamente, no se espera a que se desconecten los usuarios. Las transacciones no confirmadas (commit) SI se deshacen, por lo que la Base de Datos se podrá mantener consistente
Shutdown abort	Se usa cuando muere un proceso background y no se puede tirar la Base de Datos en modo normal o immediate. Las transacciones no confirmadas (commit) NO se deshacen y la Base de Datos puede necesitar crash recovery durante el siguiente startup

La siguiente figura muestra la arquitectura de una instancia de Oracle:

Figura 21. Arquitectura de una Instancia en Oracle.



4.1.2 Procesos que dan soporte a una instancia.

Listener: El listener es un proceso que escucha en la red las peticiones que vienen a la instancia de Oracle. Puede dar soporte a más de una instancia de Oracle. Actúa como un intermediario entre los requerimientos de usuario desde máquinas remotas y la instancia. La falla del Listener provoca que la instancia que soporta no sea accesible por clientes remotos así como ninguna aplicación que acceda a esa instancia.

Procesos de Fondo

Un grupo de procesos se ejecutan conjuntamente con el Servicio de Oracle, dando soporte a las acciones de la instancia. Los principales procesos son listados a continuación con sus respectivos acrónimos:

- **Database Writer (DBWR).** Lee los bloques de datos y los almacena en el SGA, todas las transacciones manipulan la información contenida en los database buffers, los bloques de la base de datos modificados se graban en los archivos de datos periódicamente o cuando existe algún checkpoint o algún punto de sincronización mediante el algoritmo LRU (Least Recently Used) último recientemente utilizado.
- **Log Writer (LGWR).** Escribe los registros redo log en los archivos redo log. Los registros redo log son una copia de cada transacción que se produce en la base de datos permitiendo mantener la integridad de la misma en caso de algún fallo, esto permite un mejor rendimiento de Oracle ya que no consume sus recursos en escribir las modificaciones de los datos en los archivos de datos.
- **System Monitor (SMON).** Es el supervisor del sistema que se encarga de las recuperaciones fundamentales durante el arranque de la instancia. Esto puede ser necesario si la base de datos se paro inesperadamente por fallo físico, lógico o alguna otra causa. Este proceso realiza la recuperación de la instancia a partir de los archivos redo log, libera el espacio de segmentos temporales no utilizados y une extensiones libres de la base de datos para

conseguir espacio libre contiguo y mas fácil de alojar. Este proceso se ejecuta periódicamente para detectar si es necesaria su intervención.

- **Process Monitor (PMON).** Es el supervisor de procesos que asume la identidad de un usuario cuando su sesión termina de forma anormal, libera todos los recursos de la base de datos que se estaban utilizando y anula la transacción interrumpida. Este proceso se ejecuta periódicamente para detectar si es necesaria su intervención.
- **Archiver (ARC).** Proceso opcional. Es el encargado de copiar los archivos redo log en disco con propósitos de recuperación, siempre y cuando la base de datos este trabajando en modo ARCHIVELOG.

4.1.3 Memoria utilizada por una instancia

La figura anterior muestra varias áreas de memoria que son utilizadas por una instancia de Oracle.

4.1.3.1 System global area (sga)

El SGA es un área de memoria que es accesible a todos los procesos de usuario de una instancia. Existen tres áreas importantes:

- El redo log buffer (Buffer de rehacer) mantiene la información de los registros de redo log hasta que esta información pueda ser escrita en los archivos redo log.
- El Shared Pool (Repositorio Compartido) que mantiene la información que puede ser compartida entre los procesos de usuario, tales como sentencias SQL ejecutadas previamente, compilaciones de procedimientos almacenados e información recuperada del diccionario de datos de Oracle.
- El database buffer pools (Repositorios de Buffer de Base de datos) es un grupo de áreas de memoria que son usadas para mantener bloques de datos. Un bloque de datos que se mantiene en un repositorio de buffer (database buffer) puede ser accesado mucho mas rápido que si recuperamos el bloque de datos desde disco. Es por este motivo que el uso

de estos repositorios de memoria es esencial para alcanzar un óptimo desempeño.

4.1.3.2 Program global area (PGA)

El PGA es un área de memoria que esta solamente disponible para los procesos de servidor. Contiene ítems como variables de usuario e información de cada sentencia SQL, como el número de filas que fueron recuperadas. El PGA es también usado para el ordenamiento de datos para un proceso de usuario.

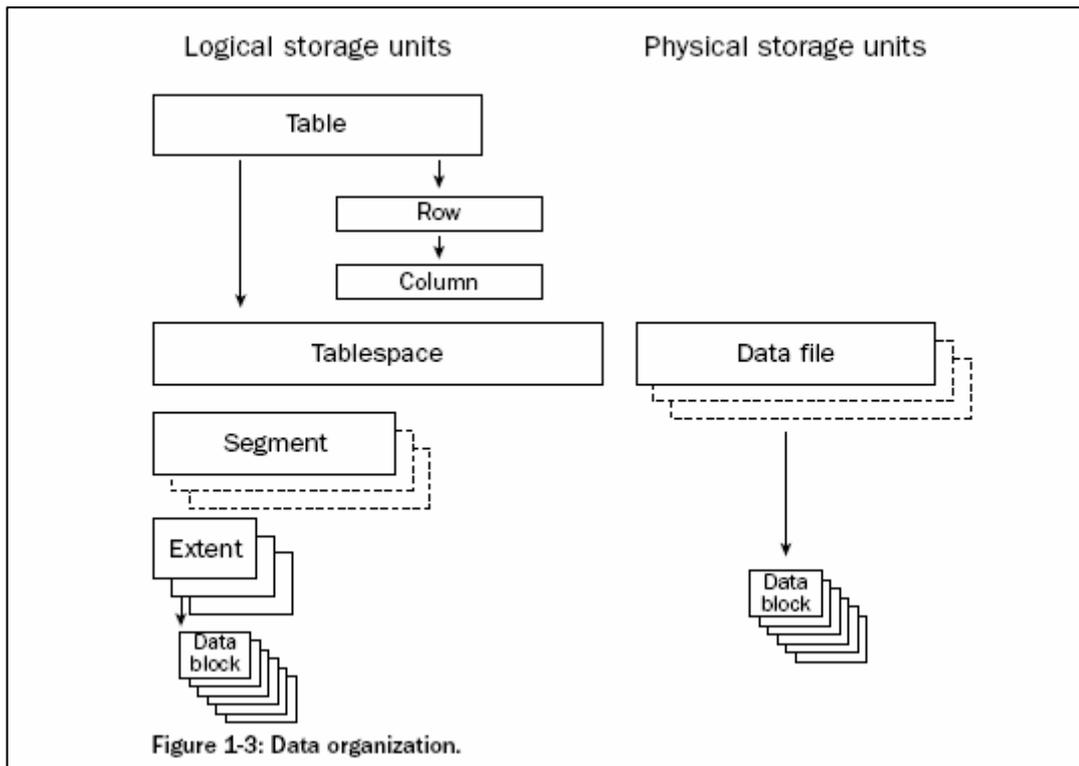
Cada vez se recibe una nueva sentencia SQL, un nuevo espacio en el PGA va a ser inicializado. Para evitar esta sobrecarga de aplicaciones con excesivas transacciones, tales como aplicaciones de procesamiento de transacciones en línea (OLTP), normalmente usa un pequeño conjunto de sentencias SQL que son continuamente reutilizadas por los procesos de usuario, reduciendo un potencial origen de sobrecarga.

4.1.4 Organización de los datos

Los datos están organizados en dos niveles: lógico y físico. En el nivel lógico, Oracle organiza sus datos en tablas, filas, columnas.

La estructura del almacenamiento lógico y físico se muestra en la siguiente figura:

Figura 22. Organización de los datos



Tablespaces

El tablespace es otra unidad de organización lógica usada por la Base de Datos Oracle, la cual actúa como el intermediario entre las tablas lógicas y el almacenamiento físico. Toda tabla o índice, cuando es creada, es almacenada dentro de un tablespace. Un tablespace puede contener más de una tabla o índice, la tabla o el índice puede ser particionado en múltiples tablespaces. Un tablespace contiene uno o más archivos, los cuales son entidades físicas usadas para almacenar los datos en el tablespace. Un archivo solo puede pertenecer a un tablespace. Este puede contener datos de fila (row data) o datos de índice (index data) y una tabla y un índice para una tabla pueden estar en diferentes tablespaces. También tiene un número de atributos, tales como el tamaño de bloque (block size) utilizado para el tablespace. Dicho tamaño de bloque es unidad de almacenamiento más pequeña en Oracle, devuelta por una sencilla operación de entrada y salida (I/O - input/output)

4.2 Conexiones a base de datos

Como describimos en el punto anterior nuestro Gestor de Base de Datos será Oracle 9i, ahora debemos definir un tipo conexión confiable y segura para satisfacer las necesidades de nuestra aplicación, de esta manera evitaremos la fuga de información o el fácil acceso a esta por parte de personas no autorizadas, además de esto, se tomara en cuenta que para el desarrollo de la aplicación se escogerá una herramienta que permita elaborar nuestra aplicación para que funcione bajo el sistema operativo Windows, debido a que todas las maquinas en las que va a trabajar nuestro software dentro de la Universidad realizan sus actividades sobre esta plataforma.

Por estos motivos se utilizara un administrador de conexiones de base de datos de la misma empresa que desarrolla el sistema operativo Windows, es decir, una aplicación elaborada por Microsoft Corporation.

El administrador de conexión OLE DB permite a una aplicación conectarse a un origen de datos mediante un proveedor de datos OLE DB.

Existe un administrador para cada una de las diferentes Bases de Datos, por ejemplo: existe el administrador de conexión OLE DB para SQL Server, el administrador de conexión OLE DB a Bases de Datos Oracle, etc. Con esto podemos deducir que no existirá ningún inconveniente en lograr una conexión confiable entre nuestra aplicación y los datos de nuestra base de datos, incluso con los datos de la base de datos de la Universidad del Azuay.

A este administrador de conexión OLE DB se lo puede configurar de varias maneras:

Dotándole de una cadena de conexión específica configurada para cumplir con los requisitos del proveedor seleccionado.

Dependiendo del proveedor, incluir el nombre del origen de datos a conectarse. Esto es de vital importancia para nuestro proyecto ya que al trabajar con bases de datos Oracle se debe definir un archivo (TNSNAMES.ORA) en el cual se

define mucha información que necesitara el administrador de base de datos OLE DB para realizar de mejor manera su trabajo, como por ejemplo: Credenciales de seguridad, indicar si la conexión creada desde el administrador de conexión se conserva durante la ejecución.

Otro punto muy importante a considerarse para el momento de la programación de nuestro software es que el origen de datos OLE DB proporciona cuatro modos de acceso distintos para extraer datos:

- Una tabla o vista
- Una tabla o vistas especificadas en una variable
- Los resultados de una instrucción SQL, esta consulta puede contar con parámetros.
- Los resultados de una instrucción SQL, almacenados en una variable.

Pero así como esta manera de enlazarnos a un origen de datos presenta muchas y muy importantes ventajas, presenta ciertas debilidades que deberán ser consideradas, entre ellas tenemos:

El proveedor Microsoft OLE DB para Oracle no admite los tipos de datos BLOB, CLOB, NCLOB, BFILE y UROWID de Oracle, y el origen de OLE DB no puede extraer los datos de tablas que contengan columnas asociadas a estos tipos de datos.

4.3 Programación orientada a objetos.

La Programación Orientada a Objetos (POO u OOP según sus siglas en ingles), intenta simular el mundo real a través del significado de objetos, además estos objetos colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar.

La programación orientada a objetos introduce nuevos conceptos, que superan y amplían conceptos antiguos ya conocidos. Entre ellos destacan los siguientes:

- **Objeto:** entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad ("métodos"). Corresponden

a los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa).

- **Clase:** definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.
- **Método:** algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, y/o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.
- **Evento:** un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente.
- **Mensaje:** una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.
- **Propiedad o atributo:** contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto, y cuyo valor puede ser alterado por la ejecución de algún método.
- **Estado interno:** es una propiedad invisible de los objetos, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos).

Asimismo, la programación orientada a objetos posee características que la distinguen de las demás metodologías de programación, las más importantes son:

- **Abstracción:** Cada objeto en el sistema sirve como modelo de un "agente" abstracto que puede realizar trabajo, informar y cambiar su estado, y "comunicarse" con otros objetos en el sistema sin revelar *cómo* se implementan estas características. Los procesos, las funciones o los

métodos pueden también ser abstraídos y cuando lo están, una variedad de técnicas son requeridas para ampliar una abstracción.

- **Encapsulamiento:** También llamado "ocultación de la información". Cada objeto está aislado del exterior, es un módulo natural, y cada tipo de objeto expone una *interfaz* a otros objetos que especifica cómo pueden interactuar con los objetos de la clase. El aislamiento protege a las propiedades de un objeto contra su modificación por quien no tenga derecho a acceder a ellas, solamente los propios métodos internos del objeto pueden acceder a su estado. Esto asegura que otros objetos no pueden cambiar el estado interno de un objeto de maneras inesperadas, eliminando efectos secundarios e interacciones inesperadas. Algunos lenguajes relajan esto, permitiendo un acceso directo a los datos internos del objeto de una manera controlada y limitando el grado de abstracción. La aplicación entera se reduce a un agregado o rompecabezas de objetos.
- **Polimorfismo:** comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando. O dicho de otro modo, las referencias y las colecciones de objetos pueden contener objetos de diferentes tipos, y la invocación de un comportamiento en una referencia producirá el comportamiento correcto para el tipo real del objeto referenciado. Cuando esto ocurre en "tiempo de ejecución", esta última característica se llama *asignación tardía* o *asignación dinámica*. Algunos lenguajes proporcionan medios más estáticos (en "tiempo de compilación") de polimorfismo, tales como las plantillas y la sobrecarga de operadores de C++.
- **Herencia:** las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen. La herencia organiza y facilita el polimorfismo y el encapsulamiento permitiendo a los objetos ser definidos y creados como tipos especializados de objetos preexistentes. Estos pueden compartir (y extender) su comportamiento sin tener que reimplementar su comportamiento. Esto suele hacerse habitualmente agrupando los

objetos en *clases* y estas en *árboles* o *enrejados* que reflejan un comportamiento común. Cuando un objeto pertenece a más de una clase se llama herencia múltiple; esta característica no está soportada por algunos lenguajes (como Java, Visual Basic .NET, etc.).

A los lenguajes de programación orientados a objetos se los denomina de quinta generación.

Pero como todo modelo de programación tiene sus ventajas y desventajas:

Debido a que esta metodología de trabajo se centra principalmente en lo que se refiere al diseño y definición de clases y considerando que los objetos son módulos completos de código, pequeños e independientes del contexto en el que se los utilicen, permite trabajar en un equipo de desarrolladores que basaran sus aplicaciones en la personalización de las funciones heredadas de los objetos bases y que permitirá luego reunir las todas en una solución final.

Esto nos indica que la POO permite una gran reutilización de código, facilitando el trabajo y reduciendo el tiempo de desarrollo de nuestras aplicaciones.

Para conseguir las ventajas antes mencionadas es necesario un gran esfuerzo en el momento de la definición y diseño de dichos objetos, esto involucra un exhaustivo trabajo de análisis que requiere de mucha disciplina para que los resultados sean los deseados y no una catástrofe completa.

Como otra desventaja podemos mencionar que la depuración de código orientado a objeto es algo más compleja que la depuración de código estructurado. Ya que debido a la presencia de la herencia se establece un árbol de jerarquía de objetos que en caso de que se presente un error deberá ser recorrido desde el inicio hasta el lugar en donde se produce dicho error, esto no se tiene que hacer en la programación estructurada.

4.4 Desarrollo de aplicaciones distribuidas

Las aplicaciones distribuidas o aplicaciones basadas en lo modelo n-tier (n-capas) de informática distribuida son la arquitectura de ultima generación para la construcción de aplicaciones multiplataforma.

El cambio que han sufrido los modelos de computación desde sus inicios, el evolucionar desde los sistemas compactos basados en mainframe y los tradicionales cliente-servidor, hasta los sistemas distribuidos multiplataforma altamente modulares, se debe a las últimas tendencias de las grandes empresas de tecnología, como Sun con su estrategia Sun Tone, o Microsoft con DotNET(.Net).

Ventajas de este modelo

- Desarrollo paralelo (en cada capa).
- Aplicaciones más robustas debido al encapsulamiento.
- Mantenimiento y soporte más sencillo (es más sencillo realizar una modificación sobre un componente que modificar una aplicación monolítica).
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es case lineal y no es necesario añadir mas código para conseguir esta escalabilidad.

Hay que tomar muy en cuenta que el diseño de una aplicación distribuida conlleva a la toma de decisiones con respecto a su arquitectura lógica y física, así como sobre la tecnología e infraestructura que se emplearan para implementar su funcionalidad. Para llegar a estas decisiones se debe tener un claro conocimiento de los requisitos funcionales, así como de los niveles de escalabilidad, disponibilidad, seguridad y mantenimiento necesarios (requisitos no funcionales, funcionales u operativos).

Al momento de diseñar una aplicación hay que considerar que:

- Solucione el problema para el que se diseña.
- Tenga en consideración la seguridad desde el principio, teniendo en cuenta los mecanismos adecuados de autenticación, la lógica de autorización y la comunicación.
- Proporcione un alto rendimiento y este optimizada para operaciones frecuentes entre patrones de implementación.
- Se puede administrar, permitiendo a los operadores implementar, supervisar y resolver los problemas de la aplicación en función del escenario.
- Funcione en los distintos escenarios de aplicaciones y patrones de implementación.

Componentes y niveles en aplicaciones

Se ha establecido un principio en el diseño de aplicaciones distribuidas, este es el de la división de la aplicación en componentes que ofrezcan servicios de presentación, empresariales y de datos. Una vez establecido esto, los componentes se pueden agrupar entre si, si las funciones que realizan son similares.

Esto también se puede aplicar a los servicios, desde un punto de vista de alto nivel ya que se pueden considerar que la solución basada en servicios esta formada por varios servicios, los cuales mantienen una comunicación mediante mensajes.

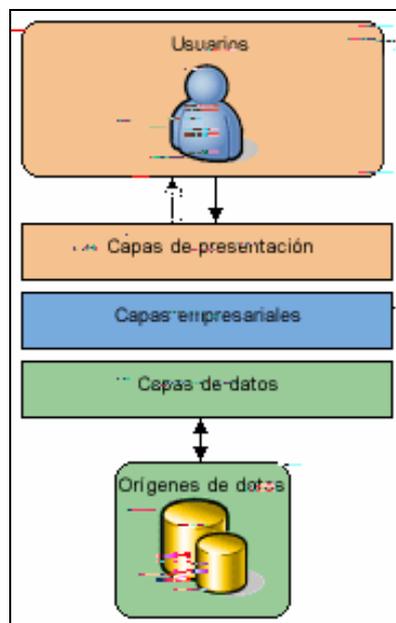
A partir de una visión conceptual, los servicios se pueden considerar como componentes de la solución global. Pero hay que tener en cuenta que un servicio esta conformado por componentes de software, al igual que cualquier otra aplicación, los cuales se pueden agrupar de forma lógica en servicios de presentación, empresariales y de datos.

Hay que tomar en cuenta que las capas son simplemente agrupaciones lógicas de los componentes de software que conforman la aplicación o servicio. Ayudan a diferenciar entre los distintos tipos de tareas que realizan los componentes, facilitando el diseño de la reutilización en la solución. Cada capa lógica contiene un número de tipos de componentes discretos agrupados en subcapas, cada una de las cuales realiza el mismo tipo de tarea específica. Al identificar los tipos genéricos de componentes que existen en la mayoría de las soluciones, puede construir un mapa coherente de una aplicación o servicio y, a continuación, utilizar este mapa como plano técnico para el diseño.

Con esta breve introducción queremos dar a conocer la forma en la que se desarrollara la aplicación de nuestra tesis, las ventajas y prestaciones de esta, así como todo el trabajo que involucrara el diseño de la misma.

La siguiente figura muestra como se encontraran agrupados los componentes de nuestro sistema.

Figura 23. Agrupación de los componentes de nuestro sistema.



Como podemos observar nuestro sistema contara con tres capas, cada una de las cuales realizara una función específica.

La capa de presentación hace referencia a la interfaz de usuario, es decir, es el medio por el cual cada usuario podrá interactuar y manipular todas las opciones de la aplicación.

La capa empresarial define a los componentes que se encargaran de validar y filtrar todos los datos que necesita la aplicación para realizar todas sus tareas de una manera correcta, en caso de presentarse alguna eventualidad esta es la encargada de presentar al usuario las advertencias pertinentes.

Por ultimo, la capa de datos es la encargada de extraer la información requerida por la capa empresarial desde la base de datos.

4.5 Herramientas de desarrollo

Para el desarrollo de nuestro sistema GEHO se requiere de una herramienta de desarrollo potente y que se adapte a nuestros requerimientos de programación como por ejemplo que permita una programación Orientada a Objetos, etc.

Para determinar dicha herramienta se ha realizado un análisis de las características (ver Anexo 8) de varios lenguajes de programación, a las mismas que se les ha establecido unos valores de importancia que hemos considerado con respecto al desarrollo de nuestra aplicación. Ver anexo 9.

Luego de haber realizado este análisis hemos determinado que la herramienta optima para el desarrollo de nuestra aplicación es Visual Basic .NET.

Además necesitaremos otras herramientas para cubrir el resto de características que le agregaremos a nuestro sistema, a continuación las describimos:

Nuestro sistema contara con ayudas para facilitar el aprendizaje del manejo de nuestra aplicación, para esto existe una herramienta llamada HTML Help Workshop que de un conjunto de paginas Web previamente elaboradas genera un archivo de ayudas con varias opciones de búsqueda, índice, etc. Debido a que es 100% compatible con Visual Basic .NET, además de otras ventajas ha

sido seleccionada como la herramienta que nos va a ayudar a cubrir esta parte del proyecto.

Como se indico anteriormente necesitaremos crear y diseñar paginas Web, para este cometido hemos seleccionado un paquete que abarca todas las opciones que se les pueden agregar a dichas paginas, este paquete es el Macromedia Studio MX, su fácil manejo y familiaridad que tenemos con esta herramienta nos ayudara a cumplir esta parte del proyecto en un tiempo mas corto.

4.6 Conclusiones

A lo largo de este capitulo obtuvimos una visión con las características mas importante de las herramientas de desarrollo de nuestra tesis, es decir, el lenguaje de programación en el que se codificara nuestra aplicación que será la herramienta Visual Basic .NET, el software de gestión de base de datos (DBMS por sus siglas en ingles) que será Oracle 9i ya que se encuentra en uso en la universidad, además de sus características que brindaran un mejor desenvolvimiento de nuestro software, para de esta manera ahorrar tiempo y esfuerzos al momento de obtener, proveer y dar mantenimiento a la información que es requerida para nuestro proyecto.

Otro aspecto importante que se abordo es, la manera en la que estas dos herramientas se conectarán para dar el mejor rendimiento y resultados de nuestra aplicación, basándonos en las prestaciones de los controladores OLE DB provistos por Microsoft Corporation y debido a la compatibilidad con las dos herramientas las escogimos como los administradores de la conexión aplicación – base de datos.

También se definió la metodología de programación que es la de Programación Orientada a Objetos (POO) de la cual se da una introducción muy interesante para familiarizarse con la misma (sección 4.3). Además, tomando en cuenta la institución a la que está orientado el presente proyecto, era necesario establecer a nuestro sistema como una Aplicación Distribuida, cuyas características están claramente especificadas en uno de los puntos del capitulo 4.4.

CAPITULO V : PROGRAMACION Y PRUEBAS DE LA APLICACION

En este capítulo se establece el ciclo de vida de nuestro software, que hace referencia a las fases en las que se desarrollara nuestra aplicación, lo cual nos permitirá llevar a cabo nuestra labor con el máximo desempeño y de una manera organizada y confiable. La información acerca de los modelos del ciclo de vida fue obtenida de *UNSA. Ingeniería de software, ciclos de vida...* descrita en la bibliografía.

Conjuntamente se determinara la manera en la cual se van a controlar los cambios que requiera nuestro sistema para satisfacer todas las necesidades que requieran dentro de las facultades de la Universidad del Azuay a las que esta orientada nuestra aplicación. Un aspecto que va de la mano con el anteriormente mencionado es el que hace referencia a la manera en la cual se realizaran las pruebas del sistema, es decir a probar nuestro sistema en situaciones reales para lo cual se contara con la colaboración de personal de la universidad que la labora en las dichas facultades.

5.1 Modelado de ciclo de vida

Un modelo de ciclo de vida de software es una visión de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas.

Un modelo de ciclo de vida del software:

- Describe las fases principales de desarrollo de software.
- Define las fases primarias esperadas de ser ejecutadas durante esas fases.
- Ayuda a administrar el progreso del desarrollo, y
- Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

De esta manera podemos decir que los modelos por una parte suministran una guía para los ingenieros de software con el fin de ordenar las diversas

actividades técnicas en el proyecto, por otra parte suministran un marco para la administración del desarrollo y el mantenimiento, en el sentido en que permiten estimar recursos, definir puntos de control intermedios, monitorear el avance, etc.

A continuación damos a conocer las características más relevantes de los modelos de ciclo de vida que mejor se podrían facilitarnos el desarrollo de nuestra aplicación.

5.1.1 Modelo en cascada

Fue propuesto por Boehm en el año de 1976. Desde el nacimiento de la Ingeniería del Software como un área se define el proceso del software para grupos de desarrollo llamado "Ciclo de Vida en Cascada".

El desarrollo del Software se realiza a través de una secuencia de fases. Estas fases tienen un conjunto de metas bien definidas y las actividades dentro de una fase contribuye a la satisfacción de las metas.

Este modelo posee seis fases:

- 1- Estudio de Factibilidad
- 2- Requerimientos
- 3- Diseño
- 4- Codificación y Test Unitario
- 5- Integración del Sistema
- 6- Operación y mantenimiento

Para utilizar este modelo se debe tener una muy buena disciplina al momento de la planeación y administración de las fases, debido a que si se presentan fallas en alguna de ellas provocaría un retraso en el desarrollo, hasta la reconstrucción de una fase. Otro punto favorable es que la fase de codificación se realiza siempre y cuando se entienda bien el objetivo del proyecto (requerimientos) y hasta que se haga el diseño del sistema.

Este modelo también presenta ciertas falencias que se deben considerar al momento de implementarlo, como por ejemplo que en la realidad no se siguen adecuadamente los pasos propuesto por el modelo, es dificultoso para el cliente exponer todos los requerimientos del sistema al inicio del desarrollo del proyecto, los responsables del desarrollo del software siempre se retrasan innecesariamente, produciendo “Estados de Bloqueo”. En el que algunos miembros del proyecto deben esperar a otros miembros del equipo para completar tareas dependientes y por último, presenta dificultad en modificaciones esencialmente en la etapa de mantenimiento puesto que para realizarlo se tendrá que empezar de nuevo todo el ciclo, modificando la información de cada fase para así tener una documentación fiable y consistente.

5.1.2 Modelo construcción de prototipos

Consiste en capturar un conjunto inicial de necesidades e implementarlas rápidamente con la intención declarada de expandirlas y refinarlas iterativamente al ir aumentando la comprensión que del sistema tienen los usuarios y quien lo desarrolla. Ofrece una alternativa atractiva y practicable a los métodos de especificación para tratar mejor la incertidumbre, la ambigüedad y la volubilidad de los proyectos reales.

Dentro del enfoque de prototipos se pretende que el modelo sea operante, es decir, una colección de programas de computadora que simulan algunas o todas las funciones que el usuario desea. Para lograr lo anterior se utilizan las siguientes herramientas de software:

- Un diccionario de datos integrado
- Un generador de pantallas
- Un generador de reportes no guiado por procedimientos
- Un lenguaje de consultas no guiado por procedimientos
- Medios poderosos de administración de base de datos

Comienza con una actividad de sondeo; de esto sigue una determinación de sí el proyecto es un buen candidato. Los buenos candidatos son aquellos que tienen las siguientes características:

- El usuario no puede o no está dispuesto a examinar modelos abstractos en papel, tales como diagramas de flujo de datos.
- El usuario no puede o no está dispuesto a articular sus requerimientos de ninguna forma y sólo se pueden determinar sus requerimientos mediante un proceso de tanteo, o ensayo y error.
- Se tiene la intención de que el sistema sea en línea y con operación total por la pantalla, en contraposición con los sistemas de edición, actualización y reportes operados por lote.
- El sistema no requiere la especificación de grandes cantidades de detalles algorítmicos, ni de muchas especificaciones de procesos para describir los algoritmos con los cuales se obtienen resultados.

5.1.3 Modelo de desarrollo incremental

Los riesgos asociados con el desarrollo de sistemas largos y complejos son enormes. Una forma de reducir los riesgos es utilizar el desarrollo incremental que es un proceso de construcción siempre incrementando subconjuntos de requerimientos del sistema.

Típicamente, un documento de requerimientos es escrito al capturar todos los requerimientos para el sistema completo.

Proceso: Es un modelo interactivo, el primer incremento es un producto esencial (núcleo), se afrontan requisitos básicos y muchas funciones extras (conocidas o no) quedan para los siguientes incrementos.

El cliente usa el producto central y en base a la utilización y/o evaluación se desarrolla un plan para el incremento siguiente, dando un producto por cada incremento. Remitiéndose hasta que se elabora el producto completo. De esta manera se evitan proyectos largos y se entrega “algo de valor” a los usuarios con cierta frecuencia, permitiendo que el usuario se involucre mas con el producto que se esta desarrollando, facilitando a que el resultado sea muy positivo con lo que se obtiene un mayor retorno de la inversión.

También hay que considerar que es más fácil determinar si los requerimientos planeados para los niveles subsiguientes son correctos, permite descartar solo la última iteración en caso de que se presente algún error importante, aunque también facilita que se puedan solucionar dichos problemas antes del comienzo del siguiente incremento.

Para que este modelo trabaje de una manera efectiva se necesita de gestores experimentados, aun así es difícil de aplicar a sistemas transaccionales que tienden a ser integrados y a operar como un todo, esto hace que sea muy difícil de evaluar su coste total.

5.1.4 Modelo espiral

Ha sido desarrollado para cubrir las mejores características tanto del ciclo de vida clásico, como de la creación de prototipos, añadiendo al mismo tiempo un nuevo elemento "El análisis de riesgo".

Actividades principales:

- Planificación: determinación de objetivos, alternativas y restricciones.
- Análisis de riesgo: análisis de alternativas e identificación/resolución de riesgos.
- Ingeniería: desarrollo del producto del "siguiente nivel",
- Evaluación del cliente: Valorización de los resultados de la ingeniería.

Con este modelo el software se desarrolla en una serie de versiones incrementales como un mecanismo de reducción de riesgos, las fases que se proponen en este modelo no son fijas, es decir, dependiendo de nuestro proyecto es que deberían tomarse en cuenta las fases que nos permitan un mejor desarrollo de nuestra aplicación.

Cada paso de una fase a otra produce ajustes en el plan del proyecto con lo que nos permite darnos cuenta de cómo es el enfoque realista del desarrollo de sistemas y de software a gran escala. Si se aplica adecuadamente debe reducir los riesgos antes de que se conviertan en problemáticos.

Como todos los modelos este también posee sus contras para la aplicación del modelo, por ejemplo, Resulta difícil convencer a grandes clientes de que el enfoque evolutivo es controlable y requiere una considerable habilidad para la evaluación del riesgo y cuenta con esa habilidad para su éxito. Si un riesgo no es descubierto y gestionado surgirán problemas.

5.2 Establecimiento del modelo de ciclo de vida

Con las descripciones de los diferentes modelos de ciclo de vida descritas en el punto anteriormente debemos determinar el modelo que se ha de utilizar para el desarrollo de nuestro sistema, para lo cual se deben considerar varios aspectos con respecto al proyecto:

Criterios a considerar:

- Madurez de la aplicación (relacionado a la probabilidad que muchos requerimientos comenzarán a conocerse sólo después del uso del sistema).
- Complejidad del problema y de la solución.
- Frecuencias y magnitudes esperadas de los cambios de requerimientos.
- Financiamiento disponible, y su perfil como una función del tiempo.
- Acceso de los desarrolladores a los usuarios.
- Certeza de requerimientos conocidos.
- Tolerancia al riesgo.
- Planes y presupuestos críticos
- Grado de lentitud de construcción dentro de los planes y presupuestos.

Además de estos criterios en la siguiente tabla se muestran las características de cada uno de los modelos y los elementos requeridos para el proyecto software, lo cual facilitara la selección del modelo mas adecuado para nuestro proyecto.

Tabla 11. Selección del Modelo de Ciclo de Vida

Requerimientos	Cascada	Incremental	Prototipo	Espiral
¿Los requerimientos son fácilmente definidos y/o bien conocidos?	SI	SI	NO	NO
¿Pueden ser los requerimientos definidos temprano durante el ciclo de vida?	SI	SI	NO	NO
¿Cambiarán los requerimientos frecuentemente?	NO	NO	NO	SI
¿Hay la necesidad de demostrar los requerimientos para alcanzar la definición?	NO	NO	SI	SI
¿Hay la necesidad de demostrar la validez de concepto para el sistema?	NO	NO	SI	SI
¿Especifican los requerimientos un sistema complejo?	NO	SI	SI	SI
¿Es uno de los requerimientos la funcionalidad temprana del sistema?	NO	SI	SI	SI

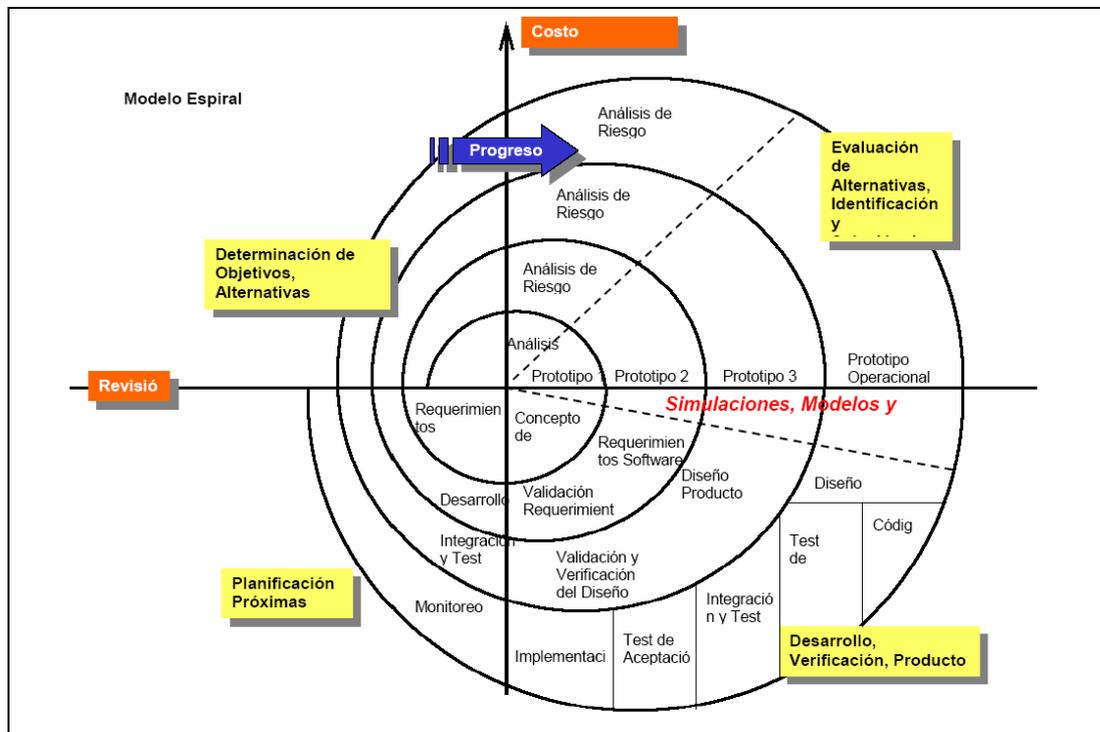
Otros Factores

¿El gerente de proyecto supervisará cercanamente el progreso del proyecto?	SI	SI	NO	SI
¿Es una prioridad la facilidad de asignación de recursos?	SI	SI	NO	NO

Considerando los puntos anteriores llegamos a la conclusión de que el modelo que nos permitiría realizar nuestro proyecto de mejor manera es el Modelo en Espiral, por su gran versatilidad, además de que en cada una de sus fases se puede incluir algún otro modelo que se ajuste perfectamente a la fase en desarrollo.

Para un mayor entendimiento del modelo a continuación ponemos a consideración una grafica que describe cada una de sus fases:

Figura 24. Modelo en Espiral.



5.3 Control de cambios

Todo sistema esta sujeto a modificaciones que pueden ser desde una simple modificación de interfaz de usuario pasando por cambios en su estructura de base de datos incluso hasta la forma en la que esta programado, estos cambios obedecen a múltiples factores tales como: facilidad de manipulación, cambios en los requerimientos de usuario, cambio en la metodología de resolución de un problema, etc.

Un factor adicional en el desarrollo de nuestro proyecto es el que al tratarse de un sistema que procurara estandarizar la labor dentro de varias facultades de la Universidad del Azuay, se debe considerar cada una de las metodologías y restricciones que manejen las mismas, con lo cual es de suma importancia mantener un control riguroso sobre los cambios que se nos sean solicitados para no poner en riesgo el perfecto funcionamiento del proyecto.

La recepción de las solicitudes de cambio se las recogerá en cada prueba que se haga del sistema con personal de la universidad, los detalles de las pruebas se explicaran de mejor manera mas adelante en este mismo capitulo, al realizarse estas pruebas conjuntamente con los desarrolladores del sistema se puede determinar el posible origen de la falla, en caso de que se trate de un error, así como de presentar una solución si se trata de alguna mejora en el sistema.

Una vez captados los cambios que se requieren, estos se discutirán entre los desarrolladores del proyecto, para medir sus impactos sobre la base de datos y cambios subsecuentes por la modificación de la misma, impacto sobre cambios en la metodología de algún proceso o procedimiento dentro de lo que se refiere a la programación, así como, el tiempo que tomara realizar dicho cambio y cualquier otro aspecto que involucre cada cambio.

En caso de que el cambio sea aceptado se diseñara una estrategia de desarrollo e implementación de dicho cambio, es decir, se determinara la manera más sencilla de corregir y poner en funcionamiento al problema para reducir sus efectos colaterales y errores que se puedan presentar.

En la siguiente prueba a la que se someta el sistema se notificara al personal de la universidad cuales han sido los cambios que se han realizado y cuales no, detallando los motivos por los que se considero un cambio o no.

5.4 Gestión de pruebas

Como se menciona anteriormente una parte muy importante para obtener un sistema que satisfaga a la mayoría de los requerimientos que presenten las diferentes facultades de la universidad es necesario ir obteniendo periódicamente los comentarios, críticas y sugerencias, etc. De las personas que se van a encontrar en constante interacción con nuestro sistema que es el personal administrativo de las diferentes facultades.

Las pruebas se las realizara desde el mismo momento en el que comience a desarrollarse el sistema, por parte de los programadores verificando su funcionamiento.

Una vez concluida cierta etapa del desarrollo del sistema, se presentara al personal administrativo de las facultades de la universidad, a la cual se le explicara el manejo y se captara sus impresiones sobre el software, esto nos permitirá mejorar el sistema utilizando la metodología establecida en el punto anterior de este capitulo. De esta manera se procura satisfacer el mayor numero de necesidades que se tengan dentro de las facultades a las cuales esta orientado nuestro software.

5.5 Conclusiones

La importancia de manejar con disciplina la programación y pruebas de la aplicación nos ha llevado a establecer el modelo de ciclo de vida, es decir, definir las etapas de la programación de nuestro proyecto, además de los lineamientos o reglas que van a ser aplicadas y respetadas por las personas que desarrollarán el software, tanto para lo que se refiere a la programación, así como para la realización de las pruebas y modificaciones del sistema.

El ciclo de vida definido para nuestro proyecto es el Modelo en Espiral, gracias a que se acopla perfectamente a nuestras expectativas de desarrollo y a las expectativas de presentación de avances y control de cambios del personal de la Universidad del Azuay.

Para cubrir estas expectativas es importante mantener una excelente comunicación entre los desarrolladores y el personal de la universidad, que permita a través de pruebas efectuadas periódicamente por ambas partes determinar los cambios o mejoras que se pueden implementar en el proyecto; luego de esto se continuará con un proceso de selección de cambios descartando los innecesarios o poco genéricos, esta selección la realizarán los desarrolladores del sistema en reuniones informales en donde se estudiarán: el impacto, beneficio, costo (tanto en tiempo como en esfuerzo) y perjuicio, si es que lo hubiera, para el sistema.

CAPITULO VI : IMPLEMENTACION

6.1 Creación de la base de datos en el servidor de la Universidad del Azuay.

Para la creación de la base de datos tuvimos la colaboración de la Ing. Janela Encalada, encargada del Centro de Computo de la Universidad del Azuay.

La base de datos en la Universidad del Azuay esta bajo Oracle 8i, nuestro software esta desarrollado en la versión siguiente Oracle 9i, en las pruebas realizadas en la fase de desarrollo pudimos darnos cuenta que existían ciertas limitaciones como es la compatibilidad entre Visual .Net y Oracle 8i, pero que este inconveniente no existía al probar Oracle 9i, por ello decidimos utilizar esta versión y acordamos con el Centro de Cómputo para que nos provean de un servidor que contenga Oracle 9i.

Este servidor esta bajo el sistema operativo Linux y la base de datos en Oracle 9i, al momento de desarrollar acordamos crear un Tablespace Asistencia donde vamos a depositar todos los datos. Un Tablespace son piezas lógicas de la base de datos que se utilizan para separar dichos datos en grupos y así simplificar su administración.

Por medidas de seguridad y para facilitar el manejo de los datos se colocaron Sinónimos y permisos solo de lectura a todas las tablas de la universidad necesarias para nuestra aplicación. Los sinónimos son objetos del sistema que apuntan a otros objetos, es decir, alias que se utilizan para esconder ciertos criterios o permisos sobre el objeto. La utilización de sinónimos es muy favorable para nuestro desarrollo ya que evitamos utilizar diferentes usuarios para acceder a la base de datos.

6.2 Establecer la conexión a la base de datos

La conexión a base de Datos dentro de Visual Basic .Net se puede realizar mediante wizards o dentro del código. Creímos más conveniente crear una clase la cual contenga la conexión y todas las variables necesarias para el acceso a la base de datos. Esta clase es heredada por todas las demás clases, y nos da la facilidad de realizar aquí cualquier cambio para las distintas pruebas que se realizaron en las fases anteriores con respecto a la cadena de conexión.

Como ya sabemos una cadena de conexión es una expresión con la cual podemos acceder a la base de datos y poder manipularla de ahí su importancia referente en compatibilidad, seguridad, etc. En Visual Basic mediante ADOs conseguimos ver la base de datos.

Dentro de la Clase ClsConexión_Horario apreciamos el siguiente código:

```
Public Class ClsConexion_Horario
    Public Conexion_horario As OracleClient.OracleConnection
    Public Transaccion_horario As OracleClient.OracleTransaction
    Public Comando_horario As OracleClient.OracleCommand
    Public DataReader_horario As OracleClient.OracleDataReader

    'Procedimiento que abre la conexión
    Public Sub New()
        Conexion_horario = New OracleClient.OracleConnection(cadena_horario)
    End Sub
```

Finalmente toda la información para tener conexión con Oracle se localiza en el archivo TSNAMES.ORA, el cual se encuentra dentro de la carpeta oracle, en nuestro caso tenemos la siguiente dirección C:\oracle\ora92\network\admin\tnsnames.ora, debido a que configuramos en nuestras máquinas el Oracle como administradores. El archivo Tnsnames.ora contiene el TNS (The Transparent Network Substrate), protocolos a utilizar como TCP, nombre del host, des servidor, además permite la comunicación distribuida entre base de datos.

A continuación colocamos parte del TNSNAMES.ora la cual añadimos dentro del archivo para poder realizar las pruebas de conexión y de todo el sistema en si.

Tnsnames.ora

```

HORARIO =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = vero)
        (PORT = 1521) )
      )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = Horario) )
    )
  
```

6.3 Instalación del software.

Antes de proceder a realizar la instalación de la aplicación se debió verificar que computadores contienen las facultades de la universidad para en base a ello, tomar las medidas necesarias respecto a programas previos, controladores, drivers, etc. Según nuestras observaciones y las confirmaciones del centro de cómputo la mayoría de las computadoras que se encuentran en las facultades de la universidad tienen las siguientes características:

Tabla 12. Características computadores de cada facultad

HARDWARE	Pentium 4 entre 1.6 a 3 GHz. Memoria RAM entre 128 y 256 Disco Duro de 40 Gb. Video: Monitores con soporte a resolución 800 x6 00. Unidad de CD ROM
SOFTWARE	Microsoft Windows Xp Profesional

Estas características cumplen con los requerimientos mínimos para el buen funcionamiento de nuestro sistema, ya que nuestra aplicación precisa aquellas que detallamos a continuación:

Tabla 13. Requerimientos mínimos de hardware

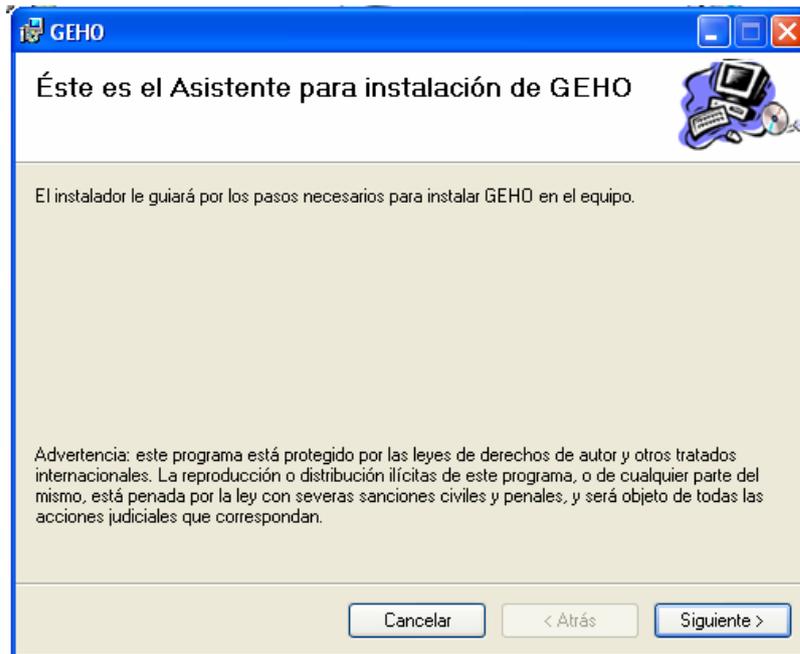
HARDWARE	Pentium 4 de 1.6 GHz Memoria RAM de 128 Disco Duro: espacio libre de 200 Mb. Video: Monitor súper VGA con soporte a resolución 800 x600 o superior. Tarjeta de Red. 10/100 Mbps.
SOFTWARE	Microsoft Windows 2000 o posterior. .NET Framework 1.1

Para poder colocar en las computadoras de cada facultad nuestro sistema realizamos un paquete de instalación el cual contiene la aplicación y ciertos elementos necesarios para su óptimo funcionamiento. Entre ellos .NET framework, plataforma necesaria para que nuestra aplicación funcione.

A continuación indicamos las pantallas que se mostrarán al momento de la instalación:

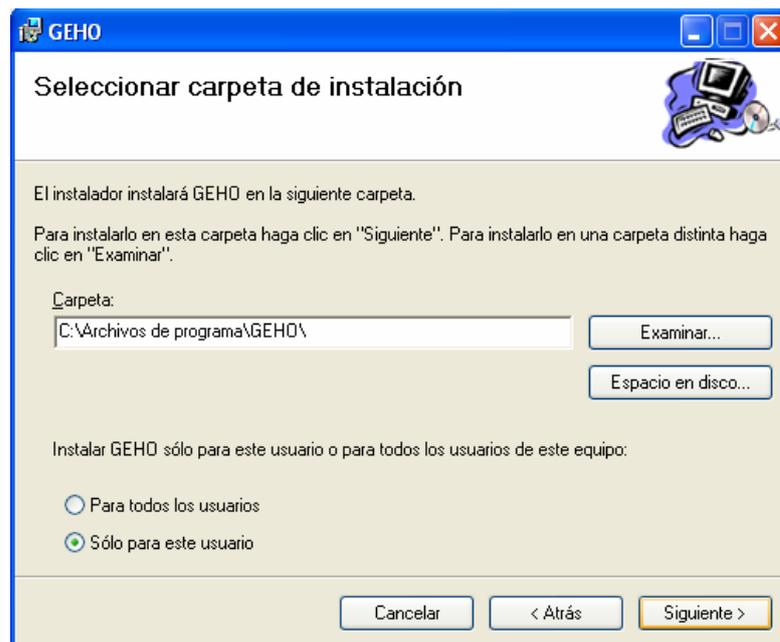
1. La primera pantalla de la instalación de GEHO:

Figura 25. Pantalla del asistente de instalación



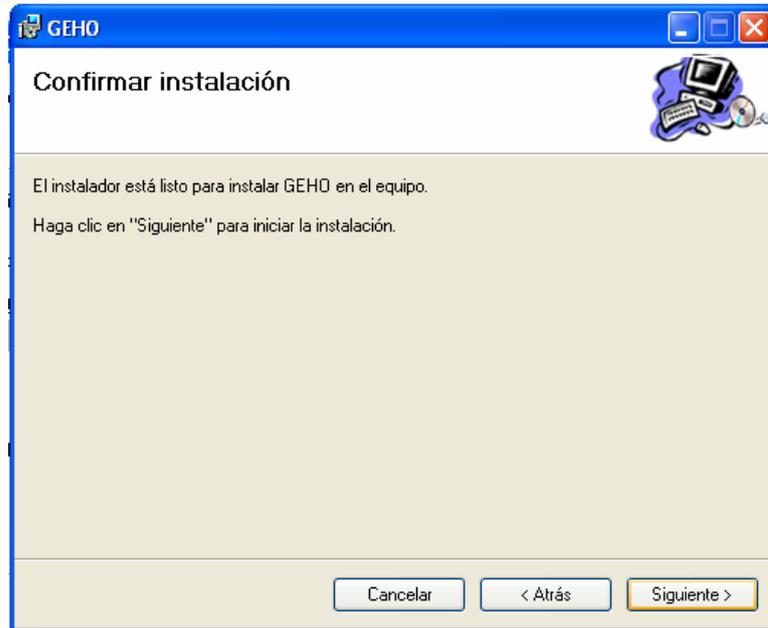
2. En esta pantalla debemos ingresar la ruta donde vamos a depositar la aplicación. Por defecto se guardará en c:\Archivos de Programa\GEHO. También es posible especificar si deseamos que se instale solo para nuestro usuario o para todo e incluso verificar el espacio que ocupa la aplicación y el espacio libre que tenemos en el disco duro.

Figura 26. Asistente que define donde instalar el programa



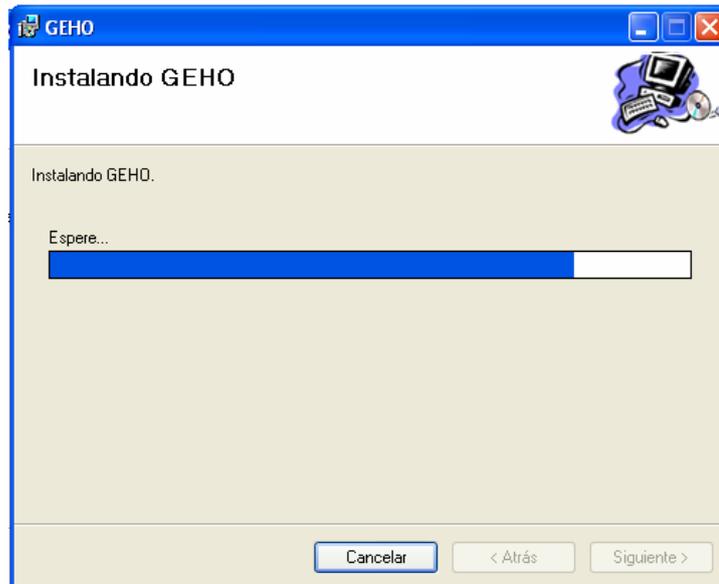
3. Ya indicada la dirección donde se guardará la aplicación procedemos a la misma instalación del sistema en la PC.

Figura 27. Pantalla de confirmación



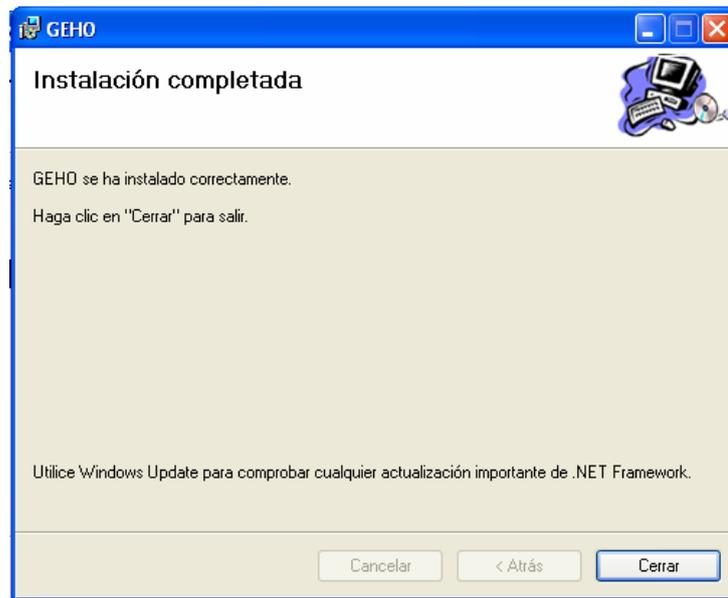
4. Mientras se instala la toda la aplicación saldrá esta pantalla indicando su avance.

Figura 28. Pantalla de instalación



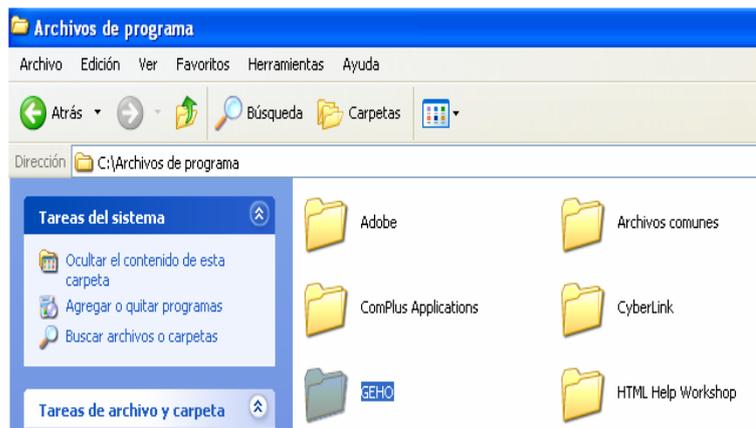
5. Y por último tenemos la pantalla de finalización una vez instalado el software

Figura 29. Características computadores de cada facultad



Podemos observar que una vez instalado GEHO se guarda por defecto en la dirección c:\Archivos de Programa.

Figura 30. Ubicación general luego de haber instalado el sistema.



6.4 Asignación de claves a los usuarios.

Hoy en día es muy importante proteger nuestra información de terceras personas, esta es muy valiosa y debe ser manipulada por el personal adecuado.

Generalmente utilizamos como claves nuestros nombres, fechas de aniversarios, fechas de nacimiento, etc., ya que son muy fáciles de recordar, pero no es recomendable ya que proveemos el acceso a personal no autorizado a nuestros datos y a manipular nuestras aplicaciones.

Pensando en este inconveniente para acceder a GEHO designamos nombres y claves de usuarios, las mismas que son creadas solo por el administrador del sistema. Dicha asignación varía los permisos y accesos a diferentes formularios dependiendo el grado de responsabilidad en el manejo de la aplicación otorgado únicamente por el usuario administrador.

Por tanto la clave que el administrador designe al usuario debe ser guardada con absoluta discreción ya que se le responsabiliza directamente de los cambios que realice. En caso que el usuario desee cambiar su clave deberá notificar al administrador del sistema para que realice el cambio. Se recomienda que periódicamente se realicen variaciones en las claves como medida de precaución.

El siguiente gráfico muestra el inicio de sesión a GEHO, un cuadro de diálogo donde debemos ingresar nuestro nombre de usuario y nuestra contraseña:

Figura 31. Inicio de Sesión



The image shows a standard Windows-style dialog box titled "Inicio de Sesión". It has a light beige background and a blue title bar. The dialog contains two text input fields. The first field is labeled "Nombre de Usuario:" and the second is labeled "Contraseña:". Below the input fields, there are two buttons: "Aceptar" and "Salir".

6.5 Conclusiones

Aprendimos en la práctica las actividades de un administrador de base de datos Oracle, conocimos limitaciones en cuanto a compatibilidad, y pequeños “tips” que nos enseñó el personal del Centro de Cómputo basados en su larga experiencia. Es así que la base de datos de nuestro sistema esta bajo Oracle 9i; para mantener la confidencialidad de la información propia de la universidad nos crearon sinónimos, para poder acceder a ellos sin manipularlos.

Las pruebas realizadas en el desarrollo del sistema nos permitieron establecer los requerimientos mínimos en hardware y software, los cuales cumplen a cabalidad todos los equipos de computación de cada facultad. En el proceso adquirimos nuevos conocimientos acerca de cómo crear un instalador con todos los programas previos a la ejecución de nuestro sistema, de esta manera crear un solo paquete con todas las necesidades de la aplicación.

La asignación de claves a usuario, establecidas por el administrador del sistema permitirá llevar mejor control en cuanto a seguridad y manejo de información. A más de administrar en caso de existir manipulación de la información, dependiendo esto de la responsabilidad del usuario.

CONCLUSIONES

El sistema propuesto, integrado por la generación de horarios y la asignación de aulas concuerda con las necesidades de las Facultades estudiadas, pues, cumple y estandariza los requerimientos de cada una de ellas.

Realizada la investigación de campo dentro de la Universidad del Azuay, se logró tener un amplio conocimiento en la forma de elaborar los horarios de clase en cada facultad, la manera que realizan la asignación de aulas y que edificios suelen ocupar, el tiempo que les lleva a cada facultad realizar esta tarea manualmente, la dificultad de coordinar y evitar cruces de horas, etc. Toda esta información adquirida nos fue de vital importancia para orientarnos a crear una propuesta adecuada a la institución.

A partir de esta investigación tuvimos un conocimiento profundo del proceso actual, y en base al diagnóstico obtenido, realizamos el sistema propuesto que optimiza tiempo y recursos.

Una vez elaborado el sistema propuesto, continuamos a la fase de diseño, la cual en base al análisis se facilitó la estandarización de pantallas, procesos, campos, muy importantes de detallarlos en esta fase para optimizar tiempo en la fase de programación. Cabe recalcar que se tomó el tiempo necesario en el diseño de la base de datos, y así evitar que se produzcan cambios en el futuro, que puedan alterar todo el desarrollo de la aplicación.

La experiencia de haber utilizado la herramienta de desarrollo Visual Basic .Net para aplicaciones Windows, y el gestor de Base de Datos Oracle, nos permitió a más de enriquecer nuestra preparación académica, cumplir roles de programadores, coordinadores, administradores de base de datos, redes, etc.; conocimientos valiosos para nuestra carrera, la cual, es muy versátil y exige nociones de muchos de estos roles aprendidos.

El desarrollo de la aplicación fue avanzando por etapas; en cada una se realizaban las pruebas respectivas, de esta manera, en el proceso fuimos

descubriendo nuevos requerimientos de los usuarios, y por medio del modelo espiral fuimos depurando las necesidades y alineando nuestro sistema a la óptima utilización del mismo. La programación orientada a objetos fue de mucha utilidad en cuanto a la organización, optimización de tiempo al momento de realizar cambios y reutilización.

Una vez terminado nuestro sistema realizamos pruebas en las facultades para estar seguros que nuestra aplicación era amigable para el usuario y fácil de utilizar. Los resultados fueron satisfactorios ya que estuvieron de acuerdo en el fácil manejo de la aplicación y sobretodo en la información que manipula. Los reportes que muestra el sistema los consideran precisos, fáciles de entender y útiles para todo el personal docente y estudiantes.

La aplicación fue probada en casos favorables, es decir, ingresando los datos básicos para que el sistema funcione. Permite cambios luego de realizar la generación, además de tener seguridades para la consistencia de los datos. Cabe destacar que para su óptimo funcionamiento se debe ingresar toda la información requerida de los profesores, distributivos de trabajo y aulas, puesto que, en base a la información ingresada al sistema nuestra aplicación trabajará. Si se agrega luego u omiten información significativa antes de realizar la generación de horarios, esta no va ser totalmente práctica.

Nuestro sistema está creado con parámetros, con los cuales es posible adaptarlo a otra institución que requiera la generación de horarios y asignación de aulas como universidades, escuelas y colegios, en sí unidades educativas donde utilicen las variables de docentes, aulas y materias.

RECOMENDACIONES

Luego de haber realizado las pruebas respectivas del sistema, hemos llegado a la conclusión de introducir ciertos lineamientos que deberán cumplir todo el personal que utilice o manipule el sistema para obtener los resultados deseados, estos parámetros son:

Se deberá ingresar toda la información de las materias que van a estar vigente durante el periodo para el cual se van a generar los horarios.

Se deberá establecer que profesores van a dictar que materia o materias, así como, el número de alumnos y número de paralelos de cada una de ellas, en caso de ser necesarios en que aula se va a dictar la materia, fecha de inicio y fecha de terminación de la materia, etc.

El horario de disponibilidad del profesor es de suma importancia para la generación de los horarios, por lo cual se requiere que el número de horas disponibles del profesor corresponda al 150% del total del número de horas que necesita para dictar todas sus cátedras, es decir, por ejemplo si un profesor dicta tres cátedras, las dos primeras de cuatro y la tercera de dos horas semanales, el profesor necesitara de diez horas para cumplir con las tres materias semanalmente. Nuestro sistema requeriría que se ingrese quince horas para garantizar la confiabilidad de los resultados que el sistema devuelva.

Se deben tener asignadas todas las aulas que cada una de las escuelas utiliza, así como su horario de utilización (mañana, tarde y/o noche) y la capacidad de las mismas.

Los autores del software no nos responsabilizamos de errores producidos o la poca satisfacción en los resultados que este devuelva, por la indebida manipulación de la información o de la aplicación o por la ausencia total o parcial de la información descrita anteriormente.

GLOSARIO

ALGORITMO

Un algoritmo es un conjunto finito de instrucciones o pasos que sirven para ejecutar una tarea o resolver un problema. De un modo más formal, un algoritmo es una secuencia finita de operaciones realizables, no ambiguas, cuya ejecución da una solución de un problema en un tiempo finito.

ANALISIS

El análisis de sistemas es la ciencia encargada del análisis de sistemas grandes y complejos y la interacción entre esos sistemas. Esta área se encuentra muy relacionada con la Investigación de operaciones. También se denomina análisis de sistemas a una de las etapas de construcción de un sistema informático, que consiste en relevar la información actual y proponer los rasgos generales de la solución futura.

APLICACION

En informática las aplicaciones son los programas con los cuales el usuario final interactúa, es decir, son aquellos programas que permiten la interacción entre el usuario y la computadora. Esta comunicación se lleva a cabo cuando el usuario elige entre las diferentes opciones o realiza actividades que le ofrece el programa.

ARQUITECTURA CLIENTE – SERVIDOR

La arquitectura cliente-servidor llamado modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realizada se efectúe con la mayor eficiencia, y permita simplificarlas.

ARQUITECTURA DEL SOFTWARE

Una Arquitectura Software, también denominada Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información.

ATRIBUTO

Cada una de las cualidades o propiedades de un objeto.

BASE DE DATOS

Una base o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

CREDITOS

Número de horas a la semana en las que deberá ser dictada una materia

DBA

El administrador de base de datos (DBA) es la persona responsable de los aspectos ambientales de una base de datos

DEPURAR

Un depurador, es una herramienta para depurar o limpiar de errores de algún programa informático. El depurador permite monitorizar la ejecución de un programa, asociando el código fuente con las instrucciones ejecutadas por el programa, detener la ejecución, seguir el flujo y comprobar los valores de las variables.

DISEÑO

El Diseño de sistema es el arte de definir la arquitectura de hardware y software, componentes, módulos y datos de un Sistema de cómputo para satisfacer ciertos requerimientos.

DISTRIBUTIVO

Documento el cual contiene información correspondiente las materias vigentes para el periodo de estudio, los profesores que las dictaran, numero de créditos, etc.

ESCALABILIDAD

Es la capacidad de un sistema informático de adaptarse a un número de usuarios cada vez mayor, sin perder calidad en los servicios.

GESTOR DE BASE DE DATOS

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

GUI – INTERFAZ DE USUARIO

La interfaz de usuario es la forma en que los usuarios pueden comunicarse con una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo.

HARWARE

Se denomina hardware al conjunto de elementos materiales que componen un computador. Hardware también son los componentes físicos de una computadora tales como el disco duro, dispositivo de CD-ROM, disquetera, etc. En dicho conjunto se incluyen los dispositivos electrónicos y electromecánicos, circuitos, cables, tarjetas, armarios o cajas, periféricos de todo tipo y otros elementos físicos.

LLAVE FORANEA

Cuando se tienen dos tablas o más, una clave ajena es aquella columna de una tabla que hace referencia a una clave primaria de otra tabla.

LLAVE PRINCIPAL

Una clave primaria es aquella columna (pueden ser también dos columnas o más) que identifica únicamente a esa tupla.

MODELO E-R

El 'Modelo Entidad/Relación (E/R)' es un concepto de modelado para Bases de Datos, mediante el cual se pretende 'visualizar' los objetos que pertenecen a la Base de Datos como Entidades (esto es similar al modelo de Programación

Orientada a Objetos) las cuales tienen unos atributos y se vinculan mediante Relaciones.

MULTIPLATAFORMA

Multiplataforma es un término utilizado frecuentemente en informática para indicar la capacidad o características de poder funcionar o mantener una interoperabilidad de forma similar en diferentes sistemas operativos o plataformas.

OBJETO

Un objeto es una representación detallada, concreta y particular de un "algo". Tal representación determina su identidad, su estado y su comportamiento particular en un momento dado.

PASSWORD

Cadena de caracteres que define una clave de seguridad para el acceso a una aplicación, no puede contener caracteres especiales tales como: coma (,), punto y coma (;), punto (.), etc.

PESTAÑA

Desempeñar una tarea a través de pestañas permite cargar varios elementos separados, dentro de una misma ventana. De esta manera, es posible alternar entre ellos (entre los elementos) con una mayor comodidad.

PROCESO

Un proceso es un concepto manejado por el sistema operativo que consiste en el conjunto formado por: instrucciones de un programa destinadas a ser ejecutadas por el microprocesador. Su estado de ejecución en un momento dado, esto es, los valores de los registros de la CPU para dicho programa. Su memoria de trabajo, es decir, la memoria que ha reservado y sus contenidos.

PROGRAMACION POR CAPAS

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un

ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

REGISTRO

En informática, y concretamente en el contexto de una base de datos relacional, un registro (también llamado fila o tupla) representa un ítem único de datos implícitamente estructurados en una tabla.

RESOLUCION

La resolución de pantalla es el número de píxeles (o máxima resolución de imagen) que puede ser mostrada en la pantalla. Viene dada por el producto de las columnas ("X"), el cual se coloca al principio y el número de filas ("Y") con el que se obtiene una razón.

SOFTWARE

Se denomina software a todos los componentes intangibles de un ordenador, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica.

TABLA

Tabla en las bases de datos, se refiere al tipo de modelamiento de datos, en que se guarda en el los datos recolectados por un programa. Su estructura general se asemeja a la vista general de un programa de Hoja de cálculo.

TRANSACCION

Una transacción en SQL es un conjunto de ordenes hacia el DBMS que se ejecutan en una unidad de trabajo, es decir en forma atómica

UML

Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está apoyado en gran manera por el OMG (Object Management Group).

BIBLIOGRAFIA

PETROUTSOS. Evangelos & RIDGEWAY. Mark. Visual Basic .NET Developer's Handbook. Estados Unidos. Sybex. 2003. 980. 1era Edición

RUMBAUGH. James. JACOBSON. Ivar. & BOOCH. Grady. El Lenguaje Unificado de Modelado. Manual de Referencia. España. Grafillles,S.L. 20000. 497. 1era Edición.

ROMERO. Gesvin. UML con Rational Rose. Perú. Megabyte S.A.C. 2004. 503. 1era Edición.

Microsoft Corporation. Arquitectura del software.
http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arg/arquitectura_soft.asp. 2006 [4 de mayo del 2006].

Microsoft Corporation. Arquitectura de aplicaciones de .NET: Diseño de aplicaciones y servicios.
<http://www.microsoft.com/spanish/msdn/arquitectura/default.asp>. 2006 [30 de Marzo de 2006]

Alzado. Como estructurar un equipo de interfaz de usuario.
http://www.alzado.org/articulo.php?id_art=486. 2006 [8 de abril del 2006].

CAPIS. Ciclo de Vida Orientado a Objetos según el Método de Grady Booch
<http://www.itba.edu.ar/capis/rtis/articulosdeloscuadernosetaaprevia/cao12.pdf>. 2006. [19 de mayo del 2006].

Desarrollo Web. Diccionario de Datos.
<http://www.desarrolloweb.com/fag/452.php>. 2006 [5 de abril del 2006].

Desarrollo Web. Resolución de Pantallas.
<http://www.desarrolloweb.com/articulos/1286.php>. 2006 [5 de abril del 2006]

Diseño de base de datos relacionales. Principios básicos de diseño.
<http://mit.ocw.universia.net/curso11208/11/11.208/IAP02/lecture-notes/lecture5-2.html>. 2006 [6 de abril del 2006]

Diseño Base de Datos. Principios.

http://www.programacion.com/bbdd/articulo/bbdd_diseno/. 2006 [6 de abril del 2006].

Enciclopedia en línea Wikipedia. Oracle: Concepto, evolución y aplicaciones.

<http://es.wikipedia.org/wiki/Oracle.htm>. 2005 [15 de marzo del 2006]

Enciclopedia en línea Wikipedia. Conceptos: Significados, definiciones de las palabras del glosario. <http://es.wikipedia.org/wiki/Portada>. 2005 [05 de junio del 2006]

Estructura de Base de Datos. Componentes funcionales.

http://www.itlp.edu.mx/publica/tutoriales/basedat1/tema1_12.htm. 2006 [4 de febrero del 2006]

Gestión de Proyectos. Ciclos de vida.

<http://www.getec.etsit.upm.es/docencia/qproyectos/planificacion/cvida.htm>. 2006 [19 mayo del 2006].

Misco. Herramientas de desarrollo.

<http://www.misco.es/categories/~102103~/Herramientas%20de%20Desarrollo.htm>. 2006. [20 de mayo del 2006].

Modelo Espiral. <http://www.itlp.edu.mx/publica/tutoriales/analisis/24.htm>. 2006.

[15 de mayo del 2006].

Red científica. Definición de Oracle. Gestor de Base de Datos.

<http://www.redcientifica.com/oracle/c0007p0002.html>. 2005 [15 de marzo de 2006]

<http://nutabe.udea.edu.co/~andrear/funinfo2/guia/bajamedia.htm>, 30/7/2005, 14:00.

UML- Diagramas Casos de Uso.

<http://ccc.inaoep.mx/~dtapia/francisco/UML%20--%20Diagramas%20de%20Casos%20de%20Uso.htm>. 2006. [17 de marzo del 2006]

UML- Diagrama de Secuencia. [http://www-](http://www-gris.det.uvigo.es/~avilas/UML/node42.html)

[gris.det.uvigo.es/~avilas/UML/node42.html](http://www-gris.det.uvigo.es/~avilas/UML/node42.html). 2006 [17 de marzo del 2006]

Universidad del Azuay. Mapa físico: Ubicación de cada bloque y facultad.

<http://www.uazuay.edu.ec/campus/mapa.htm>, 2005 [30 de julio del 2005]

Universidad del Azuay. Reglamentos: Reglamento interno de la institución para cada facultad

<http://www.uazuay.edu.ec/documentos/Reglamento%20de%20facultades.pdf>, 2005 [30 de julio 2005].

UNSA. Ingeniería de software, ciclos de vida.

<http://www.unsa.edu.ar/~japeralt/06PAR/ppframe.htm>. 2006 [20 de mayo del 2006].

GREENWALD, Rick & Stackowiak, Robert. Professional Oracle Programming. Estados Unidos. Wiley Publishing Inc. 2005. 758. 1era Edición

ANEXOS

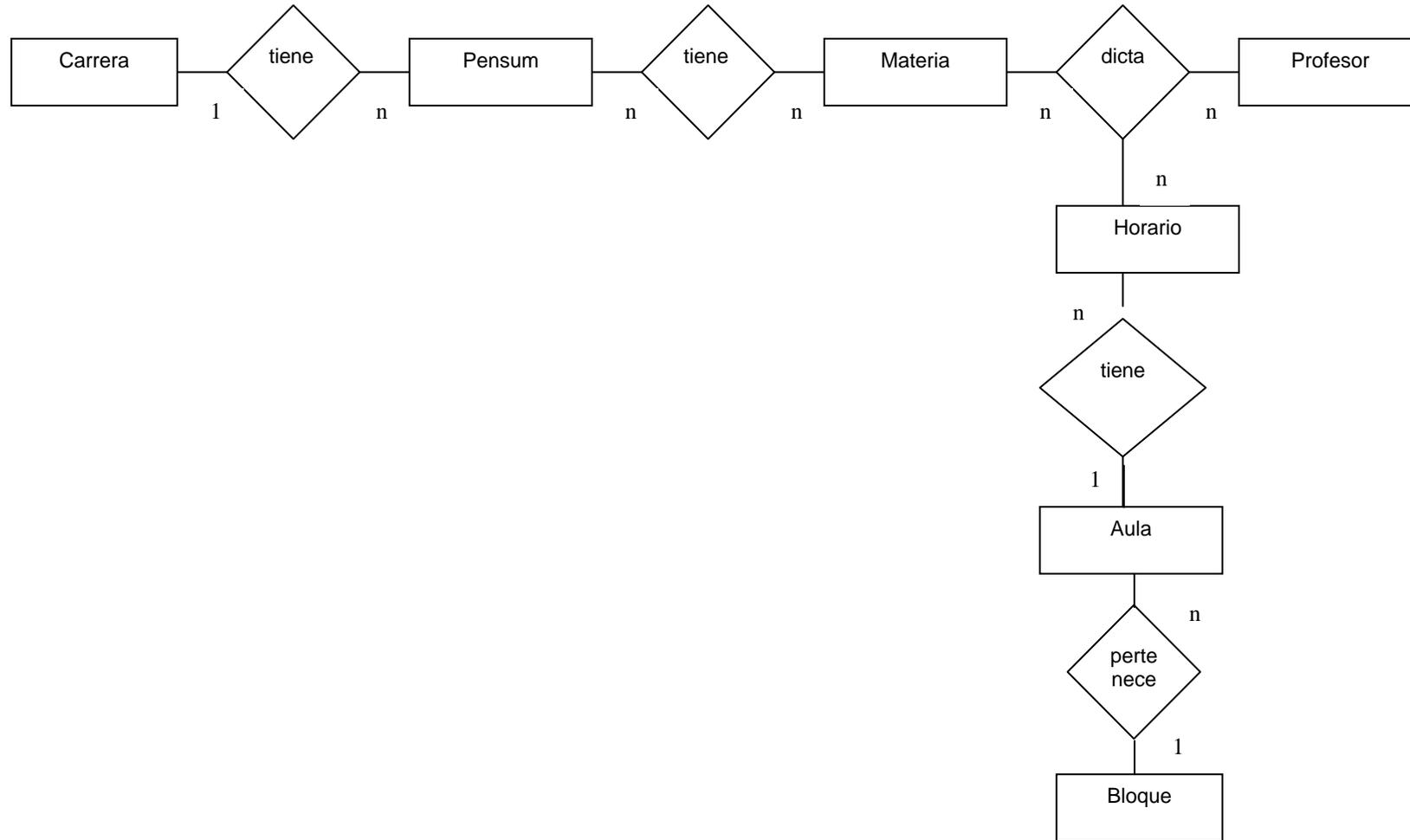
Anexo 1: Ficha de aspiración de horario

UNIVERSIDAD DEL AZUAY
FACULTAD DE FILOSOFIA, LETRAS Y CIENCIAS DE LA EDUCACION
ASPIRACION DE HORARIO DEL PROFESOR

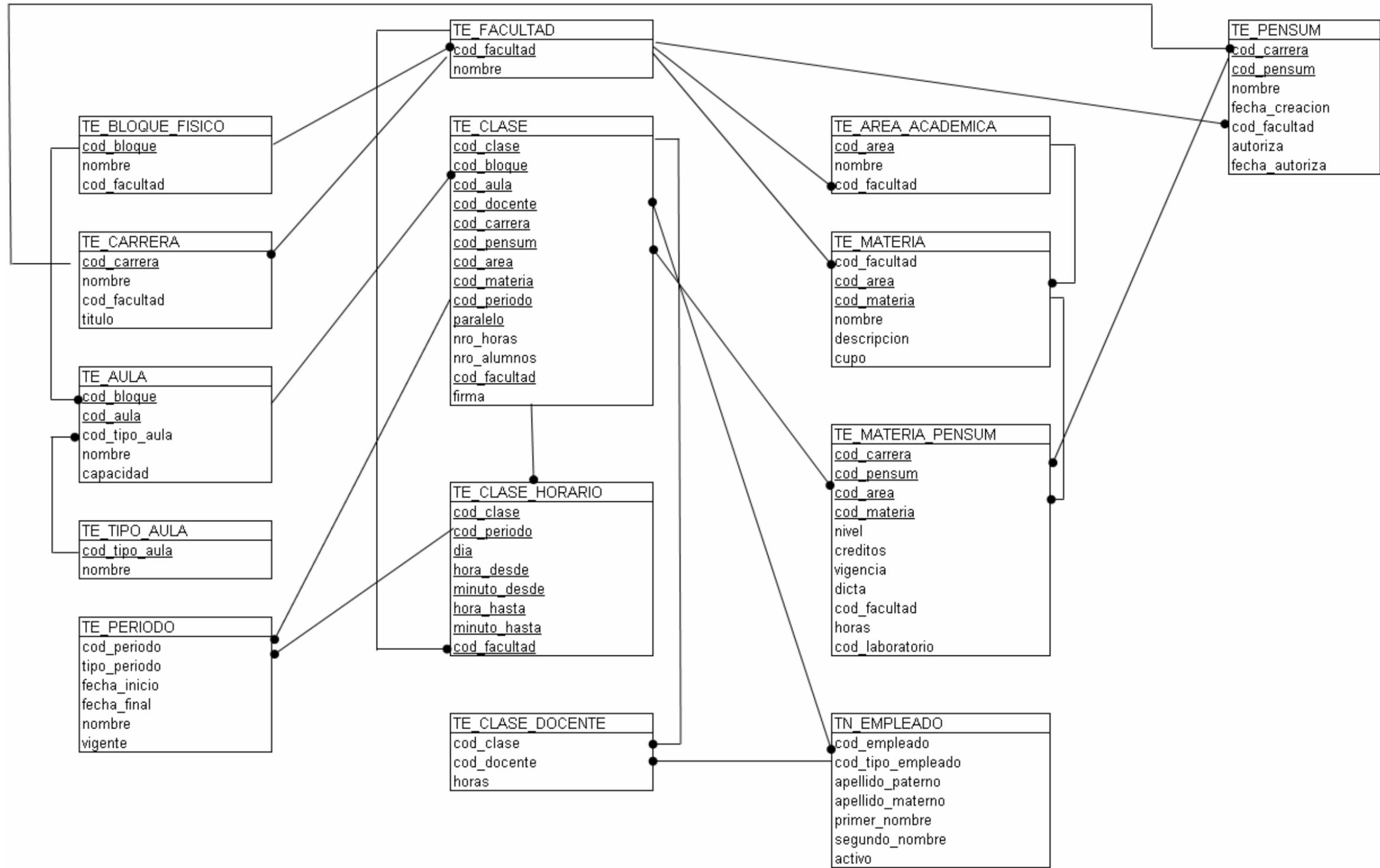
NOMBRE: _____

HORA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES
07:00 - 08:00					
08:00 - 09:00					
09:00 - 10:00					
10:00 - 11:00					
11:00 - 12:00					
12:00 - 13:00					
14:00 - 15:00					
15:00 - 16:00					
16:00 - 17:00					
17:00 - 18:00					
18:00 - 19:00					
19:00 - 20:00					
20:00 - 21:00					
21:00 - 22:00					

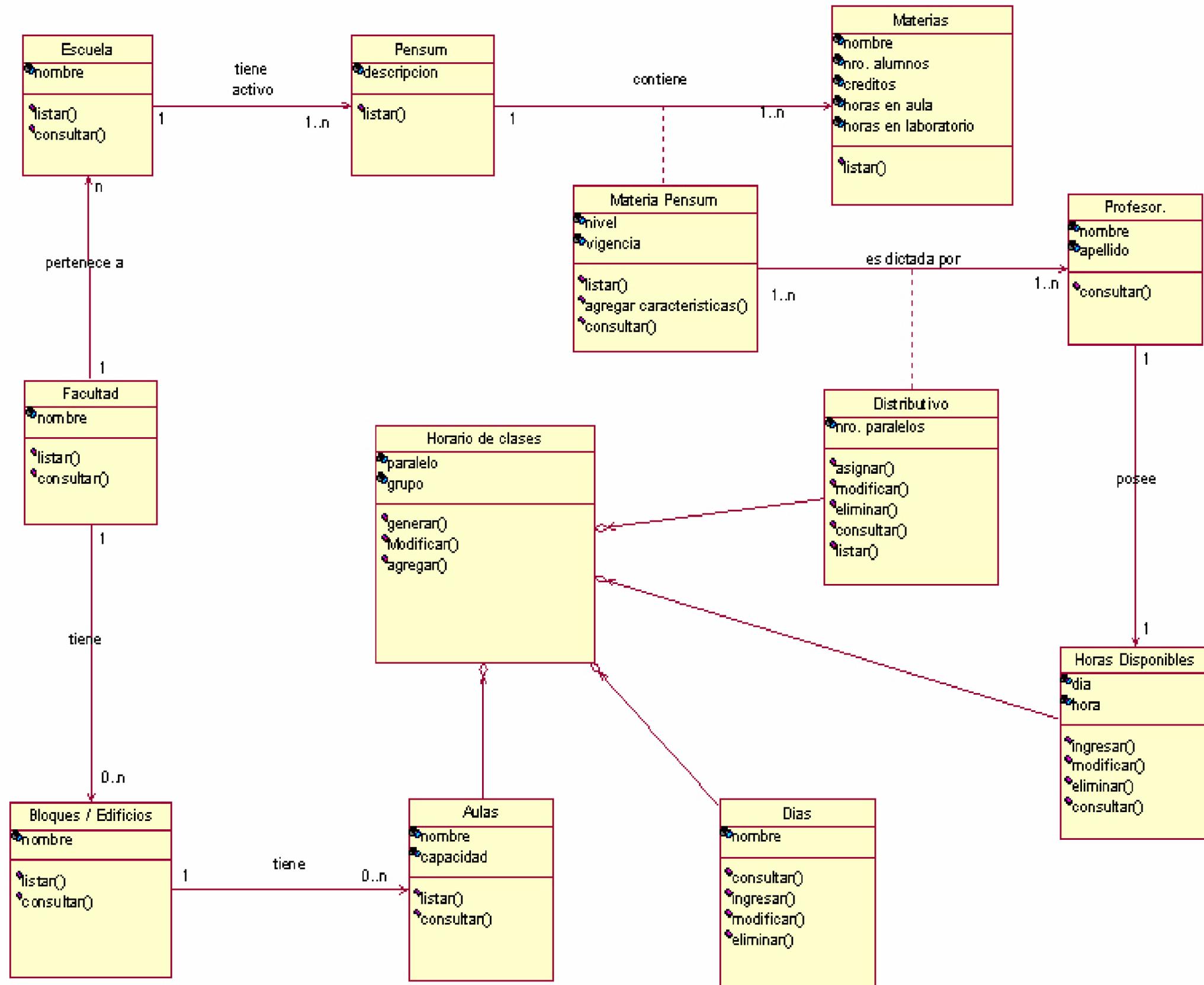
Anexo 2: Modelo Entidad Relación del sistema actual



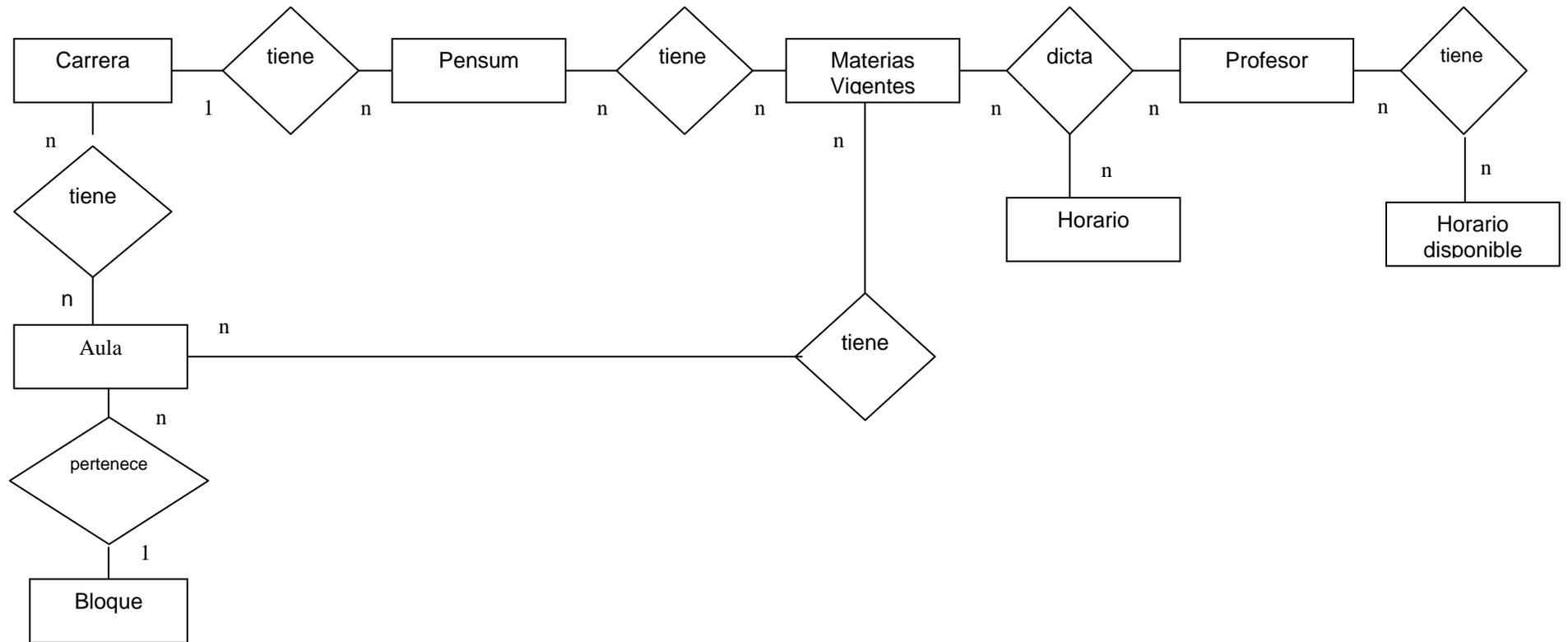
Anexo 3: Estructura de la base de datos del sistema actual



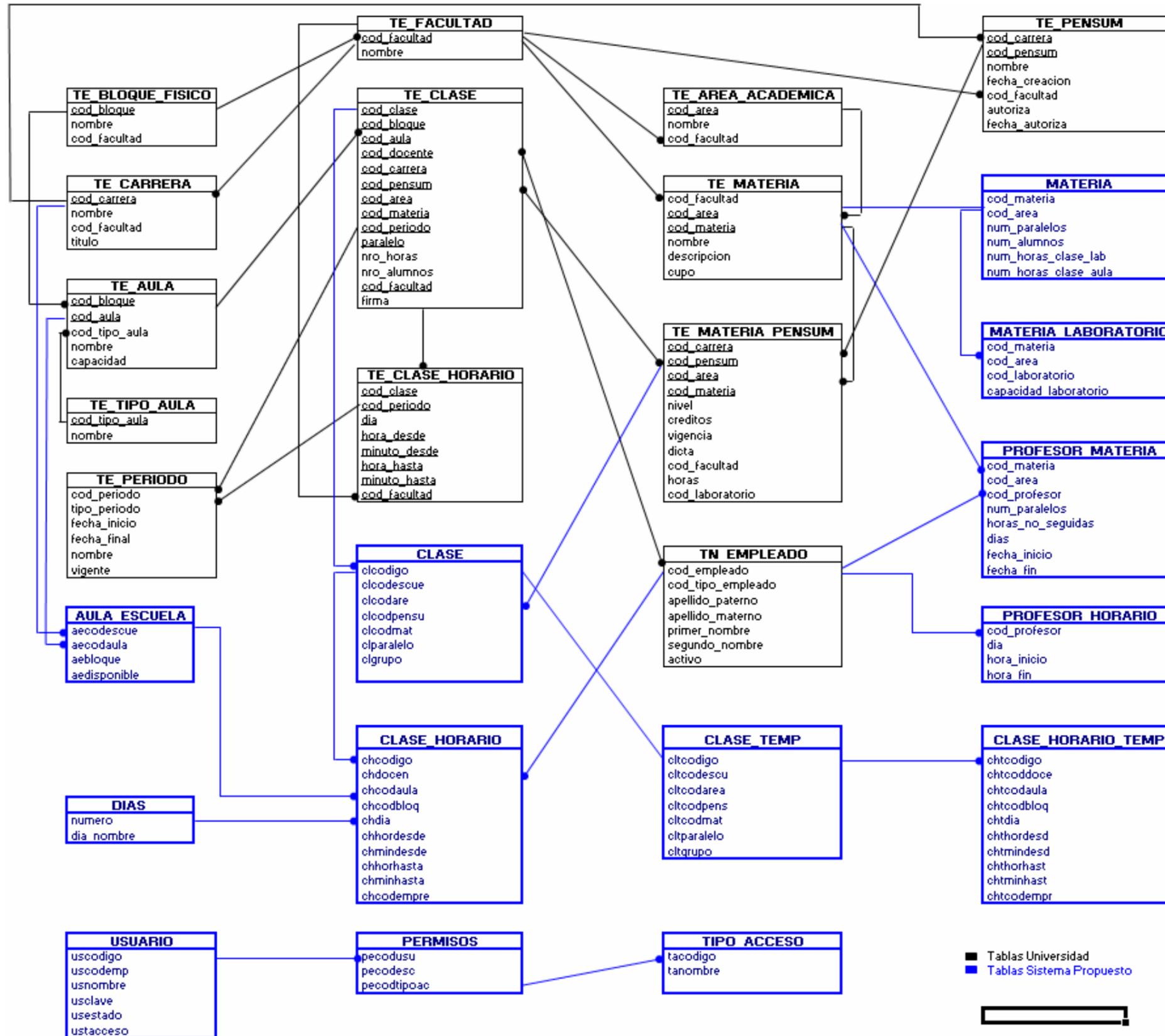
Anexo 4: Diagrama de Clases sistema propuesto



Anexo 5: Modelo Entidad – Relación del sistema propuesto



Anexo 6: Estructura de la base de datos sistema propuesto



Anexo 7: Diccionario de datos

TABLA	CAMPO	DESCRIPCION	LP	LF	RELACION
MATERIA	macodigo	Código de la materia	SI	SI	TE_MATERIA
MATERIA	macodarea	Código del área académica	SI	SI	TE_MATERIA
MATERIA	manumparal	Número de paralelos en que se dictara la materia			
MATERIA	manualumn	Número de alumnos por paralelo (estimado)			
MATERIA	manuhoclla	Numero de horas dictadas en laboratorio			
MATERIA	manuhoclau	Número de horas dictadas en aula de clases			
MATERIA_LABORATORIO	macodigo	Código de la materia que necesita de laboratorio	SI	SI	TE_MATERIA
MATERIA_LABORATORIO	macodiarea	Código área académica	SI	SI	TE_MATERIA
MATERIA_LABORATORIO	macodlabob	Código del laboratorio donde se dictara la materia	NO	SI	TE_AULA
MATERIA_LABORATORIO	macapalabo	Capacidad de la materia en dicho laboratorio			
PROFESOR_MATERIA	pmprofesor	Código del profesor	SI	SI	TN_EMPLEADO
PROFESOR_MATERIA	pmmateria	Código de la materia	SI	SI	TE_MATERIA
PROFESOR_MATERIA	pmcodarea	Código del área académica	SI	SI	TE_MATERIA
PROFESOR_MATERIA	pmnumparal	Núm paralelos que el profesor esta dispuesto a dar esta materia			
PROFESOR_MATERIA	pmhoranose	Boleano si NO dar 2 o + horas seguidas			
PROFESOR_MATERIA	pmdias	Días en que se dictará la materia			
PROFESOR_MATERIA	pmfechaini	Fecha de inicio en la que dictará la materia			
PROFESOR_MATERIA	pmfechafin	Fecha en que termina de dictar la materia			
PROFESOR_MATERIA	pmempresa	Código de empresa del profesor	NO	SI	TN_EMPLEADO
PROFESOR_MATERIA	pmcodescue	Código de la escuela a la que pertenece la materia	NO	SI	TE_CARRERA
PROFESOR_MATERIA	pmcodpensu	Código del pensum al que pertenece la materia	NO	SI	TE_PENSUM
PROFESOR_HORARIO	phprofesor	Código del profesor	SI	SI	TN_EMPLEADO
PROFESOR_HORARIO	phdia	día de la semana	SI	NO	
PROFESOR_HORARIO	phhorainic	hora disponible para comenzar a dictar clase	SI	NO	

TABLA	CAMPO	DESCRIPCION	LP	LF	RELACION
PROFESOR_HORARIO	phorafin	hora disponible en la que finaliza dictar clase	SI	NO	
PROFESOR_HORARIO	phempresa	Código de la empresa del profesor	NO	SI	TN_EMPLEADO
AULA_ESCUELA	aecodescue	Código de la escuela	SI	SI	TE_CARRERA
AULA_ESCUELA	aecodaula	código del aula	SI	SI	TE_AULA
AULA_ESCUELA	aecodbloque	Código del edificio	SI	SI	TE_AULA
AULA_ESCUELA	aedisponib	Indica que el aula esta disponible para esa escuela en horario que se especifique	SI	NO	
DIAS	numero	Número del día	SI	NO	
DIAS	dia_nombre	Nombre del día			
CLASE	clcodigo	Indica el número del horario (secuencial para identificar el horario)	SI	NO	
CLASE	clcodescue	Código de la escuela	NO	SI	TE_CARRERA
CLASE	clcodpensu	Código del pensum	NO	SI	TE_PENSUM
CLASE	clcodmat	Código de la materia	NO	SI	TE_MATERIA
CLASE	clcodarea	Código del área académica	NO	SI	TE_AREA_ACADEMICA
CLASE	clparalelo	Paralelo al cual corresponde el horario			
CLASE	clgrupo	Número del grupo al cual corresponde el horario			
CLASE_HORARI	chcodigo	Indica el número del horario (secuencial para identificar el horario)	SI	SI	CLASE
CLASE_HORARI	chcoddocen	Código del profesor que dictara la materia	SI	SI	TN_EMPLEADO
CLASE_HORARI	chcodempre	Código de empresa del profesor	SI	SI	TN_EMPLEADO
CLASE_HORARI	chcodaula	Código del aula en la que se dictara la materia	SI	SI	TE_AULA
CLASE_HORARI	chcodbloq	Código del bloque al cual pertenece el aula asignada	SI	SI	TE_AULA
CLASE_HORARI	chdia	Número del día en el que se dictara la materia	SI	SI	DIAS
CLASE_HORARI	chhordesde	Hora en la que tendrá que iniciarse las clases	SI	NO	
CLASE_HORARI	chmindesde	Complemento de hora_desde (minuto en el que tendrá que iniciarse clases)	SI	NO	

TABLA	CAMPO	DESCRIPCION	LP	LF	RELACION
CLASE_HORARI	chhorhasta	Hora en la que tendrá que finalizar las clases	SI	NO	
CLASE_HORARI	chminhasta	Complemento de hora_hasta (minuto en el que tendrá que finalizar las clases)	SI	NO	
CLASE_TEMP	cltcodigo	Indica el número del horario (secuencial para identificar el horario)	SI		
CLASE_TEMP	cltcodescu	Código de la escuela		SI	TE_CARRERA
CLASE_TEMP	cltcodpens	Código del pensum		SI	TE_PENSUM
CLASE_TEMP	cltcodmat	Código de la materia		SI	TE_MATERIA
CLASE_TEMP	cltcodarea	Código del área académica		SI	TE_AREA_ACADEMICA
CLASE_TEMP	cltparalelo	Paralelo al cual corresponde el horario			
CLASE_TEMP	cltgrupo	Número del grupo al cual corresponde el horario			
CLASE_H_TEMP	chtcodigo	Indica el número del horario (secuencial para identificar el horario)	SI	SI	CLASE
CLASE_H_TEMP	chtcoddoce	Código del profesor que dictara la materia	SI	SI	TN_EMPLEADO
CLASE_H_TEMP	chtcodempr	Código de la empresa en la que trabaja el profesor.	SI	SI	TN_EMPLEADO
CLASE_H_TEMP	chtcodaula	Código del aula en la que se dictara la materia	SI	SI	TE_AULA
CLASE_H_TEMP	chtcodbloq	Código del bloque al cual pertenece el aula asignada	SI	SI	TE_AULA
CLASE_H_TEMP	chtdia	Número del día en el que se dictara la materia	SI	SI	DIAS
CLASE_H_TEMP	chthordesd	Hora en la que tendrá que iniciarse las clases	SI		
CLASE_H_TEMP	chtmindesd	Complemento de hora_desde (minuto en el que tendrá que iniciarse clases)	SI		
CLASE_H_TEMP	chthorhast	Hora en la que tendrá que finalizar las clases	SI		
CLASE_H_TEMP	chtminhast	Complemento de hora_hasta (minuto en el que tendrá que finalizar las clases)	SI		
PERMISOS	pecodusu	código del usuario	SI	SI	USUARIO
PERMISOS	pecodesc	código de la escuela a la cual tiene acceso		SI	TE_CARRERA
PERMISOS	pecodtipoac	Tipo de acceso		SI	TIPO_ACCESO
TIPO_ACCESO	tacodigo	código del tipo de acceso	SI		

TABLA	CAMPO	DESCRIPCION	LP	LF	RELACION
TIPO_ACCESO	tanombre	Nombre del acceso			
USUARIO	uscodigo	Código del usuario	SI		
USUARIO	uscodemp	Código de la empresa			
USUARIO	usnombre	Nombre del usuario			
USUARIO	usclave	Clave del usuario para acceder al sistema			
USUARIO	usestado	Si esta vigente o no el usuario			
USUARIO	ustacceso	Tipo de acceso del usuario		SI	TIPO_ACCESO

Anexo 8: Descripción de las herramientas de desarrollo

Visual Basic .NET

Es un lenguaje de programación desarrollado por Microsoft Corporation, además es el sucesor del tan popular Visual Basic 6.0 incluido en el paquete Visual Studio, esta última tecnología .NET presenta mejoras sustanciales en todas sus características y ambientes como se describen a continuación:

En la página de inicio se cuenta con la posibilidad de con un clic acceder a información acerca de los proyectos usados recientemente, las preferencias personales, las actualizaciones de productos y la comunidad MSDN online.

Una característica muy importante de Visual Basic .NET es que cuenta con ayuda dinámica a la que se puede acceder con solo dar un clic o pulsando la tecla F1, independientemente de la tarea que se está realizando. MSDN Online Access proporciona vínculos directos a ejemplos, grupos de noticias, actualizaciones y descargas de Visual Basic .NET en el entorno de desarrollo integrado (IDE).

Permite una conexión segura y confiable con Bases de Datos Oracle, además que brinda una fiabilidad y estabilidad en aplicaciones distribuidas.

Una de las características a la que se le va a sacar el mejor provecho es la de la programación orientada a objetos, ya que como se describió anteriormente es la metodología de programación con la que se va a desarrollar nuestra aplicación.

Permite una gran reutilización de código por lo que le hace una herramienta muy útil para el desarrollo de la aplicación.

Una ventaja ajena a las características propias de este lenguaje es que gracias al desarrollo de aplicaciones durante nuestra vida estudiantil dentro de la universidad con la ayuda de esta herramienta, nos provee de un nivel de conocimiento aceptable sobre el manejo de este lenguaje de programación.

Java

Es un lenguaje de programación diseñado a partir de los lenguajes C y C++ pero con unas mejoras significativas, gracias a esta semejanza con estos lenguajes resulta un poco más sencillo su aprendizaje.

Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- aritmética de punteros
- no existen referencias
- registros (struct)
- definición de tipos (typedef)
- macros (#define)
- necesidad de liberar memoria (free)

Una de las prestaciones más significativas de Java es su facilidad para realizar una programación orientada a objetos, ya que Java trabaja con sus datos como objetos y con interfaces a esos objetos.

Las conexiones con Bases de Datos Oracle son fiables y eficientes.

Permite el desarrollo de aplicaciones distribuidas seguras.

Las dificultades que se nos presenta con este lenguaje de programación son nuestro limitado conocimiento sobre la herramienta y la nula experiencia desarrollando aplicaciones de este tipo.

Anexo 9: Selección del software de desarrollo

Opción 1: Java

Opción 2: Visual Basic.Net

Factor	Peso	Opción 1		Opción 2	
		Puntos	Total	Puntos	Total
Conocimiento	0,15	3	0,45	9	1,35
Experiencia	0,10	3	0,30	8	0,80
Facilidad de Manejo	0,08	8	0,64	9	0,72
Compatibilidad con BD Oracle	0,13	10	1,30	10	1,30
Implementación de POO	0,10	10	1,00	8	0,80
Costo de Licencia	0,10	1	0,10	1	0,10
Disponibilidad de documentación	0,10	9	0,90	8	0,80
Disponibilidad de asesoría	0,06	6	0,36	6	0,36
Soporta aplicaciones distribuidas	0,08	10	0,80	8	0,64
Reutilización de código	0,10	5	0,50	7	0,70
Total	1,00		6,35		7,57

Anexo 10: Manual del programador

Clase: ClsAula

Atributos

Nombre	Tipo	Descripción
AtrCodigo	carácter	Código del Aula
AtrBloque	entero	Código del Bloque al que pertenece el aula
AtrTipoAula	carácter	Código del tipo de aula
AtrNombre	carácter	Identificación del aula
AtrCapacidad	entero	Capacidad del aula

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Listar	DataSet	Devuelve un listado general de las aulas con sus características principales	Ninguno		
Listado_Aula	DataReader	Devuelve un listado general de las aulas con sus características adicionales	Ninguno		

Clase: ClsAulaEscuela

Atributos

Nombre	Tipo	Descripción
AtrCodigoEscuela	carácter	Código de la carrera / escuela
AtrCodigoAula	carácter	Código del aula
AtrCodBloque	entero	Código del bloque
AtrDisponible	entero	Atributo que indica si esta ocupada el aula o no

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
GuardarAsignacion	boolean	Verifica si se ha ingresado correctamente la información en el inicio de sesión, devuelve falso en caso de existir algún error al guardar dentro de la base de datos en la tabla aula_escuela.	AtrCodigoEscuela	carácter	Código de la carrera / escuela
			AtrCodigoAula	carácter	Código del aula
			AtrCodBloque	entero	Código del bloque
			AtrDisponible	entero	Atributo que indica si esta ocupada el aula o no
Eliminar	boolean	Elimina de la base de datos toda asignación de una escuela en especial	AtrCodigoEscuela	carácter	Código de la carrera / escuela

ListarAulaEscuela	DataSet	Mediante una consulta muestra todas las aulas que existen en la universidad	Ninguno		
Llenar_fg	DataSet	Función que se utiliza para tomar todas las aulas asignadas a una escuela	Ninguno		
AulaUnicas	DataSet	Función que verifica si ya existe asignada el aula a una escuela	Ninguno		
NumAulasRepetidas	DataSet	verifica si el aula ya existen en la tabla de aula_escuela	Ninguno		
Listado_Laboratorios_x_Escuela	datareader	Muestra las aulas o laboratorios por escuela	AtrCodigoEscuela	carácter	Código de la carrera / escuela
			AtrDisponible	entero	Atributo que indica si esta ocupada el aula o no

Clase: ClsBloque

Atributos

Nombre	Tipo	Descripción
AtrCodigo	entero	Código del Bloque
AtrNombre	carácter	Nombre del Bloque
AtrFacultad	carácter	Código de la facultad a la que se le asigno el bloque

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Listado_Aula	DataReader	Devuelve un listado general de los Bloques con sus características principales	Ninguno		
NombreBloque	Cadena	Obtiene el nombre de un bloque en específico	Ninguno		

Clase: ClsCarrera

Atributos

Nombre	Tipo	Descripción
AtrCodigo	carácter	Código de la Carrera
AtrNombre	carácter	Nombre de la Carrera
AtrFacultad	carácter	Código de la facultad a la que pertenece la carrera

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
BusquedaIncrementalNombre	DataSet	Devuelve un listado de las carreras cuando se realiza búsqueda incremental	Ninguno		
Listado_Carrera_Facultad	DataReader	Listado general de Carreras	Ninguno		
Listado_CarrerasXFacultad	DataReader	Obtiene un listado de carreras por facultad	Ninguno		

Clase: ClsConexion_Horario

Atributos

Nombre	Tipo	Descripción
Conexión_horario	oracle conection	Atributo de conexión propio de oracle
Transaccion_horario	oracle transaction	Atributo para realizar la transacción en oracle
Comando_Horario	oraclecommand	Atributo que ejecuta comando en oracle
Datareader_Horario	oracledatareader	Atributo datareader para oracle

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Sub New()		Procedimiento que abre la conexión	conexión_horario	oracle conection	Atributo de conexión propio de oracle

Clase: ClsFacultad

Atributos

Nombre	Tipo	Descripción
AtrCodigo	carácter	Código de la Facultad
AtrNombre	carácter	Nombre de la Facultad
AtrMateria	carácter	Código de la Materia que pertenece a la facultad
AtrArea	carácter	Área Académica de la materia
AtrEscuela	carácter	Escuela a la que pertenece la materia

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Listado_facultades	DataReader	Listado general de Facultades que tengan pensum con materias	Ninguno		
Listado_facultades1	DataReader	Listado general de Facultades	Ninguno		
Obtiene_codigo_facultad	cadena	Devuelve el código de la facultad a la que pertenece una materia	Atrmateria	carácter	Código de la materia
Materias_facultad	DataReader	Obtiene un listado de las materias que pertenecen a una facultad	AtrFacultad	carácter	Código de la facultad
Listado_Facultades_Usuario	DataReader	Listado de las facultades a las que tiene acceso un usuario	AtrUsuario	entero	Código del Usuario

Clase: ClsGeneración

Atributos

Nombre	Tipo	Descripción
AtrCodigo	entero	Código de la generación de horario
AtrProfCod	entero	Código del docente
AtrMatCod	carácter	Código de la Materia
AtrAreaCod	carácter	Área Académica de la materia
AtrEsccod	carácter	Escuela a la que pertenece la materia
AtrPensum	entero	Pensum al que pertenece la materia
AtrNivel	entero	Ciclo en el que se dicta la materia
Atrdias	entero	Número de días en los que se debe dictar la materia
AtrFechaIni	fecha	Fecha desde la que se va a dictar la materia
AtrFechaFin	fecha	Fecha hasta la que se va a dictar la materia
AtrEmpresa	entero	Empresa a la que hace referencia el horario generado
AtrDia	entero	Día de la semana en la que se va a dictar la materia (1-Lunes; 7 Domingo)
AtrHoraDesde	entero	Hora de inicio de la clase
AtrHorahasta	entero	Hora de finalización de la clase
AtrCodigoAula	carácter	Código del aula en el que se va a dictar la materia
AtrCodBloque	entero	Bloque al que pertenece el aula
Atrhordesde	entero	Hora desde la que se dictara la materia. Trabaja como parámetro
Atrparalelo	entero	Paralelo al que corresponde el horario generado
Atrgrupo	entero	Número del grupo para las clases practicas de la materia

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Obtiene_paralelo	entero	Genera el siguiente paralelo de una materia para la generación	AtrEsccod	carácter	Código de la carrera
			AtrPensum	entero	Pensum al que pertenece la materia
			AtrMatCod	carácter	Código de la Materia
			AtrAreaCod	carácter	Área Académica de la materia
Hay_cruce_materias	boolean	Verifica si hay cruce con alguna materia del mismo nivel	AtrMatCod	carácter	Código de la Materia
			AtrAreaCod	carácter	Área Académica de la materia
			AtrDia	entero	Día de la semana en la que se va a dictar la materia (1-Lunes; 7 Domingo)

			AtrHoraDesde	entero	Hora de inicio de la clase
			AtrHorahasta	entero	Hora de finalización de la clase
Aula_disponible		Verifica si hay cruce con alguna materia que se dicte en la misma aula	AtrDia	entero	Día de la semana en la que se va a dictar la materia (1-Lunes; 7 Domingo)
			AtrHoraDesde	entero	Hora de inicio de la clase
	cadena		AtrCodigoAula	carácter	Código del aula en el que se va dictar la materia
			AtrCodBloque	entero	Bloque al que pertenece el aula
obtiene_codigo	entero	Genera la secuencia de registros de la generación de horarios	Ninguno		
Grabar_cabecera_horario	boolean	Inserta un registro en la tabla Clase	Ninguno		
Grabar_detalle_horario	boolean	inserta un registro en la tabla clase_horari	Ninguno		
Obtiene_Listado_pensum	datareader	Listado de los pensum que están grabados en la tabla clase	AtrEsccod	carácter	Código de la carrera
Obtiene_listado_nivel	datareader	Listado de los niveles que posee una escuela	AtrEsccod	carácter	Código de la carrera
			AtrPensum	entero	Pensum al que pertenece la materia
Obtiene_listado_paralelo	datareader	Listado general de los paralelos existentes	Ninguno		
Obtiene_distributivo			AtrEsccod	carácter	Código de la carrera
	DataSet	Obtiene el distributivo de una escuela y pensum específicos	AtrPensum	entero	Pensum al que pertenece la materia
			Atrparalelo	entero	Paralelo al que corresponde el horario generado
			AtrNivel	entero	Ciclo en el que se dicta la materia
Profesor_materia			AtrEsccod	carácter	Código de la carrera
	datareader	Obtiene la información de la manera en que un docente dicta una materia	AtrPensum	entero	Pensum al que pertenece la materia
			AtrMatCod	carácter	Código de la Materia
			AtrAreaCod	carácter	Área Académica de la materia
Eliminar_detalle_horario	boolean	Elimina toda la información de la tabla clase_horari	Ninguno		
Eliminar_cabecera_horario	boolean	Elimina toda la información de la tabla clase	Ninguno		

Clase: ClsHorario

Atributos

Nombre	Tipo	Descripción
AtrCodigo	entero	Código del horario de clase
AtrCodEscue	carácter	Código de la escuela
AtrCodPensu	entero	Código del pensum de estudio
AtrCodMat	carácter	Código de la materia
AtrCodArea	carácter	Código del área de estudio

AtrParalelo	entero	Paralelo
AtrGrupo	entero	Grupo. Utilizado en caso de haber varios grupos en un mismo paralelo
AtrCHcodigo	entero	Código del horario de clase, de la tabla clase_horari
AtrDocente	entero	Código del profesor
AtrEmpresa	entero	Código de la empresa, dentro de la institución
AtrAula	carácter	Código del aula de clase
AtrBloque	entero	Código del bloque
AtrDia	entero	Código del día de la semana
AtrHoraDesde	entero	Hora a la que comienza la clase
AtrMinDesde	entero	Minuto al que comienza la clase
AtrHoraHasta	entero	Hora a la que termina la clase
AtrMinHasta	entero	Minuto al que termina la clase

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Llenar_grilla	DataReader	Muestra el horario dependiendo la carrera, ciclo, pensum, paralelo, grupo que corresponda	AtrCodEscue	carácter	Código de la escuela
			AtrParalelo	entero	Paralelo
			AtrGrupo	entero	Grupo. Utilizado en caso de haber varios grupos en un mismo paralelo
GuardarCabeceraHorario	Boolean	Función que guarda la cabecera del horario dentro de la tabla clase	AtrCodigo	entero	Código del horario de clase
			AtrCodEscue	carácter	Código de la escuela
			AtrCodPensu	entero	Código del pensum de estudio
			AtrCodMat	carácter	Código de la materia
			AtrCodArea	carácter	Código del área de estudio
			AtrParalelo	entero	Paralelo
GuardarDetalleHorario	Boolean	Una vez guardada la cabecera se ejecuta esta función para guardas las horas y los profesores que darán cierta materia	AtrGrupo	entero	Grupo. Utilizado en caso de haber varios grupos en un mismo paralelo
			AtrCHcodigo	entero	Código del horario de clase, de la tabla clase_horari
			AtrDocente	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa, dentro de la institución
			AtrAula	carácter	Código del aula de clase
			AtrBloque	entero	Código del bloque
			AtrDia	entero	Código del día de la semana
			AtrHoraDesde	entero	Hora a la que comienza la clase
AtrMinDesde	entero	Minuto al que comienza la clase			

			AtrHoraHasta	entero	Hora a la que termina la clase
			AtrMinHasta	entero	Minuto al que termina la clase
Nombres	Boolean	Muestra los nombres del bloque, materia, profesor	AtrBloque	entero	Código del bloque
			AtrCodMat	carácter	Código de la materia
			AtrCodArea	carácter	Código del área de estudio
			AtrDocente	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa, dentro de la institución
CodigoHorario	entero	Genera un código para el nuevo horario, este es secuencial	Ninguno		
EliminaHorario	Boolean	Elimina el horario de clase , de las tablas clase y clase_horari	AtrCodigo	entero	Código del horario de clase
			AtrCHcodigo	entero	Código del horario de clase, de la tabla clase_horari
Existe	Boolean	Verifica si existe el horario de clase	AtrCodigo	entero	Código del horario de clase
			AtrCodEscue	carácter	Código de la escuela
			AtrParalelo	entero	Paralelo
			AtrGrupo	entero	Grupo. Utilizado en caso de haber varios grupos en un mismo paralelo
VerificaProfeAsignado	Boolean	Función que verifica si el profesor que desean asignar a una materia no ha sido antes asignado a otra	AtrDocente	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa, dentro de la institución
			AtrDia	entero	Código del día de la semana
			AtrHoraDesde	entero	Hora a la que comienza la clase
AulasOcupadas	DataSet	Función que verifica que no se crucen dos o más materias en una misma aula a la misma hora y día.	AtrDia	entero	Código del día de la semana
			AtrHoraDesde	entero	Hora a la que comienza la clase
			AtrBloque	entero	Código del bloque

Funciones para guardar en tablas clase_temp y clase_h_temp

EliminaHorarioTemp	Boolean	Elimina el horario de clase temporal	AtrCodigo	entero	Código del horario de clase
			AtrCHcodigo	entero	Código del horario de clase, de la tabla clase_horari
NuevoCodigo	entero	Función que genera un nuevo código secuencial al horario temporal	Ninguno		
GuardarCabeceraHorarioTemp	Boolean	Guarda la cabecera del horario preestablecido	AtrCodigo	entero	Código del horario de clase
			AtrCodEscue	carácter	Código de la escuela
			AtrCodPensu	entero	Código del pensum de estudio
			AtrCodMat	carácter	Código de la materia
			AtrCodArea	carácter	Código del área de estudio
			AtrParalelo	entero	Paralelo

			AtrGrupo	entero	Grupo. Utilizado en caso de haber varios grupos en un mismo paralelo
GuardarDetalleHorarioTemp	Boolean	Guarda el detalle del horario pre-establecido	AtrCHcodigo	entero	Código del horario de clase, de la tabla clase_horari
			AtrDocente	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa, dentro de la institución
			AtrAula	carácter	Código del aula de clase
			AtrBloque	entero	Código del bloque
			AtrDia	entero	Código del día de la semana
			AtrHoraDesde	entero	Hora a la que comienza la clase
			AtrMinDesde	entero	Minuto al que comienza la clase
			AtrHoraHasta	entero	Hora a la que termina la clase
			AtrMinHasta	entero	Minuto al que termina la clase
LlenarGrillaTemp	DataSet	Muestra los datos según el paralelo, escuela, pensum, para llenar una grilla a cual es posible visualizar	AtrCodEscue	carácter	Código de la escuela
			AtrCodPensu	entero	Código del pensum de estudio
			AtrCodMat	carácter	Código de la materia
			AtrCodArea	carácter	Código del área de estudio
			AtrDocente	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa, dentro de la institución

Clase:

ClsMateria

Atributos

Nombre	Tipo	Descripción
AtrCodigo	carácter	Código de la Materia
AtrArea	carácter	Código del área académica a la que pertenece la materia
AtrNombre	carácter	Nombre de la Materia
AtrNumParalelo	entero	Número de paralelos de la materia
AtrNumAlumnos	entero	Número de alumnos que toman la materia
AtrLaboratorio	carácter	Código del laboratorio asignado a la materia
AtrNumHorasLab	entero	Número de horas de la materia a dictarse en el laboratorio
AtrNumHorasAula	entero	Número de horas de la materia a dictarse en aulas normales
AtrCapLaboratorio	entero	Capacidad del laboratorio en caso de que se tenga asignado uno
AtrEstado	entero	Bandera para establecer eliminaciones lógicas
AtrUsuario	carácter	Código de usuario

AtrFecha	carácter	Fecha de vigencia de la materia
AtrPensum	entero	Pensum al que pertenece la materia

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Ingresar	Boolean	Inserta un registro en la tabla materia	Ninguno		
Modificar	Boolean	Actualiza un registro de la tabla materia	Ninguno		
Eliminar	Boolean	Elimina un registro de la tabla materia	Ninguno		
Propiedades_Materia	DataReader	Obtiene todas las características de una materia es específico	Ninguno		
Num_Creditos	Entero	Devuelve el número de créditos que tiene una materia	Ninguno		
Cupo	Entero	Devuelve el número de alumnos que pueden tomar una materia	Ninguno		
Num_paralelos	Entero	Devuelve el número de paralelos que van a existir de una materia	Ninguno		
Nivel_horas	DataReader	Obtiene el nivel al que pertenece una materia y el número de horas que se dan a la semana de esa materia	Ninguno		

Clase: ClsMatLab

Atributos

Nombre	Tipo	Descripción
AtrCodMateria	carácter	Código de la Materia
AtrCodLaboratorio	carácter	Código del Laboratorio
AtrCodArea	carácter	Área Académica de la materia
AtrCodBloque	entero	Bloque al que pertenece el laboratorio
AtrCapLaboratorio	entero	Capacidad del laboratorio

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Listado_laboratorios	DataReader	Listado de los laboratorios de un bloque asignados a una materia	AtrCodMateria	carácter	Código de la Materia
			AtrCodArea	carácter	Área Académica de la materia
			AtrCodBloque	entero	Bloque al que pertenece el laboratorio
Listado_Todos_Laboratorios	DataReader	Listado de los laboratorios asignados a una materia	AtrCodMateria	carácter	Código de la Materia
			AtrCodArea	carácter	Área Académica de la materia
Capacidad_Laboratorio	entero	Devuelve la capacidad de un laboratorio asignado a una materia	AtrCodMateria	carácter	Código de la Materia

			AtrCodArea	carácter	Área Académica de la materia
			AtrCodBloque	entero	Bloque al que pertenece el laboratorio
			AtrCodLaboratorio	carácter	Código del Laboratorio
Eliminar_laboratorio	boolean	Quita todos los laboratorios de un bloque asignados a una materia	AtrCodMateria	carácter	Código de la Materia
			AtrCodArea	carácter	Área Académica de la materia
Ingresar_laboratorio	boolean	Agrega un registro a la tabla mat_lab	AtrCodBloque	entero	Bloque al que pertenece el laboratorio
			ninguno		
Eliminar_Materia	boolean	Quita todos los laboratorios asignados a una materia	AtrCodMateria	carácter	Código de la Materia
			AtrCodArea	carácter	Área Académica de la materia
Listado_Laboratorios_x_materia	DataReader	Listado de todos los laboratorios asignados a una materia	AtrCodMateria	carácter	Código de la Materia
			AtrCodArea	carácter	Área Académica de la materia

Clase: ClsMigracion

Atributos

Nombre	Tipo	Descripción
AtrClase	entero	Código del horario de clase
AtrBloque	entero	Código del Bloque
AtrAula	carácter	Código del aula
AtrDocente	entero	Código del profesor
AtrCarrera	carácter	Código de la escuela
AtrPensum	entero	Código del pensum de estudio
AtrArea	carácter	Área Académica de la materia
AtrMateria	carácter	Código de la materia
AtrPeriodo	entero	Código del periodo
AtrModelo	carácter	Código del modelo (campo establecido en la tabla te_clase, de la base de datos de la universidad)
AtrGeneracion	carácter	Código de generación (campo establecido en la tabla te_clase, de la base de datos de la universidad)
AtrParalelo	carácter	Paralelo
AtrNroHoras	entero	Numero de horas que se va a dictar la clase
AtrNroAlumnos	entero	Numero de Alumnos
AtrFacultad	carácter	Código de facultad
AtrDia	entero	Código del día de la semana
AtrHoraDesde	entero	Hora en la que comienza la clase
AtrHoraHasta	entero	Hora en la que termina la clase

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Materias_Vigentes	DataSet	Verifica si están todas las materias del horario vigentes para el nuevo periodo	AtrCarrera	carácter	Código de la escuela
Verifica_Materia_en_Distributivo	Boolean	Verifica que las materias estén dentro del distributivo	AtrCarrera	carácter	Código de la escuela
			AtrMateria	carácter	Código de la materia
			AtrArea	carácter	Área Académica de la materia
Nuevo_Codigo	entero	Determina el nuevo código con el que se guardara el horario, es incremental	Ninguno		
Cabecera_Horario	DataSet	Lista todo el horario de clases según la carrera	AtrCarrera	carácter	Código de la escuela
Detalle_Horario	DataSet	Lista el detalle del horario de clases enlazado con el código del horario	AtrClase	entero	Código del horario de clase
Periodo	entero	Determina el periodo que va estar vigente para el nuevo ciclo	Ninguno		
Facultad	carácter	Obtiene la facultad en base a la carrera dentro de la tabla te_carrera	AtrCarrera	carácter	Código de la escuela
			AtrClase	entero	Código del horario de clase
Migrar_Cabecera	Boolean	MIGRA la cabecera a la tabla te_clase de la base de datos de la universidad	AtrBloque	entero	Código del Bloque
			AtrAula	carácter	Código del aula
			AtrDocente	entero	Código del profesor
			AtrCarrera	carácter	Código de la escuela
			AtrPensum	entero	Código del pensum de estudio
			AtrArea	carácter	Área Académica de la materia
			AtrMateria	carácter	Código de la materia
			AtrPeriodo	entero	Código del periodo
			AtrModelo	carácter	Código del modelo (campo establecido en la tabla te_clase, de la base de datos de la universidad)
			AtrGeneracion	carácter	Código de generación (campo establecido en la tabla te_clase, de la base de datos de la universidad)
			AtrParalelo	carácter	Paralelo
			AtrNroHoras	entero	Numero de horas que se va a dictar la clase
			AtrNroAlumnos	entero	Numero de Alumnos
Migrar_Detalle	Boolean	Migra el detalle en la tabla te_clase_horario de la base de datos	AtrFacultad	carácter	Código de facultad
			AtrClase	entero	Código del horario de clase

		de la universidad	AtrPeriodo	entero	Código del periodo
			AtrDia	entero	Código del día de la semana
			AtrHoraDesde	entero	Hora en la que comienza la clase
			AtrHoraHasta	entero	Hora en la que termina la clase
			AtrFacultad	carácter	Código de facultad
			AtrDocente	entero	Código del profesor
Migrar_Docente_Clase	Boolean	Migra a la tabla te_clase_docente de la base de datos de la universidad	AtrClase	carácter	Código de la escuela
			AtrDocente	entero	Código del profesor
			AtrNroHoras	entero	Numero de horas que se va a dictar la clase

Clase: ClsPensumMateria

Atributos

Nombre	Tipo	Descripción
AtrMateria	carácter	Código de la materia
AtrArea	carácter	Área Académica de la materia
AtrNombreMateria	carácter	Nombre de la materia
AtrCarrera	carácter	Código de la escuela
AtrPensum	entero	Código del pensum de estudio
AtrNivel	entero	Nivel o ciclo
AtrHoras	entero	Horas que debe darse por semana
AtrVigencia	carácter	Especifica si se va a dictar este ciclo o no
AtrDicta	carácter	Especifica si esta vigente o anulada la materia

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Listado_pensum_vigente	DataReader	Lista los pensums que estarán vigentes el presente ciclo	AtrCarrera	carácter	Código de la escuela
			AtrPensum	entero	Código del pensum de estudio
Listado_Materias_Vigentes	DataReader	Lista las materias del pensum que estarán vigentes el presente ciclo	AtrPensum	entero	Código del pensum de estudio
			AtrCarrera	carácter	Código de la escuela
			AtrMateria	carácter	Código de la materia
Consulta_Materias_vigentes	DataSet	Función que devuelve todas las materias vigentes para especificar el formulario frmmatvigente	AtrCarrera	carácter	Código de la escuela
			AtrPensum	entero	Código del pensum de estudio
MatVigentesXCiclo	DataSet	Lista las materias vigentes para un ciclo /nivel	AtrPensum	entero	Código del pensum de estudio
Pensum	DataSet	Lista los pensums vigentes para una carrera	AtrCarrera	carácter	Código de la escuela
NomMateria	carácter	Muestra el nombre de la materia	AtrMateria	carácter	Código de la materia
Horas	entero	Calcula el numero de horas que se dictará la clase a la semana	AtrCarrera	carácter	Código de la escuela

			AtrPensum	entero	Código del pensum de estudio
			AtrMateria	carácter	Código de la materia
			AtrArea	carácter	Área Académica de la materia
Ciclo	entero	Obtiene el ciclo /nivel de una materia	AtrCarrera	carácter	Código de la escuela
			AtrPensum	entero	Código del pensum de estudio
			AtrMateria	carácter	Código de la materia
			AtrArea	carácter	Área Académica de la materia

Clase: ClsPeriodo

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
listado_periodo	DataSet	Devuelve un listado de los periodos de clase de la tabla te_periodo	Ninguno		

Clase: ClsPermisos

Atributos

Nombre	Tipo	Descripción
AtrCodUsuario	entero	Código del Usuario
AtrUsuario	carácter	Nombre de Usuario
AtrCodCarrera	carácter	Código de la Carrera
AtrCarrera	carácter	Nombre de la Carrera
AtrCodTipoAc	entero	Código de Tipo de Acceso

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Ingreso_Permisos	Boolean	Graba en la Tabla Permisos los datos de una nueva asignación de permisos	Ninguno		
Eliminar_permisos	Boolean	Elimina los registros de la tabla permisos dependiendo del usuario	Ninguno		
Verifica_permisos	Boolean	Determina si un usuario tiene acceso a una carrera en específico	Ninguno		
Listado Permisos	DataReader	Listado de las Carreras con permisos a un Usuario en búsqueda incremental	Ninguno		

Listado_Escuelas	DataReader	Listado de las Carreras con permisos a un Usuario	Ninguno		
------------------	------------	---	---------	--	--

Clase: ClsProfeHorario

Atributos

Nombre	Tipo	Descripción
AtrCodigo	entero	Código del profesor
AtrEmpresa	entero	Código de la empresa
AtrDia	entero	Código del día de la semana
AtrHoraIni	entero	Hora de inicio
AtrHoraFin	entero	Hora de termino

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
GuardarProfeHorari	Boolean	Guarda en la bd el horario disponible del profesor	AtrCodigo	entero	Código del profesor
			AtrDia	entero	Código del día de la semana
			AtrHoraIni	entero	Hora de inicio
			AtrHoraFin	entero	Hora de termino
			AtrEmpresa	entero	Hora de termino
Elimina	Boolean	Elimina todo el horario disponible del profesor	AtrCodigo	entero	Código del profesor
			AtrEmpresa	entero	Hora de termino
Listado	DataSet	Lista el horario disponible del profesor según elección	AtrCodigo	entero	Código del profesor
			AtrEmpresa	entero	Hora de termino
Existe	Boolean	Verifica si existe o no el profesor dentro de la tabla profesor_horari	AtrCodigo	entero	Código del profesor
			AtrEmpresa	entero	Hora de termino
Llenar_Grilla	DataSet	Toma las horas libres de un profesor antes ingresadas al sistema	Ninguno		
VerificaProfeDisponible	Boolean	Verifica que el profesor este disponible a esa hora	AtrCodigo	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa
			AtrDia	entero	Código del día de la semana
			AtrHoraIni	entero	Hora de inicio

Clase: ClsProfesor

Atributos

Nombre	Tipo	Descripción
--------	------	-------------

AtrCodigo	entero	Código del Profesor
AtrPrimerNombre	carácter	Primer nombre del profesor
AtrSegundoNombre	carácter	Segundo nombre del profesor
AtrPrimerApellido	carácter	Primer apellido del profesor
AtrSegundoApellido	carácter	Segundo apellido del profesor

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Listar	Dataset	Obtiene un listado general de los docentes ordenados por su apellido paterno	Ninguno		
Busqueda_incremental_codigo	Dataset	Listado resultante de búsqueda incremental por código	cod	entero	Código del Profesor
			Apaterno	carácter	Primer apellido del profesor
			Amaterno	carácter	Segundo apellido del profesor
			Pnombre	carácter	Primer nombre del profesor
Busqueda_incremental_pnombre	Dataset	Listado resultante de búsqueda incremental por el primer nombre	Snombre	carácter	Segundo nombre del profesor
			cod	entero	Código del Profesor
			Apaterno	carácter	Primer apellido del profesor
			Amaterno	carácter	Segundo apellido del profesor
Busqueda_incremental_snombre	Dataset	Listado resultante de búsqueda incremental por segundo nombre	Pnombre	carácter	Primer nombre del profesor
			Snombre	carácter	Segundo nombre del profesor
			cod	entero	Código del Profesor
			Apaterno	carácter	Primer apellido del profesor
Busqueda_incremental_apaterno	Dataset	Listado resultante de búsqueda incremental por apellido paterno	Amaterno	carácter	Segundo apellido del profesor
			Pnombre	carácter	Primer nombre del profesor
			Snombre	carácter	Segundo nombre del profesor
			cod	entero	Código del Profesor
Busqueda_incremental_amaterno	Dataset	Listado resultante de búsqueda incremental por apellido materno	Apaterno	carácter	Primer apellido del profesor
			Amaterno	carácter	Segundo apellido del profesor
			Pnombre	carácter	Primer nombre del profesor
			Snombre	carácter	Segundo nombre del profesor

Clase: ClsProfesorHorario

Atributos

Nombre	Tipo	Descripción
AtrDia	entero	Código de día de la semana
AtrHoraInicio	entero	Hora inicio de clases
AtrHoraFin	entero	Hora fin de clases
AtrUsuario	carácter	Código de usuario
AtrFecha	carácter	Fecha
AtrProfCod	entero	Código del profesor
AtrEmpresa	entero	Código de la empresa

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
HoraDisponible	DataReader	Verifica si el profesor tiene horas disponibles en un día específico	AtrProfCod	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa
			AtrDia	entero	Código de día de la semana
Disponible	Boolean	Función que verifica si tiene disponibilidad de horario	AtrProfCod	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa
			AtrDia	entero	Código de día de la semana
			AtrHoraInicio	entero	Hora inicio de clases
			AtrHoraFin	entero	Hora fin de clases
Total_Horas_Disponibles	entero	Suma el número de horas disponibles para verifacas si cumple con el numero de créditos que va a dictar	AtrProfCod	entero	Código del profesor
			AtrEmpresa	entero	Código de la empresa

Clase: ClsProfMateri

Atributos

Nombre	Tipo	Descripción
AtrProfCod	entero	Código del profesor
AtrMatCod	carácter	Código de la materia
AtrAreaCod	carácter	Área Académica de la materia
AtrEscCod	carácter	Escuela a la que pertenece la materia
AtrPenCod	carácter	Pensum al que pertenece la materia
AtrNumParal	entero	Número de paralelos de la materia
AtrHorasnose	entero	Identifica si se dicta la materia en horas seguidas o no
Atrdias	entero	Número de dias en los que se dictara la materia

Atrfechaini	fecha	Fecha desde la que se va a dictar la materia
AtrFechafin	fecha	Fecha hasta la que se va a dictar la materia
AtrEmpresa	entero	Empresa con la que se esta trabajando

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
ListProfMat	DataSet	Listado de materias dictadas por un profesor características principales	AtrProfCod	entero	Código del profesor
Listado_materias_profesor	DataReader	Listado de materias dictadas por un profesor características adicionales	AtrProfCod	entero	Código del profesor
Ingresar	boolean	Ingresar un registro en la tabla profe_materi	Ninguno		
Eliminar	boolean	Elimina los registros que pertenecen a un profesor	AtrProfCod	entero	Código del profesor
profesor	DataReader	Listado general de los códigos de los docentes	AtrUsuario	entero	Código del Usuario
Horas_materia	entero	Número de horas a la semana en la que se dicta la materia	AtrMatCod	carácter	Código de la materia
			AtrAreaCod	carácter	Área Académica de la materia
			AtrEscCod	carácter	Escuela a la que pertenece la materia
			AtrPenCod	carácter	Pensum al que pertenece la materia

Clase: ClsTipoAula

Atributos

Nombre	Tipo	Descripción
AtrCodigo	carácter	Código del tipo de aula
AtrNombre	carácter	Nombre del tipo de aula
AtrCodBloque	entero	Código del bloque

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Listado_Tipo_aula	DataSet	Listado de tipos de aula de un bloque	AtrCodBloque	entero	Código del bloque
BADisponilbe	DataSet	Verifica si el aula de un bloque específico esta disponible	Ninguno		
NomBloque	cadena	Devuelve el nombre del Bloque	bloque	entero	Código del bloque

Clase: ClsUdaAulaEscuela

Atributos

Nombre	Tipo	Descripción
--------	------	-------------

AtrCodAula	carácter	Código del aula de clase
AtrBloAula	entero	Código del bloque
AtrTipAula	carácter	Código de tipo de aula
AtrNomAula	carácter	Nombre del aula de clase
AtrCapAula	entero	Capacidad del aula
AtrCodEscuela	carácter	Código de la carrera
AtrNomEscuela	carácter	Nombre de la carrera
AtrFacEscuela	carácter	Facultad a la que pertenece la carrera

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
ListarAula	DataSet	Muestra un listado de todas las aulas de la universidad	Ninguno		
BusquedaIncrementalNombreEscuela	DataSet	Busca el nombre de la escuela con las letras escritas	AtrNomEscuela	carácter	Nombre de la carrera
CodigoBloque	entero	Toma el codigo del edificio donde se encuentra el aula	Ninguno		
NomBloque		Toma el nombre del edificio donde se encuentra el aula	Ninguno		
NombreAula		Toma el codigo del aula y capacidad	Ninguno		

Clase: ClsUsuario

Atributos

Nombre	Tipo	Descripción
AtrCodUsuario	entero	Código del usuario
AtrUsuario	carácter	Nombre de usuario
AtrClave	carácter	Contraseña del usuario
AtrNClave	carácter	Nueva Contraseña del usuario
AtrCodEmpleado	entero	Código del docente al que pertenece el usuario
AtrAcceso	entero	Tipo de acceso que tiene el usuario

Operaciones

Nombre	Valor Devuelto	Descripción	Parámetro	Tipo	Descripción
Valida_Clave	boolean	Verifica si se ha ingresado correctamente la información en el inicio de sesión	AtrCodUsuario	entero	Código del usuario
			AtrClave	carácter	Contraseña del usuario
Listado_Usuarios	DataSet	Listado general de usuarios	Ninguno		
Consulta_datos_usuario	DataSet	Devuelve toda la información del usuario	AtrCodUsuario	entero	Código del usuario
Ingreso_usuario	boolean	Ingresa un nuevo usuario	Ninguno		
Modificacion_usuario	boolean	Modifica la información del usuario	AtrCodUsuario	entero	Código del usuario

Eliminacion_usuario	boolean	Elimina un usuario	AtrCodUsuario	entero	Código del usuario
Codigo_nuevo_usuario	entero	Genera el secuencial para el nuevo usuario	Ninguno		
Valida_codigo_employado	boolean	Verifica si el codigo de docente es un codigo valido	AtrCodEmpleado	entero	Código del docente al que pertenece el usuario
Modificacion_clave	boolean	Modifica la clave de inicio de sesión	AtrCodUsuario	entero	Código del usuario
			AtrClave	carácter	Contraseña del usuario
			AtrNClave	carácter	Nueva Contraseña del usuario
Tipo_usuario	cadena	Devuelve el tipo de acceso que tiene el usuario	Ninguno		

Anexo 11: Manual de Usuario

GUIA DE USUARIO



GEHO
Generación de Horarios

Cristian Xavier Montero Mariño
Maria Verónica Vásquez Vítores

Todos los derechos reservados.
Prohibida su reproducción total o parcial de este material,
Su tratamiento informático la transmisión de ninguna otra forma
O por cualquier otro medio ya sea electrónico, mecánico,
por fotocopia, por registro u otros métodos sin permiso previo
y por escrito de los titulares de Copyright

Cuenca – Ecuador

INDICE

INDICE	2
INTRODUCCIÓN A GEHO	3
Introduccion.....	4
Que Es Geho?	4
Ayudas	5
DESCRIPCIÓN DEL SISTEMA	6
Descripcion Geho.....	7
Modulo Usuarios	7
Gestión	7
Ingreso	7
Consulta	7
Modulo Profesores	8
Modulo Materias.....	8
Modulo Aulas	8
ENTORNO DEL SISTEMA	9
Ingresando Al Sistema	10
Inicio En Geho.....	10
Menu Geho	11
Menu Usuario.....	11
Menu Profesor.....	12
Menu Materias	13
Menu Escuela	14
Menu Horario	14
Menu Reportes.....	15
Menu Migracion.....	16
MANEJO DEL SISTEMA.....	17
Ventana Cambiar Clave	18
Ventana Inicio De Sesión.....	19
Ventana Mantenimiento De Usuarios	19
Ventana Materias Dictadas	21
Ventana Horario Específico Del Profesor	24
Cuadro De Dialogo Datos Paralelo Y Grupo.....	25
Ventana Horario Disponible Del Profesor	26
Ventana Materias Vigentes	27
Ventana Asignar Aulas.....	29
Ventana Generar.....	32
Ventana Modificar Horario	32
Ventana Migracion	36

Capitulo No.1

Introducción a GEHO

- Introducción
- ¿Qué es GEHO?
- Ayudas

INTRODUCCION

Los horarios de clase y la asignación de aulas describen la estructura organizacional y distributiva con las que cuentan las materias dentro de las diferentes horas del día, así como la ubicación física del local en el que se impartirá dicha cátedra por parte del docente, dando a conocer tanto a estudiantes como al personal docente las horas designadas para realizar sus actividades estudiantiles y de enseñanza respectivamente.

Para la universidad permiten cumplir una función de control de Asistencia eficiente al personal docente ya que cuenta con la información completa y personalizada de las horas en las que dicta cátedra.

Estas ayudas están diseñadas para brindar un soporte a lo largo del proceso de generación de horario y asignación de aulas al usuario y pueda entender el manejo del sistema completo.

QUE ES GEHO?

GEHO es un software desarrollado para la Generación de Horarios y Asignación de Aulas para instituciones educativas.

REDUCCIÓN DE TIEMPO EN LA ELABORACIÓN

En el caso de las personas que se dedican a elaborar los horarios dentro de cada una de las facultades, se les reduciría de manera considerable el tiempo que les toma realizar esta labor.

FIABILIDAD

Para los estudiantes los beneficios serán contar con información detallada y fiable de los horarios de clases al momento de hacer las prematriculas, además de tener conocimientos de las aulas en que recibirán clase dependiendo de las materias que elijan para el nuevo periodo.

CONTROL

El personal administrativo tendrá control más detallado de la asistencia del personal docente, así como de las aulas en donde dictará clase.

AYUDAS

El siguiente manual es una guía para el cliente durante la utilización del sistema. Pretende proveer información específica acerca del funcionamiento de GEHO.

Estas ayudas incluyen el proceso paso a paso de cada uno de los procedimientos necesarios para el desarrollo de la Generación de Horarios y Asignación de Aulas.

Capitulo No.2

Descripción del Sistema

- Descripción GEHO
- Módulo Usuarios
- Módulo Profesores
- Módulo Materias

DESCRIPCION GEHO

GEHO es un sistema que automatiza el proceso de Generación de Horarios y Asignación de Aulas para instituciones educativas, que se ajusta a las necesidades de cada institución.

Cuenta con los siguientes módulos:

- Usuarios.
- Profesores
- Materias
- Aulas

MODULO USUARIOS

Este modulo permite definir los usuarios que tendrán acceso al sistema, pudiendo asignarles permisos de: Administrador, Gestión, Ingreso y de Consulta.

Administrador

Este usuario es el que puede gestionar o tiene acceso a todas las opciones del programa, es el responsable de la creación del resto de usuarios.

Gestión

Tiene permisos para modificar los horarios ya generados, modificar la información de profesores, aulas, etc.

Ingreso

Es el responsable de proveer al sistema de toda la información requerida por este para realizar la generación.

Consulta

Tiene permisos para visualizar consultas y obtener los reportes que necesite.

Todos estos usuarios deberán ser asignados solo a personal de la Universidad.

MODULO PROFESORES

Gestiona la información de todos los docentes, como es el horario disponible de cada profesor, es decir, el tiempo que tiene disponible el docente para dictar clases.

Asimismo de las materias que van a ser dictadas por cada uno de los profesores, con el respectivo número de paralelos.

MODULO MATERIAS

Al igual que en los otros módulos se gestiona la información de las materias. El número de paralelos que van a existir de esa materia en el siguiente periodo educativo, el número de alumnos que se permitirán por paralelo.

Dependiendo la materia que sea, existirá la posibilidad de asignarle aulas específicas en las que necesite dictarse dicha asignatura.

MODULO AULAS

En este módulo se gestionan las aulas que serán asignadas a cada escuela para recibir clases ya sea por la mañana, tarde o noche.

Capitulo No.3

Entorno del Sistema

- Ingresando al Sistema
- Inicio en GEHO
- Menú GEHO
- Menú Usuario
- Menú Profesor
- Menú Materias
- Menú Escuela
- Menú Horario
- Menú Reportes
- Menú Migración

INGRESANDO AL SISTEMA

Asegúrese de que posea un nombre de usuario y contraseña de acceso al sistema, caso contrario póngase en contacto con la persona encargada de la administración del sistema o directamente con los desarrolladores del sistema.

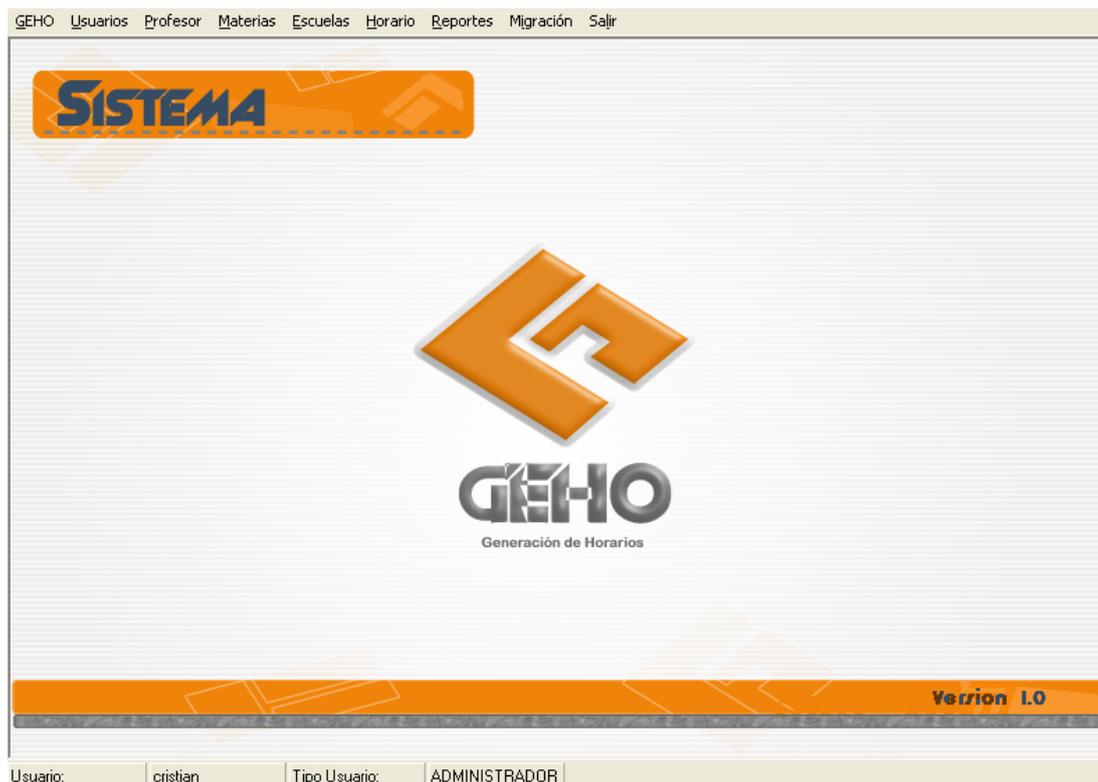
En la pantalla de inicio del sistema digite su nombre de usuario y contraseña, a continuación de clic sobre el botón Aceptar.

La imagen muestra una ventana de inicio de sesión con un título "Inicio de Sesión" en un encabezado azul. Dentro de la ventana, hay dos campos de texto: "Nombre de Usuario:" y "Contraseña:". Debajo de los campos, hay dos botones: "Aceptar" y "Salir".

Nota: Ingrese su nombre de usuario y contraseña deben ser ingresados tal cual como le fue asignado, es decir, si su nombre contiene letras mayúsculas debe ingresarlo exactamente igual.

INICIO EN GEHO

Una vez que haya ingresado al sistema se le presentara una pantalla principal con su respectivo menú, las opciones que le presente dicho menú dependerá del tipo de usuario que posea, para mayor información sobre los tipos de usuario ver el tema Descripción del sistema.



A continuación se describen todas las opciones de GEHO con el listado de los usuarios que pueden acceder a cada una de las opciones.

MENU GEHO

Usuarios que acceden:

Todos.

Cambiar Usuario

Esta opción le permitirá cambiarse de usuario en caso de que sea necesario, se debe repetir el proceso de *Ingresando al Sistema* explicado anteriormente.

MENU USUARIO

Mantenimiento

Usuario que acceden:

Administrador

Tiene como objetivo la manipulación de todo lo concerniente a usuarios, es decir, dar las facilidades para la creación, modificación, eliminación, consulta a cada uno de los usuarios.

Pero la función mas importante de esta opción es la de asignar escuelas a cada uno de estos usuarios para que el usuario pueda modificar la información solo de las escuelas que le han sido asignadas.

Cambio de Contraseña

Usuario que acceden:

Todos.

Le da la opción de que cada usuario cambie su clave de acceso al sistema, lo cual se recomienda hacerlo periódicamente y además de eso deben ser claves que no involucren nombres propios, fechas de nacimiento, etc. Con lo que se lograra que ninguna persona pueda “adivinar” su clave y tenga acceso al sistema.

En los menús que se describen a continuación pueden acceder todos los usuarios pero el control de permisos se realiza en cada uno de los formularios, permitiendo, según sea el usuario, ingresar, modificar, o solamente consultar la información que se pueda que maneje dicho formulario.

MENU PROFESOR

- **Horario Disponible**

Usuarios que acceden:

Ingreso y Gestión

Para iniciar el proceso de Generación y Asignación es necesario ingresar la aspiración de horario de cada profesor es decir, todas las horas en las que podrá dictar cátedra durante la semana. Esta información es de vital importancia para el proceso de Generación de Horarios y Asignación de aulas, puesto que dependiendo del tiempo disponible que disponga el profesor se permitirá asignarle un número de materias a dictar para un nuevo ciclo.

- **Asignar Materias**

Usuarios que acceden:

Ingreso y Gestión

Aquí se puede definir las materias que cada profesor dictara durante el siguiente periodo lectivo, cada usuario podrá asignarle materias de la o las escuelas que les hayan sido asignadas.

MENU MATERIAS

- **Materias Vigentes**

Usuario que acceden:

Ingreso y Gestión

Se deben definir todas las materias que van a ser dictadas en el próximo período lectivo, todas las materias que no tengan definida la información que se solicita en esta parte de la aplicación no será considerada para la generación del horario, es por esto, que es sumamente importante que se ingrese esta información.

MENU ESCUELA

- **Asignar Aulas**

Usuarios que acceden:

Ingreso y Gestión.

Es necesario definir las aulas de que edificio están establecidas para cada escuela incluyendo las horas ya sea en la mañana, tarde o noche, debido a que muchas escuelas tienen clases en distintos edificios y otras requieren de aulas especiales. Para estas especificaciones es de mucha utilidad este formulario puesto que asigna las aulas de clases o laboratorios a escuelas.

MENU HORARIO

Gestiona la Generación de horarios y Asignación de Aulas, así como también la modificación de este proceso.

- Generación de Horarios

Usuarios que acceden:

Ingreso

Este proceso es aquel que realiza la generación de horarios y asignación de aulas es por ello de mucha importancia, que todos los datos estén correctamente ingresados, de esta forma se evitara molestias con el nuevo horario generado.

Tome en cuenta que solo se almacena el último horario generado en la base de datos, y no es posible recuperar un horario anterior.

- **Modificar Horario**

Usuarios que acceden:

Gestión.

Una vez generado el horario de clases y asignadas las aulas, es posible que surjan ciertos cambios inesperados por el cliente, para ello esta aplicación permite realizar modificaciones en los horarios siempre y cuando cumplan con las condiciones antes ingresadas, por ejemplo no es posible asignar a un profesor una materia en la mañana si dicho profesor tiene asignado su horario disponible en la tarde.

MENU REPORTES

Usuarios que acceden:

Todos

- **Distributivos**
- **Por escuela**

Aquí se muestra toda la información concerniente a las materias de cada nivel, los correspondientes docentes que van a dictar la cátedra durante el siguiente periodo.

Toda esta información se encuentra agrupada por pensum, nivel y paralelo.

Cada usuario podrá obtener este reporte únicamente de las escuelas a las que se le dio acceso para manipular su información.

- **Por profesor**

En este reporte se encuentra organizada la información por profesor, es decir, la materia, el nivel y paralelo al que corresponde la materia que será dictada por cada profesor, así como, el número de créditos que posee dicha materia.

- **Aulas Escuelas**
 - **Asignación de Aulas a Escuela**

Se detalla las aulas de clase que están asignadas a una escuela, incluyendo al edificio que pertenece.

- **Profesor**
 - **Disponibilidad**

Presenta la disponibilidad de horas para dar clases del docente.

- **Horario de Clases**

Una vez realizado el proceso de generación, se habilita este reporte el cual permite visualizar el horario de clases del profesor que se desee observar.

MENU MIGRACION

Usuario que acceden:

Ingreso y Gestión.

Este proceso se debe ejecutar una vez concluida la generación de los horarios.

La función de este proceso es la de exportar la información del horario generado desde la base de datos del sistema a la base de datos de la Universidad del Azuay.

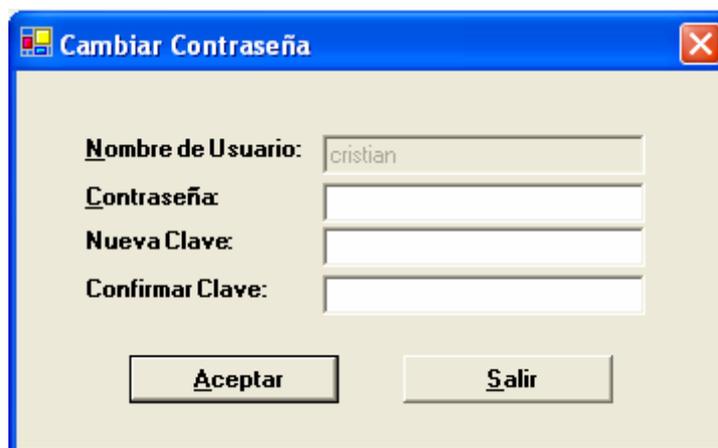
Capitulo No.4

Manejo del Sistema

- Ventana Cambiar Clave
- Ventana Inicio de Sesión
- Ventana Mantenimiento de Usuarios
- Ventana Materias Dictadas
- Ventana Horario Especifico del Profesor
- Ventana Horario Disponible del Profesor
- Ventana Materias Vigentes
- Ventana Asignar Aulas
- Ventana Generar
- Ventana Modificar Horario
- Ventana Migración

VENTANA CAMBIAR CLAVE

Si desea cambiar su clave de acceso, lo puede realizar ingresando al sistema, ubíquese en el menú Gestión y luego de clic en la opción cambiar clave, aparecerá la ventana que tenemos a continuación:



La imagen muestra una ventana de software con el título "Cambiar Contraseña". El título está en un recuadro azul con un icono de Windows a la izquierda y un botón de cerrar (una 'X' roja) a la derecha. El fondo de la ventana es beige. Hay cuatro campos de texto con sus respectivos labels a la izquierda: "Nombre de Usuario:" con el valor "cristian" ingresado; "Contraseña:"; "Nueva Clave:"; y "Confirmar Clave:". Debajo de los campos hay dos botones: "Aceptar" y "Salir".

Nombre de Usuario: Se visualizará su nombre de usuario, el cual no se podrá modificar.

Contraseña: Ingrese la clave de acceso que sea cambiar. Se recomienda realizar este cambio periódicamente.

Nueva Clave: Ingrese la nueva contraseña, es preferible evitar claves con nombres propios o fechas de nacimiento para evitar que "adivinen" su contraseña e ingresen al sistema.

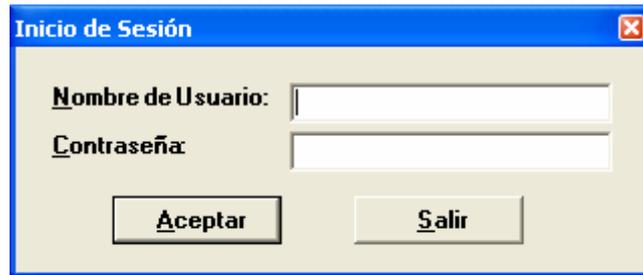
Confirmar Clave: Vuelva a digitar la nueva clave, tome en cuenta que el sistema hace diferencia entre mayúsculas y minúsculas, por tanto, es recomendable que la escriba igual, caso contrario no se realizará el cambio de contraseña.

Aceptar: Pulse el botón aceptar para guardar el cambio de contraseña, la siguiente ocasión que ingrese al sistema utilizará su nueva contraseña.

Salir: Cierra la ventana y regresa al menú principal.

VENTANA INICIO DE SESIÓN

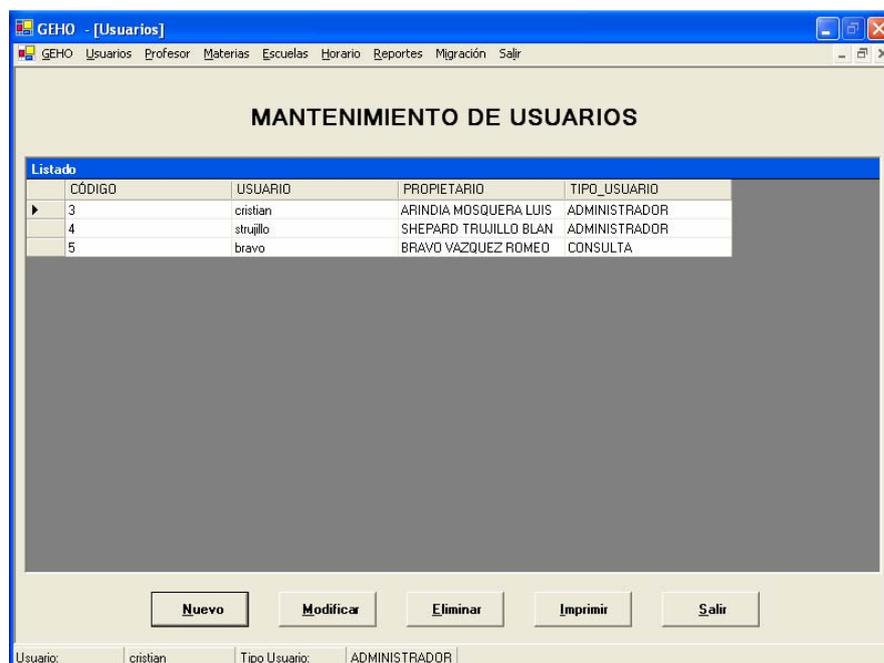
Cuando necesite cambiar de usuario ubíquese en el menú GEHO y luego de clic en la opción cambio de usuario, aparecerá la misma ventana de inicio de sesión que se visualizó al iniciar el sistema, digite su nombre de usuario, contraseña y pulse el botón aceptar de esta manera ingresará al sistema como otro usuario el cual probablemente tenga diferentes permisos de acceso.



La imagen muestra una ventana de inicio de sesión con el título "Inicio de Sesión". Contiene dos campos de texto: "Nombre de Usuario:" y "Contraseña:". Debajo de los campos hay dos botones: "Aceptar" y "Salir".

VENTANA MANTENIMIENTO DE USUARIOS

Esta ventana permite administrar los diferentes usuarios que acceden al sistema. Tiene acceso solo el usuario administrador. Para ingresar pulse en el menú Usuarios y luego de clic en Mantenimiento. Muestra en primera instancia el listado de todos los usuarios, los cuales pueden ser modificados, eliminados o imprimirlos en un reporte.



La imagen muestra una ventana de "MANTENIMIENTO DE USUARIOS" dentro del sistema GEHO. La ventana tiene un menú superior con opciones: GEHO, Usuarios, Profesor, Materias, Escuelas, Horario, Reportes, Migración, Salir. El título principal es "MANTENIMIENTO DE USUARIOS".

Debajo del título hay un "Listado" con una tabla de usuarios:

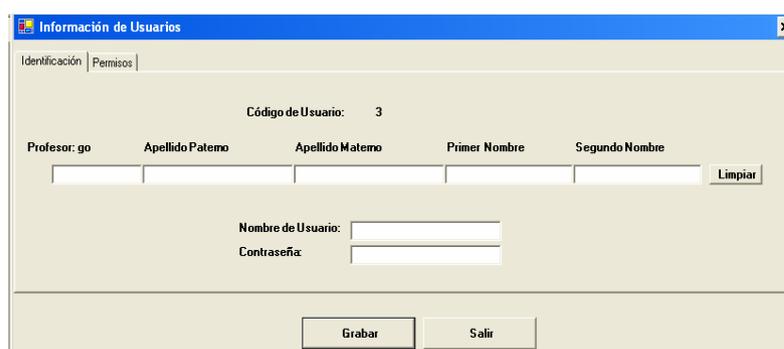
CÓDIGO	USUARIO	PROPIETARIO	TIPO_USUARIO
3	cristian	ARINDIA MOSQUERA LUIS	ADMINISTRADOR
4	strujillo	SHEPARD TRUJILLO BLAN	ADMINISTRADOR
5	bravo	BRAVO VAZQUEZ ROMEO	CONSULTA

Debajo de la tabla hay cinco botones: "Nuevo", "Modificar", "Eliminar", "Imprimir" y "Salir".

En la parte inferior de la ventana, se muestra el usuario actual: "Usuario: cristian" y "Tipo Usuario: ADMINISTRADOR".

Listado de Usuarios: Visualiza los usuarios con sus respectivos nombre de usuario, propietario y tipo de usuario.

Botón Nuevo: El botón nuevo visualiza otra ventana de información de Usuarios, para ingresar al nuevo usuario, dicha ventana contiene dos pestañas, la de identificación y la de permisos.

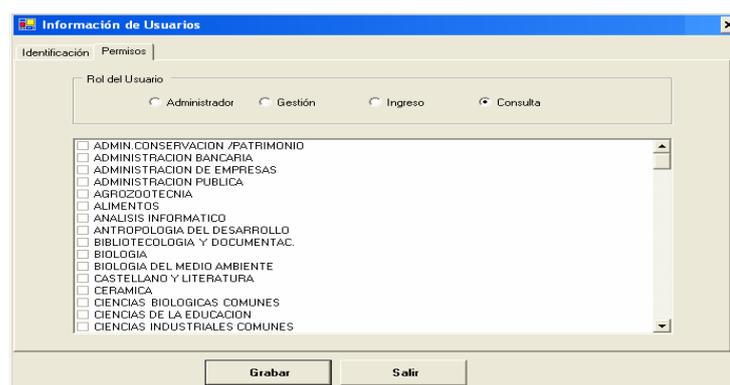


The screenshot shows a window titled "Información de Usuarios" with two tabs: "Identificación" (selected) and "Permisos". The "Identificación" tab contains the following fields and controls:

- Código de Usuario:** 3
- Profesor:** go
- Apellido Paterno:** [Empty text box]
- Apellido Materno:** [Empty text box]
- Primer Nombre:** [Empty text box]
- Segundo Nombre:** [Empty text box]
- Limpiar:** Button
- Nombre de Usuario:** [Empty text box]
- Contraseña:** [Empty text box]
- Grabar:** Button
- Salir:** Button

En la pestaña de Identificación se ingresa al usuario el cual debe estar antes ingresado en el sistema de la universidad. El sistema previamente le asignará un código de usuario, y luego el usuario administrador le asignará un nombre de usuario y una contraseña.

La pestaña permisos, contiene el rol de usuario sea este Administrador, Gestión, Ingreso o Consulta, es decir los permisos que tendrá sobre las escuelas especificadas.



The screenshot shows the "Permisos" tab of the "Información de Usuarios" window. It features the following elements:

- Rol del Usuario:** Radio buttons for "Administrador", "Gestión", "Ingreso", and "Consulta". The "Consulta" option is selected.
- Lista de Permisos:** A list of checkboxes corresponding to various university departments, including:
 - ADMIN. CONSERVACION /PATRIMONIO
 - ADMINISTRACION BANCARIA
 - ADMINISTRACION DE EMPRESAS
 - ADMINISTRACION PUBLICA
 - AGROZOO TECNIA
 - ALIMENTOS
 - ANALISIS INFORMATICO
 - ANTROPOLOGIA DEL DESARROLLO
 - BIBLIOTECOLOGIA Y DOCUMENTAC.
 - BIOLOGIA
 - BIOLOGIA DEL MEDIO AMBIENTE
 - CASTELLANO Y LITERATURA
 - CERAMICA
 - CIENCIAS BIOLÓGICAS COMUNES
 - CIENCIAS DE LA EDUCACION
 - CIENCIAS INDUSTRIALES COMUNES
- Grabar:** Button
- Salir:** Button

Botón Modificar: Para modificar un usuario debe previamente escogerlo del listado de usuarios (1), que se muestra en pantalla. Pulse en el botón modificar y aparecerá la ventana de Información de Usuario (descrita anteriormente) con los datos del usuario seleccionado. En esta ventana podrá realizar las modificaciones convenientes.

Botón Eliminar: Si desea eliminar un usuario, selecciónelo de la lista de usuarios (1), pulse el botón eliminar, se visualizará la ventana de Información de Usuario y de clic en el botón eliminar.

Botón Imprimir: El botón imprimir es útil para visualizar un reporte de todos los usuarios ingresado, si desea imprimir este listado haga clic en el botón de imprimir que se encuentra en la parte superior izquierda del reporte. 

Botón Salir: Pulse el botón salir cuando quiera cerrar a la ventana y regresar al menú principal.

VENTANA MATERIAS DICTADAS

Esta ventana se encuentra dentro del menú Profesor, no es más que el distributivo Profesor – Materia que se realiza antes de cada ciclo.



CODIGO	NOMBRE	AREA	COD.ESC	ESCUELA	PENSUM	PARAL	HS	DIA

En esta ventana se realiza la asignación de materias a un profesor con ciertas especificaciones importantes y necesarias que deben ser tomadas en cuentas al momento de la generación de horarios.

Para la manipulación de esta ventana se requiere seguir ciertos pasos en orden; a continuación detallamos el significado y utilización cada uno de los parámetros requeridos:

En primer lugar se debe elegir el profesor a quien se le va a asignar materias a dictar, es de mucha utilidad que escriba el nombre del profesor ya que el sistema buscará de inmediato los coincidentes, seleccione el docente dando clic sobre el nombre.

Procesa a dar clic en el botón Cargar Info para mostrar las materias asignadas. Si no aparece nada en pantalla es porque no hay materias asignadas a ese profesor, caso contrario aparece un listado de materias con su descripción.

Si desea asignar una materia pulse en el botón Nuevo, se habilitarán las opciones que se encuentran debajo del botón. A continuación detallamos cada una de las opciones:

Escuela: Digite el nombre de la escuela a la que pertenece la materia, en esta opción podrá tener acceso solo a las escuelas de su facultad.

Pensum: Digite el código del pensum al que pertenece la materia (valor numérico). Es necesario que indique este parámetro ya que una misma materia se encuentra en varios pensum y no todos están vigentes.

Materia: Una vez elegida la escuela y el pensum, digite el nombre de la materia que desea asignar, el sistema mostrará un listado de materias coincidentes, de clic en la materia deseada.

Asignar Horario: Este botón permite especificar un horario de clases predeterminado para el profesor y la materia dictada. Se explicará en la siguiente ventana con detalle su función.

No. Paralelos: Este parámetro especifica a cuantos paralelos el profesor va a dar la materia escogida. Por ejemplo, si elige tres paralelos da a entender que la materia 'Y', será dictada en los paralelos A, B y C por el profesor 'X'.

Horas: Indica el número de horas en las que se dictará esa materia a la semana.

¿Horas no Seguidas?: Active esta casilla en caso de no querer que se dicte la materia en horas seguidas. Por ejemplo de 7 a 9.

Días: Especifique en cuantos días desea dictar la materia. Para este parámetro se debe tomar en cuenta las horas asignadas a la materia en la semana, Por ejemplo, si las horas son 8 y elige dictar en 2 días, el sistema generará para esta materia 4 horas en un mismo día, lo cual sería una recarga para el alumno y el profesor. Se aconseja que como máximo se dicte la materia 2 horas por día.

Fecha Inicio: Indique la fecha en la cual se va a comenzar a dictar la materia, generalmente es igual a la fecha de inicio de ciclo.

Fecha Fin: Indique la fecha en la cual va a culminar el dictado de la materia, generalmente es igual a la fecha de fin de ciclo.

Una vez ingresados todos los parámetros descritos, procedemos a pulsar el botón Agregar, de esta manera agrega en el listado de materias, la materia con toda su descripción.

Repita el proceso si desea asignar otra materia.

Una vez asignadas todas las materias al docente (ya sean estas de diferentes escuelas), se procede a Guardar los cambios pulsando el botón Guardar. Cabe recalcar si usted ha predeterminado un horario de clases para el profesor y una materia específica, estos datos también se guardan (Ventana Horario Especifico del Profesor).

Cuando desee borrar una materia del listado, seleccione la materia y luego pulse el botón Quitar.

Con el botón salir cierra la ventana y regresa al menú principal. Si cierra la ventana sin haber guardado previamente, todos los datos ingresados en último momento se borrarán y no guardará los cambios.

VENTANA HORARIO ESPECÍFICO DEL PROFESOR

Existen ocasiones en las que es preciso determinar un horario de clases exacto para el profesor y la asignatura especificada. Para acceder a esta ventana pulsamos clic en el botón Asignar Horario que se encuentra en la Ventana Materias Dictadas.

Asignación de Materias a Profesores

Código: 3360 Apellido Paterno: CUEVA Apellido Materno: MALO Primer Nombre: RODRIGO Segundo Nombre: ALEJANDRO

CODIGO	NOMBRE	AREA	COD.ESC	ESCUELA	PENSUM	PARAL	HS	DIAS	F

Escuela: ADMINISTRACION DE EMPRESAS Pensum: 99 Materia: ADMINISTRACION FINANCIERA III No. Paralelos: 1 Créditos: 4

Fecha Inicia: Lunes, 13 de Marzo de 2006 Fecha Fin: Viernes, 07 de Julio de 2006

Asignar Horario

Una vez pulsado el botón aparece la Ventana de Horario Especifico del Profesor, semejante a un horario de clases. En la parte superior se encuentra detallado el docente, la materia, pensum y ciclo al que pertenece. Luego el horario de clases donde se podrá indicar los días, horas, paralelos y grupos en los que se desea dictar la materia, pulsando cada celda se desplegará la ventana datos paralelo y grupo que se explica más adelante.

Asignación de Horario...

Docente: RODRIGO ALEJANDRO CUEVA MALO Pensum: 99
Materia: ADMINISTRACION FINANCIERA III Ciclo: 9

Especifique Paralelo y Grupo al cual se va a dictar clase:

HORAS	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO
7 - 8						
8 - 9						
9 - 10						
10 - 11						
11 - 12						
12 - 13						
13 - 14						
14 - 15						
15 - 16						
16 - 17						
17 - 18						
18 - 19						
19 - 20						
20 - 21						
21 - 22						

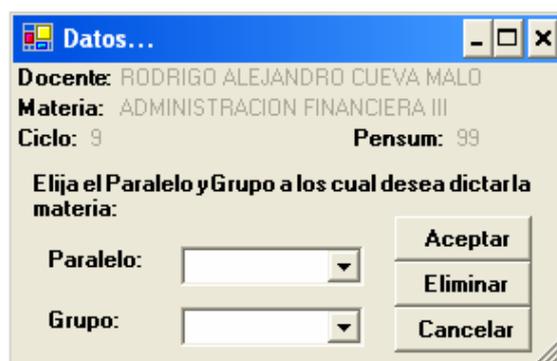
Aceptar Cancelar

Los botones Aceptar y Cancelar que se encuentran en la parte inferior de la pantalla son utilizados en caso de que se desee mantener el horario de clases o no. Si desea guardar el horario debe cerrar la pantalla y luego pulsar el botón Guardar ubicado en la Ventana Materias Asignadas.

NOTA: Utilice esta opción solo si es estrictamente necesario, caso contrario se sugiere que el sistema genere automáticamente el horario de clases y designe por su cuenta las horas y días en que el docente dictará la materia.

CUADRO DE DIALOGO DATOS PARALELO Y GRUPO

Para establecer dentro de la Ventana Horario Especifico del Profesor el paralelo y el grupo al que se va a dictar clase, tan solo de clic en una de las celdas y aparecerá un cuadro diálogo como el siguiente:



The dialog box, titled "Datos...", contains the following information and controls:

- Docente:** RODRIGO ALEJANDRO CUEVA MALO
- Materia:** ADMINISTRACION FINANCIERA III
- Ciclo:** 9
- Pensum:** 99
- Instruction:** Elija el Paralelo y Grupo a los cual desea dictarla materia:
- Paralelo:** A dropdown menu.
- Grupo:** A dropdown menu.
- Buttons:** Aceptar, Eliminar, and Cancelar.

Este cuadro de diálogo indica en la parte superior el docente, materia, ciclo y pensum correspondientes. A continuación es posible elegir el paralelo y grupo al que desea dar clase.

Para mostrar los datos dentro del horario de clases pulse el botón Aceptar, caso contrario pulse el botón Cancelar. El botón Eliminar es utilizado en el caso de haber tenido antes ya datos en una celda y quiere borrarlos del horario de clases.

Nota: Se utilizará en la opción grupo un número mayor que 1 en caso de que al paralelo se divida en grupos debido al exceso de alumnos.

VENTANA HORARIO DISPONIBLE DEL PROFESOR

Se encuentra ubicada debajo el menú profesor. Es una de las primeras ventanas en ser utilizadas, ya que de la información que se ingrese aquí, se ingresarán las demás.

En el siguiente gráfico observamos la ventana de horario disponible del profesor y a continuación explicaremos su funcionamiento:

The screenshot shows a software window titled 'HORARIO DISPONIBLE DEL DOCENTE'. At the top, there are search fields for 'Código', 'Apellido Paterno', 'Apellido Materno', 'Primer Nombre', and 'Segundo Nombre', along with a 'Limpiar' button. Below these is a 'Cargar Horario' button. The main area is a table with columns for 'HORAS' (7-8 to 21-22) and days of the week (LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO), each containing a checkbox. At the bottom, there are buttons for 'Limpiar', 'Grabar', 'Imprimir', and 'Salir'. The status bar at the very bottom shows 'Usuario: WV' and 'Tipo Usuario: ADMINISTRADOR'. Numbered callouts (1-7) point to: 1. The search fields; 2. The 'Cargar Horario' button; 3. A checkbox in the 'SABADO' column; 4. The 'Limpiar' button; 5. The 'Grabar' button; 6. The 'Imprimir' button; 7. The 'Salir' button.

El funcionamiento de esta ventana es muy sencillo, es aquí donde se debe indicar la aspiración de horario que tiene el profesor para dictar clases, es recomendable ingresar el doble horas disponibles para que no haya conflictos al momento de generar los horarios de clases.

Nombre de Profesor: Ingrese en cualquiera de los campos según el título el nombre del profesor, inmediatamente se desplegará una lista de nombres que coincidan con el que usted necesita, haga doble clic en el nombre que desee y el nombre completo del profesor se mostrará en pantalla.

Mostrar Horario: Una vez elegido el profesor, de clic en el botón Mostrar Horario para que presente en pantalla el horario disponible. En caso de no

presentarse nada en pantalla significa que el profesor no ha sido antes ingresado.

Horario Disponible: El cuadro de Horario Disponible es semejante a la hoja de papel que entregan a los docentes para que llenen su horario para dar clases. El día y la hora que el profesor tenga disponible de clic, de inmediato se activará la casilla de verificación. Si desea desactivarla de nuevamente clic.

Limpiar: El botón limpiar le permite borrar de pantalla todas las casillas de verificación activadas, es necesario recordar que se borran solo de la pantalla si desea guardar los cambios debe pulsar el botón guardar.

Grabar: El botón guardar como su nombre lo indica guarda los cambios realizados en pantalla, es decir guarda el horario disponible del profesor para dictar clases, el cual será utilizado luego para la generación de horarios.

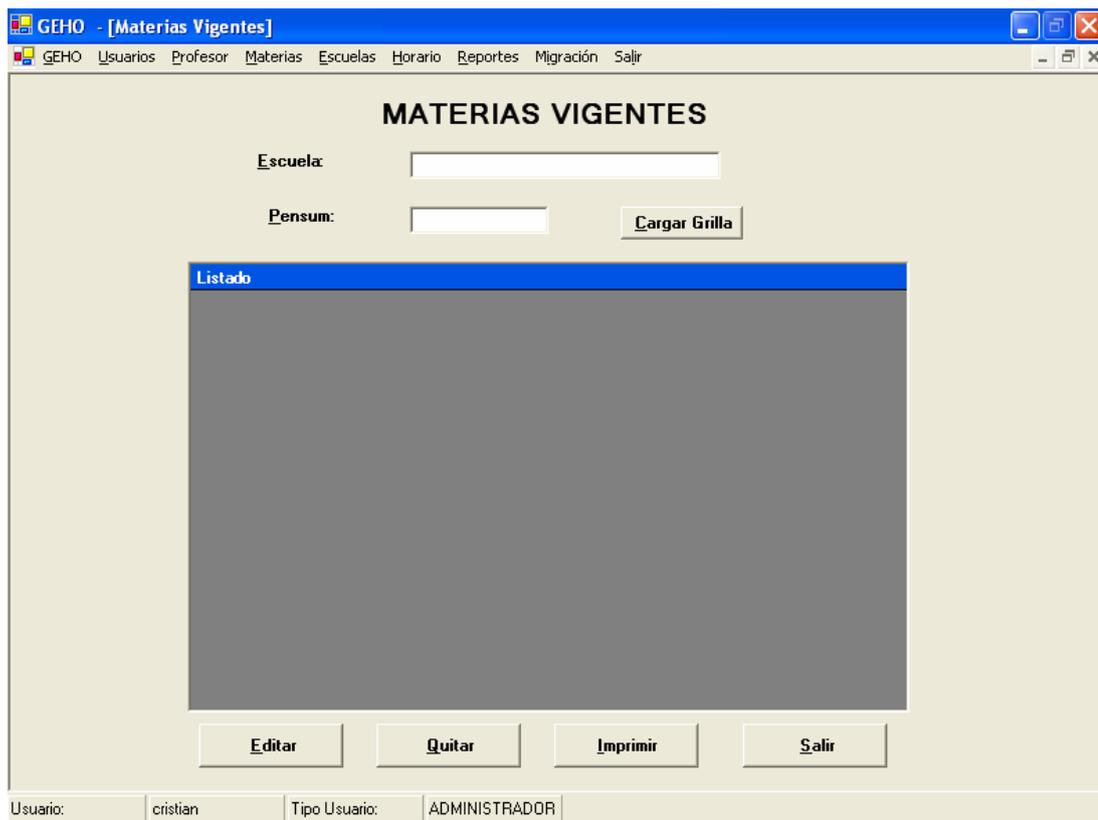
Imprimir: Al dar clic en el botón Imprimir, se visualizará un reporte previo a la impresión del horario disponible del profesor. Si desea imprimir el reporte tan solo de clic en el botón de impresora que se encuentra en el reporte.

Salir: Cierra el formulario y se visualiza el menú principal. Antes de pulsar este botón asegúrese de haber guardado primero los datos.

VENTANA MATERIAS VIGENTES

Materias Vigentes se encuentra dentro del menú materias. Siendo parte del módulo de Materias, es de mucha importancia que sea una de las primeras ventanas en ser utilizada, puesto que dependiendo de las materias que se acepten, se podrá realizar el distributivo de profesor – materia, y posteriormente la generación de horarios. Es aquí donde se dan ciertas características a cada materia vigente para el próximo ciclo.

El primer paso a realizar es digitar el nombre de la escuela, digitar el código de pensum y pulsar el botón Mostrar información; a continuación se desplegará un listado con todas las materias vigentes de la escuela y pensum escogidos.



Botón Editar: Luego de seleccionar una materia del listado, pulsamos el botón editar el cual nos presentara una pantalla con la información de la materia seleccionada.

Botón Quitar: Borra de pantalla la materia seleccionada.

Botón Imprimir: Muestra un listado de todas las materias que se visualizan en pantalla con sus características. Si desea imprimir el reporte de clic en botón imprimir que se encuentra en el reporte. 

Botón Salir: Cierra la ventana y regresa al menú principal, se recomienda guardar todos los cambios antes de cerrar la ventana caso contrario no se guardará la información que haya ingresado.

Seleccionando una de las materias del listado y pulsando el botón Editar es posible asignar ciertas características, las cuales detallaremos a continuación:

No. Paralelos: Especifica el número de paralelos en los que se dictará la materia.

No. Alumnos: Especifica el número máximo de alumnos que podrá tener esta materia, esta opción es muy útil en caso de tener materias técnicas dictadas en laboratorios donde pueden recibir un número reducido de alumnos clases.

Cabe recalcar que el máximo permitido por la universidad es de 45 alumnos por aula.

No. Horas Aula: Indica el número de horas a la semana que se dictará la materia en un aula común. Por ejemplo si una materia técnica es dada en 6 horas a la semana, y utiliza un laboratorio 2 horas semanales, en el campo No. Horas Aula se colocará 4.

En caso la materia requiera de un aula o varias específicas durante todo el ciclo, sea esta laboratorio, audiovisuales o aula común:

Bloque:, En primer lugar digite el edificio al que pertenece.

Tipo Aula: A continuación se especifica el tipo de aula que se requiere asignar, aula, bodega, laboratorio, oficina, otros.

Aula: De clic en la casilla del aula (s) que desee asignar a la materia.

Capacidad: Cada vez que seleccione un aula en este campo se mostrará su capacidad máxima.

VENTANA ASIGNAR AULAS

Esta ventana se encuentra ubicada bajo el menú Escuelas. La información que se ingrese aquí es de vital importancia al momento de realizar el proceso de asignación las aulas a cada curso.

Para comenzar debe digitar el nombre de la escuela a la que desea asignar aulas, luego pulse el botón Cargar Aulas, a continuación mostrará el listado de aulas y en la parte inferior el listado de las aulas ya asignadas a la escuela, las cuales se detalla a continuación:

Asignacion de Aulas a Escuela

ASIGNACION DE AULAS A ESCUELAS

Escuela:

AULAS					
CODIGO	NOMBRE	BLOQUE	CAPACIDAD	TIPO	
101	TALLER GENERICO	BLOQUE GENERICO	40	TALLER	
102	AULA GENERICA	BLOQUE GENERICO	40	AULA	
9001	CERAMICA	BLOQUE GENERICO	35	TALLER	
901	AULA TEORICA	BLOQUE GENERICO	30	AULA	
903	AULA PRACTICA	BLOQUE GENERICO	30	LABORATORIO	
904	AULA TEORICA	BLOQUE GENERICO	35	AULA	
9080	CONFORT	BLOQUE GENERICO	55	LABORATORIO	
910	ALIMENTOS	BLOQUE GENERICO	20	LABORATORIO	
9999	LABORATORIO	BLOQUE GENERICO	0	AULA	
9999	AULA GENERICA	BLOQUE GENERICO	0	AULA	

CODIGO	NOMBRE	BLOQUE FISICO	CAPACIDAD	MAÑANA	TARDE	NOCHE
202	AULA (202)	ADMINISTRACION DE E	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
203	AULA (203)	ADMINISTRACION DE E	0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
203	AULA (203)	COLEG.ASUNCION	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
204	AULA (204)	ADMINISTRACION DE E	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
601	AULA 601	COLEG.ASUNCION	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
701	AULA 701	COLEG.ASUNCION	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
702	AULA 702	COLEG.ASUNCION	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
703	AULA 703	COLEG.ASUNCION	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
704	AULA 704	COLEG.ASUNCION	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
705	AULA 705	COLEG.ASUNCION	0	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Listado Aulas: Luego de pulsar el botón Cargar Aulas se habilita el listado aulas y muestra todas las aulas de la universidad indicando código, nombre, edificio, capacidad y tipo. Dando clic en un aula esta se asignará a la escuela y se mostrará en el listado de la parte inferior.

Listado Aulas Asignadas: En este listado se encuentran todas las aulas asignadas a la escuela.

Es necesario que especifique a que hora del día esta disponible el aula para la escuela ya que muchas aulas son compartidas por diferentes escuelas en distintas horas del día.

De un clic en la casilla de verificación sea esta mañana, tarde o noche, puede elegir las tres opciones si es requerido.

Eliminar Registro: Si desea eliminar una asignación seleccione el aula de listado Aulas Asignadas y pulse el botón Eliminar registro.

Otra manera práctica de eliminar un registro asignado es seleccionando el aula y pulsando la tecla suprimir.

Limpiar: Borra todas las aulas del listado de aulas asignadas.

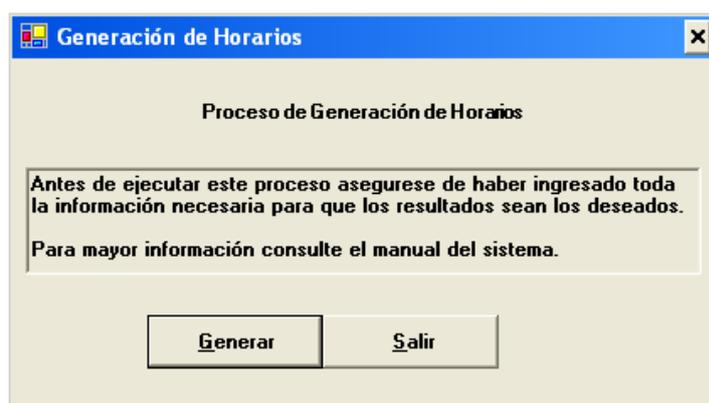
Guardar Asignación: Una vez realizado el proceso de asignación debe guardar los cambios dando clic en el botón Guardar Asignación, este realiza una verificación en la información ingresada, en caso de existir algún inconveniente el sistema mostrará un mensaje de error.

Imprimir: Si ha digitado una escuela, al dar clic en el botón Imprimir mostrará el reporte de aulas asignadas a dicha escuela, caso contrario, mostrará el reporte de todas las escuelas a las que tiene permiso de acceso con sus respectivas aulas asignadas. Para imprimir de clic en el botón imprimir ubicado en la parte superior izquierda del reporte.

Salir: Cierra la ventana y regresa al menú principal. No olvide guardar los cambios realizados antes de salir.

VENTANA GENERAR

Una vez que haya ingresado correctamente los datos requeridos, se procede a generar el horario de clases y asignar las aulas. Para ello tenemos la ventana de Generación de Horarios, de clic en el botón Generar, si la generación se ha realizado con éxito aparecerá un mensaje indicándolo. Caso contrario, verifique la información ingresada o comuníquese con el administrador del sistema.



VENTANA MODIFICAR HORARIO

Una vez generado el horario de clases, se pueden presentar ligeros cambios por parte del cliente, es por ello, que esta ventana permite modificar un horario generado o ingresar otro. Tan solo el usuario con acceso de Gestión podrá realizar los cambios respectivos.

GEHO - [Ingreso de Horario]

GEHO Usuarios Profesor Materias Escuelas Horario Reportes Migración Salir

Escuela: Ciclo: Paralelo: Pensum: Grupo: Cargar Datos

HORARIO DE CLASES

HORAS	LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
7 - 8					
8 - 9					
9 - 10					
10 - 11					
11 - 12					
12 - 13					
13 - 14					
14 - 15					

Guardar Horario Imprimir Salir

Usuario: cristian Tipo Usuario: ADMINISTRADOR

Ya sea para modificar un horario o para ingresar uno nuevo debe primero ingresar los datos dentro de:

Escuela: Digite el nombre de la escuela

Ciclo: Elija el ciclo al cual desea cambiar el horario.

Paralelo: Escoja el paralelo.

Pensum: Elija el pensum vigente para el nuevo ciclo

Grupo: Esta opción es nueva para el sistema tradicional, es utilizado para casos especiales en los que a un paralelo se lo divide en varios grupos en una o varias materias debido al exceso de alumnos. Esta opción permite indicar a los alumnos y los docentes que días exactamente tendrán clase dependiendo el grupo que se les asigne.

Mostrar Datos: Una vez llenados los campos anteriores pulse el botón Mostrar Datos para que se muestre en pantalla el horario ya generado. En caso de no mostrar nada se entiende que no ha sido generado ningún horario.

Una vez cargados los datos se habilitará el cuadro donde muestra el horario de clases y será posible modificar la información.

Horario de Clases: En cada celda muestra la materia que corresponde a esa hora y día, el aula y edificio donde será dictada, y el nombre del profesor que dará la clase.

Si desea modificar esta información de clic en la celda correspondiente, aparecerá otro cuadro de diálogo como el siguiente:

MATERIA:

Codigo	Materia
ISI0701	INVESTIGACION
ISI0702	TEORIA DE LA C
ISI0703	ANALISIS DE SIS
ISI0704	LABORATORIO D
ISI0705	REDES Y COMU
ISI0791	PENSAMIENTO S

UBICACION:

Bloque:

Tipo Aula:

Aula:

PROFESOR:

Código	Apellido Paterno	Apellido Materno	Primer Nombre	Segundo Nombre
<input type="text"/>				

Limpiar

Regresar Eliminar Registro Cancelar

El listado de materias que se muestren dependerá del ciclo que haya escogido, seleccione una materia dando un clic sobre ella, y de inmediato esta se mostrará en la parte superior.

Digite el nombre del profesor para que se despliegue un listado de nombres coincidentes, de doble clic en el nombre que corresponda.

Pueden existir dos razones por las que el sistema no permita elegir a un profesor:

El profesor ya tiene asignada esa hora para dictar clases.

El profesor no tiene ingresado en la ventana de Horario Disponible del Profesor esa hora ni día, por ejemplo si el docente "X" ha ingresado como horario disponible para dictar clases todos los días de 7H00 a 9H00, es imposible que se le asigne en Modificar Horario una materia a las 20H00. Para solucionar este inconveniente debe primero agregarle horas disponibles en la ventana Horario Disponible del Profesor ubicado en el menú profesor.

A continuación especifique donde se irá a dictar la materia, comencemos por digitar el nombre del edificio, generalmente tienen el nombre de la facultad a la que han sido asignados ya sea Filosofía, Administración, Diseño, CCTT, Colegio Asunción, etc.

Una vez elegido el Bloque o Edificio, tomamos el tipo de aula en el que vamos a dar clase sea aula normal, laboratorio, etc., de inmediato vemos un listado de aulas disponibles. Marcamos el aula en el listado, cabe recalcar que solo puede marcar un aula.

Por último pulsamos el botón Regresar para cerrar el cuadro de diálogo y ver en pantalla la nueva asignación.

En el caso de modificar un registro, se realiza el mismo proceso anterior con la diferencia que se muestran los datos actuales a lo que podrán realizar cambios.

Si desea eliminar un registro de un clic sobre él y en la ventana Datos Horario pulse el botón Eliminar Registro.

Una vez realizadas todas las modificaciones o ingresadas todas las materias de un ciclo es necesario guardar el nuevo horario (8) para que se deposite en la base de datos de la universidad y así tengan acceso también los alumnos.

Imprimir: Si desea ver el horario de clases pulse el botón Imprimir, a continuación aparecerá un reporte con el horario de clases modificado o ingresado siempre y cuando haya guardado los cambios antes. Si desea imprimirlo pulse el botón imprimir que se encuentra en la parte superior izquierda del reporte.

Salir: Cierra la ventana y regresa al menú principal. Recuerde guardar los cambios antes de salir.

VENTANA MIGRACION

A esta ventana solo tiene acceso el usuario de Gestión, no es más que actualizar la información en la base de datos de la universidad.

Lo único que deberá hacer es escoger el periodo al que corresponde el horario generado y luego pulsar en el botón migrar. La velocidad del proceso dependerá de la cantidad de información que se posea en ese momento.

Migración de la Información

Escuela:

Especifique el periodo al que pertenecera el horario

Periodo:

CODIGO	FECHA_INICI	FECHA_FINA	DESCRIPCIO
[Empty table body]			

Migrar **Salir**