BENEMERITA UNIVERSIDAD AUTÓNOMA DE PUEBLA
FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

"SISTEMA WEB DE ADMINISTRACION ESCOLAR PARA INSTITUCIONES DE EDUCACION MEDIA SUPERIOR"

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE
LIC. EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA

SANDY CECILIA OROZCO VELÁZQUEZ

ASESORA: M.E. MARÍA DEL CARMEN CERÓN GARNICA

PUEBLA, PUE. DICIEMBRE 2009

INDICE

INTRODUCCION

I. Antecedentes ...1
II. Objetivos ...1
III. Organización de la Tesis ...2

CAPITULO I. MARCO TEÓRICO

1.1. Ingeniería de Software ...3
1.1.1. Modelos Genéricos del Desarrollo de Software ...4
1.1.2. Notación UML ...9
1.1.2.1. La Ingeniería de Software ...9
1.1.2.2. La complejidad del Software ...10
1.1.2.3. Principios de Modelado ...10
1.1.2.4. El Lenguaje de Modelado Unificado UML ...11
1.1.2.5. El proceso Unificado de Modelado (RUP) ...12
1.1.2.6. Diagramas de UML ...13
1.1.3. Ingeniería Web ...13
1.1.3.1. Atributos de las Aplicaciones Basadas en Web ...14
1.1.3.2. Características de las Webapps ...14
1.1.3.3. Categorías de Aplicación de las Webapps ...15
1.2. Bases de Datos ...15
1.2.1. Sistema Gestor de Bases de Datos ...16
1.2.2. Sistema de Bases de Datos ...17
1.2.3. Modelos de Datos Conceptuales ...19
1.2.3.1. Modelo entidad-relación ...19
1.2.3.2. Metodología del Diseño Conceptual ...21
1.2.4. Herramientas de Bases de Datos Web ...22
1.3. Modelo Cliente-Servidor ...23
1.4. World Wide Web ...25
1.5. MySQL ...25
1.6. PHP ...29

CAPITULO 2. ADMINISTRACION ESCOLAR

2.1. Antecedentes UIA ...31
2.2. Modelo UIA ...32
2.3. Sistemas de Administración Escolar ...32
2.4. Políticas de la SEP ...35
2.5. Objetivo de la Administración Escolar ...36
2.6. Objetivos Específicos de la Administración Escolar ...36
2.7. Descripción de la Preparatoria UIA ...37

CAPITULO 3. ANÁLISIS DEL SISTEMA

3.1. Planteamiento del Problema ...39
3.1.1. Especificación de Requerimientos ...39
3.1.2. Antecedentes ...41
3.1.3. Objetivos ...41
3.1.4. Alcances ...41
3.1.5. Metodología ...42

3.1.6. Definición de Requerimientos Funcionales ...42
3.1.7. Definición de Requerimientos No Funcionales ...42
3.1.8. Diagrama de Casos de Uso ...43
3.1.8.1. Casos de Uso y sus Descripciones ...44
3.1.8.2. Diagramas de Secuencia ...48
3.1.8.3. Subcasos de Uso ...49

CAPITULO 4. DISEÑO DEL SISTEMA
4.1. Arquitectura del Sistema ...55
4.2. Diagrama de Navegación ...55
4.3. Modelo Conceptual de la Base de Datos ...58
4.4. Diseño Inicial de la Base de Datos ...64
4.4.1. Diagrama Entidad-Relación ...64
4.4.2. Normalización ...66
4.4.3. Diseño Físico de la Base de Datos ...72

CAPITULO 5. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA
5.1. Herramientas para el Desarrollo del Sistema ...74
5.1.1. Implementación de la Base de Datos en MySQL ...75
5.2. Conexión MySQL con PHP ...82
5.3. Interfaces del Sistema de Administración Escolar ...84
5.3.1. Interfaces del Usuario Administrador ...85
5.3.2. Interfaces del Usuario Auxiliar ...86
5.3.3. Interfaces del Usuario Docente ...88
5.3.4. Interfaces del Usuario Alumno ...90
5.3.5. Reportes ...91

CONCLUSIONES ...94
PERSPECTIVAS ...94
BIBLIOGRAFÍA ...95

ANEXO A. Simbología de los Diagramas ...98
ANEXO B. Manual del Usuario ...100

INTRODUCCION

I. Antecedentes

En los últimos años, la rápida expansión de Internet y del uso de intranets corporativas ha supuesto una transformación en las necesidades de información de las organizaciones. En particular esto afecta a la necesidad de que:

1. La información sea accesible desde cualquier lugar dentro de la organización e incluso desde el exterior.
2. Esta información sea compartida entre todas las partes interesadas, de manera que todas tengan acceso a la información completa (o a aquella parte que les corresponda según su función) en cada momento.

Estas necesidades han provocado un movimiento creciente de cambio de las aplicaciones tradicionales de escritorio hacia las aplicaciones Web, que cumplen a la perfección con las necesidades mencionadas anteriormente. Por tanto, los sitios Web tradicionales que se limitaban a mostrar información se han convertido en aplicaciones capaces de una interacción más sofisticada con el usuario. Inevitablemente, esto ha provocado un aumento progresivo de la complejidad de estos sistemas y, por ende, la necesidad de buscar opciones de diseño nuevas que permitan dar con la arquitectura óptima que facilite la construcción de los mismos, y que al mismo tiempo proporcionen la funcionalidad y comodidad que las organizaciones y sus usuarios requieren para el manejo de su información.

II. Objetivos

Objetivo General

El objetivo de este proyecto es realizar un sistema de Bases de Datos basado en Web, el cual cumpla las funciones de la administración escolar para nivel preparatoria (caso de estudio: Preparatorias de la Universidad Iberoamericana.)

Objetivos Específicos

- Estudiar el lenguaje de programación Web PHP, el sistema de gestión de base de datos relacional MySQL y la integración de las bases de datos en las aplicaciones Web, para utilizarlos en la creación del sistema Web propuesto.
- Control escolar que por el momento gestione a 150 alumnos y 50 profesores (plantel actual) pero que cada año incremente la matricula en un 5%.
- Control de alumnos inscritos y de nuevo ingreso con formato de inscripción, ficha médica y registro de documentos.
- Control de profesores para registrar datos personales y capturar calificaciones.
- Control de grupos que incluyen 3 años de preparatoria, asignando las materias con su respectivo profesor.
- Consultas Web por parte del administrador, profesor y alumno.
- Reportes por parte del administrador, profesor y alumno.

III. Organización de la Tesis

El contenido de la tesis cubre el marco tanto teórico como práctico, los cuales nos permiten alcanzar el objetivo anteriormente planteado. El marco teórico está formado por los dos primeros capítulos, los cuales cubren los temas necesarios para la comprensión del desarrollo del sistema. En los siguientes capítulos encontraremos el análisis, diseño, implementación y evaluación del sistema. A continuación se muestra un panorama general de cada uno de los capítulos que conforman la tesis:

- En el Capítulo 1 se da una breve explicación de los conceptos generales de los sistemas Web, presentando conceptos importantes como Ingeniería de Software, Bases de Datos y sus manejadores en particular MySQL, Modelo cliente-servidor, World Wide Web y el lenguaje de programación PHP.
- En el Capítulo 2 se describe el concepto de administración escolar, en específico el modelo, políticas, y objetivos de la Preparatoria UIA.
- En el Capítulo 3 se da la descripción del análisis del sistema, planteando el problema y proponiendo una estrategia de solución.
- En el Capítulo 4 se describe el diseño del sistema, se incluye el diseño de módulos e interfaces y la estructura de la base de datos.
- En el Capítulo 5 se describe el desarrollo del sistema, se implementa y se realizan las pruebas mostrando el producto final.
- Finalmente se adjuntan a estos capítulos los resultados y conclusiones, las perspectivas y limitaciones. En el apéndice se incluyen los anexos y el manual de usuario para el sistema.

CAPÍTULO I. MARCO TEÓRICO

En este capítulo se incluyen los conceptos básicos utilizados para el desarrollo del sistema, haciendo un breve resumen y una recapitulación de la información que ha servido de base para sustentar todos los procesos que se llevan a cabo, logrando también una mejor comprensión el sistema.

1.1. Ingeniería de Software

El software de computadora es el producto que diseñan y construyen los ingenieros de software. Esto abarca programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, documentos que comprenden formularios virtuales e impresos y datos que combinan números y texto, así como representaciones de información de audio, video e imágenes.

El impacto del software en nuestra sociedad y en la cultura es profundo, por ello la comunidad del software trata continuamente de desarrollar tecnologías que hagan más sencillo, rápido y menos costosa la construcción de programas de computadora de alta calidad.

La **Ingeniería de Software** es una disciplina o área de la informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. [1]

La ingeniería de Software trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como la construcción de compiladores, sistemas operativos o desarrollos en Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, administración, derecho, Internet, etc.

Independientemente de la complejidad del sistema y de su área de aplicación, la ingeniería de software puede considerarse como una tecnología multicapas (ver **figura 1.1**), donde la primera capa enfatiza que los cimientos de la ingeniería de software están orientadas hacia la *calidad*.

Un *proceso de software* es el conjunto de actividades, métodos, prácticas y tecnologías aplicables a todos los proyectos de software, un proceso básico (también conocido como ciclo de vida básico) está conformado por el análisis, diseño, codificación, pruebas y mantenimiento.

Los *métodos* o *modelos* de la ingeniería de software indican cómo realizar los pasos necesarios del ciclo de vida, así pues está el modelo de construcción de prototipos, el modelo de desarrollo rápido de aplicaciones, el de procesos evolutivos (que se divide en modelo incremental, en espiral, de ensamblaje de componentes y de desarrollo concurrente), el modelo de métodos formales y por último las técnicas de cuarta generación.

Las *herramientas* ayudan a organizar tareas de trabajo, controlar y supervisar los progresos y administrar la calidad técnica, su objetivo principal es proporcionar un soporte automático o semiautomático para los procesos y los métodos.

[1] Ingeniería de Software-Roger S. Pressman-P XXIX

Figura 1.1. Capas de la Ingeniería del Software.

El trabajo que se asocia a la ingeniería de software se puede dividir en tres fases genéricas, con independencia del área de aplicación, tamaño o complejidad del proyecto [**PrR2002**]:

En la *fase de definición* se identifica qué información ha de ser procesada, que función y rendimiento se desea, que comportamiento del sistema, que interfaces van a ser establecidas, que restricciones de diseño existen y que criterios de validación se necesitan para definir un sistema correcto.

En la *fase de desarrollo* se define cómo han de diseñarse las estructuras de datos, cómo ha de implementarse una función dentro de una arquitectura de software, como han de implementarse los detalles procedimentales, cómo han de caracterizarse las interfaces, cómo ha de traducirse el diseño en un lenguaje de programación y cómo han de realizarse las pruebas.

La *fase de mantenimiento* se centra en el cambio que va asociado a la corrección de errores, a las adaptaciones requeridas a medida que evoluciona el entorno del software y a los cambios debidos de las mejoras producidas por los requisitos cambiantes del cliente.

1.1.1. Modelos Genéricos del Desarrollo de Software

Para resolver los problemas reales de una empresa u organización, un ingeniero de software debe incorporar una estrategia de desarrollo que acompañe al proceso, métodos y herramientas ya descritas, así como las fases genéricas.

La ingeniería de software tiene varios modelos o paradigmas de desarrollo en los cuales se puede apoyar para la realización de software, de los cuales podemos destacar a éstos por ser los más utilizados y los más completos:

- Modelo en cascada o Clásico (modelo tradicional)
- Modelo en espiral (modelo evolutivo)
- Modelo de prototipos
- Desarrollo por etapas
- Desarrollo iterativo y creciente o Iterativo Incremental
- RAD (Rapid Application Development)

MODELO EN CASCADA O CLASICO

Es un enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Un ejemplo de una metodología de desarrollo en cascada es [**PrR2002**]:

1. Análisis de requisitos
2. Diseño del Sistema
3. Diseño del Programa
4. Codificación
5. Pruebas
6. Implantación
7. Mantenimiento

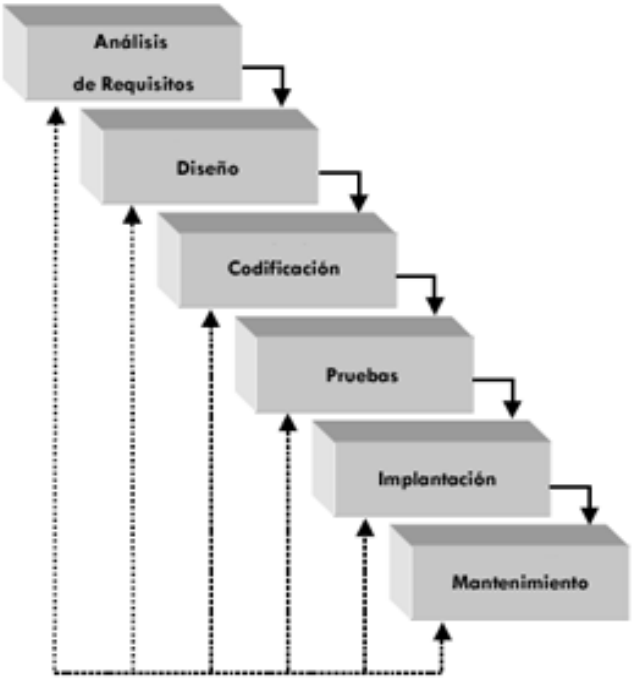


Figura 1.2. Modelo en Cascada.

Cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costes del desarrollo. La palabra *cascada* sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto (ver **figura 1.2**). A continuación se describe cada una de las fases:

Análisis de requisitos

Se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. En esta etapa se debe consensuar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas. Como resultado al concluir esta etapa surge el SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Diseño del Sistema

Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras.

Diseño del Programa

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

Codificación

Es la fase de programación o implementación propiamente dicha. Aquí se implementa el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores.

Pruebas

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de ser puesto en explotación.

Implantación

El software obtenido se pone en producción. Se implantan los niveles software y hardware que componen el proyecto. La implantación es la fase con más duración y con más cambios en el ciclo de elaboración de un proyecto. Es una de las fases finales del proyecto
Durante la explotación del sistema software pueden surgir cambios, bien para corregir errores o bien para introducir mejoras. Todo ello se recoge en los Documentos de Cambios.

MODELO EN ESPIRAL O MODELO EVOLUTIVO

Las actividades de este modelo son una espiral, cada bucle es una actividad. Las actividades no están fijadas a prioridad, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior [**PrR2002**].

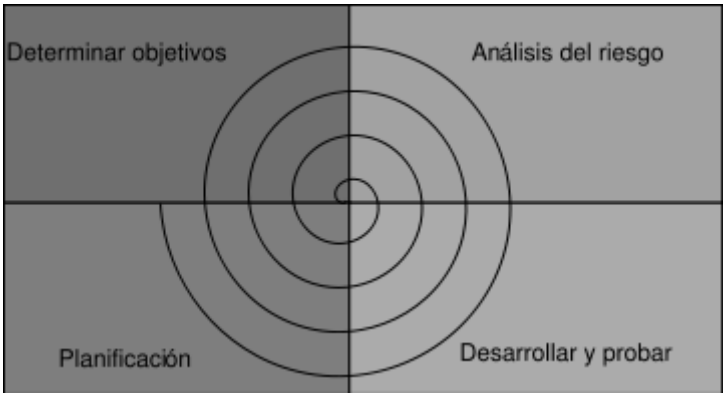


Figura 1.3. Modelo en Espiral.

Para cada actividad habrá cuatro tareas (ver **figura 1.3**):

Determinar o fijar objetivos

Fijar los productos definidos a obtener: requerimientos, especificación, manual de usuario, restricciones. Identificar riesgos del proyecto y estrategias alternativas para evitarlos.

Análisis del riesgo

Se estudian todos los riesgos potenciales y se seleccionan una o varias alternativas propuestas para reducir o eliminar los riesgos.

Desarrollar, verificar y validar (probar)

Se realizan las tareas de la actividad propia y de prueba, se realiza un análisis de alternativas e identificación resolución de riesgos, y dependiendo de estos resultados, se elige un modelo para el desarrollo, el que puede ser cualquiera de los otros existentes (formal, evolutivo, cascada, etc.)

Planificar

Se revisa todo lo hecho, evaluándolo, y con ello se decide si se continúa con las fases siguientes y se planifica la próxima actividad.

MODELO DE PROTOTIPOS O DE DESARROLLO EVOLUTIVO

Este desarrollo se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado (en forma de un diseño rápido) [**PrR2002**].

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final. El diseño rápido conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La iteración ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

MODELO DE DESARROLLO POR ETAPAS

Este modelo es similar al Modelo de prototipos ya que se muestra al cliente el software en diferentes estados sucesivos de desarrollo, se diferencia en que las especificaciones no son conocidas en detalle al inicio del proyecto y por tanto se van desarrollando simultáneamente con las diferentes versiones del código [**PrR2002**].

Pueden distinguirse las siguientes fases:

- Especificación conceptual
- Análisis de requerimientos
- Diseño inicial
- Diseño detallado, codificación, depuración y liberación

Estas diferentes fases se van repitiendo en cada etapa del diseño.

MODELO DE DESARROLLO ITERATIVO Y CRECIENTE (O INCREMENTAL)

Este modelo es un proceso de desarrollo de software, creado en respuesta a las debilidades del modelo tradicional de cascada. Para apoyar el desarrollo de proyectos por medio de este modelo se han creado Framework (entornos de trabajo), de los cuales los dos más famosos son el

Rational Unified Process y el Dynamic Systems Development Method. El desarrollo incremental e iterativo es también una parte esencial de un tipo de programación conocido como Extreme Programming y los demás frameworks de desarrollo rápido de software**[PrR2002]**.

La idea principal detrás de mejoramiento iterativo es desarrollar un sistema de programas de manera incremental, permitiéndole al desarrollador sacar ventaja de lo que se ha aprendido a lo largo del desarrollo anterior, incrementando, versiones entregables del sistema. El aprendizaje viene de dos vertientes: el desarrollo del sistema, y su uso (mientras sea posible). Los pasos claves en el proceso eran comenzar con una implementación simple de los requerimientos del sistema, e iterativamente mejorar la secuencia evolutiva de versiones hasta que el sistema completo esté implementado. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema.

El proceso en sí mismo consiste de:

- Etapa de inicialización
- Etapa de iteración
- Lista de control de proyecto

Etapa de inicialización

Se crea una versión del sistema. La meta de esta etapa es crear un producto con el que el usuario pueda interactuar, y por ende retroalimentar el proceso. Debe ofrecer una muestra de los aspectos claves del problema y proveer una solución lo suficientemente simple para ser comprendida e implementada fácilmente.

Etapa de iteración

Esta etapa involucra el rediseño e implementación de una tarea de la lista de control de proyecto, y el análisis de la versión más reciente del sistema. La meta del diseño e implementación de cualquier iteración es ser simple, directa y modular, para poder soportar el rediseño de la etapa o como una tarea añadida a la lista de control de proyecto. El código puede, en ciertos casos, representar la mayor fuente de documentación del sistema. El análisis de una iteración se basa en la retroalimentación del usuario y en el análisis de las funcionalidades disponibles del programa. Involucra el análisis de la estructura, modularidad, usabilidad, confiabilidad, eficiencia y eficacia (alcanzar las metas).

Lista de control del proyecto

Esta se modifica bajo la luz de los resultados del análisis.

DESARROLLO RAPIDO DE APLICACIONES (RAD)

Rapid Application Development (**RAD**), es un proceso de desarrollo de software. El método comprende el desarrollo iterativo, la construcción de prototipos y el uso de utilidades CASE (Computer Aided Software Engineering). Tradicionalmente, el desarrollo rápido de aplicaciones tiende a englobar también la usabilidad, utilidad y la rapidez de ejecución.

Hoy en día se suele utilizar para referirnos al desarrollo rápido de GUIs tal como Glade, o IDEs de desarrollo completas como Delphi, Foxpro o Anjuta. Algunas de las herramientas más usadas para hacer aplicaciones rápidamente son **[MeW2009]**:

Herramientas RAD Multiplataforma

- NetBeans
- WaveMaker WaveMaker Herramienta Open Source para aplicaciones Web2.0

- Revolution Studio Es una avanzada herramienta cross-platform RAD que deriva ejecutables sobre Windows, Linux, Solaris, MacOS X Universal Binary and MacOS Classic.
- Lazarus Es un IDE cross-platform similar a Borland Delphi.
- Real Basic Es un IDE cross-platform similar a Visual Basic.
- Microsoft Visual Studio 2005

Herramientas RAD para Escritorio

- Borland C++Builder
- Borland Delphi
- Visual Basic
- Ultimate++

Herramientas RAD para Bases de Datos

- Oracle Forms
- Oracle Application Express o APEX
- Sybase PowerBuilder
- Microsoft Access

Herramientas RAD Orientadas a la WEB

- Adobe ColdFusion
- Symfony (PHP)
- WebDev
- Microsoft Visual Studio 2005
- ycube RAD Plus Herramienta Open Source para aplicaciones comerciales.
- NConstruct - .NET RAD herramienta para Winforms & Web

1.1.2. Notación UML (Proceso RUP)

A lo largo de los años, el desarrollo de los proyectos de software causan bastantes confusiones y malas interpretaciones en los requerimientos de los clientes y usuarios, en parte debido a la abundancia de notaciones, metodologías y conceptos que hace que los desarrolladores de sistemas no se pongan de acuerdo en que es lo que realmente están elaborando. En un esfuerzo para estandarizar las notaciones y procesos a utilizar, se conformó un consorcio liderado por la empresa Rational y por las principales empresas del mundo de la industria de la informática, entre ellas, Microsoft, Oracle, Sun Microsystems, Intellicorp, IBM, AMD y otras, quienes desarrollaron una notación llamada UML y el proceso de desarrollo RUP.

1.1.2.1. La Ingeniería de Software

La ingeniería del Software nace como una disciplina para aplicar los principios técnicas y herramientas de desarrollo de software, surgió porque todos los desarrolladores en la década de los 80's, realizaban el software de forma artística, es decir utilizando métodos y técnicas adhoc donde la experiencia (el ensayo-error) era el camino a seguir. Este enfoque produjo grandes y exitosos productos de programación pero conforme los proyectos se volvieron más complejos debido al avance del hardware y software y la penetración cada vez mayor de la informática en todos los ámbitos de la sociedad, llevó a que se produjera software sin calidad, se incumplieran los presupuestos y se incrementara dramáticamente los costos de mantenimiento.

La solución propuesta fue aplicar métodos y principios que han sido utilizados y probados en la experiencia de desarrollo de software para producir de forma inequívoca productos que corran eficientemente y se ejecuten sobre máquinas reales. En la década de los 70 surgieron una gran variedad de metodologías basadas en los principios de la programación estructurada. En los 80's y 90's el paradigma estructurado evolucionó hacia el paradigma orientado a objetos, en el período de 1989 y 1994 se creó la llamada guerra de métodos dentro de la comunidad orientada a objetos existiendo un incremento de menos de diez a más de cincuenta metodologías, es así que los desarrolladores de software quedaron muy confundidos sin saber cuál era la metodología más adecuada para elaborar sus proyectos.

Ante lo enunciado, el UML oficialmente se presentó cuando Rumbaugh, Booch y Jacobson unifican sus estudios con una semántica y notación, para lograr compatibilidad en el análisis y diseño orientado a objetos, permitiendo que los proyectos se asentaran en un lenguaje de modelado maduro, permitiendo a los constructores de herramientas enfocarse en producir características más útiles.

1.1.2.2. La complejidad del Software

Al observar sistemas complejos sociales como una gran empresa, los naturales como el universo y los sistemas creados por el hombre como el computador, se observa que exhiben una jerarquía de clases (conceptos) y otra de objetos (instancias). En una empresa donde conjuntos de personas forman un departamento y un conjunto de departamentos forman divisiones se describe la forma canónica de un sistema complejo que exhibe dos jerarquías: Una jerarquía de clases y otra jerarquía de objetos, donde cada objeto es una instancia de la una clase. Este es el modelo del cual se apropia el análisis y diseño orientado a objetos para desarrollar sistemas donde hay gran cantidad de software. (Ver **Figura 1.4**)

1.1.2.3. Principios de Modelado

En cualquier proyecto de ingeniería como la construcción de un gran edificio, un avión, una represa hidroeléctrica, la construcción de un procesador de textos o un software de comunicaciones para Internet, requieren de etapas de modelamiento que permitan experimentar y visualizar el sistema que se construirá. De la experiencia en ingeniería se extractan los siguientes principios de modelado [**MaA2001**]:

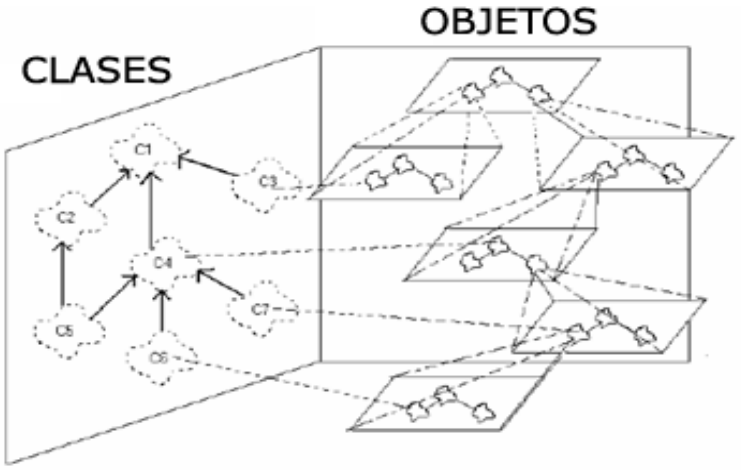


Figura 1.4. Forma Canónica de un Sistema Complejo

a) La forma como vemos el problema tiene una profunda influencia en forma como acometemos el problema y le damos solución al mismo.

Si pensamos que el mundo está compuesto de clases (Abstracciones de la realidad y de la solución del problema) y objetos (instancias de éstas abstracciones) que interactúan entre sí para realizar una funcionalidad, así veremos el mundo. Este es precisamente al paradigma a que le apuesta UML: el modelo orientado a objetos. Si vemos la realidad como compuesta de procesos donde cada uno a su vez se puede descomponer en subprocesos entonces estamos concibiendo la realidad según el modelo estructurado y la arquitectura del sistema en desarrollo estará conformada de programas y subprogramas.

b) Para modelar un sistema complejo no es suficiente un único modelo se requieren múltiples modelos donde cada uno representa una vista (aspecto) del sistema; estos modelos se complementan entre sí.

Esta es la razón de la existencia de varios diagramas en UML que modelan diferentes aspectos del sistema, desde las vistas lógicas y físicas del sistema hasta los aspectos dinámicos, estáticos y funcionales del mismo.

c) Cualquier modelo puede ser representado con diferentes grados de precisión.

La precisión se puede ver desde dos ópticas: La primera es el grado de detalle con que se representa un modelo; por ejemplo, si lo que se desea es razonar acerca de los requerimientos del sistema con un cliente o usuario final, se puede elaborar un diagrama de clases que muestra las clases, sus atributos y operaciones así como varios adornos(multiplicidad) en las relaciones; por otro lado, si lo que se desea es transmitir el diagrama de clases para que sea implementado en un DBMS (Data Base Management System, Sistema Administrador de Bases de Datos) por un programador, el diagrama con toda seguridad contendrá la visibilidad de las características (atributos y operaciones) de las clases, los tipos de datos de los atributos y las signaturas de las métodos de las clases.

La segunda forma de ver la precisión de un modelo se refiere al nivel de abstracción, ese decir, a los detalles y la vista (porción del sistema o realidad) que presenta un modelo al lector; por ejemplo, en un sistema Bancario que maneja los retiros que hacen los clientes ya sea en un cajero automático o humano, el diagrama de clases contiene decenas de éstas; sin embargo las personas encargadas de desarrollar la interfaz de un cajero electrónico estarían interesadas en las clases necesarias para realizar el comportamiento del cajero y omiten el resto de clases del sistema.

d) Los mejores Modelos están ligados a la realidad.

El símbolo de un actor en un diagrama de casos de uso representa, de hecho, un actor en el sistema real; así como un componente en un diagrama de componentes representa un componente físico del software. Cada elemento de UML como una clase, objeto, estado, componente o nodo tiene su correspondencia con algún elemento conceptual o físico del mundo real.

1.1.2.4. El Lenguaje de Modelado Unificado UML

"El Lenguaje de Modelado Unificado UML es un lenguaje estándar para escribir planos de software. UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software" **[BoG1999].**

El UML es el Lenguaje de Modelado Unificado Orientado a Objetos, UML no es un método porque no tiene noción de proceso el cual es una parte importante de un método. Ahora bien si UML no es método; entonces ¿Cuáles son las etapas a seguir en el desarrollo de sistemas con UML?, varios especialistas en desarrollo de sistemas de información arguyen de que existe la necesidad de

adoptar un Proceso de Desarrollo de sistemas para enmarcar las fases importantes que sigue el UML, por ello los desarrolladores de proyectos de sistemas de información emplean el **Procesos Unificado** para dar soluciones adecuadas a las necesidades de los clientes.

El desarrollo de sistemas con UML siguiendo el proceso unificado incluye actividades específicas, cada una de ellas a su vez contienen otras subactividades las cuales sirven como una guía de cómo deben ser las actividades desarrolladas y secuenciadas con el fin de obtener sistemas exitosos; consecuentemente el desarrollo de los sistemas puede variar de desarrollador en desarrollador, de proyecto en proyecto, de empresa en empresa adoptando siempre un Proceso de Desarrollo (ver **Figura 1.5**).

1.1.2.5. El proceso Unificado de Modelado (RUP)

A través de la historia se han desarrollado varios modelos de proceso de software (paradigmas de desarrollo) cada uno con sus ventajas, desventajas y utilidad en algunos tipos de proyectos y problemas. Al igual que cualquier notación, el proceso unificado actúa como un modelo que puede adaptarse a cualquier tipo de proyecto y empresa (grandes y pequeñas). Las características del proceso unificado de modelado son [**BoG1999**]:

- **Centrado en los Modelos:** Los diagramas son un vehículo de comunicación más expresivo que las descripciones en lenguaje natural. Se trata de minimizar el uso de descripciones y especificaciones textuales del sistema.
- **Guiado por lo casos de uso:** Los casos de uso son el instrumento para validar la arquitectura del software y extraer los casos de prueba.
- **Centrado en la arquitectura:** Los modelos son proyecciones del análisis y el diseño constituye la arquitectura del producto a desarrollar.
- **Iterativo e incremental:** Durante todo el proceso de desarrollo se producen versiones incrementales (que se acercan al producto terminado) del producto en desarrollo.

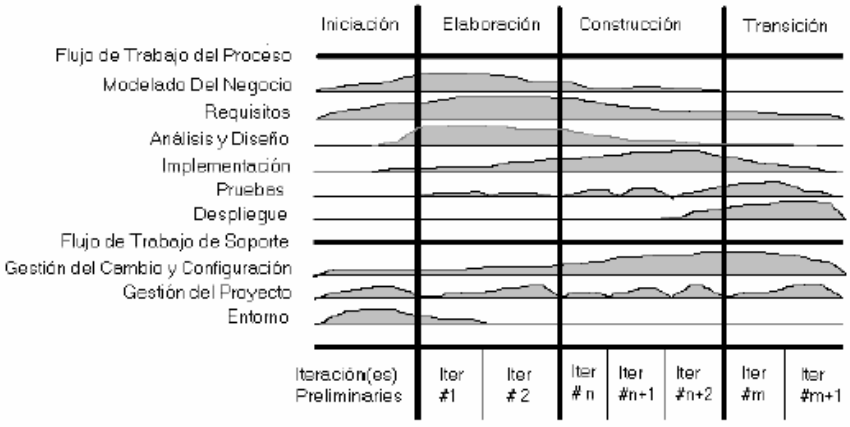


Figura 1.5. El Proceso de Modelado Unificado

El gráfico que representa el RUP incluye las cuatro etapas importantes que son: la **iniciación, elaboración, construcción y transición,** las cuales muestran que para producir una versión del producto en desarrollo se aplican todas las actividades de ingeniería pero con diferente énfasis; en las versiones preliminares, como además indica la intuición, hay más énfasis en actividades de modelado del negocio, requisitos, análisis y diseño; conforme se producen versiones el énfasis pasa a las actividades de implementación, pruebas y despliegue.

1.1.2.6. Diagramas de UML

Los elementos de UML se muestran mediante diagramas que presentan múltiples vistas del sistema, ese conjunto de vistas son conocidos como ***modelos***.

UML presenta varios diagramas donde cada uno representa un aspecto del sistema. De ahí que varios investigadores según sus criterios y puntos de vista mencionan qué diagramas emplear en el desarrollo de los sistemas de información; sin mencionar cuáles son los diagramas más adecuados en las distintas etapas de desarrollo del Proceso Unificado, viendo esta necesidad, Aleida Barrientos [**BaA2009**] propone un conjunto de diagramas necesarios para cada etapa según la complejidad del sistema de información a solucionar.

Dado un sistema a desarrollar no es necesario emplear todos los diagramas; para sistemas sencillos un diagrama de clases junto con un par de diagramas de actividades e interacción sería suficiente, asimismo si los sistemas son complejos requieren de la utilización de más diagramas, debido a que requieren de etapas incrementales e iterativas(ciclos de desarrollo) en el análisis, diseño e implementación, por ello es que el conjunto actividades deberá especificar la etapa de desarrollo y los diagramas recomendados como muestra la **figura 1.6.**

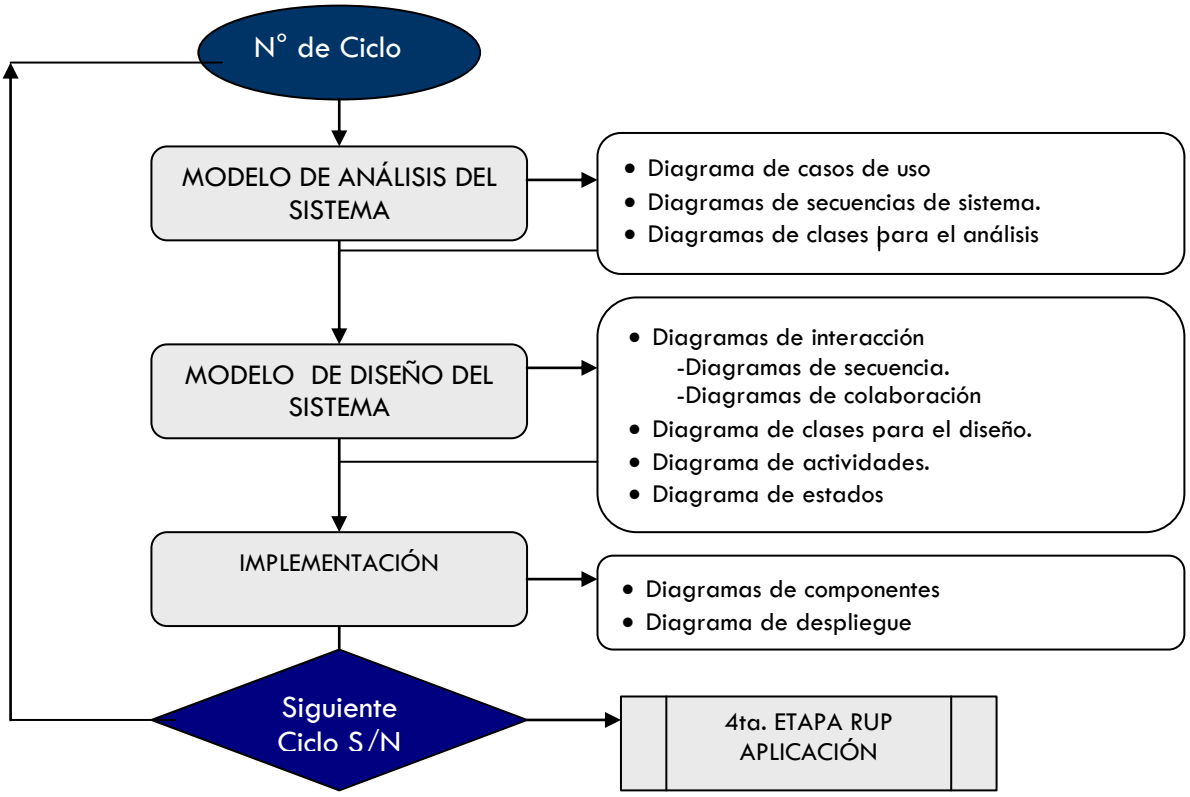


Figura 1.6. Diagramas recomendados en el desarrollo de los sistemas de información

1.1.3. Ingeniería Web

La WWW (que se describe en el apartado 1.4 del presente capítulo) e Internet han introducido a la población en general dentro del mundo de la informática. Se puede decir que Internet y la Web (ver apartado 1.4) son los avances más importantes en la historia de la informática, ya que estas tecnologías informáticas nos han llevado a la era de la informática.

Con objeto de evitar una Web enmarañada y lograr un mayor éxito en el desarrollo y aplicación de sistemas basados en Web complejos y a gran escala, existe una necesidad apremiante de métodos y herramientas nuevos para el desarrollo, empleo y evaluación de sistemas y aplicaciones basados en Web. Tales métodos y herramientas deberán tener en cuenta las características especiales en el medio nuevo, en los entornos y escenarios operativos, y en la multiplicidad de perfiles de usuario implicando todo ello un reto adicional para el desarrollo de aplicaciones basadas en Web.

El proceso de ingeniería de Web comienza con la formulación, que es una actividad que identifica las metas y los objetivos de la WebApp (apartado 1.1.3.1.). La planificación estima el coste global del proyecto, evalúa los riesgos asociados con el esfuerzo del desarrollo y define una programación temporal del desarrollo. El análisis establece requisitos técnicos e identifica los objetos del contenido que se incorporarán a la WebApp. La actividad de ingeniería incorpora dos tareas paralelas: diseño del contenido y diseño técnico. La generación de páginas es una actividad de construcción que hace uso extenso de herramientas automatizadas para la creación de WebApps, y la comprobación ejercita la navegación de la WebApp, intentando descubrir errores de la función y el contenido, y asegurando mientras tanto que la WebApp funcione correctamente en diferentes entornos.

1.1.3.1. Atributos de las Aplicaciones Basadas en Web

Las aplicaciones basadas en Web son también conocidas como **WebApps**. Una aplicación Web abarca desde una página Web simple hasta sitios Web completos que residen en Internet, en una Intranet o en una Extranet.

Powell dice que los sistemas basados en Web implican una mezcla de publicación impresa y desarrollo de software, de marketing, de informática, de comunicaciones internas y relaciones externas, de arte y de tecnología. Los siguientes atributos se van a encontrar en la gran mayoría de las WebApps **[PrR2002]**:

- *Intensivas de Red:* una WebApp reside en una red y debe dar servicio a las necesidades de una comunidad diversa de clientes. Una WebApp puede residir en Internet, haciendo posible una comunicación abierta por todo el mundo; o bien, se puede ubicar en una Intranet, implementando una comunicación a través de redes de una organización, o Extranet para la comunicación entre redes.
- *Controlada por el Contenido:* en la mayoría de los casos una WebApp utiliza hipermedia[2] para presentar al usuario el contenido de textos, gráficos, sonido y video.
- *Evolución Continua:* a diferencia del software de aplicaciones convencionales, que evolucionan con una serie de versiones planificadas y cronológicamente espaciadas, las aplicaciones Web están en constante evolución, lo que permite que un sitio Web crezca en robustez e importancia para servir y adaptarse a las necesidades de muchos usuarios.

1.1.3.2. Características de las Webapps

Para conducir el proceso de desarrollo de las WebApps y su cont{inua evolución, se utilizan sus siguientes características**[PrR2002]:**

[2] **Hipermedia** es el término con que se designa al conjunto de métodos o procedimientos para escribir, diseñar, o componer contenidos que tengan texto, video, audio, mapas u otros medios, y que además tenga la posibilidad de interactuar con los usuarios.

Inmediatez: el tiempo que se tarda en comercializar un sitio Web completo puede ser cuestión de dias o semanas, así que los desarrolladores deberán utilizar métodos de planificación, análisis, diseño, implementación y comprobación que se hayan adaptado a planificaciones apretadas en tiempo para el desarrollo de WebApps.
Seguridad: debido a que las WebApps están disponibles a través del acceso a la red, es difícil limitar la población de usuarios finales que puedan acceder a la aplicación. Por tal motivo, deberán implementarse fuertes medidas de seguridad dentro de la aplicación de la WebApp y en toda la infraestructura que la apoya, esto para proteger el contenido confidencial y para proporcionar formas seguras de transmisión de datos.
Estética: cuando se diseña una aplicación con el fin de comercializarse o vender productos o ideas, la estética tiene mucho que ver con el éxito del diseño técnico, ya que una parte del atractivo de una WebApp es su apariencia e interacción.

1.1.3.3. Categorías de Aplicación para las Webapps

Dentro del trabajo de la Web, podemos encontrar distintas categorías de aplicaciones para las WebApps, las cuales representan hechos reales para los ingenieros de la Web [**PrR2002**]:

Informativa: se proporciona un contenido de solo lectura con navegación y enlaces simples.
Descarga: un usuario descarga la información desde el servidor apropiado.
Personalizable: el usuario personaliza el contenido de sus necesidades específicas.
Interacción: la comunicación entre una comunidad de usuarios ocurre mediante un espacio chat (charla), tablones de anuncio (tagboards) o mensajería instantánea.
Entrada del Usuario: la entrada basada en formularios es el mecanismo primario de la necesidad de comunicación.
Orientada a transacciones: el usuario hace una solicitud que es cumplida por la WebApp.
Orientado a servicios: la aplicación proporciona un servicio al usuario.
Portal: la aplicación canaliza al usuario llevándolo a otros contenidos o servicios Web fuera del dominio de la aplicación del portal.
Acceso a bases de datos: el usuario consulta en una base de datos grande y extrae información.

1.2. Bases de Datos

Una **base de datos** o **banco de datos** es un conjunto de datos pertenecientes al un mismo contexto y almacenados sistemáticamente para su posterior uso. En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital (electrónico), que ofrece un amplio rango de soluciones al problema de almacenar datos.

Una base de datos está conformada por tablas, registros y campos. **Tabla** se refiere al tipo de modelamiento de datos, donde se guardan los datos recolectados por un programa. Su estructura general se asemeja a la vista general de un programa de Hoja de cálculo.
Las tablas se componen de dos estructuras[**Lyc2008**]:

Campo: Corresponde al nombre de la columna. Debe ser único y además de tener un tipo de dato asociado.
Registro: Corresponde a cada fila que compone la tabla. Allí se componen los datos y los registros. Eventualmente pueden ser nulos en su almacenamiento.

En la actualidad las bases de datos ocupan un papel muy importante en la organización de información para el mejor funcionamiento de las organizaciones, y se han convertido en uno de los principales componentes del software.

1.2.1. Sistema Gestor de Bases de Datos

Los **sistemas de gestión de base de datos** (**SGBD**); (en inglés: *Database management system*, abreviado **DBMS**) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, entendiendo por *aplicación* un tipo de programa informático diseñado para facilitar al usuario la realización de un determinado tipo de trabajo. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de datos. De manera conceptual funciona de la siguiente forma[**MeW2009**]:

1. El usuario emite una petición de acceso, utilizando algún sublenguaje de datos específico (por ejemplo SQL).
2. El DBMS intercepta esa petición y la analiza.
3. El DBMS inspecciona, en su momento, el esquema externo para ese usuario, la transformación externa correspondiente, el esquema conceptual, la transformación interna y la definición de la estructura de almacenamiento.
4. El DBMS ejecuta las operaciones necesarias sobre la base de datos almacenada.

En la **figura 1.7** se muestra el esquema de un DBSM.

En la estructura básica de un Sistema de Gestión de Base de Datos se enuncian dos lenguajes que permiten trabajar sobre la base de datos. Estos lenguajes estándar son [**MeW2009**]:

DDL (Data Definition language): Lenguaje de Definición de Datos. Por medio de este el DBMS identifica las descripciones de los elementos de los esquemas y almacena la descripción del esquema en el catálogo del DBMS. Por medio de este el DBMS especifica el esquema conceptual e interno (Base de datos almacenada).
SDL (Store Definition language): Lenguaje de definición de almacenamiento. Es utilizado por el DBMS para especificar el esquema interno que corresponde a la Base de Datos Almacenada.
VDL (View Definition language): Lenguaje de Definición de Vistas. Es utilizado por el DBMS para especificar las vistas del usuario y sus correspondencias con el esquema conceptual.
DML (Data Manipulation language): Lenguaje de Manipulación de Datos. Permite la manipulación de las operaciones de Inserción, Eliminación y Modificación.

Existen diversos DBMS disponibles en el mercado, por ejemplo [**MeW2009**]:

LIBRES (con licencias de dominio público)

- PosgreSQL
- MySQL
- Apache Derby

GRATUITOS

- Microsoft SQL Server Compact Edition

- Sybase ASE Express Edition para Linux

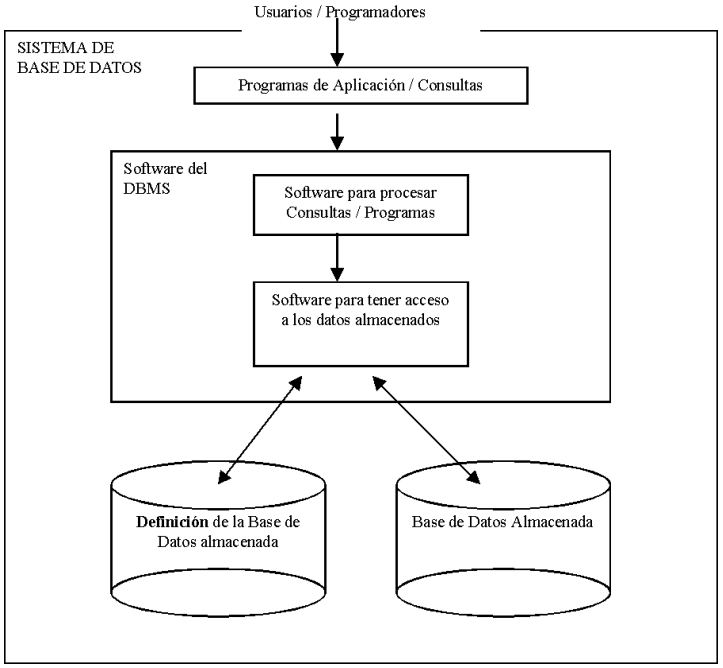


Figura 1.7. Esquema de un DBSM

COMERCIALES

- Fox Pro
- Microsoft Access
- Microsoft SQL Server
- Paradox
- Progress

1.2.2. Sistema de Bases de Datos

Un sistema de base de datos es algo más que simples datos o que un conjunto de datos en combinación con unos programas de gestión. Un sistema de base de datos está formado por los siguientes componentes [**Lyc2008**]:

Datos

Las características más importantes de la información en estos sistemas es que va a estar integrada y compartida.

Integrada: La Base de datos puede considerarse como una unificación de varios ficheros de datos, que son tratados como uno solo, y en el que se ha eliminado totalmente, o en parte, la redundancia de datos.

Compartida: Los datos pueden compartirse entre varios usuarios distintos. Es posible que varios de estos usuarios accedan al mismo tiempo al mismo elemento de información (acceso concurrente).

Equipo (Hardware)

Conjunto de dispositivos físicos utilizados para almacenar y procesar los datos.

Computadoras utilizadas para procesar los datos de la base de datos: pueden ser mainframe, miniordenador u ordenador personal. El mainframe y los miniordenadores fueron utilizados tradicionalmente para soportar el acceso de varios usuarios a una base de datos común. Los ordenadores personales eran empleados, inicialmente, para manejar bases de datos

autónomas controladas y manipuladas por un usuario único. No obstante, actualmente, también pueden conectarse a una red *cliente/servidor,* garantizando el acceso de varios usuarios a una base de datos común almacenada en unidades de disco y controladas por un ordenador servidor. El servidor puede ser otro ordenador personal más potente, o bien, un miniordenador o un mainframe.

Volúmenes de almacenamiento. Generalmente son unidades de disco que constituyen el mecanismo de almacenamiento principal para las bases de datos.

Otros dispositivos, como unidades de cinta, terminales, impresoras, etc.

Programas (Software)

Un sistema de base de datos incluye dos tipos de programas:

El *software de propósito general,* para la gestión de la base de datos, comúnmente llamado *Sistema Gestor de Bases de Datos* (apartado 1.2.1.). El SGBD maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios y los programas de aplicación.

El *software de aplicación,* que usa las facilidades del SGBD para manipular la base de datos con el fin de llevar a cabo una función específica en la gestión de la empresa (por ejemplo: la gestión de almacén). Puede estar desarrollado en un lenguaje de programación estándar, tal como COBOL o C, o en un lenguaje propio de los SGBD denominados lenguajes de cuarta generación (4GL).

Personal

En un sistema de base de datos intervienen un número importante de usuarios, que podemos clasificar en tres grupos:

Administrador de la base de datos (ABD.) Son los encargados de diseñar la estructura de la base de datos y los responsables de que el sistema funcione correctamente. El ABD se encarga de autorizar el acceso a la base de datos, de coordinar y vigilar su utilización y de adquirir los recursos necesarios de software y hardware. Es el responsable cuando surgen problemas como violaciones de seguridad o una respuesta lenta del sistema. El ABD tiene, entre otras, las siguientes funciones:

- Definición del esquema: Decidir el contenido de la base de datos, eligiendo cuales son los datos que interesa tener almacenados y organizarlos de la mejor forma posible, creando el esquema conceptual, que se escribirá mediante un lenguaje de definición de datos (DDL).
- Definición de las estructuras de almacenamiento y método de acceso: Debe decidir sobre la forma en que se van a almacenar los datos sobre los soportes físicos en los que se grabará la base de datos y la correspondencia entre esta estructura de almacenamiento y el esquema conceptual.
- Modificación del esquema y de la organización física si los requerimientos cambian.
- Decidir los controles de autorizaciones para el acceso a los datos: Es el que concede diferentes tipos de autorizaciones al resto de los usuarios de la base de datos.
- Especificar las restricciones de integridad: Debe definir los procedimientos de validación que habrán de ejecutarse cada vez que se actualiza la base de datos. Estas restricciones son consultadas por el SGBD cada vez que se realiza una actualización de los datos.

Programadores de aplicaciones, que se encargan de desarrollar las aplicaciones que manejan datos de la base de datos. Estas aplicaciones contendrán solicitudes de datos al SGBD que luego serán procesados por los programas de la aplicación que tendrán como finalidad resolver problemas específicos de la empresa.

Usuarios finales, que son personas que no tienen por que tener conocimientos informáticos y que pueden manipular los datos (examinarlos y actualizarlos) con la ayuda de las aplicaciones, o bien de lenguajes de consulta no procedimentales (no es necesario indicar el algoritmo de acceso a los datos), tipo SQL, o bien, mediante herramientas basadas en sistemas de menús. Se distinguen tres tipos de usuarios finales:

- Usuarios especializados: Aquellos que son capaces de escribir ciertas aplicaciones para la BD, para su uso propio.
- Usuarios casuales: Aquellos que realizan consultas a través de un procesador de consultas. Esas consultas pueden ser creadas por ellos mismos o por otras personas.
- Usuarios ingenuos: Aquellos que solo acceden a través de aplicaciones previamente escritas por otros usuarios.

1.2.3. Modelos de Datos Conceptuales

Un *modelo de datos* es una serie de conceptos que puede utilizarse para describir un conjunto de datos y las operaciones para manipularlos. Hay dos tipos de modelos de datos [**MaA2001**]: los modelos conceptuales y los modelos lógicos. Los ***modelos conceptuales*** se utilizan para representar la realidad a un alto nivel de abstracción. Mediante los modelos conceptuales se puede construir una descripción de la realidad fácil de entender. En los ***modelos lógicos***, las descripciones de los datos tienen una correspondencia sencilla con la estructura física de la base de datos.

El diseño de bases de datos se descompone en tres etapas: diseño conceptual, diseño lógico y diseño físico. El diseño conceptual es el proceso por el cual se construye un modelo de la información que se utiliza en una empresa u organización, independientemente del SGBD que se vaya a utilizar para implementar el sistema y de los equipos informáticos o cualquier otra consideración física.

Un modelo conceptual es un conjunto de conceptos que permiten describir la realidad mediante representaciones lingüísticas y gráficas. Los modelos conceptuales deben poseer una serie de propiedades: expresividad, simplicidad, minimalidad y formalidad.

El modelo conceptual más utilizado es el **modelo entidad-relación**, que posee los siguientes conceptos: entidades, relaciones, atributos, dominios de atributos, identificadores y jerarquías de generalización.

En el diseño de bases de datos se usan primero los modelos conceptuales para lograr una descripción de alto nivel de la realidad, y luego se transforma el esquema conceptual en un esquema lógico. El motivo de realizar estas dos etapas es la dificultad de abstraer la estructura de una base de datos que presente cierta complejidad. Un *esquema* es un conjunto de representaciones lingüísticas o gráficas que describen la estructura de los datos de interés.

Los modelos conceptuales deben ser buenas herramientas para representar la realidad, por lo que deben poseer las siguientes cualidades:

Expresividad: deben tener suficientes conceptos para expresar perfectamente la realidad.
Simplicidad: deben ser simples para que los esquemas sean fáciles de entender.
Minimalidad: cada concepto debe tener un significado distinto.
Formalidad: todos los conceptos deben tener una interpretación única, precisa y bien definida.

1.2.3.1. El modelo entidad-relación

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Fue introducido por Peter Chen en 1976. El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas y lingüísticas [**Hal1994**].

Originalmente, el modelo entidad-relación sólo incluía los conceptos de entidad, relación y atributo. Más tarde, se añadieron otros conceptos, como los atributos compuestos y las jerarquías de generalización, en lo que se ha denominado *modelo entidad-relación extendido*. (Ver **Figura 1.8**)

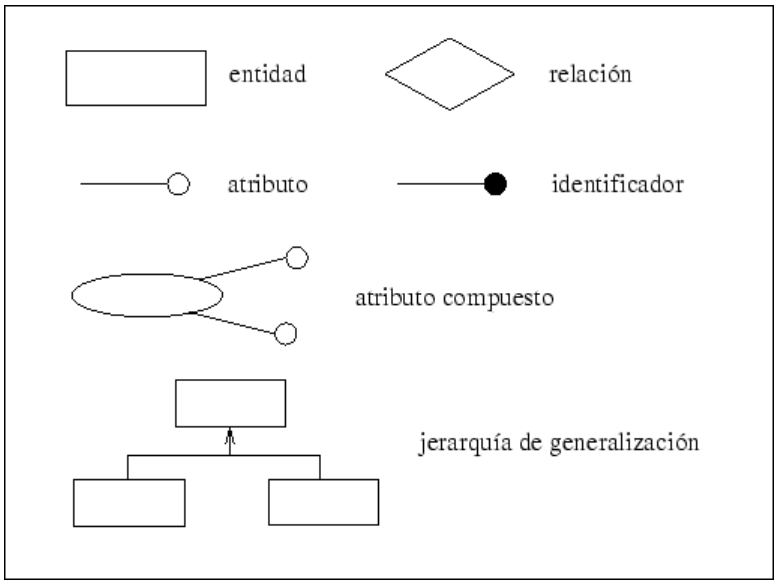


Figura 1.8. Conceptos del modelo entidad-relación extendido.

Entidad

Cualquier tipo de objeto o concepto sobre el que se recoge información: cosa, persona, concepto abstracto o suceso. Por ejemplo: coches, casas, empleados, clientes, empresas, oficios, diseños de productos, conciertos, excursiones, etc. Hay dos tipos de entidades: fuertes y débiles. Una *entidad débil* es una entidad cuya existencia depende de la existencia de otra entidad. Una *entidad fuerte* es una entidad que no es débil.

Relación (interrelación)

Es una correspondencia o asociación entre dos o más entidades. Cada relación tiene un nombre que describe su función. Las entidades que están involucradas en una determinada relación se denominan *entidades participantes*. El número de participantes en una relación es lo que se denomina *grado* de la relación. Por lo tanto, una relación en la que participan dos entidades es una relación *binaria*; si son tres las entidades participantes, la relación es *ternaria*; etc. Una *relación recursiva* es una relación donde la misma entidad participa más de una vez en la relación con distintos papeles. La *cardinalidad* con la que una entidad participa en una relación especifica el número mínimo y el número máximo de correspondencias en las que puede tomar parte cada ocurrencia de dicha entidad. La participación de una entidad en una relación es *obligatoria (total)* si la existencia de cada una de sus ocurrencias requiere la existencia de, al menos, una ocurrencia de la otra entidad participante. Si no, la participación es *opcional (parcial)*. Las reglas que definen la cardinalidad de las relaciones son las *reglas de negocio*.

Atributo

Es una característica de interés o un hecho sobre una entidad o sobre una relación. Los atributos representan las propiedades básicas de las entidades y de las relaciones. Cada atributo tiene un conjunto de valores asociados denominado *dominio*. El dominio define todos los valores posibles que puede tomar un atributo. Puede haber varios atributos definidos sobre un mismo dominio. Los atributos pueden ser simples o compuestos. Un *atributo simple* es un atributo que tiene un solo componente, que no se puede dividir en partes más pequeñas que tengan un significado propio. Un

atributo compuesto es un atributo con varios componentes, cada uno con un significado por sí mismo. Un grupo de atributos se representa mediante un atributo compuesto cuando tienen afinidad en cuanto a su significado, o en cuanto a su uso. Los atributos también pueden clasificarse en monovalentes o polivalentes. Un *atributo monovalente* es aquel que tiene un solo valor para cada ocurrencia de la entidad o relación a la que pertenece. Un *atributo polivalente* es aquel que tiene varios valores para cada ocurrencia de la entidad o relación a la que pertenece. A estos atributos también se les denomina *multivaluados*, y pueden tener un número máximo y un número mínimo de valores. La *cardinalidad* de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. Los atributos pueden ser derivados. Un *atributo derivado* es aquel que representa un valor que se puede obtener a partir del valor de uno o varios atributos, que no necesariamente deben pertenecer a la misma entidad o relación.

Identificador

Un identificador de una entidad es un atributo o conjunto de atributos que determina de modo único cada ocurrencia de esa entidad. Un identificador de una entidad debe cumplir dos condiciones:

1. No pueden existir dos ocurrencias de la entidad con el mismo valor del identificador. Si se omite cualquier atributo del identificador, la condición anterior deja de cumplirse.
2. Toda entidad tiene al menos un identificador y puede tener varios identificadores alternativos. Las relaciones no tienen identificadores.

Jerarquía de generalización

Una entidad *E* es una generalización de un grupo de entidades E^1, E^2, ... E^n, si cada ocurrencia de cada una de esas entidades es también una ocurrencia de *E*. Todas las propiedades de la entidad genérica *E* son heredadas por las subentidades. Cada jerarquía es total o parcial, y exclusiva o superpuesta. Una jerarquía es *total* si cada ocurrencia de la entidad genérica corresponde al menos con una ocurrencia de alguna subentidad. Es *parcial* si existe alguna ocurrencia de la entidad genérica que no corresponde con ninguna ocurrencia de ninguna subentidad. Una jerarquía es *exclusiva* si cada ocurrencia de la entidad genérica corresponde, como mucho, con una ocurrencia de una sola de las subentidades. Es *superpuesta* si existe alguna ocurrencia de la entidad genérica que corresponde a ocurrencias de dos o más subentidades diferentes. Un *subconjunto* es un caso particular de generalización con una sola entidad como subentidad. Un subconjunto siempre es una jerarquía parcial y exclusiva.

1.2.3.2. Metodología de Diseño Conceptual

El primer paso en el diseño de una base de datos es la producción del esquema conceptual. Normalmente, se construyen varios esquemas conceptuales, cada uno para representar las distintas visiones que los usuarios tienen de la información. Cada una de estas visiones suelen corresponder a las diferentes áreas funcionales de la empresa como, por ejemplo, producción, ventas, recursos humanos, etc.

Estas visiones de la información, denominadas *vistas*, se pueden identificar de varias formas. Una opción consiste en examinar los diagramas de flujo de datos, que se pueden haber producido previamente, para identificar cada una de las áreas funcionales. La otra opción consiste en entrevistar a los usuarios, examinar los procedimientos, los informes y los formularios, y también observar el funcionamiento de la empresa.

A los esquemas conceptuales correspondientes a cada vista de usuario se les denomina *esquemas conceptuales locales*. Cada uno de estos esquemas se compone de entidades, relaciones,

atributos, dominios de atributos e identificadores. El esquema conceptual también tendrá una documentación, que se irá produciendo durante su desarrollo. Las tareas a realizar en el diseño conceptual son las siguientes [**Hal1994**]:

- Identificar las entidades.
- Identificar las relaciones.
- Identificar los atributos y asociarlos a entidades y relaciones.
- Determinar los dominios de los atributos.
- Determinar los identificadores.
- Determinar las jerarquías de generalización (si las hay).
- Dibujar el diagrama entidad-relación.
- Revisar el esquema conceptual local con el usuario.

1.2.4. Herramientas de Bases de Datos Web

Las bases de datos son hoy en día el corazón de una organización, una pieza clave de las aplicaciones de negocio. De la calidad de una base de datos depende la productividad y la ventaja competitiva de muchas compañías.

En entornos heterogéneos, o en aquellos en los que se dispone de una misma plataforma (Oracle, DB2, SQL Server, etc.), los desarrolladores y responsables de las bases de datos han de contar con herramientas de software y soluciones para poder garantizar la disponibilidad y el rendimiento de sus bases de datos.

Estas herramientas pueden ser desde un Sistema Gestor de Base de Datos (MySQL, PostgreSQL Oracle, Sybase, SQLServer, Access, etc.), hasta un software que permita diseñar base de datos (CAD de diseño de base de datos) como PowerDesigner de Powersoft (Sybase).

Existen diversos tipos de herramientas que brindan soporte para todos los aspectos del desarrollo de una base de datos, como la administración y manejo de bases de datos, así como componentes que permiten a las aplicaciones trabajar con diferentes formatos de bases de datos. A continuación mostraremos diversos tipos de herramientas con algunos ejemplos del software utilizado actualmente [**ScA1996**]:

Generadores de Código y Herramientas para mapeo objeto-relación
Estos productos eliminan la necesidad de escribir código de acceso a la base de datos repetitivamente, generando los objetos y el código necesario, o bien proporcionando una interface de objetos dinámicos. Ejemplos de estos son: MagicDraw UML, ADO Data Class Builder , eXpress Persistent Objects for .NET 2007.

Herramientas de Calidad de Información
Se usan para mantener la información en buenas condiciones, asegurando la consistencia y limpieza, eliminando duplicados, reformateando más. Un ejemplo de estas: AddressDoctor Enterprise.

Herramientas de Administración de Bases de Datos
Estas herramientas permiten al usuario realizar tareas de mantenimiento y administración en las bases de datos, incluyen a SQL Server, Oracle, MySQL y MS Access. Estas herramientas hacen posible poder administrar la base de datos desde cualquier localidad por medio de páginas Web.

APIs de Bases de Datos y Componentes de Conectividad
Estos componentes proporcionan al desarrollador un API por medio del cual pueden implementar operaciones para acceso a la información. Estos productos varían según el tipo de

base de datos (SQL Server, Oracle, MySQL, MS Access, BDE) , los tipos de acceso y los mecanismos (ODBC, JDBC, ADO, JET, etc.) Algunos ejemplos son: Database Workbench Pro, SQL Examiner Suite 2008.

Herramientas de Comparación de Bases de Datos

Estas herramientas le permiten al usuario final comparar la estructura y la información entre dos bases de datos, reportar diferencias y sincronizar ambas bases. Ejemplos de estas herramientas son: EasySQL Data Compare, SQL Examiner Suite 2008.

Herramientas para Desarrollo de Bases de Datos

Ambientes de desarrollo integrado para desarrolladores de proyectos con bases de datos. Ejemplos son: Advanced Data Generator, MySQL Edition.

Herramientas para Documentación de Bases de Datos

Generan un esquema de documentación para bases de datos, incluyendo diagramas de tablas y relaciones, así como referencias cruzadas. Ejemplos de estas herramientas: SQL Documentation Tool, SQL Help Builder.

Motores de Bases de Datos

Dan a las aplicaciones capacidades de persistencia sin sacrificar la portabilidad e independencia, proporcionan capacidades para almacenamiento de información que pueden ser integradas dentro de aplicaciones contruidas para todas las plataformas: Net, ActiveX, Java, Delphi. Ejemplos de estas son: TurboDB para Delphi & C++Builder.

Sistemas Administradores de Bases de Datos

Permiten organizar y almacenar información par alas aplicaciones. Algunos proporcionan ambientes de desarrollo en los cuales se puede diseñar, implementar y montar aplicaciones de bases de datos administradas completas. Access 2007, Microsoft SQL Server 2000 , Microsoft Visual FoxPro , MySQL database server son ejemplos importantes.

Componentes Constructores de Consultas para Bases de Datos

Permiten al desarrollador proporcionar al usuario final consultas de información de forma fácil y accesible. Ejemplos:Active Query Builder .NET , Automated-SQL-Builder.

Componentes de Procesamiento Analitico en Línea (OLAP)

Son herramientas que proporcionan el análisis de información almacenada en una base de datos. Ejemplos de estas son: Data Dynamics Analysis, RadarCube ASP.NET for MS Analysis.

Herramientas y Componentes para Escritura de Reportes

Herramientas que generan reportes de la información contenida en una base de datos. Por ejemplo: ActiveReports, Visual Reports , Crystal Reports.

1.3. Modelo Cliente-Servidor

Esta arquitectura consiste básicamente en que un programa -el cliente- que realiza peticiones a otro programa -el servidor- que le da respuesta (ver **Figura 1.9**). Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras [**MeW2009**].

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes **computadoras** aumentando así el grado de distribución del sistema.

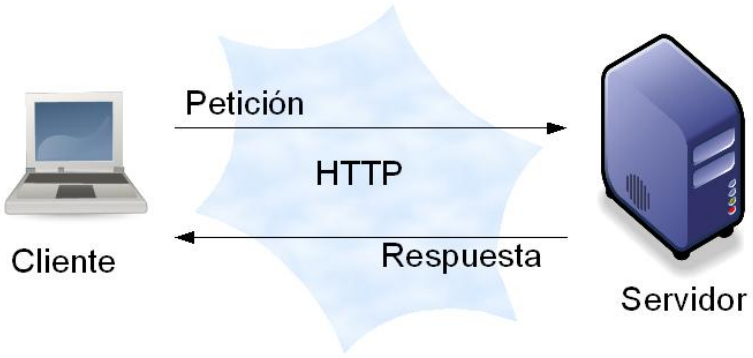


Figura 1.9. Modelo Cliente-Servidor

La **arquitectura cliente-servidor (C/S)** sustituye a la arquitectura **monolítica** en la que no hay distribución, tanto a nivel físico como a nivel lógico.

Características de un cliente

En la arquitectura C/S el **remitente de una solicitud** es conocido como cliente. Sus características son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación (dispositivo **maestro** o **amo**).
- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.

Características de un servidor

En los sistemas C/S el **receptor de la solicitud** enviada por cliente se conoce como servidor. Sus características son:

- Al iniciarse esperan a que lleguen las solicitudes de los clientes, desempeñan entonces un papel pasivo en la comunicación (dispositivo **esclavo**).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

Arquitectura multi-capas

La arquitectura cliente/servidor genérica tiene dos tipos de nodos en la red: clientes y servidores. Consecuentemente, estas arquitecturas genéricas se refieren a veces como arquitecturas de dos niveles o **dos capas**.

1.4. World Wide Web

World Wide Web (o la "Web") o **Red Global Mundial** es un sistema de documentos de hipertexto[3] y/o hipermedios enlazados y accesibles a través de Internet[4]. Con un navegador Web, un usuario visualiza páginas Web que pueden contener texto, imágenes, vídeos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces[**MeW2009**].

La visualización de una página web de la World Wide Web normalmente comienza tecleando la URL[5] de la página en el navegador web, o siguiendo un enlace de hipertexto a esa página o recurso. En ese momento el navegador comienza una serie de comunicaciones, transparentes para el usuario, para obtener los datos de la página y visualizarla.

El primer paso consiste en traducir la parte del nombre del servidor de la URL en una dirección IP usando la base de datos distribuida de Internet conocida como DNS. Esta dirección IP es necesaria para contactar con el servidor web y poder enviarle paquetes de datos.

El siguiente paso es enviar una petición HTTP al servidor Web solicitando el recurso. En el caso de una página web típica, primero se solicita el texto HTML y luego es inmediatamente analizado por el navegador, el cual, después, hace peticiones adicionales para los gráficos y otros ficheros que formen parte de la página.

Así que se han recibido los ficheros solicitados desde el servidor web, el navegador renderiza la página tal y como se describe en el código HTML, el CSS y otros lenguajes web. Al final se incorporan las imágenes y otros recursos para producir la página que ve el usuario en su pantalla.

Esta colección de recursos útiles y relacionados, interconectados a través de enlaces de hipertexto, es lo que ha sido denominado como 'red' (web, en inglés) de información. Al trasladar esta idea a Internet, se creó lo que Tim Berners-Lee llamó **WorldWideWeb** (un término escrito en CamelCase, posteriormente desechado) en 1990.

1.5. MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo[6] y multiusuario con más de seis millones de instalaciones.[1] MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems— desarrolla MySQL como software libre en un esquema de licenciamiento dual [**MeW2009**].

Su popularidad como aplicación web está muy ligada a PHP (ver apartado 1.6), que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la

[3] **Hipertexto**, en informática, es el nombre que recibe el texto que en la pantalla de una computadora conduce a su usuario a otro texto relacionado.
[4] **Internet** es un conjunto descentralizado de redes de comunicación interconectadas, que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.
[5] **URL** significa ***U**niform **R**esource **L**ocator*, es decir, localizador uniforme de recurso. Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, como documentos e imágenes en Internet, por su localización.
[6] Un **hilo de ejecución**, en sistemas operativos, es una característica que permite a una aplicación realizar varias tareas concurrentemente.

modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

MySQL es un sistema de administración relacional de bases de datos. Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.

MySQL es software de fuente abierta. Fuente abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar.

Seguridad para MySQL

Con el objetivo de mantener segura la información almacenada en la base de datos, evitando ataques e intrusiones, MySQL proporciona un sistema de privilegios, cuya función primaria es autenticar un usuario conectándose desde un equipo dado, y asociar dicho usuario con privilegios en una base de datos tales como SELECT, INSERT, UPDATE, y DELETE. Además incluye la habilidad de tener usuarios anónimos y de dar privilegios para funciones específicas de MySQL tales como LOAD DATA INFILE y operaciones administrativas.

El sistema de privilegios de MySQL asegura que todos los usuarios pueden ejecutar sólo la operación permitida a los mismos. Como usuario, cuando conecta a un servidor MySQL, su identidad se determina mediante *el equipo desde el que se conecta* y *el nombre de usuario que especifique*. Cuando efectúe peticiones tras conectar, el sistema le otorga privilegios acorde a su identidad y *lo que quiera hacer*. En la **tabla 1.1** se muestra el listado de privilegios MySQL y su contexto.

El control de acceso de MySQL implica dos etapas [**MSQ2009**]:

- Etapa 1: El servidor comprueba si debe permitirle conectarse.
- Etapa 2: Asumiendo que se conecta, el servidor comprueba cada comando que ejecuta para ver si tiene suficientes permisos para hacerlo. Por ejemplo, si intenta seleccionar registros de una tabla en una base de datos o eliminar una tabla de la base de datos, el servidor verifica que tenga el permiso SELECT para la tabla o el permiso DROP para la base de datos.

Privilegio	**Contexto**
CREATE	bases de datos, tablas, o índices
DROP	bases de datos o tablas
GRANT OPTION	bases de datos, tablas, o procedimientos almacenados
REFERENCES	bases de datos o tablas
ALTER	tablas
DELETE	tablas
INDEX	tablas
INSERT	tablas
SELECT	tablas
UPDATE	tablas
CREATE VIEW	vistas
SHOW VIEW	vistas
ALTER ROUTINE	procedimientos almacenados
CREATE ROUTINE	procedimientos almacenados
EXECUTE	procedimientos almacenados
FILE	acceso a archivos en la máquina del servidor

CREATE TEMPORARY TABLES	administración del servidor
LOCK TABLES	administración del servidor
CREATE USER	administración del servidor
PROCESS	administración del servidor
RELOAD	administración del servidor
REPLICATION CLIENT	administración del servidor
REPLICATION SLAVE	administración del servidor
SHOW DATABASES	administración del servidor
SHUTDOWN	administración del servidor
SUPER	administración del servidor

Tabla 1.1. Privilegios de los que provee MySQL [MSQ2009]

Los privilegios CREATE y DROP permiten crear nuevas bases de datos y tablas, o eliminar las existentes.

Los privilegios SELECT, INSERT, UPDATE, and DELETE permiten realizar operaciones en registros de tablas existentes en una base de datos.

El privilegio INDEX permite crear o eliminar índices. INDEX es aplicable a tablas existentes. Si tiene el privilegio CREATE para una tabla, entonces puede incluir definiciones de índices en la sentencia CREATE TABLE.

El privilegio ALTER permite utilizar ALTER TABLE para cambiar la estructura de o renombrar tablas.

El privilegio CREATE ROUTINE es necesario para crear procedimientos almacenados (funciones y procedimientos). El privilegio ALTER ROUTINE se necesita para modificar o eliminar procedimientos almacenados, y EXECUTE es necesario para ejecutarlos.

El privilegio GRANT permite dar a otros usuarios los privilegios que uno mismo posee. Puede ser utilizado para bases de datos, tablas, y procedimientos almacenados.

El privilegio FILE otorga permiso para leer y escribir archivos en la máquina del servidor utilizando las sentencias LOAD DATA INFILE y SELECT ... INTO OUTFILE. Un usuario que tiene el privilegio FILE puede leer cualquier archivo de la máquina del servidor que sea legible por cualquiera o por el usuario que ejecuta el servidor MySQL. (Esto implica que el usuario puede leer cualquier archivo en el directorio de datos, porque el servidor puede acceder a cualquiera de estos archivos.) El privilegio FILE también permite al usuario crear archivos nuevos en cualquier directorio en que el servidor MySQL tiene acceso de escritura. Los archivos existentes no pueden ser sobreescritos.

El comando RELOAD comunica al servidor que debe releer las tablas GRANT a memoria. FLUSH-PRIVILEGES es un sinónimo de RELOAD. El comando RELOAD cierra y vuelve a abrir los archivos de registro y vuelca todas las tablas. Los otros comandos FLUSH-*XXX* realizan funciones similares a REFRESH, pero son más específicas y pueden ser preferibles en algunos casos. Por ejemplo, si quiere tan solo volcar los archivos de registro, FLUSH-LOGS es mejor opción que REFRESH.

El comando SHUTDOWN apaga el servidor.

El comando PROCESSLIST muestra información sobre los subprocesos que se están ejecutando en el servidor (es decir, sobre las sentencias que se están ejecutando por parte de clientes asociados con otras cuentas).

El comando KILL mata los subprocesos del servidor. Siempre puede mostrar información sobre sus propios subprocesos, o matarlos, pero necesita el privilegio PROCESS para ver subprocesos iniciados por otros usuarios, y el privilegio SUPER para matarlos.

El privilegio CREATE TEMPORARY TABLES permite la utilización de la palabra clave TEMPORARY en sentencias CREATE TABLE.

El privilegio LOCK TABLES permite la utilización de sentencias LOCK TABLES explícitas para bloquear tablas para las que tiene el privilegio SELECT. Esto incluye el uso de bloqueos de escritura, que evita que cualquier otra persona lea la tabla bloqueada.

El privilegio REPLICATION CLIENT permite la utilización de las sentencias SHOW MASTER STATUS y SHOW SLAVE STATUS.

El privilegio REPLICATION SLAVE debería otorgarse a cuentas que son utilizadas por servidores esclavos para conectarse al servidor actual como su maestro. Sin este privilegio, la cuenta esclava no puede pedir actualizaciones que se hayan hecho a las bases de datos del servidor maestro.

El privilegio SHOW DATABASES permite a la cuenta ver los nombres de las bases de datos mediante la ejecución de la sentencia SHOW DATABASE. Cuentas que no tengan este privilegio solo pueden ver las bases de datos para las que tienen privilegios, y no pueden utilizar la sentencia de ninguna manera si el servidor ha sido iniciado con la opción --skip-show-database.

En general, es una buena idea garantizar a una cuenta solo aquellos privilegios que necesita. Se debe tener especial cuidado en seguir esta regla con los privilegios administrativos y FILE [**MSQ2009**]:

- El privilegio FILE puede utilizarse inadecuadamente para introducir en una tabla de la base de datos cualquier archivo que el servidor MySQL sea capaz de leer en la máquina del servidor. Esto incluye todos los archivos que sean legibles para cualquiera, además de los archivos almacenados en el directorio de datos del servidor. Esta tabla puede entonces ser accedida utilizando una sentencia SELECT para transferir sus contenidos a la máquina cliente.
- El privilegio GRANT permite a los usuarios otorgar sus mismos privilegios a otros usuarios. Dos usuarios con diferentes privilegios y con el privilegio GRANT pueden combinar sus privilegios.
- El privilegio ALTER puede ser utilizado inadecuadamente para sabotear el sistema de privilegios mediante el renombrado de tablas.
- El privilegio SHUTDOWN puede utilizarse inadecuadamente para denegar el servicio a otros usuarios de manera total, cerrando el servidor.
- El privilegio PROCESS puede utilizarse para ver el texto de las consultas que se estén ejecutando actualmente, incluyendo consultas que establecen o modifican passwords.
- El privilegio SUPER puede utilizarse para cerrar la conexión a otros clientes o cambiar como el servidor funciona.
- Los privilegios otorgados para la propia base de datos mysql pueden utilizarse para cambiar passwords y otra información de privilegios de acceso. Las passwords se almacenan cifradas, así que un usuario malicioso no puede simplemente leerlas para conocer la password. Aún así, el usuario con privlegio de escritura a la columna Password de la tabla user puede cambiar la password de una cuenta, y seguidamente conectarse al servidor MySQL utilizando esa cuenta.

Hay algunas cosas que no se pueden hacer con el sistema de privilegios de MySQL [**MSQ2009**]:

- No se puede especificar explícitamente que a un usuario se le deba denegar el acceso.
- No se puede especificar que un usuario tenga privilegios para crear o eliminar tablas en una base de datos, pero que no pueda crear o eliminar la propia base de datos.

1.6. PHP

PHP es un lenguaje de programación interpretado[7], diseñado originalmente para la creación de páginas web dinámicas. PHP significa ***P**HP **H**ypertext **P**re-processor* (inicialmente PHP Tools, o, *P*ersonal *H*ome *P*age Tools) [**MeW2009**].

PHP es un lenguaje interpretado de propósito general ampliamente usado y que está diseñado especialmente para desarrollo web y puede ser embebido dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. La más reciente versión principal del PHP fue la versión 5.3.0 (30 de junio de 2009).

Características de PHP

- Es un lenguaje multiplataforma.
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- No requiere definición de tipos de variables.

Seguridad para PHP

Una de las ventajas más grandes de PHP es la facilidad con la cual puede comunicarse con las bases de datos, lo más normal con MySQL. Sin embargo, con esa ventaja hay problemas suficientemente grandes en la seguridad a los que se tienen que hacer frente. A continuación se muestran algunas recomendaciones de seguridad para mantener al sistema y la información libre de ataques e intrusiones [**PHP2009**]:

- Disponer siempre de versiones actualizadas de Apache y PHP
- Cuando la directiva *register_globals* está activada en el fichero **php.ini**, PHP crea automáticamente variables globales a partir de los datos de los formularios y de las *cookies*. *Es recomendable d*eshabilitar *register_globals* en el fichero php.ini, a partir de PHP 4.2.0 el valor por defecto de esta directiva es off.
- Es recomendable inicializar todas las variables antes de usarlas.

[7] Se conoce como **lenguaje interpretado** a un lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados. También se les conoce como lenguajes de ***script***.

- En un entorno de producción debe evitarse la aparición de mensajes de aviso o error. Para ello se utilizan las siguientes directivas en **php.ini**: *display_errors=off, log_errors=on, error_log= /var/log/php_errors.log*, con esto los errores irán al fichero especificado en lugar de mostrarse en la pantalla.
- Además hay que tener en cuenta que las funciones de manejo de ficheros como *include()* o *require()* admiten nombres de ficheros remotos, lo que podría provocar la ejecución de código maligno cargado de otro servidor. Sea, por ejemplo, el código *include($libdir. "/conecta.php");* si un atacante modifica el valor de la variable *$libdir* a, pongamos por caso, "*http://atacante/*", y coloca en la raíz del mismo un fichero de nombre *conecta.php*, su código sería ejecutado. Se puede desactivar la funcionalidad de acceso a ficheros remotos con la siguiente directiva en **php.ini**: *allow_url_fopen= off*.
- Otra defensa contra los nombres de ficheros incorrectos es la directiva de **php.ini** open_basedir: *open_basedir= /alguna/ruta*, PHP limitará las operaciones sobre ficheros al directorio especificado y sus subdirectorios:
include("/alguna/ruta/lib.inc");// permitido
include("/otra/ruta/lib.inc");// da error
- Es recomendable validar todos los datos provenientes de formularios para asegurarse de que los valores recibidos son los esperados. Si un usuario introduce algún código HTML en el texto del formulario, el código será interpretado y sus efectos podrían ser graves, para evitar esto se puede utilizar la función *htmlspecialchars(),* que impide que se interpreten los caracteres especiales de HTML (<, >, &). El código quedaría de la siguiente manera: *print("Nombre: " . $nombre); print("Comentario: " . htmlspecialchars($comentario));*
- La Inyección SQL consiste en inyectar un mandato dentro de una consulta SQL. Por ejemplo, sea la consulta:
$consulta= "SELECT titulo FROM libros WHERE codigo = $codigo";* (siendo *$codigo un valor introducido desde un formulario).
Si el valor es '*23*' la consulta será:
SELECT titulo FROM libros WHERE codigo= 23
Si el valor es '*23; DROPTABLE users*' la consulta es:
SELECT titulo FROM libros WHERE codigo= 23; DROP TABLE users (que destruiría la tabla de usuarios de MySQL)
Otro ejemplo es: Se puede saltar la comprobación del password introduciendo el valor 'juan'--' como username o el valor ' *OR* ''='como password. Las consultas que quedarían en ambos casos son, respectivamente:
SELECT id FROM usuarios WHERE username= 'juan'--' AND password = ''SELECT id FROM usuarios WHERE username= 'juan' AND password = '' OR ''=''
Nótese que -- es un comentario de línea en MySQL y provoca que se ignore todo lo que viene tras él en la línea.
Por esto, para evitar la inyección SQL es muy importante validar los valores que se han de integrar en la consulta SQL, y utilizar *htmlentities* y *html_entity_decode*.
- Es recomendable hacer uso de la encriptación de las contraseñas mediante MD5, que se utiliza para comprobar la integridad de un archivo descargado de Internet y prevenir posibles vulnerabilidades en nuestro servidor. Esto con el objeto de que si alguien puede acceder a las contraseñas, no pueda ver la contraseña si no su encriptación. Esto se logra haciendo uso de la función *md5()* de PHP [**MD52009**].

CAPITULO 2. ADMINISTRACION ESCOLAR

2.1. Antecedentes UIA (Universidad Iberoamericana)

Misión, principios y filosofía educativa

La Universidad Iberoamericana asume el proyecto educativo de la Compañía de Jesús, y tiene como misión contribuir en un ambiente de participación responsable, apertura, libertad, respeto y crítica propositiva al desarrollo y la difusión del conocimiento, y a la formación de profesionales e investigadores con calidad humana y académica, que se comprometan en el servicio a los demás para el logro de una sociedad más libre, productiva, justa y solidaria.

La Compañía de Jesús, cuyos miembros son conocidos como los jesuitas, es una orden religiosa fundada por San Ignacio de Loyola en 1540 junto con nueve compañeros. Este grupo quiso ponerse especialmente bajo la autoridad del Sumo Pontífice para ser enviados a aquellos lugares donde él viera que podían dar un mayor servicio a los demás. Pronto el papa Paulo III los distribuyó por diferentes países para hacer tareas muy variadas.

Desde entonces la Compañía de Jesús se ha extendido por todos los continentes y continúa teniendo como un rasgo fundamental el servir a la Iglesia a través de la obediencia especial al Vicario de Cristo. Actualmente son casi 25,000 jesuitas en el mundo, de los cuales casi 500 pertenecen a la Provincia Mexicana [**UIA2009**].

Asociación de Universidades Confiadas a la Compañía de Jesús

Es una Asociación de 29 universidades confiadas a la Compañía de Jesús en 14 países de América Latina, a la luz de la realidad de nuestros pueblos y en fidelidad a la específica inspiración cristiana y a la identidad y tradición espiritual y pedagógica de la Compañía de Jesús. La Universidad Iberoamericana Puebla, junto con los planteles de la Ciudad de México, León, Tijuana y Torreón, así como el ITESO de Guadalajara y la Universidad Loyola del Pacífico, conforman el Sistema Universitario Jesuita.

Plan Estratégico 2007-2011

El Plan Estratégico 2007-2011 pone al día la misión y visión de la Universidad Iberoamericana, describe las áreas prioritarias en donde se concentrarán los esfuerzos institucionales, y detalla las metas que se ha propuesto alcanzar en los próximos cinco años. Este plazo se escogió por ser suficiente para superar una mentalidad de corto plazo conservando una visión realista, que permita guiar la acción.

El proceso en el que se apoya la definición de las áreas y metas estratégicas aquí presentadas, dio inicio con un amplio diagnóstico que tomó en cuenta tanto elementos externos como internos. Entre los primeros, fueron considerados la demanda de educación superior, el análisis de la competencia y las tendencias del sector, que permitieron definir un escenario educativo plausible en el mediano plazo. Del contexto interno, se analizaron las características de nuestro alumnado, de la planta docente, de los programas académicos, y de los procesos sustantivos, así como el de enseñanza-aprendizaje, entre otros.

De esta forma, se establecieron los lineamientos que sirvieron de base para elaborar los planes estratégicos de cada una de las dependencias universitarias. Estos planes sirvieron, a su vez, para afinar y completar los planteamientos iniciales que derivaron en el plan institucional. Por consiguiente, además de este, existen planes estratégicos de los departamentos y de las unidades administrativas, que, en algunos casos, complementan y detallan las líneas institucionales.

2.2. Modelo UIA

La Universidad Iberoamericana, institución confiada a la Compañía de Jesús, tiene como ***misión***:

Formar profesionales e investigadores de proyección internacional, con calidad humana y académica, que se comprometan en el servicio a los demás; y desarrollar y difundir el conocimiento para el logro de una sociedad libre, justa, solidaria y productiva.

La ***visión*** de la Universidad Iberoamericana, reconocida por su excelencia académica, pretende ofrecer a sus alumnos una formación integral que vaya más allá de la mera adquisición de información o conocimientos, al favorecer su crecimiento como personas, mediante el desarrollo de sus capacidades, su sentido de ser con y para los demás, y sus actitudes de compromiso, de manera especial, con los más necesitados de la sociedad [**UIA2009**].

Forma profesionales e investigadores:

a) De proyección nacional e internacional
b) Con calidad humana y académica
c) Que se comprometan en el servicio a los demás.

Con programas innovadores de calidad internacional:

- Desarrolla y difunde el conocimiento para el logro de una sociedad libre, justa, solidaria y productiva
- Promueve la transformación social

Con una administración servicial, eficaz y eficiente:

- En una comunidad que actúa conforme a lo que pregona.
- Áreas prioritarias de acción

La planeación estratégica establece seis áreas prioritarias de acción en las que la UIA deberá concentrar sus esfuerzos durante los próximos cinco años, con la finalidad de fortalecerse frente a los mayores desafíos que apunta nuestro contexto actual, y cumplir eficazmente con su Misión y Visión. Dichas áreas son [**UIA2009**]:

1. Calidad Académica
2. Formación Integral
3. Investigación y Transformación Social
4. Selectividad de Alumnos y Matrícula Adecuada
5. Uso óptimo de Recursos Materiales y Humanos
6. Procesos de Administración, Gestión y Comunicación Interna.

2.3. Sistemas de Administración Escolar

Los Sistemas de Administración Escolar, permiten a las instituciones educativas llevar un control administrativo, académico, y de procesos dentro de la institución, en sitio o a través de Internet por lo que es un software altamente adaptable y flexible.

La finalidad de estos sistemas es proveer una herramienta para la administración de académica, y con ello poder brindar a los usuarios (Maestros, Alumnos y Personal Administrativo) el poder consultar información sobre su desempeño escolar. Además pueden proporcionar funciones como: consultar saldos o pagar las colegiaturas en línea, hasta ver las calificaciones y ausencias de los alumnos.

La Universidad Iberoamericana en su plantel Puebla, tiene más de 25 años, y maneja un sistema de administración escolar que cumple con los estándares y necesidades propios de la educación superior, pero que no es adaptable para su aplicación en las preparatorias de reciente creación. Por tanto, surge la necesidad de crear un sistema especializado para las Preparatorias UIA en sus tres planteles, ya que actualmente se realizan tareas de organización y archivo manualmente, y se manejan de archivos elaborados en Excel que son difíciles de controlar al momento de tratar de controlar una matrícula de alumnos cada vez más amplia [**IbP2009**].

En la actualidad existen diversos sistemas de administración escolar implementados en los planteles educativos en todos sus niveles, ya sean aplicaciones intranet, o bien aplicaciones Web.

La mayoría de los sistemas son desarrollados específicamente para cada institución, pero también existen en el mercado sistemas adaptables que ofrecen diversas soluciones. Estos sistemas son una solución perfecta para cualquier institución educativa que quiera tomar ventaja de la tecnología, y repercutirla a sus alumnos, sus maestros y sus procesos administrativos. A los alumnos les ayuda, ya que pueden consultar sus tareas en línea, y en muchos casos se pueden encontrar guías de estudio en el mismo sitio. También permiten a las instituciones educativas llevar un control administrativo, académico, y de procesos dentro de la institución, en sitio o a través de Internet. Incluso permite a los padres de familia estar en contacto constante con la institución por medio de Internet, por lo que los padres de familia pueden desde consultar saldos o pagar las colegiaturas en línea, hasta ver las calificaciones y ausencias de sus hijos. Ejemplos de estos sistemas de administración escolar actualmente a la venta son el Sistema de Administración Integral Escolar (SAIE XP) desarrollado por Soluciones Integrales en Software y Consultoría en México D.F., el Sistema de Administración Escolar Syscol 2008 para colegios

Instituciones de educación superior, universidades reconocidas a nivel nacional e internacional, utilizan sistemas de administración escolar desarrollados según las necesidades específicas de cada institución, a continuación se mencionan algunos de los más importantes:

- La Dirección General de Administración Escolar de la **UNAM**, a través de la Subdirección de Sistemas de Registro Escolar (S.S.R.E.), desarrolló el **Sistema Integral de Administración Escolar (S.I.A.E.)**, para otorgar un servicio de calidad y eficiencia a cada miembro que integra la máxima casa de estudios, en todo lo referente al registro y seguimiento académico de la trayectoria escolar de los alumnos de la UNAM. El objetivo de este sistema es lograr dentro de la Administración Escolar de la UNAM, la simplificación, agilización y descentralización de los trámites académico-administrativos de los alumnos en todas las escuelas y facultades, integrando información confiable y consistente como el producto de la coordinación de todos los elementos de la Institución. Los servicios que presta son: la actualización de carreras, planes de estudio, asignaturas y planteles que se ofrecen en la UNAM; proporciona acceso a escuelas, facultades, alumnos y público en general para consulta sobre asignaturas, créditos, seriación, equivalencia de asignaturas entre diferentes planes de estudio de una misma carrera, así como consulta de los requisitos de ingreso y titulación a la carrera; registro de alumnos de primer ingreso al ciclo (secundaria, bachillerato, técnico y licenciatura); solicitud y autorización de trámites; calificación de actas

en línea por parte del personal docente; impresión de documentación oficial en el plantel; consulta a la historia y trayectoria académica del alumno; emisión e impresión del directorio de alumnos; impresión de estadísticas de procesos como reinscripción, inscripción a extraordinarios, control de cupos y grupos, estadísticas de aprobación y reprobación, por asignatura, etc. [**SIAE2009**]

- La **Benemérita Universidad Autónoma de Puebla** es la primera institución de educación superior en nuestro país que certifica todos sus procesos escolares bajo la norma ISO 9001-2000, actualmente cuenta con 6 dependencias certificadas bajo la misma: Tesorería, Contabilidad, Dirección de Patrimonio, Dirección de Adquisiciones, Proveeduría e Inventarios y recientemente la **Dirección de Administración Escolar (DAE)**. Hablar del proceso de certificación en la DAE nos obliga a remontarnos a 1998, año en que se sentaron las bases de una nueva forma de trabajo, a través de lo que se llamó en su momento MODELO INTEGRAL DE ADMINISTRACIÓN ESCOLAR, que fue el resultado de un diagnóstico de las condiciones en que la Dirección venía trabajando en esos años. Al paso de los años este Modelo de trabajo se transformó en lo hoy conocemos como PROCESO INTEGRAL DE ADMINISTRACIÓN ESCOLAR, que permitió entender que los diferentes departamentos funcionaban de acuerdo a procesos y no como departamentos aislados. Para entender la complejidad de la DAE es importante mencionar, que la BUAP tiene en la actualidad una población estudiantil de cerca de 50 mil alumnos, distribuidos en los diferentes niveles, de tal forma que de preparatoria atendemos una población de 9508 alumnos, de nivel subprofesional 1854, de licenciatura 33860 y de posgrado 3337 alumnos, datos correspondientes a Primavera 2004, lo que representa un total de 45222 alumnos. La Dirección de Administración Escolar tiene como objetivo atender a toda esta población, desde su ingreso hasta su egreso, otorgando diferentes servicios escolares que van desde la selección de aspirantes a ingresar hasta la tramitación de su título profesional. Esto por medio de un manual de calidad, procedimientos documentados y un sistema de cómputo muy completo que apoya y agiliza estos procedimientos [**DAE2009**].

- La Dirección de Operación Escolar de Servicios Escolares y Efectividad Institucional del Sistema del Instituto **Tecnológico de Estudios Superiores de Monterrey**, ofrece a la comunidad Tec (con el apoyo de la Vicerrectoría de Recursos Humanos y Tecnologías de Información) algunos servicios e información para apoyar actividades académicas y escolares del Instituto mediante su **Sistema de Administración Académica (SADA), en el cual se encuentra** el punto de entrada a los siguientes servicios y al Manual de Banner Escolar:
    - Alumnos:
    - Evaluanet
    - Permisos Intercampus
    - Empleados:
    - Administración y Programación de grupos
    - Banner Escolar (entrada al manual)
    - Data Warehouse de Escolar
    - Encuestas de profesores a directivos / Resultados de ECOAS y ECOPS
    - Permisos Intercampus

  Sólo pueden acceder a cada servicio aquellas personas (empleados o alumnos) que tengan los privilegios requeridos. Cada servicio tiene su propia administración de cuentas de ingreso ya conocida por el usuario de cada uno [**SADA2009**].

- La **Universidad Iberoamericana (UIA)** cuenta con un sistema intranet que ofrece servicios como: recursos humanos online; asignatura; formatos DICA para solicitudes de servicios como bajas o cambios activo fijo, informática, mantenimiento, reserva y préstamo de espacios institucionales; sistema de guías de estudio modelo; planes de estudio; sistema de guías de estudios del profesor; asignación de funciones, sistema de gestión académica; indicadores de gestión y portal de finanzas. Todo esto mediante un sistema de control de acceso para usuarios registrados únicamente [**UIA2009**].

2.4. Políticas de la SEP

La **Secretaría de Educación Pública (SEP)** tiene como propósito esencial crear condiciones que permitan asegurar el acceso de todas las mexicanas y mexicanos a una educación de calidad, en el nivel y modalidad que la requieran y en el lugar donde la demanden.

Su visión es que para el año 2025, México cuente con un sistema educativo amplio, articulado y diversificado, que ofrezca educación para el desarrollo humano integral de su población. El sistema será reconocido nacional e internacionalmente por su calidad y constituirá el eje fundamental del desarrollo cultural, científico, tecnológico, económico y social de la Nación.

La Secretaría de Educación Pública en el desarrollo de sus funciones para proporcionar los servicios educativos que ofrece, requiere del soporte de una infraestructura tecnológica en materia de Información y comunicaciones **(TIC)**, y de los servicios que con ella se proporcionan a sus usuarios tanto internos como externos.

En el marco de la evolución tecnológica exponencial que nos ofrece el mercado mundial y dentro de esta modernidad tecnológica, que busca simplificar y optimizar el procesamiento de grandes volúmenes de datos y agilizar las formas y medios de comunicación entre las personas, la **Dirección General de Tecnología de la Información** en la Secretaría de Educación Pública, tiene entre otros objetivos, atender los requerimientos de los usuarios, cuidando que las erogaciones en tecnología de la información cumplan con los requisitos técnicos que favorezcan la continuidad de las operaciones y la estandarización de la infraestructura tecnológica, conforme con la disponibilidad de los recursos económicos asignados para este fin, en el marco normativo vigente.

Debido a que en la Secretaría de Educación Pública, la Oficialía Mayor a través de su **Dirección General de Tecnología de la Información (DGTEC)** es rectora, en esta materia, la infraestructura, soluciones y servicios que se requieran en la Secretaría en el Sector Central deberán regirse por lineamientos, acuerdos y normatividades, como instrumento s para dar atención a los requerimientos de TIC, de las diversas Unidades Administrativas que la integran, mediante el desarrollo de programas y proyectos.

Considerando que para resolver una necesidad de equipamiento de cómputo, de equipos y medios de comunicación, de sistemas o paquetes de cómputo y en general de cualquier herramienta de tecnología de la información, intervienen diversas instancias para su estudio, posibilidad de adoptarla y gestión de adquisición, a través de la **Dirección de Servicios y Gestión de Bienes Informáticos**, se emiten lineamientos con el propósito de apoyar la decisión de los Comités correspondientes, en la aprobación de proyectos de tecnología de la información.

El objetivo de estos lineamientos es establecer las bases que se deberán considerar para elaborar los estudios de viabilidad técnica de los proyectos y programas que en materia de tecnología de la información y comunicaciones y que se requiere llevar a cabo en la Secretaría de

Educación Pública a través de la **Dirección General de Tecnología de la Información**, adscrita a la Oficialía Mayor; para emitir un dictamen sobre la factibilidad técnica existente para satisfacer los requerimientos de las Unidades Administrativas de la Secretaría.

Dentro de las normatividades y lineamientos establecidos por los órganos de la SEP, se encuentra una serie de documentos que sirven de base y guía para el desarrollo de sistemas informáticos para instituciones educativas. A continuación se listan algunos de ellos:

- Lineamientos que regulan el uso y control de la infraestructura tecnológica de la SEP. (Dirección General de Tecnología de la Información, Normateca Interna SEP [**CII1**]).
- Lineamientos que regulan la elaboración de los estudios de viabilidad técnica para proyectos de tecnología de información. (Dirección General de Tecnología de la Información, Normateca Interna SEP [**CII1**]).
- Lineamientos que deberán observar las dependencias y entidades de la administración pública federal para notificar al instituto del listado de sistemas de datos personales. (Transparencia y Acceso a la Información, 20 de agosto 2003 [**CII2**]).
- Lineamientos de protección de datos personales. (Transparencia y Acceso a la Información, 30 de septiembre 2005 [**CII2**]).

2.5. Objetivo de la Administración Escolar

La Administración Escolar, es una disciplina de programación, organización, integración y control de los recursos materiales, humanos y financieros necesarios para la prestación del servicio educativo.

Los procesos que acompañan a la administración son [**OAE2009**]: *la planeación, la organización y la gestión* como factores determinantes. El papel de la organización escolar comprende desde la participación, colaboración, y organización del profesorado, la organización académica del alumnado, la organización del tiempo, las instituciones educativas, el sistema escolar y el sistema educativo, así como el currículum, autoridades, calidad, y competitividad del proceso. Los procesos de comunicación, organización y planeación son elementos centrales que requieren principal atención. Esto debido a que la planeación no es una actividad exclusiva de los sujetos educacionales, sino que conlleva políticas económicas, sociales, internacionales y educativas determinadas para el sistema educativo.

2.6. Objetivos Específicos de la Administración Escolar

Establecer y difundir las normas que deberán aplicarse en la administración escolar durante las etapas de inscripción, reinscripción, acreditación, regularización y certificación de alumnos, así como de la administración financiera de la institución [**OAE2009**].

INSCRIPCION.

El Área de Administración Escolar en la Entidad recibirá conforme al calendario que elabora para tal efecto la documentación para la autorización de la inscripción de los alumnos tanto de escuelas oficiales como de particulares con Reconocimiento de Validez Oficial de Estudios.

REINSCRIPCIÓN

Regular y controlar el registro del reingreso de los alumnos que son promovidos de semestre o que regresan de una baja temporal, con el propósito de continuar su historial académico.

ACREDITACIÓN

Establecer los requisitos para el reconocimiento oficial de la aprobación de una materia, semestre, grado o nivel escolar; regular la periodicidad y determinar los criterios con los que se realizarán las evaluaciones del aprendizaje, normar el registro de los resultados de los alumnos y actualizar su historial académico.

REGULARIZACIÓN

Establecer el proceso de acreditación de conocimientos de los alumnos irregulares y actualizar su historial académico.

CERTIFICACIÓN

Otorgar el reconocimiento oficial a los estudios realizados por los alumnos, conforme a los planes y programas de estudio correspondientes.

ADMINISTRACION

Llevar el control administrativo y financiero (en caso de ser una institución educativa privada) de cada alumno. Registrando pagos de inscripciones, reinscripciones, becas y colegiaturas.

2.7. Descripción de la Preparatoria UIA

La Compañía de Jesús se distingue por acompañar el crecimiento personal de cada estudiante. Más allá de la excelencia académica, favorece el desarrollo integral humano que capacita para enfrentar a tiempo los complejos retos de la realidad. La comprensión de Dios, del ser humano y de la realidad que caracteriza a San Ignacio de Loyola, es su apoyo e inspiración para este propósito. Por consiguiente, las Preparatorias Ibero ofrecen educación integral a nivel de preparatoria en las ciudades de Puebla y Tlaxcala a partir de agosto de 2007 [**IbP2009**].

FORMACIÓN INTEGRAL

La Preparatoria Ibero es una institución de educación media superior cuyo centro es el ser humano no sólo como estudiante, sino también como persona y como ser social, político y religioso.

El objetivo de la preparatoria es la formación de hombres y mujeres integrales, lo que implica motivarles a buscar las habilidades necesarias para poder desenvolverse en la enseñanza superior y en la vida profesional, a aprender y pensar por sí mismos, a ser capaces de encontrar razones por las que vale la pena tomar decisiones y dedicar su vida; así como a respetar al prójimo en el marco de una comunidad global.

RESPONSABILIDAD PARA DECIDIR

En la propuesta curricular se incluyen seis competencias genéricas que se implican unas a otras de tal manera que constituyen una interacción compleja. Estas competencias son: comunicación, liderazgo intelectual, organización de personas y ejecución de tareas, innovación y cambio, perspectiva global humanista y el manejo de sí.

PROGRAMA ACADÉMICO

El programa académico cuenta, además, con el área de Formación Propedéutica en temas selectos como: Arte, Sociología, Cálculo, Biología, Química, Ciencias de la Salud, Física, Dibujo, Contabilidad, Economía, Administración, Derecho y Comunicación.

IDIOMAS

El Centro Intercultural de Lenguas de la Universidad Iberoamericana Puebla (CIDEL) comparte el objetivo general de la Ibero, formar hombres y mujeres con y para los demás. Esto significa que, además de transmitir conocimientos, enriquecerlos con la investigación y aplicarlos en el servicio, buscamos participar en la construcción de la justicia social, la paz y la defensa de la dignidad humana en México y otras comunidades del mundo.

FORMACIÓN CULTURAL Y DEPORTIVA

El programa ofrece diversas opciones en actividades culturales, deportivas y estéticas, para la formación integral del estudiante.

AMBIENTE

La Preparatoria Ibero ofrece un ambiente de plena libertad de creencias. No se molesta ni se pone traba alguna a ningún maestro o alumno por razón de sus convicciones religiosas, filosóficas o políticas. Sólo se pide a unos y otros que conozcan y respeten los principios y valores de la propia institución.

PERFIL DE LOS EGRESADOS DE LA PREPARATORIA IBERO

- Son hombres y mujeres que se preparan para su inserción en la sociedad con sentido crítico para superar toda manipulación y con espíritu solidario para responder a las necesidades de los demás.
- Asumen su misión de educarse y desarrollan sus competencias en un ambiente responsable y de progresiva autonomía. Manifiestan con sus actitudes y comportamientos su vocación cristiana.
- Al mismo tiempo actúan con respeto con quienes no tienen la misma fe, y en la medida de sus posibilidades están abiertos al diálogo ecuménico e interreligioso.
- Cooperan entre sí y crean un ambiente de verdadero compañerismo con espíritu de apertura, diálogo, tolerancia, comprensión y colaboración con todos sus compañeros y compañeras, teniendo especial sensibilidad en favor de los diferentes.
- Alientan entre sí y en sus ambientes la firme decisión por la honestidad y por luchar con sus medios en contra de la corrupción en cualquiera de sus formas.
- Participan en asociaciones y movimientos sociales, propios de su edad y de su responsabilidad como estudiantes, y contribuyen a la promoción y defensa

COMPROMISO SOCIAL

Este programa permite a los alumnos estar en contacto con su entorno y sensibilizarse para obtener un aprendizaje formativo. Para ello, los alumnos realizan [**IbP2009**]:

- Servicio Social (primero y segundo semestre): Servicio social en instituciones y organizaciones dedicadas a este fin.
- Experiencia Rural (tercero y cuarto semestre): Servicio comunitario en varias poblaciones de Puebla y estados circunvecinos.
- Experiencia Laboral (quinto y sexto semestre): Prácticas en establecimientos determinados que le proporcionen un acercamiento con el ámbito laboral.

CAPITULO 3. ANÁLISIS DEL SISTEMA

3.1. Planteamiento del Problema

Se requiere un sistema de información para realizar consultas y manipular la información almacenada en una base de datos; la interfaz del sistema deberá ser de fácil manipulación para el usuario. Se propone un Sistema Web de Administración Escolar, en sus planteles Puebla, Tlaxcala y Veracruz, estándar para las preparatorias de Universidad Iberoamericana Plantel Puebla. Este sistema estará conformado por un control escolar de gestión, inscripción y reinscripción de alumnos, un control de grupos, materias y profesores, un control de acceso para usuarios, formularios para captura de calificaciones por parte del personal docente, generación de consultas Web y reportes para alumnos, profesores y directivos, impresión de certificados.

Existirán cuatro tipos de usuarios para el acceso y manipulación del sistema: administrador, auxiliar, profesor y alumno. Cada usuario tendrá los permisos correspondientes a su tipo de usuario, esto es con el fin de restringir el acceso a procesos del sistema para cada tipo de usuario.

El sistema será una aplicación Web, donde el acceso estará restringido por un usuario y una clave asignados por el administrador, para que de esta manera sea accesible a todos usuarios autorizados en cualquier lugar y en cualquier momento que se requiera.

3.1.1. Especificación de Requerimientos

Se requiere un sistema Web de control escolar, este sistema deberá contar con las tareas propias para el mantenimiento de un sistema, como son: Altas, Bajas, Modificaciones, Consultas y Reportes.

- La información que se va a almacenar en la base de datos y que se va a manipular en este sistema es información de alumnos, profesores, información del plantel e información de pagos.
- El sistema se ejecutará en el Web y su acceso será restringido.
- El acceso al sistema se realizará a través de un usuario y clave asignados.
- Deberá haber diferentes tipos de usuarios para el acceso al sistema. Esto es, cada usuario podrá realizar sólo las tareas permitidas por su tipo de usuario (o permisos asignados).
  - *Usuario Administrador*: Tendrá acceso a todos los procesos del sistema excepto a los módulos destinados para el registro de información propia de los alumnos.
  - *Usuario Profesor*: Tendrá acceso sólo a Altas, Bajas, Modificaciones Consulta y Búsqueda de calificaciones.
  - *Usuario Auxiliar*: Tendrá acceso a todos los procesos del sistema, esto es, este tipo de usuario podrá manipular toda la información del sistema.
  - *Usuario Alumno*: Este tipo de usuario, sólo podrá hacer búsqueda de calificaciones.
- En el módulo de altas, se deben registrar todos los datos requeridos para los alumnos, profesores, planteles y pagos. Este proceso sólo lo podrán realizar usuarios con permisos para dar de alta datos.
- En el módulo de bajas, se debe dar de baja los datos que se elijan de acuerdo a los filtros. Este proceso sólo lo podrán realizar usuarios con permisos para eliminar datos.
- En el módulo de modificaciones, se podrán realizar modificaciones a los datos correspondientes a los filtros que se elijan. Este proceso sólo lo podrán realizar usuarios con permisos para modificar datos.

- En el módulo de consultas, se podrán realizar consultas para los datos almacenados en la base de datos; se dará al usuario la opción para visualizar sólo los datos que desea ver, el reporte se desplegará en pantalla y el sistema también dará la opción de imprimir este reporte. Este proceso sólo lo podrán realizar usuarios con permisos para consultas.
- El módulo de búsqueda fungirá como una consulta más específica; el sistema le pedirá al usuario los datos del registro que desea ver; esto es para que el sistema despliegue los datos correspondientes a ese registro y además dará la opción de impresión del reporte.
- En el módulo reportes, el sistema dará opción al usuario para elegir los datos que desea visualizar, luego entonces el sistema sólo deberá crear y almacenar en algún dispositivo de salida el reporte en formato Excel. En este módulo también existirá la opción para imprimir certificado" y libro de control de folios de certificados.
- Para permitir la impresión de certificados de alumnos, el alumno deberá haber concluido el tercer grado y haber cubierto satisfactoriamente el número total de materias establecidas por la institución; esto es que las calificaciones de cada materia deben ser mayores a 5.
- En el reporte del libro de control de folios de certificados, se visualizarán los folios que se han generado para los certificados.
- Se deberá registrar la siguiente información de alumnos: nombre, folio de ingreso, matrícula, periodo, fecha de solicitud de examen, fecha de examen, hora de examen, salón, entrevistador, fecha de entrevista, hora de entrevista, pago, folio recibo de pago, estado civil, medio de transporte, trabaja, giro de la empresa, escuela de procedencia, calle, número interior, número exterior, colonia, cp, ciudad, estado, país, teléfono, teléfono celular, otro teléfono, email, calificación de examen, número de cuenta, grupo, nivel de inglés, estado del alumno, pago por entrega extemporánea de docs, primer pago de inscripción, segundo pago de inscripción, primer pago de aportación única, segundo pago de aportación única, pago de curso propedéutico, solicitud de beca, deportes taller artístico, registro médico, fotos, carta compromiso, recibo de documentación, acta de nacimiento original, acta de nacimiento copia, curp, certificado parcial primer año, certificado de secundaria original, certificado de primaria original y copia, carta de buena conducta, constancia de estudios de secundaria, calificaciones, comprobante de domicilio, test de estilos de aprendizaje, número de cuenta del alumno, No. de IMSS/ISSSTE, grupo sanguíneo, enfermedades que ha padecido o padece, alérgico a algún factor externo?, alérgico a algún medicamento ?, usa anteojos graduados, ha sido intervenido quirúrgicamente, ha consumido o consume alcohol, consume tabaco, consume drogas, tiene algún tatuaje o perforación?.
- Se deberá registrar la siguiente información para profesores: número de cuenta, nombre, dirección, ciudad, estado, país, teléfono de domicilio, teléfono celular, fecha de nacimiento, correo, nombre de usuario, password, plantel.
- Se deberá registrar la siguiente información para los grupos: plantel, semestre, nombre de grupo, folio del alumno.
- Se deberá registrar la siguiente información para las materias: plantel, nombre, grupo y profesor.
- Se deberá registrar la siguiente información para los padres y tutores: folio del alumno, nombre, dirección, cp, ciudad, estado, país, teléfono domicilio, teléfono oficina, teléfono celular, otro teléfono, grado académico.
- Se deberá registrar la siguiente información para el período escolar: períodos existentes, nuevo semestre, nuevo año.
- Se deberá registrar la siguiente información para planteles: nombre, dirección colonia, teléfono, ciudad, estado, email.
- Se deberá registrar la siguiente información para calificaciones: número de cuenta del alumno, materia plantel, grupo, período, mes, calificación.

- Se deberá registrar la siguiente información para actividades artísticas: nueva actividad, instructor, actividades existentes.
- Se deberá registrar la siguiente información para deportes: deportes existentes, nuevo deporte e instructor.
- Se deberá registrar la siguiente información para pagos: tipo de pago existente y nuevo tipo de pago.
- Se deberá registrar la siguiente información para becas: número de cuenta del alumno, porcentaje otorgado y período.

3.1.2. Antecedentes

La Universidad Iberoamericana en su plantel Puebla, tiene más de 25 años, y maneja un sistema de administración escolar que cumple con los estándares y necesidades propios de la educación superior, pero que no es adaptable para su aplicación en las preparatorias de reciente creación. Por tanto, surge la necesidad de crear un sistema especializado para las Preparatorias UIA en sus tres planteles, ya que actualmente se realizan tareas de organización y archivo manualmente, y se manejan de archivos elaborados en Excel que son difíciles de controlar al momento de tratar de controlar una matrícula de alumnos cada vez más amplia.

3.1.3. Objetivos

El objetivo general de este proyecto es realizar un sistema de Bases de Datos basado en Web, el cual cumpla las funciones de la administración escolar para nivel preparatoria (caso de estudio: Preparatorias de la Universidad Iberoamericana.)

Específicamente el sistema debe cumplir con los siguientes objetivos:

- Control escolar que por el momento gestione a 150 alumnos y 50 profesores (plantel actual) pero que cada año incremente la matricula en un 5%.
- Control de alumnos inscritos y de nuevo ingreso con formato de inscripción, ficha médica y registro de documentos.
- Control de profesores para registrar datos personales y capturar calificaciones.
- Control de grupos que incluyen 3 años de preparatoria, asignando las materias con su respectivo profesor.
- Consultas Web por parte del administrador, profesor y alumno.
- Reportes por parte del administrador, profesor y alumno.
- Impresión de certificados de los alumnos que concluyeron su estadía en la institución.

3.1.4. Alcances

Debido a que el sistema se ejecutará en el Web, se proporcionará información de forma inmediata a los usuarios con acceso autorizado.

El control de la información de alumnos, profesores e información administrativa se manipulará de forma controlada y segura.

La centralización de la información asegura la confiabilidad de los datos, así como la seguridad de estos.

El sistema proporcionará información en pantalla, así como, información impresa e información guardada en formato Excel.

3.1.5. Metodología

Utilizando la metodología del ciclo de vida básico, también llamado modelo en cascada o modelo lineal secuencial, se seguirá un enfoque sistemático secuencial para el desarrollo del software y el diseño e implementación de una base de datos relacional, comenzando en un nivel de sistemas y progresando con el análisis, diseño, codificación, pruebas y mantenimiento. Los pasos a considerar son:

- Especificación de Requerimientos del proyecto.
- Análisis de los requerimientos utilizando algunos elementos de notación UML (Lenguaje Unificado de Modelado) dentro de todo el desarrollo del proyecto.
- Diseño de la base de datos relacional.
- Diseño de la aplicación basada en Web.
- Implementación del sistema con PHP y MySQL, Apache.
- Pruebas del sistema.

3.1.6. Definición de Requerimientos Funcionales

- El acceso al sistema será restringido por un usuario y password asignados por el administrador del sistema.
- Los tipos de usuarios para el acceso al sistema son: administrador, auxiliar, profesor y alumno. Cada tipo de usuario tendrá sus restricciones o permisos para realizar tareas específicas en el sistema.
- El sistema permitirá realizar los procesos necesarios para el mantenimiento del sistema.
- Dependiendo el tipo de usuario se podrán realizar las siguientes tareas o procesos.
    - El usuario de tipo auxiliar tendrá acceso a los módulos de altas, bajas, modificaciones, consultas, reportes e impresión de certificados. Además tendrá acceso a módulos destinados al registro de los alumnos.
    - El usuario de tipo administrador tendrá acceso a los módulos de altas, bajas, modificaciones, consultas, reportes e impresión de certificados.
    - El usuario de tipo profesor sólo podrá dar de alta, dar de baja, modificar, consultar y hacer búsqueda de calificaciones.
    - El usuario de tipo alumno sólo tendrá acceso para hacer búsqueda de sus calificaciones.

3.1.7. Definición de Requerimientos No Funcionales

Para que el usuario pueda visualizar las páginas correspondientes al sistema, el usuario debe contar con un equipo cliente con opción para la conexión a un servidor web, donde reside el sistema en cuestión.

- El equipo del usuario debe tener instalado algún browser de su elección para poder accesar a las pantallas del sistema.
- La velocidad de respuesta del sistema, dependerá de los componentes del hardware y en gran medida de la velocidad y tipo de conexión del equipo cliente y del servidor.

3.1.8. Diagrama de Casos de Uso

Dentro del análisis inicial realizado se llegó a determinar, usando un Modelo de Casos de Uso lo siguiente:

Actores

Representaciones de entidades externas que integran el sistema durante su ejecución:

- Usuario Registrado.
- Administrador

Casos de Uso

Representan o describen procesos a realizar por el sistema (**Figura 3.1**):

- Altas
- Bajas
- Modificaciones
- Reportes
- Consultas

Relación entre Actores y Casos de Uso

Se tiene una descripción de los servicios ofrecidos por el sistema bajo el Modelado de un diagrama de Casos de Uso, donde cada caso de uso es todo un procedimiento que se ha de llevar. En la **Figura 3.1** sólo se muestra en forma general la interacción y relación de los diversos actores con los Casos de Uso.

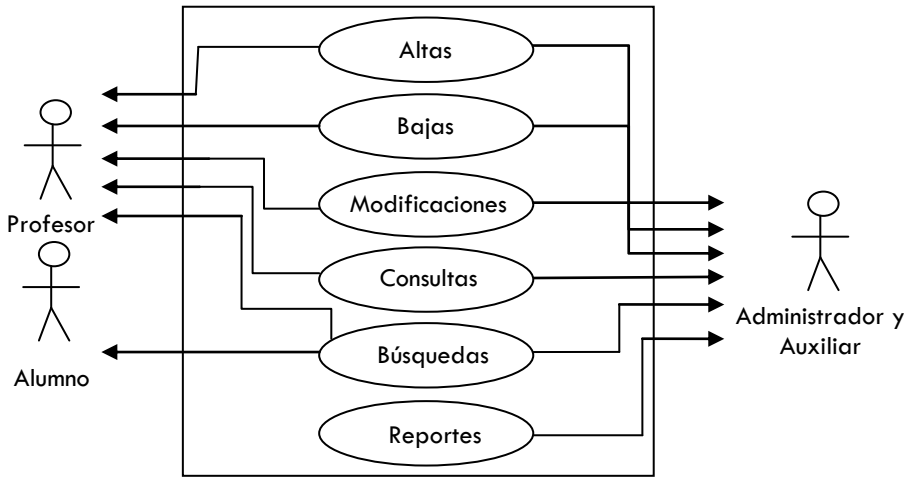


Figura 3.1 Diagrama General de Casos de Uso

Siendo una de las herramientas utilizadas los diagramas de casos de uso para el análisis del sistema y una vez determinados los principales casos de uso, se hace un análisis de las actividades que se realizarán dentro de cada uno de ellos y se realiza una serie de diagramas de subcasos de uso considerando cada uno de los casos de uso principales y los actores que intervienen en ellos.

Como el sistema tendrá diferentes acciones y opciones a seguir dependiendo del actor, se ha realizado un diagrama de subcasos de uso por cada uno de los actores que intervienen en el sistema.

3.1.8.1. Casos de Uso y sus Descripciones

ALTAS

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de registrar la información de los alumnos, padres, profesores, planteles, materias, calificaciones y usuarios del sistema. Los actores de este caso serán el administrador, auxiliar; el profesor sólo tendrá acceso a alta de calificaciones.

Flujo Básico

1. El usuario ingresa los datos de cualquiera de las siguientes entidades: alumnos, padres, profesores, materias, calificaciones e ir a paso 3.
2. El sistema indicará al usuario los datos (mandatorios y opcionales) que se deben capturar según la entidad que se esté registrando. Una vez capturados los datos, se procede a aceptar el registro de estos.
3. Al realizar el registro con éxito se desplegará un mensaje para avisar que los datos se guardaron correctamente. En otro caso el sistema avisará que no se guardaron los datos correctamente.

Flujos Alternativos

- En el punto 2, en caso de no haber capturado los datos que son mandatorios, se desplegará un mensaje de error para avisar que faltan datos, y se dará la oportunidad de regresar para capturar los datos faltantes.

Precondiciones

- Para que el usuario pueda registrar materias, calificaciones, certificados y datos de padres de familia de un alumno, los datos de dicho alumno deben registrarse previamente.

BAJAS

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de dar de baja la información de los alumnos, padres, profesores, planteles, materias, calificaciones y usuarios del sistema. Los actores de este caso serán el administrador, auxiliar y el profesor. El profesor sólo tendrá acceso a dar de baja calificaciones.

Flujo Básico

1. El sistema indicará al usuario, los datos necesarios que debe proporcionar para realizar la búsqueda de los datos que desea dar de baja, e ir a paso 3.

2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema dará la opción al usuario para confirmar la baja de esta información.
3. Una vez aceptado o confirmado el paso 2, la información se dará de baja.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda dar de baja materias, calificaciones, certificados y datos de padres de familia de un alumno, los datos de dicho alumno deben registrarse previamente.

Postcondiciones

- Una vez que se ha dado de baja la información, los datos correspondientes ya no aparecerán en reportes o consultas que se generen en el sistema.

MODIFICACIONES

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de realizar modificaciones a la información de los alumnos, padres, profesores, planteles, materias, calificaciones y usuarios del sistema. Los actores de este caso serán el administrador, auxiliar y el profesor. El profesor sólo tendrá acceso a realizar modificaciones de las calificaciones.

Flujo Básico

1. El sistema indicará al usuario, los datos necesarios que debe proporcionar para realizar la búsqueda de los datos que desea modificar, e ir a paso 3.
2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema proporcionará los datos en modo de escritura para que el usuario pueda realizar la modificación de estos. El sistema dará la opción para enviar la modificación de los datos.
3. Una vez aceptado o confirmado el paso 2, la información se almacenará en la base de datos con las respectivas modificaciones realizadas.
4. El sistema enviará un mensaje para indicar que se guardaron los datos correctamente.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda modificar materias, calificaciones, certificados y datos de padres de familia de un alumno, los datos de dicho alumno deben registrarse previamente.

CONSULTAS

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de consultar la información de los alumnos, padres, profesores, planteles, materias, calificaciones y usuarios del sistema. Los actores de este caso serán el administrador, auxiliar, profesores y alumnos. Los profesores y alumnos sólo tendrán acceso a realizar consulta de calificaciones.

Flujo Básico

1. El sistema dará la opción al usuario para elegir los datos de la información que desea visualizar e ir a paso 2.
2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema desplegará la información correspondiente y dará la opción de impresión.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda realizar consultas de alumnos, materias, calificaciones, certificados y datos de padres de familia, los datos deben registrarse previamente.

Poscondiciones

- Los datos resultantes de la búsqueda, se desplegarán sólo para consultar, esto es, no se permitirá modificar los datos.

BUSQUEDAS

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de consultar de manera más específica información de cada alumno, padre, profesor, plantel y materia. Los actores de este caso serán el administrador, auxiliar, profesores y alumnos. Los profesores y alumnos sólo tendrán acceso a realizar búsqueda de calificaciones.

Flujo Básico

1. El sistema indicará al usuario, los datos necesarios que debe proporcionar para realizar la búsqueda de los datos e ir a paso 2.
2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema desplegará en pantalla la información correspondiente y dará la opción para imprimir.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda realizar consultas de alumnos, materias, calificaciones, certificados y datos de padres de familia, los datos deben registrarse previamente.

REPORTES

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de generar reportes sobre la información de los alumnos, padres, profesores, planteles, materias, calificaciones y usuarios del sistema. Los actores de este caso serán el administrador, auxiliar.

Flujo Básico

1. El sistema indicará al usuario, los datos necesarios que debe proporcionar para realizar la búsqueda de los datos y generar el reporte correspondiente, e ir a paso 2.
2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema creará los reportes en formato Excel y dará la opción para guardar o abrir el reporte.
3. Una vez aceptado o confirmado el paso 2, los reportes se imprimirán o guardarán en formato Excel.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda generar reportes de alumnos, materias, calificaciones, certificados y datos de padres de familia, los datos deben registrarse previamente.

3.1.8.2. Diagramas de Secuencia

El diagrama de secuencias del UML muestra la forma en que los objetos se comunican entre sí al transcurrir el tiempo, a continuación se muestra en la **figura 3.2** los diagramas de secuencia para los cuatro usuarios: administrador, auxiliar, profesor y alumno.

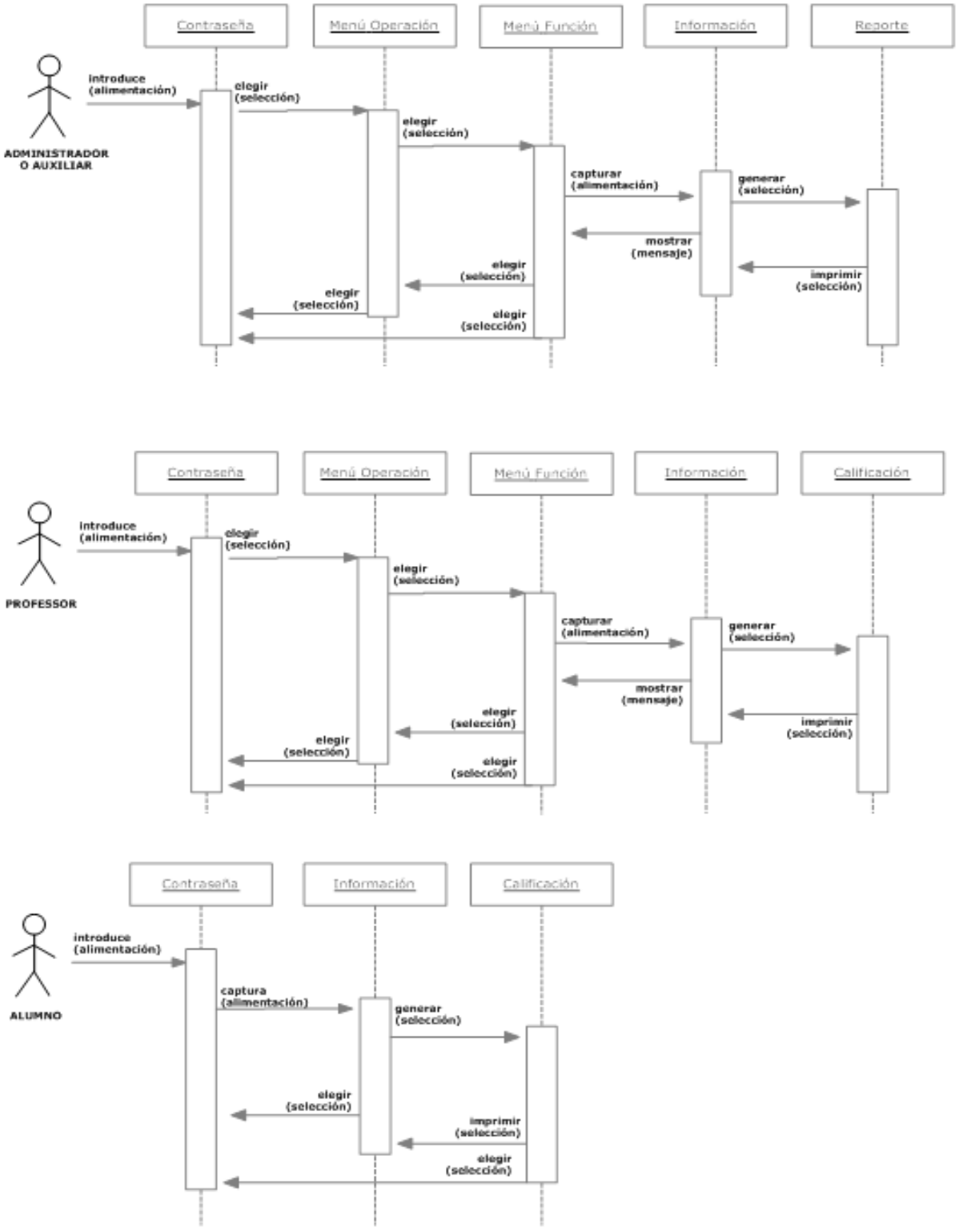


Figura 3.2 Diagramas de Secuencia para los usuarios del sistema

3.1.8.3. Subcasos de Uso

Una vez definido los casos de uso y tomando en cuenta que el acceso a las diferentes tareas que se pueden realizar en el sistema, va a estar restringido por el tipo de usuario que haya ingresado; donde los tipos de usuario son: administrador, auxiliar, profesor y alumno; se ha hecho la consideración de subdividir algunos de los casos de uso que contienen más procesos internos que se pueden agrupar como subcasos de uso tomando en cuenta a cada uno de los tipos de usuario definidos. A continuación se muestran los diagramas de subcasos para todos los usuarios en las **figuras 3.3, 3.4 y 3.5**

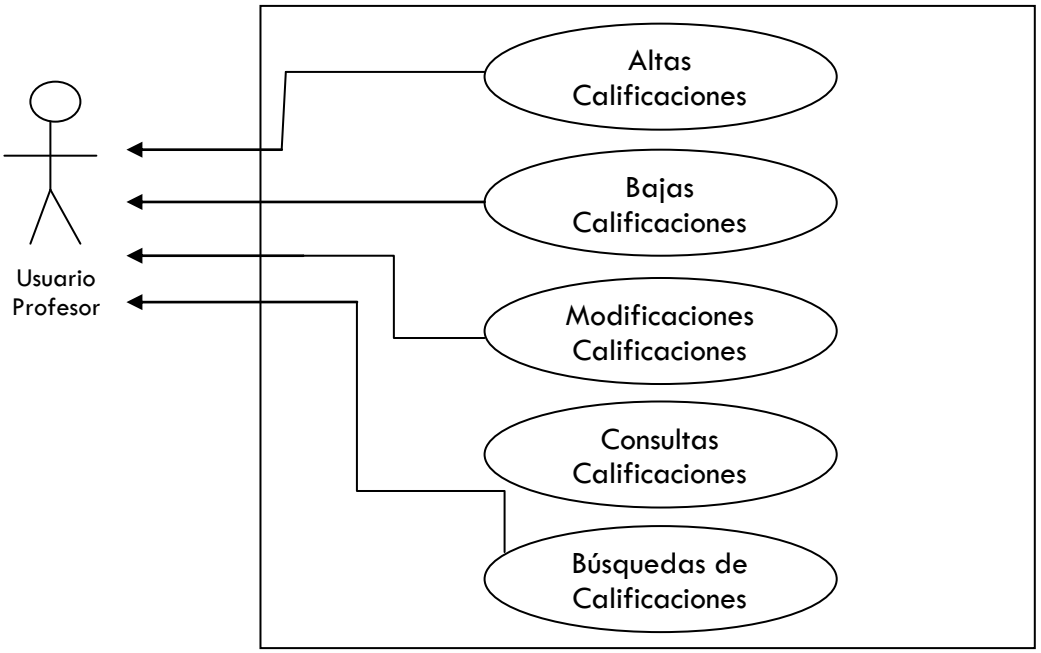


Figura 3.3 Diagrama de Subcasos de Uso del usuario Profesor

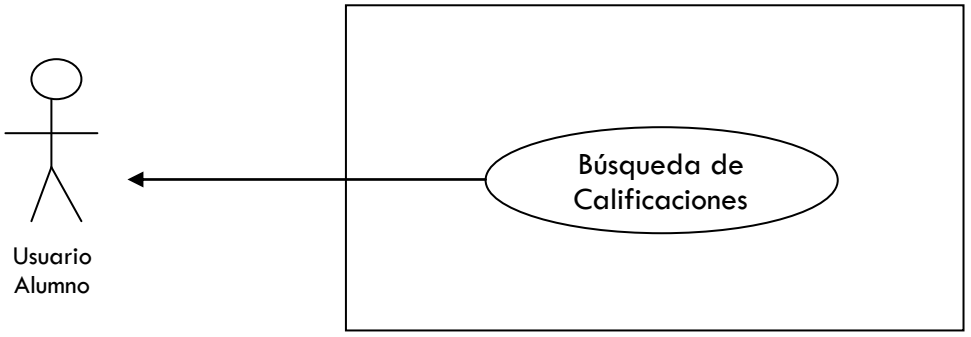


Figura 3.4 Diagrama de Subcasos de Uso del usuario Alumno

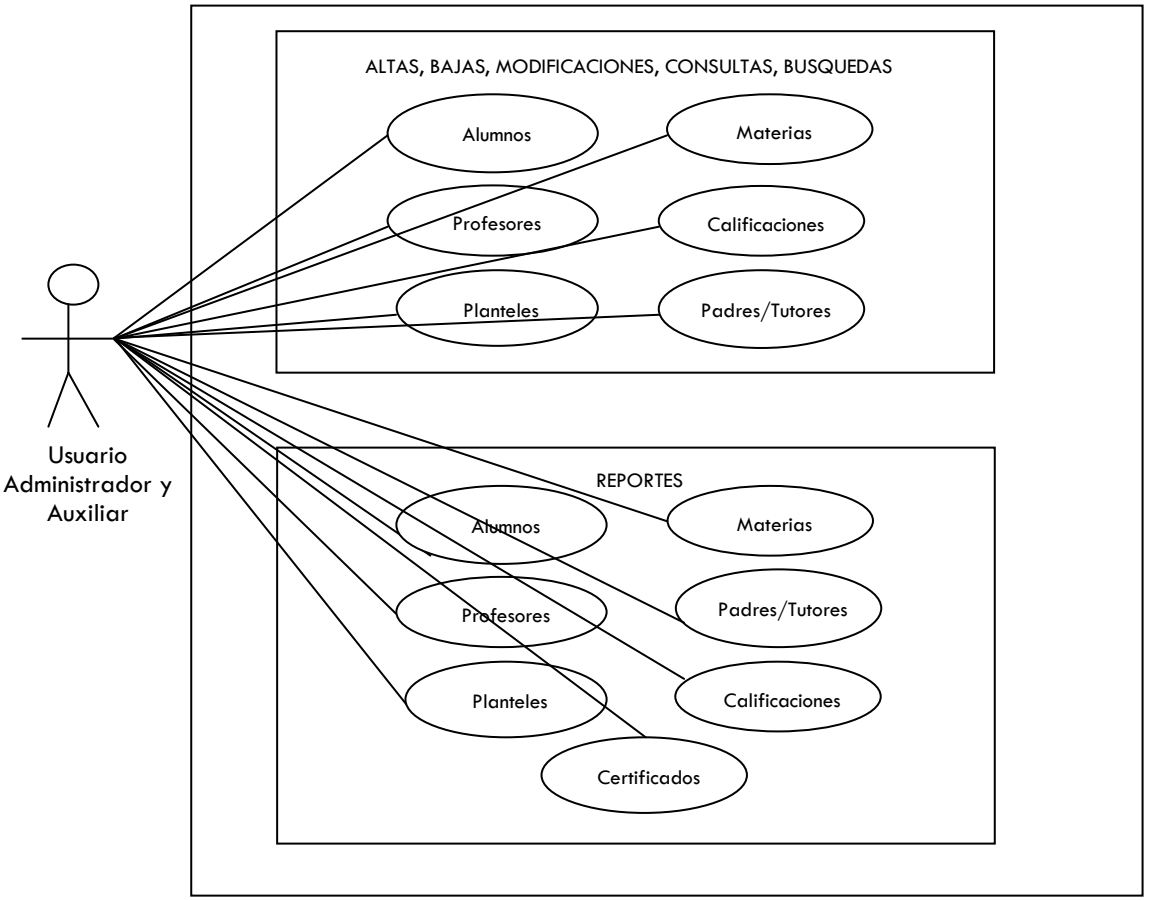


Figura 3.5 Diagrama de Subcasos de Uso del usuario Administrador y Auxiliar

SUBCASO DE USO ALTA DE ALUMNOS, PROFESORES, PLANTELES, MATERIAS, CALIFICACIONES Y PADRES/TUTORES

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de registrar la información referente a los alumnos, profesores, planteles, materias, calificaciones y padres/tutores.

Flujo Básico

1. El sistema indicará al usuario, los datos que debe capturar para registrar al alumno, profesor, planteles, materia, calificación y padre/tutor.
2. El sistema indicará al usuario los datos (mandatorios y opcionales) que se deben capturar según la entidad (alumno, profesor, planteles, materia, calificación y padre/tutor) selccionada. Una vez capturados los datos, se procede a aceptar el registro de estos e ir a paso 3.
3. Al realizar el registro con éxito se desplegará un mensaje para avisar que los datos se guardaron correctamente. En otro caso el sistema avisará que no se guardaron los datos correctamente.

Flujos Alternativos

- En el punto 2, en caso de no haber capturado los datos que son mandatorios, se desplegará un mensaje de error para avisar que faltan datos, y se dará la oportunidad de regresar para capturar los datos faltantes.

Precondiciones

- Para que el usuario pueda registrar alumnos, se debe haber registrado previamente algún plantel.

SUBCASO DE USO BAJA DE ALUMNOS, PROFESORES, PLANTELES, MATERIAS, CALIFICACIONES Y PADRES/TUTORES

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de dar de baja la información referente a los alumnos, profesores, planteles, materias, calificaciones y padres/tutores.

Flujo Básico

1. El sistema indicará al usuario, los datos necesarios que debe proporcionar (filtros) para realizar la búsqueda de los datos que desea dar de baja, e ir a paso 3.
2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema dará la opción al usuario para confirmar la baja de esta información.
3. Una vez aceptado o confirmado el paso 2, la información se dará de baja.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda dar de baja los datos del alumno, profesor, plantel, materia, calificación y padre/tutor, este debe haber sido registrado previamente.

Poscondiciones

- Una vez que se ha dado de baja la información del alumno, profesor, plantel, materia, calificación y padre/tutor, los datos correspondientes ya no aparecerán en reportes o consultas que se generen en el sistema.

SUBCASO DE USO MODIFICACION DE ALUMNOS, PROFESORES, PLANTELES, MATERIAS, CALIFICACIONES Y PADRES/TUTORES

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de modificar la información referente a los alumnos, profesores, planteles, materias, calificaciones y padres/tutores.

Flujo Básico

1. El sistema indicará al usuario, los datos necesarios que debe proporcionar para realizar la búsqueda de los datos que desea modificar, e ir a paso 3.
2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema proporcionará los datos en modo de escritura para que el usuario pueda realizar la modificación de estos. El sistema dará la opción para enviar la modificación de los datos.
3. Una vez aceptado o confirmado el paso 2, la información se almacenará en la base de datos con las respectivas modificaciones realizadas.
4. El sistema enviará un mensaje para indicar que se guardaron los datos correctamente.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda modificar información de alumnos, profesores, planteles, materias, calificaciones y padres o tutores; la información debe ser previamente registrada.

SUBCASO DE USO CONSULTA DE ALUMNOS, PROFESORES, PLANTELES, MATERIAS, CALIFICACIONES, PADRES/TUTORES Y CERTIFICADOS

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de consultar la información referente a los alumnos, profesores, planteles, materias, calificaciones y padres/tutores.

Flujo Básico

1. El sistema dará la opción al usuario para elegir los datos de la información que desea visualizar e ir a paso 2.
2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema desplegará la información correspondiente y dará la opción de impresión.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda realizar consultas de alumnos, materias, calificaciones, certificados y datos de padres de familia, los datos deben registrarse previamente.

Poscondiciones

- Los datos resultantes de la búsqueda, se desplegarán sólo para consultar, esto es, no se permitirá modificar los datos.

SUBCASO DE USO BUSQUEDA DE ALUMNOS, PROFESORES, PLANTELES, MATERIAS, CALIFICACIONES, PADRES/TUTORES Y CERTIFICADOS

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de realizar búsquedas para la información referente a los alumnos, profesores, planteles, materias, calificaciones y padres/tutores.

Flujo Básico

3. El sistema indicará al usuario los datos que debe proporcionar para realizar la búsqueda relacionada con estos datos e ir a paso 2.
4. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema desplegará la información correspondiente y dará la opción de impresión.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda realizar búsquedas de alumnos, materias, calificaciones, certificados y datos de padres de familia, los datos deben registrarse previamente.

Poscondiciones

- Los datos resultantes de la búsqueda, se desplegarán sólo para consultar, esto es, no se permitirá modificar los datos.

SUBCASO DE USO REPORTES DE ALUMNOS, PROFESORES, PLANTELES, MATERIAS, CALIFICACIONES, PADRES/TUTORES Y CERTIFICADOS

Descripción

En este caso de uso el usuario previamente registrado en el sistema, será capaz de generar reportes de la información referente a los alumnos, profesores, planteles, materias, calificaciones, padres/tutores y certificados. Estos reportes se imprimar de manera directa, o bien se guardarán en el equipo cliente en formato Excel.

Flujo Básico

1. El sistema indicará al usuario, los datos necesarios que debe proporcionar para realizar la búsqueda de los datos y generar el reporte correspondiente, e ir a paso 2.
2. Al realizarse con éxito la búsqueda de la información correspondiente a los datos que se proporcionaron en el paso 1, el sistema proporcionará los reportes, así como la opción para impresión de estos reportes o bien la generación de estos reportes en formato Excel.
3. Una vez aceptado o confirmado el paso 2, los reportes se imprimirán o guardarán en formato Excel.

Flujos Alternativos

- En el punto 2, en caso de que no exista información correspondiente a los datos (filtros) proporcionados por el usuario, el sistema enviará un mensaje de error, para avisar que no existe información correspondiente a estos datos.

Precondiciones

- Para que el usuario pueda generar reportes de alumnos, profesores, planteles, materias, calificaciones, certificados y datos de padres de familia, los datos deben registrarse previamente.
- Para que se pueda imprimir el certificado de un alumno, el alumno debe ser de 3 grado y haber concluido satisfactoriamente las materias.

CAPITULO 4. DISEÑO DEL SISTEMA

En este capítulo se va a presentar el diseño de la base de datos, esto incluye: el diagrama de navegación, el modelo conceptual, el modelo entidad-relación, la normalización de las relaciones y el modelo físico de la base de datos.

4.1. Arquitectura del Sistema

Se ha implementado un servidor Web, al cual almacena la base de datos y los archivos del sistema. Esta máquina espera a que lleguen las solicitudes de los clientes, tras la recepción de una solicitud, la procesa y envía la respuesta al cliente.

Los clientes hacen su solicitud a través de la Web y esperan la respuesta del servidor, en este caso los clientes pueden ser administradores o auxiliares que capturan, modifican y consultan información, estos también pueden generar reportes. También clientes son los profesores que capturan y consultan calificaciones, y los alumnos que consultan e imprimen sus calificaciones.

Esta arquitectura está diseñada para que cada plantel cuente con su propio servidor, pero que al mismo tiempo los tres planteles puedan compartir la información necesaria. Se muestra el diseño de esta arquitectura en la **Figura 4.1**

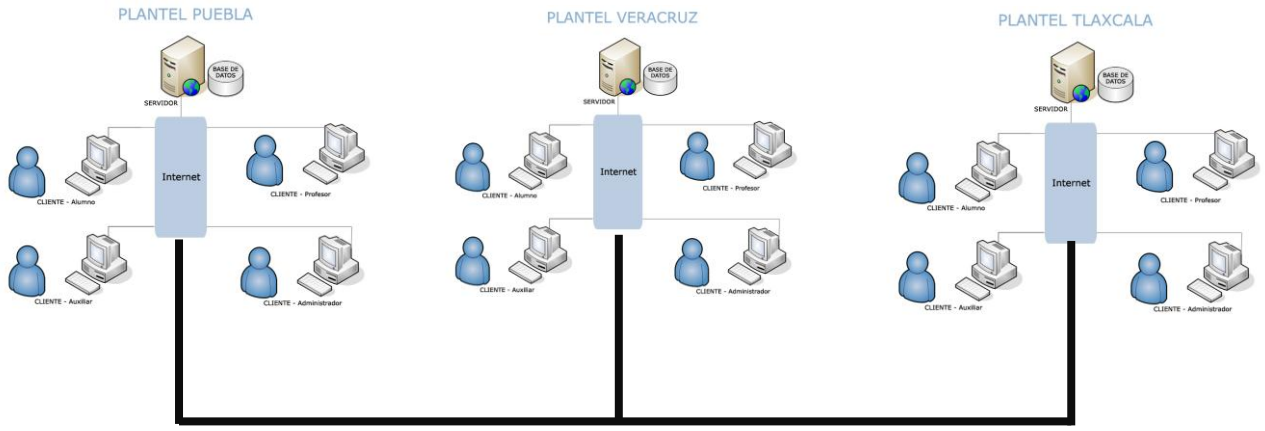


Figura 4.1 Arquitectura Cliente-Servidor del Sistema Web de Administración Escolar para Instituciones de Educación Media Superior

4.2. Diagrama de Navegación

En la **figura 4.2** se muestra un diagrama para explicar la navegación entre los contenidos del Sistema, la navegación dentro del sistema varia en cuanto a la validación que se haga sobre el tipo de usuario.

Diagrama de navegación para los procesos de: Altas, Bajas, Modificaciones, Consultas, Búsquedas y Reportes.

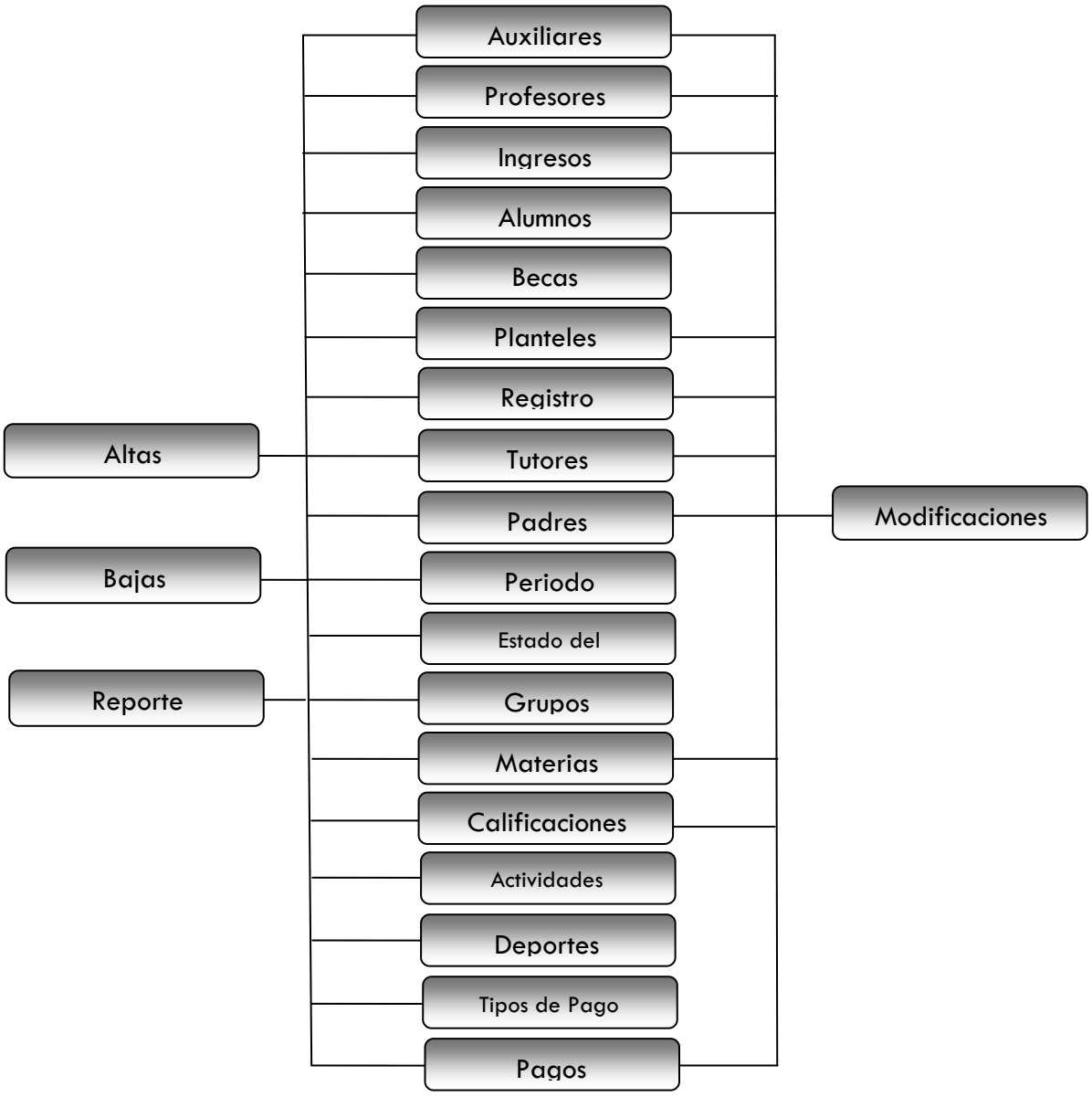


Figura 4.2 Diagrama de Navegación para: Altas, Bajas, Modificaciones y Reportes

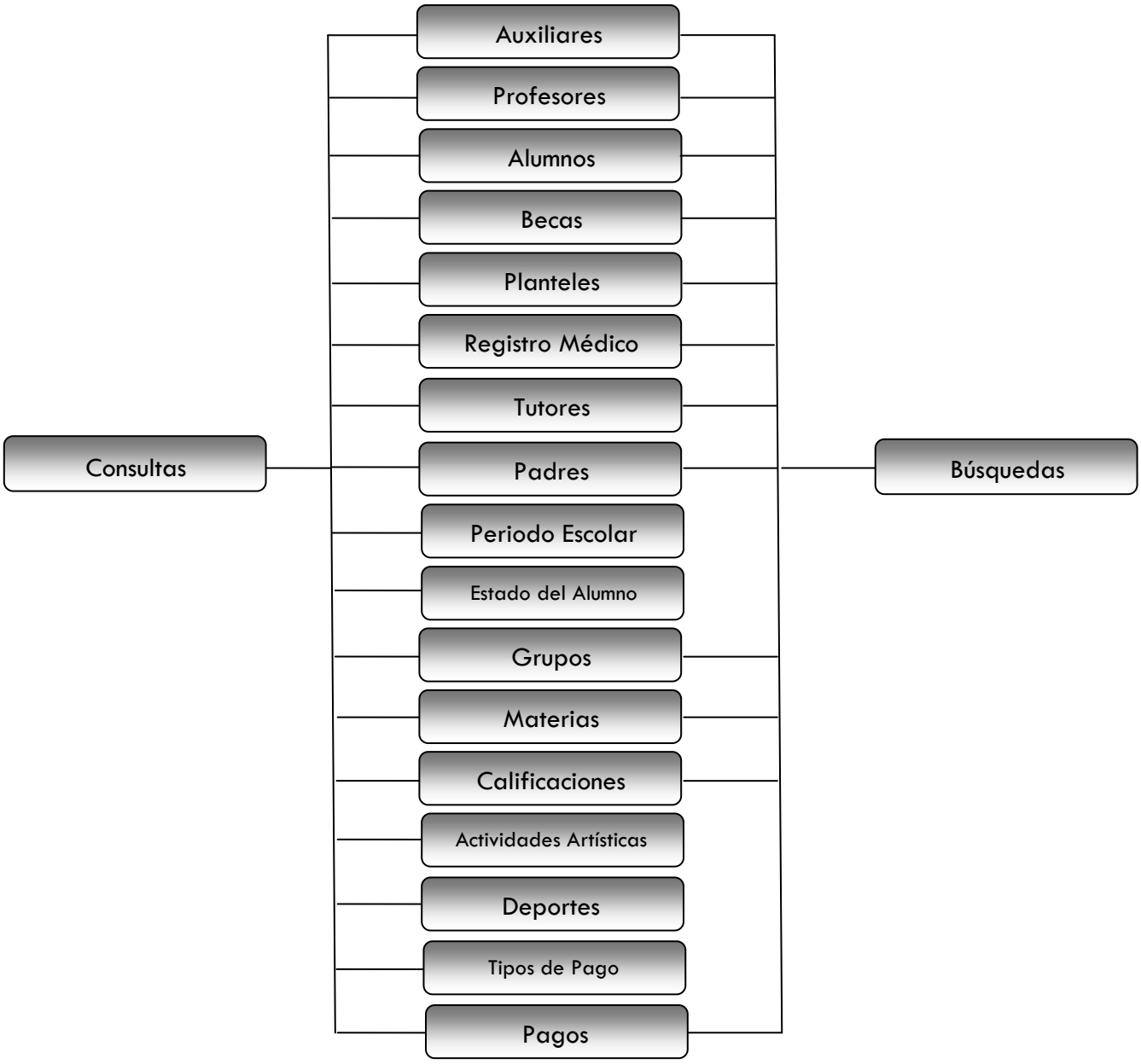
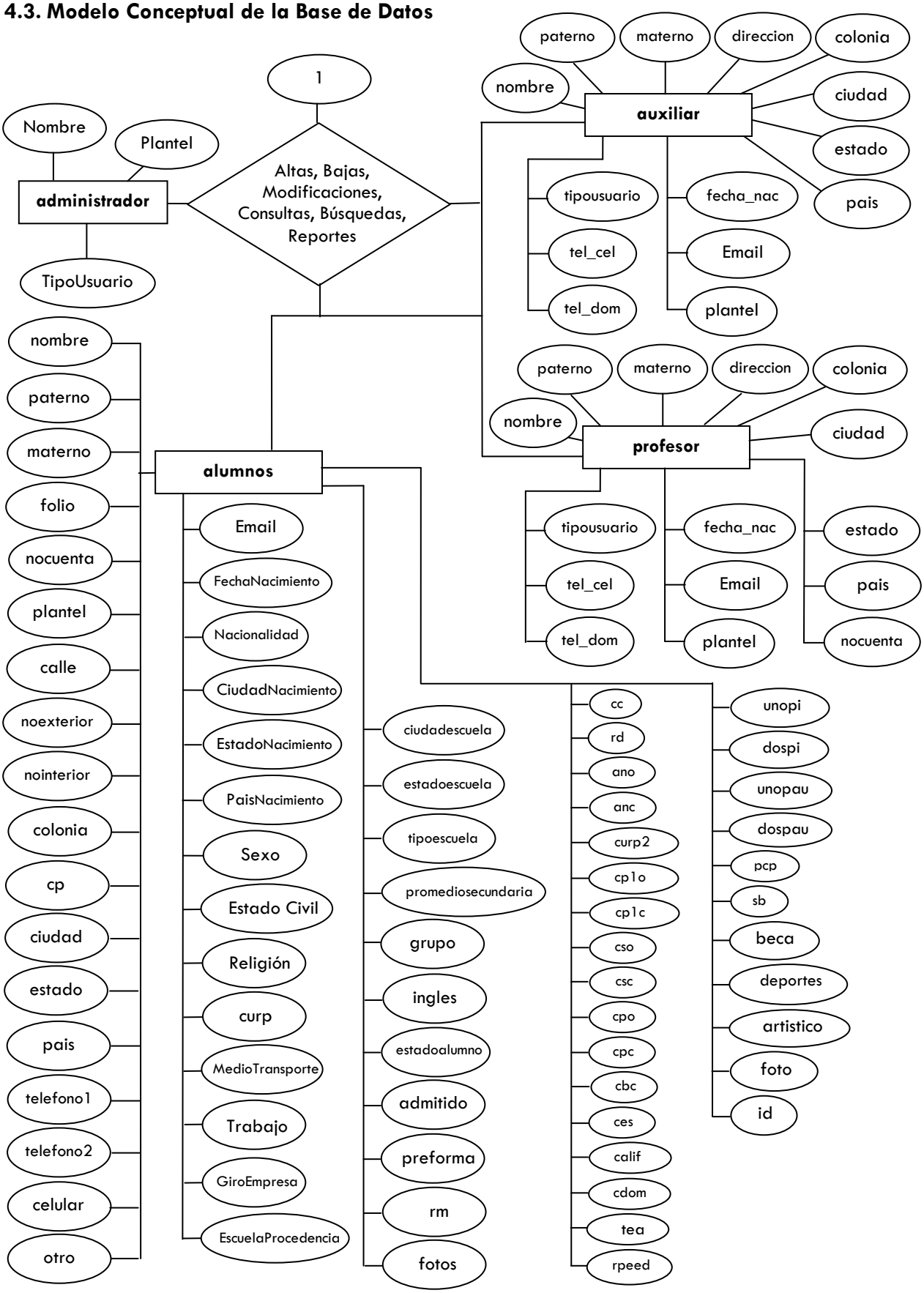
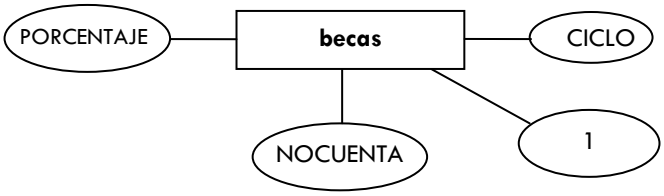
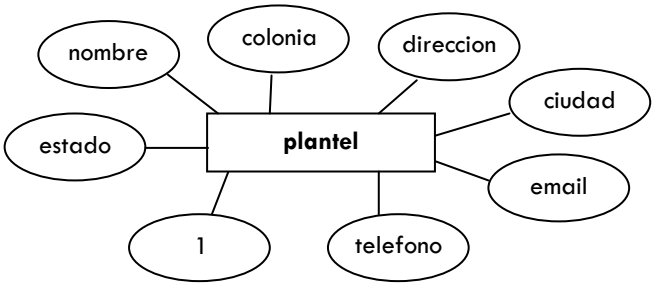


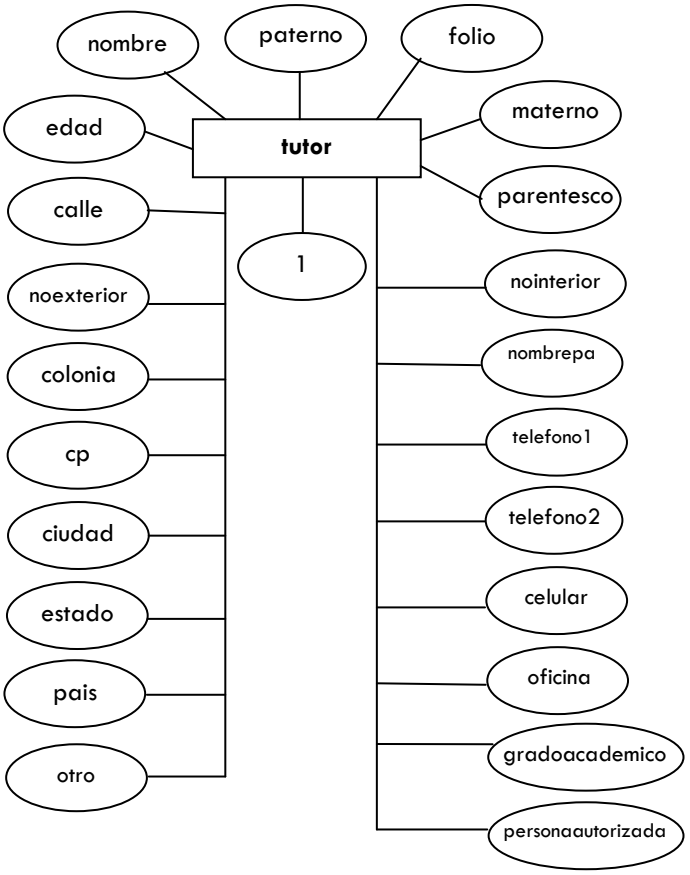
Figura 4.3 Diagrama de Navegación para: Consultas y Búsquedas

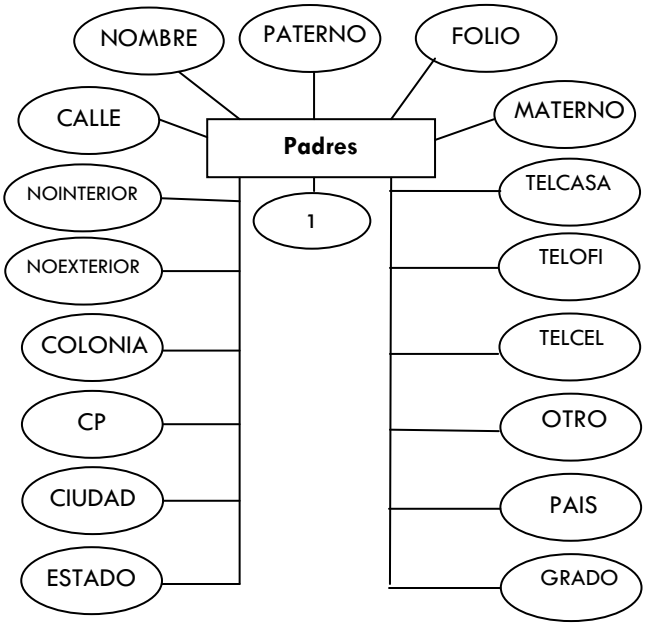
4.3. Modelo Conceptual de la Base de Datos

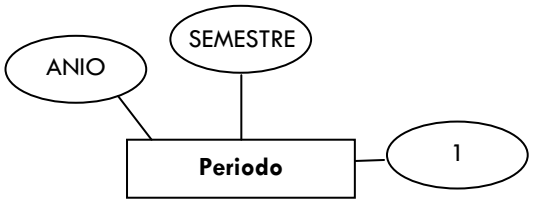


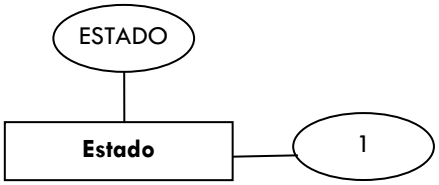

PORCENTAJE
becas
CICLO
NOCUENTA
1

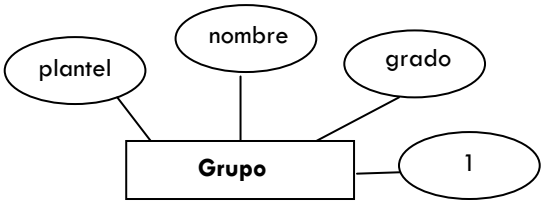

nombre
colonia
direccion
ciudad
estado
plantel
email
1
telefono

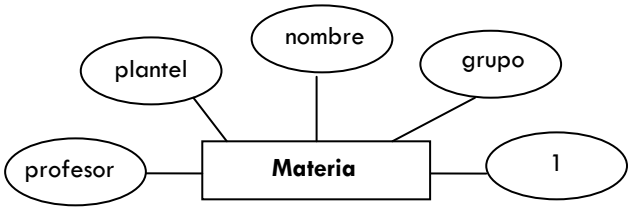

nombre
paterno
folio
edad
tutor
materno
calle
parentesco
1
noexterior
nointerior
colonia
nombrepa
cp
telefono1
ciudad
telefono2
estado
celular
pais
oficina
otro
gradoacademico
personaautorizada

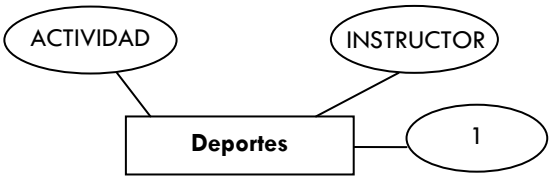

NOMBRE
PATERNO
FOLIO
CALLE
MATERNO
Padres
NOINTERIOR
TELCASA
1
NOEXTERIOR
TELOFI
COLONIA
TELCEL
CP
OTRO
CIUDAD
PAIS
ESTADO
GRADO

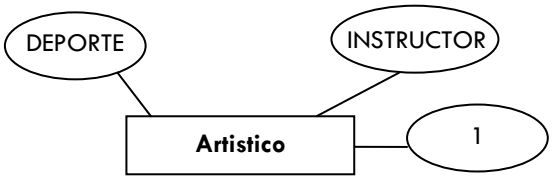

SEMESTRE
ANIO
Periodo
1

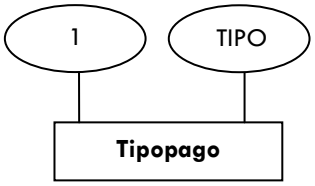

ESTADO
Estado
1

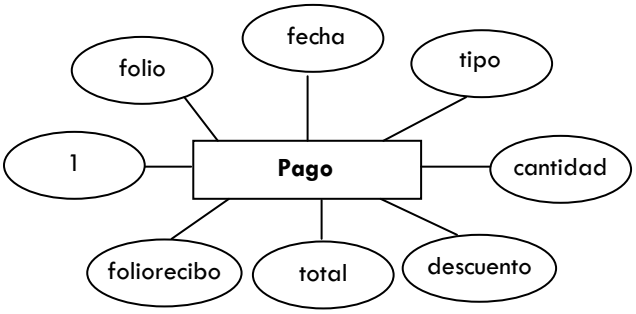

nombre
plantel
grado
Grupo
1

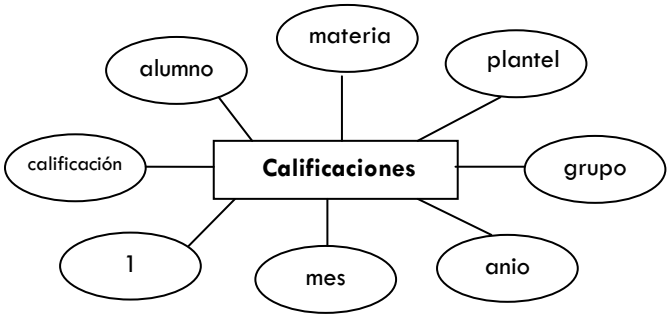

nombre
plantel
grupo
profesor
Materia
1

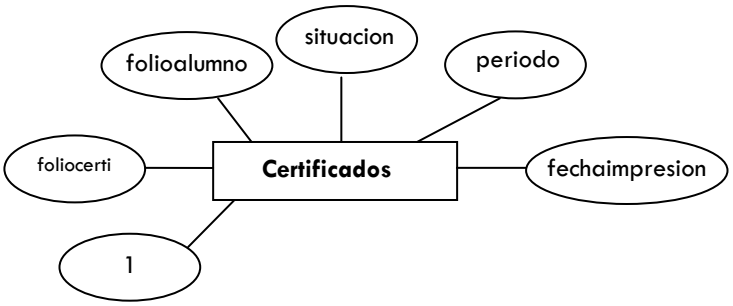

ACTIVIDAD
INSTRUCTOR
Deportes
1


DEPORTE
INSTRUCTOR
Artistico
1

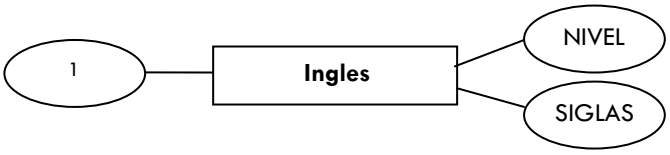

1
TIPO
Tipopago

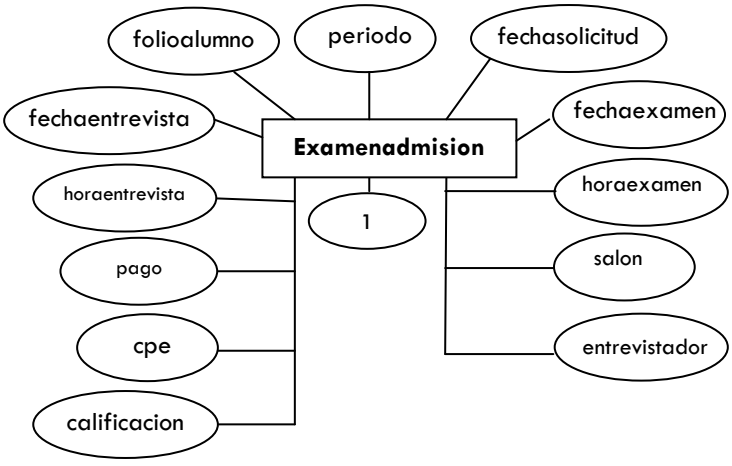

fecha
folio
tipo
1
Pago
cantidad
foliorecibo
total
descuento

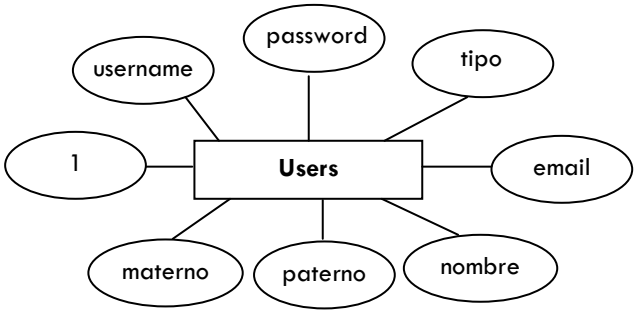

materia
alumno
plantel
calificación
Calificaciones
grupo
1
mes
anio

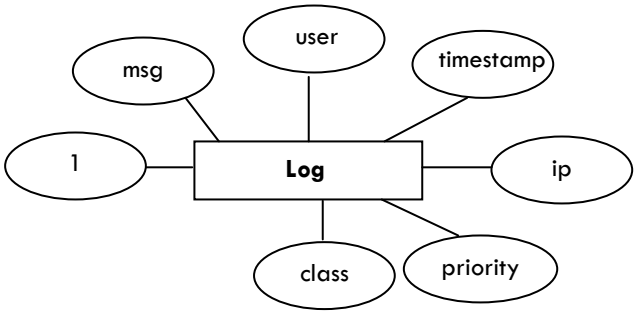

situacion
folioalumno
periodo
foliocerti
Certificados
fechaimpresion
1

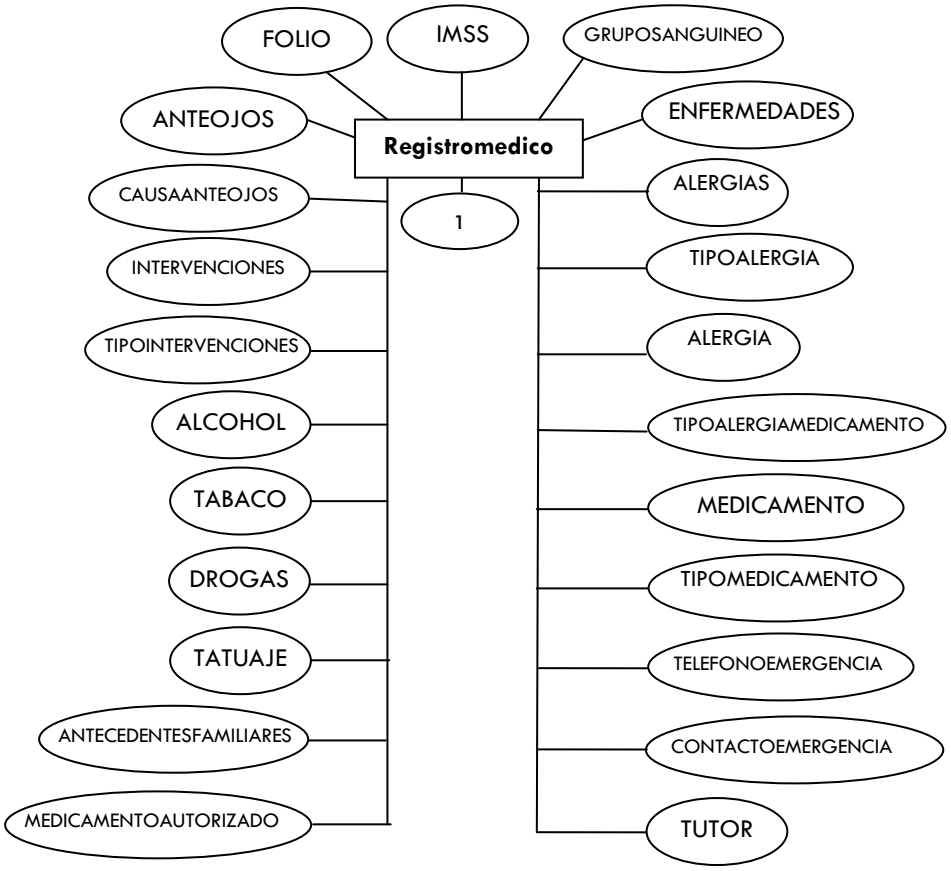
1
Fotos
foto


NIVEL
1
Ingles
SIGLAS


folioalumno
periodo
fechasolicitud
fechaentrevista
fechaexamen
Examenadmision
horaentrevista
horaexamen
1
pago
salon
cpe
entrevistador
calificacion


password
username
tipo
1
Users
email
materno
paterno
nombre


user
msg
timestamp
1
Log
ip
class
priority


FOLIO
IMSS
GRUPOSANGUINEO
ANTEOJOS
ENFERMEDADES
Registromedico
CAUSAANTEOJOS
ALERGIAS
1
INTERVENCIONES
TIPOALERGIA
TIPOINTERVENCIONES
ALERGIA
ALCOHOL
TIPOALERGIAMEDICAMENTO
TABACO
MEDICAMENTO
DROGAS
TIPOMEDICAMENTO
TATUAJE
TELEFONOEMERGENCIA
ANTECEDENTESFAMILIARES
CONTACTOEMERGENCIA
MEDICAMENTOAUTORIZADO
TUTOR

4.4. Diseño Inicial de la Base de Datos

Como ya se ha mencionado, el diseño de la base de datos se realizará bajo los principios del modelo relacional. Analizando los requerimientos de información se identificaron las entidades, atributos y llaves iniciales para cada entidad.

Para continuar con nuestras actividades, es importante mencionar la siguiente terminología:

Formal	**Informal**
Relación	Tabla
Tupla	Renglón
Atributo	Columna, Campo
Grado	Nombre de Columna
Cardinalidad	Número de Renglones
Dominio	Fondo de valores permitidos

4.4.1. Diagrama Entidad-Relación

La convención de modelado de datos que utilizamos para realizar el modelo E-V de nuestra base de datos es CDM (Custom Development Method). En la **figura 4.4**. se mencionan los elementos principales de CDM [**ScA1996**].

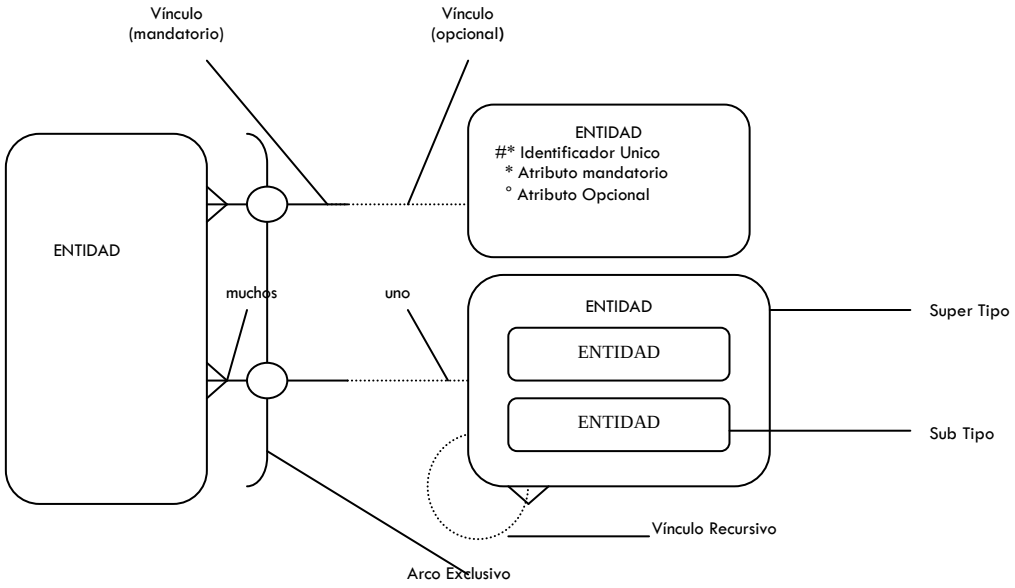


Figura 4.4 Elementos Principales de un Diagrama E-V

Analizando el diseño inicial, se realizó la división de algunos atributos, como por ejemplo: en la entidad Alumnos, el atributo dirección se descompuso en calle, colonia, número interior, número exterior, estado. De esta manera se obtiene el diseño de la base de datos.

En la siguiente **figura 4.5** se presenta el modelo entidad-relación de nuestra base de datos:

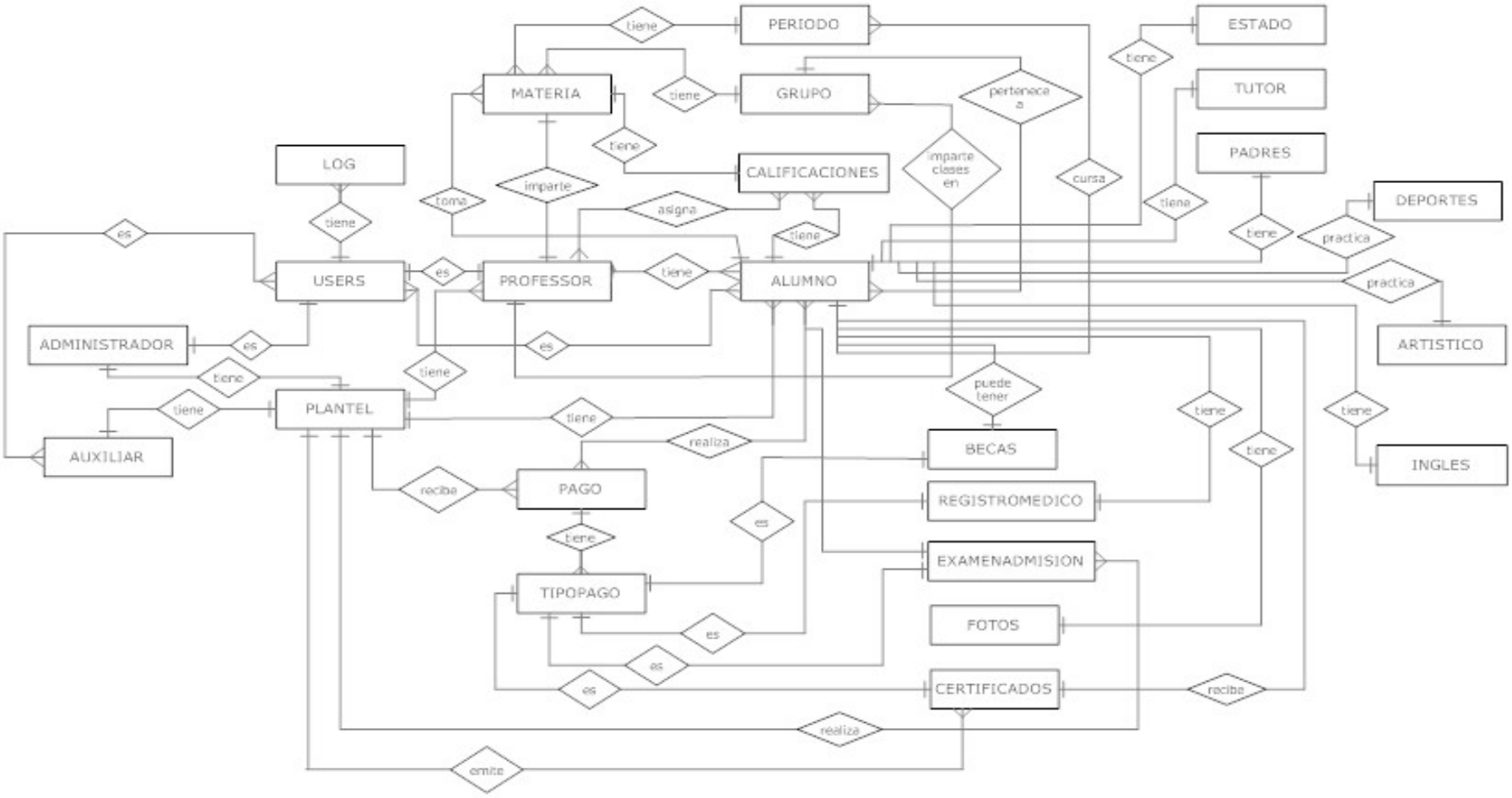


Figura 4.5 Modelo Entidad-Relación

4.4.2. Normalización

Un esquema relacional, debe cumplir con las formas normales, las cuales se consideran como propiedades; son de gran importancia en el diseño de bases de datos. Particularmente importantes son las denominadas "***Tercera Forma Normal*** y ***Forma Normal de Boyce-Codd*** ". Las formas normales garantizan que varios de los problemas de redundancia y de anomalías en una base de datos no ocurran.

El objetivo de la normalización es identificar relaciones que tengan grupos repetidos en el mismo nivel, y eliminar estos grupos repetidos durante el proceso de la normalización [Hal1994].

Primera Forma Normal (FN1)

Las relaciones están en ***primera forma normal*** si todos los dominios son simples. En un dominio simple, todos los elementos son atómicos (el objetivo es eliminar grupos repetidos). Las relaciones con dominios no simples no están en primera forma normal; a estas relaciones se les llama, ***relaciones no normalizadas***.

Segunda Forma Normal (FN2)

Una relación *R* está en *segunda forma normal* si cada atributo no primo de *R* es por completo funcionalmente dependiente con cada candidato a llave de *R*.

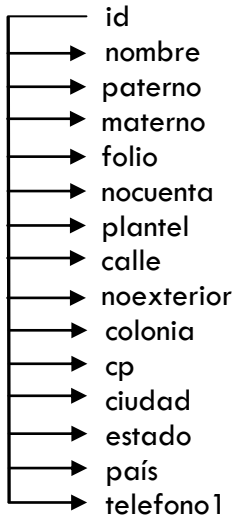
Tercera Forma Normal (FN3)

Una relación *R* está en ***tercera forma normal*** si:

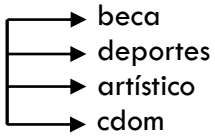
1. La relación *R* está en la segunda forma normal.
2. Los atributos no primos son independientes en forma mutua; esto es, si ningún atributo no primo es funcionalmente dependiente de otro atributo no primo (no hay dependencias transitorias).

De esta manera al aplicar la teoría de la normalización, nos ayuda a prevenir redundancias e inconsistencias en nuestra base de datos. El proceso de normalización es el siguiente:

Alumno

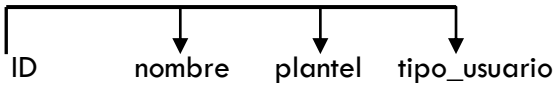


- telefono2
- celular
- otro
- email
- fechanacimiento
- ciudadnacimiento
- fotos
- estadonacimiento
- paisnacimiento
- sexo
- estadocivil
- religión
- curp
- mediotransporte
- trabajo
- giroempresa
- escuelaprocedencia
- ciudadprocedencia
- estadoescuela
- tipoescuela
- promediosecundaria
- grupo
- ingles
- estadoalumno
- admitido
- preforma
- rm
- fotos
- cc
- rd
- ano
- anc
- curp2
- cp1o
- cp1c
- cso
- csc
- cpo
- cpc
- cbc
- ces
- calif
- tea
- rpeed
- unopi
- dospi
- unopau
- dospau
- pcp
- sb



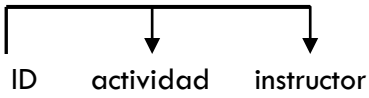
- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Administrador



- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Artístico



- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Auxiliar

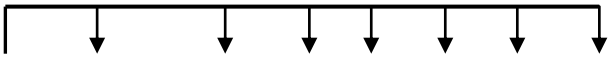
- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Becas

ID nocuenta porcentaje ciclo

- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Calificaciones



ID alumno materia plantel grupo anio mes calificación

- Inicialmente se acepta que se encuentra en FN1.

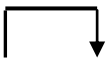
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Certificados

ID foliocerti folioalumno fechaimpresion situación periodo

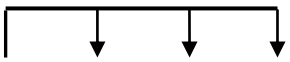
- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Estado Fotos Grupo



ID estado

ID foto

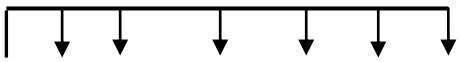


ID nombre grado plantel

- Inicialmente se acepta que se encuentra en FN1.

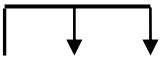
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Log



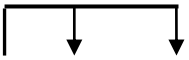
ID msg user timestamp pi priority class

ingles



ID nivel siglas

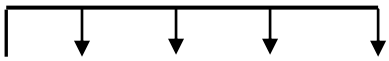
Periodo



ID semester anio

- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Materia



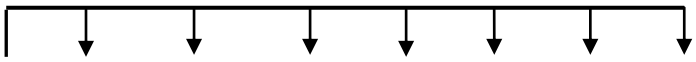
ID nombre grupo plantel profesor

Pago

ID folio fecha tipo cantidad descuento total folio recibo

- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Plantel



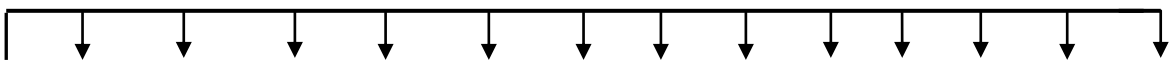
ID nombre dirección colonia ciudad estado teléfono email

Tipopago

ID Tipo

- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Profesor

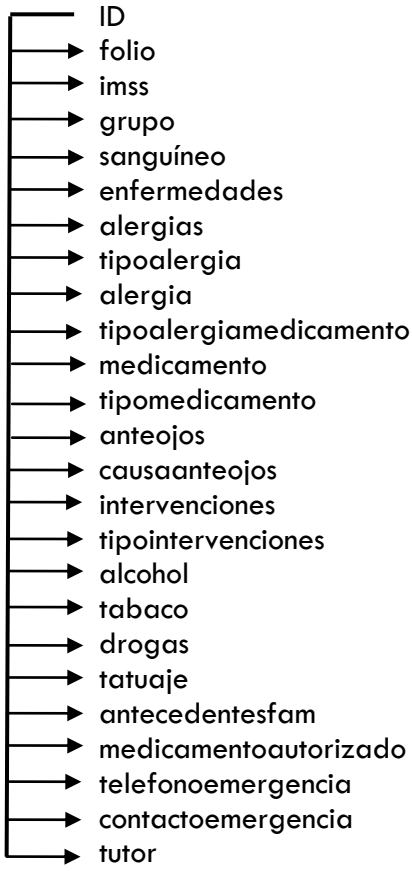


ID nombre paterno materno plantel direccion teldom telcel ciudad estado país fechanac email Tipousuario

- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.

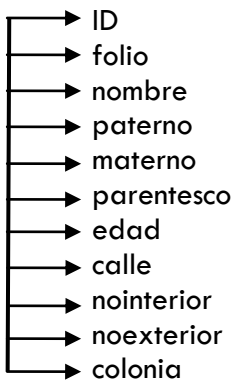
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

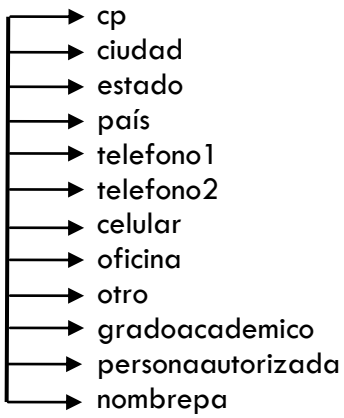
Registromedico



- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

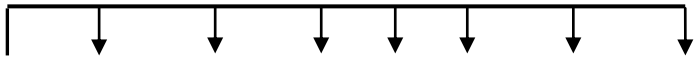
Tutor





- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

Users



ID username password email tipo nombre paterno materno

- Inicialmente se acepta que se encuentra en FN1.
- Todos los campos no clave son dependientes de la clave principal, por lo que esta entidad se encuentra en FN2.
- Ningún atributo no clave es funcionalmente dependiente de algún otro atributo no clave FN3.

4.4.3. Diseño Físico de la Base de Datos

En el Modelo Relacional todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura distinta.

La base de datos de este proyecto se diseñó en base al modelo relacional, partiendo de la identificación de entidades, atributos y relación que llevaron un modelo Entidad-Relación posteriormente se convierten en tablas según el Modelo Relacional y están relacionadas entre sí por medio de llaves foráneas.

A continuación en la **figura 4.6** se presentan las tablas que forman la base de datos, este esquema ya incluye las relaciones existentes entre ellas con la inclusión de las llaves foráneas que hacen alusión a la correspondiente tabla relacionada.

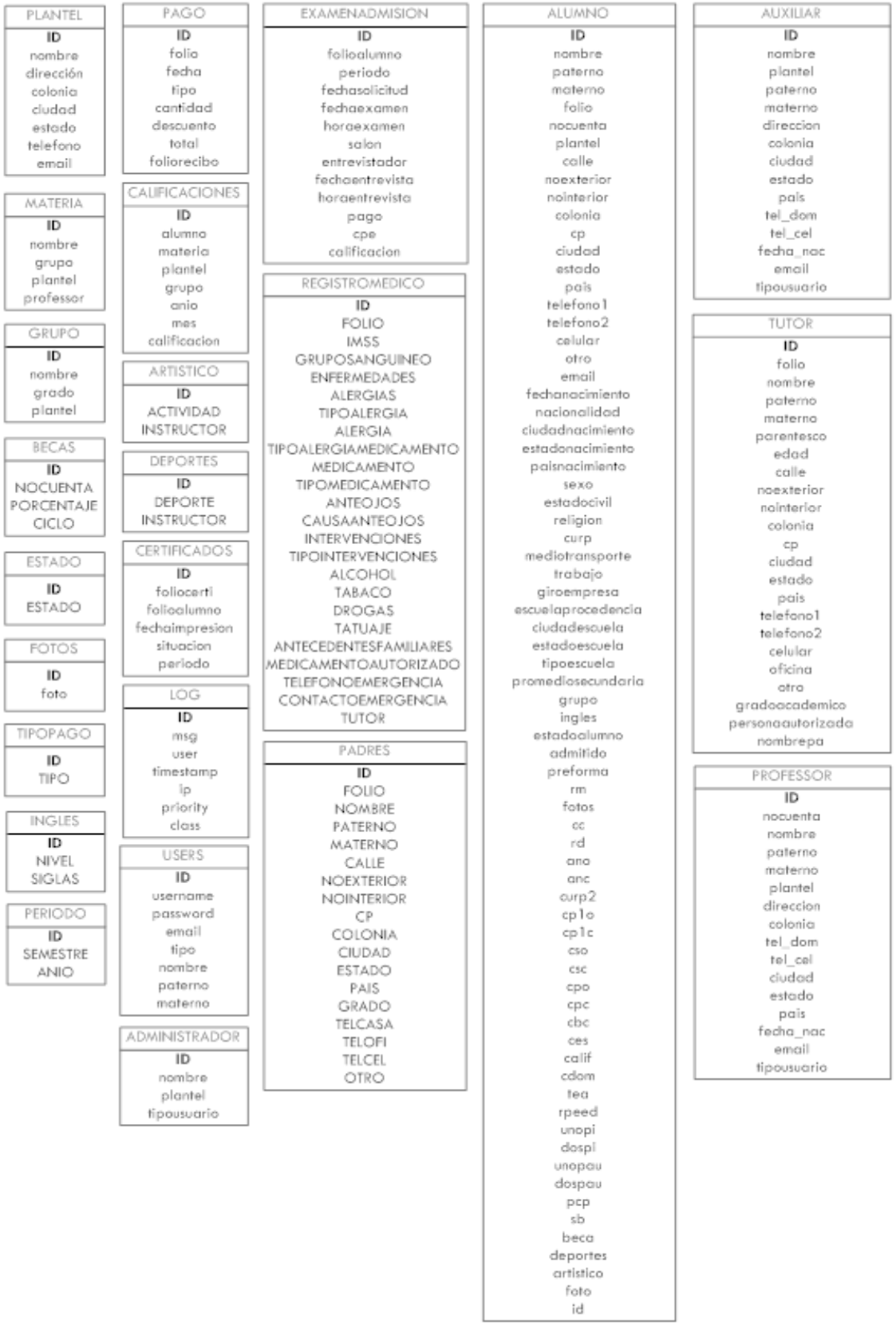


Figura 4.6 Tablas de la Base de Datos

CAPITULO 5. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

Después de haber realizado los procesos de análisis, diseño del sistema, realizando el desglose de los diferentes casos de uso, el flujo básico de cada uno de ellos y finalmente con el diseño de la base de datos, en la siguiente fase se procede a la implementación del sistema.

Para la fase de desarrollo del proyecto, se utilizará: un lenguaje de programación, un administrador de base de datos y herramientas para el diseño de las pantallas que formarán el sistema de Administración Escolar.

5.1 Herramientas para el Desarrollo del Sistema

La implementación de la base de datos se realizará con el DBMS (Data Base Management System) MySQL versión 4.1.22 standard; dado que el sistema se ejecutará en Web, se utilizará tecnología Web como: PHP Versión 5.2.5, sobre una plataforma Apache web server: Versión 1.3.41 (Linux), HTML, Javascript y CSS para el desarrollo de interfaz de usuario. Todas estas tecnologías están descritas en el capítulo 1 de este documento.

Cabe hacer mención que se tomaron en cuenta y aplicaron las recomendaciones de seguridad para MySQL y PHP mencionadas en el Capítulo 1. Para MySQL (ver **figura 5.1**) se crearon diferentes tipos de usuarios relacionados a la base de datos, según la función que desempeñan (administrador, auxiliar, profesor o alumno) editando los privilegios MySQL para los mismos de la siguiente forma:

- Administrador: permitidos todos los privilegios.
- Auxiliar: bloqueados todos los privilegios de estructura y administración, activados todos los privilegios de Datos.
- Profesor: bloqueados todos los privilegios de estructura y administración, activados todos los privilegios de Datos.
- Alumno: bloqueados todos los privilegios de estructura, administración. En datos se bloquean las opciones de INSERT, UPDATE y DELETE.

Figura 5.1. Configuración de privilegios con phpMyAdmin

5.1.1 Implementación de la Base de Datos en MySQL

Dentro de MySQL se realizó la implementación de la base datos, que en lenguaje SQL es la siguiente:

Se crea la base de datos:

```
CREATE DATABASE uia
```

Se crea la tabla alumno:

```
CREATE TABLE `alumno` (
 `nombre` varchar(255) NOT NULL DEFAULT 'NOMBRE',
 `paterno` varchar(100) NOT NULL DEFAULT 'PATERNO',
 `materno` varchar(100) NOT NULL DEFAULT 'MATERNO',
 `folio` varchar(15) NOT NULL DEFAULT 'FOLIO',
 `nocuenta` varchar(15) NOT NULL DEFAULT 'NOCUENTA',
 `plantel` varchar(100) NOT NULL DEFAULT 'PLANTEL',
 `calle` varchar(255) NOT NULL DEFAULT 'CALLE',
 `noexterior` varchar(5) NOT NULL DEFAULT '0',
 `nointerior` varchar(5) NOT NULL DEFAULT '0',
 `colonia` varchar(100) NOT NULL DEFAULT 'COLONIA',
 `cp` varchar(5) NOT NULL DEFAULT '00000',
 `ciudad` varchar(100) NOT NULL DEFAULT 'CIUDAD',
 `estado` varchar(50) NOT NULL DEFAULT 'ESTADO',
 `pais` varchar(50) NOT NULL DEFAULT 'PAIS',
 `telefono1` varchar(20) NOT NULL DEFAULT 'TELEFONO1',
 `telefono2` varchar(20) NOT NULL DEFAULT 'TELEFONO2',
 `celular` varchar(20) NOT NULL DEFAULT 'CELULAR',
 `otro` varchar(20) NOT NULL DEFAULT 'OTRO',
 `email` varchar(50) NOT NULL DEFAULT 'EMAIL',
 `fechanacimiento` date NOT NULL DEFAULT '0000-00-00',
 `nacionalidad` varchar(100) NOT NULL DEFAULT 'NACIONALIDAD',
 `ciudadnacimiento` varchar(50) NOT NULL DEFAULT 'CIUDADNACIMIENTO',
 `estadonacimiento` varchar(50) NOT NULL DEFAULT 'ESTADO NACIMIENTO',
 `paisnacimiento` varchar(50) NOT NULL DEFAULT 'PAISNACIMIENTO',
 `sexo` varchar(20) NOT NULL DEFAULT 'SEXO',
 `estadocivil` varchar(20) NOT NULL DEFAULT 'ESTADOCIVIL',
 `religion` varchar(50) NOT NULL DEFAULT 'RELIGION',
 `curp` varchar(20) NOT NULL DEFAULT 'CURP',
 `mediotransporte` varchar(50) NOT NULL DEFAULT 'MEDIOTRANSPORTE',
 `trabajo` char(2) NOT NULL DEFAULT 'NO',
 `giroempresa` varchar(255) NOT NULL DEFAULT 'GIROEMPRESA',
 `escuelaprocedencia` varchar(100) NOT NULL DEFAULT 'ESCUELAPROCEDENCIA',
 `ciudadescuela` varchar(50) NOT NULL DEFAULT 'CIUDADESCUELA',
 `estadoescuela` varchar(50) NOT NULL DEFAULT 'ESTADOESCUELA',
 `tipoescuela` varchar(50) NOT NULL DEFAULT 'TIPOESCUELA',
 `promediosecundaria` decimal(4,2) NOT NULL DEFAULT '0.00',
 `grupo` varchar(10) NOT NULL DEFAULT '0',
 `ingles` varchar(10) NOT NULL DEFAULT 'INGLES',
 `estadoalumno` varchar(100) NOT NULL DEFAULT 'ESTADO',
 `admitido` char(2) NOT NULL DEFAULT 'NO',
 `preforma` char(2) NOT NULL DEFAULT 'NO',
 `rm` char(2) NOT NULL DEFAULT 'NO',
```

```
 `fotos` char(2) NOT NULL DEFAULT 'NO',
 `cc` char(2) NOT NULL DEFAULT 'NO',
 `rd` char(2) NOT NULL DEFAULT 'NO',
 `ano` char(2) NOT NULL DEFAULT 'NO',
 `anc` char(2) NOT NULL DEFAULT 'NO',
 `curp2` char(2) NOT NULL DEFAULT 'NO',
 `cp1o` char(2) NOT NULL DEFAULT 'NO',
 `cp1c` char(2) NOT NULL DEFAULT 'NO',
 `cso` char(2) NOT NULL DEFAULT 'NO',
 `csc` char(2) NOT NULL DEFAULT 'NO',
 `cpo` char(2) NOT NULL DEFAULT 'NO',
 `cpc` char(2) NOT NULL DEFAULT 'NO',
 `cbc` char(2) NOT NULL DEFAULT 'NO',
 `ces` char(2) NOT NULL DEFAULT 'NO',
 `calif` char(2) NOT NULL DEFAULT 'NO',
 `cdom` char(2) NOT NULL DEFAULT 'NO',
 `tea` char(2) NOT NULL DEFAULT 'NO',
 `rpeed` char(2) NOT NULL DEFAULT 'NO',
 `unopi` char(2) NOT NULL DEFAULT 'NO',
 `dospi` char(2) NOT NULL DEFAULT 'NO',
 `unopau` char(2) NOT NULL DEFAULT 'NO',
 `dospau` char(2) NOT NULL DEFAULT 'NO',
 `pcp` char(2) NOT NULL DEFAULT 'NO',
 `sb` char(2) NOT NULL DEFAULT 'NO',
 `beca` int(11) NOT NULL DEFAULT '0',
 `deportes` varchar(100) NOT NULL DEFAULT 'DEPORTES',
 `artistico` varchar(100) NOT NULL DEFAULT 'ARTISTICO',
 `foto` text NOT NULL,
 `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
 KEY `id` (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=23 DEFAULT CHARSET=latin1
```

Se crea la tabla administrador:

```
CREATE TABLE `administrador` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `nombre` varchar(255) NOT NULL DEFAULT 'NOMBRE',
 `plantel` varchar(255) NOT NULL DEFAULT 'PLANTEL',
 `tipousuario` smallint(1) NOT NULL DEFAULT '0',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Se crea la tabla artístico:

```
CREATE TABLE `artistico` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `ACTIVIDAD` varchar(100) NOT NULL DEFAULT 'ACTIVIDAD',
 `INSTRUCTOR` varchar(255) NOT NULL DEFAULT 'INSTRUCTOR',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=5 DEFAULT CHARSET=latin1
```

Se crea la tabla auxiliar:

```
CREATE TABLE `auxiliar` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `nombre` varchar(255) NOT NULL DEFAULT 'NOMBRE',
 `plantel` varchar(255) NOT NULL DEFAULT 'PLANTEL',
 `paterno` varchar(100) NOT NULL DEFAULT 'PATERNO',
 `materno` varchar(100) NOT NULL DEFAULT 'MATERNO',
 `direccion` varchar(255) NOT NULL DEFAULT 'DIRECCION',
 `colonia` varchar(100) NOT NULL DEFAULT 'COLONIA',
 `ciudad` varchar(100) NOT NULL DEFAULT 'CIUDAD',
 `estado` varchar(100) NOT NULL DEFAULT 'ESTADO',
 `pais` varchar(100) NOT NULL DEFAULT 'PAIS',
 `tel_dom` varchar(50) NOT NULL DEFAULT 'TEL_DOM',
 `tel_cel` varchar(50) NOT NULL DEFAULT 'TEL_CEL',
 `fecha_nac` varchar(10) NOT NULL DEFAULT 'FECHA_NAC',
 `email` varchar(100) NOT NULL DEFAULT 'EMAIL',
 `tipousuario` varchar(100) NOT NULL DEFAULT 'USUARIO',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=26 DEFAULT CHARSET=latin1
```

Se crea la tabla becas:

```
CREATE TABLE `becas` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `NOCUENTA` varchar(20) NOT NULL DEFAULT 'NOCUENTA',
 `PORCENTAJE` int(11) NOT NULL DEFAULT '0',
 `CICLO` varchar(11) NOT NULL DEFAULT '-',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1
```

Se crea la tabla calificaciones:

```
CREATE TABLE `calificaciones` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `alumno` varchar(20) NOT NULL DEFAULT 'FOLIO',
 `materia` varchar(100) NOT NULL DEFAULT 'MATERIA',
 `plantel` varchar(100) NOT NULL DEFAULT 'PLANTEL',
 `grupo` varchar(10) NOT NULL DEFAULT 'GRUPO',
 `anio` varchar(12) NOT NULL DEFAULT 'AÑO',
 `mes` varchar(15) NOT NULL DEFAULT 'MES',
 `calificacion` decimal(4,2) NOT NULL DEFAULT '0.00',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=4 DEFAULT CHARSET=latin1
```

Se crea la tabla certificados:

```
CREATE TABLE `certificados` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `foliocerti` varchar(30) NOT NULL,
 `folioalumno` int(11) NOT NULL,
 `fechaimpresion` date NOT NULL,
 `situacion` varchar(4) NOT NULL,
 `periodo` varchar(11) NOT NULL,
 PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=latin1
```

Se crea la tabla deportes:

```
CREATE TABLE `deportes` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `DEPORTE` varchar(100) NOT NULL DEFAULT 'DEPORTE',
 `INSTRUCTOR` varchar(255) NOT NULL DEFAULT 'INSTRUCTOR',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1
```

Se crea la tabla estado:

```
CREATE TABLE `estado` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `ESTADO` varchar(100) NOT NULL DEFAULT 'ESTADO',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=15 DEFAULT CHARSET=latin1
```

Se crea la tabla examenadmision:

```
CREATE TABLE `examenadmision` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `folioalumno` varchar(15) NOT NULL DEFAULT 'FOLIO',
 `periodo` varchar(20) NOT NULL DEFAULT 'PERIODO',
 `fechasolicitud` date NOT NULL DEFAULT '0000-00-00',
 `fechaexamen` date NOT NULL DEFAULT '0000-00-00',
 `horaexamen` time NOT NULL DEFAULT '00:00:00',
 `salon` varchar(5) NOT NULL DEFAULT 'SALON',
 `entrevistador` varchar(100) NOT NULL DEFAULT 'ENTREVISTADOR',
 `fechaentrevista` date NOT NULL DEFAULT '0000-00-00',
 `horaentrevista` time NOT NULL DEFAULT '00:00:00',
 `pago` decimal(10,0) NOT NULL DEFAULT '0',
 `cpe` tinyint(1) NOT NULL DEFAULT '0',
 `calificacion` decimal(4,2) NOT NULL DEFAULT '0.00',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=11 DEFAULT CHARSET=latin1
```

Se crea la tabla fotos:

```
CREATE TABLE `fotos` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `foto` text NOT NULL,
 PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
```

Se crea la tabla grupo:

```
CREATE TABLE `grupo` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `nombre` varchar(100) NOT NULL DEFAULT 'NOMBRE',
 `grado` varchar(50) NOT NULL DEFAULT 'GRADO',
 `plantel` varchar(50) NOT NULL DEFAULT 'PLANTEL',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
```

Se crea la tabla imagen:

```
CREATE TABLE `imagen` (
 `idfoto` int(4) NOT NULL AUTO_INCREMENT,
 `thumb` blob NOT NULL,
 `thumbdos` blob NOT NULL,
 `foto` blob NOT NULL,
 `mime` varchar(20) NOT NULL DEFAULT '',
 PRIMARY KEY (`idfoto`)
) ENGINE=MyISAM AUTO_INCREMENT=810 DEFAULT CHARSET=latin1
```

Se crea la tabla ingles:

```
CREATE TABLE `ingles` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `NIVEL` varchar(100) NOT NULL DEFAULT 'NIVEL',
 `SIGLAS` varchar(5) NOT NULL DEFAULT 'XX',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1
```

Se crea la tabla log:

```
CREATE TABLE `log` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `msg` varchar(255) NOT NULL DEFAULT '',
 `user` varchar(255) NOT NULL DEFAULT '',
 `timestamp` int(11) NOT NULL DEFAULT '0',
 `ip` varchar(255) NOT NULL DEFAULT '',
 `priority` varchar(255) NOT NULL DEFAULT '',
 `class` varchar(255) NOT NULL DEFAULT '',
 PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=9 DEFAULT CHARSET=latin1
```

Se crea la tabla materia:

```
CREATE TABLE `materia` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `nombre` varchar(255) NOT NULL DEFAULT 'NOMBRE',
 `grupo` varchar(5) NOT NULL DEFAULT 'GRUPO',
 `plantel` varchar(50) NOT NULL DEFAULT 'PLANTEL',
 `profesor` varchar(100) NOT NULL DEFAULT 'PROFESOR',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=latin1
```

Se crea la tabla padres:

```
CREATE TABLE `padres` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `FOLIO` varchar(20) NOT NULL DEFAULT 'FOLIO',
 `NOMBRE` varchar(100) NOT NULL DEFAULT 'NOMBRE',
 `PATERNO` varchar(100) NOT NULL DEFAULT 'PATERNO',
 `MATERNO` varchar(100) NOT NULL DEFAULT 'MATERNO',
 `CALLE` varchar(100) NOT NULL DEFAULT 'CALLE',
```

```
 `NOEXTERIOR` varchar(10) NOT NULL DEFAULT 'NOEXTERIOR',
 `NOINTERIOR` varchar(10) DEFAULT 'NOINTERIOR',
 `CP` varchar(10) NOT NULL DEFAULT 'CP',
 `COLONIA` varchar(100) NOT NULL DEFAULT 'COLONIA',
 `CIUDAD` varchar(100) NOT NULL DEFAULT 'CIUDAD',
 `ESTADO` varchar(100) NOT NULL DEFAULT 'ESTADO',
 `PAIS` varchar(100) NOT NULL DEFAULT 'PAIS',
 `GRADO` varchar(50) NOT NULL DEFAULT 'GRADO',
 `TELCASA` varchar(20) NOT NULL DEFAULT 'TELCASA',
 `TELOFI` varchar(20) DEFAULT 'TELOFI',
 `TELCEL` varchar(20) DEFAULT 'TELCEL',
 `OTRO` varchar(20) DEFAULT 'OTRO',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1
```

Se crea la tabla pago:

```
CREATE TABLE `pago` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `folio` varchar(20) NOT NULL DEFAULT 'FOLIO',
 `fecha` date NOT NULL DEFAULT '0000-00-00',
 `tipo` varchar(50) NOT NULL DEFAULT 'TIPO',
 `cantidad` decimal(7,0) NOT NULL DEFAULT '0',
 `descuento` decimal(7,0) NOT NULL DEFAULT '0',
 `total` decimal(7,0) NOT NULL DEFAULT '0',
 `foliorecibo` varchar(10) NOT NULL DEFAULT 'XXXXXXXXXX',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=12 DEFAULT CHARSET=latin1
```

Se crea la tabla periodo:

```
CREATE TABLE `periodo` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `SEMESTRE` char(1) NOT NULL DEFAULT '-',
 `ANIO` varchar(9) NOT NULL DEFAULT '-',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=latin1
```

Se crea la tabla plantel:

```
CREATE TABLE `plantel` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `nombre` varchar(255) NOT NULL DEFAULT 'NOMBRE',
 `direccion` varchar(255) NOT NULL DEFAULT 'DIRECCION',
 `colonia` varchar(100) NOT NULL DEFAULT 'COLONIA',
 `ciudad` varchar(50) NOT NULL DEFAULT 'CIUDAD',
 `estado` varchar(50) NOT NULL DEFAULT 'ESTADO',
 `telefono` varchar(50) NOT NULL DEFAULT 'TELEFONO',
 `email` varchar(50) NOT NULL DEFAULT 'EMAIL',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=7 DEFAULT CHARSET=latin1
```

Se crea la tabla profesor:

```
CREATE TABLE `profesor` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `nocuenta` varchar(20) NOT NULL DEFAULT 'NOCUENTA',
 `nombre` varchar(255) NOT NULL DEFAULT 'NOMBRE',
 `paterno` varchar(100) NOT NULL DEFAULT 'PATERNO',
 `materno` varchar(100) NOT NULL DEFAULT 'MATERNO',
 `plantel` varchar(255) NOT NULL DEFAULT 'PLANTEL',
 `direccion` varchar(255) NOT NULL DEFAULT 'DIRECCION',
 `colonia` varchar(255) NOT NULL DEFAULT 'COLONIA',
 `tel_dom` varchar(20) NOT NULL DEFAULT 'TEL_DOM',
 `tel_cel` varchar(20) NOT NULL DEFAULT 'TEL_CEL',
 `ciudad` varchar(50) NOT NULL DEFAULT 'CIUDAD',
 `estado` varchar(50) NOT NULL DEFAULT 'ESTADO',
 `pais` varchar(100) NOT NULL DEFAULT 'PAIS',
 `fecha_nac` date NOT NULL DEFAULT '0000-00-00',
 `email` varchar(50) NOT NULL DEFAULT 'EMAIL',
 `tipousuario` varchar(20) NOT NULL DEFAULT 'TIPOUSUARIO',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1
```

Se crea la tabla registromedico:

```
CREATE TABLE `registromedico` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `FOLIO` varchar(10) NOT NULL DEFAULT 'FOLIO',
 `IMSS` varchar(20) NOT NULL DEFAULT 'IMSS/ISSSTE',
 `GRUPOSANGUINEO` varchar(5) NOT NULL DEFAULT 'XXXXX',
 `ENFERMEDADES` varchar(255) NOT NULL DEFAULT 'ENFERMEDADES',
 `ALERGIAS` char(2) NOT NULL DEFAULT 'NO',
 `TIPOALERGIA` varchar(255) NOT NULL DEFAULT 'TIPOALERGIA',
 `ALERGIA` char(2) NOT NULL DEFAULT 'NO',
 `TIPOALERGIAMEDICAMENTO` varchar(255) NOT NULL DEFAULT 'TIPOALERGIAMEDICAMENTO',
 `MEDICAMENTO` char(2) NOT NULL DEFAULT 'NO',
 `TIPOMEDICAMENTO` varchar(255) NOT NULL DEFAULT 'TIPOMEDICAMENTO',
 `ANTEOJOS` char(2) NOT NULL DEFAULT 'NO',
 `CAUSAANTEOJOS` varchar(255) NOT NULL DEFAULT 'CAUSAANTEOJOS',
 `INTERVENCIONES` char(2) NOT NULL DEFAULT 'NO',
 `TIPOINTERVENCIONES` varchar(255) NOT NULL DEFAULT 'TIPOINTERVENCIONES',
 `ALCOHOL` char(2) NOT NULL DEFAULT 'NO',
 `TABACO` char(2) NOT NULL DEFAULT 'NO',
 `DROGAS` char(2) NOT NULL DEFAULT 'NO',
 `TATUAJE` char(2) NOT NULL DEFAULT 'NO',
 `ANTECEDENTESFAMILIARES` varchar(255) NOT NULL DEFAULT 'ANTECEDENTES',
 `MEDICAMENTOAUTORIZADO` varchar(255) NOT NULL DEFAULT 'MEDICAMENTOAUTORIZADO',
 `TELEFONOEMERGENCIA` varchar(20) NOT NULL DEFAULT 'TELEFONOEMERGENCIA',
 `CONTACTOEMERGENCIA` varchar(100) NOT NULL DEFAULT 'CONTACTOEMERGENCIA',
 `TUTOR` varchar(255) NOT NULL DEFAULT '0',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=8 DEFAULT CHARSET=latin1
```

Se crea la tabla tipopago:

```
CREATE TABLE `tipopago` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
```

```
 `TIPO` varchar(100) NOT NULL DEFAULT 'TIPO',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=6 DEFAULT CHARSET=latin1
```

Se crea la tabla tutor:

```
CREATE TABLE `tutor` (
 `ID` int(11) NOT NULL AUTO_INCREMENT,
 `folio` varchar(20) NOT NULL DEFAULT 'FOLIO',
 `nombre` varchar(255) NOT NULL DEFAULT 'NOMBRE',
 `paterno` varchar(100) NOT NULL DEFAULT 'PATERNO',
 `materno` varchar(100) NOT NULL DEFAULT 'MATERNO',
 `parentesco` varchar(50) NOT NULL DEFAULT 'PARENTEZCO',
 `edad` int(2) NOT NULL DEFAULT '0',
 `calle` varchar(255) NOT NULL DEFAULT 'CALLE',
 `noexterior` int(11) NOT NULL DEFAULT '0',
 `nointerior` int(11) DEFAULT '0',
 `colonia` varchar(100) NOT NULL DEFAULT 'COLONIA',
 `cp` varchar(5) NOT NULL DEFAULT '00000',
 `ciudad` varchar(50) NOT NULL DEFAULT 'CIUDAD',
 `estado` varchar(50) NOT NULL DEFAULT 'ESTADO',
 `pais` varchar(50) NOT NULL DEFAULT 'PAIS',
 `telefono1` varchar(20) NOT NULL DEFAULT 'TELEFONO1',
 `telefono2` varchar(20) DEFAULT 'TELEFONO2',
 `celular` varchar(20) DEFAULT 'CELULAR',
 `oficina` varchar(20) DEFAULT 'OFICINA',
 `otro` varchar(20) DEFAULT 'OTRO',
 `gradoacademico` varchar(50) NOT NULL DEFAULT 'GRADO',
 `personaautorizada` char(2) NOT NULL DEFAULT 'NO',
 `nombrepa` varchar(100) DEFAULT 'PERSONA AUTORIZADA',
 PRIMARY KEY (`ID`)
) ENGINE=MyISAM AUTO_INCREMENT=3 DEFAULT CHARSET=latin1
```

Se crea la tabla users:

```
CREATE TABLE `users` (
 `id` int(11) NOT NULL AUTO_INCREMENT,
 `username` varchar(30) NOT NULL DEFAULT '',
 `password` varchar(20) NOT NULL DEFAULT '',
 `email` varchar(40) NOT NULL DEFAULT '',
 `tipo` int(1) NOT NULL DEFAULT '0',
 `nombre` varchar(250) NOT NULL DEFAULT 'NOMBRE',
 `paterno` varchar(250) NOT NULL DEFAULT 'APELLIDO PATERNO',
 `materno` varchar(250) NOT NULL DEFAULT 'APELLIDO MATERNO',
 PRIMARY KEY (`id`)
) ENGINE=MyISAM AUTO_INCREMENT=34 DEFAULT CHARSET=latin1
```

5.2 Conexión MySQL con PHP

Para realizar la conexión, se realizó un script en PHP con los datos de la base de datos, y se utilizó una función propia del lenguaje para realizar la conexión con la base de datos.

Script _ *config.php* con datos de la base de datos:

```
<?php
  $dbhost='localhost';
  $dbusername= 'root';
  $dbuserpass='';
  $dbname='sandyoro_uia';
?>
```

Proceso para realizar la conexión:

```
require("_config.php");
// Conexión a la base de datos
mysql_connect ($dbhost, $dbusername, $dbuserpass);
mysql_select_db($dbname) or die("Cannot select database");
```

En este caso sólo tomaremos la tabla alumnos para exponer el código que se generó para la inserción de registros, baja de registros, modificación de registros y consulta de registros de esta tabla; ya que para el resto de las tablas de la base de datos, el código que se generó para el mantenimiento de estas es el mismo que se utiliza para la tabla alumnos.

Inserción de alumnos:

```
$sql = "INSERT INTO ALUMNO
(nocuenta,admitido,ingles,estadoalumno,grupo,rm,cc,rd,ano,anc,curp2,cp1o,cp1c,cso,csc,cpo,cpc,cbc,ces,calif,cd
om,tea,rpeed,unopi,dospi,unopau,dospau,pcp,sb,beca,deportes,artistico,foto,folio) VALUES
('$mat','$adm','$ing','$edo','$gru','$rm','$cc','$rd','$ano','$anc','$curp2','$cp1o','$cp1c','$cso','$csc','$cpo','$cpc'
,'$cbc','$ces','$calif','$cdom','$tea','$rpeed','$unopi','$dospi','$unopau','$dospau','$pcp','$sb','0','$deportes','$art
istico','$arf','$fol')";
$result = mysql_query($sql);
$my_error = mysql_error($link);
```

Baja de alumnos:

```
$cadbusca="DELETE FROM ALUMNO WHERE nocuenta='$nocuenta'";
$result=mysql_query($cadbusca);
$cadbusca2="DELETE FROM CALIFICACIONES WHERE alumno='$folio'";
$result2=mysql_query($cadbusca2);
$cadbusca3="DELETE FROM PADRES WHERE FOLIO='$folio'";
$result3=mysql_query($cadbusca3);
$cadbusca4="DELETE FROM PAGO WHERE folio='$folio'";
$result4=mysql_query($cadbusca4);
$cadbusca5="DELETE FROM REGISTROMEDICO WHERE FOLIO='$folio'";
$result5=mysql_query($cadbusca5);
$cadbusca6="DELETE FROM TUTOR WHERE folio='$folio'";
$result6=mysql_query($cadbusca6);
$cadbusca7="DELETE FROM BECAS WHERE NOCUENTA='$folio'";
$result7=mysql_query($cadbusca7);
if(! $result or ! $result2 or ! $result3 or ! $result4 or ! $result5 or ! $result6 or ! $result7) {
echo "Ha habido un error al buscar los valores. $result";
} else {
echo "Registro con el no. de cuenta: ".$folio." eliminado";
}
} else {
echo "Error, no ha introducido todos los datos <br>";
echo "<a href='javascript:history.back(1)'>Regresar</a>";
```

}

La eliminación de registros se realizará de acuerdo al filtro seleccionado por el usuario.

Modificación de Alumnos:

```
$sql = "UPDATE ALUMNO SET folio =
$fol,nocuenta='$mat',admitido='$adm',ingles='$ing',estadoalumno='$edo',grupo='$gru',rm='$rm',fotos='$fotos',cc='$cc',rd='$rd',ano='$ano',anc='$anc',curp2='$curp2',cp1o='$cp1o',cp1c='$cp1c',cso='$cso',csc='$csc',cpo='$cpo',cpc='$cpc',cbc='$cbc',ces='$ces',calif='$calif',cdom='$cdom',tea='$tea',rpeed='$rpeed',unopi='$unopi',dospi='$dospi',unopau='$unopau',dospau='$dospau',pcp='$pcp',sb='$sb',beca='0',deportes='$deportes',artistico='$artistico' WHERE nombre='$nombre' AND paterno = '$paterno' AND materno = '$materno'";
   $result = mysql_query($sql);
   $my_error = mysql_error($link);
```

Consulta de Alumnos:

```
$cadbusca=mysql_query("select * FROM ALUMNO") or die(mysql_error());
//$result=mysql_fetch_array($cadbusca);
```

5.3 Interfaces del Sistema de Administración Escolar

La interfaz del sistema inicia con una pantalla, donde el sistema pide un usuario y un password , esto es para permitir el acceso al sistema de acuerdo al perfil del usuario que se esté firmando. Esto se muestra en la **figura 5.2**:

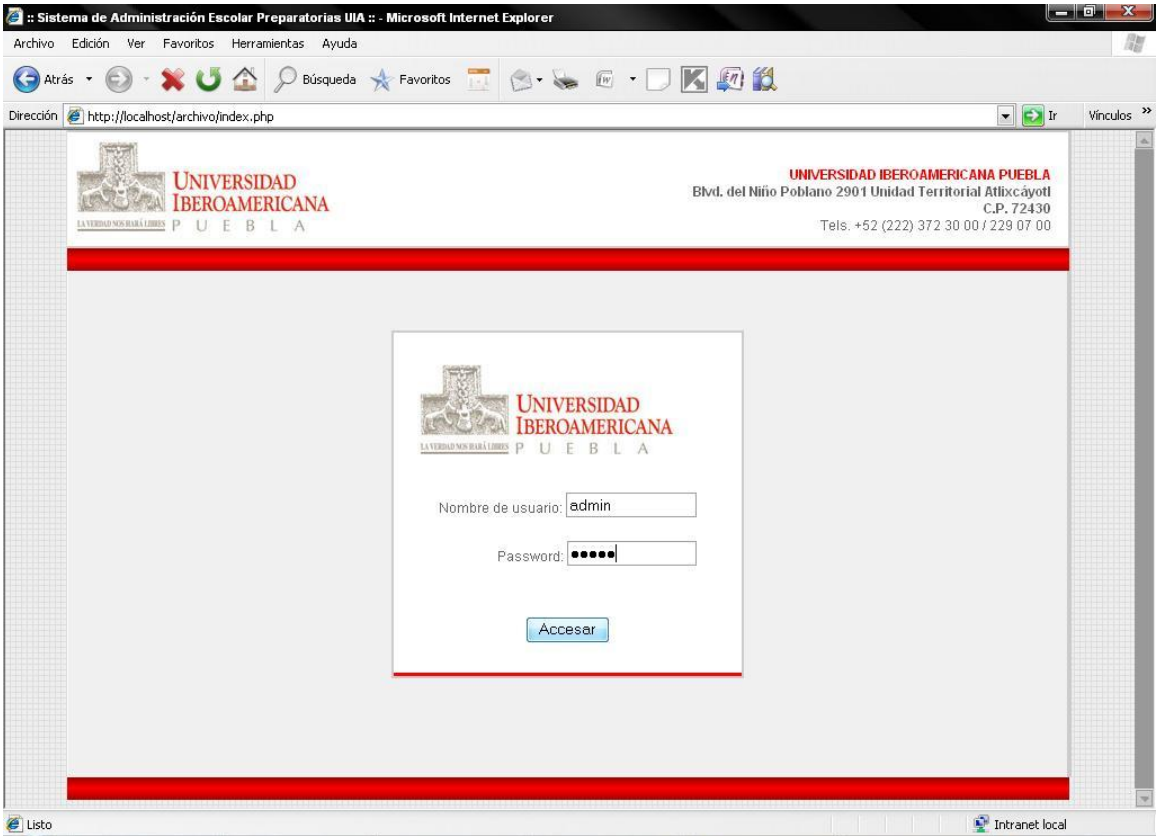


Figura 5.2 Interfaz de entrada al sistema

5.3.1 Interfaces del Usuario Administrador

La interfaz del administrador se muestra en la **figura 5.3**:

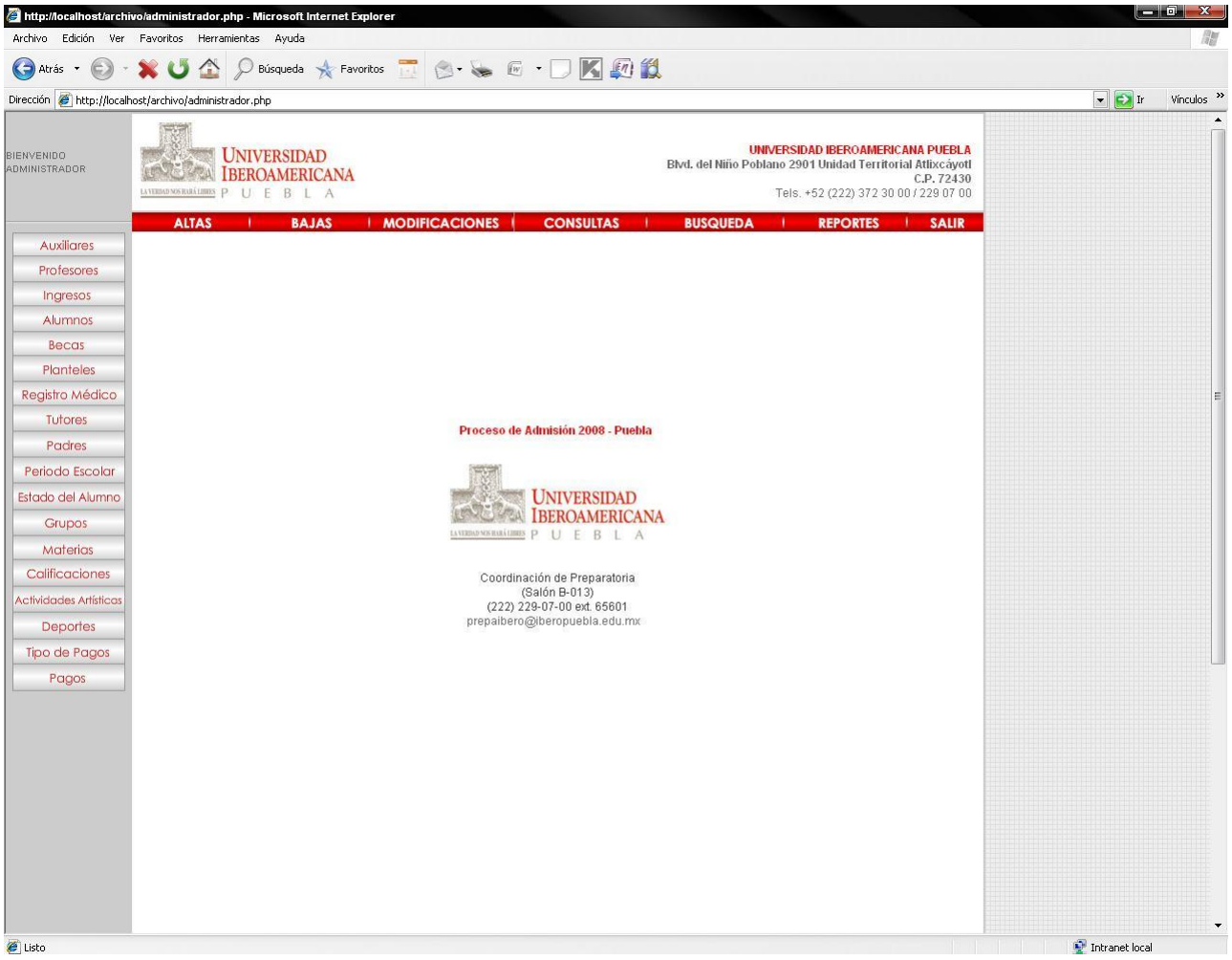


Figura 5.3 Interfaz para usuario Administrador

El usuario firmado como administrador del sistema, tiene acceso para manipular toda la información de las entidades que se indican en la parte izquierda de la pantalla, así como a todas las tareas que se indican en la parte superior (barra de menús).

Por ejemplo se desplegará la siguiente pantalla al elegir opción ALTAS y elegir botón Calificaciones en la **figura 5.4**:

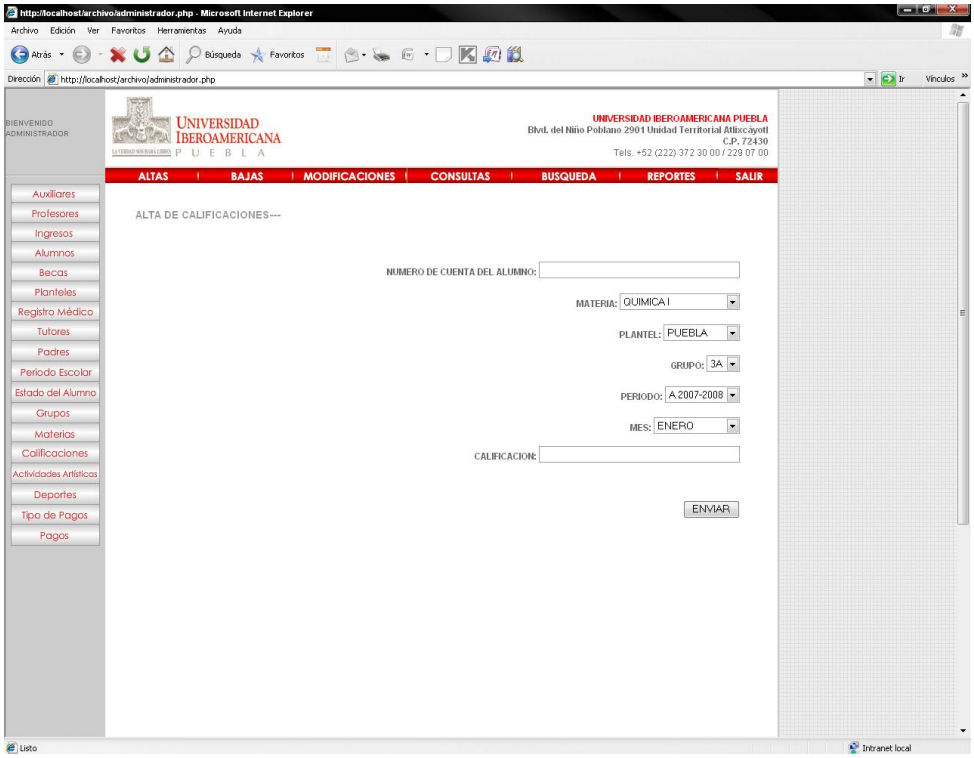


Figura 5.4 Pantalla para dar de alta calificaciones

5.3.2 Interfaces del Usuario Auxiliar

La interfaz del auxiliar se muestra en la **figura 5.5**:

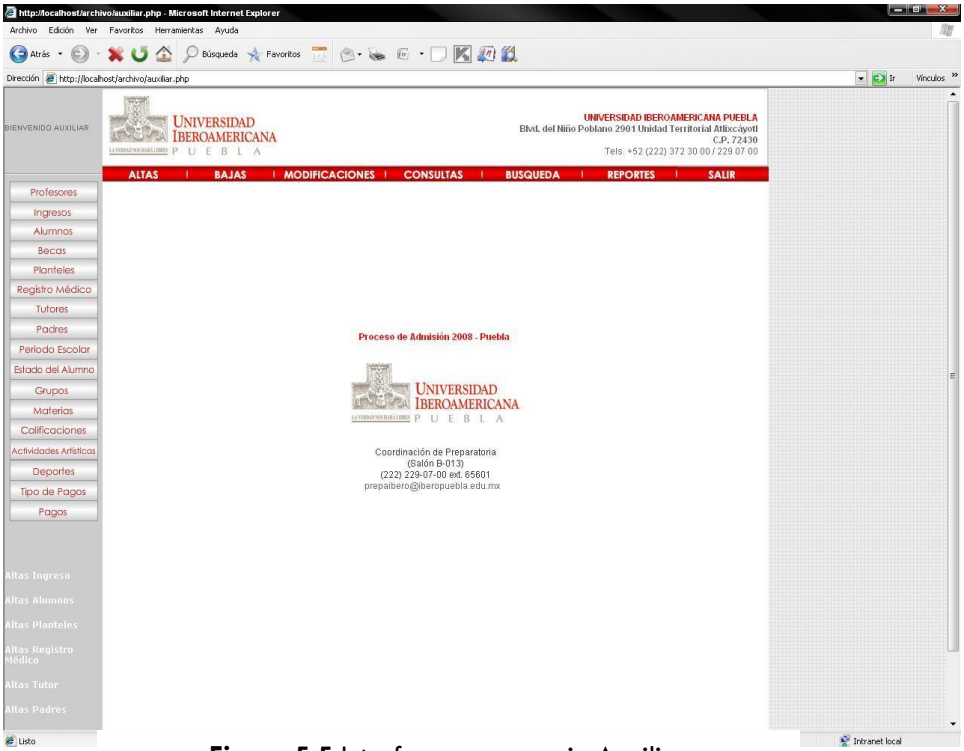


Figura 5.5 Interfaz para usuario Auxiliar

El usuario firmado como auxiliar, tiene acceso para manipular toda la información de las entidades que se indican en la parte izquierda de la pantalla, así como a todas las tareas que se indican en la parte superior (barra de menús).

Por ejemplo se desplegará la siguiente pantalla al elegir opción ALTAS y elegir opción Ingreso en la **figura 5.6**:

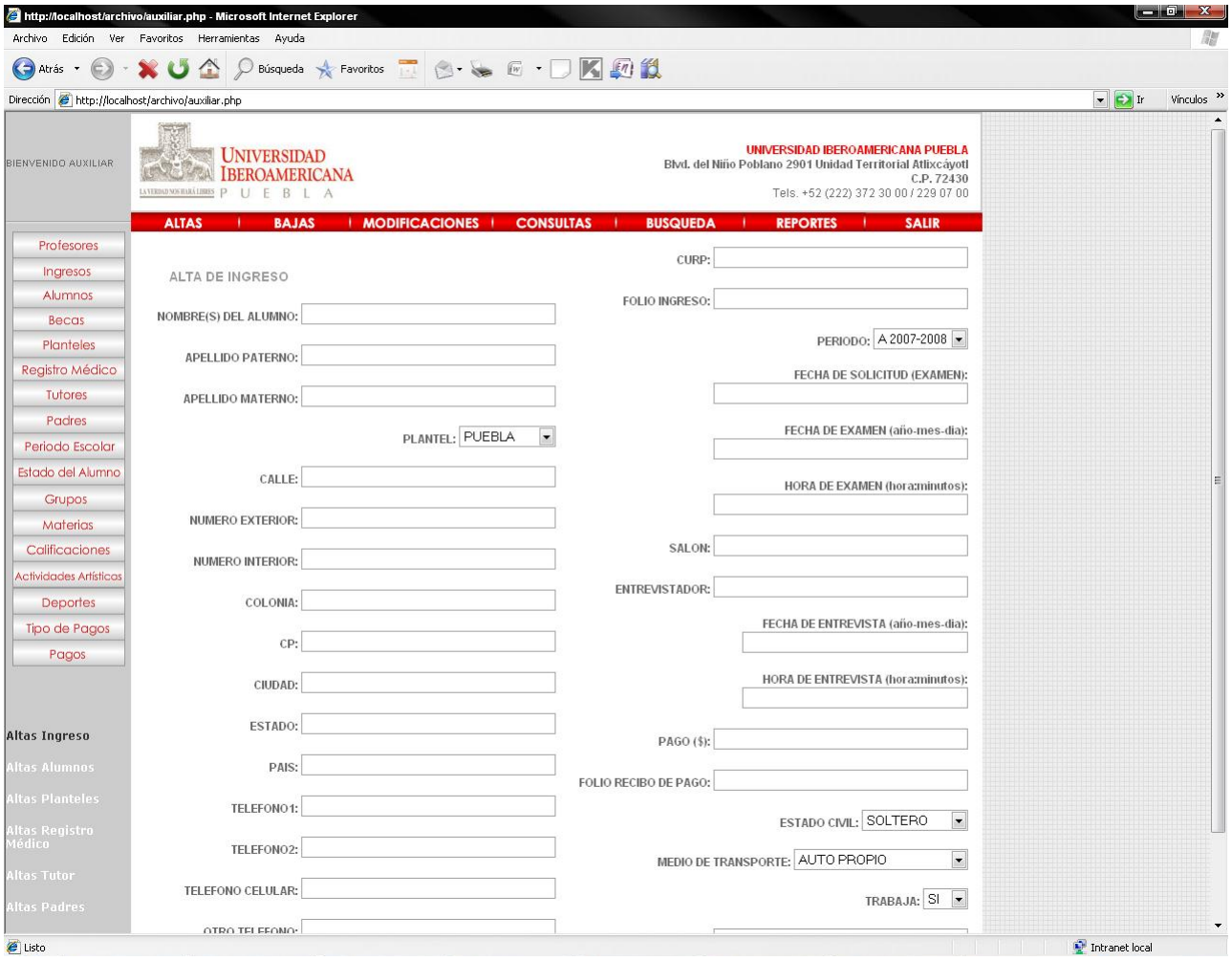


Figura 5.6 Pantalla de registro de Ingreso de alumnos

5.3.3 Interfaces del Usuario Docente

La interfaz del profesor se muestra en la **figura 5.7**:

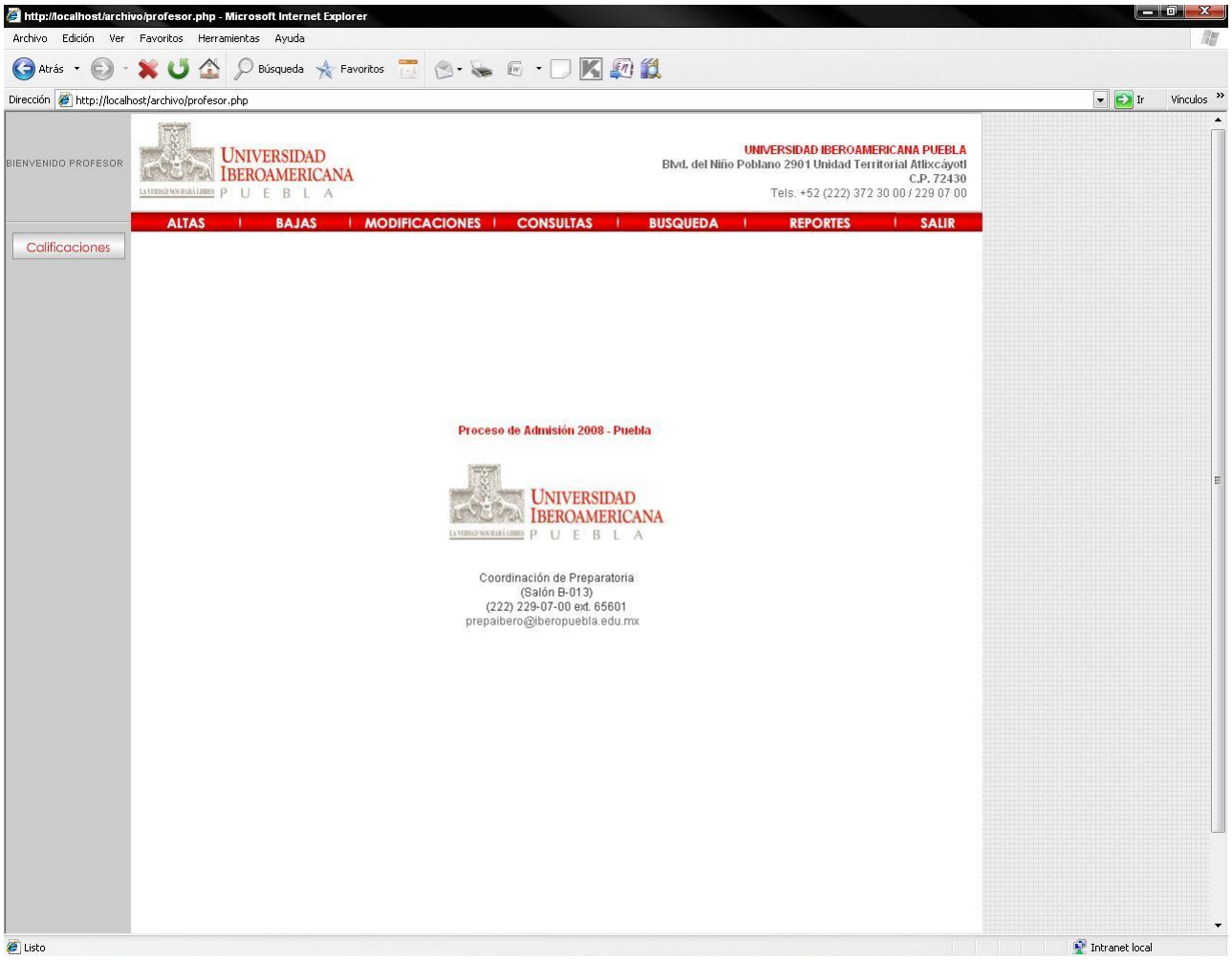


Figura 5.7 Interfaz para usuario Profesor

El usuario firmado como profesor, tiene acceso para manipular toda la información de las entidades que se indican en la parte izquierda de la pantalla, así como a todas las tareas que se indican en la parte superior (barra de menús).

Por ejemplo se desplegará la siguiente pantalla al elegir opción CONSULTA y elegir opción Calificaciones en la **pantalla 5.8**:

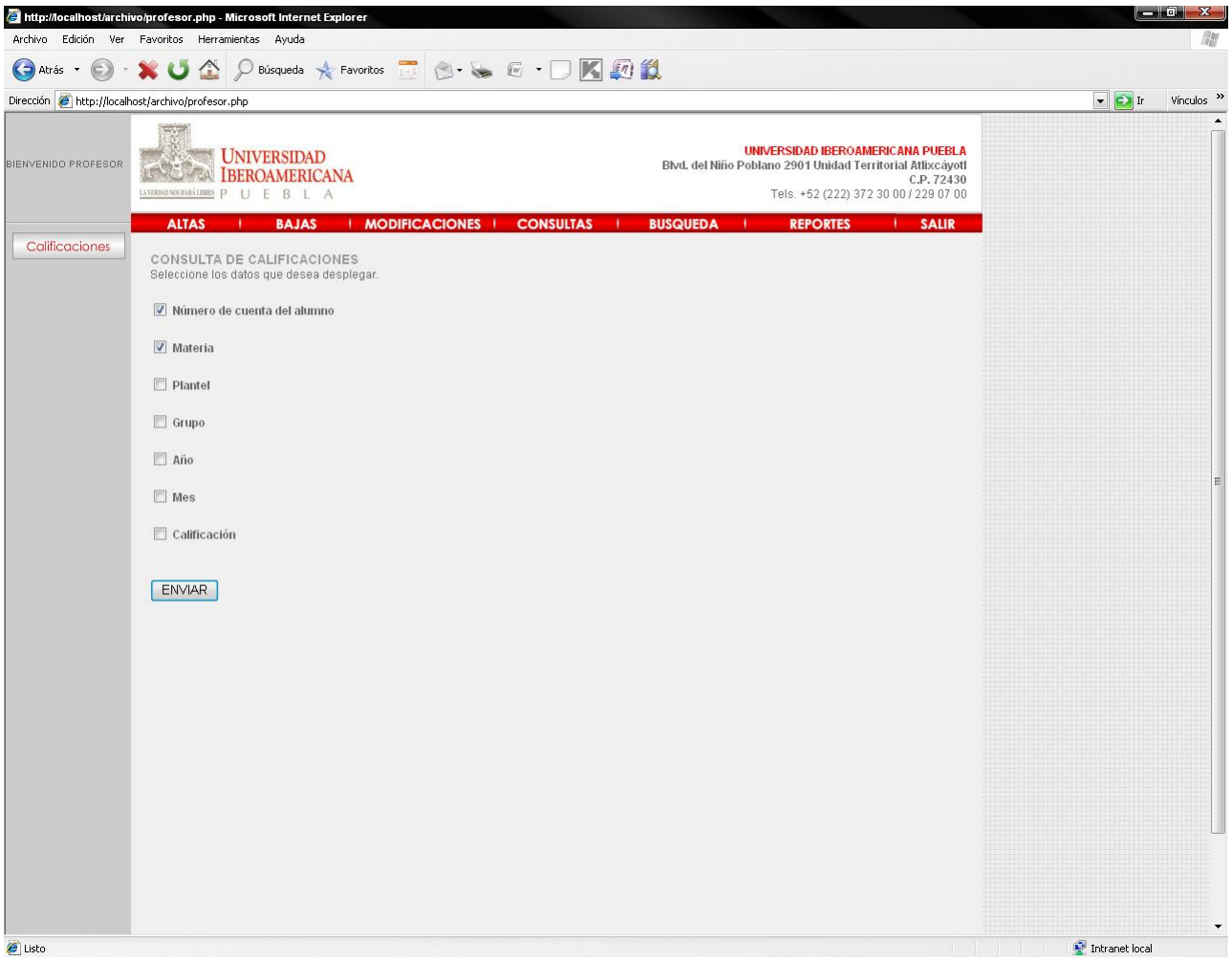


Figura 5.8 Pantalla de consulta de Calificaciones

5.3.4 Interfaces del Usuario Alumno

La interfaz del alumno se muestra en la **figura 5.9**:

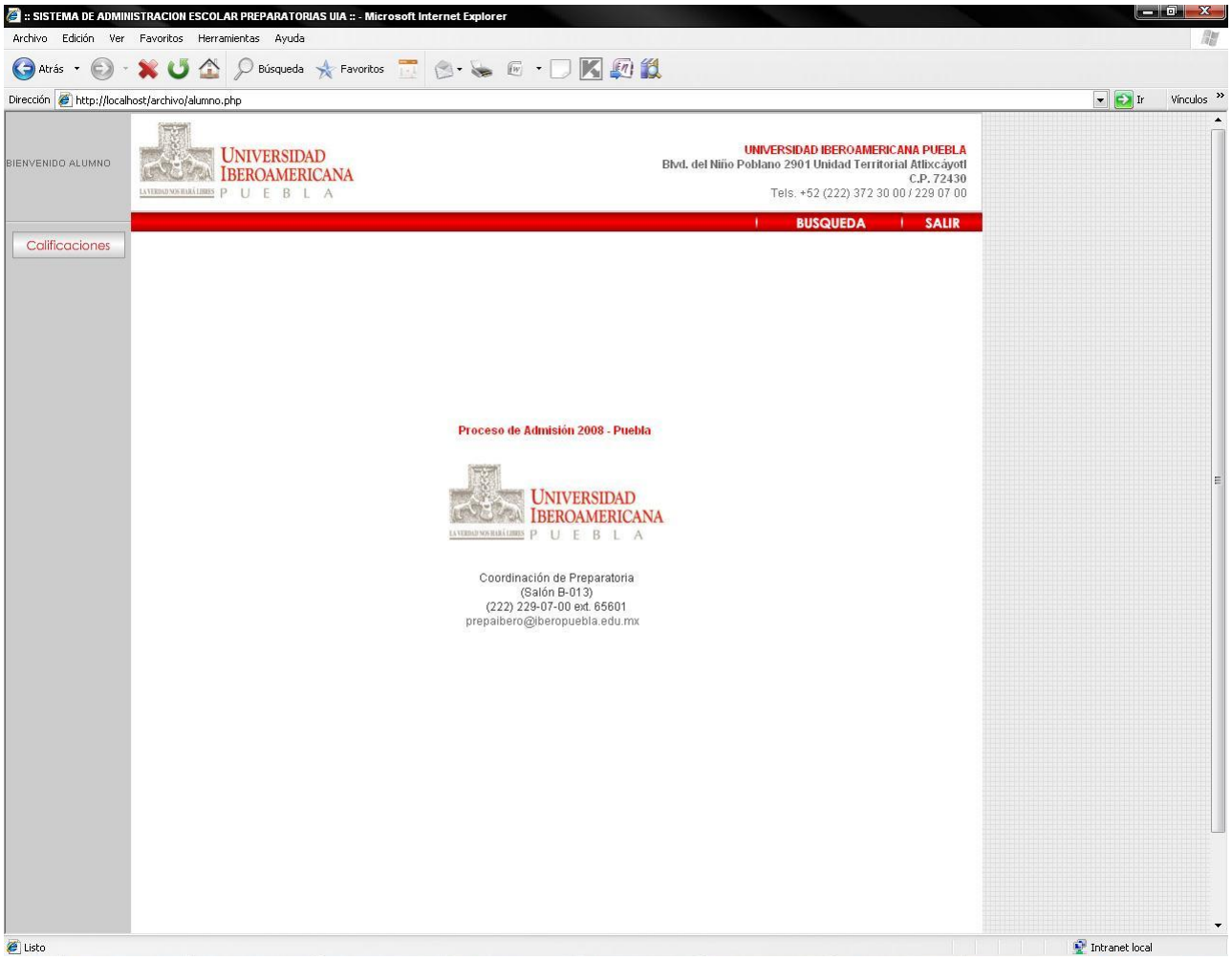


Figura 5.9 Interfaz para usuario Alumno

El usuario firmado como profesor, tiene acceso para manipular toda la información de las entidades que se indican en la parte izquierda de la pantalla, así como a todas las tareas que se indican en la parte superior (barra de menús).

Por ejemplo se desplegará la siguiente pantalla en la **figura 5.10** al elegir opción CONSULTA y elegir opción Calificaciones:

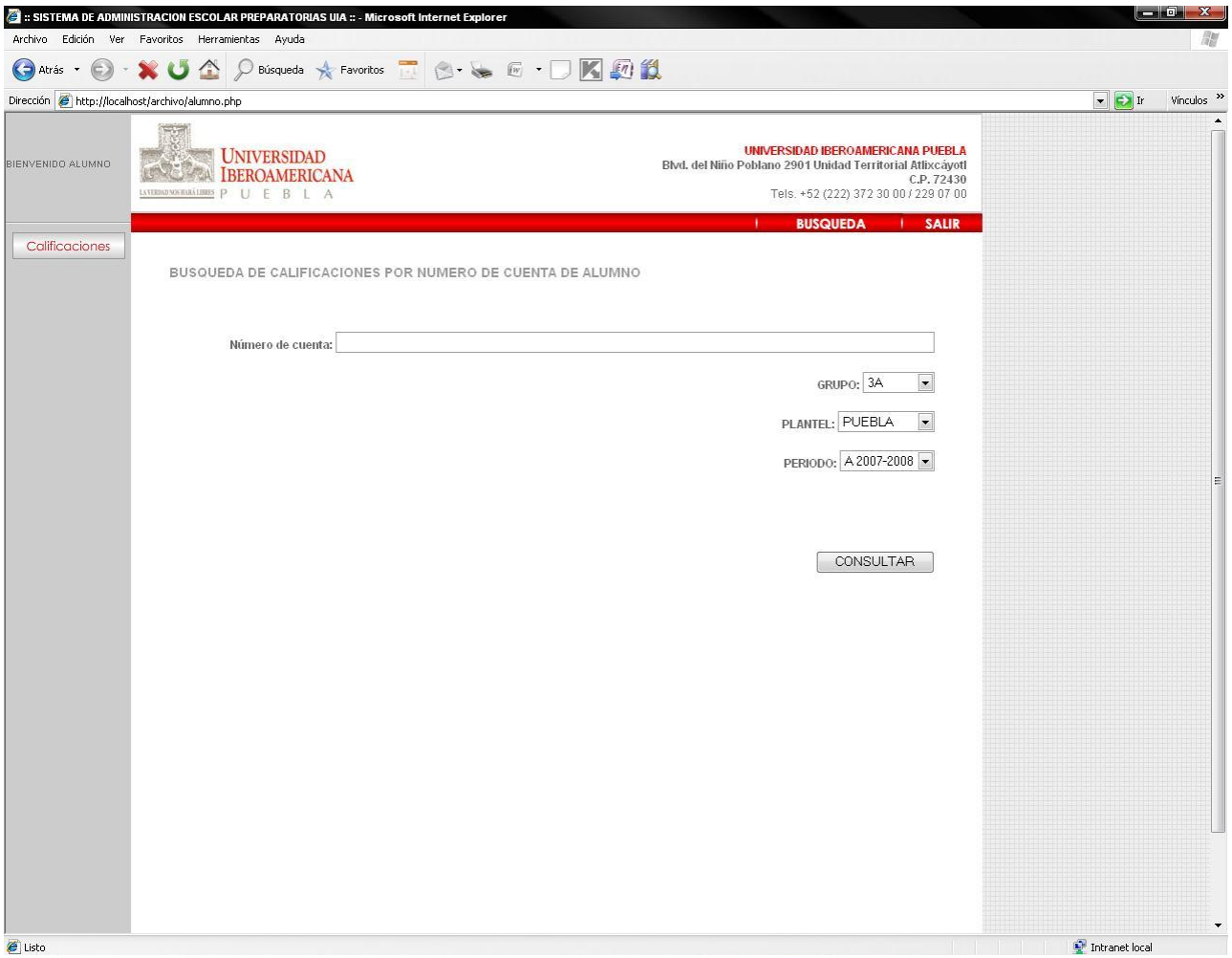


Figura 5.10 Pantalla de Búsqueda de calificaciones

5.3.5 Reportes

Se pueden obtener tres tipos de reportes, según el tipo de usuario que utilice el sistema.

Uno de ellos es obtenido al realizar una consulta mediante la selección de los datos necesarios e imprimir el resultado de la búsqueda, disponible para los usuarios: administrador y auxiliar, un ejemplo se muestra en la **figura 5.11**:

CONSULTAR ALUMNO Página 1 de 1

No. CUENTA	FOLIO	NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO	PLANTEL	GRUPO	CIUDAD	ESTADO
065456	1	SANDY	OROZCO	VELAZQUEZ	PUEBLA	3A	PUEBLA	PUEBLA
123456	3	FERNANDO	JUAREZ	HERNANDEZ	PUEBLA	3A	PUEBLA	PUEBLA
789012	4	ALEJANDRA	GUTIERREZ	MARTINEZ	PUEBLA	3A	PUEBLA	PUEBLA
432156	5	SERGIO	PINEDA	PEREZ	TLAXCALA	3A	TLAXCALA	TLAXCALA

Figura 5.11 Impresión de resultado de una consulta de Alumnos

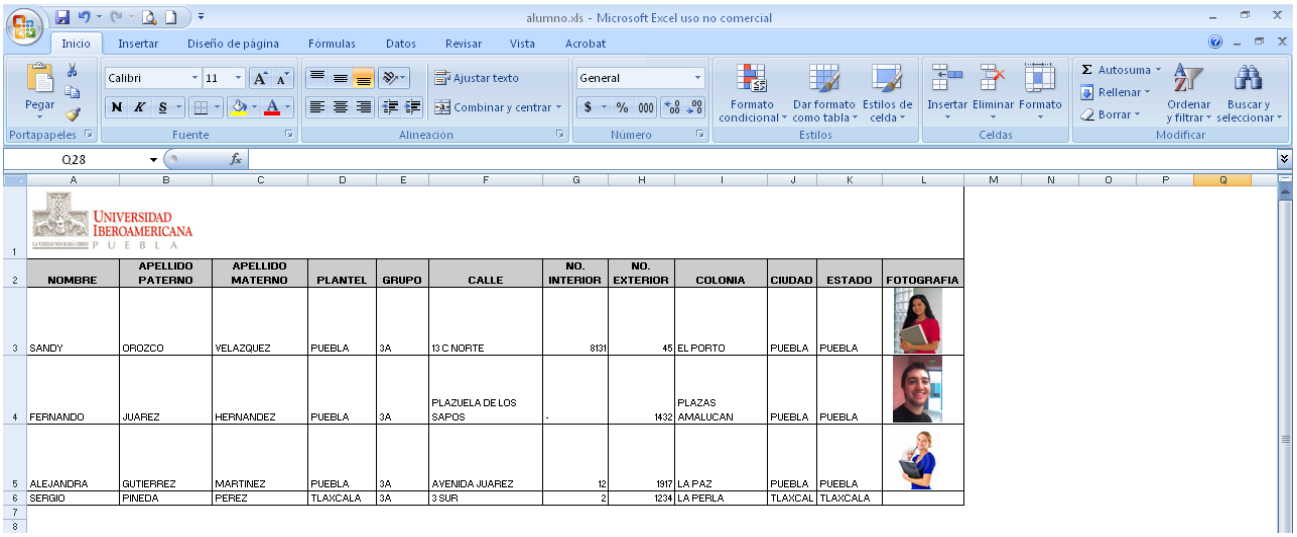
Los usuarios: profesor y alumno, únicamente pueden generar reportes de calificaciones, como se muestra en la **figura 5.12**:

CONSULTAR CALIFICACION Página 1 de 1

NUMERO DE CUENTA	MATERIA	PLANTEL	GRUPO	AÑO	MES	CALIFICACION
123456	QUIMICA I	PUEBLA	3A	A 2008-2009	MARZO	7.00
123456	HISTORIA DEL ARTE	PUEBLA	3A	A 2008-2009	MARZO	8.60
123456	QUIMICA I	PUEBLA	3A	A 2008-2009	ENERO	10.00
123456	BIOLOGIA	PUEBLA	3A	A 2008-2009	ENERO	9.00
123456	HISTORIA DEL ARTE	PUEBLA	3A	A 2008-2009	ENERO	8.00
123456	QUIMICA I	PUEBLA	3A	A 2008-2009	FEBRERO	7.00
123456	BIOLOGIA	PUEBLA	3A	A 2008-2009	FEBRERO	10.00
123456	HISTORIA DEL ARTE	PUEBLA	3A	A 2008-2009	FEBRERO	9.00

Figura 5.12 Impresión de resultado de una consulta de Calificaciones

Otro reporte que se puede generar es exportado en Excel, disponible para los usuarios: administrador y auxiliar, dicho reporte está basado en una selección previa de la información que se quiere obtener, como se muestra en la **figura 5.13**:



NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO	PLANTEL	GRUPO	CALLE	NO. INTERIOR	NO. EXTERIOR	COLONIA	CIUDAD	ESTADO	FOTOGRAFIA
SANDY	OROZCO	VELAZQUEZ	PUEBLA	3A	13 C NORTE	8131	45	EL PORTO	PUEBLA	PUEBLA	
FERNANDO	JUAREZ	HERNANDEZ	PUEBLA	3A	PLAZUELA DE LOS SAPOS	-	1432	PLAZAS AMALUCAN	PUEBLA	PUEBLA	
ALEJANDRA	GUTIERREZ	MARTINEZ	PUEBLA	3A	AVENIDA JUAREZ	12	1917	LA PAZ	PUEBLA	PUEBLA	
SERGIO	PINEDA	PEREZ	TLAXCALA	3A	3 SUR	2	1234	LA PERLA	TLAXCAL	TLAXCALA	

Figura 5.13 Reporte exportado en Excel de Alumnos

El tercer tipo de reporte, es la impresión del certificado con los datos del alumno y el plantel almacenados en la base de datos, como se muestra en la **figura 5.14**:

SEP

SISTEMA EDUCATIVO NACIONAL

DIRECCION GENERAL DEL BACHILLERATO

PREPARATORIA IBERO Puebla

LA DIRECCIÓN DE SERVICIOS ESCOLARES CERTIFICA QUE EN LA
CON CLAVE DEL CENTRO DE TRABAJO

CON CLAVE ÚNICA DE REGISTRO DE POBLACIÓN (CURP) SCOV171276 Y MATRICULA 123439
ACREDITO SATISFACTORIAMENTE EL CICLO DE BACHILLERATO GENERAL AL HABER CURSADO
ASIGNATURAS DE UN TOTAL DE 2 QUE INTEGRA EL PLAN DE ESTUDIOS SEGÚN CONSTA
QUE OBRAN EN EL ÁREA DE LA DIRECCIÓN DE SERVICIOS ESCOLARES DE ESTA INSTITUCIÓN

ASIGNATURAS	CALIF. FINAL	ASIGNATURAS
QUIMICA I	7.00	
HISTORIA DEL ARTE	8.60	

Promedio General de Aprovechamiento

7.8

Folio

IBERO-21

A 29 DEL PRESENTE SE EXPIDE EN PUEBLA, PUEBLA
DEL MES DE JUNIO DE DOS MIL NUEVE

ESTUDIOS RECONOCIDOS POR LA SECRETARÍA DE EDUCACIÓN PÚBLICA SEGÚN DECRETO PRESIDENCIAL DE FECHA
ACUERDO DE R.V.O.E. No.
ESTE DOCUMENTO ES VALIDO EN LOS ESTADOS UNIDOS MEXICANOS Y NO REQUIERE TRAMITES ADICIONALES DE LEGALIZACIÓ

Figura 5.14 Impresión de un certificado de alumno

CONCLUSIONES

Una vez implementado el sistema, se probó y cumplió con los requisitos planteados al inicio del proyecto, así como algunos otros requerimientos que se fueron dando durante el proceso de desarrollo.

Se realizó un sistema de Bases de Datos basado en Web, que cumple las funciones de la administración escolar para las preparatorias de la Universidad Iberoamericana, desempeñando satisfactoriamente las siguientes funciones:

- Seguridad de acceso, con 4 tipos de usuario. Los cuales deben contar con datos válidos de acceso para su manejo.
- Control escolar que por el momento gestiona a 150 alumnos y 50 profesores (plantel actual).
- Control de alumnos inscritos y de nuevo ingreso con formato de inscripción, ficha médica y registro de documentos.
- Control de profesores para registrar datos personales y capturar calificaciones.
- Control de grupos que incluyen 3 años de preparatoria, asignando las materias con su respectivo profesor.
- Consultas Web por parte del administrador, auxiliar, profesor y alumno.
- Generación de reportes solicitados por administrador y auxiliar.
- Consulta de calificaciones por parte de alumnos y profesores.

El sistema cuenta con los procesos necesarios para el mantenimiento y manipulación de la base de datos, como son: altas, bajas, consultas y modificaciones.

El desarrollo del sistema, se llevó a cabo bajo metodología de Ingeniería de Software y el diseño de bases de datos relacionales, se estudió el lenguaje de programación Web PHP, el sistema de gestión de base de datos relacional MySQL y la integración de las bases de datos en las aplicaciones Web. Se utilizaron también elementos de notación UML (Lenguaje Unificado de Modelado) dentro de todo el desarrollo del proyecto con la ayuda de herramientas como SmartDraw 2009 y StarUML.

Cabe recalcar que el sistema se probó en Web sobre una plataforma Apache web server Versión 1.3.41 (Linux), utilizando PHP Versión 5.2.5 y MySQL versión 4.1.22 standard; y se probó también sobre plataforma Windows XP Professional SP3, con la ayuda de Wamp Server 2.0 que incluye: Apache 2.2.11, MySQL 5.1.36 y PHP 5.3.0, ambos con buenos resultados.

PERSPECTIVAS

- Conveniencia de agregar un módulo para creación y manejo de usuarios, y mejorar la seguridad del sistema mediante la utilización de sesiones para mantener el estado entre diferentes conexiones y disponer de unas variables con valores persistentes durante toda la conexión del usuario, así como la utilización de SSL (Secure Sockets Layer, que es un protocolo usado para establecer comunicación segura entre un Servidor y un Cliente).
- Generación de estadísticas para graficar la cantidad de alumnos reprobados y aprobados por materia.
- Conveniencia de complementar el sistema con un módulo para el manejo de caja y pagos.

BIBLIOGRAFÍA

[PrR2002] Pressman R. S.
"Ingeniería de Software"
Mc. Graw Hill
4ª Edición
España, 2002

[ScA1996] Scott, Annette
"Data Modeling and Relational Database Design"
4ª Edición
USA, 1996

[Hal1994] Hawryszkiewycz, I.T
"Análisis y Diseño de Bases de Datos"
Edit. Megabyte
1ª Edición
1994.

[BaA2009] Barrientos Aleída
"*Proceso Metodológico de Auditoría Informática aplicado a la evaluación y seguimiento de Sistemas de Gestión desarrollados con el estándar de modelado UML*"
Tesis de Maestría en Ingeniería Informática
Universidad de Oriente La Habana Cuba - Universidad Autónoma Tomás Frías Potosí-Bolivia, 2002

[BoG1999] Booch Grady
"*El lenguaje Unificado de Modelado*"
Editorial Addison Wesley
1ª Edición
1999

[LaC2002] Larman Craig
"*UML y Patrones Una introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado*"
Editorial Prentice Hall
2ª Edición
2002

[Jal1999] Jacobson Ivar
"*El Proceso Unificado de Modelado*"
Editorial Addison Wesley
1ª Edición
1999

[RuJ1998] Rumbaugh James
"*Modelado y Diseño Orientado a Objetos con OMT*"
Editorial Addison Wesley
1ª Edición
1998

[**MSQ2009**] MySQL
http://dev.mysql.com/doc/refman/5.0/es/security.html
Noviembre 2009

[**PHP2009**] PHP
http://www.lsi.us.es/cursos/cursophp/
Noviembre 2009

[**MD52009**] MD5
http://www.maestrosdelweb.com/editorial/md5/
Noviembre 2009

[**MeW2009**] MediaWiki
"Encriptación MD5"
http://es.wikipedia.org/wiki/MD5
Noviembre 2009

[**MeW2009**] MediaWiki
"Cliente-servidor"
http://es.wikipedia.org/wiki/Cliente-servidor
Noviembre 2009

[**MeW2009**] MediaWiki
"MySQL"
http://es.wikipedia.org/wiki/MySQL
Noviembre 2009

[**DeD2009**] Database Tools
http://www.devdirect.com/All/DatabaseTools_SGROUP_0005.aspx
Agosto 2009

[**UIA2009**] Universidad Iberoamericana
http://www.uia.mx
Noviembre 2009

[**SIAE2009**] Universidad nacional Autónoma de México
http://www.dgae-siae.unam.mx/
Noviembre 2009

[**IbP2009**] Universidad Iberoamericana Plantel Golfo-Centro
http://www.iberopuebla.edu.mx/Prepalbero/
Noviembre 2009

[**DAE2009**] http://www.escolar.buap.mx/
Noviembre 2009

[**SADA2009**] http://escolar.tol.itesm.mx/sada/SADA-CCF.htm
Noviembre 2009

[OAE2009] (Objetivos de Administración Escolar)
Manual de Administración Escolar para los Centros de Estudios Tecnológicos, Industrial y de Servicios, Centros de Estudios Científicos y Tecnológicos en los Estados y Escuelas Particulares con Reconocimiento de Validez Oficial de Estudios
http://www.guerrero.gob.mx/pics/art/articles/329/file.manadesc.pdf
Noviembre 2009

[CII1]
[CII2] (Políticas de la SEP)
http://normatecainterna.sep.gob.mx:7007/
http://www.sep.gob.mx/wb/sep1/sep1_XII_Transparencia

[MeW2009]
MediaWiki
"*Bases de Datos*"
http://es.wikipedia.org/wiki/Base_de_datos
Julio 2009

[Lyc2008]
Lycos
"*Sistemas de Bases de Datos*"
http://usuarios.lycos.es/cursosgbd/UD2.htm
Julio 2009

[MeW2009]
MediaWiki
"*Ingeniería de Software*"
http://es.wikipedia.org/wiki/Desarrollo_de_software
Junio 2009

[MaA2001]
María Mercedes Marqués Andrés
"*Modelos de Datos*"
http://www3.uji.es/~mmarques/f47/apun/node82.html
2001-02-12

[RiV1998]
El Rincón del Vago
"*El Desarrollo de Sistemas de Información Empleando el Lenguaje de Modelado Unificado UML*"
http://html.rincondelvago.com/uml_5.html
Salamanca, 1998

ANEXO A. Simbología de los Diagramas

Diagrama de Casos de Uso

Diagramas de Secuencia

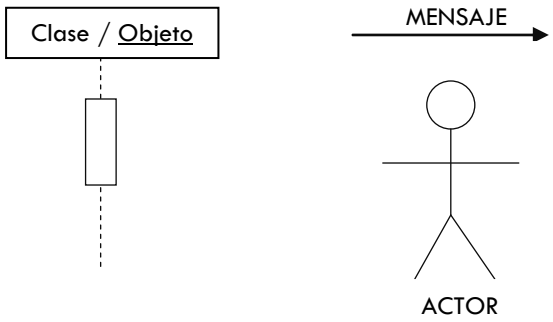


Diagrama de Navegación

Modelo Conceptual de la Base de Datos

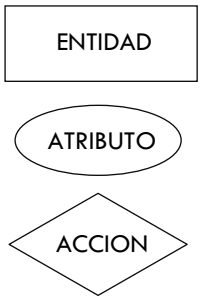
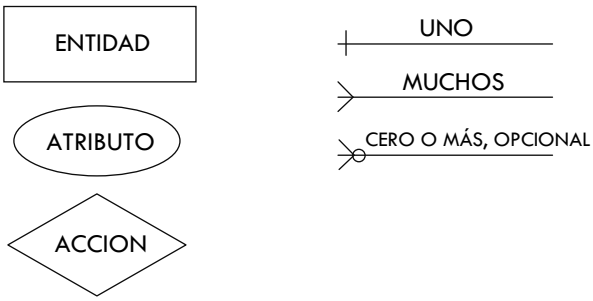


Diagrama Entidad-Relación



ANEXO B. Manual del Usuario

Manual del usuario para el sistema Administrador Escolar.

Barra de Menús

Opciones

Módulo de Altas (de la barra de menús)

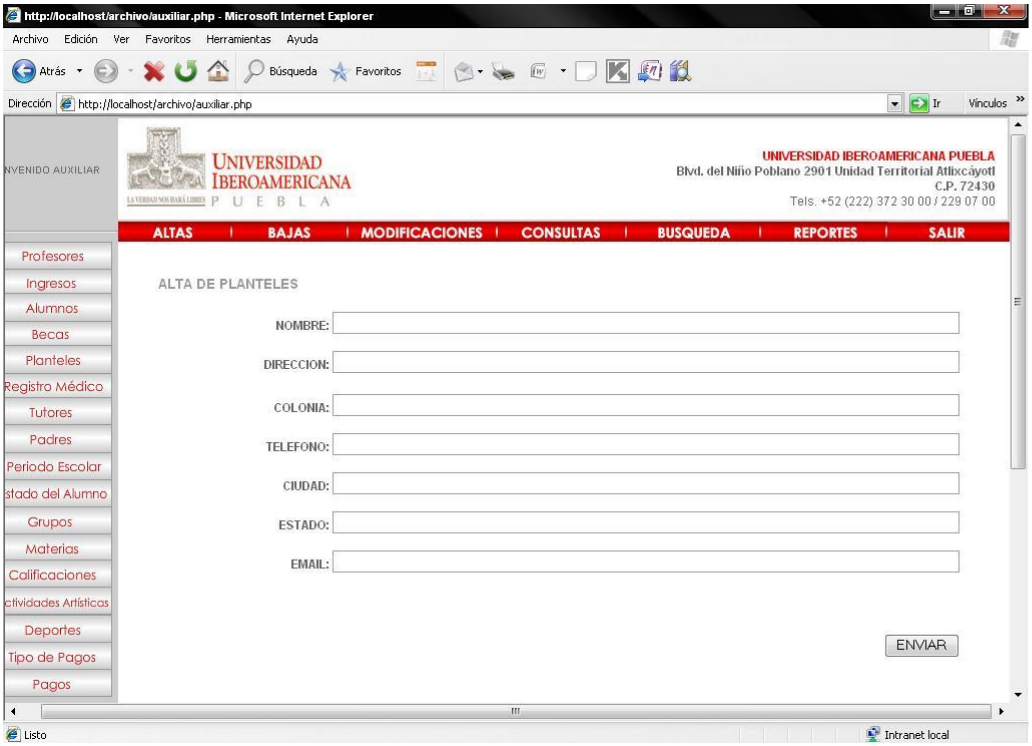
En este módulo el usuario podrá realizar el registro de información para: profesores, ingresos, alumnos, becas, planteles, registro médico, tutores, padres, periodo escolar, estado del alumno, grupos, materias, calificaciones, actividades artísticas, deportes, tipos de pagos y pagos; que en este caso llamaremos entidades.

Cada entidad tiene su pantalla de registro; en cada una de estas pantallas se despliegan los datos que el usuario debe capturar, una vez capturados los datos en cada campo o casilla, el usuario debe elegir el botón "ENVIAR" para que los datos sean guardados en la base de datos.

Esto es, en cada pantalla se visualizará una descripción de los datos, su casilla para insertar datos o elegir una opción y un botón de "ENVIAR", el cual tendrá la función de guardar la información. Por ejemplo:

Alta de Planteles

El usuario debe elegir opción Planteles y se desplegará la siguiente pantalla:



En este caso el usuario debe capturar: nombre, dirección, colonia, teléfono, ciudad, estado, email y posteriormente elegir botón "ENVIAR".

El proceso es el mismo para el resto de las entidades.

Módulo de Bajas (de la barra de menús)

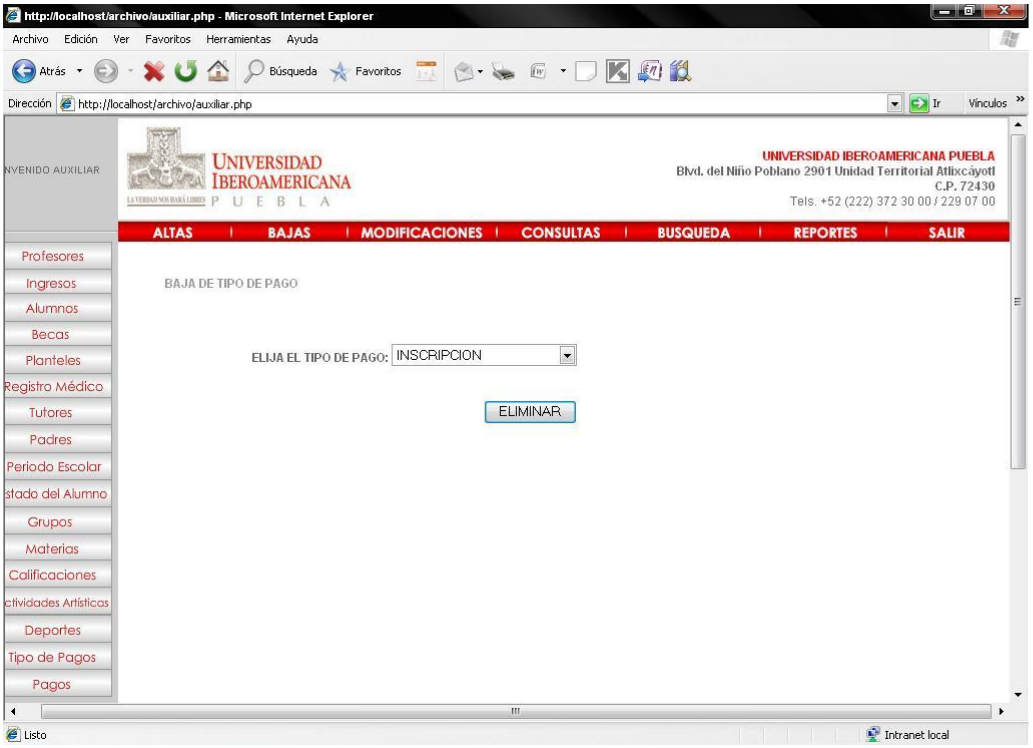
En este módulo el usuario podrá realizar la eliminación de información para: profesores, ingresos, alumnos, becas, planteles, registro médico, tutores, padres, periodo escolar, estado del alumno, grupos, materias, calificaciones, actividades artísticas, deportes, tipos de pagos y pagos.

Cada entidad tiene su pantalla de baja; en cada una de estas pantallas se despliegan los datos que el usuario debe capturar, una vez capturados los datos en cada campo o casilla, el usuario debe elegir el botón "ELIMINAR" para que los datos sean eliminados en la base de datos; el sistema desplegará un mensaje de confirmación para eliminación de información.

Esto es, en cada pantalla se visualizará una descripción de los datos, su casilla para insertar datos de la información que se desea eliminar y un botón de "ELIMINAR", el cual tendrá la función de eliminar la información encontrada de acuerdo a los datos que proporcionó el usuario. Por ejemplo:

Baja de Tipo de Pago

El usuario debe elegir opción Tipo de Pago y se desplegará la siguiente pantalla:



En este caso el usuario debe elegir: tipo de pago y posteriormente elegir botón "ELIMINAR". El sistema desplegará el siguiente mensaje: "¿Confirma eliminar registro?" en caso de elegir si, la información se eliminará de la base de datos, en caso contrario el sistema no hará nada.

El proceso es el mismo para el resto de las entidades.

Módulo de Modificaciones (de la barra de menús)

En este módulo el usuario podrá realizar la modificación de información para: profesores, ingresos, alumnos, becas, planteles, registro médico, tutores, padres, periodo escolar, estado del alumno, grupos, materias, calificaciones, actividades artísticas, deportes, tipos de pagos y pagos.

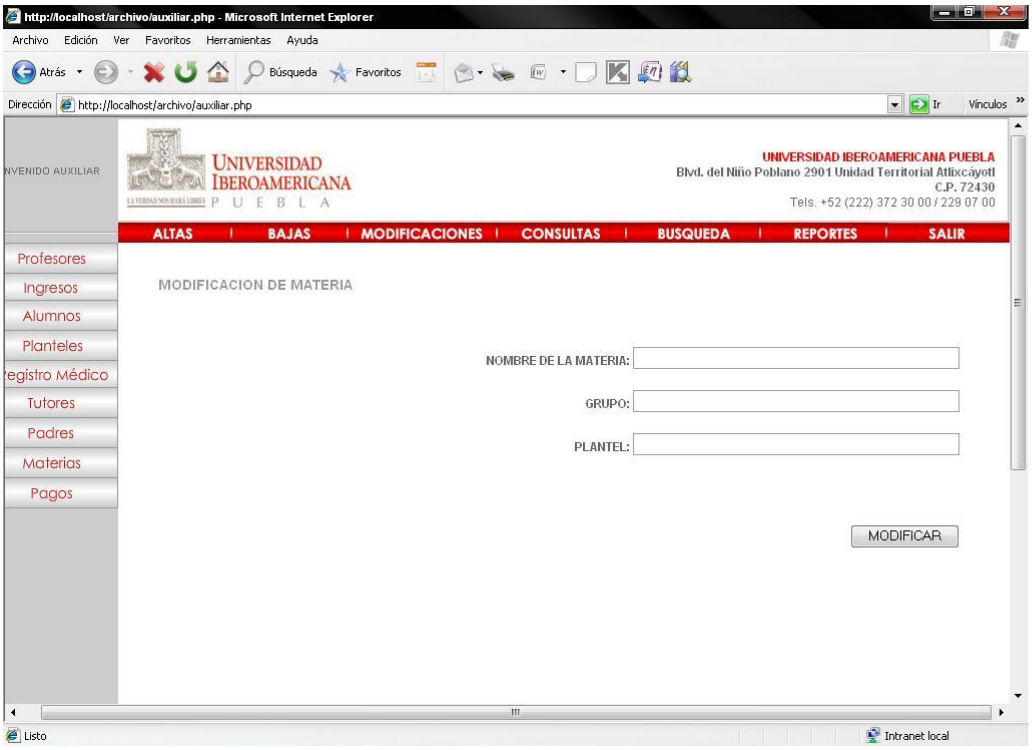
Cada entidad tiene su pantalla de modificación; en cada una de estas pantallas se despliegan los datos que el usuario debe capturar, una vez capturados los datos en cada campo o casilla, el usuario debe elegir el botón "MODIFICAR" para que los datos sean modificados en la base de datos.

Esto es, en cada pantalla se visualizará una descripción de los datos, su casilla para insertar datos de la información que se desea modificar y un botón de "MODIFICAR", el cual tendrá la función de modificar la información encontrada de acuerdo a los datos que proporcionó el usuario.

Por ejemplo:

Modificar Materias

El usuario debe elegir opción Materias y se desplegará la siguiente pantalla:



En este caso el usuario debe capturar: nombre de la materia, grupo, plantel y posteriormente elegir botón "MODIFICAR".

El proceso es el mismo para el resto de las entidades.

Módulo de Consultas (de la barra de menús)

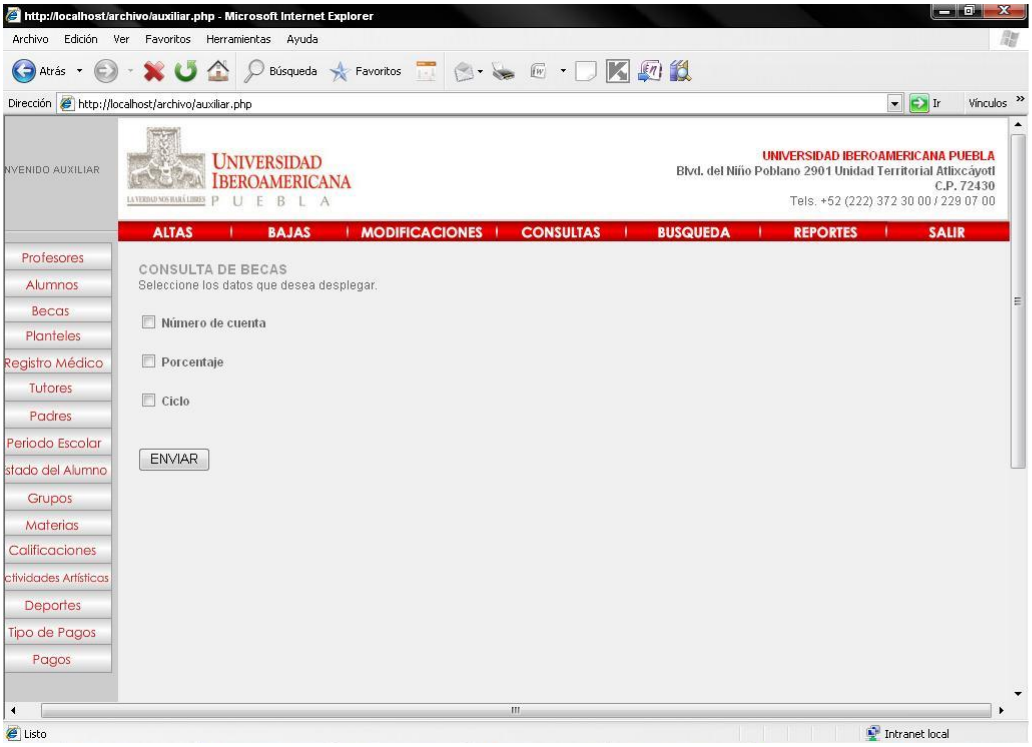
En este módulo el usuario podrá realizar la consulta de información para: profesores, ingresos, alumnos, becas, planteles, registro médico, tutores, padres, periodo escolar, estado del alumno, grupos, materias, calificaciones, actividades artísticas, deportes, tipos de pagos y pagos.

Cada entidad tiene su pantalla de consulta; en cada una de estas pantallas se despliegan los datos que el usuario debe capturar, una vez capturados los datos en cada campo o casilla, el usuario debe elegir el botón "ENVIAR".

Esto es, en cada pantalla se visualizará una descripción de los datos, su casilla para insertar datos de la información que se desea visualizar y un botón de "ENVIAR", el cual tendrá la función de buscar la información correspondiente a los datos que proporcionó el usuario. Por ejemplo:

Consultar Becas

El usuario debe elegir opción Becas y se desplegará la siguiente pantalla:



En este caso el usuario debe elegir las casillas que desea visualizar en la consulta y posteriormente elegir botón "ENVIAR".

El proceso es el mismo para el resto de las entidades.

Módulo de Búsquedas (de la barra de menús)

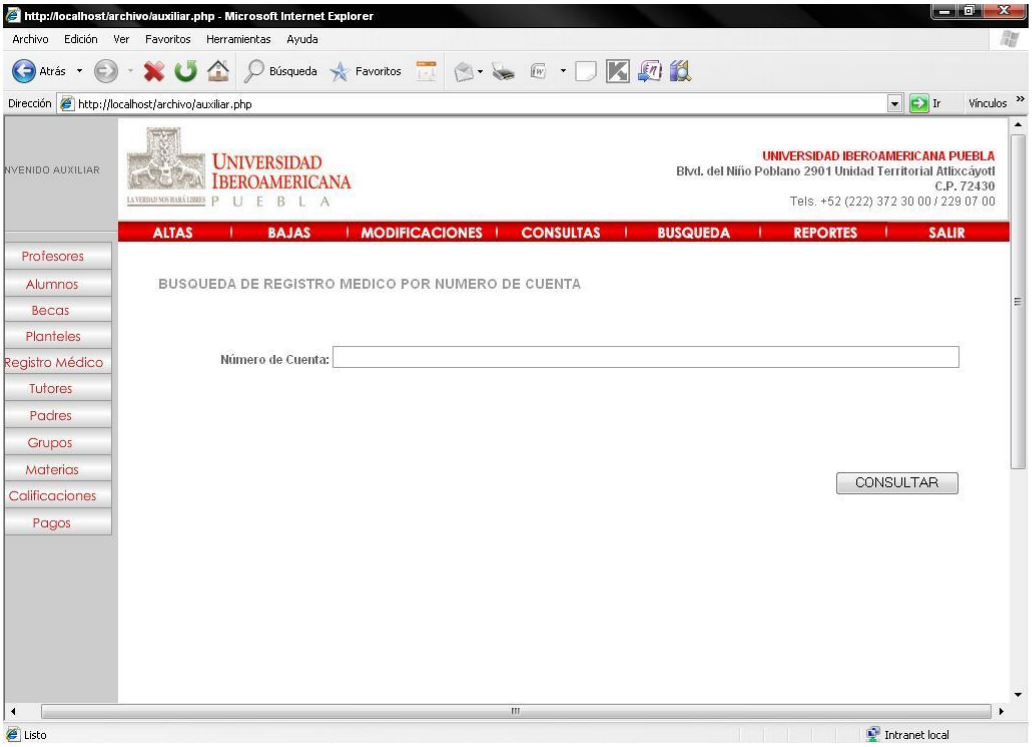
En este módulo el usuario podrá realizar la búsqueda de información para: profesores, ingresos, alumnos, becas, planteles, registro médico, tutores, padres, periodo escolar, estado del alumno, grupos, materias, calificaciones, actividades artísticas, deportes, tipos de pagos y pagos.

Cada entidad tiene su pantalla de búsqueda; en cada una de estas pantallas se despliegan los datos que el usuario debe capturar, una vez capturados los datos en cada campo o casilla, el usuario debe elegir el botón "CONSULTAR".

Esto es, en cada pantalla se visualizará una descripción de los datos, su casilla para insertar datos de la información que se desea visualizar y un botón de "CONSULTAR", el cual tendrá la función de buscar la información correspondiente a los datos que proporcionó el usuario. Por ejemplo:

Consultar Registro Médico

El usuario debe elegir opción Registro Médico y se desplegará la siguiente pantalla:



En este caso el usuario debe capturar el número de cuenta del alumno y posteriormente elegir botón "CONSULTAR".

El proceso es el mismo para el resto de las entidades.

Módulo de Reporte (de la barra de menús)

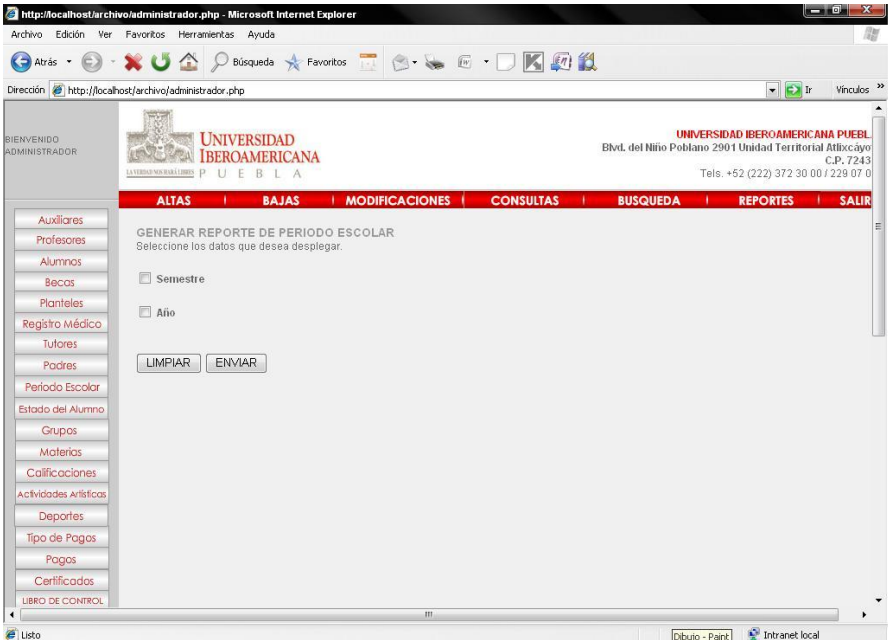
En este módulo el usuario podrá realizar reportes para: profesores, ingresos, alumnos, becas, planteles, registro médico, tutores, padres, periodo escolar, estado del alumno, grupos, materias, calificaciones, actividades artísticas, deportes, tipos de pagos y pagos; estos reportes se crearán en Excel.

Cada entidad tiene su pantalla de reporte; en cada una de estas pantallas se despliegan los datos que el usuario debe capturar, una vez capturados los datos en cada campo o casilla, el usuario debe elegir el botón "ENVIAR".

Esto es, en cada pantalla se visualizará una descripción de los datos, su casilla para insertar datos de la información que se desea visualizar y un botón de "ENVIAR", el cual tendrá la función de buscar la información correspondiente a los datos que proporcionó el usuario y generar los reportes en Excel, el usuario deberá elelgir donde guardar esos reportes. Por ejemplo:

Reporte de Período Escolar

El usuario debe elegir opción Período Escolar y se desplegará la siguiente pantalla:

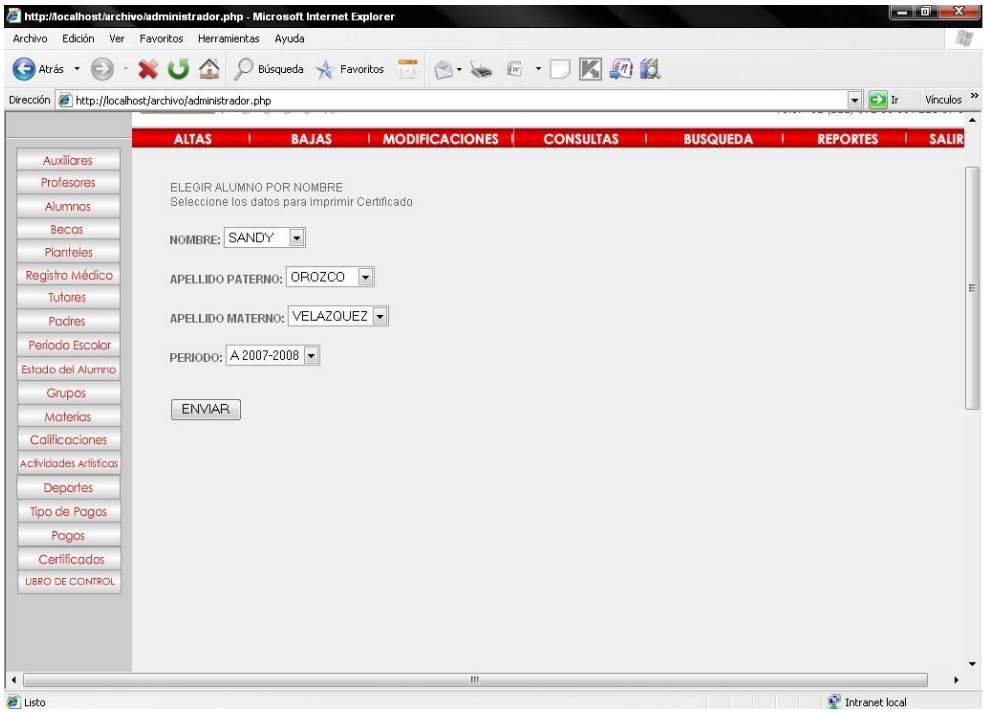


En este caso el usuario debe elegir semestre y/o año para visualizarlos en el reporte y posteriormente elegir botón "ENVIAR".

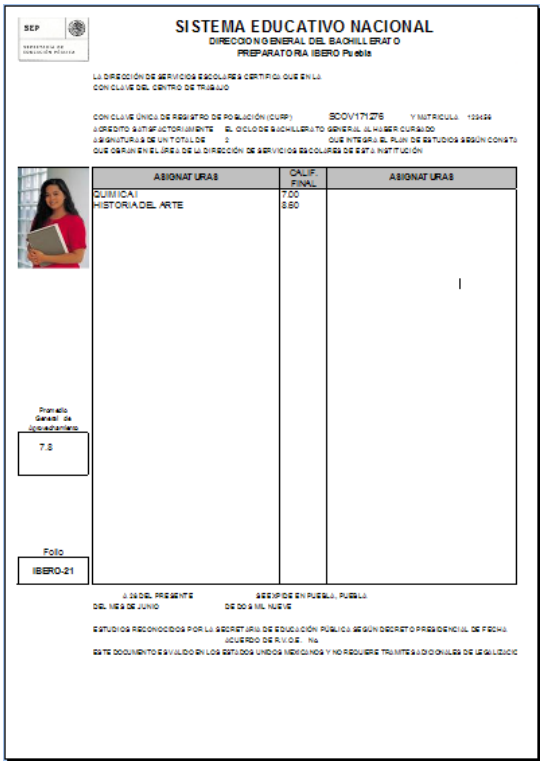
El proceso es el mismo para el resto de las entidades.

Reporte Certificado

En esta opción el usuario podrá generar certificados para los alumnos que hayan concluido con la carga de materias que indica la institución. Por ejemplo, el usuario debe elegir la opción Certificados y se desplegará la siguiente pantalla:



En este caso el usuario debe elegir los datos del alumno para el cual desea generar el certificado, este certificado se generará con un folio asignado por el sistema. En la siguiente figura se da un ejemplo del certificado que se generaría con los datos que están en pantalla:



SEP

SISTEMA EDUCATIVO NACIONAL

DIRECCION GENERAL DEL BACHILLERATO

PREPARATORIA IBERO Puebla

LA DIRECCIÓN DE SERVICIOS ESCOLARES CERTIFICA QUE EN LA
CON CLAVE DEL CENTRO DE TRABAJO

CON CLAVE ÚNICA DE REGISTRO DE POBLACIÓN (CURP) SCOV171276 Y MATRICULA 123456
ACREDITO SATISFACTORIAMENTE EL CICLO DE BACHILLERATO GENERAL AL HABER CURSADO
ASIGNATURAS DE UN TOTAL DE 2 QUE INTEGRA EL PLAN DE ESTUDIOS ARRIBA CONSTA
QUE OBRAN EN EL ARCHIVO DE LA DIRECCIÓN DE SERVICIOS ESCOLARES DE ESTA INSTITUCIÓN

ASIGNATURAS	CALIF. FINAL	ASIGNATURAS
QUIMICA I	7.00	
HISTORIA DEL ARTE	8.60	

Promedio General de Aprovechamiento
7.8

Folio
IBERO-21

A 24 DEL PRESENTE
DEL MES DE JUNIO

SE EXPIDE EN PUEBLA, PUEBLA
DE DOS MIL NUEVE

ESTUDIOS RECONOCIDOS POR LA SECRETARIA DE EDUCACIÓN PUBLICA SEGÚN DECRETO PRESIDENCIAL DE FECHA
ACUERDO DE R.V.O.E. No.
ESTE DOCUMENTO ES VALIDO EN LOS ESTADOS UNIDOS MEXICANOS Y NO REQUIERE TRAMITES ADICIONALES DE LEGALIZACIO

Módulo de Salir (de la barra de menús)

Esta opción tiene la funcionalidad de cerrar la sesión del sistema.