

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Computación

**“Sistema WEB para Control Escolar
de un Bachillerato General”**

Tesina

que para obtener el Título de:

Licenciado en Ciencias de la Computación

Presenta:

Guillermo Pérez Arrieta

Asesor:

Mtra. Alma Delia Ambrosio Vázquez

Puebla, Pue. Abril 2013

INDICE

Agradecimientos.....	5
Prefacio.....	7
Introducción.....	9
Capítulo I. Estado del Arte.....	14
1.1. Estado del Arte.....	12
1.1.1. Coleg-ERP.....	12
1.1.2. School Manager.....	15
1.1.3. Mi escuela.com.....	16
1.2. Justificación.....	18
Capítulo II Marco Teórico.....	20
2.1. Paradigma Orientada a Objetos.....	20
2.1.1. Antecedentes.....	21
2.2 Metodologías de Análisis y Diseño de Software.....	23
2.2.1. Proceso Unificado.....	25
2.3. Lenguaje Unificado de Modelado(UML).....	27
2.4. Diseño de la Base de Datos.....	29
2.4.1. Modelo entidad-relación.....	30
2.4.1.1. Relaciones.....	31
2.4.1.2. Atributos.....	32
2.4.1.3. Identificadores Únicos.....	33
2.4.2. Normalización.....	34
2.4.2.1. Dependencia Funcional.....	35
2.4.2.2. Formas Normales.....	35
2.5. Plataforma de Implementación.....	38
2.5.1. WAMP.....	39
2.5.1.1. Windows.....	39

2.5.1.2. Apache.....	40
2.5.1.3. MySQL.....	42
2.5.1.4. PHP.....	44
Capítulo III. Analisis y Diseño.....	47
3.1. Requisitos funcionales	47
3.2. Requisitos no funcionales.....	49
3.3. Análisis del Sistema.....	50
3.3.1. Modelo de casos de uso.....	50
3.3.1.1. Diagrama de casos de uso.....	50
3.3.1.2. Especificación de casos de uso.....	51
3.3.1.3. Escenarios.....	64
3.3.2. Modelo conceptual.....	73
3.3.2.1. Diagrama de clases.....	73
3.3.2.2. Diccionario del modelo.....	74
3.4. Diseño del Sistema.....	77
3.4.1. Modelo de análisis.....	77
3.4.1.1. Clases de análisis.....	77
3.4.1.2. Diagramas de clases de análisis.....	79
3.4.1.3. Diagramas de interacción.....	82
3.4.1.3.1. Diagramas de secuencia.....	82
3.4.1.3.2. Diagramas de colaboración	86
Capítulo IV. Diseño de la Base de Datos.....	91
4.1. Diseño conceptual.....	91
4.1.1. Modelo entidad-relación.....	91
4.1.1.1. Entidades.....	92
4.1.1.2. Atributos.....	93
4.1.1.3. Identificadores únicos.....	94
4.1.1.4. Relaciones.....	99
4.1.1.5. Normalización.....	103

4.2. Diseño lógico.....	106
Capítulo V. Implementación.....	112
5.1. Implementación.....	112
5.1.1. Base de datos.....	113
5.1.2. Interfaz del sistema.....	117
5.1.2.1. Página de inicio.....	117
5.1.2.2. Interfaz del usuario.....	118
5.1.2.3. Interfaz de administración.....	128
5.2. Pruebas del Sistema.....	134
5.2.1. Caja blanca.....	134
5.2.2. Caja negra.....	140
Conclusiones.....	146
Perspectivas.....	148
Bibliografía.....	149

AGRADECIMIENTOS

A Dios

*Por la vida y las bendiciones que me otorgas aún
sin pedir las.*

A mi Madre

*Por brindarme tu amor incondicional e inculcarme
grandes valores, por ser ejemplo de esfuerzo
constante para sacar adelante a nuestra familia y
darme la oportunidad de ser un profesionalista,
gracias Mamá.*

(A ti mi mamita querida, dedico este trabajo)

A mi Padre(Finado)

*Por ser un ejemplo de trabajo, responsabilidad y
amor a la familia. Gracias por mostrarme el
camino.*

***(Este trabajo es también para ti, sé que en
donde te encuentres, lo apreciarás)***

A mi Abue

*Por ser el pilar de la familia y mostrarnos que
estando unidos podemos vencer la adversidad, tu
fortaleza y consejos son una inspiración para ser
mejores.*

A mi Esposa

*Por aceptar ser mi compañera de viaje, tu cariño
y amor me impulsan para lograr viejos y nuevos
propósitos, gracias por brindar alegría y calor a
mi vida.*

A mis hermanos

*Por compartir conmigo los triunfos y fracasos, las
risas y lágrimas, los juegos y obligaciones, los
consejos y reprimendas, gracias por caminar
conmigo y ayudarme a ser un mejor profesionalista
y a continuar creciendo como persona.*

A mis Familiares

A mis Tíos, Primos y Sobrinos que hemos tenido la oportunidad de convivir, gracias por compartir conmigo sus experiencias y darme ánimo para nunca desistir en los propósitos que me planteé.

A mi Asesora

Por su apoyo y paciencia en la elaboración de este trabajo, por contribuir a realizar uno de los logros más anhelados en mi vida académica, Gracias por compartir su tiempo y conocimientos.

A mis Maestros

Por colaborar con sus conocimientos a mi formación académica, desde Preescolar hasta la Universidad, a todos Ustedes mi sincero agradecimiento por formarme como estudiante y enriquecer mi mente y mi persona.

A Pipis

Por el cariño y apoyo brindado en los momentos necesarios y por contribuir de manera especial a la formación de mí familia.

A mi amigo Ramón

Por sus consejos y apoyo desinteresado en la elaboración de este proyecto, tus comentarios y conocimientos técnicos fueron de gran utilidad para que este trabajo fuera posible.

A mis Amigos

Con todos aquellos que de una u otra forma he compartido momentos significativos, y han dejado una huella difícil de borrar, a todos ellos gracias por su presencia en mi vida.

PREFACIO

El presente trabajo se encuentra enfocado a desarrollar un sistema WEB que pueda ser utilizado en un Bachillerato General, el cual es una Institución pública que pertenece a la SEP del Estado de Puebla; con el fin de facilitar las tareas administrativas que se realizan en el mismo.

Para desarrollar el sistema mencionado se analizaron e incluyeron las funciones que son más susceptibles de ser automatizadas, así como las que demandan una mayor atención dentro de los procesos de control escolar en la Institución.

Sustentado en una metodología robusta para el desarrollo de software como lo es UP (Proceso Unificado), se trazó un diseño que cumpliera con los requisitos y funcionalidades esperadas, perfiladas a partir de casos de uso y escenarios que establecen las características necesarias del sistema para obtener una alternativa de solución.

Así mismo, mediante la aplicación de la teoría de Bases de Datos Relaciones se elaboraron los modelos Entidad-Relación y Relacional que nos ayudan a implementar la Base de Datos, la cual permite ejecutar las funciones de integridad, persistencia y transacciones del sistema propuesto.

Por otra parte, se plantea que la naturaleza de la herramienta propuesta, sea utilizando las tecnologías de desarrollo de software más actuales, además de cuidar el aspecto de inversión económica en su elaboración, implantación y manejo, por tanto se eligió la plataforma WAMP (acrónimo de Windows, Apache, MySQL, PHP) para su fabricación.

Windows como Sistema Operativo nativo en la Institución que se desea implementar; Apache como Servidor de páginas WEB; MySQL como Administrador

de la base de Datos del Sistema y PHP para la implantación de las reglas de negocio.

Adicionalmente se empleó HTML (Lenguaje de Marcado de Hipertexto) y CSS (Hojas de Estilo en Cascada) para complementar el desarrollo de la interfaz del usuario.

Por ultimo se realizaron y documentaron las pruebas aplicadas al sistema WEB desarrollado, las cuales fueron: modulares e integrales.

INTRODUCCIÓN

Administración Escolar [9]

La administración escolar entendida como el conjunto de actividades dirigidas a la ordenación de esfuerzos; determinación de objetivos académicos, y a la creación y aplicación de normas para los alumnos, personal docente y administrativo; tiene la finalidad de establecer en la Institución las reglas de enseñanza-aprendizaje y un gobierno escolar eficiente y exitoso.

Tiene como objetivo primordial facilitar el proceso de enseñanza-aprendizaje, mediante la maximización de los recursos de la Institución; y para lograrlo se requiere de la ejecución de actividades tales como: establecer relaciones entre la escuela y la comunidad; desarrollar planes y programas de estudios, organizar y evaluar a los alumnos; gestionar y administrar los recursos materiales, humanos y financieros, etc.

Al mismo tiempo la administración escolar es un proceso, el cual está conformado por funciones como: planificación, organización, dirección, coordinación, ejecución y **control**.



Y es precisamente la función de control escolar que se desarrolla en las instituciones educativas públicas y específicamente en los Bachilleratos Generales del Estado el tema de interés del presente trabajo.

Control Escolar [10]

El Control Escolar entendido como: la función administrativa por medio de la cual se evalúa el rendimiento, es un elemento del proceso educativo que incluye todas las actividades que se emprenden para garantizar que las acciones reales coincidan con las acciones planificadas.

Permite evaluar los resultados y saber si estos son adecuados a los planes y objetivos que se desea conseguir, a través de esta función se pueden registrar los datos pertinentes, precisar los errores y corregir las fallas, para que la organización se encuentre encaminada de manera correcta. El control debe llevarse en cualquier nivel de la organización, garantizando de esta forma que en la misma se cumplan los objetivos programados.

Ámbito

En el Bachillerato General “Benito Juárez” se llevan a cabo diversas actividades para realizar la función de control escolar, sin embargo, de todas ellas las que representan el objetivo de este estudio, son las que están dedicadas al control y registro de la información académica y de evaluación de los alumnos.

Como actividades principales de control escolar en la Institución mencionada, se tienen las de registro de datos de aspirantes; inscripción y asignación de grupos a los alumnos inscritos; así mismo el control de evaluación y seguimiento de asistencias y la emisión de documentos que comprueban el desempeño académico de estos.

Se debe tener en cuenta el hecho de que al realizar las actividades mencionadas anteriormente, estas desencadenan una serie de procesos paralelos u otras acciones que se deben realizar de manera anticipada para que se pueda cumplir con la función deseada.

Se debe destacar que dentro de la Institución mencionada, en la mayoría de los casos no se cuenta con herramientas adecuadas que apoyen a las áreas encargadas de aplicar las funciones de control escolar.

Descripción del problema

El Bachillerato General “Benito Juárez” es una Institución perteneciente a la Secretaría de Educación Pública del Estado de Puebla, sus objetivos están enfocados a brindar servicios educativos de calidad los jóvenes de la comunidad de San Francisco Ocotlán del Municipio de Coronango, que año con año ingresan al mismo.

Por tanto el personal encargado de Control Escolar en el Bachillerato, debe realizar las tareas de registro, validación, control e impresión de la documentación solicitada para cada uno de los procesos.

Sin embargo, dichas tareas al ser realizadas de manera manual y algunas con el uso de herramientas como Excel, limitan la eficiencia de los procesos ya que la información es generada de forma repetitiva; debido a esto su integridad se ve comprometida derivando en una inversión extra de tiempo y recursos al tener que estar revisando y validando constantemente la información que se genera de los alumnos de la Institución.

Por lo anterior se plantea elaborar para el Bachillerato, un software que pueda ser utilizado desde el Internet y/o la intranet de la escuela, para que el personal de control escolar tenga un almacén de datos único y realice las tareas de administración de manera más ágil y eficiente.

Objetivos

General.-

Desarrollar un sistema WEB que permita administrar de manera eficiente la información de los alumnos que se genera en el Bachillerato General.

Específicos.-

- Elegir y aplicar la metodología de desarrollo de software más apropiada para el diseño y construcción del sistema.
- Realizar el diseño y la implementación de la Base de datos, de conformidad con los datos obtenidos a través del modelo entidad-relación.
- Construir la interfaz del usuario de acuerdo con los diseños y modelos obtenidos a partir de la metodología de software y teoría de BD.
- Elaborar un sistema con la tecnología de mayor actualidad mediante herramientas de bajo costo para la Institución.
- Proporcionar un sistema que facilite la realización de funciones de control escolar específicas para el Bachillerato General.

Metodología de desarrollo

Para el diseño del sistema que permite cumplir con los objetivos señalados en el apartado anterior se realizaron una serie de actividades organizadas que se describen a continuación:

1. Se revisaron las diferentes metodologías de desarrollo de software para elegir la más adecuada para nuestro proyecto.
2. Elegida la metodología de desarrollo UP, se aplicaron los procesos de análisis y diseño indicados en la misma, obteniendo como resultado diagramas y modelos para la creación del sistema.
3. Así mismo, se aplicó la teoría de Bases de Datos Relacionales, para el elaborar el diseño de la Base de Datos del sistema.

4. Posteriormente se implementaron los modelos y diagramas obtenidos en los procesos anteriores, mediante las herramientas proporcionadas por la plataforma de implementación WAMP.
5. Por último se realizaron pruebas conforme se desarrollaba cada módulo para revisar su funcionalidad y para concluir se hicieron pruebas al sistema en su conjunto para verificar que presentaba el desempeño esperado.

La consecución de este proyecto facilitará muchas de las funciones de control escolar dentro del Bachillerato General, además de conformarse como un sistema base con todas las características para agregar más funcionalidades en un futuro.

CAPITULO I

ESTADO DEL ARTE

1.1. Estado del Arte

Actualmente existe diversidad de software para su aplicación en las áreas de Control Escolar, algunas de las posibilidades representativas se describen a continuación:

1.1.1. Coleg-ERP [11]

Es una plataforma informática que permite el manejo y control de las calificaciones, circulares, calendarios, reportes, tareas y aplicaciones que servirán como una herramienta para la gestión estudiantil para evitar errores humanos.

Controla el cobro de colegiaturas de forma automática y permite emitir recordatorios de pago de forma inmediata a los padres de familia para mantener una estadística de cobro favorable, enviando mail's o bien de forma impresa con la frecuencia que se decida; así mismo los padres de familia pueden imprimir CFD's.

Permite mantener informados a los padres de familia de la conducta estudiantil, emocional y personal de sus hijos; El sistema: envía de manera segura, personalizada y confidencial, el avance educativo, social, psicológico y personal de los alumnos.

*El modelo de implementación se muestra en la **figura 1.1**. El modelo muestra el esquema de operación de la herramienta, sin embargo no se menciona explícitamente el equipamiento o infraestructura con que debe contar la Institución para el funcionamiento del mismo. Situación que representa un problema significativo debido a la inversión que se debe realizar por las escuelas públicas, las cuales*

no cuentan con los recursos para implementar una herramienta de esta magnitud.



ación

Otra característica relevante de este sistema es el precio, el cual menciona se indexa en proporción al cobro de las colegiaturas. Sin explicar en qué consiste realmente este modelo de licenciamiento, y por lo tanto se debe considerar como una inversión adicional.

1.1.2. School Manager [12]

Es un Sistema de Administración Escolar en aplicación de escritorio, basada en infraestructura Cliente/Servidor. Cuenta con manejo de usuarios y perfiles para controlar el acceso y la seguridad del sistema. Esta desarrollado con Tecnología .NET. Tiene soporte técnico por Skype, Teléfono, y Redes sociales (Facebook, Twitter, Youtube, MSN.)

Puede ser usado en la red local o en servidores remotos, permite capturar fichas médicas y psicológicas de los alumnos, así como dar

seguimiento a las incidencias de las mismas. Permite elaborar cartas, memos u oficios y permite la exportación a hojas de cálculo en Excel y PDF.

Para este software se mencionan los requerimientos de una infraestructura accesible en el equipamiento, sin embargo no hace diferencia de si las características mencionadas son para el cliente o para el servidor, situación que no deja clara la inversión para la implementación del mismo.

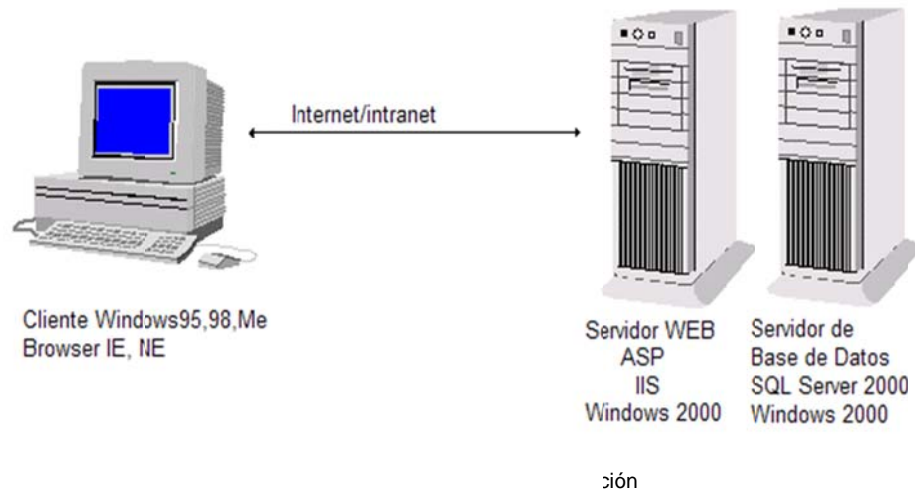
En el aspecto de licenciamiento, el sistema se puede ir adquiriendo de manera modular, lo que indica que la inversión puede no ser tan accesible, aunque el beneficio puede estar en adquirir únicamente el o los módulos que sean necesarios para la Organización.

1.1.3. Mi-escuela.com [13]

Es un Sistema de Administración y Control Escolar integral y flexible ideal para todo tipo de planteles educativos (escuelas primarias, secundarias, bachilleratos, universidades, etc.) que facilita a los usuarios el control escolar, académico y administrativo de Alumnos, Cuentas por Cobrar, Maestros, Aulas, Grupos, Calificaciones, Materias, etc.

Debido a su diseño internet/intranet, el software puede ser ejecutado desde cualquier computadora que cuente con un Browser o Navegador conectado a un Servidor WEB ya sea local o remoto, de forma que personal administrativo, académico, alumnos y padres de familia puedan tener acceso al sistema desde cualquier lugar en el momento que lo deseen.

El modelo de operación de la herramienta mi-escuela.com es el siguiente:



En el caso de esta herramienta se menciona explícitamente los requerimientos en materia de infraestructura para su implementación, haciendo la aclaración de que para el caso de su instalación en una intranet, únicamente se necesitaría un equipo con características de servidor.

Lo anterior indica que se debe invertir en más equipamiento para instalar este sistema.

Así mismo, el esquema de licenciamiento de la herramienta nos muestra dos opciones:

- 1. La primera, en la cual se paga una cuota fija mensual por número de alumnos, lo que permite a la escuela conectarse a los servidores del proveedor, el beneficio observado es en materia de infraestructura y licenciamiento de los servidores, ya que no se necesita adquirir ningún software o equipo adicional, sin embargo, este esquema hace dependiente a la*

Institución con el proveedor, ya que la información reside en los equipos de este último, así como la dependencia económica al pagar cuotas anuales por número de alumnos;

- 2. La segunda opción, permite la compra de la licencia del sistema, siendo el costo de la misma por cada servidor y basada en el número de alumnos inscritos, sin embargo si se incrementa la matrícula de alumnos se deberán comprar licencias adicionales, además de tener que adquirir los equipos servidores y el correspondiente licenciamiento de sistema operativo y base de datos para su funcionamiento. Lo cual genera una inversión considerable en materia de infraestructura y licenciamiento.*

1.2. Justificación

De los ejemplos mostrados anteriormente para las herramientas de control escolar, se puede observar que los sistemas actuales de control escolar han evolucionado hacia funcionalidades adicionales como es: el cobro y la emisión de comprobantes fiscales digitales, emisión de mails para mantener informados a los padres de familias sobre la situación escolar de los alumnos, registro y seguimiento de evaluaciones psicológicas, elaboración de reportes, cartas y oficios integrados en las mismas, control de inventarios, etc., por mencionar solo algunas, las cuales otorgan una capacidad de operación y funcionalidad extraordinaria en las Instituciones.

Sin embargo, también es visible el hecho de que dichas soluciones de control escolar se encuentran enfocadas hacia Instituciones con un nivel más que aceptable en captación de recursos económicos, el cual está presente generalmente en la educación privada, misma que les permite realizar egresos adicionales en materia de licenciamiento o infraestructura, y poder acceder a implementar una solución de esta magnitud.

Así mismo, teniendo en cuenta que los sistemas comerciales habitualmente son de propósito general, es posible que se deban adecuar diversas funcionalidades a los procesos específicos del Bachillerato, ya que es conocido el hecho de que las aplicaciones de propósito general normalmente no se adaptan al 100% a los requerimientos de las Organizaciones que los emplean.

Es por ello que las características de la solución que se generará en este proyecto, estarán enfocadas a atender procesos específicos definidos para un Bachillerato General, además de no generar gastos extra en materia de licenciamiento; también contemplará que la aplicación utilice la infraestructura con que cuenta actualmente la Institución, permitiendo optimizar los recursos y resolver la problemática específica del Bachillerato.

CAPITULO II

MARCO TEÓRICO

Para iniciar con la construcción de la herramienta software, es necesario establecer los conceptos teóricos que marcaran los procesos y actividades que deberemos ir cumpliendo para la consecución de los objetivos planteados.

En mención de lo anterior, procederemos a describir los elementos teóricos, que establecen la guía para orientar nuestros esfuerzos en la aplicación de los procesos de análisis, diseño, desarrollo, pruebas y liberación del sistema que queremos implantar.

2.1. Paradigma Orientado a Objetos

La tecnología para lograr la Orientación a Objetos (OO) ha logrado madurar comercialmente en la última década. Las siguientes generaciones de productos OO han tenido cada vez más éxito, hasta el punto de que actualmente la tecnología de objetos es el estándar de facto para el desarrollo de nuevos sistemas. [5]

Esto se debe en gran parte a que se dispone ya de una infraestructura de Ingeniería de Software con metodologías, técnicas y herramientas que aseguran la calidad del proceso de desarrollo. [5]

Parte fundamental de esta infraestructura es el UML (Unified Modeling Language), que permite modelar las diferentes fases del proceso de desarrollo de sistemas, pero que se enfoca principalmente en el análisis y diseño orientado a objetos. [5]

2.1.1. Antecedentes

- 1967 nace SIMULA precursor del concepto.
- Se hablaba de “Crisis del Software”, provocó el surgimiento de la Programación Estructurada.
- Surge otro problema: los sistemas no se terminaban, pocos terminaban cumpliendo los requisitos iniciales, no se usaban según lo planificado.
- El problema era cómo adaptar el software a nuevos requerimientos imposibles de haber sido planificados.
- El hardware, su disminución de precios y la explotación de PC fue el detonante final.
- Se necesitan desarrollar prototipos rápidos sin esperar que los requisitos iniciales estén totalmente precisos.
- Es necesario aprovechar el software existente. [5]

De aquí nace el concepto Orientado a Objetos (OO)

- En lugar de tratar de modelar un problema en algo familiar a la computadora, se trata ahora de acercar la computadora al problema.
- Modelar la realidad del problema a través de **entidades** independientes pero que interactúan entre si y cuyas fronteras no estén determinadas por su instrumentación computacional sino por la naturaleza del problema.
- Estas **entidades** serán denominadas objetos, por analogía con el concepto objeto en el mundo real. [5]

El principio más importante de la OO presupone un cambio radical en la forma de cómo se “piensan” los sistemas.

- Un objeto es una entidad (con estado e identidad) que tiene un comportamiento definido.

- Agrupa los procedimientos y datos relacionados, para manejarlos como unidades integrales e independientes.
- Se debe hacer un cambio de paradigma, donde la solución a un problema no se plantea en términos de un modelo computacional, sino en términos del problema mismo. **[5]**

Realizar análisis y diseño orientado a objetos es una manera de aplicar los conceptos básicos de la orientación a objetos:

- *Abstracción*: es la capacidad de conceptualizar entidades genéricas de información a partir de cosas concretas, enfatizando las características comunes de interés. (*Minimiza la complejidad*);
- *Encapsulación*: es la capacidad de esconder los detalles de cómo funciona un ente, detrás de su interface, solo es necesario conocer esta última. (*Esconde la complejidad*);
 - Polimorfismo : es la habilidad de esconder diferentes implementaciones tras una sola interface;
- *Modularidad*: es la capacidad de particionar algo complejo y difícil de manejar, en partes más sencillas y fáciles de manejar. (*Administra la complejidad*);
- *Jerarquía*: es la capacidad de manejar niveles de abstracción. (*Organiza la complejidad*);
 - Herencia: es la capacidad de los elementos de una jerarquía, de transmitir sus características desde los niveles más abstractos hasta los más concretos. **[5]**

Para definir modelos que representen:

- sistemas del mundo real (dominio del problema)
- sistemas de software que automaticen soluciones a los sistemas del mundo real (dominio de la solución) **[5]**

Para realizar análisis y diseño orientado a objetos, este se debe aplicar dentro del contexto de una *metodología* o proceso de desarrollo de software; utilizando una notación estandarizada que sea completa, robusta y extensible; apoyado por herramientas que permitan modelar visualmente; y buscando y aplicando “patrones” que promuevan la reutilización de elementos de análisis y diseño. [5]

2.2. Metodologías de Análisis y Diseño de Software

Una metodología de desarrollo de software describe un marco de trabajo que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información. [14]

Cada metodología de desarrollo de software cuenta con su propio enfoque para el desarrollo de software. Los enfoques más generales, que se desarrollan en las metodologías son los siguientes:

- *Modelo en cascada*: Marco de trabajo lineal.
- *Prototipado*: marco de trabajo iterativo.
- *Incremental*: Combinación de marco de trabajo lineal e iterativo.
- *Espiral*: Combinación de marco de trabajo lineal e iterativo.
- *RAD*: Rapid Application Development, marco de trabajo iterativo.
- *OO*: Orientado a Objetos, combina lo mejor de las demás metodologías. [14]

Modelo en cascada

Es un proceso secuencial de desarrollo en el que los pasos de desarrollo son vistos hacia abajo (como en una cascada de agua) a través de las fases de análisis de las necesidades, el diseño, implementación, pruebas (validación), la integración y el mantenimiento. [14]

Prototipado

El prototipado es el marco de trabajo de actividades dedicadas al desarrollo de software prototipo, es decir, versiones incompletas del software a desarrollar. **[14]**

Incremental

Provee una estrategia para controlar la complejidad y los riesgos, desarrollando una parte del producto software reservando el resto de aspectos para el futuro. **[14]**

Espiral

La atención se centra en la evaluación y reducción del riesgo del proyecto dividiendo el proyecto en segmentos más pequeños y proporcionar más facilidad de cambio durante el proceso de desarrollo, Cada viaje alrededor de la espiral atraviesa cuatro cuadrantes básicos: (1) determinar objetivos, alternativas, y desencadenantes de la iteración; (2) Evaluar alternativas; Identificar y resolver los riesgos; (3) desarrollar y verificar los resultados de la iteración, y (4) plan de la próxima iteración. **[14]**

Rapid Application Development (RAD)

El *desarrollo rápido de aplicaciones* (RAD) es una metodología de desarrollo de software, que implica el desarrollo iterativo y la construcción de prototipos. **[14]**

Metodologías de desarrollo Orientado a objetos

- *Diseño orientado a objetos (OOD) de Grady Booch*: también conocido como Análisis y Diseño Orientado a Objetos (OOAD). El modelo incluye seis diagramas: de clase, objeto, estado de transición, la interacción, módulo, y el proceso. **[14]**

- *Top-down programming*: evolucionado en la década de 1970 por el investigador de IBM Harlan Mills (y Niklaus Wirth) en Desarrollo Estructurado.
- *Proceso Unificado*: es una metodología de desarrollo de software, basado en UML. Organiza el desarrollo de software en cuatro fases, cada una de ellas con la ejecución de una o más iteraciones de desarrollo de software: creación, elaboración, construcción y las directrices. Hay una serie de herramientas y productos diseñados para facilitar la aplicación. Una de las versiones más populares es la de Rational Unified Process. [15]

2.2.1. Proceso Unificado (UP)

Teniendo en cuenta que en el sistema que está desarrollando se van a aplicar métodos y tecnologías de vanguardia que ofrezcan ventajas en materia de análisis, diseño, desarrollo y mantenimiento del mismo. Y considerando que las características y prestaciones que ofrece la metodología del Proceso Unificado (UP), cumplen con los requerimientos que se tienen planteados para su elaboración, se ha optado por aplicar dicha metodología para la elaboración del presente trabajo, la cual se describe a continuación:

El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado (UP) es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP. [15]

El Proceso Unificado es un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, el *Proceso Unificado de Rational (RUP)*, también es un

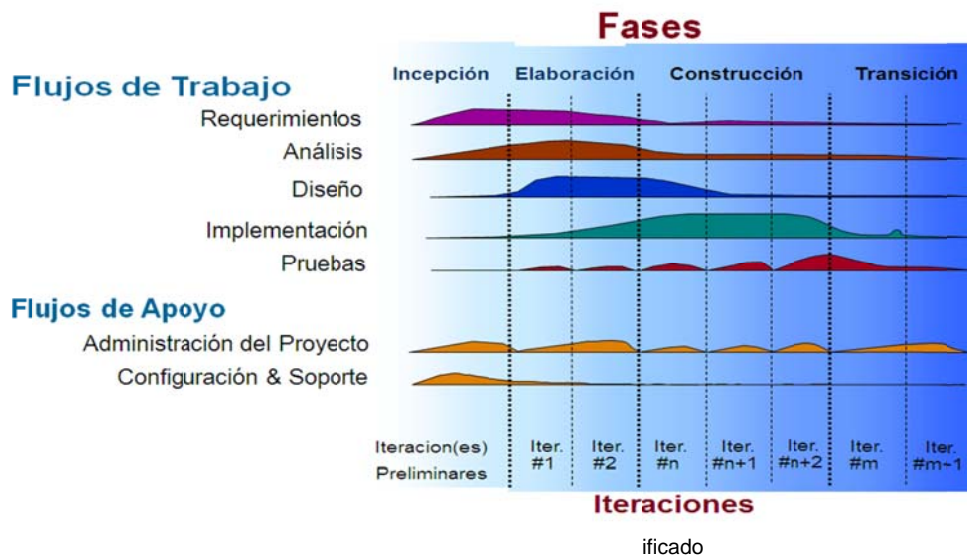
marco de trabajo extensible, por lo que muchas veces los dos nombres suelen utilizarse para referirse a un mismo concepto. [15]

Características:

Iterativo e Incremental

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio puede incluir varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un *incremento* del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo. [15]

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de requisitos, Diseño, Implementación y Prueba. [15]



Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto. El diagrama anterior ilustra como el énfasis relativo en las distintas disciplinas cambia a lo largo del proyecto. [15]

Dirigido por los casos de uso

En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc. el proceso dirigido por casos de uso es el RUP. [15]

Centrado en la arquitectura

El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando construyes un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc. [15]

Enfocado en los riesgos

El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero. [15]

2.3. Lenguaje Unificado de Modelado (UML)

La metodología de desarrollo de software (UP), utiliza como soporte para diseñar los modelos visuales de un sistema, el Lenguaje Unificado de Modelado o UML.

UML por sus siglas en inglés (*Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la

actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados. **[16]**

Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema, para documentar y construir. **[16]**

El UML combina lo mejor de

- Métodos centrados en datos (Métodos estructurados: ERD, DFD, STD).
- Métodos centrados en escenarios de uso.
- Métodos centrados en responsabilidades. **[5]**

UML se utiliza para modelar visualmente:

- La interacción del sistema con el mundo externo
- El comportamiento del sistema
- La estructura del sistema
- La arquitectura del sistema
- Los componentes del sistema. **[5]**

Para poder elaborar los modelos visuales mencionados anteriormente, UML cuenta con un catálogo de diagramas específico para cada necesidad, a continuación se describen de manera general los que tienen mayor importancia para nuestro proyecto.

Diagramas de:

Casos de uso.- se utilizan para visualizar la interacción del sistema con el mundo exterior.

Secuencia.- muestran cómo se realiza paso a paso cierta funcionalidad en el sistema.

Colaboración.- muestran cómo interactúan los objetos para lograr cierta funcionalidad en el sistema.

Clase.- modelan la estructura del sistema.

Componentes.- ilustran la organización y dependencia entre los componentes que definen la arquitectura del sistema.

Distribución.- permiten visualizar la distribución de los componentes del sistema a través de la empresa. [5]

El análisis y diseño del sistema para el desarrollo del sistema de control escolar para el “Bachillerato Benito Juárez”, se aplicara de conformidad con los puntos descritos anteriormente, utilizando el Paradigma Orientado a Objetos (POO), aplicándolo a través de la metodología del Proceso Unificado (UP), el cuál hará uso de la herramienta Lenguaje Unificado de Modelado (UML) para modelar los diagramas de la solución.

2.4. Diseño de la Base de Datos

La Base de datos del presente trabajo está fundamentada en el modelo de diseño de Bases de Datos relacionales el cual se describe a continuación:

Definición: Una Base de datos se puede definir como un conjunto de información relevante organizada de forma sistemática representada por entidades y sus interrelaciones. Los datos almacenados pueden ser de diversa índole y generalmente está estructurada por tablas. [4]

Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. [6]

Existen dos métodos convencionales que son de notable importancia para el diseño de bases de datos:

- El modelo entidad-relación; y
- La normalización. [6]

2.4.1. Modelo Entidad-Relación

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de un conjunto de objetos básicos llamados entidades y de relaciones entre estos objetos. Una entidad es una “cosa u objeto” en el mundo real que es distinguible de otros. [6]

Un modelo entidad-relación asiste al diseñador para definir y entender las cosas significativas acerca de cuál información necesita ser conocida o manejada y también las relaciones entre estas. [6]

El modelo entidad-relación categoriza todos los elementos de un sistema como una entidad (una persona, lugar o cosa) o una relación entre entidades. Ambas construcciones son representadas por la misma estructura, una tabla. [6]

La aplicación del modelo entidad-relación requiere los siguientes pasos:

1. Identificar las entidades del sistema y construir una tabla para representar cada entidad.

2. Identificar las relaciones entre las entidades y las tablas actuales extendidas o crear tablas nuevas para representar estas relaciones. [6]

2.4.1.1. Relaciones.

Las relaciones representan las reglas y la información que el negocio necesita.

- Una relación es una asociación importante entre dos entidades.
- Una relación es “que tiene que ver una cosa con otra”.
- Una relación es bi-direccional, asociación importante entre dos entidades. [6]

Los tipos de relaciones se expresan en el modelo entidad-relación, como lo indican las figuras: [6]

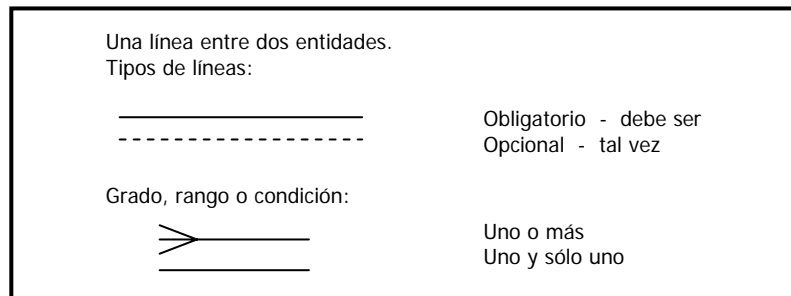


Figura 2.2. Expresión de las Relaciones

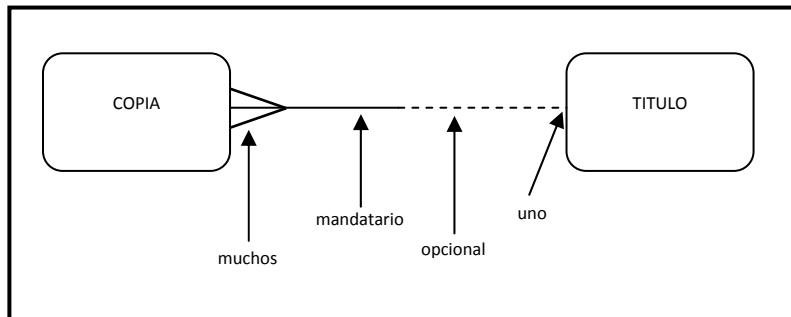


Figura 2.3. Ejemplo Relación entre dos entidades

Los tipos de relaciones pueden ser:

- *Relaciones Muchos-a-Uno*: Estas son las más comunes y muestran que una relación tiene un grado de uno o más en una dirección y uno y solamente uno en la otra dirección.
- *Relaciones Muchos-a-Muchos*: Hay un grado de uno o más en ambas direcciones y también son muy comunes. Usualmente son opcionales en ambas direcciones.
- *Relaciones Uno-a-Uno*: Hay un grado de uno y sólo uno en ambas direcciones. Este tipo de relaciones es bastante raro, comúnmente indican que las dos entidades son realmente la misma entidad en términos de negocio. **[6]**

2.4.1.2. Atributos.

Una entidad se representa mediante un conjunto de atributos. Los atributos describen propiedades que posee cada miembro de un conjunto de entidades. Los atributos describen una entidad por que la: **[6]**

- Habilitan.
- Identifican.
- Clasifican.
- Numeran.
- Expresan su estado.

Los atributos representan un tipo de descripción o detalles para cada instancia. **[6]**

Pueden ser opcionales u obligatorios, si son obligatorios se anteceden por el carácter “*”, en caso contrario con el carácter “O”. **[6]**

Cada atributo está conformado de los siguientes datos: **[6]**

Descripción: nombre que identifica al atributo.

Formato:

- Tipo. Carácter, fecha, imagen, entero, dinero, fotografía, sonido, texto.
- Longitud Máxima. Solamente es usado con los tipos como son: carácter o número.
- Promedio longitud. El tamaño normal del atributo.
- Lugar decimal. El número de dígitos después del lugar decimal.
- Unidad de medida. Puede ser peso, dimensión o capacidad.

Valores admisibles: Especificar valores válidos para los atributos:

- Rango de valores, por ejemplo 0-65 o A-M.
- Especificar la lista de valores, por ejemplo 4, 6, 8, 32 o Y, N.
- Una serie de rangos, por ejemplo 12-16, 20-15, 30-38. **[6]**

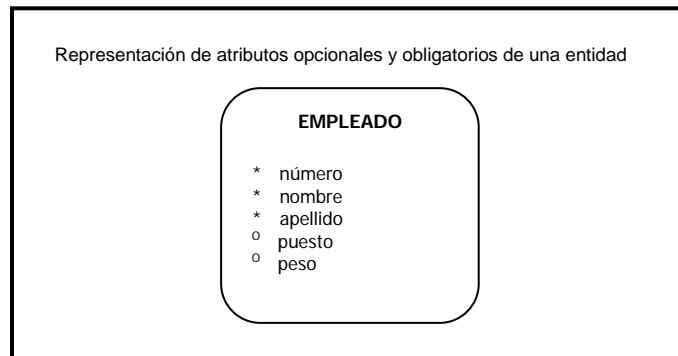


Figura 2.4. Ejemplo de la Entidad EMPLEADO y sus atributos.

2.4.1.3. Identificadores Únicos.

Cada entidad debe ser identificada, un identificador único (UID) es una combinación de atributos y relaciones que sirven para identificar una instancia de una entidad. **[6]**

Dentro de los conjuntos de entidades existen los siguientes tipos de identificadores únicos:

- *Superclave*: Es un subconjunto de atributos que permite distinguir unívocamente cada una de las entidades de un conjunto de entidades.
- *Clave candidata*: Dada una superclave, si ésta deja de serlo removiendo únicamente uno de los atributos que la componen, entonces ésta es una clave candidata.
- *Clave primaria*: Es una clave candidata, elegida por el diseñador de la base de datos, para identificar unívocamente las entidades en un conjunto de entidades.

[17]

Los identificadores únicos se representan anteponiendo el signo “#” como indica la figura siguiente: [6]

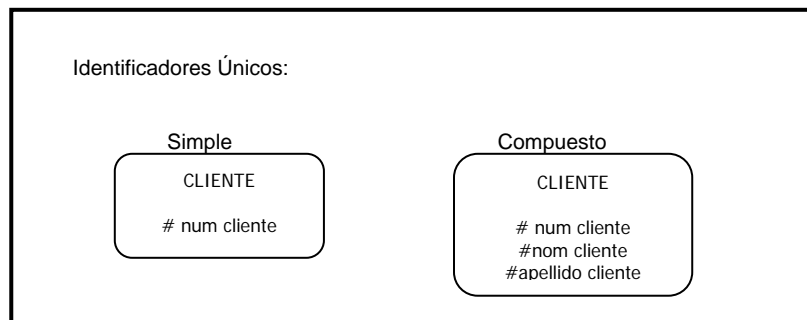


Fig. 2.5. Identificadores Únicos

2.4.2. Normalización

La teoría de la normalización es una ayuda que proporciona un procedimiento riguroso para el diseño de bases de datos. Una base de datos mal diseñada puede funcionar inicialmente pero puede mostrar anomalías en el almacenamiento debidas al agrupamiento indiscriminado de los campos cuando se efectúan en los archivos las operaciones de inserción, actualización o eliminación. La teoría de la

normalización ayuda a reconocer las cualidades no deseadas en un archivo y la forma de corregirlas. [6]

Para realizar el proceso de normalización, se debe determinar la forma normal de una relación a partir de su relación de atributos conocida como “dependencia funcional”. [6]

2.4.2.1. Dependencia Funcional

El análisis de una relación de dependencia funcional entre los campos de una relación permite clasificar la relación en una de las cuatro formas normales. [6]

El concepto de dependencia funcional (DF), se tomó de las matemáticas elementales. Se dice que Y es función de X, $Y = f(X)$, si el valor de Y está siempre determinado por el valor de X. [6]

Si se aplica la misma terminología a una relación, la dependencia funcional entre los atributos A y B en una relación se define como sigue: El atributo A es funcionalmente dependiente del atributo B si el valor de A esta determinado por el valor de B. [6]

A partir de la dependencia funcional se determina la forma normal en que se encuentra la relación, para que en acto seguido se pueda realizar el proceso de normalización.

2.4.2.2. Formas Normales

El proceso de normalización empieza con la combinación de todos los datos de la base en una relación, la que a su vez se descompone en dos o más relaciones más pequeñas. Se

efectúan descomposiciones sucesivas de las relaciones intermedias aplicando las formas normales hasta que todas las relaciones obtenidas pertenecen a la cuarta forma normal (4FN).

[6]

Primera Forma Normal (1FN):

Una relación está en la primera forma normal (o 1FN) si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos. El dominio de un campo es el rango de valores permitidos para el campo. **[6]**

Segunda Forma Normal (2FN):

Una relación es o pertenece a la segunda forma normal si es 1FN y cada atributo no-clave de la relación es total y funcionalmente dependiente de su clave principal. **[6]**

Tercera Forma Normal (3FN):

Una relación es 3FN, si es 2FN y ningún atributo no-clave en la relación es funcionalmente dependiente de algún otro atributo no-clave. **[6]**

Forma normal Boyce/Codd (BFCN)

Una relación es BFCN si cada determinante en la relación es una clave aspirante. **[6]**

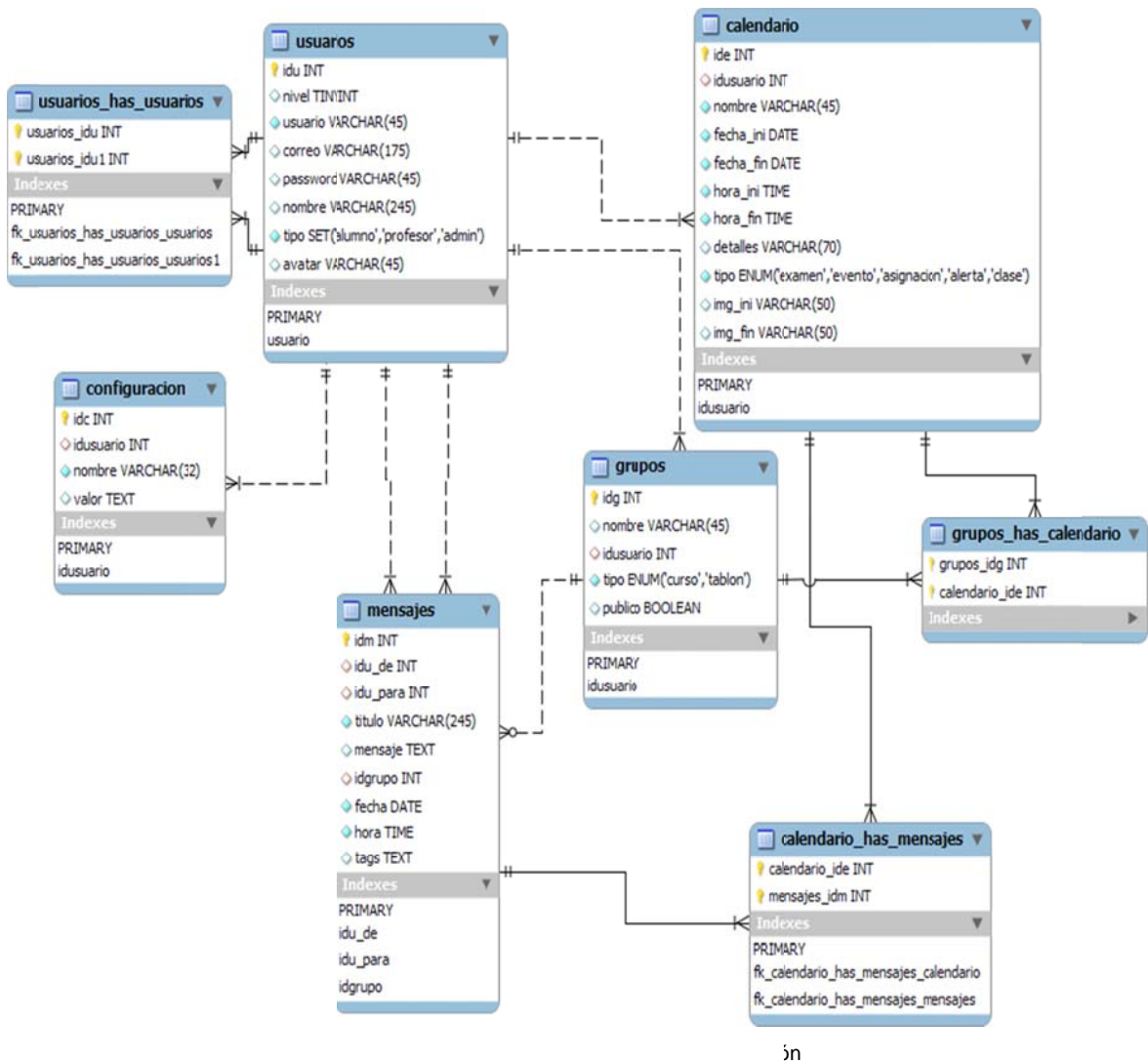
Cuarta Forma Normal (4FN):

Una relación es en 4FN (cuarta forma normal) si es BFCN y no contiene dependencias multivalores. **[6]**

La meta final del proceso de normalización es la agrupación de todos los atributos (o campos) de una base de datos en relaciones adecuadas para que la base se pueda almacenar con el mínimo de datos redundantes. **[6]**

Aplicar los dos métodos mencionados (Modelo E-R y normalización) para el diseño de la Base de Datos nos permite obtener un diagrama, que representa un lenguaje gráfico para describir conceptos detallados del modelo E-R, el cuál muestra gráficamente un diseño general de nuestra Base de Datos (BD).

En función de la herramienta de diseño que se emplee un diagrama entidad-relación muestra una forma similar a la siguiente:



2.5. Plataforma de Implementación

La elección de la plataforma de implementación del sistema, motivo de este trabajo, fue debido a las características que ofrece la misma y a los requerimientos planteados para su funcionalidad.

Dichos requerimientos están basados en la necesidad de desarrollar una aplicación con tecnología de mayor actualidad, que brinde facilidad en su desarrollo, implantación, mantenimiento y actualización.

El sistema para control escolar fue pensado para funcionar en un ambiente WEB, ya que los sistemas o aplicaciones WEB cuentan con un gran número de ventajas en comparación con las aplicaciones de escritorio e incluso con las cliente-servidor.

Algunas de sus ventajas se listan a continuación:

WEB vs Escritorio. [18]

Característica	WEB	Escritorio
Accesibilidad y cobertura	Se puede acceder desde cualquier lugar con acceso a Internet	Solo en el computador donde se haya instalado previamente el software
Infraestructura y movilidad	Solo se tiene que conectar a la Internet	Está restringido a la ubicación del computador local.
Personalización, actualización y soporte	Es suficiente con realizar los cambios en el servidor WEB	Hay que realizarlos en cada estación de trabajo (PC) donde se tenga la aplicación
Capacidad de usuarios concurrentes	Alta debido a la arquitectura de clientes livianos que la pueden usar	Baja ya que la forma de diseño es centrada en un único usuario local

Tabla 2.1. Comparación Aplicaciones WEB vs Escritorio

WEB vs Cliente-Servidor [19]

Característica	WEB	Escritorio
Costos del sistema	Se ofrecen como aplicaciones “llave en mano” hechas a la medida, permiten iniciar pequeños y crecer sin desembolsar grandes cantidades	Se pagan por completo y los costos iniciales pueden ser altos, riesgo alto si no cumple las necesidades.
Ambiente	El hardware y software necesarios están incluidos en el precio de la aplicación	Costos extras de mantenimiento y software recaen en el cliente.
Licenciamiento	El uso concurrente depende de la configuración del servidor y el ancho de banda de internet.	Cada licencia de usuario debe pagarse, y puede existir dispendio de recursos económicos
Acceso al sistema	Conexión desde cualquier parte con un navegador e internet	Conexión a través de la red local, mayor gasto si se requiere acceso remoto.

Tabla 2.2. Comparación Aplicaciones WEB vs Cliente-Servidor

2.5.1. WAMP

WampServer es un entorno de desarrollo WEB de Windows. Permite crear aplicaciones web con Apache 2, PHP y bases de datos MySQL. Además, PhpMyAdmin permite administrar fácilmente las bases de datos. [20]

2.5.1.1. Windows

Windows es el nombre de una familia de sistemas operativos desarrollados y vendidos por Microsoft. [21]

Un **sistema operativo**, es un conjunto de programas que posibilita la administración de los recursos de una **computadora**.

Este tipo de sistemas empieza a trabajar cuando se enciende el equipo para gestionar el **hardware** a partir desde los niveles más básicos. [22]

2.5.1.2. Apache

Es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. **[23]**

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero no cuenta con interfaz gráfica que ayude en su configuración. **[23]**

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo. **[23]**

Ventajas:

- Modular
- Código abierto
- Multi-plataforma
- Extensible
- Popular (fácil conseguir ayuda/suporte)

Módulos

La arquitectura del servidor Apache es muy modular. El servidor consta de un núcleo y diversos módulos que aportan mucha de la funcionalidad que podría considerarse básica para un servidor web. **[23]**

Uso

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web. [23]

Además es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición en una forma segura y confiable. Un ejemplo es al momento de compartir archivos desde una computadora personal hacia Internet. [23]

Configuración

La mayor parte de la configuración se realiza en el fichero apache2.conf o httpd.conf, según el sistema donde esté corriendo. Cualquier cambio en este archivo requiere reiniciar el servidor, o forzar la lectura de los archivos de configuración nuevamente. [23]

Licencia

La licencia de software bajo la cual el software de la fundación Apache es distribuido es una parte distintiva de la historia de Apache HTTP Server y de la comunidad de código abierto. La Licencia Apache permite la distribución de derivados de código abierto y cerrado a partir de su código fuente original. [23]

Además de ser una aplicación con buena reputación, tiene la ventaja de que su licenciamiento es de código abierto, que si

bien no es significado de gratuito, (aunque en la mayoría de casos lo es) puede ser adquirido a un bajo costo.

2.5.1.3. MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual. [24]

Aplicaciones

MySQL es muy utilizado en aplicaciones web, en plataformas (Linux/Windows). Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. [24]

MySQL es una base de datos muy rápida en la lectura, pero puede provocar problemas de integridad en entornos de alta concurrencia. En aplicaciones web hay baja concurrencia en la 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. [24]

Características

Entre las características disponibles en las últimas versiones se puede destacar:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.

- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones..
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.
- Permite escoger entre múltiples motores de almacenamiento para cada tabla.
- Permite agrupación de transacciones. [24]

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. [24]

Licencia

MySQL se distribuye bajo la licencia GNU GPL que obliga a que la distribución de cualquier producto derivado (aplicación) se haga bajo esa misma licencia. [24]

De las características mencionadas para MySQL, es notable el hecho de que es ideal para realizar aplicaciones WEB, así como su naturaleza de ser un administrador de BD relacional, características que son requeridas por el sistema a desarrollar.

2.5.1.4. PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. **[25]**

Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. **[25]**

PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. **[25]**

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. **[25]**

Características

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender.
- El código fuente escrito en PHP es invisible al navegador web y al cliente.
- Programación segura y confiable.

- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Posee una amplia documentación en su sitio web oficial.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones (desde PHP5). [25]

Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. [25]

Inconvenientes

- Como es un lenguaje que se interpreta en ejecución, para ciertos usos puede resultar un inconveniente que el código fuente no pueda ser ocultado.
- Debido a que es un lenguaje interpretado, un script en PHP suele funcionar considerablemente más lento que su equivalente en un lenguaje de bajo nivel.
- Las variables al no ser tipadas dificulta a los diferentes IDEs para ofrecer asistencias para el tipeado del código. [25]

De las particularidades mencionadas con respecto a PHP, hay que destacar la de ser un lenguaje de programación para desarrollo WEB, con código interpretado por el servidor, además

de ser fácil de aprender y que no tiene costo alguno. Características más que deseables para el proyecto a elaborar.

De lo anterior deducimos que la plataforma elegida para el desarrollo de nuestra aplicación, cuenta con las herramientas necesarias para implementar las características funcionales deseadas para el sistema, además de ser de código abierto, lo cual satisface el aspecto económico del mismo.

Por lo que se considera una elección adecuada para cumplir con los requerimientos del sistema, apartado que comentaremos en el siguiente capítulo.

CAPITULO III

ANÁLISIS Y DISEÑO

Tomando en consideración que la metodología aplicada para el ciclo de vida de desarrollo de este sistema es el UP (Proceso Unificado), y de acuerdo a los objetivos planteados con anterioridad, procederemos a realizar el análisis y diseño del sistema. Para lo cual describiremos a continuación los requisitos funcionales y no funcionales.

3.1. Requisitos funcionales

El sistema debe incluir un control de acceso que permita que los usuarios ingresen con un nombre de usuario y una contraseña, los cuales serán entregados únicamente al personal administrativo y encargado de control escolar para su manejo y resguardo.

Cada semestre de ciclo escolar y durante transcurso del mismo, el sistema debe permitir registrar alumnos de nuevo ingreso con todos sus datos personales. De igual forma debe permitir buscar, modificar y dar de baja a alumnos que se encuentren inscritos, para los casos que causen baja el sistema únicamente deberá cambiar el estatus del alumno para que no aparezca en las operaciones diarias de control escolar, pero conservará el registro del mismo para futuras consultas, también debe permitir activar o cambiar su estatus de baja a activo en el caso de que se presente un reingreso del mismo.

Debe admitir el registro de profesores y personal administrativo con los datos personales y profesionales que se indican en los lineamientos. En el caso de personal docente deberá permitir la función de asignación a materias y grupos. De igual forma tendrá la opción de buscar, modificar y dar de baja a docentes y administrativos. En el caso de baja únicamente se debe

cambiar el estatus del personal pero se conservará el registro del mismo. Así también se debe poder realizar el cambio de estatus de la persona.

Debe permitir generar y modificar un número determinado de grados, semestres y grupos con que cuenta la Institución, para el grado se utilizara un número progresivo del uno al tres que determina los grados oficiales de un bachillerato, para el semestre se describirá con números ordinales escrito con letra del primero al sexto y para el grupo se asignará una letra del alfabeto en mayúscula y con comillas dobles iniciado en la letra "A" y como límite la letra "Z". Así mismo se debe generar y registrar el ciclo escolar que está en curso, que estará indicado en forma de número por el año que inicia y termina dicho curso.

El sistema debe permitir generar y modificar el catálogo de materias. Cada materia debe permitir el registro de su aprobación. También se debe poder registrar el porcentaje de asistencias que tiene un alumno a dicha materia, el cual será calculado a partir del número de asistencias establecidas para cada una de ellas.

El bachillerato tiene un horario de atención de 8:00 a 16:00hrs. y los alumnos toman clase de 8:00 a 13:30hrs. por lo que el sistema debe permitir generar y modificar el horario oficial que se encuentra indicado en los lineamientos emitidos por la Secretaría. Dicho horario será de manera semanal y se deben poder asignar y modificar materias, grados y grupos para los alumnos y maestros. Se deberá verificar que las materias, grados y grupos asignados a un día y hora específica no se repitan o se traslapen.

El personal de control escolar asignará un grado y grupo a cada alumno, de conformidad a los criterios que establezcan, en todo caso el criterio final podrá ser de manera aleatoria pero verificando la equidad en el número y género de alumnos por salón.

Así también el personal de control escolar asignará a cada personal docente las materias que le corresponden impartir, así como la hora en que se impartirá la misma.

El sistema debe permitir registrar la calificación y el número de asistencias del alumno en el periodo de estudio llamado Momento. También deberá permitir modificar o continuar una captura parcial, así como la consulta de las mismas. Únicamente el personal de control escolar y previa autorización del Director podrá capturar y realizar modificaciones en el sistema.

El sistema deberá permitir la generación, consulta e impresión de la documentación oficial para el alumno, específicamente los documentos siguientes: boletas de los alumnos las cuales podrán generarse por alumno o por grupo de alumnos, así mismo los formatos de certificado, los cuales mostrarán la información concerniente a este documento y también permitirá generar las listas de grupos, para control de asistencias de los alumnos.

3.2. Requisitos no funcionales

Algunas de las características que también son importantes a considerar en el diseño del sistema son las siguientes:

Disponibilidad: Debido a que el sistema únicamente funcionará en la intranet de la institución será suficiente con el hecho de que tenga una disponibilidad aceptable.

Seguridad: La seguridad es únicamente a nivel de autenticación del usuario y encriptación de clave mediante algoritmo MD5.

Accesibilidad: Está permitida únicamente a personal de Control Escolar.

Portabilidad: El sistema se debe diseñar para la WEB, facilitando su portabilidad hacia cualquier plataforma, además en la implementación del código fuente utilizará el paradigma orientado a objetos lo que permitirá ser reutilizado.

Escalabilidad: En su diseño se debe contemplar el crecimiento y adaptación de nuevas funcionalidades.

Operatividad: el sistema debe funcionar en una PC con un solo procesador, que pueda albergar un servidor de páginas WEB y al mismo tiempo pueda ser cliente.

Interfaz: la interacción con el usuario será por medio de páginas WEB, las cuales muestran menús para elegir las operaciones a realizar, así como presentar formularios de captura de información y botones para accionar los comandos.

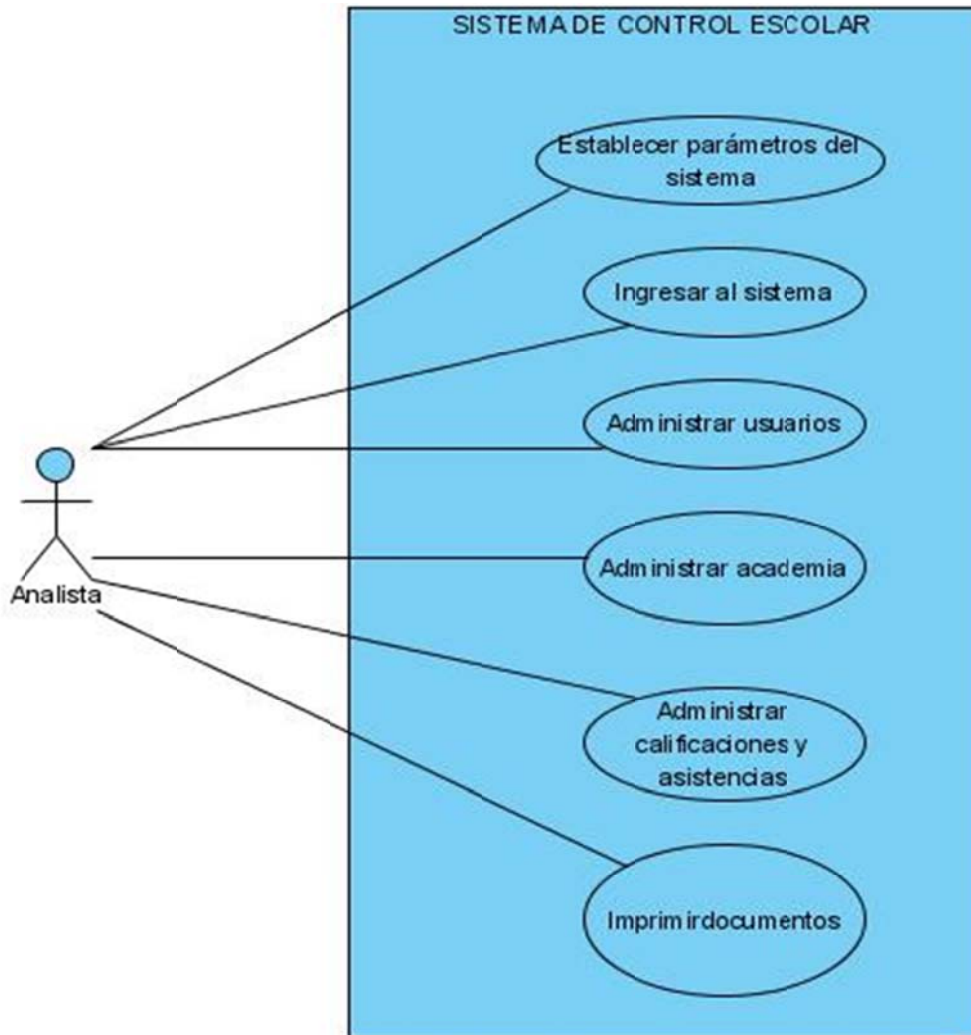
Costo: en su diseño, implementación y despliegue, se consideran las tecnologías que ofrezcan un equilibrio aceptable entre costo-beneficio, privilegiando las plataformas de código abierto.

3.3. Análisis del sistema

Para iniciar con esta etapa, partimos de la especificación del sistema descrita en los requerimientos funcionales y no funcionales, y definimos el comportamiento mediante el Modelo de Casos de Uso.

3.3.1. Modelo de Casos de Uso

3.3.1.1. Diagrama de casos de uso



s

La figura anterior muestra el diagrama general de casos de uso del sistema de control escolar, cada uno de los casos de uso muestra el comportamiento del sistema, es decir funcionalidad que realiza el sistema ante estímulos de actores externos.

3.3.1.2. Especificación de casos de uso

Un caso de uso se debe documentar para describir el propósito que tiene y el flujo de eventos que realiza, así como para tener en cuenta las pre/post condiciones que se deben cumplir para su funcionamiento, a continuación se lista la documentación de los casos de uso del diagrama anterior: [5]

1. Nombre:	Establecer parámetros del sistema
2. Autor:	Guillermo Pérez Arrieta
3. Fecha:	28-mayo-2012
4. Descripción:	En este caso de uso, el sistema le permite al analista crear y modificar los grados, grupos, semestres, ciclos escolares, materias, cursos y horarios de acuerdo con las necesidades académicas de la Institución.

5. Actores:

5.1. Analista

6. Pre-condiciones:

6.1. El Analista debe haber ingresado al sistema.

7. Flujo básico:

7.1. El Analista selecciona la función “**Establecer parámetros**”.

7.2. El sistema muestra al Analista un menú **Parámetros** con las opciones: **Grados, Grupos, Semestres, Ciclos Escolares, Materias, Cursos y Horarios**.

7.3. El Analista escoge la opción “**Grados**”. Si se escoge la opción:

- “**Grupos**” se realiza el punto 8.1.
- “**Semestres**” se realiza el punto 8.2.
- “**Ciclos Escolares**” se realiza el punto 8.3.
- “**Materias**” se realiza el punto 8.4.
- “**Cursos**” se realiza el punto 8.5.
- “**Horarios**” se realiza el punto 8.6.

7.4. El sistema muestra al Analista las opciones de funciones: **Crear, Modificar, Eliminar, Buscar o Salir**.

7.5. Si se escoge la opción **Crear**:

- “**Modificar**” se realiza el punto 8.7.
- “**Buscar**” se realiza el punto 8.8.
- “**Eliminar**” se realiza el punto 8.9.
- “**Salir**” se realiza el punto 8.10.

7.6. El sistema muestra una pantalla con formulario para ingresar los datos.

7.7. El Analista ingresa los datos en el formulario.

7.8. El Analista envía los datos al sistema.

7.9. El sistema guarda la información y responde mostrando los resultados al Analista.

7.10. Termina el caso de uso.

8. Flujo alternativo:

8.1. Grupos.

8.1.1. El Analista escoge la opción “**Grupos**”.

8.1.2. Continúa el flujo en el punto 7.4. del flujo básico.

8.2. Semestres.

8.2.1. El Analista escoge la opción “**Semestres**”.

8.2.2. Continúa el flujo en el punto 7.4. del flujo básico.

8.3. Ciclos Escolares.

8.3.1. El Analista escoge la opción “**Ciclos Escolares**”.

8.3.2. Continúa el flujo en el punto 7.4. del flujo básico.

8.4. Materias.

8.4.1. El Analista escoge la opción “**Materias**”.

8.4.2. Continúa el flujo en el punto 7.4. del flujo básico.

8.5. Cursos.

8.5.1. El Analista escoge la opción “**Cursos**”.

8.5.2. Continúa el flujo en el punto 7.4. del flujo básico.

8.6. Horarios.

8.6.1. El Analista escoge la opción “**Horarios**”.

8.6.2. Continúa el flujo en el punto 7.4. del flujo básico.

8.7. Modificar.

8.7.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de la información en que se está interesado.

8.7.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

8.7.3. El sistema muestra al Analista una pantalla con formulario que contiene los resultados de la búsqueda para que pueda modificarlos.

8.7.4. El Analista modifica cualquier dato del formulario.

8.7.5. Continúa el flujo en el punto **7.1.9.** del flujo básico.

8.8. Buscar.

8.8.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de la información en que se está interesado.

8.8.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

8.8.3. El sistema muestra al Analista una pantalla con los resultados de la búsqueda para que pueda revisarlos.

8.8.4. Continúa el flujo en el punto **7.10.** del flujo básico.

8.9. Eliminar.

8.9.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de la información en que se está interesado.

8.9.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

8.9.3. El sistema muestra al Analista una pantalla con la lista de resultados de la búsqueda para que pueda seleccionarlos.

8.9.4. El Analista selecciona datos de los resultados.

8.9.5. Continúa el flujo en el punto **7.1.9.** del flujo básico.

8.10. Salir

8.10.1. El sistema termina la función y muestra la pantalla inicial.

9. Flujo de excepción:

9.1.E1. El sistema no puede guardar la información porque no hay conexión a la BD.

9.2.E2. El sistema no puede guardar la información porque ya existe.

9.3.E3. El sistema no puede guardar porque no se ingresó ninguna información.

9.4.E4. No existe información con los criterios de búsqueda indicados.

10. Post-condiciones:

1. Nombre:	Administrar usuarios
2. Autor	Guillermo Pérez Arrieta
3. Fecha	06-junio-2012
4. Descripción:	En este caso de uso, el sistema permite agregar, modificar o eliminar usuarios que harán uso del sistema.

5. Actores:

5.1. Analista

6. Pre-condiciones:

6.1. El Analista debe haber ingresado al sistema.

7. Flujo básico:

7.1. El Analista inicia el caso de uso “**Administrar usuarios**”.

7.2. El sistema muestra al Analista una página con las opciones:
Agregar, Modificar, Eliminar usuario o **Salir**.

7.3. El Analista selecciona “**Agregar usuario**” como flujo básico. Si la acción a realizar es:

- “**Modificar**” se realiza el punto **8.1**.
- “**Buscar**” se realiza el punto **8.2**.
- “**Eliminar**” se realiza el punto **8.3**.
- “**Salir**” se realiza el punto **8.4**.

7.4. El sistema muestra una pantalla con el formulario vacío donde se registran los datos del nuevo usuario.

7.5. El Analista ingresa los datos del usuario.

7.6. El Analista envía la información al sistema.

7.7. El sistema guarda los datos y muestra los resultados al Analista.

7.8. Termina el caso de uso.

8. Flujo alternativo:

8.1. Modificar

- 8.1.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de los usuarios en que se está interesado.
- 8.1.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.
- 8.1.3. El sistema muestra al Analista una pantalla con listado de usuarios para que pueda seleccionarlos.
- 8.1.4. El Analista selecciona el usuario a modificar.
- 8.1.5. El sistema muestra una pantalla con formulario del usuario a modificar.
- 8.1.6. El Analista modifica sus datos.
- 8.1.7. El flujo continúa en el punto 7.6. del flujo básico.

8.2. Buscar.

- 8.2.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de la información en que se está interesado.
- 8.2.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.
- 8.2.3. El sistema muestra al Analista una pantalla con los resultados de la búsqueda para que pueda revisarlos.
- 8.2.4. Continúa el flujo en el punto 7.8. del flujo básico.

8.3. Eliminar usuario

- 8.3.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de los usuarios en que se está interesado.
- 8.3.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.
- 8.3.3. El sistema muestra al Analista una pantalla con listado de usuarios para que pueda seleccionarlos.
- 8.3.4. El Analista selecciona el usuario a eliminar.
- 8.3.5. El flujo continúa en el punto 7.6. del flujo básico.

8.4. Salir

- 8.4.1. El sistema termina la función y muestra la pantalla inicial.

9. Flujo de excepción:

9.1.E1. El sistema no puede guardar la información porque no hay conexión a la BD.

9.2.E2. El sistema no puede guardar la información porque ya existe.

9.3.E3. El sistema no puede guardar porque no se ingresó ninguna información.

9.4.E4. No existe información con los criterios de búsqueda indicados.

10. Post-condiciones:

1. Nombre:	Administrar academia
2. Autor	Guillermo Pérez Arrieta
3. Fecha	06-junio-2012
4. Descripción:	En este caso de uso, el sistema permite agregar, modificar, buscar o dar de baja docentes y personal administrativo en el sistema.

5. Actores:

5.1. Analista

6. Pre-condiciones:

6.1. El analista debe haber ingresado al sistema.

7. Flujo básico:

7.1. El Analista inicia el caso de uso "**Administrar academia**".

7.2. El sistema muestra al Analista un menú con los tipos de personal: "**Docentes**" o "**Personal Administrativo**".

7.3. El Analista escoge la opción "**Docentes**". Si se escoge la opción:

- "**Personal Administrativo**" se realiza el punto **8.1**.

7.3.1. El sistema muestra al Analista las opciones de funciones: **Crear, Modificar, Eliminar, Buscar** o **Salir**.

7.3.2. El Analista selecciona "**Crear**". Si se escoge la opción:

- "**Modificar**" se realiza el punto **8.2**.
- "**Buscar**" se realiza el punto **8.3**.
- "**Dar de baja**" se realiza el punto **8.4**.
- "**Salir**" se realiza el punto **8.5**.

7.4. El sistema muestra una pantalla con el formulario vacío donde se registran los datos de la persona.

7.5. El Analista ingresa los datos.

7.6. El Analista envía la información al sistema.

7.7. El sistema guarda los datos y muestra los resultados al Analista.

7.8. Termina el caso de uso.

8. Flujo alternativo:

8.1. Personal Administrativo

8.1.1. El Analista escoge la opción “**Personal Administrativo**”.

8.1.2. Continúa el flujo en el punto 7.4. del flujo básico.

8.2. Modificar

8.2.1. El sistema muestra una pantalla con formulario para realizar la búsqueda del personal en que se está interesado.

8.2.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

8.2.3. El sistema muestra al Analista una pantalla con listado de personal para que pueda seleccionarlos.

8.2.4. El Analista selecciona el personal a modificar.

8.2.5. El sistema muestra una pantalla con formulario del personal a modificar.

8.2.6. El Analista modifica los datos.

8.2.7. El flujo continúa en el punto 7.6. del flujo básico.

8.3. Buscar.

8.3.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de la información en que se está interesado.

8.3.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

8.3.3. El sistema muestra al Analista una pantalla con los resultados de la búsqueda para que pueda revisarlos.

8.3.4. Continúa el flujo en el punto 7.8. del flujo básico.

8.4. Dar de baja

8.4.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de los usuarios en que se está interesado.

8.4.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

8.4.3. El sistema muestra al Analista una pantalla con listado de personal para que pueda seleccionarlos.

8.4.4. El Analista selecciona el personal a eliminar.

8.4.5. El flujo continúa en el punto **7.6.** del flujo básico.

8.5. Salir

8.5.1. El sistema termina la función y muestra la pantalla inicial.

9. Flujo de excepción:

9.1.E1. El sistema no puede guardar la información porque no hay conexión a la BD.

9.2.E2. El sistema no puede guardar la información porque ya existe.

9.3.E3. El sistema no puede guardar porque no se ingresó ninguna información.

9.4.E4. No existe información con los criterios de búsqueda indicados.

10. Post-condiciones:

1. Nombre:	Ingresar al sistema
2. Autor:	Guillermo Pérez Arrieta
3. Fecha:	06-junio-2012
4. Descripción:	En este caso de uso, el sistema permite los usuarios ingresar al sistema con un nombre de usuario y una contraseña o password.

5. Actores:

5.1. Analista

6. Pre-condiciones:

7. Flujo básico:

7.1. El usuario solicita la ejecución del caso de uso.

7.2. El sistema muestra una página con los campos nombre de usuario y contraseña para que ingrese estos datos.

7.3. El usuario ingresa su nombre de usuario y su contraseña.

7.4. El usuario envía al sistema los datos capturados.

7.5. El sistema recibe los datos y valida:

7.5.1. Que el usuario exista en el sistema.

7.5.2. Que la contraseña sea correcta.

7.6. El sistema da ingreso al usuario mostrando la pantalla inicial.

Termina el caso de uso

8. Flujo alternativo:

9. Flujo de excepción:

9.1. E1. El usuario no existe, el sistema muestra un mensaje indicando al usuario que el nombre de usuario no es correcto

9.2. E2. La contraseña no es correcta, El sistema muestra una ventana con mensaje indicando al usuario que la contraseña está equivocada.

9.3. E4. No existe información con los criterios de búsqueda indicados.

10. Post-condiciones:

1. Nombre:	Administrar calificaciones y asistencias
2. Autor	Guillermo Pérez Arrieta
3. Fecha	07-junio-2012
4. Descripción:	En este caso de uso, el sistema permite agregar o modificar calificaciones y asistencias de los diferentes momentos y materias del ciclo escolar.

5. Actores:

5.1. Analista

6. Pre-condiciones:

6.1. El Analista debe haber ingresado al sistema.

7. Flujo básico:

- 7.1. El analista inicia el caso de uso **“Administrar calificaciones y asistencias”**.
 - 7.2. El sistema muestra un menú al Analista con las opciones: **“Calificaciones”** o **“Asistencias”**.
 - 7.3. El Analista escoge la opción **“Calificaciones”**. Si se escoge la opción:
 - **“Asistencias”** se realiza el punto **8.1**.
 - 7.4. El sistema muestra al Analista las opciones de funciones: **Agregar, Modificar, Buscar o Salir**.
 - 7.5. El Analista selecciona **“Agregar”**. Si se escoge la opción:
 - **“Modificar”** se realiza el punto **8.2**.
 - **“Buscar”** se realiza el punto **8.3**.
 - **“Salir”** se realiza el punto **8.4**.
 - 7.6. El sistema muestra una pantalla con formulario para realizar la búsqueda de la información en que se está interesado.
 - 7.7. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.
 - 7.8. El sistema muestra una pantalla con el curso y formulario vacío de para agregar calificaciones.
 - 7.9. El Analista ingresa las calificaciones.
 - 7.10. El Analista envía los datos al sistema.
 - 7.11. El sistema guarda los datos y muestra los resultados al Analista.
 - 7.12. Termina el caso de uso.
- 8. Flujo alternativo:**
- 8.1. Asistencias**
 - 8.1.1. El Analista escoge la opción **“Asistencias”**.
 - 8.1.2. Continúa el flujo en el punto **7.4**. del flujo básico.
 - 8.2. Modificar**
 - 8.2.1. El sistema muestra una pantalla con formulario para realizar la búsqueda del curso en que se está interesado.

8.2.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

8.2.3. El sistema muestra al Analista una pantalla con listado del curso para que pueda modificar los datos.

8.2.4. El Analista modifica los datos.

8.2.5. El flujo continúa en el punto **7.10.** del flujo básico.

8.3. Buscar.

8.3.1. El sistema muestra una pantalla con formulario para realizar la búsqueda de la información en que se está interesado.

8.3.2. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

8.3.3. El sistema muestra al Analista una pantalla con los resultados de la búsqueda para que pueda revisarlos.

8.3.4. Continúa el flujo en el punto **7.12.** del flujo básico.

8.4. Salir

8.4.1. El sistema termina la función y muestra la pantalla inicial.

9. Flujo de excepción:

9.1.E1. El sistema no puede guardar la información porque no hay conexión a la BD.

9.2.E2. El sistema no puede guardar la información porque ya existe.

9.3.E3. El sistema no puede guardar porque no se ingresó ninguna información.

9.4.E4. No existe información con los criterios de búsqueda ingresados.

10. Post-condiciones:

1. Nombre:	Imprimir documentos
2. Autor	Guillermo Pérez Arrieta
3. Fecha	07-junio-2012
4. Descripción:	En este caso de uso, el sistema permite imprimir boletas de calificaciones, listas de asistencia y certificados de los alumnos.

5. Actores:

5.1. Analista

6. Pre-condiciones:

6.1. El Analista debe haber ingresado al sistema.

7. Flujo básico:

7.1. El Analista inicia el caso de uso “**Imprimir Documentos**”.

7.2. El sistema muestra al Analista un menú con las opciones: **Boletas, Listas de asistencia o Certificados.**

7.3. El Analista escoge la opción “**Boletas**”. Si se escoge la opción:

- “**Listas**” se realiza el punto **8.1.**
- “**Certificados**” se realiza el punto **8.2.**

7.4. El sistema muestra al Analista las opciones de funciones: **Generar o Salir.**

7.5. El Analista selecciona “**Generar**”. Si se escoge la opción:

- “**Salir**” se realiza el punto **8.3.**

7.6. El sistema muestra una pantalla con formulario para realizar la búsqueda de la información en que se está interesado.

7.7. El Analista indica los criterios de búsqueda en el formulario y se acciona el comando.

7.8. El sistema muestra una pantalla con las boletas generadas en formato PDF.

7.9. El Analista indica al sistema que imprima el documento.

7.10. El sistema envía los datos a la impresora. Termina el caso de uso.

8. Flujo alternativo:

8.1. Listas

8.1.1. El Analista escoge la opción “**Listas**”.

8.1.2. Continúa el flujo en el punto **7.4.** del flujo básico.

8.2. Certificados

8.2.1. El Analista escoge la opción “**Certificados**”.

8.2.2. Continúa el flujo en el punto **7.4.** del flujo básico.

8.3. Salir

8.3.1. El sistema termina la función y muestra la pantalla inicial.

9. Flujo de excepción:

9.1.E1. El sistema no puede generar la información porque no hay conexión a la BD.

9.2.E4. No existe información con los criterios de búsqueda ingresados.

10. Post-condiciones:

3.3.1.3. Escenarios

Los escenarios definen situaciones en las que se encuentra un sistema, por tanto es una instanciación de un caso de uso, esto quiere decir que cada caso de uso tiene varios escenarios. **[5]**

El conjunto de escenarios, definen el sistema o lo que puede hacer. Por tanto los escenarios sirven para definir, que hace y que no hace un sistema. **[5]**

En todo sistema tenemos escenarios Primarios o de flujo básico y Secundarios: Alternos y de Excepción; y se deben elaborar tantos como sean necesarios para el entendimiento del problema. **[5]**

Tomando en cuenta lo anterior describiremos los escenarios más representativos para el sistema de control escolar.

a) Caso de uso: Establecer parámetros del sistema

Escenario 1.-

1. Melina ya realizó el caso de uso “**ingresar al sistema**”.
2. Melina inicia el proceso “**Establecer parámetros del sistema**” y en el menú selecciona la opción “**Cursos**”.
3. En la pantalla elige la opción “**Crear**”.

4. En la pantalla para crear el curso se muestra un formulario donde ingresa: **3º** para el “Grado”, la letra **“C”** para el Grupo, **“Sexto”** para el Semestre, **“2011-2012”** para el Ciclo escolar, **“Química”** para la Materia, **“Química Sexto C”** para Horario y envía la información al sistema para guardarla.
5. El sistema guarda la información y muestra la página con los datos de curso generado.
6. Termina el caso de uso

Escenario 2.-

1. Rocío ya realizó el caso de uso **“Ingresar al sistema”**.
2. Rocío inicia el proceso **“Establecer parámetros del sistema”** y en el menú selecciona la opción **“Ciclo Escolar”**.
3. En la pantalla elige la opción **“Modificar”**.
4. El sistema le muestra una pantalla con formulario para buscar del ciclo que se desea modificar.
5. Rocío elige la opción **“Todos”** y le envía el comando al sistema.
6. El sistema muestra todos los Ciclos Escolares que se encuentran registrados para que seleccione cual desea modificar
7. Rocío selecciona el Ciclo **2011-2012** y le indica al sistema que lo muestre.
8. El sistema muestra el formulario con los parámetros que se encuentran ingresados previamente.
9. Rocío modifica **2012-2013** para el **“Ciclo escolar”** y envía la información al sistema para guardarla.
10. El sistema guarda la información y muestra una pantalla con el resultado de la modificación.
11. Termina el caso de uso

Escenario 3.-

1. Guillermo realizó el caso de uso **“Ingresar al sistema”**.
2. Inicia el proceso **“Establecer parámetros del sistema”** y en el menú selecciona la opción **“Materias”**.

3. En la pantalla elige la opción **“Crear”**.
4. En pantalla de captura se muestra el formulario donde ingresa el código y el nombre de la materia siguiente: **“100 Álgebra”**, y envía los datos al sistema.
5. El sistema guarda la información y muestra una pantalla con el resultado.
6. Termina el caso de uso

Escenario 4.-

1. Fernando ya realizo el caso de uso **“Ingresar al sistema”**.
2. Inicia el proceso **“Establecer parámetros del sistema”** y en el menú selecciona la opción **“Semestres”**.
3. En la pantalla elige la opción **“Eliminar”**.
4. El sistema le muestra una pantalla con formulario para buscar del Semestre que se desea eliminar.
5. Fernando elije la opción **“Todos”** y le envía el comando al sistema.
6. El sistema muestra todos los Semestres que se encuentran registrados para que seleccione cual desea eliminar
7. Fernando selecciona el **“Sexto”** Semestre y le indica al sistema que lo elimine.
8. El sistema guarda la información y muestra una pantalla con el resultado.
9. Termina el caso de uso

Escenario 5.-

1. Leticia realizo el caso de uso **“Ingresar al sistema”**.
2. Inicia el proceso **“Establecer parámetros del sistema”** y en el menú selecciona la opción **“Horarios”**.
3. En la pantalla elige la opción **“Modificar”**.
4. El sistema le muestra una pantalla con formulario para buscar el horario que se desea modificar.
5. Leticia elije la opción **“Todos”** y le envía el comando al sistema.

6. El sistema muestra todos los Horarios que se encuentran registrados para que seleccione cual desea modificar.
7. Leticia selecciona el horario **Ingles I** y le indica al sistema que lo muestre.
8. El sistema muestra el formulario con los parámetros que se encuentran ingresados previamente.

Descripción	LunIni	LunFin	MarIni	MarFin	MieIni
Ingles I 2A	8:00	8:50	8:00	8:50	8:00
	MieFin	JueIni	JueFin	VieIni	VieFin
	8:50	8:00	8:50	8:00	8:50

9. En la pantalla horario Ingresa en el formulario:

Descripción	LunIni	LunFin	MarIni	MarFin	MieIni
Ingles I 2A	11:00	11:50	11:00	11:50	12:40
	MieFin	JueIni	JueFin	VieIni	VieFin
	13:30	12:40	13:30	11:00	11:50

y lo envía al sistema para almacenamiento.

10. El sistema guarda la información y muestra una pantalla con los resultados.
11. Termina el caso de uso

b) Caso de uso: Administrar usuarios

Escenario 1.-

1. Margarita realizo el caso de uso “Ingresar al sistema”.
2. Margarita selecciona el CU “**Administrar usuarios**” y selecciona la opción: “**Crear usuario**”.
3. El sistema muestra el formulario y Margarita ingresa: “**Nombre: Manuel Lima Jiménez, Rol: Analista, nombre de usuario: Mlima, Contraseña: Mlima**” y lo envía al sistema para almacenamiento.
4. El sistema guarda los datos y muestra los resultados.
5. Termina el caso de uso

Escenario 2.-

1. Alberto realizo el caso de uso “**Ingresar al sistema**”.

2. Alberto selecciona el CU “**Administrar usuarios**” y selecciona la opción: “**Eliminar usuario**”.
3. El sistema le muestra una pantalla con formulario para buscar el usuario que se desea eliminar.
4. Alberto elige la opción “**Todos**” y le envía la instrucción al sistema.
5. El sistema muestra todos los Usuarios que se encuentran registrados para que seleccione cual desea eliminar.
6. Alberto selecciona el usuario **mperez** y le indica al sistema que lo elimine.
7. El sistema elimina al usuario y muestra una pantalla con los resultados.
8. Termina el caso de uso.

c) Caso de uso: Administrar academia

Escenario 1.-

1. Roberto realizo el caso de uso “**Ingresar al sistema**”.
2. Roberto selecciona el CU “**Administrar academia**” y selecciona la opción: “**Alumnos**”
3. El sistema muestra la ventana de opciones y elige: “**Agregar**”.
4. En el formulario ingresa los datos siguientes: “**Nombre: Camilo, Apellidos: Arteaga Morales, CURP: ARMC781209HPLL09, NIA: 78945, Edad: 16, Sexo: H, Dirección: Av. Zaragoza 213, Status: Ingreso**” y envía los datos para ser guardados.
5. El sistema guarda los datos y muestra una pantalla con los resultados.
6. Termina el caso de uso.

Escenario 2.-

1. Omar realizo el caso de uso “**Ingresar al sistema**”.
2. Omar selecciona el CU “**Administrar academia**” y selecciona la opción: “**Docentes**”
3. En la ventana de opciones Omar selecciona: “**Dar de baja**”.

4. El sistema le muestra una pantalla con formulario para buscar el docente que se desea eliminar.
5. Omar elije la opción “**Todos**” y le envía la instrucción al sistema.
6. El sistema muestra todos los Docentes que se encuentran registrados para que seleccione cual desea dar de baja.
7. Omar selecciona docente Miguel Cervantes y le indica al sistema que lo ponga en bajas.
8. El sistema guarda los datos y muestra una pantalla con los resultados.
7. Termina el caso de uso.

d) Caso de uso: Ingresar al sistema

Escenario 1.-

1. Leonor inicia la función “**Ingresar al sistema**”.
2. En la ventana de login, ingresa “**Nombre de usuario: Lmorales y Contraseña: Lmorales**” y los envía al sistema para su validación.
8. El sistema determina que el usuario y la contraseña son válidos y muestra la pantalla de inicio del sistema.
9. Termina el caso de uso.

Escenario 2.-

1. Guadalupe inicia la función “**Ingresar al sistema**”.
2. En la ventana de login, ingresa “**Nombre de usuario: Gsosa y contraseña: Gsosa**” y los envía al sistema para su validación.
3. El sistema valida que el usuario no existe y muestra una ventana con el mensaje “**Usuario o contraseña no son válidos**”.
4. El sistema le muestra nuevamente la pantalla de login.
5. Termina el caso de uso.

e) Caso de uso: Administrar calificaciones y asistencias

Escenario 1.-

1. Gabriel realizo el caso de uso “**Ingresar al sistema**”.

2. Gabriel selecciona el CU **“Administrar calificaciones y asistencias”** y selecciona la opción: **“Calificaciones”** y la función **“Agregar”**.
3. El sistema le muestra una pantalla con formulario para buscar el curso en que se desea agregar calificaciones.
4. Gabriel elije la opción **“Todos”** y le envía la instrucción al sistema.
5. El sistema muestra todos los Cursos que se encuentran registrados para que seleccione en cual desea agregar datos.
6. Gabriel selecciona el curso **“Materia: Química I, Grado: 2, Grupo: B, Semestre: Segundo, Ciclo Escolar: 2012-2013”** y lo envía al sistema para que lo muestre.
7. El sistema muestra una pantalla con la lista de alumnos y momentos de la materia y asigna calificaciones a los siguientes alumnos:

n. Z.	NOMBRE DEL ALUMNO	CALIFICACIONES
1	AGUIRRE ROJAS ANAYELI	10
2	AMARO RAMIREZ YANELI	10
3	ATONAL HERNANDEZ YANETT	9
4	COSME CALDERON MIGUEL ANGEL	8
5	COSME CANTOR JUAN MANUEL	6
6	COSME CUEVAS JORGE	6
7	DIAZ TECPIL JOSE GUADALUPE	0
8	ESCAMILLA ROMERO CRISTIAN	0
9	ESPINDOLA AGUIRRE FLAVIO CESAR	6
10	ESPINDOLA ESCAMILLA EDUARDO	5

Y remite estos datos al sistema para almacenarlos.

8. El sistema guarda los datos y muestra una pantalla con los resultados.
9. Termina el caso de uso.

Escenario 2.-

1. Victoria realizó el caso de uso **“Ingresar al sistema”**.
2. Victoria selecciona el CU **“Administrar calificaciones y asistencias”** y selecciona la opción: **“Asistencias”** y la función **“Modificar”**.

3. El sistema le muestra una pantalla con formulario para buscar el curso en que se desea modificar asistencias.
4. Victoria elige la opción **“Todos”** y le envía la instrucción al sistema.
5. El sistema muestra todos los Cursos que se encuentran registrados para que seleccione en cual desea modificar datos.
6. El sistema muestra una ventana con la lista de materias y Victoria selecciona **“Materia: Razonamiento Verbal”, Grado: 2, Grupo: B, Semestre: Segundo, Ciclo Escolar: 2012-2013”**.
7. El sistema muestra una pantalla con la lista de alumnos y los datos previamente almacenados, Victoria modifica las asistencias en los siguientes alumnos:

N ^o	NOMBRE DEL ALUMNO	ASISTENCIAS
1	AGUIRRE ROJAS ANAYELI	23
2	AMARO RAMIREZ YANELI	22
3	ATONAL HERNANDEZ YANETT	21
4	COSME CALDERON MIGUEL ANGEL	23
5	COSME CANTOR JUAN MANUEL	23

Y remite estos datos al sistema para actualizarlos.

8. El sistema guarda los datos y muestra una pantalla con los resultados.
9. Termina el caso de uso.

f) Caso de uso: Imprimir documentos

Escenario 1.-

1. Ana realizo el caso de uso **“Ingresar al sistema”**.
2. Ana selecciona el CU **“Imprimir documentos”** y en el menú que muestra el sistema elije **“Boletas”**.
3. El sistema le muestra una pantalla con formulario para buscar el curso del que se desean las boletas.

4. Ana elige el curso **Materia: Razonamiento Verbal**, **Grado: 2, Grupo: B, Semestre: Segundo, Ciclo Escolar: 2012-2013**.y le envía la instrucción al sistema.
5. El sistema muestra una pantalla con la lista de boletas preparadas para imprimir.
6. Ana envía la impresión de boletas al sistema.
7. El sistema envía los datos a la impresora.
8. Termina el caso de uso

Escenario 2.-

1. Jorge realizo el caso de uso **“Ingresar al sistema”**.
2. Jorge selecciona el CU **“Imprimir documentos”** y en el menú que muestra el sistema elije **“Listas”**.
3. El sistema le muestra una pantalla con formulario para buscar el grupo del que se desean las listas.
4. Jorge elige el **Grado: 1, Grupo: C, Semestre: Tercero, Ciclo Escolar: 2012-2013**.y le envía la instrucción al sistema.
5. El sistema muestra una pantalla con la lista del grupo preparada para imprimir.
6. El sistema no puede enviar los datos a la impresora.
7. Jorge envía la impresión de listas al sistema.
8. El sistema muestra una ventana con mensaje **“No se puede enviar el documento a la impresora, verifique su conexión”** para su aceptación.
9. Jorge acepta el mensaje.
10. Termina el caso de uso.

3.3.2. Modelo conceptual

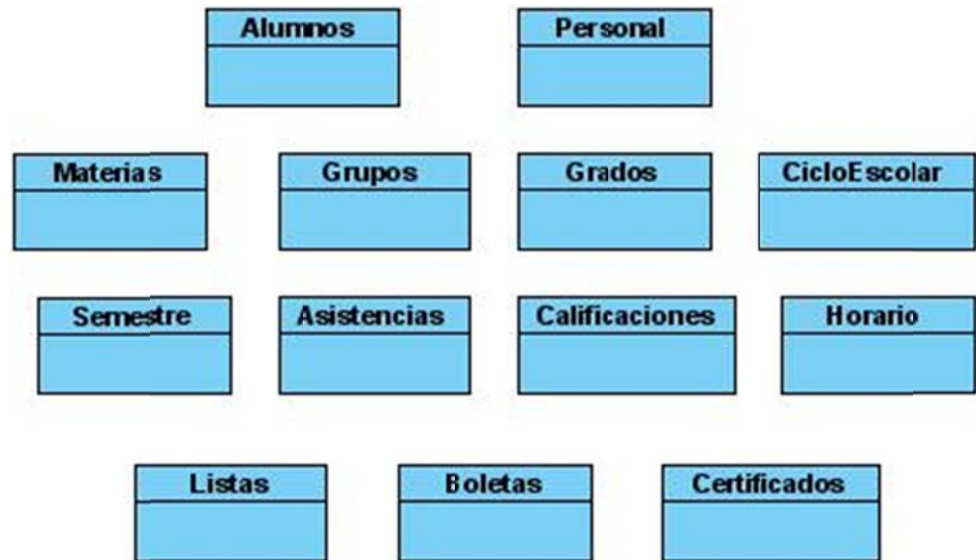
3.3.2.1. Diagrama de clases

En este punto del análisis es posible aplicar los conceptos del paradigma orientada a objetos, que nos permite detectar los grupos de objetos y abstraerlos en clases.

De la descripción de los requerimientos del sistema, planteados en los apartados anteriores, es posible empezar a definir la estructura, iniciamos examinando la descripción de los requerimientos descritos y construimos un primer modelo para identificar las clases que integran el sistema.

Así mismo, describimos el propósito de cada una de las clases en el diccionario del modelo, que define la semántica de las mismas.

La siguiente figura muestra un primer diagrama de clases de nuestro sistema.



ceptual)

3.3.2.2. Diccionario del modelo

El diccionario del modelo que describe a las clases del diagrama anterior es el siguiente:

Nombre: Alumno

Descripción: Información acerca de una persona registrada para realizar diversas actividades en el Bachillerato (generalmente recibir clases), con el fin de completar los cursos que forman parte del plan de estudios.

Nombre: Personal

Descripción: Información acerca de una persona registrada para realizar diversas actividades en el Bachillerato, con el fin de atender las necesidades de los alumnos, puede ser docente (impartir clases) o administrativo (actividades de control escolar).

Nombre: Semestre

Descripción: Es un periodo de tiempo oficial establecido con fechas de inicio y fin, el objetivo es delimitar el plan de estudios y las actividades realizadas para la evaluación del desempeño de los alumnos.

Nombre: Materias

Descripción: Información que describe las disciplinas impartidas en el Bachillerato que forman parte del plan de estudios, cuyo fin es registrar la evaluación de conocimientos y asistencias de los alumnos.

Nombre: Grados

Descripción: Es un nivel de formación establecido por el plan de estudios, su objetivo es determinar el grado de conocimientos de los alumnos y su avance paulatino a través del plan.

Nombre: Grupos

Descripción: Es el establecimiento de conjuntos de alumnos que comparten actividades y materias en común, su objetivo es dividir en agrupaciones el total de alumnos de cierto grado para facilitar la administración del conocimiento.

Nombre: Ciclo Escolar

Descripción: Es periodo de tiempo oficial establecido con fechas de inicio y final, dentro del cual se realizan las actividades del Bachillerato.

Nombre: Horario

Descripción: Información acerca del periodo de tiempo que dura una actividad o materia durante el día, su objetivo es establecer la hora de inicio y fin de cada actividad para cada día de la semana.

Nombre: Calificaciones

Descripción: Es información acerca de la evaluación de los alumnos para cada una de las materias asignadas durante un momento o semestre, su objetivo es registrar un valor numérico que expresa el grado de aprendizaje de los alumnos.

Nombre: Asistencias

Descripción: Es información acerca de la evaluación de los alumnos para cada una de las materias asignadas durante un momento o semestre, su objetivo es registrar un valor numérico que expresa número de veces que el alumnos asistió a clase.

3.4. Diseño del sistema

3.4.1. Modelo de Análisis

El modelo de análisis representa la conceptualización del entorno de desarrollo del sistema, se agrupa en dos partes o vistas:

Modelo estático: que permite representar la estructura del modelo de análisis.

Modelo dinámico: que muestra las interacciones y responsabilidades que se manejan en el sistema. [5]

Para construir este modelo realizamos análisis de casos de uso tomando como base los modelos de casos de uso y conceptual, así como los documentos de requerimientos funcionales y no funcionales.

Iniciamos analizando los casos de uso para identificar los objetos y clases del sistema, obteniendo “clases de análisis” las cuales pueden ser entidad, límite, control, excepción, meta clase y utilitario.

Para nuestro caso particular es suficiente con la identificación de las clases: entidad, límite y control; mismas que son presentadas a continuación:

3.4.1.1. Clases de análisis

a) Caso de uso: Establecer parámetros del sistema

Clases.-

Entidad	Limite	Control
Grado	PantallaParámetrosSistema	AlmacenarParametros
Grupo	PantallaCrearParámetros	EliminarParametros
Semestre	PantallaEliminarParámetros	ModificarParámetros
CicloEscolar	PantallaBuscarPárametros	BuscarPárametros
Materias	PantallaModificarParámetros	
Cursos	PantallaMostrarParámetros	
Horarios	PantallaMensajeError	

b) Caso de uso: Administrar usuarios
Clases.-

Entidad	Limite	Control
Personal	PantallaAdministrarUsuarios	AlmacenarUsuario
Usuario	PantallaAgregarUsuario	EliminarUsuario
	PantallaEliminarUsuario	Modificarusuario
	PantallaModificarUsuario	BuscarUsuario
	PantallaBuscarUsuario	
	PantallaMostrarUsuarios	
	PantallaMensajeError	

c) Caso de uso: Administrar academia
Clases.-

Entidad	Limite	Control
Alumno	PantallaAdministrarAcademia	AlmacenarAlumno
Personal	PantallaAgregarAlumno	AlmacenarPersonal
	PantallaAgregarPersonal	EliminarAlumno
	PantallaDarBajaAlumno	EliminarPersonal
	PantallaDarBajaPersonal	ModificarAlumno
	PantallaModificarAlumno	ModificarPersonal
	PantallaModificarPersonal	BuscarAlumno
	PantallaBuscarAlumno	BusacarPersonal
	PantallaBuscarPersonal	
	PantallaMostrarAlumno	
	PantallaMostrarPersonal	
	PantallaMensajeError	

d) Caso de uso: Ingresar al sistema
Clases.-

Entidad	Limite	Control
Usuario	PantallaRegistroUsuario	IngresarAlSistema
	PantallaInicioSistema	
	VentanaMensajeUsuarioNoValido	

e) Caso de uso: Administrar calificaciones y asistencias
Clases.-

Entidad	Limite	Control
Materias	PantallaAdminCalifyAsist	AlmacenarCalificación
Grado	PantallaAgregarCalificación	AlmacenarAsistencia
Grupo	PantallaAgregarAsistencia	BuscarCalificación
CicloEscolar	PantallaModificarCalificación	BuscarAsistencia
Calificaciones	PantallaModificarAsistencia	ModificarCalificación
	PantallaMensajeError	ModificarAsistencia

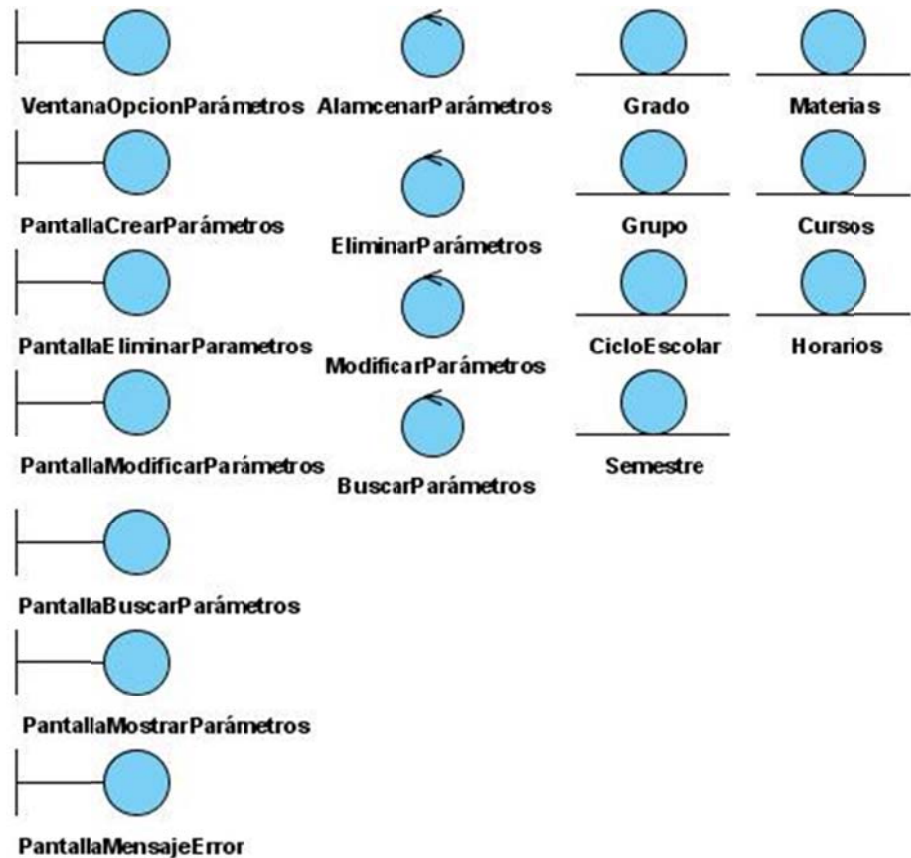
f) Caso de uso: Imprimir documentos para el alumno
Clases.-

Entidad	Limite	Control
Materias	PantallaDocumentos	GenerarBoletas
Grado	PantallaGeneraBoletas	GenerarCertificados
Grupo	PantallaGeneraCertificados	GenerarListAsistencia
CicloEscolar	PantallaGeneraConstancias	BuscarDocumentos
Calificaciones	PantallaGeneraListAsistencia	ImprimirDocumentos
Boletas	PantallaBuscarDocumentos	
Constancias	PantallaMuestraBoletas	
Certificados	PantallaMuestraListAsistencia	
ListasAsistencia	PantallaMuestraCertificados	
Alumnos	PantallaMuestraConstancias	
	PantallaMensajeError	

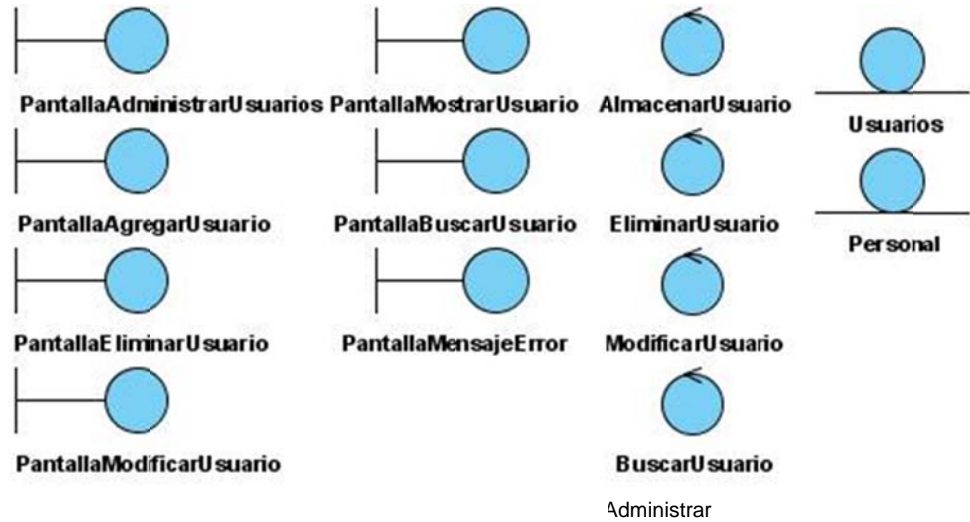
3.4.1.2. Diagramas de clases de análisis

A partir de la identificación de nuestras clases de análisis, podemos elaborar los diagramas de clases.

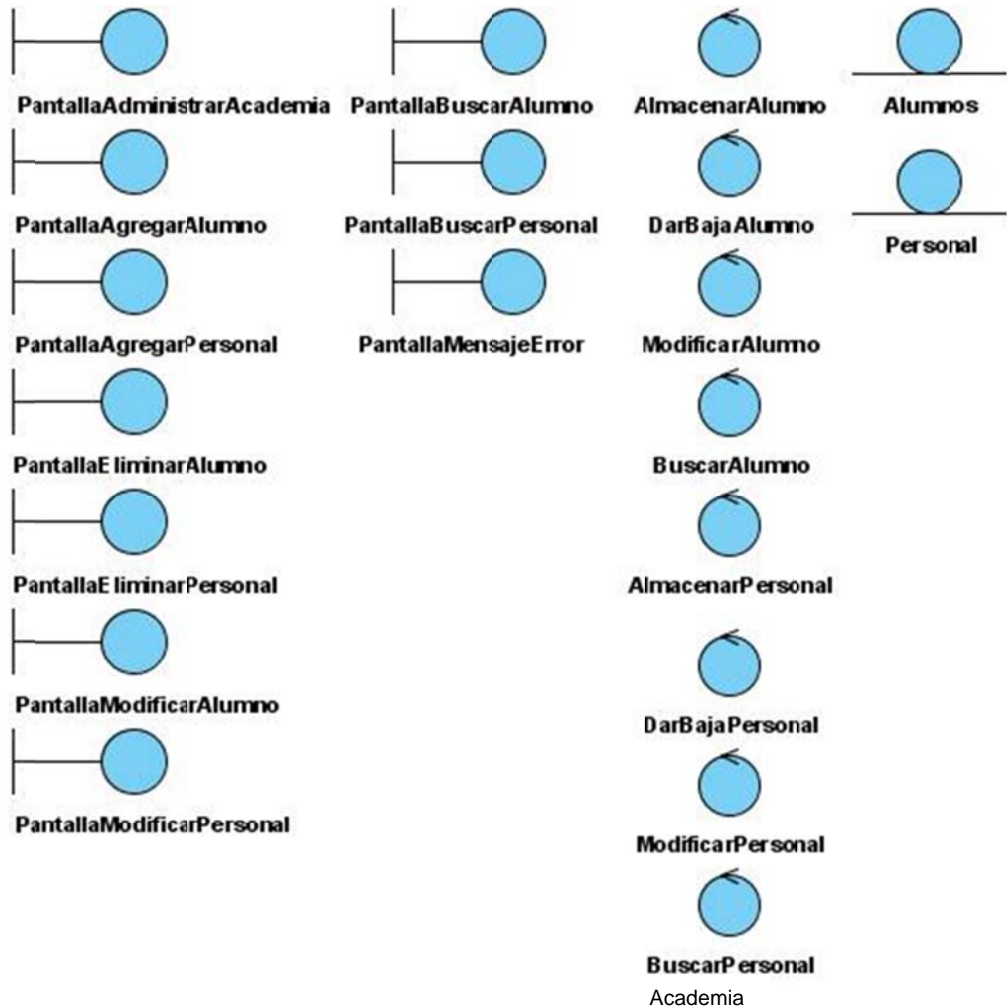
a) Caso de uso Establecer Parámetros



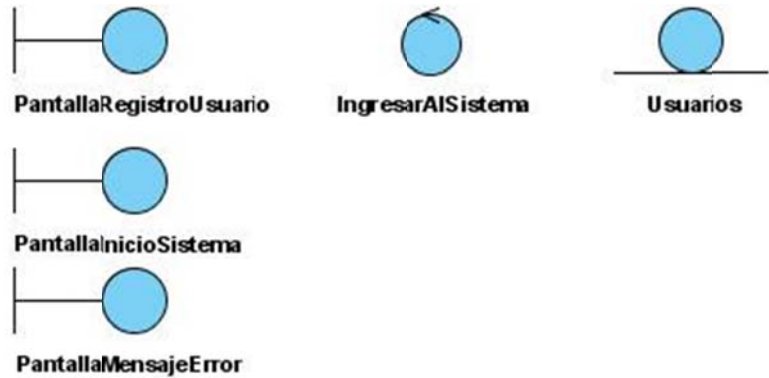
b) Caso de uso Administrar Usuarios



c) Caso de uso Administrar Academia

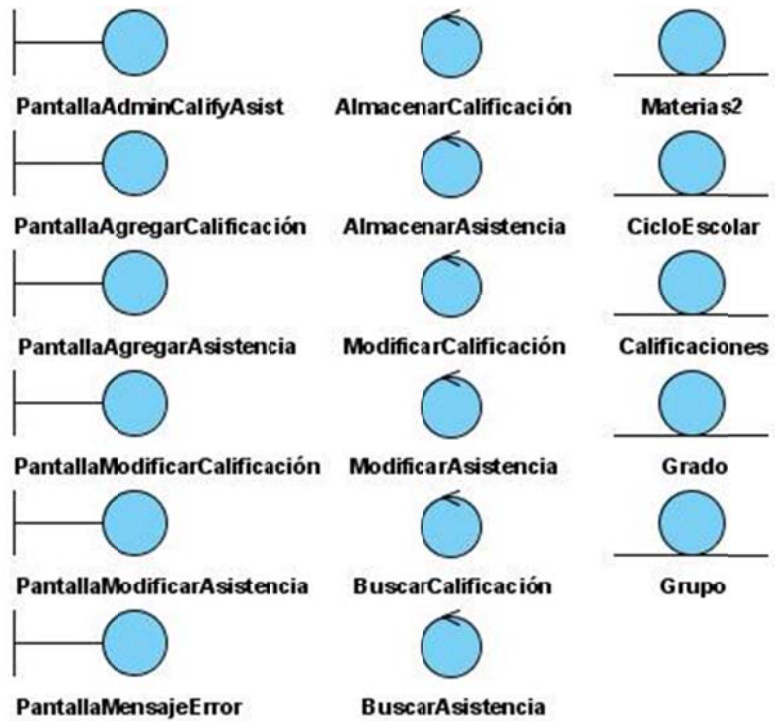


d) Caso de uso Ingresar al Sistema



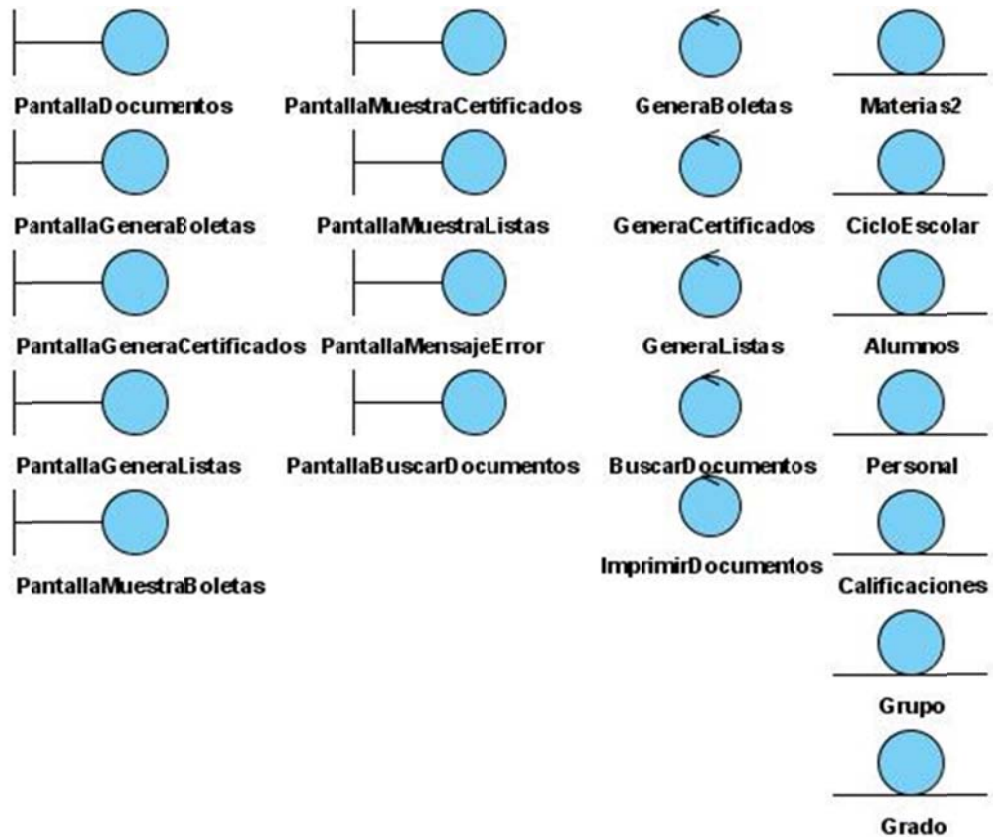
ii

a) Caso de uso Administrar Calificaciones y Asistencias



inistrar

a) Caso de uso Imprimir Documentos



3.4.1.3. Diagramas de Interacción

Después de obtener las clases participantes y sus diagramas, que muestran el modelo estático del sistema, el siguiente paso es obtener el modelo dinámico.

Partiendo de los escenarios de CU, elaboramos diagramas de interacción para identificar las propiedades y responsabilidades de los objetos y clases.

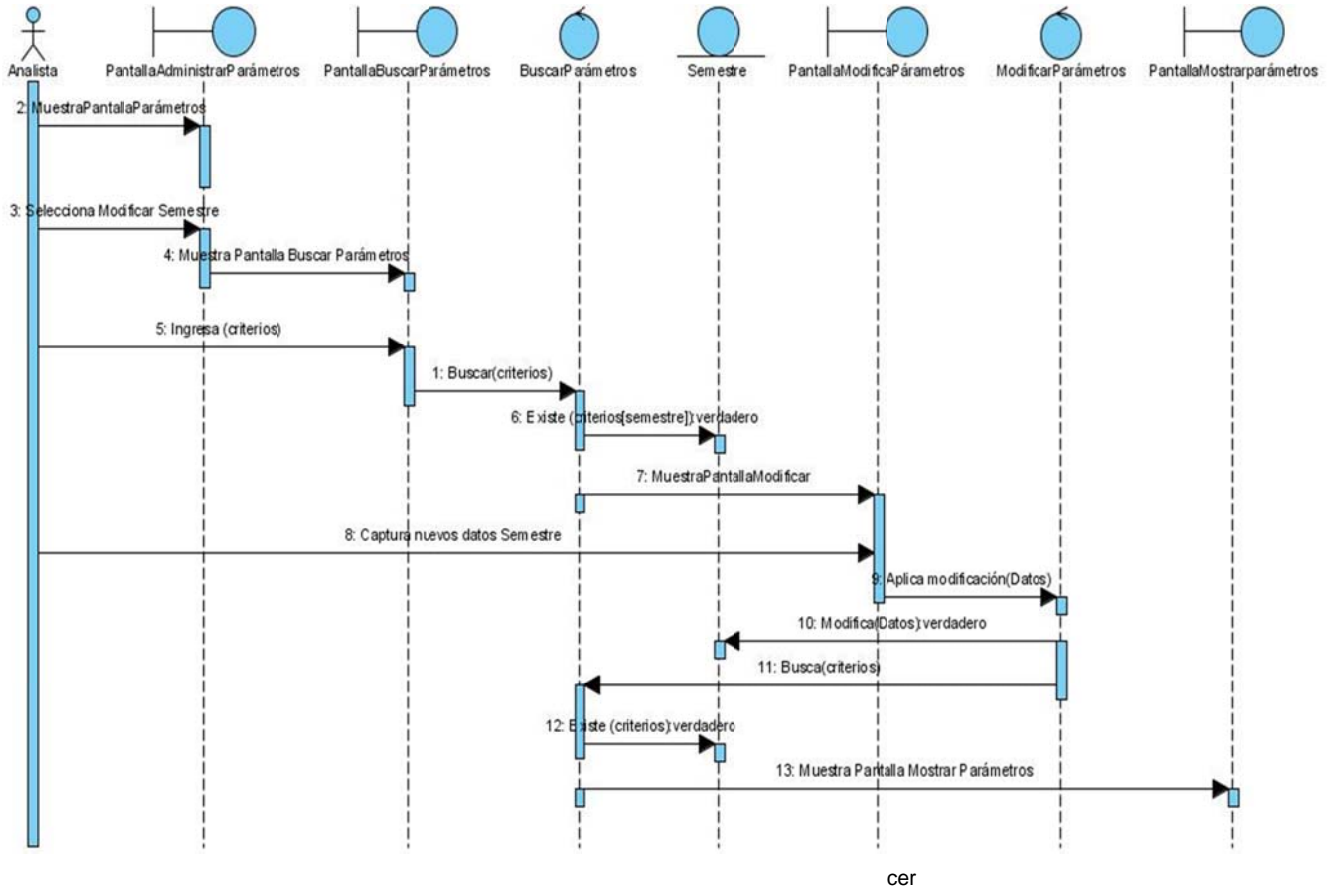
3.4.1.3.1. Diagramas de Secuencia

Un diagrama de secuencia sirve para modelar las interacciones entre objetos ordenadas en una secuencia en el tiempo. Incluye: [5]

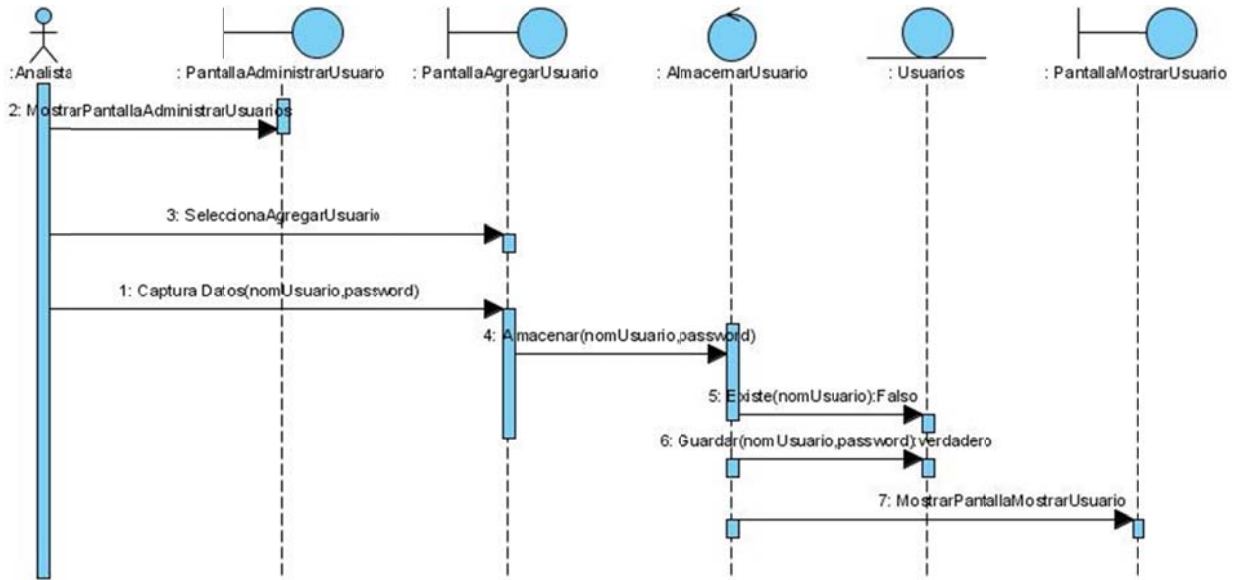
- Los objetos que participan en el escenario con sus líneas de vida
- Los mensajes intercambiados, que representan el flujo de eventos.
- El enfoque de control sobre los objetos. [5]

Mostramos a continuación los diagramas de secuencia para cada caso de uso:

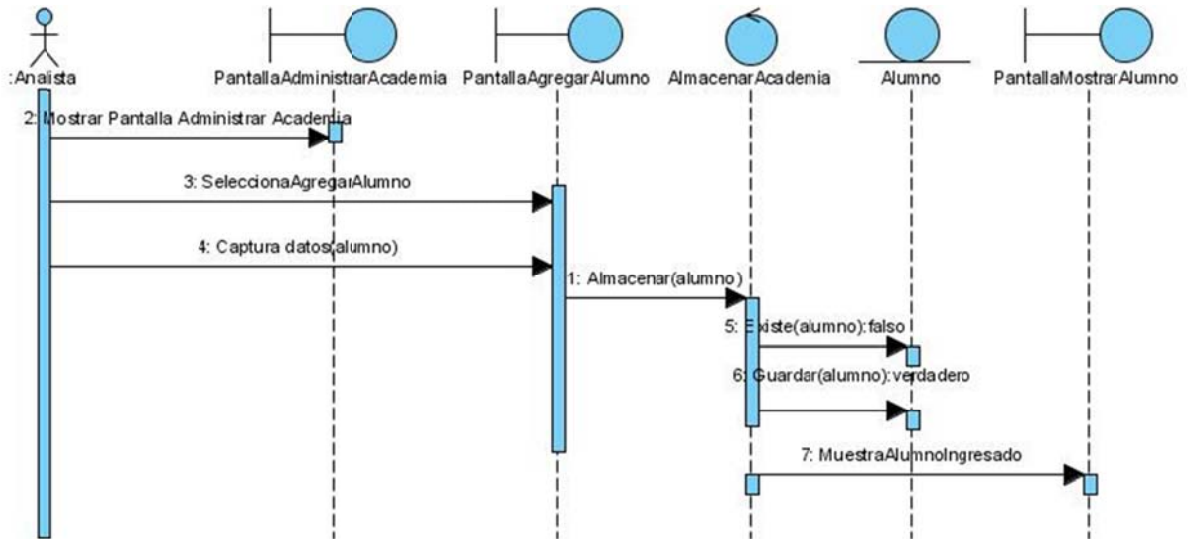
a) Establecer Parámetros



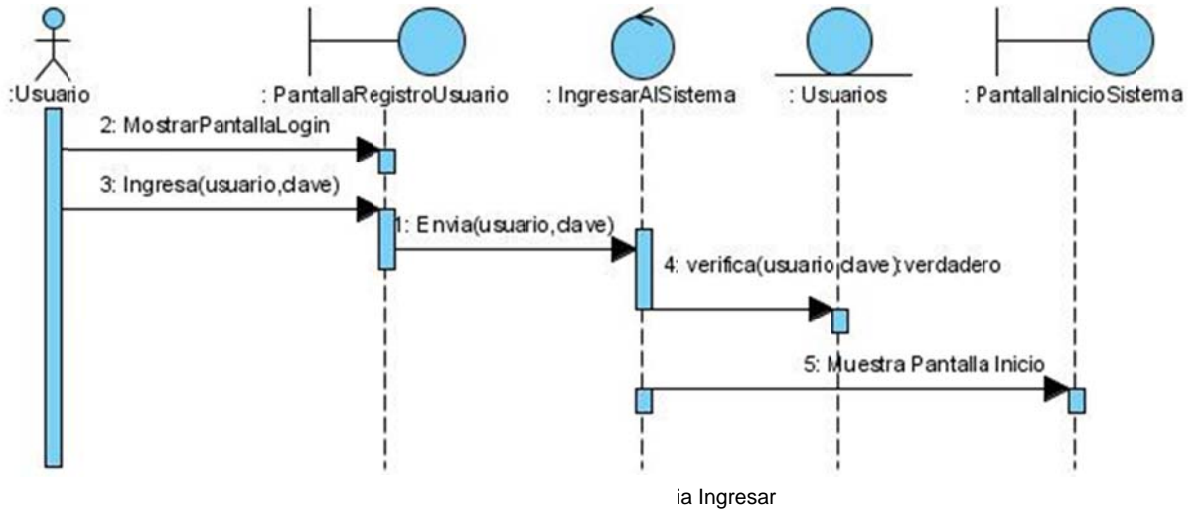
b) Administrar Usuarios



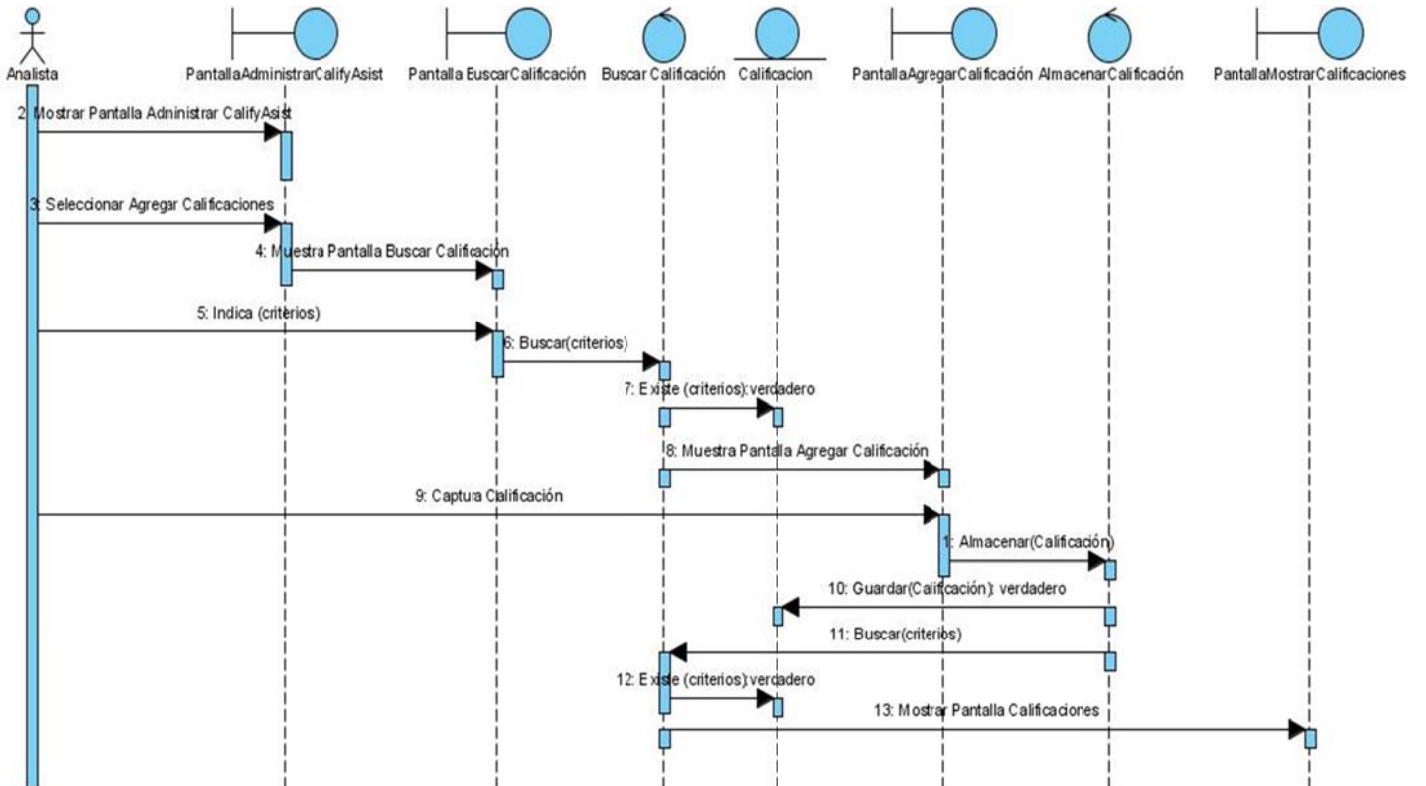
c) Administrar Academia



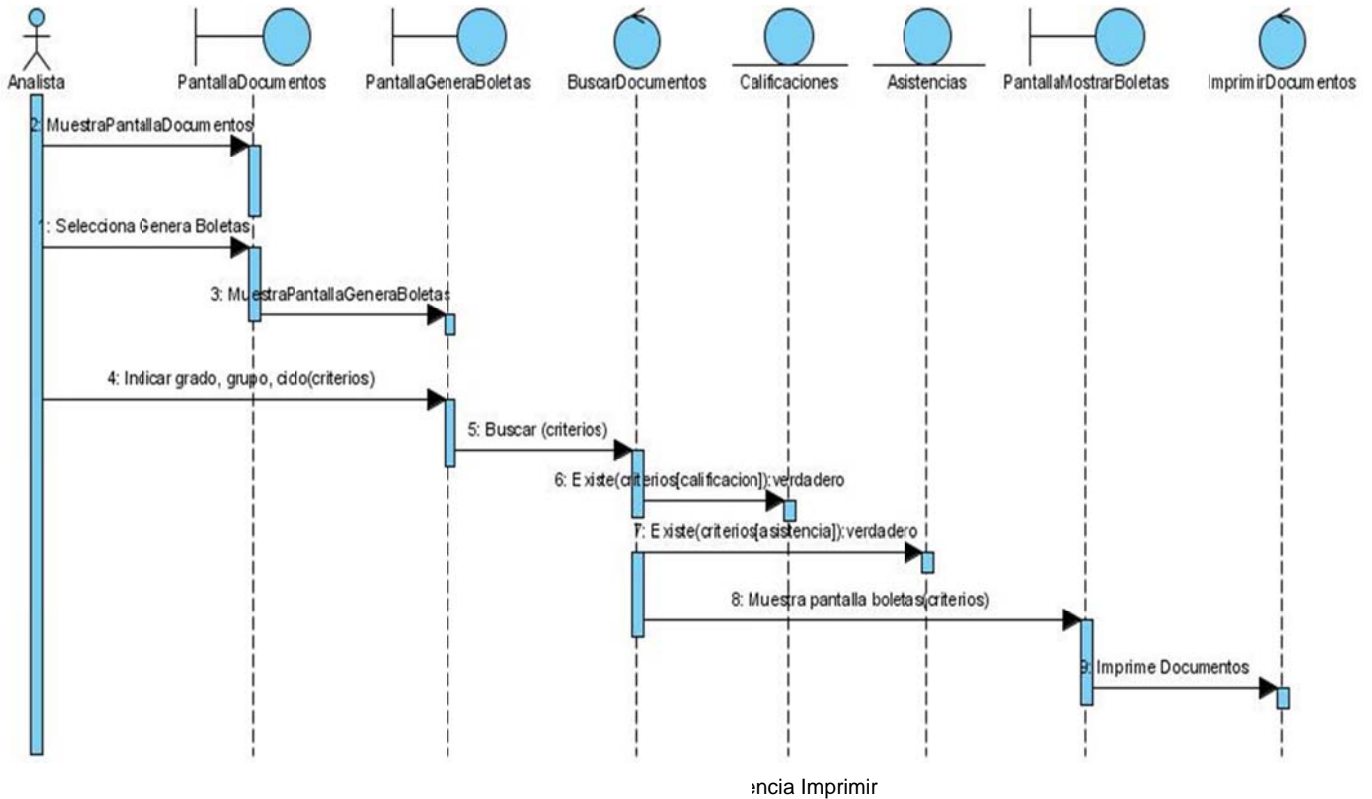
d) Ingresar al sistema



e) Administrar Calificaciones y Asistencias



f) Imprimir documentos



3.4.1.3.2. Diagramas de colaboración

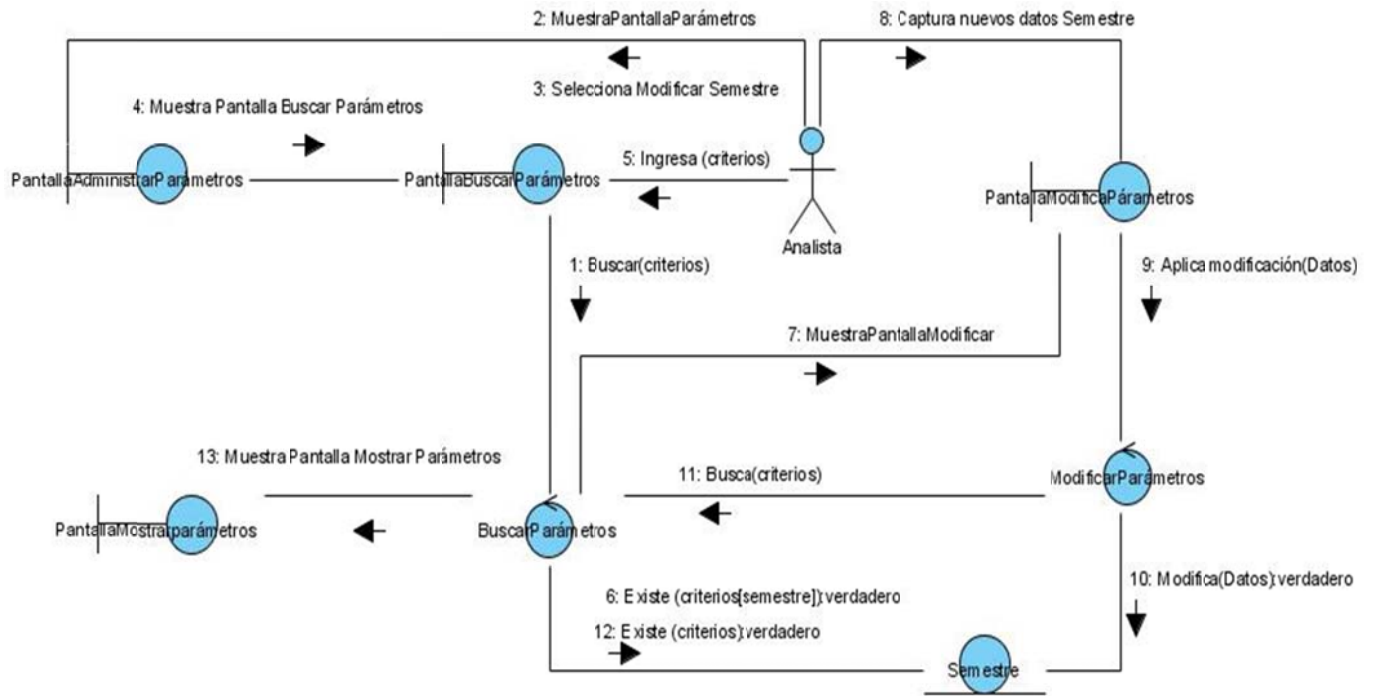
Son una forma alternativa de representar los mensajes intercambiados por un conjunto de objetos. El diagrama muestra los objetos con enlaces entre ellos cuando hay una o más interacciones y las interacciones se muestran como mensajes dibujados sobre los enlaces. [5]

Este diagrama incluye:

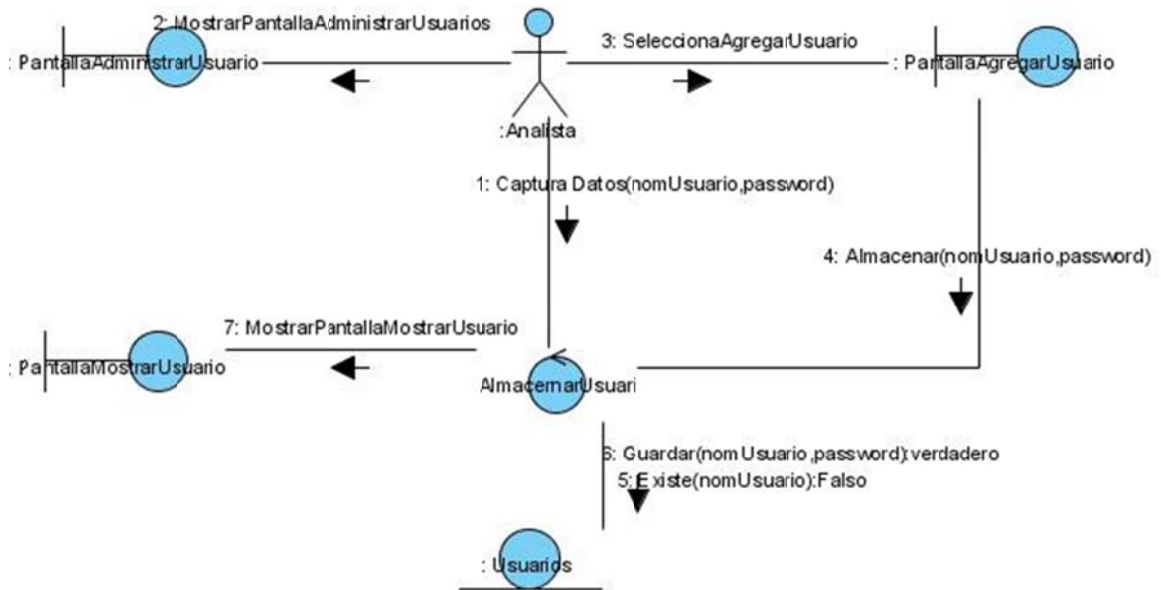
- Objetos
- Enlaces entre objetos
- Mensajes intercambiados entre objetos; y
- Flujo de datos entre objetos, si existen. [5]

Mostramos a continuación dichos diagramas:

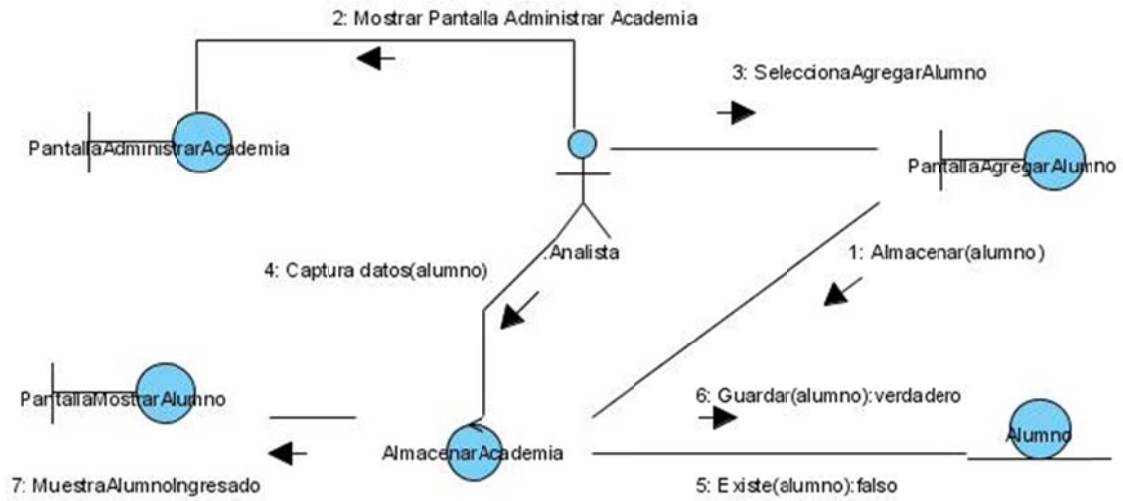
a) Establecer parámetros



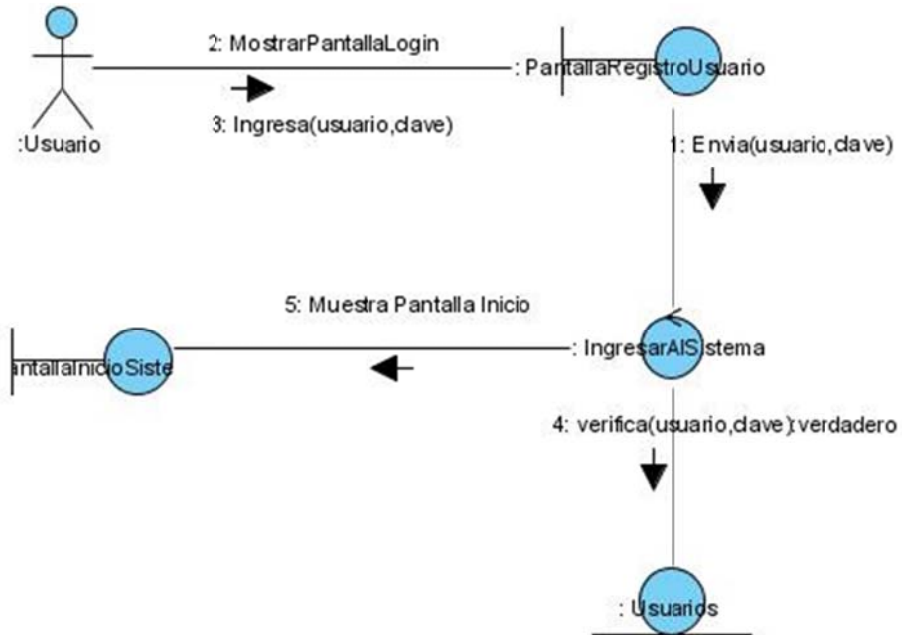
b) Administrar usuarios



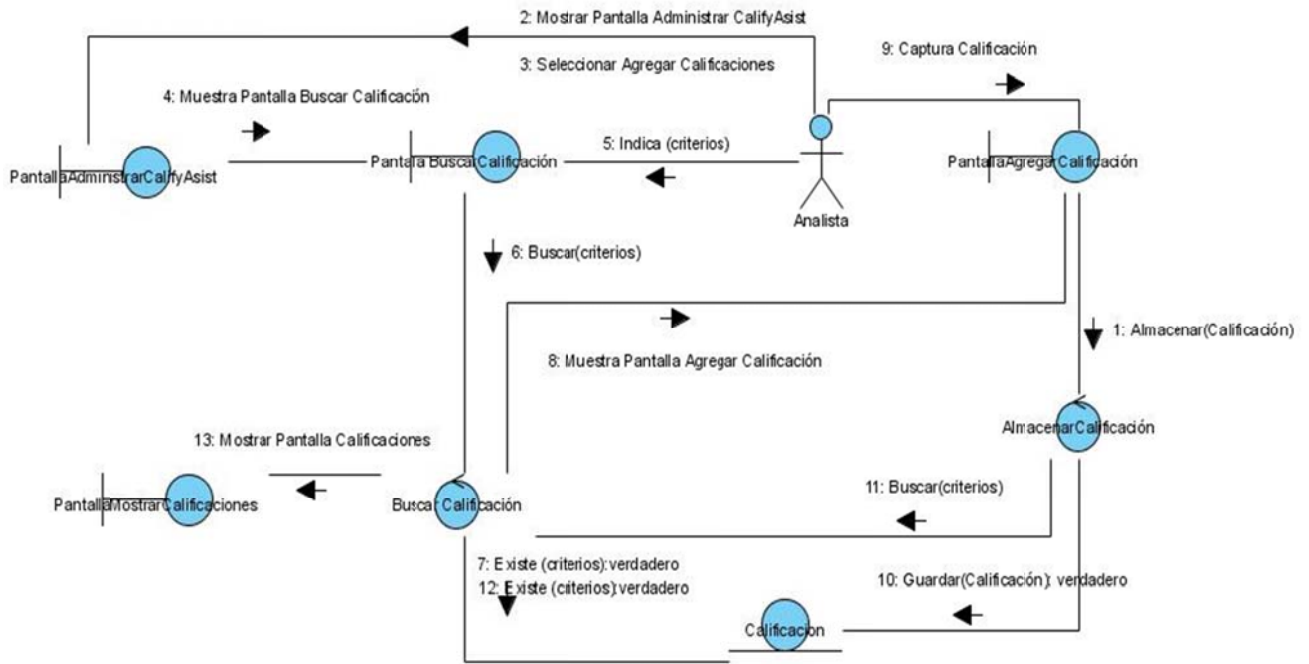
c) Administrar Academia



d) Ingresar al sistema

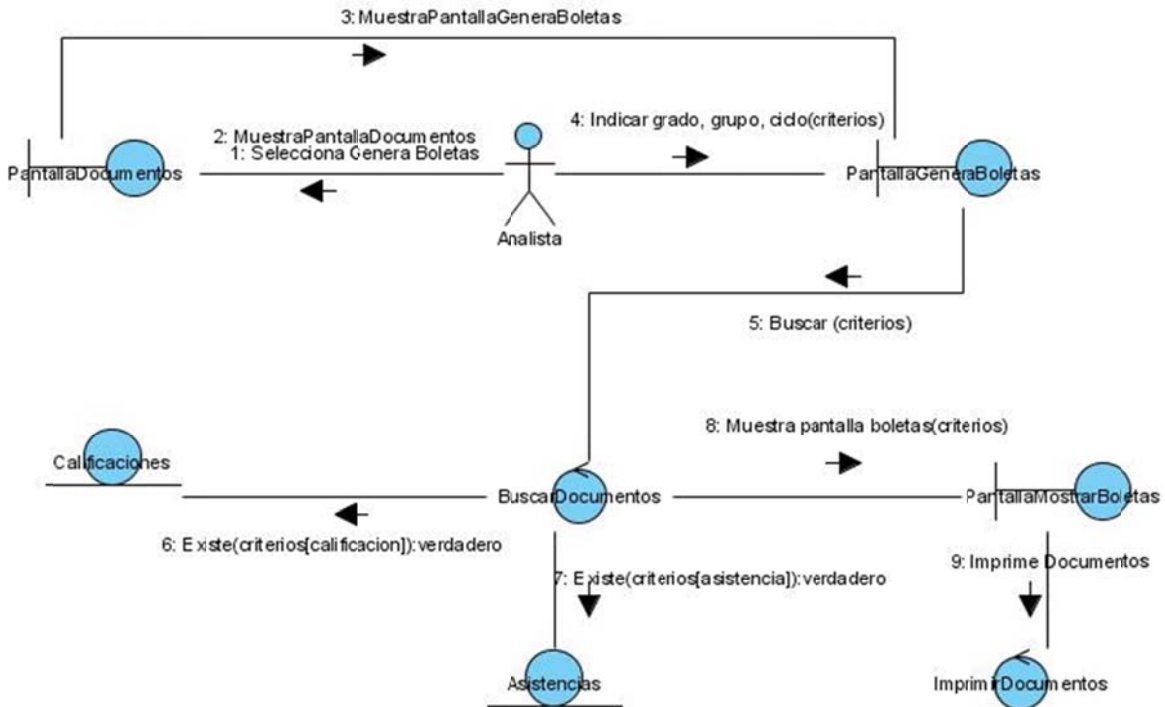


e) Administrar Calificaciones y Asistencias



s

f) Imprimir documentos



En este punto hemos realizado el análisis del sistema utilizando los requerimientos funcionales y no funcionales, elaboramos el diagrama de casos de uso y la documentación de cada uno de ellos, describimos escenarios para estos casos de uso y realizamos nuestra primera abstracción de clases para graficar el diseño conceptual del sistema.

Posteriormente realizamos análisis de casos de uso tomando los modelos de casos de uso y conceptual, agregando clases en algunos puntos y eliminando en otros, las clases resultantes se estereotiparon en clases: límite, control y entidad lo que permitió diseñar diagramas VOPC, así como diagramas de secuencia y colaboración, para obtener los modelos estático y dinámico, que muestran la estructura e interacción del sistema respectivamente.

Estos modelos son la base para realizar la implementación, el modelo estático sirve para diseñar la BD, que es la información persistente o que debe almacenarse en el sistema; y el modelo dinámico permite desarrollar las reglas de negocio y la interfaz del usuario, etapas que se describen en los siguientes capítulos.

CAPITULO IV

DISEÑO DE LA BASE DE DATOS

El diseño consiste en definir la estructura de los datos que debe tener la base de datos del sistema de información, así como las relaciones que existen entre estos. [4]

Para realizar lo anterior utilizamos las teorías que permiten diseñar el modelo de base de datos de nuestro sistema.

4.1. Diseño conceptual

En esta etapa obtenemos una estructura de la información de la BD independiente de la tecnología que hay que emplear. No se tiene en cuenta qué tipo de base de datos se utilizará (relacional, orientada a objetos, jerárquica, etc.). [4]

Para expresar los resultados del diseño conceptual utilizaremos el modelo Entidad-Relación, así también, se hace la observación que para fines de este trabajo se utilizará la notación de Oracle en el diseño del modelo.

4.1.1. Modelo entidad-relación

Es un enfoque de modelización de datos y su nombre proviene de que los principales elementos que incluye son las entidades y las relaciones. [4]

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de un conjunto de objetos básicos llamados entidades y de relaciones entre estos objetos. [4]

4.1.1.1. Entidades

Una entidad es una “cosa” u “objeto” en el mundo real que podemos distinguir de otros objetos y del que nos interesan sus propiedades. [4]

El método entidad-relación parte del hecho de que uno de los resultados del análisis del sistema es la comprensión clara de cuáles son las entidades incluidas en él. [6]

Teniendo esto en cuenta y considerando que previamente hemos realizado el análisis de requerimientos donde obtuvimos las clases de análisis de tipo entidad, las cuales representan la información persistente en el sistema, tomaremos dichas clases como las entidades del presente modelo, ya que representan los objetos de información del mismo:

- ALUMNO
- PERSONAL
- USUARIO
- MATERIA
- GRADO
- GRUPO
- SEMESTRE
- CICLOESCOLAR
- HORARIO
- ASISTENCIA
- CALIFICACION

4.1.1.2. Atributos

Los atributos son información acerca de una entidad que necesita ser conocida o descrita. Los atributos representan un tipo de descripción o detalles para cada instancia. [6]

Aplicando la teoría y los conceptos del modelo para encontrar los atributos, así como la notación elegida para su representación, tenemos que las Entidades con sus atributos quedarían de la siguiente forma:

- ALUMNO (*NIA, *CURP, *aPaterno, *aMaterno, *nomAlumno, °calle, °numInterior, °numExterior, °colonia, °localidad, °estado, °fechaNac, °edad, °sexo, *status, °telCasa).
- PERSONAL (aPaterno, aMaterno, nomPersonal, °calle, °numInterior, °numExterior, °colonia, °localidad, °estado, °categoria, °plaza, °tipoPersonal, °profesión, *status).
- USUARIO (*nomUsuario, *password,).
- CICLOESCOLAR (*añoInicio, *añoFin, *desc).
- SEMESTRE (*nomSemestre).
- GRADO (*nivel).
- GRUPO (*letraGrupo).
- HORARIO (°lunInicio, °lunFin, °marInicio, °marFin, °mieInicio, °mieFin, °jueInicio, °jueFin, °vieInicio, °vieFin).
- MATERIA (*codMateria, *nomMateria, °especialidad).
- CALIFICACION (°califMomento1, °califMomento2, °califMomento3, °califFinal).
- ASISTENCIA (asistMomento1, asistMomento2, asistMomento3, porcentajeAsist).

En la lista de entidades con sus atributos distinguimos los atributos obligatorios con el símbolo “*”, de los opcionales que llevan el símbolo “o”

4.1.1.3. Identificadores Únicos

Una entidad debe ser distinguible del resto de los objetos del mundo real. Esto hace que para toda Entidad sea posible encontrar un conjunto de atributos que permitan identificarla. [4]

Procedemos a analizar los atributos de las entidades para distinguir un identificador único o clave primaria, en este caso y debido a que en las reglas de negocio no se encuentra especificada una clave como identificador, se asignará un identificador único para cada entidad, tal como se muestra:

- ALUMNO (#idAlumno)
- PERSONAL (#idPersonal)
- Usuario(#idUsuario)
- CICLOESCOLAR (#idCicloEscolar)
- SEMESTRE (#idSemestre)
- GRADO (#idGrado)
- GRUPO (#idGrupo)
- HORARIO (#idHorario)
- MATERIA (#idMateria)
- CALIFICACION (#idCalificacion)
- ASISTENCIA (#idAsistencia)

Notamos que el identificador único se antepone el símbolo “#”.

Así mismo, detallamos los atributos de cada Entidad como sigue:

ALUMNO

Atributo	Descripción	tipo	Longitud	Rango
idAlumno	Identificador único	Entero		0..*
aPaterno	Apellido paterno	Carácter	20	a-z, A-Z
aMaterno	Apellido materno	Carácter	20	a-z, A-Z
nomAlumno	Nombre(s) del alumno	Carácter	20	a-z, A-Z
NIA	Registro del Alumno	Entero	8	0..9
CURP	Registro de población	Combinación caracteres y enteros	20	A-Z, 0-9
fechaNac	Fecha de nacimiento	Fecha		
Sexo	Genero	Carácter	1	H o M
calle	parte del domicilio	Carácter	10	a-z, A-Z
numInterior	parte domicilio	Entero		0..*
numexterior	parte del domicilio	Entero		0..*
Colonia	parte del domicilio	Carácter	20	a-z, A-Z
Localidad	parte del domicilio	Carácter	20	a-z, A-Z
Municipio	parte del domicilio	Carácter	20	a-z, A-Z
Estado	parte del domicilio	Carácter	20	a-z, A-Z
telCasa	Teléfono casa	Numero	10	0-9

Tabla 4.1. Detalle Atributos de ALUMNO

PERSONAL

Atributo	Descripción	tipo	Longitud	Rango
idPersonal	Identificador único	Entero		0..*
aPaterno	Apellido paterno	Carácter	20	A-Z
aMaterno	Apellido materno	Carácter	20	A-Z
nomPersonal	Nombre del trabajador	Carácter	20	A-Z
Categoría	Nivel de plaza del trabajador	Entero	8	A-Z, 0..9
Plaza	Identificador de la plaza	Entero	6	0..9
tipoPersonal	Tipo de trabajador (Administrativo o Docente)	Carácter	15	A-Z
Profesión	Especialidad del	Carácter	20	A-Z

	trabajador			
calle	parte del domicilio	Carácter	10	A-Z
numInterior	parte domicilio	Entero		0..*
numexterior	parte del domicilio	Entero		0..*
Colonia	parte del domicilio	Carácter	20	A-Z
Localidad	parte del domicilio	Carácter	20	A-Z
Municipio	parte del domicilio	Carácter	20	A-Z
Estado	parte del domicilio	Carácter	20	A-Z
telCasa	Teléfono casa	Numero	10	0-9

Tabla 4.2. Detalle Atributos de PERSONAL

USUARIO

Atributo	Descripción	tipo	Longitud	Rango
idUsuario	Identificador único	Entero		0..*
nomUsuario	Nombre de usuario	Carácter	20	a-z, A-Z, 0..9, caracteres especiales
password	Clave de acceso al sistema	Carácter	20	a-z, A-Z, 0..9, caracteres especiales

Tabla 4.3. Detalle Atributos de USUARIO

CICLOESCOLAR

Atributo	Descripción	tipo	Longitud	Rango
idCiclo	Identificador único	Entero		0..*
anoIni	Año de inicio del ciclo escolar	Entero	4	0..9
anoFin	Año de finalización del ciclo escolar	Entero	4	0..9
desc	Nombre descriptivo para el ciclo escolar	Carácter	20	A-Z

Tabla 4.4. Detalle Atributos de CICLOESCOLAR

SEMESTRE

Atributo	Descripción	tipo	Longitud	Rango
idSemestre	Identificador único	Entero		0..*
nomSemestre	Nombre descriptivo para el semestre	Carácter	7	{Primer, Segundo, Tercero, Cuarto, Quinto, Sexto}

Tabla 4.5. Detalle Atributos de SEMESTRE

GRADO

Atributo	Descripción	tipo	Longitud	Rango
idGrado	Identificador único	Entero		0..*
nivel	Nivel escolar	Entero	1	{1º, 2º, 3º}

Tabla 4.6. Detalle Atributos de GRADO

GRUPO

Atributo	Descripción	tipo	Longitud	Rango
idGrupo	Identificador único	Entero		0..*
letraGrupo	Indicador de agrupación de alumnos	Carácter	1	A-Z

Tabla 4.7. Detalle Atributos de GRUPO

HORARIO

Atributo	Descripción	tipo	Longitud	Rango
idHorario	Identificador único	Entero		0..*
lunInicio	Hora de Inicio y Finalización de la materia por día	Hora		8:00-16:00
lunfin		Hora		
marInicio		Hora		
marfin		Hora		
mielInicio		Hora		
miefin		Hora		
jueInicio		Hora		

juefin		Hora		
vielnicio		Hora		
viefin		Hora		

Tabla 4.8. Detalle Atributos de HORARIO

CALIFICACION

Atributo	Descripción	tipo	Longitud	Rango
idCalificacion	Identificador único	Entero		0..*
califMom1	Calificación del primer periodo de evaluación	Entero	1	0-10
califMom2	Calificación del segundo periodo de evaluación	Entero	1	0-10
califMom3	Calificación del tercer periodo de evaluación	Entero	1	0-10
califFin	Calificación final del Semestre	Entero	1	0-10

Tabla 4.9. Detalle Atributos de CALIFICACIÓN

ASISTENCIA

Atributo	Descripción	tipo	Longitud	Rango
idAsistencia	Identificador único	Entero		0..*
asistMom1	Total de asistencias del primer periodo de evaluación	Entero	2	0-30
asistMom2	Total de asistencias del segundo periodo de evaluación	Entero	2	0-30
asistMom3	Total de asistencias del tercer periodo de evaluación	Entero	2	0-30
porcentajeAsist	Porcentaje de asistencias del Semestre por materia	Entero	3	0-100

Tabla 4.10. Detalle Atributos de ASISTENCIA

4.1.1.4. Relaciones

Las relaciones representan las reglas y la información que el negocio necesita. [6]

De las Entidades que tenemos detectadas de nuestro análisis de requerimientos, debemos establecer las relaciones que existen entre estas y que tipo de relación tienen establecida.

Del análisis para el diseño de la BD, se observó que hace falta una entidad que permita crear un curso para que los alumnos puedan inscribirse en él, esta nueva entidad se compone de propiedades de las entidades MATERIA, SEMESTRE, HORARIO, GRADOS, GRUPO, CICLOSESCOLAR y PERSONAL. Se determinó llamarla REGISTROCURSO y quedó formada como se muestra:

- REGISTROCURSO(#NRC, *idPersonal, *idMateria, *idCicloEscolar, *idSemestre, *idGrado, *idGrupo, *idHorario)

En este caso el identificador único es #NRC y los demás atributos son los identificadores únicos de las entidades que lo componen, lo cuáles se encuentran detallados en el apartado anterior.

Ahora pasaremos a describir las relaciones que existen entre las Entidades descritas con anterioridad:

PERSONAL con USUARIO

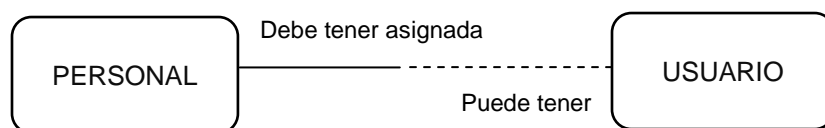


Fig. 4.1. Diagrama Relación PERSONAL-USUARIO

Al relacionar las entidades REGISTROCURSO y ALUMNO se genera una relación con cardinalidad muchos a muchos, por lo que se creó una nueva entidad que permita establecer una relación de uno a muchos entre estas, la que se nombró como INSCRIPCIÓN y está compuesta por propiedades de REGISTROCURSO y ALUMNOS, quedando como sigue:

- INSCRIPCIÓN(#NRC,#idAlumno)

ALUMNO CON INSCRIPCIÓN

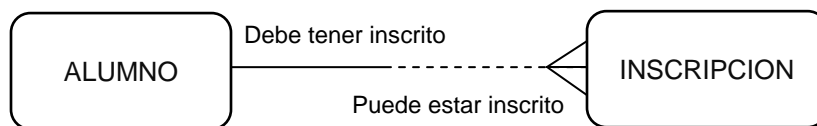


Fig. 4.2. Diagrama Relación ALUMNO-INSCRIPCIÓN

INSCRIPCIÓN CON REGISTRO CURSO

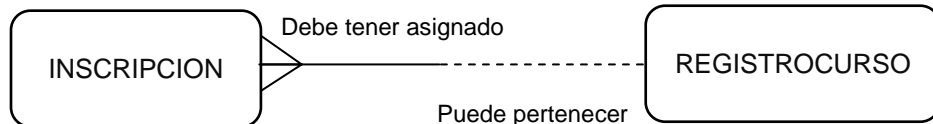


Fig. 4.3. Diagrama Relación INSCRIPCIÓN-REGISTROCURSO

Se observa que CALIFICACIÓN y ASISTENCIA pueden ser unidas en una sola entidad, ya que comparten similitud en el identificador único así como en la relación entre ellos, por lo tanto se eliminó la entidad ASISTENCIA y se unieron sus atributos para formar parte de la entidad CALIFICACIÓN. La relación quedó establecida como se muestra:

INSCRIPCIÓN CON CALIFICACIÓN

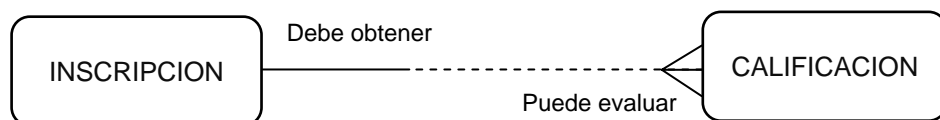


Fig. 4.4. Diagrama Relación INSCRIPCIÓN-CALIFICACIÓN

REGISTROCURSO concentra diferentes propiedades de las entidades, la mayoría de estas tienen una relación con REGISTROCURSO, por lo que se vuelve una entidad central en el sistema.

Por lo anterior las relaciones para REGISTROCURSO, quedan establecidas como lo muestra el diagrama:

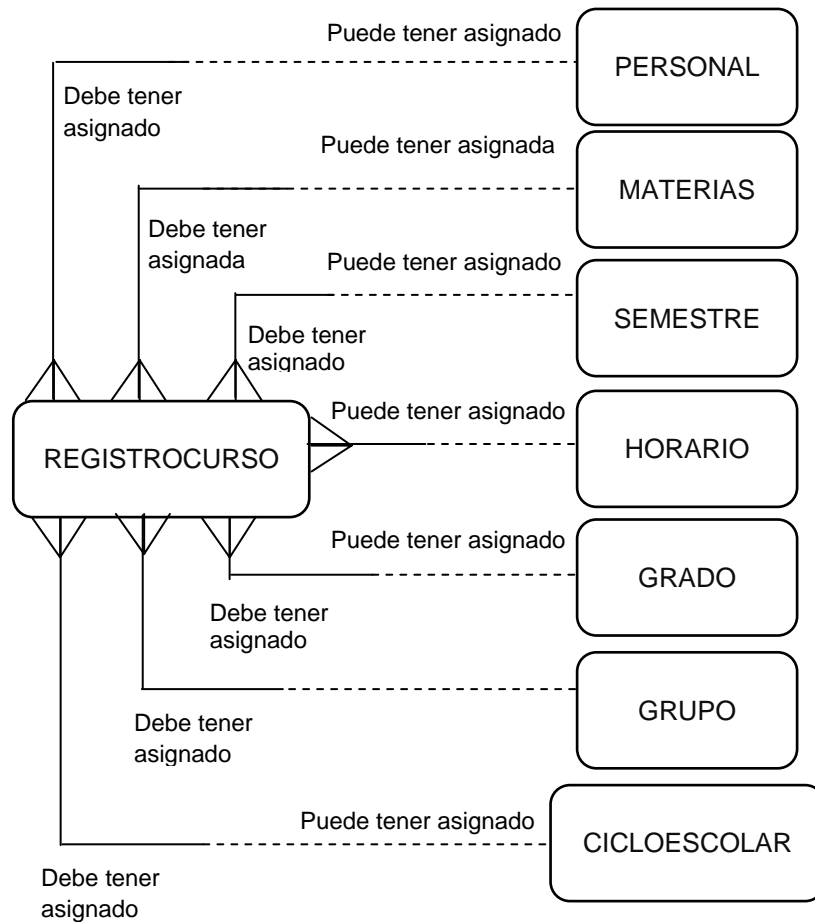
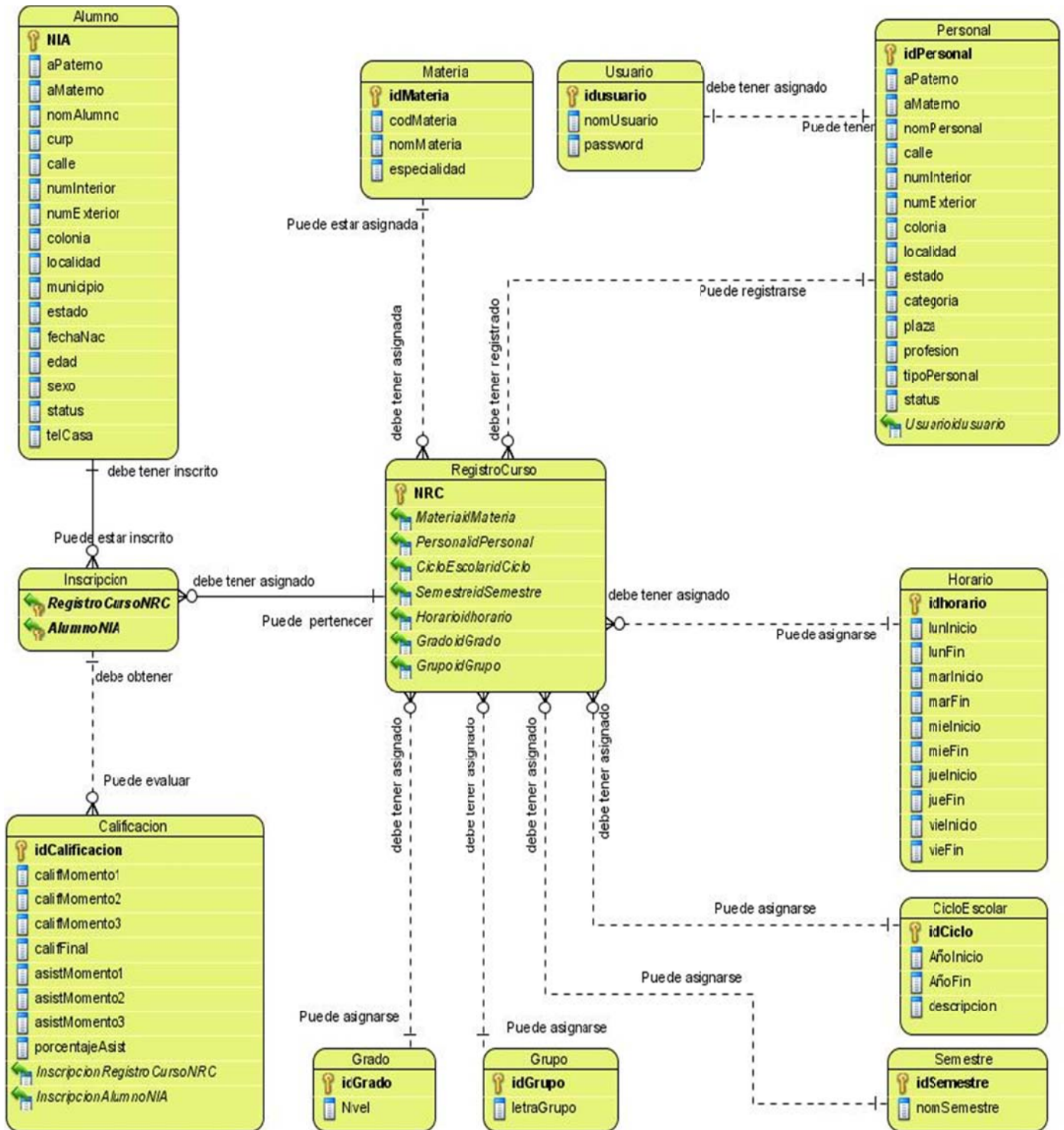


Fig. 4.4. Diagrama Relaciones REGISTROCURSO

Al unir los diagramas de las entidades con sus relaciones y atributos obtenemos el Diagrama Entidad-Relación para nuestro sistema, el cual se muestra en la siguiente figura.

Dicho diagrama fue diseñado en una herramienta de modelado especial.



- Relación

4.1.1.5. Normalización

Ahora que contamos con las entidades, sus atributos y los identificadores únicos definidos, procederemos a normalizar cada una de las entidades.

Iniciaremos mostrando la Dependencia funcional de los atributos de las entidades.

Alumno

idAlumno → NIA, CURP, aPaterno, aMaterno, nomAlumno, calle, numInterior, numExterior, colonia, localidad, estado, fechaNac, edad, sexo, status, telCasa

Personal

idPersonal → aPaterno, aMaterno, nomPersonal, calle, numInterior, numExterior, colonia, localidad, estado, categoría, plaza, tipoPersonal, profesión, status

Usuario

idUsuario → nomUsuario, password

Grado

idGrado → nivel

Grupo

idGrupo → letraGrupo

Semestre

idSemestre → nomSemestre

Ciclo

idCiclo → añoIni, añoFin, desc

Horario

idHorario —————> lunInicio, lunFin, marInicio, marFin,
mielInicio, mieFin, jueInicio, jueFin,
vieInicio, vieFin

Materias

idMateria —————> codMateria, nomMateria, especialidad

Calificaciones

idCalificacion —————> califMomento1, califMomento2,
califMomento3, califFinal,

Asistencias

idAsistencias —————> asistMomento1, asistMomento2,
asistMomento3, porcentajeAsist

De lo anterior observamos que Calificaciones y Asistencias pueden unirse, debido a que el momento escolar que utilizan es el mismo y la dependencia funcional no se afecta con la unión. Por lo que quedaría:

CalifyAsist

idCalificacion —————> califMomento1, califMomento2,
califMomento3, califFinal,
asistMomento1, asistMomento2,
asistMomento3, porcentajeAsist

Lo anterior nos muestra que todos los atributos de las entidades dependen funcionalmente de sus claves primarias.

Ahora analizaremos las entidades y realizaremos los procesos necesarios para llevarlas a la 4FN forma normal, de ser necesario.

Primera Forma Normal (1FN):

Una relación está en la primera forma normal (o 1FN) si todos los campos en cada registro contienen un solo valor tomado de sus dominios respectivos. **[6]**

Para verificar que las entidades cumplen con estar en primera forma normal, en el apartado diseño lógico se muestran dichas entidades como tablas con datos que cumplen con el requisito marcado en la primera forma normal.

Segunda Forma Normal (2FN)

Una relación es o pertenece a la segunda forma normal si es 1FN y cada atributo no-clave de la relación es total y funcionalmente dependiente de su clave principal. **[6]**

Como observamos en la descripción de dependencia funcional, se nota como cada atributo no clave de cada entidad, es total y funcionalmente dependiente de la clave principal.

Por lo que podemos asegurar que está en 2FN

Tercera Forma Normal (3FN)

Una relación es 3FN, si es 2FN y ningún atributo no-clave en la relación es funcionalmente dependiente de algún otro atributo no-clave. **[6]**

Nuevamente al analizar la descripción de dependencia funcional podemos apreciar que no existen atributos no clave, que sean dependientes de otros no clave en las relaciones.

Permitiendo afirmar que se encuentran en 3FN.

Cuarta Forma 4FN

Una relación esa 4FN (cuarta forma normal) si es BFCN y no contiene dependencias multivalores. [6]

Observamos que las entidades no contienen claves primarias compuestas, lo que nos indica que no tiene dependencias multivaluadas.

Por lo tanto se encuentran en 4FN.

4.2. Diseño lógico

Para realizar el diseño lógico de la BD, debemos tomar el resultado del diseño conceptual, es decir, el diagrama entidad-relación obtenido en el apartado anterior y transformarlo en una estructura de datos del modelo relacional.

El modelo relacional se basa en el hecho de que la estructura fundamental del modelo es una "relación", es decir una tabla bidimensional constituida por líneas (tuplas) y columnas (atributos).

Las relaciones representan las entidades de la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad.

De conformidad con el resultado del modelo entidad relación y aplicando la teoría del modelo relacional, tenemos que las entidades, sus atributos y relaciones, se representarían de la siguiente manera:

Sistema WEB para Control Escolar de un Bachillerato General

ALUMNO.-

idAlumno	nia	curp	aPaterno	aMaterno	nomAlumno	fNacimiento	edad	sexo
1	1022317	AURA960508MPLGJN05	ROJAS	AGUIRRE	ANAYELI	08/05/1996	15	M
2	1572396	AARY941122MPLMMN07	AMARO	RAMIREZ	YANELI	22/11/1994	17	M
3	1089278	AOHY960713MPLTRN05	ATONAL	HERNANDEZ	YANETT	13/07/1996	15	M
4	1022392	COCM960704HPLSLG09	COSME	CALDERON	MIGUEL ANGEL	04/07/1996	15	H
5	1572406	CO CJ931225HPLSNN08	COSME	CANTOR	JUAN MANUEL	25/12/1993	18	H
6	1022356	CO CJ960423HPLSVR06	COSME	CUEVAS	JORGE	23/04/1996	15	H
7	1007431	DITG941212HPLZCD09	DIAZ	TECPIL	JOSE GUADALUPE	12/12/1994	17	H
8	1013824	EARC960724HPLSMR02	ESCAMILLA	ROMERO	CRISTIAN	24/07/1996	15	H
9	1013866	EIAF960507HPLSGL09	ESPINDOLA	AGUIRRE	FLAVIO CESAR	07/05/1996	15	H
10	1022323	EIEE940825HPLSSD03	ESPINDOLA	ESCAMILLA	EDUARDO	25/08/1994	17	H

Tabla 4.11a. Relación ALUMNO

calle	numExt	numInt	colonia	Localidad	municipio	estado	telCasa	Status
Av. Zaragoza	563			San Francisco Ocotlán	Coronango	Puebla	2891497	NUEVO INGRESO
Mariano Abasolo	12	3		San Francisco Ocotlán	Coronango	Puebla	2261297	NUEVO INGRESO
3 Oriente	7		Centro	Puebla	Puebla	Puebla	2892240	NUEVO INGRESO
7 Poniente	4910	A	Nueva Antequera	Puebla	Puebla	Puebla	6251596	REINGRESO
Blvd las Torres	1290	8	Unversidades	Puebla	Puebla	Puebla	2892240	REINGRESO
Circuito del Bosque	3498		Lomas de Loreto	San Francisco Ocotlán	Coronango	Puebla	2894196	REINGRESO
Av. Nacional	123	B	Mayorazgo	Puebla	Puebla	Puebla	2894732	NUEVO INGRESO
25 Sur	3692		ReformaSur	Puebla	Puebla	Puebla	2891069	NUEVO INGRESO
43 Oriente	138		Chulavista	Puebla	Puebla	Puebla	2894281	BAJA
Niño artillero	22	C	Leobardo Coca	Puebla	Puebla	Puebla		BAJA

Tabla 4.11b. Continuación Relación ALUMNO

PERSONAL.-

idPersonal	aPaterno	aMaterno	nomPersonal	categoría	plaza	profesión	tipoPersonal	idUsuario
1	CORTES	BARRIENTOS	PILAR	E7817	256	Lic. Biología	Docente	
2	PEREZ	SALAZAR	CLAUDIA	E4618	089	Lic. Contaduría Pública	Docente	
5	RODRIGUEZ	PEREZ	CESAR	E4618	753	Lic. Derecho	Analista	2
7	LIMA	JIMENEZ	MANUEL	E7817	486	Lic. Informática	Docente	
9	CALDERON	CANIZAL	MIGUEL	E4517	356	Lic. Derecho	Docente	
10	PEREZ	ARRIETA	GUILLERMO	P7539	784	Lic. Ciencias de la Computación	Analista	2

Tabla 4.12a. Relación PERSONAL

Sistema WEB para Control Escolar de un Bachillerato General

calle	numExt	numInt	colonia	Localidad	municipio	estado	Status
Av. 18 Poniente	2345	2	Santa Anita	Puebla	Puebla	Puebla	NUEVO INGRESO
C. Argentina	1416		Las Hadas	Puebla	Puebla	Puebla	NUEVO INGRESO
C. Niño Artillero	89	6	Leobardo coca	Puebla	Puebla	Puebla	REINGRESO
Av. Fidel Velázquez	5312		La Margarita	Puebla	Puebla	Puebla	NUEVO INGRESO
C. 21 Sur	5432		Las Animas	Puebla	Puebla	Puebla	BAJA
Plaza del Palmar	Edif. DII	Depto. 1	La Margarita	Puebla	Puebla	Puebla	NUEVO INGRESO

Tabla 4.12b. Continuación Relación PERSONAL

USUARIO.-

idUsuario	nomUsuario	password
1	pegui	pegui
2	rope	rope
3	raga	raga

Tabla 4.13. Relación USUARIO

MATERIA.-

idMateria	codigoMateria	nomMateria	especialidad
1	80112	EDUCACIÓN FISICA I	
2	90112	EDUCACION ARTISTICA I	
3	15112	ESTADISITICA	
4	65112	FORM.PROP.	DESARROLLO HUMANO
5	75112	FORM.PROP.	PENSAMIENTO CRITICO Y CREATIVO
6	85112	FORM.PROP.	T. MED. COM. I
7	95112	CAP.P/TRAB.	SIST. TEC. IMP.

Tabla 4.14. Relación MATERIA

HORARIO.-

idHorario	LunIni	LunFin	MarIni	MarFin	MieIni	MieFin	JueIni	JueFin	VieIni	VieFin
1	8:00	8:50	10:30	11:00			14:00	15:00	13:30	14:00
2	9:00	9:50	10:30	11:00					8:50	9:00
3	9:00	9:50	9:00	9:50	9:00	9:50	9:00	9:50	9:00	9:50
4	9:50	10:30			9:50	10:30	9:50	10:30	9:50	10:30
5	10:30	11:00	10:30	11:00						

Tabla 4.15. Relación HORARIO

CICLO ESCOLAR.-

idCiclo	descripción	añoInicio	añoFin
1	Ciclo Escolar 2010-2011	2010	2011
2	Ciclo Escolar 2011-2012	2011	2012
3	Ciclo Escolar 2012-2013	2012	2013

Tabla 4.16. Relación CICLO ESCOLAR

SEMESTRE.-

idSemestre	nomSemestre
1	Primer
2	Segundo
3	Tercer
4	Cuarto
5	Quinto
6	Sexto

Tabla 4.17. Relación SEMESTRE

GRADO.-

idGrado	nivel
1	Primero
2	Segundo
3	Tercero

Tabla 4.18. Relación GRADO

GRUPO.-

idGrupo	letraGrupo
1	A
2	B
3	C

Tabla 4.19. Relación GRUPO

REGISTROCURSO.-

NCR	idMateria	idPersonal	idCiclo	idSemestre	idHorario	idGrado	idGrupo
1	1	7	1	1	1	1	1
2	2	4	2	2	2	1	2
3	3	3	3	3	3	2	2
4	4	4	1	1	4	1	2
5	3	5	1	1	1	1	3
6	4	6	3	3	3	2	3
7	2	7	3	6	3	3	3

Tabla 4.20. Relación REGISTROCURSO

INSCRIPCIÓN.-

NCR	NIA
1	1
1	3
1	5
4	5
4	5
6	2
7	1

Tabla 4.21. Relación INSCRIPCIÓN

CALIFICACION.-

idCalificacion	califMomento1	califMomento2	califMomento3	califFinal	asistMomento1	asistMomento2	asistMomento3	%Asistencias	NCR	NIA
1	7	7	9	8	19	18	19	98	1	1
2	6	8	9	8	19	19	19	100	2	2
3	8	8	7	8	19	19	18	98	3	5
4	9	8	7	8	25	24	25	98	3	6
5	6	9	7	8	25	24	25	98	2	2
5	7	10	7	8	22	23	21	95	6	2
7	8	7	8	8	20	20	19	98	6	1

Tabla 4.22. Relación CALIFICACIÓN

Lo realizado en los apartados anteriores nos ha permitido realizar el análisis y diseño de la Base de Datos de nuestro sistema.

Hemos podido describir las entidades y sus relaciones, así como los atributos de las mismas.

A partir de estas desarrollamos el diseño lógico que nos ha permitido describir la información como un conjunto de tablas, que a la vez contienen información organizada en tuplas.

En el siguiente capítulo se realiza la implementación de los diseños que se han elaborado en los capítulos III y IV; y los cuales generan el desarrollo de nuestro sistema.

CAPITULO V

IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA

En esta etapa, utilizamos todos los modelos y diagramas de las fases de análisis y diseño, y mediante las herramientas proporcionadas por la plataforma de implementación, aplicamos los procesos indicados en cada una de estas, obteniendo la realización de nuestro sistema.

Así también, en esta misma etapa, aplicamos una serie de pruebas a los módulos y al sistema en general, lo que nos permite asegurar el funcionamiento esperado para el mismo.

2.1. Implementación

Para la implementación del Sistema de Control Escolar, en razón de la aplicación de un sentido práctico y a la facilidad que otorgan algunos sitios WEB de diseño, se decidió utilizar como base de la interfaz de usuario, una plantilla WEB elaborada mediante la tecnología de Hojas de Estilo en Cascada (CSS - Cascade Style Shet).

Esta plantilla fue descargada del sitio www.freecsstemplates.org. El cual está dedicado a elaborar plantillas de páginas WEB mediante CSS, y de las que se puede hacer uso de manera gratuita, cumpliendo con algunos requisitos de licenciamiento, siendo el más importante: el dejar indicado en cada página de la que se haga uso, quién es su diseñador.

Así mismo, tomando en consideración que estamos desarrollando un sistema WEB, este se encontrará formado por un conjunto de páginas WEB, las cuales partirán de la plantilla mencionada con anterioridad.

Además se empleó el lenguaje HTML (Hypertext Markup Languaje – Lenguaje de marcado de Hipertexto) para desarrollar los elementos de la

interfaz, lo que permite a los usuarios interactuar con el sistema, mostrar los mensajes de aviso y error que emite el mismo y los resultados de las operaciones solicitadas por los operadores.

Cabe también mencionar, que en la creación de funcionalidad de las reglas de negocio se utilizó el lenguaje PHP, mismo que permitió programar las partes fundamentales como son: validaciones de datos, control de flujo de operaciones, conexiones y consultas a la BD, implementación de los objetos, así como sus métodos y atributos.

Para la parte de la implementación de la BD de nuestro sistema, hicimos uso del Sistema de Gestión de Base de Datos MySQL, el cual facilitó la creación de la BD, sus tablas y atributos, y proporcionó los mecanismos para la manipulación de los datos.

Debido a las condiciones de diseño del sistema, iniciamos con la implementación de la BD, ya que uno de los puntos centrales del sistema es el acceso al mismo, se registró de inicio un usuario en la BD.

2.1.1. Base de Datos

La transformación del diseño lógico de la BD a nuestro modelo físico, es mediante la elaboración de un script de instrucciones MySQL, que nos proporcionan la base de datos con todas las características diseñadas en capítulos anteriores.

El script de implementación es el siguiente:

```
DROP DATABASE IF EXISTS controlescolar;  
CREATE DATABASE controlescolar;  
  
USE controlescolar;  
  
DROP TABLE IF EXISTS USUARIO;  
CREATE TABLE USUARIO (  
idUsuario INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
nomUsuario VARCHAR(20),  
password VARCHAR(15));
```

```
DROP TABLE IF EXISTS ALUMNO;
CREATE TABLE ALUMNO (
idAlumno INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
aPaterno VARCHAR(15) NOT NULL,
aMaterno VARCHAR(15) NOT NULL,
nomAlumno VARCHAR(15) NOT NULL,
curp VARCHAR(20) NOT NULL,
calle VARCHAR(20),
numInt VARCHAR(6),
numExt VARCHAR(6),
colonia VARCHAR(40),
localidad VARCHAR(40),
municipio VARCHAR(40),
estado VARCHAR(20),
fechaNac VARCHAR(10) NOT NULL,
edad INT NOT NULL,
sexo VARCHAR(1) NOT NULL,
NIA INT,
telCasa VARCHAR(10),
status VARCHAR(14),
idUsuario INT,
FOREIGN KEY (idUsuario) REFERENCES USUARIO(idUsuario) ON UPDATE
CASCADE ON DELETE CASCADE );
```

```
DROP TABLE IF EXISTS PERSONAL;
CREATE TABLE PERSONAL (
idPersonal INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
aPaterno VARCHAR(15) NOT NULL,
aMaterno VARCHAR(15) NOT NULL,
nomPersonal VARCHAR(15) NOT NULL,
curp VARCHAR(20) NOT NULL,
calle VARCHAR(20),
numInt VARCHAR(6),
numExt VARCHAR(6),
colonia VARCHAR(20),
localidad VARCHAR(20),
municipio VARCHAR(15),
estado VARCHAR(20),
categoria VARCHAR(8),
plaza VARCHAR(6),
profesion VARCHAR(40),
tipoPersonal VARCHAR(14),
telCasa VARCHAR(10),
status VARCHAR(13),
idUsuario INT,
FOREIGN KEY (idUsuario) REFERENCES USUARIO (idUsuario) ON UPDATE
CASCADE ON DELETE CASCADE );
```

```
DROP TABLE IF EXISTS MATERIA;
CREATE TABLE MATERIA (
idMateria INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
codMateria INT NOT NULL,
nomMateria VARCHAR(50) NOT NULL,
especialidad VARCHAR(40) );
```

```
DROP TABLE IF EXISTS GRADO;
CREATE TABLE GRADO (
idGrado INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
nivel INT );
```

```
DROP TABLE IF EXISTS GRUPO;
CREATE TABLE GRUPO(
idGrupo INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
letraGrupo VARCHAR(1) );
```

```
DROP TABLE IF EXISTS SEMESTRE;
CREATE TABLE SEMESTRE (
idSemestre INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
nomSemestre VARCHAR(10) );
```

```
DROP TABLE IF EXISTS CICLOESCOLAR;
CREATE TABLE CICLOESCOLAR (
idCiclo INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
alni VARCHAR(10),
aFin VARCHAR(10),
descripcion VARCHAR(25) );
```

```
DROP TABLE IF EXISTS HORARIO;
CREATE TABLE HORARIO (
idHorario INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
lunIni VARCHAR(5),
lunFin VARCHAR(5),
marIni VARCHAR(5),
marFin VARCHAR(5),
mieIni VARCHAR(5),
mieFin VARCHAR(5),
jueIni VARCHAR(5),
jueFin VARCHAR(5),
vieIni VARCHAR(5),
vieFin VARCHAR(5),
descripcion VARCHAR(50) );
```

```
DROP TABLE IF EXISTS REGISTROCURSO;
CREATE TABLE REGISTROCURSO (
NRC INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
idMateria INT NOT NULL,
idPersonal INT NOT NULL,
idCiclo INT NOT NULL,
idSemestre INT NOT NULL,
idHorario INT NOT NULL,
idGrado INT NOT NULL,
idGrupo INT NOT NULL,
status VARCHAR(10),
FOREIGN KEY (idMateria) REFERENCES MATERIA(idMateria) ON UPDATE
CASCADE ON DELETE CASCADE,
FOREIGN KEY (idPersonal) REFERENCES PERSONAL(idPersonal) ON
UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (idCiclo) REFERENCES CICLOESCOLAR(idCiclo) ON
UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (idSemestre) REFERENCES SEMESTRE(idSemestre) ON
UPDATE CASCADE ON DELETE CASCADE,
```

```
FOREIGN KEY (idHorario) REFERENCES HORARIO(idHorario) ON UPDATE  
CASCADE ON DELETE CASCADE,  
FOREIGN KEY (idGrado) REFERENCES GRADO(idGrado) ON UPDATE  
CASCADE ON DELETE CASCADE,  
FOREIGN KEY (idGrupo) REFERENCES GRUPO(idGrupo) ON UPDATE  
CASCADE ON DELETE CASCADE );
```

```
DROP TABLE IF EXISTS INSCRIPCION;  
CREATE TABLE INSCRIPCION (  
NRC INT NOT NULL,  
idAlumno INT NOT NULL,  
FOREIGN KEY (NRC) REFERENCES REGISTROCURSO(NRC) ON  
UPDATE CASCADE ON DELETE CASCADE,  
FOREIGN KEY (idAlumno) REFERENCES ALUMNO(idAlumno) ON UPDATE  
CASCADE ON DELETE CASCADE,  
PRIMARY KEY (NRC,idAlumno) );
```

```
DROP TABLE IF EXISTS CALIFICACIONES;  
CREATE TABLE CALIFICACIONES (  
idCalificacion INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
califMom1 INT,  
califMom2 INT,  
califMom3 INT,  
califFinal INT,  
asistMom1 INT,  
asistMom2 INT,  
asistMom3 INT,  
porcenAsist INT,  
NRC INT NOT NULL,  
idAlumno INT NOT NULL,  
FOREIGN KEY (NRC) REFERENCES INSCRIPCION(NRC) ON UPDATE  
CASCADE ON DELETE CASCADE,  
FOREIGN KEY (idAlumno) REFERENCES INSCRIPCION(idAlumno) ON  
UPDATE CASCADE ON DELETE CASCADE );
```

2.1.2. Interfaz del Sistema

2.1.2.1. Página de Inicio

Partiendo de la plantilla elegida como base de nuestro sistema para la interfaz, se desarrolló la siguiente página de inicio del sistema.

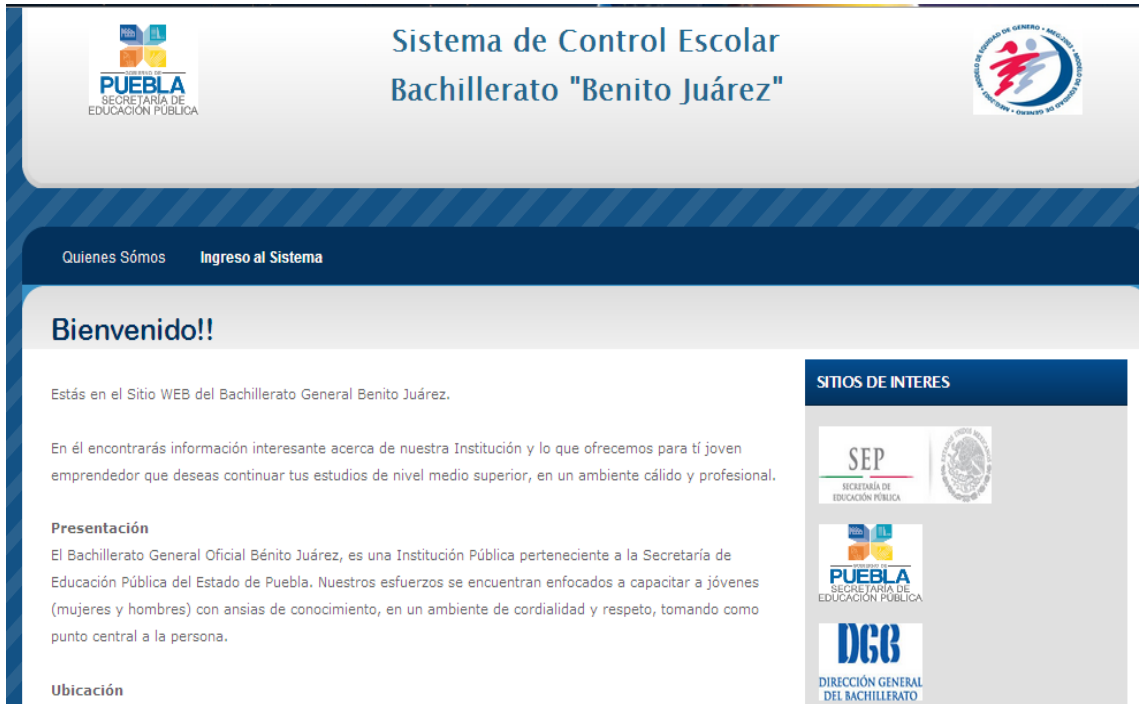


Fig. 5.1. Página inicio (¿Quiénes somos?)

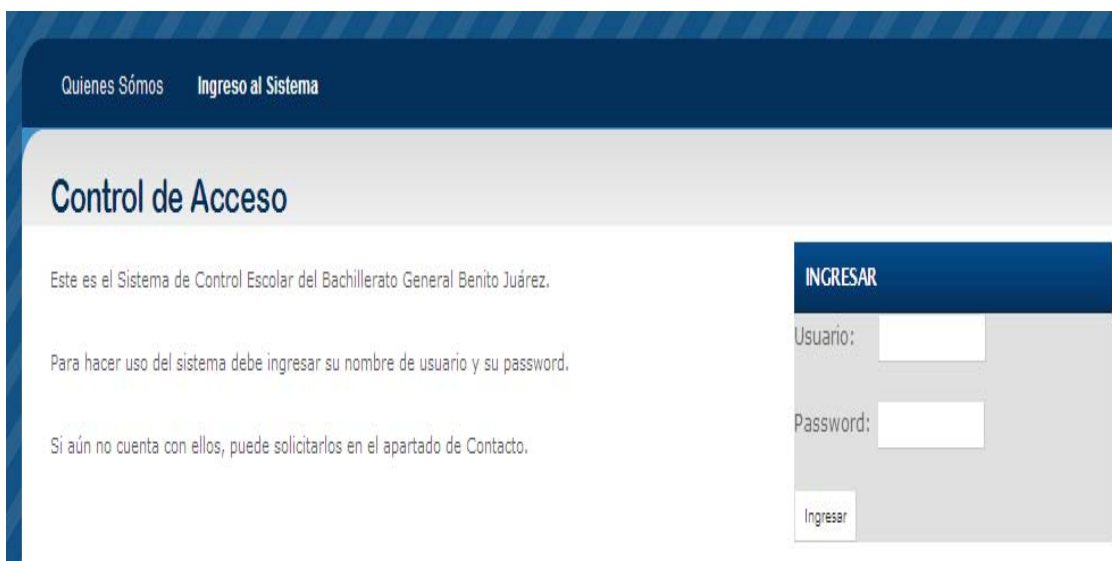


Fig. 5.2. Página inicio (Control de Acceso)

La anterior está compuesta por dos secciones:

- Quienes somos.- describe información general acerca del Bachillerato.
- Ingreso al sistema.- proporciona la interfaz a una persona registrada, para que ingrese su usuario y contraseña accediendo a las funcionalidades del mismo.

2.1.2.2. Interfaz del usuario

Si una persona está registrada e ingresa su usuario y contraseña de manera correcta, accede a las funcionalidades que ofrece el sistema.

Las siguientes imágenes muestran las diferentes herramientas para el usuario:



Fig. 5.3. Página inicial del sistema

Inicio.- Esta sección muestra la página principal de las funciones, en ella se describe el objetivo de las herramientas.

En la sección Academia, se encuentran las funciones para trabajar con Alumnos, Docentes y Administrativos, estas permiten ingresar datos, buscar, eliminar y modificar información referente a los mismos.

Las figuras siguientes muestran estas funcionalidades:



Fig. 5.4. Menú Academia



Fig. 5.5. Función Consultar Maestros (Filtros)

Sistema de Control Escolar de un Bachillerato General

Inicio Academia ▾ Registro ▾ Documentos ▾ Parámetros ▾

Maestros

Consulta – Datos personales

NO.	NOMBRE	CURP
1	Calderon Canizal Luis Miguel	CACL750418HPLJKS01
2	Cortes Barrientos Nereida	COBN600523MPLSDF00

Fig. 5.6. Función Consultar Maestros

Inicio Academia ▾ Registro ▾ Documentos ▾ Parámetros ▾ Usuarios Salir

Ingresar Alumno

DATOS PERSONALES

APELLIDO PATERNO	APELLIDO MATERNO	NOMBRE
<input type="text"/>	<input type="text"/>	<input type="text"/>

CURP	FECHA NAC.	EDAD	SEXO	NIA
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

DOMICILIO

CALLE	NO. INT.	NO. EXT.	COLONIA
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

LOCALIDAD	MUNICIPIO	ESTADO	TELEFONO
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

OPERACIONES

- ASIGNAR GRADO Y GRUPO
- ACTUALIZAR
- CONSULTAR
- DAR DE BAJA
- INGRESAR
- REGRESAR

Fig. 5.7. Función Ingresar Alumnos

Fig. 5.8. Función Modificar Administrativos (Filtros)

ACT. NO.	NOMBRE	CURP
<input type="radio"/> 1	Cortes Barrientos Pilar	COBP650912MPLHRS09
<input type="radio"/> 2	Lima Jimenez Manuel	LIJM750211HPLSDR03
<input type="radio"/> 3	Perez Salazar Claudia	PESC710512MPLTYU00
<input type="radio"/> 4	Perez Arrieta Guillermo	PEAG730625HPLRRL07
<input type="radio"/> 5	Rodriguez Perez Cesar	ROPC760723HPLMRS01

Fig. 5.9. Función Modificar Administrativos (Selección)

Sistema de Control Escolar Bachillerato "Benito Juárez"

Inicio Academia Registro Documentos Parámetros Usuarios Salir

Personal Administrativo

Actualizar – Datos personales

APELLIDO PATERNO	APELLIDO MATERNO	NOMBRE	CURP
Lima	Jimenez	Manuel	LJM750211HPLSDR03

Actualizar

Fig. 5.10. Función Modificar Administrativos (Modificar)

Inicio Academia Registro Documentos Parámetros Usuarios Salir

Baja de Alumnos

Para dar de baja a un Alumno debe seleccionar uno o más criterios de búsqueda e ingresar el valor correspondiente.

Nota: Si no se ingresa algún valor, por defecto se mostrarán todos los Alumnos Activos en el sistema.

Criterios de Búsqueda

<input type="checkbox"/>	Nombre	<input type="text"/>
<input type="checkbox"/>	CURP	<input type="text"/>
<input type="checkbox"/>	NIA	<input type="text"/>
<input type="checkbox"/>	Calificación	<input type="text"/>
<input type="checkbox"/>	Grado	1 <input type="text"/>
<input type="checkbox"/>	Grupo	A <input type="text"/>
<input type="checkbox"/>	Ciclo Escolar	Ciclo Escolar 2010-2011 <input type="text"/>

Buscar

Fig. 5.11. Dar de baja Alumnos (Filtros)

Inicio Academia Registro Documentos Parámetros Usuarios Salir

Baja de Alumnos

Seleccione los Alumnos a dar de baja:

BAJA	NO.	NOMBRE	FECHA NAC.	EDAD	SEXO	NIA	GRADO	GRUPO	SEMESTRE	CICLO ESCOLAR	ESTATUS
<input checked="" type="checkbox"/>	1	Copalca Teja Alma	12-05-1992	20	F	101347	2	B	Segundo	Ciclo Escolar 2012-2013	
<input checked="" type="checkbox"/>	2	Galicia Camargo Coral	12-08-1995	16	F	101017	1	A	Primer	Ciclo Escolar 2010-2011	NUEVO INGRESO
<input type="checkbox"/>	3	Gonzalez Tellez Ricardo	15-06-1992	19	M	100917					NUEVO INGRESO
<input checked="" type="checkbox"/>	4	Hernandez Robles Alejandro	10-09-1996	16	M	101089					NUEVO INGRESO
<input checked="" type="checkbox"/>	5	Jimenez Contreras Sergio	12-10-1997	16	M	160967	1	A	Primer	Ciclo Escolar 2012-2013	NUEVO INGRESO
<input type="checkbox"/>	6	Luna Sarmiento Omar	11-02-1997	15	M	135917					NUEVO INGRESO

Fig. 5.12. Dar de baja Alumnos (Selección)

Inicio Academia Registro Documentos Parámetros Usuarios Salir

Baja de Alumnos

Seleccione los Alumnos a dar de baja:

BAJA	NO.	NOMBRE	FECHA NAC.	EDAD	SEXO	NIA	GRADO	GRUPO	SEMESTRE	CICLO ESCOLAR	ESTATUS
<input type="checkbox"/>	1	Amaro Ramirez Yaneli	28-08-1998	16	F	114577	3	A	Tercer	Ciclo Escolar 2012-2013	BAJA
<input type="checkbox"/>	2	Carcamo Miranda Laura	15-03-1996	16	F	115617	1	A	Primer	Ciclo Escolar 2010-2011	BAJA
<input type="checkbox"/>	3	Chiquito Rojas Hipolito	23-11-1996	16	M	101916	1	B	Primer	Ciclo Escolar 2012-2013	BAJA
<input type="checkbox"/>	4	Copalca Teja Alma	12-05-1992	20	F	101347	2	B	Segundo	Ciclo Escolar 2012-2013	BAJA
<input type="checkbox"/>	5	Galicia Camargo Coral	12-08-1995	16	F	101017	1	A	Primer	Ciclo Escolar 2010-2011	BAJA

Fig. 5.13. Dar de Baja Alumnos

En la sección de Registro, es posible asentar Calificaciones y Asistencias a los grupos de Alumnos en la materia elegida y para un ciclo escolar determinado.

Las funciones se muestran a continuación:

Calificaciones Captura

Para realizar la captura o actualización de Calificaciones debe seleccionar uno o más criterios de búsqueda e ingresar el valor correspondiente.

Por Grupo

GRADO	1
GRUPO	A
MATERIA	Algebra
SEMESTRE	Primer
CICLO ESCOLAR	Ciclo Escolar 2010-2011

Por Alumno

<input type="checkbox"/> Nombre	
<input type="checkbox"/> CURP	
<input type="checkbox"/> NIA	
<input type="checkbox"/> GRADO	1
<input type="checkbox"/> GRUPO	A

Fig. 5.14. Capturar Calificaciones (Filtros)

Calificaciones Captura y Actualización

NO.	CURP	NOMBRE	MATERIA	1ER. MOMENTO	2DO. MOMENTO	3ER. MOMENTO	FINAL
1	CAML960315MPLMNG09	Carcamo Miranda Laura	Algebra	0	0	0	0
2	AURA960508MPLGJN05	Rojas Aguirre Anayeli	Algebra	0	0	0	0

Ingresar

Fig. 5.15. Capturar Calificaciones (Captura)

Sistema de Control Escolar Bachillerato "Benito Juárez"

Inicio Academia ▾ Registro ▾ Documentos ▾ Parámetros ▾ Usuarios Salir

Consultas Calificaciones

NO.	CURP	NOMBRE	1ER. MOMENTO	2DO. MOMENTO	3ER. MOMENTO	FINAL	MATERIA
1	CAML960315MPLMNG09	Carcamo Miranda Laura	10	8	9	9	Algebra
2	AURA960508MPLGJN05	Rojas Aguirre Anayeli	5	6	7	6	Algebra

Fig. 5.16. Capturar Calificaciones (Resultados)

Inicio Academia ▾ Registro ▾ Documentos ▾ Parámetros ▾ Usuarios Salir

Asistencias Consulta

Para realizar un consulta debe seleccionar uno o más criterios de búsqueda e ingresar el valor correspondiente.

Por Grupo

GRADO 1 ▾

GRUPO A ▾

MATERIA Algebra ▾

SEMESTRE Primer ▾

CICLO ESCOLAR Ciclo Escolar 2010-2011 ▾

Por Alumno

Nombre

CURP

NIA

GRADO 1 ▾

GRUPO A ▾

MATERIA Algebra ▾

SEMESTRE Primer ▾

CICLO ESCOLAR Ciclo Escolar 2010-2011 ▾

Fig. 5.17. Consulta Asistencias (Filtros)

NO.	CURP	NOMBRE	1ER. MOMENTO	2DO. MOMENTO	3ER. MOMENTO	TOTAL	MATERIA
1	CHRH961123HPLRL01	Chiquito Rojas Hipolito	0	0	0	0	Educacion Fisica
2	JICS971012HPLRSL09	Jimenez Contreras Sergio	19	18	19	98	Educacion Fisica
3	MIAH980223HPLRLL00	Miraflores Azueta Hugo	19	19	18	98	Educacion Fisica
4	POPA970925MPLLP01	Portugal Perez Aracely	25	24	25	98	Educacion Fisica
5	SALM950511HPLRLL03	Sanchez Lopez Manuel	19	19	19	100	Educacion Fisica

Fig. 5.18. Consulta Asistencias (Resultados)

El apartado Documentos permite generar una serie de información de la situación académica de los Alumnos, todo esto en los formatos predeterminados, los cuales pueden ser Boletas de calificaciones, Listas de Asistencia y Certificados de Estudios.

Las imágenes siguientes muestran la interfaz de estas funciones:

Documentos
Genera Boletas

Para generar boletas debe seleccionar uno o más criterios de búsqueda e ingresar el valor correspondiente.

Por Grupo

GRADO: 1

GRUPO: A

CICLO ESCOLAR: Ciclo Escolar 2012-2013

Por Alumno

Nombre

CURP

NIA

GRADO: 1

GRUPO: A

CICLO ESCOLAR: Ciclo Escolar 2012-2013

Fig. 5.19. Generar Boletas (Filtros)

The figure shows two screenshots of a web application interface for generating student result sheets (Boletas). Both screenshots are for the 'BACHILLERATO GRAL. OF. BENITO JUAREZ' with 'CLAVE: 21EBH0173Q'.

Top Screenshot:

- Header: PUEBLA SECRETARIA DE EDUCACION PUBLICA, SEMS SECRETARIA DE EDUCACION PUBLICA.
- Form fields: CICLO ESCOLAR: Ciclo Escolar 2012-2013; GRADO: 2; GRUPO: B; NOMBRE DEL ALUMNO: Alma Copalosa Teja.
- Table:

NO.	ASIGNATURA	CALIFICACION	% DE ASISTENCIAS	OBSERVACIONES
1	Educacion Artistica	0	0	
PROMEDIO		0.0		

- Signature area: FIRMA ENTERADO, NOMBRE DEL TUTOR, EL DIRECTOR, NOMBRE DEL DIRECTOR.

Bottom Screenshot:

- Header: PUEBLA SECRETARIA DE EDUCACION PUBLICA, SEMS SECRETARIA DE EDUCACION PUBLICA.
- Form fields: CICLO ESCOLAR: Ciclo Escolar 2012-2013; GRADO: 2; GRUPO: B; NOMBRE DEL ALUMNO: Sergio Jimenez Contreras.
- Table:

NO.	ASIGNATURA	CALIFICACION	% DE ASISTENCIAS	OBSERVACIONES
1	Educacion Artistica	9	19	
PROMEDIO		9.0		

- Signature area: FIRMA ENTERADO, NOMBRE DEL TUTOR, EL DIRECTOR, NOMBRE DEL DIRECTOR.

Fig. 5.20. Generar Boletas (Resultados)

The screenshot shows the 'Genera Listas' interface in a web application. The navigation bar includes: Inicio, Academia, Registro, Documentos, Parámetros, Usuarios, Salir.

Documentos
Genera Listas

Para generar Listas de Grupos debe seleccionar uno o más criterios de búsqueda e ingresar el valor correspondiente.

Por Grupo

- GRADO:** 1
- GRUPO:** A
- MATERIA:** Algebra
- CICLO ESCOLAR:** Ciclo Escolar 2012-2013
- FECHA INICIO:** [Empty field]

Nota: La fecha deberá tener el formato dd-mm-aaaa (Donde d= día, m = mes y a=año).

Fig. 5.21. Generar Listas de Grupo (Filtros)



Fig. 5.23. Actualizar Ciclos Escolares (Filtros)



Fig. 5.24. Crear Cursos (Filtros)



Fig. 5.25. Actualizar Grupos (Selección)



Fig. 5.26. Actualizar Grados (Modificar)

Ingresar Horario

DÍA	INICIO	FIN
LUNES		
MARTES		
MIERCOLES		
JUEVES		
VIERNES		
DESCRIPCION		

Guardar

Fig. 5.27. Crear Horario (Capturar)

Materias Consultas

CÓDIGO	MATERIA	ESPECIALIDAD
1010	Educacion Fisica	
1050	Educacion Artistica	
3010	Estadistica	
3020	Orientacion Profesiografica	
3030	Form. Prop.	Desarrollo Humano
3040	Form. Prop.	Pensamiento Critico y Creativo
3050	Cap.P/Trab.	Sist. Tec. Imp.
3016	Cap.P/Trab.	Impl. Proy.
1020	Algebra	
1030	Ingles I	
1040	Etica y Valores I	

Fig. 5.28. Consultar Materias



Fig. 5.29. Eliminar Semestre (Selección)

La segunda sección de la interfaz de administración proporciona las funciones para manejar Usuarios, estos usuarios permiten el acceso al sistema a personas registradas para realizar tareas de control escolar. Su interfaz es la siguiente:



Fig. 5.30. Crear Usuarios (Captura)



Fig. 5.31. Consultar Usuarios (Resultados)



Fig. 5.32. Consultar Usuarios (Resultados)

2.2. Pruebas

Para la verificación de la funcionalidad del sistema se aplicaron una serie de pruebas que nos permiten comprobar tanto la funcionalidad de los módulos como las interfaces con el usuario, a través de estas se detectaron y corrigieron las fallas presentadas por el sistema.

Así mismo, muestran que el sistema está realizando las funciones para las que fue diseñado e implementado.

Las pruebas que se realizaron fueron:

De caja blanca: las cuales están enfocadas a probar módulos del sistema a nivel de código, mediante la verificación de datos a través de los posibles caminos que puede tomar la información en cada módulo.

De caja negra: estas se están destinadas a probar el sistema de manera más integral, desde el punto de vista de la interfaz con los usuarios, ingresando datos y verificando que el sistema realiza las funciones solicitadas.

A continuación se describen algunas de las pruebas realizadas para el sistema:

2.2.1. Caja Blanca

Para realizar esta prueba elegimos un módulo específico del sistema, elaboramos nuestro caso de prueba y realizamos la verificación del mismo, como se muestra a continuación:

Caso de prueba 1.-

Modulo: verificar usuario

Descripción: No se ingresa un usuario y se ingresa el password:

Roguepe, por tanto la casilla de nombre de usuario se encuentra

vacía. El modulo verifica que se haya ingresado información en los cuadros de texto usuario y password.

Resultado esperado: no hay datos en el cuadro de texto usuario, el sistema envía un mensaje de Error: No se ingresó ningún usuario.

Código:

Vista

<!--Se ingresa un usuario y password en los cuadros de texto del formulario y se envía para su validación-->

<h4>Ingresar</h4>

```
<form action="valida_usuario.php" method="post">
```

```
    Usuario
```

```
    <input type="text" name="usuario" size="15" maxlength="15" />
```

```
    <br/>
```

```
    Password:
```

```
        <input type="password" name="password" size="15"
```

```
        maxlength="15" />
```

```
    <input type="submit" value="Ingresar"/>
```

```
</form>
```

Negocio

<!--Se inicia una sesión, se ponen a disposición del módulo las clases conexión y usuarios, se reciben los datos de los cuadros de texto usuario= “ ” y password = “Rogupe”, se valida si se encuentran vacíos.-->

```
<?php
```

```
    session_start();
```

```
    if(!$_SESSION["k_username"])
```

```
        header("Location: Index.php");
```

```
    require_once("Clases/Conexion.php");
```

```
    require_once("Clases/Usuarios.php");
```

```
    $usuario=$_REQUEST[""]; 
```

```
    $password=$_REQUEST['Rogupe'];
```

```
    if ($usuario=="")
```

```
{
    $_SESSION['mensaje']="No ingreso ning&uacute;n usuario";
    header("Location: ../../vista/errores/error1.php");
}
....
....
$enlace->cerrarConexion();
}
?>
```

Resultado obtenido: El sistema envía un mensaje de Error: No se ingresó ningún usuario.

Caso de prueba 2.-

Modulo: verificar usuario

Descripción: Se ingresan los datos usuario: **roger**, password: **Rogupe**, el usuario no se encuentra registrado en el sistema. El modulo realiza una conexión a la BD y busca usuario y password en la tabla Usuarios.

Resultado esperado: no se encuentra el usuario, el sistema envía un mensaje de Error: el usuario no existe.

Código:

Vista

<!--Se ingresa un usuario y password en los cuadros de texto del formulario y se envía para su validación-->

<h4>Ingresar</h4>

*...
...
</form>*

Negocio

<!--Se inicia una sesión, se ponen a disposición del módulo las clases conexión y usuarios, se reciben los datos de los cuadros de texto usuario="roger" y password="Rogupe", se valida si se encuentran vacíos o si los datos están registrados en la BD, se muestran errores o se ingresa al sistema.-->

```
<?php
...
...
$usuario=$_REQUEST['roger'];
$password=$_REQUEST['Rogupe'];

if ($usuario=="")
{
    $_SESSION['mensaje']="No ingreso ning&uacute;n usuario";
    header("Location: ../../vista/errores/error1.php");
}
else if ($password=="")
{
    $_SESSION['mensaje']="No ingreso ning&uacute;n de
password";
    header("Location: ../../vista/errores/error1.php");
}
else
{
    $enlace = Conexion::obtenerInstancia();
    $user = new Usuarios(roger,Rogupe,$enlace);
    if($user->validaUsuario())
    {
        if($user->validaPassword())
            $_SESSION["k_username"]      =
            $usuario;
            header("Location: InicioAdm.php");
        }
        else
            {
                $_SESSION['mensaje']="El password
no existe";
                header("Location:../../vista/errores/erro
r1.php");
            }
    }
    else
        {
```

```
$_SESSION['mensaje']="El usuario no existe";  
header("Location:  
../../vista/errores/error1.php");  
}
```

...

?>

Resultado obtenido: El sistema envía un mensaje de Error: El usuario no existe.

Caso de prueba 3.-

Modulo: verificar usuario

Descripción: Se ingresan los datos usuario: **pegui**, password: **Gupear**, el usuario está registrado en el sistema. El modulo realiza una conexión a la BD y busca usuario y password en la tabla Usuarios.

Resultado esperado: el usuario y el password son válidos, se da acceso al usuario y se muestra la página de inicio del sistema.

Código:

Vista

<!--Se ingresa un usuario y password en los cuadros de texto del formulario y se envía para su validación-->

<h4>Ingresar</h4>

...

...

...

</form>

Negocio

<!--Se inicia una sesión, se ponen a disposición del módulo las clases conexión y usuarios, se reciben los datos de los cuadros de texto usuario y password, se valida si se encuentran vacíos o si los datos están registrados en la BD, se muestran errores o se ingresa al sistema.-->

<?php

...

...

```
...
$usuario=$_REQUEST['pegui'];
$password=$_REQUEST['Gupear'];

if ($usuario=="")
{
    $_SESSION['mensaje']="No ingreso ning&uacute;n usuario";
    header("Location: ../../vista/errores/error1.php");
}
else if ($password=="")
{
    $_SESSION['mensaje']="No ingreso ning&uacute;n de
password";
    header("Location: ../../vista/errores/error1.php");
}
else
{
    $enlace = Conexion::obtenerInstancia();
    $user = new Usuarios(pegui,Gupear,$enlace);
    if($user->validaUsuario())
    {
        if($user->validaPassword())

        $_SESSION["k_username"] = pegui;
        header("Location: InicioAdm.php");
    }
    ...
    ...
    ...
    $enlace->cerrarConexion();
}
?>
```

Resultado obtenido: El sistema muestra la pantalla de inicio.

2.2.2. Caja Negra

Para realizar esta prueba seleccionamos la funcionalidad a probar, elaboramos el caso de prueba correspondiente y realizamos la verificación de los resultados obtenidos, algunas de las pruebas se muestran a continuación:

Caso de prueba 1:

Función: Actualizar Alumnos

Descripción: Se selecciona el Alumno(a): Mayra García Palacios y se modifican el Nombre y Apellido.

Resultado esperado: en una ventana se muestran los resultados de la modificación: Nombre: ALICIA Apellido Paterno: MORALES del alumno seleccionado.

Pantallas: Las pantallas muestran el proceso mediante el que se realiza la prueba.



Fig. 5.33. Se ingresa a la sección Alumnos

Actualiza datos de Alumnos

Para actualizar datos de un Alumno debe seleccionar el Estatus de Alumnos y los Datos a mostrar. Seleccionar uno o más criterios de búsqueda e ingresar el valor correspondiente.

Nota: Si no se ingresa algún valor, por defecto se mostrarán los Alumnos Activos en el sistema.

Estado

Activos Bajas Todos

Datos a Mostrar

Personales Académicos Domiciliares

Criterios de Búsqueda

Nombre

CURP

NIA

Calificación

Grado

Grupo

Ciclo Escolar

Fig. 5.34. Búsqueda de Alumno mediante criterios

Actualizar datos de Alumnos

Seleccione el Alumno de quién desea actualizar sus datos:

ACT. NO.	NOMBRE	CURP	FECHA NAC.	EDAD	SEXO
<input type="radio"/> 1	Copaldua Teja Alma	COTA921205MPLHKI00	12-05-1992	20	F
<input type="radio"/> 2	Galicia Camargo Coral	GACC950812MPLJSD08	12-08-1995	16	F
<input checked="" type="radio"/> 3	Garcia Palacios Mayra	GAPM960525MPLYTO07	25-05-1995	16	F
<input type="radio"/> 4	Gonzalez Tellez Ricardo	GOTR920615HPLPLJ00	15-06-1992	19	M
<input type="radio"/> 5	Hernandez Robles Alejandro	HERA960910HPLJKL00	10-09-1996	16	M
<input type="radio"/> 6	Jimenez Contreras Sergio	JICS971012HPLRSL09	12-10-1997	16	M
<input type="radio"/> 7	Luna Sarmiento Omar	LUSO970211HPLMFD01	11-02-1997	15	M
<input type="radio"/> 8	Miraflores Azueta Hugo	MIAH980223HPLRLL00	23-02-1998	17	M
<input type="radio"/> 9	MORA SUASTEGUI GABRIEL	MOSGA301007HVZORU0	30-10-2007	5	M
<input type="radio"/> 10	Ortiz Miguel Francisco	RIMF950720HPLLLT02	20-07-1995	16	H

Fig. 5.35. Selección del Alumno que se quiere Actualizar

**Sistema de Control Escolar
Bachillerato "Benito Juárez"**

Inicio Academia Registro Documentos Parámetros Usuarios Salir

Modificar Datos Personales del Alumno

APELLIDO PATERNO	APELLIDO MATERNO	NOMBRE
MORALES	Palacios	ALICIA

CURP	FECHA NAC.	EDAD	SEXO
GAPM960525MPLYTO07	25-05-1995	16	F

Actualizar

Fig. 5.36. Se modifica el Apellido Paterno: MORALES y el Nombre: ALICIA

Inicio Academia Registro Documentos Parámetros Usuarios Salir

Actualizar datos de Alumnos

Seleccione el Alumno de quién desea actualizar sus datos:

ACT.	NO.	NOMBRE	CURP	FECHA NAC.	EDAD	SEXO
<input type="radio"/>	1	Amaro Ramirez Yaneli	AMRA940213MPLTRY00	28-08-1998	16	F
<input type="radio"/>	2	Carcamo Miranda Laura	CAML960315MPLMNG09	15-03-1996	16	F
<input type="radio"/>	3	Chiquito Rojas Hipolito	CHRH961123HPLRL01	23-11-1996	16	M
<input type="radio"/>	4	Copalca Teja Alma	COTA921205MPLHKI00	12-05-1992	20	F
<input type="radio"/>	5	Galicia Camargo Coral	GACC950812MPLJSD08	12-08-1995	16	F
<input type="radio"/>	6	Gonzalez Tellez Ricardo	GOTR920615HPLPLJ00	15-06-1992	19	M
<input type="radio"/>	7	Hernandez Robles Alejandro	HERA960910HPLJKL00	10-09-1996	16	M
<input type="radio"/>	8	Jimenez Contreras Sergio	JICS971012HPLRSL09	12-10-1997	16	M
<input type="radio"/>	9	Luna Sarmiento Omar	LUSO970211HPLMFD01	11-02-1997	15	M
<input type="radio"/>	10	Miraflores Azueta Hugo	MIAH980223HPLRLL00	23-02-1998	17	M
<input type="radio"/>	11	MORA SUASTEGUI GABRIEL	MOSGA301007HVZORU00	30-10-2007	5	M
<input checked="" type="radio"/>	12	MORALES Palacios ALICIA	GAPM960525MPLYTO07	25-05-1995	16	F
<input type="radio"/>	13	Ortiz Miguel Francisco	RIMF950720HPLLLT02	20-07-1995	16	H
<input type="radio"/>	14	Palalia Rojas Camilo	PARC950630HPLRTG01	30-06-1995	16	M

Fig. 5.37. Muestra el resultado de la Actualización de datos del Alumno.

Caso de prueba 2:

Función: Ingresar Curso

Descripción: Se genera un nuevo Curso con los siguientes datos:
Materia: Biología II, Maestro: Nereida Cortés Barrientos, Grado: 2º, Grupo: C, Semestre: Segundo, Ciclo Escolar: 2012-2013, Horario: Biología II 2C.

Resultado esperado: en una ventana se muestra el listado de cursos en el sistema con los datos del nuevo curso.

Pantallas: Las pantallas muestran el proceso mediante el que se realiza la prueba.



Fig. 5.38. Ingresar a la Sección Cursos

Crear Curso

MATERIA: Biología II

PROFESOR: Nereida Cortes Barrientos

GRADO: 2

GRUPO: C

SEMESTRE: Segundo

CICLO ESCOLAR: Ciclo Escolar 2012-2013

HORARIO: Biología II 2C

Guardar

Fig. 5.39. Elegir los datos para crear el curso

Sistema de Control Escolar Bachillerato "Benito Juárez"

Inicio Academia Registro Documentos Parámetros Usuarios Salir

Cursos

MATERIA	PROFESOR	GRADO	GRUPO	SEMESTRE	CICLO	HORARIO
Biología II	Nereida Cortes Barrientos	2	C	Segundo	Ciclo Escolar 2012-2013	Biología II 2C
Algebra	Luis Miguel Calderon Canizal	3	A	Primer	Ciclo Escolar 2012-2013	Algebra 3 A

Fig. 5.40. Muestra lista con curso creado

Los ejemplos descritos con anterioridad muestran los métodos que se utilizaron en la aplicación de pruebas al sistema. Cabe destacar que dichos ejemplos representan solo una pequeña muestra del conjunto de pruebas que se realizaron al mismo.

CONCLUSIONES

El proyecto presentado nos proporcionó una visión general de la problemática que se presenta en un Bachillerato General Estatal, se analizaron las necesidades más demandantes y se eligieron las funciones principales para desarrollar un software que permita solucionarlas.

Se observó que la aplicación de una metodología para el desarrollo de software, nos proporciona una serie de instrumentos, que nos permiten: crear sistemas de manera más ordenada, estructurar adecuadamente las tareas que se realizan, administrar de forma más eficiente los cambios requeridos y mejorar el control de errores que se presenten; obteniendo un software de mejor calidad y facilidad para su mantenimiento, crecimiento o adaptación.

Analizamos que el paradigma Orientado a Objetos nos provee los métodos para modelar problemas de la realidad y transformarlos a conceptos más cercanos a la computadora; facilitándonos la labor de análisis y diseño del software; y logrando mejorar los niveles de reutilización y mantenimiento del código.

Se encontró que el empleo de UML representa una gran herramienta de apoyo para la aplicación del paradigma Orientado a Objetos, ya que facilitó los procesos de análisis y diseño del sistema, mediante la creación de diseños que mostraban las ideas expresadas en lenguaje natural, permitiendo una mejor comprensión del sistema a desarrollar.

Las características tanto iterativa como incremental de la metodología del UP, así como las fases que la componen, proporcionaron un esquema de trabajo estructurado que permitió delimitar las tareas y los productos a obtener en cada etapa atendida, funcionando como una excelente guía para el desarrollo del sistema.

El uso y aplicación de la teoría de diseño de BD, nos proporcionó los procesos adecuados para modelar y diseñar la BD de nuestro sistema, evitando la redundancia de datos y elaborando un esquema detallado de la información necesaria, así como el cuidado de la integridad de la misma; proporcionando una BD funcional.

La plataforma WAMP para la implementación de un sistema WEB, nos proporcionó una serie de herramientas muy útiles y de fácil manejo, que permitieron transformar los modelos y diseños elaborados en etapas preliminares de la metodología UP, en pequeños sistemas o prototipos de software, permitiéndonos incrementar los mismos hasta lograr el software planificado.

Para el desarrollo de un software de calidad es muy importante que las personas involucradas no escatimen tiempo ni esfuerzos en la selección de una metodología adecuada para el desarrollo del mismo, ya que proporciona un marco de trabajo ampliamente organizado para la consecución de los objetivos planteados.

De igual manera, es recomendable emplear una plataforma de implementación con ventajas tanto técnicas como económicas, ya que estas facilitan la labor de del desarrollador de software.

De las actividades realizadas se concluye que se lograron los objetivos planteados en el proyecto, al obtener un sistema WEB que proporciona las funciones necesarias para lograr que las tareas de Control Escolar se realicen de manera más eficiente.

Así mismo, al emplear tecnologías de bajo costo en su desarrollo, permitió implementar el sistema con un margen de inversión menor a las aplicaciones comerciales ofrecidas en el mercado, beneficiando su implantación en la Institución.

PERSPECTIVAS

Las tareas de Control Escolar en las Instituciones se incrementan cada vez más, por tanto es necesario que los sistemas creados para solventar estas funciones tengan la facilidad de crecer para adaptarse a las nuevas necesidades de las mismas.

El sistema que se desarrolló cuenta con características para poder crecer en funcionalidades de una manera organizada y bajo una metodología que permite controlar las modificaciones con un mínimo grado de riesgo.

La interfaz del usuario es susceptible de ser mejorada, agregando aspectos multimedia que desarrollen un aspecto más atractivo para el usuario.

Debido a que la naturaleza de las características del sistema cumple con los requerimientos en materia de Control Escolar para un Bachillerato General, este puede ser replicado en cualquier escuela del nivel medio superior, con un mínimo de adecuaciones.

Así también, el sistema puede ser tomado como base para la implementación en otros niveles educativos de Instituciones públicas, con la adaptación de las funcionalidades respectivas.

BIBLIOGRAFÍA

1. Roger Pressman, Ingeniería del Software, un enfoque práctico, Quinta Edición, Ed. Mc Graw Hill.
2. Kendall & Kendall, Análisis y diseño de sistemas, Tercera Edición, Ed. Prentice Hall.
3. Craigh Larman, UML y Patrones, Introducción al análisis y diseño orientado a objetos, Cuarta Edición, Ed. Prentice Hall.
4. Dolors Costal Costa, Introducción al diseño de Bases de Datos, P06/M2109/02150, Universidad Oberta de Cataluña, España.
5. Miguel Rodríguez, Notas del diplomado Modulo I, Analisis y Diseño Orientado a Objetos, Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla.
6. Alma Delia Ambrosio Vázquez, Notas del diplomado Modulo II. Diseño de Bases de Datos, Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla.
7. Juan Carlos Orós, Diseño de páginas Web con XHTML, JavaScript y CSS, Tercera Edición, Ed. Alfaomega Ra-Ma.
8. Abraham Gutiérrez Ginés Bravo, PHP5 a través de ejemplos, Primera Edición, Editorial Alfaomega Ra-Ma.

Webliografía

9. Administración escolar
<http://www.monografias.com/trabajos93/la-administracion-educativa/la-administracion-educativa.shtml>
10. Control Escolar
<http://www.buenastareas.com/ensayos/Control-Escolar/153571.html>
11. Enlace al sitio del software Coleg-ERP
<http://www.coleg-erp.com/>
12. Página al sitio de School Manager
<http://www.sistemexico.net/sistemas-para-escuelas/school-manager-software/>
13. Enlace al sitio del software mi-escuela.com

<http://www.mi-escuela.com/>

14. Metodologías de desarrollo de software

http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software

15. Proceso Unificado (UP)

http://es.wikipedia.org/wiki/Proceso_Unificado

16. Lenguaje Unificado de Modelado (UML)

http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

17. Bases de datos

<http://www.monografias.com/trabajos72/base-datos/base-datos2.shtml#ixzz2Lh3ECK9L>

18. Aplicaciones WEB vs Escritorio

<http://www.osdglobal.com/faq/desarrollo-software/comparativo-web-vs-escritorio>

19. Aplicaciones WEB vs Cliente-Servidor

<http://www.systemsbyrr.com/aplicaciones-web-vs-cliente-servidor/>

20. WAMP

<http://www.wampserver.com/en/>

21. Windows

http://es.wikipedia.org/wiki/Microsoft_Windows

22. Windows

<http://definicion.de/windows/#ixzz2Ln3CE0rB>

23. Apache

http://es.wikipedia.org/wiki/Servidor_HTTP_Apache

24. MySQL

<http://es.wikipedia.org/wiki/MySQL>

25. PHP

<http://es.wikipedia.org/wiki/PHP>