



**BENEMÉRITA UNIVERSIDAD
AUTÓNOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA
COMPUTACIÓN**

**SISTEMA DE CONTROL ESCOLAR
DEL INSTITUTO DE CIENCIAS
SOCIALES Y HUMANIDADES
(ICSyH)**

TESIS

**QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN**

PRESENTA:

ANA LUISA REYES ARAIZA

ASESORA:

**DRA. MARÍA JOSEFA SOMODEVILLA
GARCÍA**

PUEBLA, PUEBLA NOVIEMBRE DE 2009

RESUMEN

El presente proyecto tiene como finalidad la creación de un Sistema de Control Escolar para el Instituto de Ciencias Sociales y Humanidades (ICSyH), el cual servirá a los usuarios para llevar el control de los alumnos del instituto.

Para llevar a cabo la creación de este proyecto se utilizó una metodología, la cual nos permitirá tener resultados satisfactorios. La metodología utilizada está basada en el Ciclo de Vida de Cascada el cual presenta las siguientes fases: Requerimientos, Análisis, Diseño, Implementación y Pruebas.

El análisis de requerimientos se utiliza para identificar de manera correcta las diferentes funciones que realizará el Sistema. Para una mejor comprensión de estas funciones se utilizó una herramienta basada en UML la cual nos permitió hacer diagramas para así poder representar de una manera más sencilla estas funciones.

En la fase de diseño se desarrolló una base de datos basada en el modelo Entidad-Relación, la cual facilita la comprensión de todos los elementos de la base de datos como lo son las entidades, los atributos y las relaciones que existen entre ellos.

Para la fase de la implementación del sistema se utilizó diferentes herramientas como: MySQL que nos sirven para la manipulación de la Base de Datos y PHP como software de capa media.

Después de este proceso se procedió a realizar las pruebas pertinentes para comprobar que el Sistema cumple con las funciones antes especificadas de una manera correcta y eficiente, como lo son la manipulación de los datos de alumnos y trabajadores del instituto.

AGRADECIMIENTOS

Esta tesis esta dedicada a mis Padres, quienes me infundieron la ética que guía mi transitar por la vida y a quienes agradezco de todo corazón por su amor, su comprensión y ayuda incondicional. En todo momento los llevo conmigo, sé que cuento con ellos siempre.

Agradezco a mis hermanos por su compañía y el apoyo que me brindan.

Agradezco a Dios por llenar mi vida de dicha y bendiciones.

Agradezco a mis amigos por su confianza y lealtad.

Agradezco a mi tutora de tesis por su disposición y ayuda brindadas.

ÍNDICE

INTRODUCCIÓN	1
ESTRUCTURA DEL DOCUMENTO	2
CAPÍTULO 1. MARCO TEÓRICO	3
1.1 ANTECEDENTES	3
1.2 INGENIERÍA DE SOFTWARE	4
1.2.1 Lenguaje de Modelado Unificado (UML)	4
1.3 MODELOS DEL CICLO DE VIDA	5
1.3.1 Modelo de Cascada	5
1.4 DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE	7
1.5 VERIFICACIÓN Y VALIDACIÓN DEL SOFTWARE	7
1.6 CONCEPTOS DE BASES DE DATOS	7
1.6.1 Definición de Bases de Datos	7
1.6.2 Componentes Principales	8
1.6.3 Ventajas en el Uso de las Bases de Datos	8
1.6.4 Sistema Gestor de Bases de Datos (SGBD)	9
1.6.5 Abstracción de Datos	9
1.6.6 Esquemas	10
1.6.7 Independencia de Datos	10
1.7 MODELOS DE DATOS	11
1.7.1 Modelo Entidad-Relación	11
1.7.1.1 Diagrama Entidad-Relación	11
1.7.1.2 Correspondencia de Cardinalidades	12
1.7.2 Modelo Relacional	12
1.8 CLAVES	12
1.9 MANIPULACIÓN DE LOS DATOS	12
1.10 NORMALIZACIÓN	13
1.10.1 Primera Forma Normal (1FN)	13
1.10.2 Segunda Forma Normal (2FN)	13
1.10.3 Tercera Forma Normal (3FN)	13
1.10.4 Forma Normal de Boyce-Codd (FNBC)	13
1.10.5 Cuarta Forma Normal (4FN)	13
1.10.6 Quinta Forma Normal (5FN)	14
1.11 MYSQL	14
1.12 LENGUAJES DE PROGRAMACIÓN WEB	15
1.12.1 Lenguaje HTML	15
1.12.2 PHP	16
1.13 SERVIDOR WEB	17
1.13.1 Wampserver	19

CAPÍTULO 2. ANÁLISIS DEL SISTEMA	20
2.1 INTRODUCCIÓN	20
2.1.1 Sistema de Control Escolar del ICsYH	20
2.2 DEFINICIÓN DEL PROBLEMA	20
2.3 REQUERIMIENTOS FUNCIONALES	21
2.4 REQUERIMIENTOS NO FUNCIONALES	21
2.5 MODELO DE CASOS DE USO	22
2.5.1 Diagramas de Caso de Uso	22
2.5.1.1 Especificación de Casos de Uso	30
CAPÍTULO 3. DISEÑO DEL SISTEMA	61
3.1 MODELO DE DATOS	61
3.2 DESCRIPCIÓN DE ENTIDADES Y ATRIBUTOS	61
3.3 RELACIONES ENTRE ENTIDADES	65
3.4 DIAGRAMA ENTIDAD-RELACIÓN	66
3.5 MODELO RELACIONAL	68
3.6 DESCRIPCIÓN DE TABLAS	69
3.7 NORMALIZACIÓN	73
CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBAS	79
4.1 HERRAMIENTAS UTILIZADAS	79
4.1.1 Navicat	79
4.1.2 Dreamweaver	79
4.1.3 Wampserver	80
4.2 IMPLEMENTACIÓN	80
4.2.1 Construcción de la Base de Datos	80
4.3 CODIFICACIÓN EN PHP	81
4.3.1 Conexión a la Base de Datos	81
4.3.2 Inserción	82
4.3.3 Consulta	82
4.3.4 Modificación	83
4.3.5 Eliminación	83
4.4 PROTOTIPOS DE INTERFAZ DE USUARIO	84
4.4.1 Menú Principal	84
4.4.2 Menú Administrador	85
4.4.3 Menú Secretaria	85
4.4.4 Menú Director	86
4.4.5 Consulta de Alumno	86
4.4.6 Alta de Alumno	87
4.5 PRUEBAS	88
CONCLUSIONES Y PERSPECTIVAS	93
BIBLIOGRAFIA	94

ÍNDICE DE FIGURAS

Fig. 1.1 Modelo de Cascada	6
Fig. 1.2 Representación Gráfica del Modelo Entidad Relación	11
Fig. 1.3 Servidor Web	18
Fig. 1.4 WAMP5 1.7.3	19
Fig. 2.1 Casos de Uso del Sistema General	22
Fig. 2.2 Casos de Uso que Controla el Actor Secretaria	23
Fig. 2.3 Casos de Uso Control de Alumnos	23
Fig. 2.4 Casos de Uso Control de Personal	24
Fig. 2.5 Casos de Uso Control de Aulas	24
Fig. 2.6 Casos de Uso Control de Pagos	24
Fig. 2.7 Casos de Uso Control de Sanciones	25
Fig. 2.8 Casos de Uso Control de Asignaturas	25
Fig. 2.9 Casos de Uso que Controla el Actor Administrador	26
Fig. 2.10 Casos de Uso Control de Edificios	26
Fig. 2.11 Casos de Uso Control de Materias	27
Fig. 2.12 Casos de Uso Control de Nivel Académico	27
Fig. 2.13 Casos de Uso Control de Plan de Estudios	27
Fig. 2.14 Casos de Uso Control Posgrados	28
Fig. 2.15 Casos de Uso Control de Tipo de Pago	28
Fig. 2.16 Casos de Uso Control de Tipo de Personal	28
Fig. 2.17 Casos de Uso Control de Tipo de Sanciones	29
Fig. 2.18 Casos de Uso Control de Tipo de Sangre	29
Fig. 2.19 Casos de Uso del Actor Director	29
Fig. 3.1 Diagrama Entidad-Relación	67
Fig. 4.1 Creación de Tablas en Navicat	80
Fig. 4.2 Vista de las Tablas en la Base de Datos	81
Fig. 4.3 Menú Principal del Sistema	84
Fig. 4.4 Menú Administrador	85
Fig. 4.5 Menú Secretaria	85
Fig. 4.6 Menú Director	86
Fig. 4.7 Consulta de Alumno	86
Fig. 4.8 Alta de Alumno	87
Fig. 4.9 Datos de Alta de Alumno	88
Fig. 4.10 Consulta de Alumno desde el Sistema	89
Fig. 4.11 Datos de Alumnos desde la Base de Datos	89
Fig. 4.12 Actualizar Datos Alumno	90
Fig. 4.13 Formulario de Actualización de Datos Alumno	91
Fig. 4.14 Datos Alumno Antes de Actualizar la Base de Datos	92
Fig. 4.15 Datos Alumno Después de Actualizar la Base de Datos	92

ÍNDICE DE TABLAS

Tabla 3.1 Relaciones entre Entidades	65
Tabla 3.2 Tabla Alumno	69
Tabla 3.3 Tabla Nivel Académico	69
Tabla 3.4 Tabla Personal	70
Tabla 3.5 Tabla Tipo de Personal	70
Tabla 3.6 Tabla Tipo de Sangre	70
Tabla 3.7 Tabla Pagos	70
Tabla 3.8 Tabla Tipo de Pago	71
Tabla 3.9 Tabla Sanciones	71
Tabla 3.10 Tabla Tipo de Sanción	71
Tabla 3.11 Tabla Aulas	71
Tabla 3.12 Tabla Materias	71
Tabla 3.13 Tabla Edificios	72
Tabla 3.14 Tabla Posgrado	72
Tabla 3.15 Tabla Dentro de Plan de Estudio	72
Tabla 3.16 Tabla Asignaturas	72

INTRODUCCIÓN

La disciplina de Ciencias de la Computación abarca el estudio de las bases teóricas de la información y la computación y su aplicación en sistemas computacionales.

Existen diversos campos dentro de la disciplina de ciencias de la computación, la cual está en constante evolución, cada vez abarca diferentes ámbitos, por eso es importante especializarse dentro de una de ellas, la presente tesis se enfoca dentro del campo de las bases de datos, en la cual entra en práctica la Ingeniería de Software.

El estudio de las bases de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido, una escuela puede considerarse una base de datos compuesta en su mayoría por datos de alumnos, personal administrativo, y calificaciones para su consulta.

En la actualidad, y debido al desarrollo tecnológico, la mayoría de las bases de datos están en formato digital, que ofrece un amplio rango de soluciones al problema de almacenar datos.

OBJETIVOS

OBJETIVO GENERAL

Desarrollar un Sistema de Control Escolar para el Instituto de Ciencias Sociales y Humanidades (ICSyH), dicho sistema tiene la finalidad de proporcionar apoyo a las áreas de control escolar, académico, docencia y cobranza, entre otras.

OBJETIVOS PARTICULARES

Con la estrategia correcta, la tecnología de vanguardia y personal altamente capacitado, el Sistema de Control Escolar del ICSyH respalda a la institución para enfrentar los retos educativos actuales y construir un sistema firme para el servicio de los alumnos, docentes y personal administrativos de una manera eficiente y versátil.

Además de contar con una herramienta eficaz para la administración escolar permite implementar estrategias para mejorar la calidad de los procesos administrativos.

ESTRUCTURA DEL DOCUMENTO

CAPÍTULO 1

El capítulo de Marco Teórico abarca fundamentalmente cuatro temas; la evolución de la Computación, Ingeniería de Software, Bases de Datos y Herramientas para la creación del proyecto.

CAPÍTULO 2

En el capítulo llamado Análisis del Sistema, encontramos tres temas: el primero es una introducción sobre el Sistema de Control Escolar del ICSyH, el segundo tema abarca el planteamiento del problema y también los requerimientos del sistema y por último los casos de uso relacionados al sistema.

CAPÍTULO 3

El capítulo denominado Diseño del Sistema conceptual, lógico y físico, trata de la creación de la base de datos para el sistema, por tanto se presentan las entidades, el diagrama Entidad-Relación, el proceso del mapeo al modelo relacional, así como también la normalización de la base de datos, asegurando así un correcto funcionamiento.

CAPÍTULO 4

En este capítulo titulado Implementación y Pruebas encontramos: las herramientas empleadas mostrándose las diferentes interfaces con ejemplos de su uso.

Finalmente, se presentan las conclusiones y perspectivas, seguidas de la bibliografía.

CAPÍTULO 1. MARCO TEÓRICO

En este capítulo se reunirá información documental para tener un diseño metodológico de la investigación, la cual proporcionará un conocimiento profundo de la teoría que le da significado a la investigación de la tesis. Es a partir de las teorías existentes acerca de la tesis, como pueden generarse nuevos conocimientos.

La validez interna y externa de una investigación se demuestra en las teorías que lo apoyan y, en esa medida, los resultados pueden generalizarse.

El marco teórico incluye conceptos que ayudaran al usuario a la comprensión y aprovechamiento del sistema.

1.1 ANTECEDENTES

Todos nosotros, alguna o muchas veces nos hemos enfrentado al desgastante trato con secretarías que nos obligan a dar un sinnúmero de vueltas para trámites administrativos. Desde pagar el teléfono o la luz, hasta inscribirnos en un curso de Posgrado.

En varios casos esto se debe al deficiente control y manejo de la información. Es decir, para que nos den una respuesta, hay que esperar, a que la forma, solicitud, o lo que hayamos presentado, de un x número de vueltas, pase por un sinnúmero de personas, tenga una enorme cantidad de sellos, antes de que nos digan si está o no autorizada, registrada, aprobada o lo que se requiera.

El manejo y control de la información de forma manual, para muchos de los trámites que realizamos hoy en día, se ha vuelto no sólo lento y problemático, sino obsoleto. El uso de las computadoras ha venido a simplificar y ordenar muchos de los procesos que cotidianamente realizamos. Su utilización hoy en día, en casi todos los ámbitos en donde nos desenvolvemos, es ya común. La optimización del control de información, la rapidez en la respuesta, la flexibilidad, entre otros aspectos, hace del uso de la computadora una herramienta indispensable.

En el Instituto de Ciencias Sociales y Humanidades (ICSyH), específicamente en lo relativo a la administración escolar, hasta 1992 se contaba con terminales de computadora UNISYS para realizar los procesos de inscripción e historiales académicos.

Aún cuando en su momento dio respuestas positivas, este sistema se volvió insuficiente y poco flexible para adecuarlo a las actuales necesidades específicas del Posgrado del ICSyH. Operativamente la información se encontraba centralizada, por lo que cualquier consulta que quisiera hacer la secretaria, sobre los datos de los alumnos, tenía que consultarlos directamente en los papeles del archivo.

A pesar de que la información se encontraba en bancos de datos como Excel, éste no permitía dar solución a los requerimientos cotidianos de información, ni a las modificaciones constantes de los contenidos de los Planes de Estudio del Posgrado

(cambios en carga académica, en valor de créditos, entre otros). La dinámica de este proceso no permitía la consulta y utilización de información con la rapidez requerida.

A partir de esta situación se comenzó a trabajar en el desarrollo de un sistema en microcomputadora, para ser utilizado en dicha institución.

1.2 INGENIERÍA DE SOFTWARE

Desde que se acuñará el término de Ingeniería de Software en la conferencia de la OTAN en 1968 [1], se ha recorrido un largo camino en el que se han ido incorporando nuevos campos de investigación y desarrollo bajo el amparo del término general, que cubren de una manera excepcional cada una de las frases que componen el ciclo de vida del software.

Se encuentran diferentes definiciones de Ingeniería de Software, a continuación citaremos algunas:

Ingeniería del Software es el establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico que sea fiable y funcione de manera eficiente sobre máquinas reales. Fritz Bauer, 1968.

Construcción de software multi-versión por un equipo de varias personas. David Parnas, 1987 [2].

En la actualidad podemos decir que la **Ingeniería de Software** designa el conjunto de técnicas destinadas a la producción de un programa de computadora, más allá de la sola actividad de programación. Forman parte de esta disciplina las ciencias computacionales y el manejo de proyectos, entre otros campos, propios de la rama más genérica denominada Ingeniería de Informática [3].

1.2.1 Lenguaje de Modelado Unificado (UML)

El **Lenguaje de Modelado Unificado (UML, Unified Modeling Language)** es un estándar propuesto para la creación de especificaciones de varios componentes de un sistema de software [4].

Una de las metas principales de UML es avanzar en el estado de la integración de las partes del sistema proporcionando herramientas de operabilidad para el modelado visual de objetos.

UML es ante todo un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica del sistema.

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✓ *Visualizar*: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ *Especificar*: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ✓ *Construir*: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ *Documentar*: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión [5].

Algunos de los diagramas de UML son:

Diagrama de casos de uso. Los diagramas de caso de uso muestran la interacción entre los usuarios y el sistema, este diagrama sirve para hacer un bosquejo a grandes rasgos del sistema, en específico los pasos de las tareas que realiza el usuario [6].

Diagrama de actividad. Los diagramas de actividad describen el flujo de tareas entre varios componentes de un sistema [7].

Diagrama de secuencia. Estos constan de objetos, mensajes y líneas de tiempo, en ellos se establece la interacción entre objetos por medio de mensajes a través del tiempo [8].

1.3 MODELO DEL CICLO DE VIDA

Un modelo de ciclo de vida es una descripción de los pasos que deben realizarse cuando se construye un producto de Software [1]. Debido a que casi siempre es más fácil llevar a cabo una secuencia de tareas más pequeña que una grande, el modelo del ciclo de vida total se divide en una serie de pasos más pequeños, denominados fases [9].

1.3.1 Modelo de Cascada

Llamado algunas veces “ciclo de vida básico” o “modelo lineal secuencial” es el paradigma más antiguo y más extensamente utilizado en la ingeniería de software.

Sugiere un enfoque secuencial para el desarrollo del software que comienza en un nivel de sistemas progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

El Modelo de Cascada que se muestra gráficamente en la Fig. 1.1 y comprende las siguientes actividades:

Análisis de los requisitos del software. El proceso de reunión de requisitos se intensifica y se centra especialmente en el software. Para comprender la naturaleza del (los) programa(s) a construirse, el ingeniero (<<analista>>) del software debe comprender el dominio de información del software, así como la función requerida, comportamiento, rendimiento e interconexión.

Diseño. El diseño del software es realmente un proceso de muchos pasos que se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, representaciones de interfaz y detalle procedimental (algoritmo). El proceso de diseño traduce requisitos en una representación del software donde se pueda evaluar su calidad antes de que comience la codificación.

Generación de código. El diseño se debe traducir en una forma legible por la máquina. El paso de generación de código lleva a cabo esta tarea. Si se lleva a cabo el diseño de una forma detallada, la generación de código se realiza mecánicamente.

Pruebas. Una vez que se ha generado el código, comienzan las pruebas del programa. El proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; es decir, realizar las pruebas para la detección de errores asegurar que la entrada definida produce resultados reales de acuerdo con los resultados requeridos.

Mantenimiento. El software indudablemente sufrirá cambios después de ser entregado al cliente (una excepción posible es el software empotrado). Se producirán cambios porque se han encontrado errores, porque el software debe adaptarse para acoplarse a los cambios de su entorno externo (por ejemplo: se requiere un cambio debido a un sistema operativo o dispositivo periférico nuevo), o porque el cliente requiere mejoras funcionales o de rendimiento. El soporte y mantenimiento del software vuelve a aplicar cada una de las fases precedentes a un programa ya existente y no a uno nuevo [9].

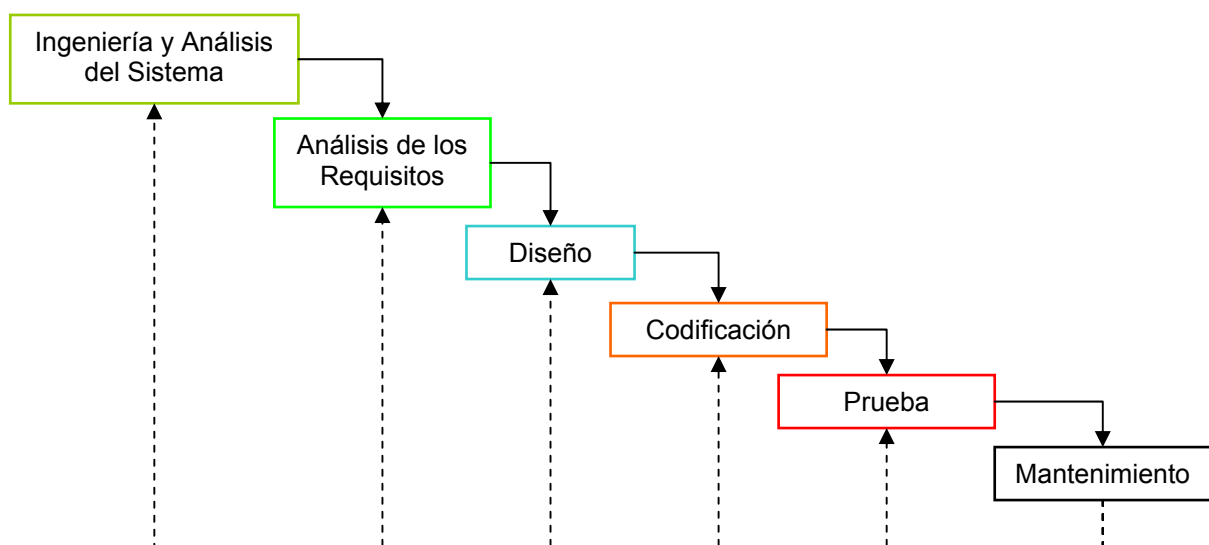


Fig. 1.1 Modelo de Cascada

1.4 DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE

En el Diseño las especificaciones atraviesan dos procesos de diseño consecutivos. Primero surge el diseño arquitectónico, en el cual producto como un todo se divide en sus componentes, denominados módulos. Después, se diseña cada módulo: este proceso se denomina diseño detallado. Los dos documentos de diseño resultantes describen “cómo lo hará el producto”. Más específicamente, su objetivo es afinar los artefactos del flujo del trabajo del análisis hasta que el material se encuentre en tal forma que pueda ser implementado por los programadores.

En la implementación cada uno de los diferentes componentes atraviesa por procesos de codificación y pruebas (pruebas unitarias). Después, los componentes del producto se combinan y prueban como un todo; esto se denomina integración. Cuando los desarrolladores están satisfechos de que el producto funciona correctamente, lo prueba el cliente (pruebas de aceptación). La fase de implementación termina cuando el cliente acepta el producto y se instala en su computadora.

1.5 VERIFICACION Y VALIDACIÓN DEL SOFTWARE

La verificación y validación del software (V y V) incluyen un conjunto de procedimientos, actividades, técnicas y herramientas que se utilizan paralelamente al desarrollo del software para asegurar que el producto resuelve el problema para el que fuera diseñado [9].

La V y V actúan sobre los productos intermedios intentando detectar y corregir cuanto antes sus defectos y desviaciones del objetivo si las hubiera. Las tareas que abarca son las siguientes: pruebas de verificación, revisiones y auditoría e incluyen las tareas de validación y pruebas de validación que se realizan durante el ciclo de vida del software para asegurar la satisfacción con los requisitos.

Cuando ya existe código ejecutable, se pueden realizar las pruebas del mismo, como verificación y validación del software. Las pruebas consisten en ejecutar el software con determinados datos de entrada y producir resultados que luego serán comparados con los teóricos.

1.6 CONCEPTOS DE BASES DE DATOS

1.6.1 Definición de Base de Datos

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso [10].

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular [11].

Una base de datos es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquinas accesibles en tiempo real y compatible con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo [12].

Las bases de datos proporcionan la infraestructura requerida para los sistemas de apoyo a la toma de decisiones y para los sistemas de información estratégicos, ya que estos sistemas explotan la información contenida en las bases de datos de la organización para apoyar el proceso de toma de decisiones o para ventajas competitivas. Por este motivo es importante conocer la forma en que están estructuradas las bases de datos y su manejo [13].

1.6.2 Componentes Principales

Datos. Los datos son la Base de Datos propiamente dicha.

Hardware. El hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.

Software. Está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DBMS: Data Base Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.

Usuarios. Existen cuatro clases de usuarios relacionados con una Base de Datos:

1. El diseñador de la Base de Datos, quien se encarga de identificar los datos que se almacenarán en la Base de Datos y elegir las estructuras apropiadas para la misma.
2. El programador de aplicaciones, quien crea programas de aplicación que utilizan la base de datos.
3. El usuario final, quien accesa a la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación.
4. El administrador de la Base de Datos (DBA: Data Base Administrador), quien se encarga del control general del Sistema de Base de Datos [14].

1.6.3 Ventajas en el Uso de Bases de Datos

Globalización de la información. Permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.

Eliminación de información redundante duplicada.

Eliminación de información inconsistente. Si el sistema está desarrollado a través de archivos convencionales.

Permite compartir información. Varios sistemas o usuarios pueden utilizar una misma entidad.

Permite mantener la integridad en la información. Solo se almacena la información correcta.

Independencia de datos. La independencia de datos implica un divorcio entre programas y datos; es decir, se pueden hacer cambios a la información que contiene la base de datos o tener acceso a la base de datos de diferente manera, sin hacer cambios en las aplicaciones o en los programas [14].

1.6.4 Sistema Gestor de Bases de Datos (SGBD)

Un Sistema Gestor de Bases de Datos (SGBD) son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan, consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. Su objetivo principal es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante, para un buen manejo de los datos [15].

1.6.5 Abstracción de Datos

Como se había mencionado, uno de los propósitos principales de un Sistema de Bases de Datos es proporcionar a los usuarios una visión *abstracta* de los datos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos.

Existen tres niveles de abstracción:

Nivel físico: el nivel más bajo de abstracción describe cómo se almacenan realmente los datos. En el nivel físico se describen a detalle las estructuras de datos complejas de bajo nivel.

Nivel lógico: el siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. Los administradores que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.

Nivel de vistas: el nivel más alto de abstracción describe sólo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. El sistema puede proporcionar muchas vistas para la misma base de datos [15].

1.6.6 Esquemas

La colección de información almacenada en la base de datos en un momento particular se denomina un ejemplar de la base de datos. El diseño completo de la base de datos se llama esquema.

Para que el sistema sea útil, debe recuperar los datos eficientemente. Como muchos usuarios de sistemas de bases de datos no están familiarizados con computadoras, los desarrolladores esconden la complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema.

Los esquemas son raramente modificados, si es que lo son alguna vez. Existen diferentes tipos de esquemas según los niveles de abstracción: físico, lógico y de vistas que se corresponden con los niveles físico, lógico y de vistas mencionados anteriormente.

1.6.7 Independencia de los Datos

La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella. Existen dos niveles de independencia de datos.

Independencia física de datos: Es la capacidad de modificar el esquema físico sin provocar que se vuelvan a escribir los programas de aplicación. Las modificaciones en el nivel físico son ocasionalmente necesarias para mejorar el funcionamiento.

Independencia lógica de datos: Capacidad de modificar el esquema conceptual sin provocar que se vuelvan a escribir los programas de aplicación. Las modificaciones en el nivel lógico son necesarias siempre que la estructura lógica de la base de datos se altere.

La independencia de datos lógica es más fácil de proporcionar que la independencia de datos física, ya que los programas de aplicación son fuertemente dependientes de la estructura lógica de los datos a los que ellos acceden.

El concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos. La independencia de datos implica un divorcio entre programas y datos [17].

1.7 MODELOS DE DATOS

Modelo de datos: es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia [18]. Los diferentes modelos de datos se clasifican en tres grupos diferentes:

- Modelos lógicos basados en objetos.
- Modelos lógicos basados en registros.
- Modelos físicos.

1.7.1 Modelo Entidad-Relación

El **modelo de datos Entidad-Relación (E-R)** está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades y de relaciones entre objetos:

- **Entidad:** es una cosa u objeto en el mundo real que es distinguible en todos los demás objetos.
- **Atributo:** nos sirve para describir las entidades en una base de datos.
- **Relación:** es una asociación entre varias entidades [11].

1.7.1.1 Diagrama Entidad-Relación

Es la representación gráfica de la estructura lógica general de una base de datos y sus componentes son:

- **Rectángulos:** que representan conjuntos de entidades.
- **Elipses:** que representan relaciones entre una entidad.
- **Rombos:** que representan relaciones entre conjuntos de entidades.
- **Líneas:** que en los atributos son los conjuntos de entidades y los conjuntos de entidades con las relaciones [11].

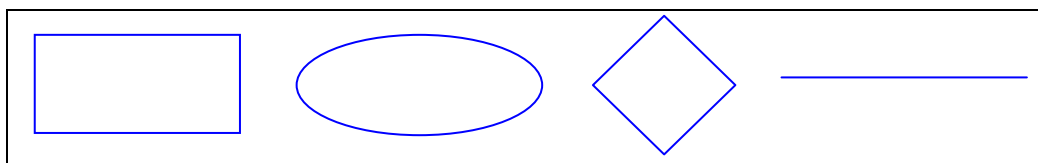


Fig. 1.2 Representación Gráfica del Modelo Entidad-Relación

1.7.1.2 Correspondencia de Cardinalidad

Es una restricción importante que expresa el número de entidades con las que otra entidad se puede asociar a través de un conjunto de relaciones.

1.7.2 Modelo Relacional

En el **modelo relacional** se utiliza un grupo de tablas para representar los datos y las relaciones entre ellos. Cada tabla está compuesta por varias columnas y cada columna tiene un nombre único [19].

1.8 CLAVES

Los valores de los atributos de una entidad deben ser tales que permitan identificar unívocamente a la entidad. En otras palabras no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos.

Una **clave** permite identificar un conjunto de atributos suficiente para distinguir las entidades entre sí.

Una **superclave** es un conjunto de uno o más atributos que tomados colectivamente permiten identificar de forma única una entidad en el conjunto de entidades.

Una **clave primaria** es una clave que es elegida por el diseñador de la base de datos como elemento principal para identificar las entidades dentro de un conjunto de entidades [19].

1.9 MANIPULACION DE LOS DATOS

Podemos definir la **manipulación de los datos** como:

- La recuperación de información almacenada en la base de datos.
- La inserción de información nueva en la base de datos.
- El borrado y modificación de información almacenada en la base de datos.

Para manipular la información utilizamos un lenguaje relacional, el más común para construir las consultas a bases de datos relacionales es **SQL** (Structured Query Language) o Lenguaje Estructurado de Consultas, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales [20].

1.10 NORMALIZACIÓN

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Es un proceso que clasifica relaciones, objetos, formas de relación y además elementos en grupos, en base a las características que cada uno posee. Si se identifican ciertas reglas, se aplica una categoría; si se definen otras reglas, se aplicará otra categoría. Cada regla está basada en la que le antecede [21].

Existen varios niveles de normalización:

1.10.1 Primera Forma Normal (1FN)

Una tabla está en Primera Forma Normal sólo si:

Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.

- ✓ La tabla contiene una clave primaria.
- ✓ La tabla no contiene atributos nulos.

Una columna no puede tener múltiples valores. Los datos son atómicos. (Si a cada valor de X le pertenece un valor de Y, entonces a cada valor de Y le pertenece un valor de X).

1.10.2 Segunda Forma Normal (2FN)

Dependencia Funcional Total. Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir, que no existen dependencias parciales. En otras palabras, podríamos decir que la segunda forma normal está completamente basada en el concepto funcional.

1.10.3 Tercera Forma Normal (3FN)

La tabla se encuentra en 3FN si es 2FN y cada atributo que no forma parte de ninguna clave, depende directamente y no transitivamente, de la clave primaria.

1.10.4 Forma Normal de Boyce-Codd (FNBC)

La tabla se encuentra en FNBC si está en 3FN y si cada determinante, atributo que determina completamente a otro, es clave candidata.

1.10.5 Cuarta Forma Normal (4FN)

Una tabla se encuentra en 4FN si está en 3FN, y sólo si, para cada una de sus dependencias múltiples no funcionales $X \twoheadrightarrow Y$, siendo X una superclave que, X es o una clave candidata o un conjunto de claves primarias.

1.10.6 Quinta Forma Normal (5FN)

Una tabla se encuentra en 5FN si: La tabla está en 4FN y no existen relaciones de dependencias no triviales que no siguen los criterios de las claves. Una tabla que se encuentra en 4FN se dice que está en la 5FN si, y sólo si, cada relación de dependencia se encuentra definida por las claves candidatas [22].

1.11 MYSQL

MySQL es un sistema gestor de bases de datos relacional, que además es muy utilizado en plataformas como PHP, Perl y HTML y funciones avanzadas de administración y optimización de bases de datos para facilitar las tareas habituales. Es un sistema cliente servidor de administración de bases de datos relacionales diseñado para el trabajo tanto en Windows como en UNIX/LINUX.

Podemos decir que MySQL es un sistema gestor de bases de datos relacional cliente-servidor de coste mínimo que incluye un servidor SQL, programas cliente para acceder al servidor, herramientas administrativas y una interfaz de programación para escribir programas, es portable y se ejecuta en sistemas operativos comerciales [23].

Existen cinco versiones de MySQL:

Estándar. Incluye el motor estándar y la posibilidad de usar bases de datos.

InnoDB. Todo el potencial de MySQL, pero sin soporte completo para utilizar transacciones.

Max. Para usuarios que quieren MySQL con herramientas de prueba para realizar opciones avanzadas de base de datos.

Pro. Versión comercial de MySQL estándar.

Classic. Igual que la estándar pero no dispone de soporte para InnoDB [24].

El uso de MySQL (excepto en la versión Pro) está sujeto a licencia GNU public license (llamada GLP). Esta licencia admite el uso de MySQL para crear cualquier tipo de aplicación.

El software MySQL tiene una doble licencia. Los usuarios pueden elegir entre usar el software MySQL como un producto Open Source bajo los términos de la licencia GNU General Public Licence o pueden adquirir una licencia comercial estándar de MySQL AB.

1.12 LENGUAJES DE PROGRAMACIÓN WEB

Existen diferentes lenguajes de programación para desarrollar en la Web, estos han sido surgiendo debido a las tendencias y necesidades de las plataformas.

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos. A continuación daremos una introducción a los diferentes lenguajes de programación para la Web.

1.12.1 Lenguaje HTML

Desde el surgimiento de Internet se han publicado sitios Web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Consortium (W3C). Los archivos pueden tener las extensiones (htm, html) [27].

Ventajas:

Sencillo que permite describir hipertexto.

Texto presentado de forma estructurada y agradable.

No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web o WYSIWYG.

Archivos pequeños.

Despliegue rápido.

Lenguaje de fácil aprendizaje.

Lo admiten todos los exploradores.

Desventajas:

Lenguaje estático.

La interpretación de cada navegador puede ser diferente.

Guarda muchas etiquetas que pueden convertirse en “basura” y dificultan la corrección.

El diseño es más lento.

Las etiquetas son muy limitadas.

1.12.2 PHP

PHP (Hypertext Pre-Processor) es un lenguaje de programación concebido principalmente como herramienta para el desarrollo de aplicaciones Web. Nos permite diseñar páginas dinámicas de servidor, es decir, páginas capaces de responder de manera inteligente a las demandas del cliente y nos permiten la automatización de gran cantidad de tareas.

PHP es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor [27].

PHP es un lenguaje muy fácil de aprender con respecto a otros lenguajes utilizados para el mismo propósito, como JAVA o ASP. Debido a esto no es necesario hacer un estudio muy concienzudo de sus funciones para realizar programas sencillos que nos resuelvan la mayoría de los problemas diarios.

Haciendo un poco de historia, PHP comenzó siendo un conjunto de scripts escritos en Perl que permitían a su creador, Rasmus Lerdorf, el control de los accesos a sus páginas personales. A este conjunto de scripts se les denominó como Personal Home Page Tools. En 1995 nació PHP/FI y a finales de 1997 se libera PHP/FI 2.0, en Junio de 1998 se liberó oficialmente PHP 3.0 siendo completamente reescrito por Andi Gutmans y Zeev Zuraski, en el PHP 4.0 la evolución consistió en la reescritura de su núcleo, dando lugar a un nuevo motor denominado Zend fue liberado en Mayo de 2000. La última y actual versión **PHP 5.0** fue liberada en Junio de 2004 está basada en el nuevo motor Zend 2, que cuenta entre sus características el completo soporte para la programación orientada a objetos, incorpora la gestión de excepciones, una nueva librería de XML y un soporte nativo para el sistema gestor de bases de datos SQLite y una nueva ampliación de MySQL.

PHP está construido de forma modular, esto quiere decir que todos sus componentes se ejecutan en el mismo espacio de memoria PHP, lo cual hace que el código PHP puede ejecutarse más rápidamente al no sustituir la sobrecarga impuesta por la comunicación con los diferentes objetos.

Haciendo así a este lenguaje el más utilizado en las aplicaciones Web, aunque debemos recordar que tiene otras aplicaciones [28].

Ventajas:

Muy fácil de aprender.

Se caracteriza por ser un lenguaje muy rápido.

Soporta en cierta medida la orientación a objetos: clases y herencia.

Es un lenguaje multiplataforma: Linux, Windows, entre otros.

Capacidad de conexión con la mayoría de los manejadores de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.

Capacidad de expandir su potencial utilizando módulos.

Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.

Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

Incluye gran cantidad de funciones.

No requiere definición de tipos de variable ni manejo detallado de bajo nivel.

Desventajas:

Se necesita instalar un servidor web.

Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.

La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.

La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.

Dificulta la modularización.

Dificulta la organización por capas de la aplicación.

Seguridad:

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

1.13 SERVIDOR WEB

Un servidor Web es un programa que implementa el protocolo HTTP (Hyper Text Transfer Protocol). Este protocolo pertenece a la capa de aplicación del modelo OSI y está diseñado para transferir lo que llamamos hipertextos, páginas Web o páginas HTML (HyperText Markup Lenguaje): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música [29].

Es un programa que se ejecuta continuamente en un ordenador (también se emplea el término para referirse al ordenador que lo ejecuta), manteniéndose a la espera de peticiones por parte de un cliente (un navegador web) y que responde a estas peticiones adecuadamente, mediante una página Web que se exhibirá en el navegador o mostrando el respectivo mensaje si se detectó algún error.

Instalar un servidor Web en nuestra PC nos permitirá, entre otras cosas, poder montar nuestra propia página web sin necesidad de contratar hosting, probar nuestros desarrollos vía local, acceder a los archivos de nuestro equipo desde un PC remoto. El problema de usar nuestro ordenador como servidor Web es que conviene tenerlo encendido permanentemente (para que esté accesible de forma continua como la mayoría de los sitios webs) con el consiguiente coste debido al consumo de electricidad.

Sobre el servicio web clásico podemos disponer de aplicaciones Web. Éstas son porciones de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

Aplicaciones en el lado del cliente: el cliente Web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java o Javascript, el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Comúnmente, los navegadores permiten ejecutar aplicaciones escritas en lenguajes javascript y java, aunque pueden añadirse más lenguajes mediante el uso de plugins.

Aplicaciones en el lado del servidor: el servidor Web ejecuta la aplicación, ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP [29].

Las aplicaciones de servidor muchas veces suelen ser la mejor opción para realizar aplicaciones Web. La razón es que, al ejecutarse ésta en el servidor y no en la máquina del cliente, éste no necesita ninguna capacidad añadida, como sí ocurre en el caso de querer ejecutar aplicaciones javascript o java. Así pues, cualquier cliente dotado de un navegador web básico puede utilizar este tipo de aplicaciones.

Un servidor Web nos ayudará a convertir cualquier PC convencional en un servidor de red que soporte hosting (alojamiento de páginas web), servidor de correo electrónico, servidor de FTP (para subir archivos para tu página), servidor IRC (servidor de chat), telnet, etc., como lo muestra la Fig. 1.3, incluso se tiene nuestro propio subdominio.

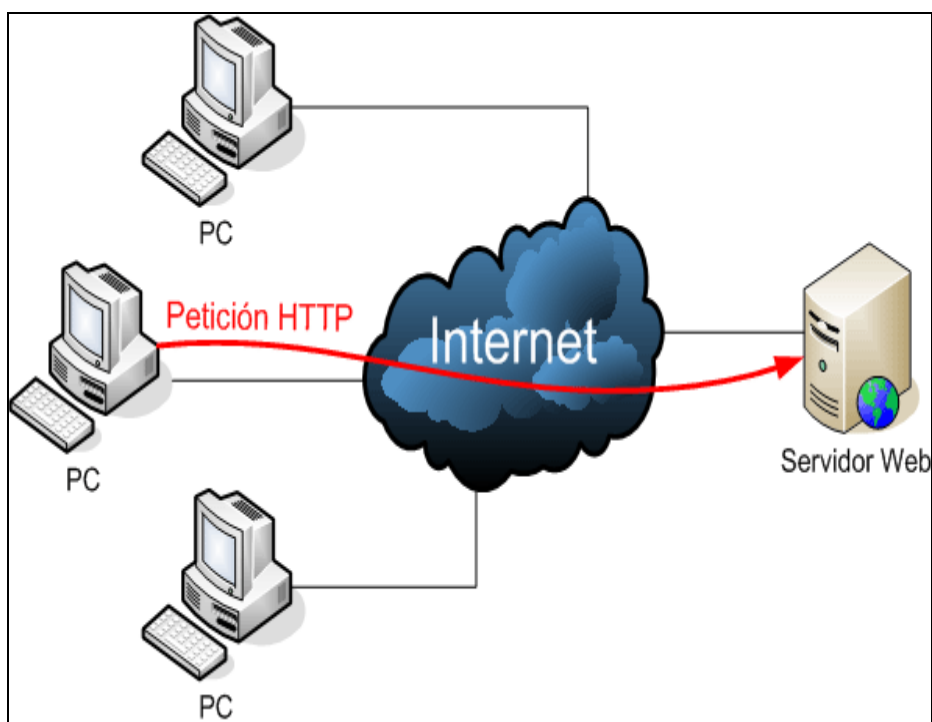


Fig. 1.3 Servidor Web

1.13.1 Wampserver

Wamp Server es un compendio de aplicaciones, que permiten tener un servidor Web en una PC. Facilita el trabajo de estar instalando independientemente Apache, PHP que después necesita adicionarse a Apache configurando sus archivos, mysql, y tb, trae PhpMyAdmin para poder configurar y manipular tus bases de datos vía web [30].

Una de las principales facilidades que trae es su interfaz amigable, que es multilinguaje, permitiendo acceder a los archivos de configuración directamente sin tener que irlos a buscar a las carpetas donde están instaladas. Permite adicionar extensiones en PHP con un solo clic, permite ponerse en modo compartido o modo local para que si no se quiere que externamente vean lo que se comparte vía web simplemente se pone local, y seguirá trabajando pero solo el usuario podrá verlo.

WAMP (Windows, Apache, MySQL, PHP) es el acrónimo para describir un conjunto de herramientas listas para ser instaladas fácilmente en nuestro servidor de internet.

La versión WAMP5 1.7.3 que se muestra en la Fig. 1.4 incorpora:

- Apache 2.2.4
- PHP 5.2.4
- MySQL 5.0.45
- SQLitemanager
- Phpmyadmin



Fig. 1.4 WAMP5 1.7.3

CAPÍTULO 2. ANÁLISIS DEL SISTEMA

2.1 INTRODUCCIÓN

Aún con su relativamente corta historia como disciplina académica formal, las Ciencias de la Computación han logrado una buena cantidad de contribuciones fundamentales a la ciencia y la sociedad. Por ejemplo, una definición formal de computación y de computabilidad, la demostración de que existen problemas a los que no hay una solución computacional, el concepto de lenguaje de programación, tecnologías revolucionarias y también ha habilitado nuevos tipos de investigación científica [31].

Existen diversos campos o disciplinas dentro de las Ciencias de la Computación, en este caso el enfoque son los problemas que requieren implementación de cómputo, utilizando la programación aplicada a los lenguajes de programación específicos para desarrollar una solución a un problema, en este caso un problema específico a resolver es la administración de una escuela dando como solución un Sistema de Control Escolar.

2.1.1 Sistema de Control Escolar

Si bien existe una gran diversidad de Sistemas de Control Escolar, pero ninguno realmente enfocado al Instituto de Ciencias Sociales y Humanidades, satisfaciendo las necesidades específicas que requiere dicha institución. Lo anterior es muy importante ya que es necesario poder tener el control de la planta administrativa y docente del instituto y así los usuarios interesados en el sistema sean capaces de tener un sistema de administración confiable para poder cubrir las necesidades del instituto. Esto nos ayudará a evitar que no se tenga la información necesaria de los alumnos y personal en el momento en que se requiere y contar con un sistema que brinde una interfaz amigable y fácil de utilizar para cualquier persona que tenga contacto con el sistema.

Día a día las instituciones de educación se enfrentan al desafío de ser más competitivas ofreciendo más y mejores opciones de educación con un alto grado de calidad y servicio. Esto trae consigo un nuevo reto a la tecnología, ya que las instituciones de educación se ven en la necesidad de buscar mecanismos que posibiliten un fácil manejo de la información de sus procesos y la pro actividad de las herramientas en cualquier lugar y en cualquier momento para ser más eficientes.

2.2 DEFINICIÓN DEL PROBLEMA

La tecnología ha probado ser una herramienta eficaz para la administración, abriendo nuevos horizontes al campo de la educación. Con el soporte del Sistema de Control Escolar, el Instituto de Ciencias Sociales y Humanidades se respalda a dicha institución para enfrentar los retos educativos actuales y construir un cambio firme para ofrecer servicios a alumnos, docentes y personal administrativos de una manera eficiente y versátil.

El Sistema de Control Escolar permite administrar y realizar los procesos de: alta, consulta, modificaciones y eliminaciones de los datos de la plata académica (control de alumnos, pagos, sanciones, planes de estudio, entre otras funciones).

También permite registrar y administrar toda la información referente a los alumnos del ICSyH. Apoya a la administración de la información de los alumnos.

El Sistema de Control Escolar en resumen:

- ✓ Facilita el trabajo de consulta para profesores y alumnos.
- ✓ Hace más eficiente el proceso de registro escolar.
- ✓ Proporciona los elementos para dar respuesta inmediata a docentes y alumnos.
- ✓ Se ahorra tiempos y esfuerzos en procesos.
- ✓ Permite una interactividad entre los módulos administrativos.

2.3 REQUERIMIENTOS FUNCIONALES

El Sistema de Control Escolar requiere que tenga las siguientes funcionalidades:

- El sistema debe llevar el control de edificios del instituto.
- En general el sistema tiene que llevar el control de pagos, personal, sanciones, nivel académico, tipos de sangre, las cuales puede agregar, modificar o eliminar y que dichos cambios se realicen en el sistema.
- Los usuarios podrán checar sanciones y pagos de cualquier alumno.
- En general el director puede hacer cualquier tipo de consulta.
- Se requiere que la secretaria tenga acceso a toda la información referente a los alumnos y profesores.
- El sistema debe llevar el control de pagos de los alumnos.
- Una nota importante es que para cada sesión el Sistema desplegará sólo el menú correspondiente para cada tipo de actor.

2.4 REQUERIMIENTOS NO FUNCIONALES

El Sistema de Control Escolar al estar en Web tiene la dificultad tal vez en el tiempo de respuesta a una petición de usuario ya que puede ser variable, y esto depende de circunstancias ajenas al sistema como lo sería la latencia y el ancho de banda de la red. Generalmente estas peticiones tardan segundos en realizarse, pero por lo mencionado anteriormente las operaciones requeridas por los usuarios pueden diferir en cuestión de algunos segundos.

2.5 MODELO DE CASOS DE USO

2.5.1 Diagramas de Caso de Uso

Para mejorar la definición de los requisitos en la realización de programas, utilizamos la creación de casos de uso, la cual es una descripción narrativa de los procesos. Fundamentalmente el modelo de casos de uso indica quien usa o interactúa con el sistema a desarrollar y como lo hace.

El análisis comprende aquellas actividades que parten de los casos de uso y de los requerimientos funcionales para producir un modelo de análisis del sistema. Mientras que el modelo de diseño es un modelo de objetos describiendo la realización de los casos de uso y sirve como una abstracción del modelo de implementación y código fuente. El modelo de diseño es usado como entrada para las actividades a realizar dentro de la implementación y pruebas del sistema.

Capturando los requerimientos y usando diagramas de casos de uso en UML los actores que intervienen en el sistema son: administrador, secretaria y director.

A continuación se presenta la representación de casos de uso del sistema en general en la figura 2.1, tomando los diagramas de casos de uso para realizar el análisis. Dichos diagramas se encuentran especificados en UML.

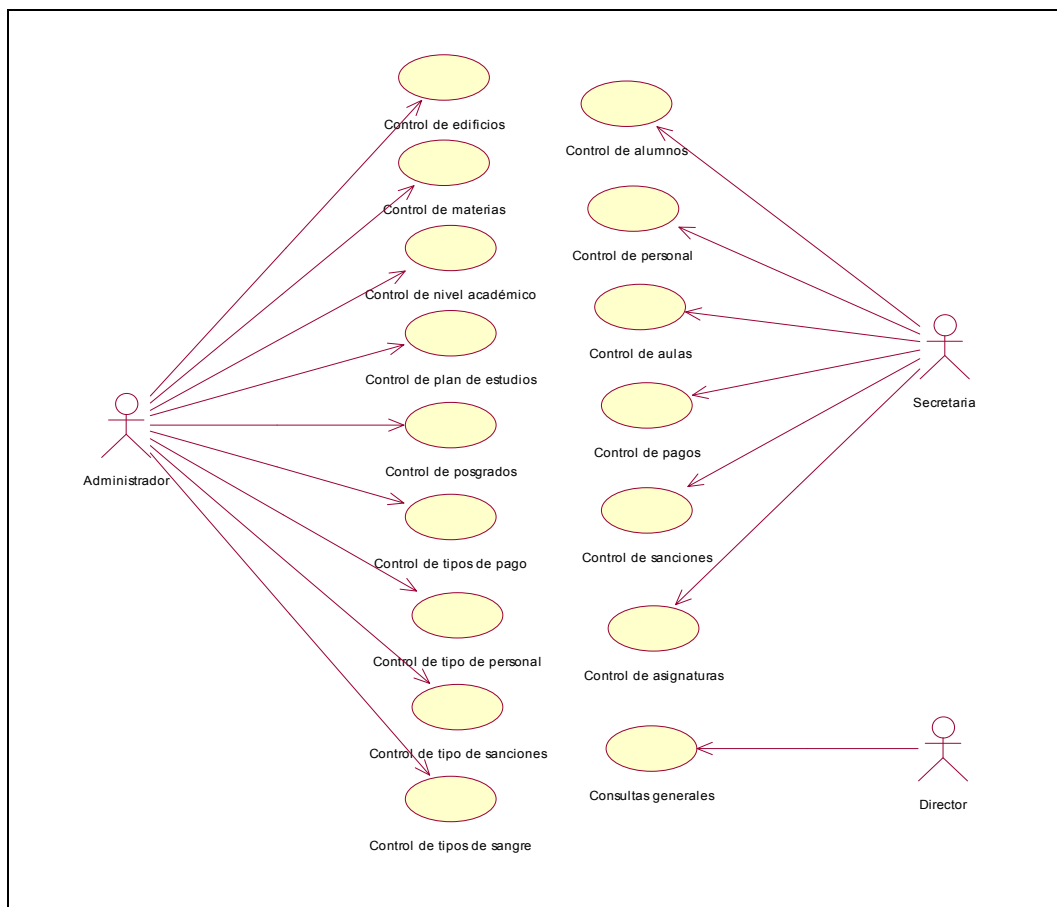


Fig. 2.1 Casos de Uso del Sistema General

El siguiente caso de uso es una forma más general de los casos de uso para el actor secretaria representado en la figura 2.2.

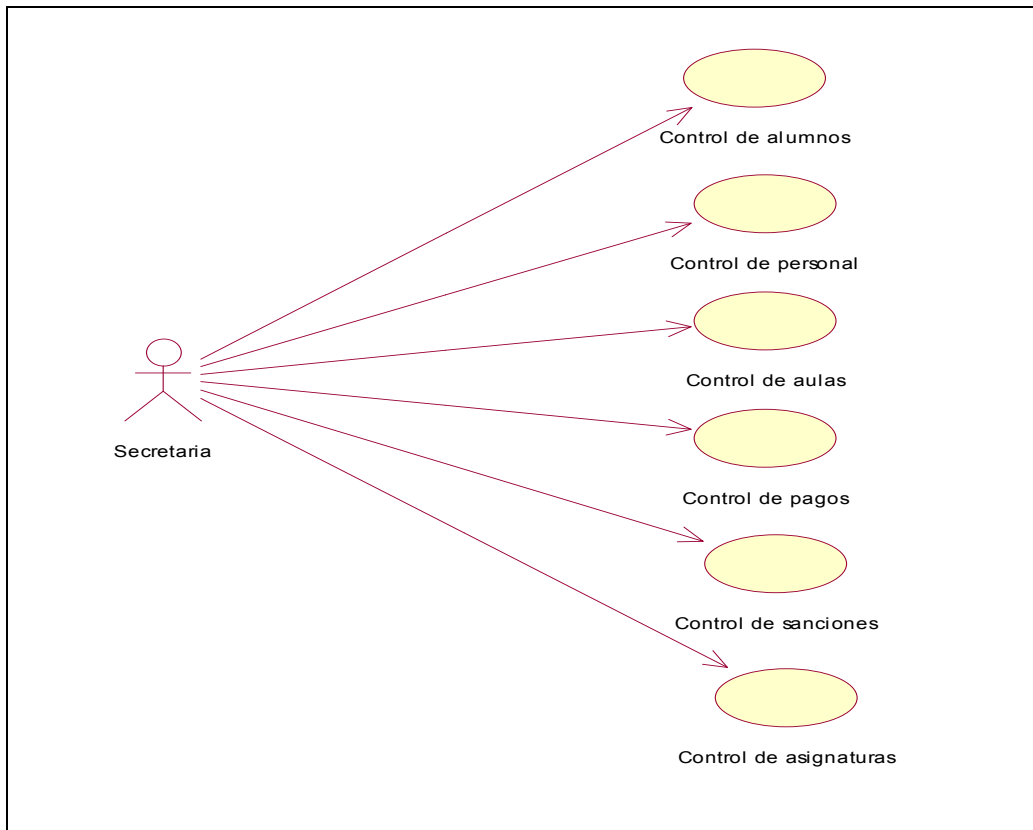


Fig. 2.2 Casos de Uso que Controla el Actor Secretaria

Los siguientes casos de uso especifican en forma más detallada los casos de uso que controla el actor Secretaria.

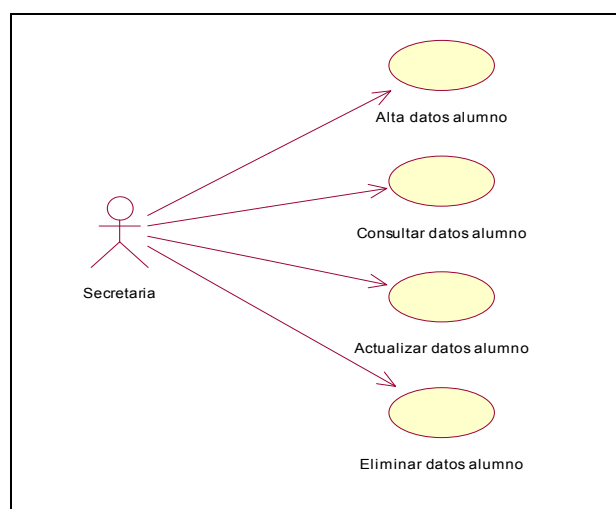


Fig. 2.3 Casos de Uso Control de Alumnos

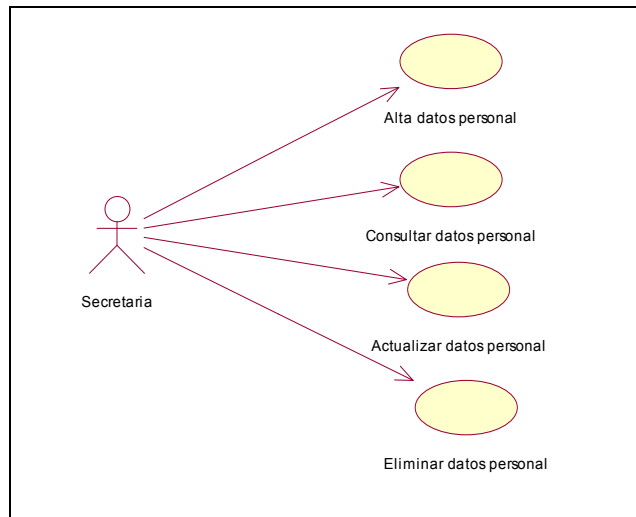


Fig. 2.4 Casos de Uso Control de Personal

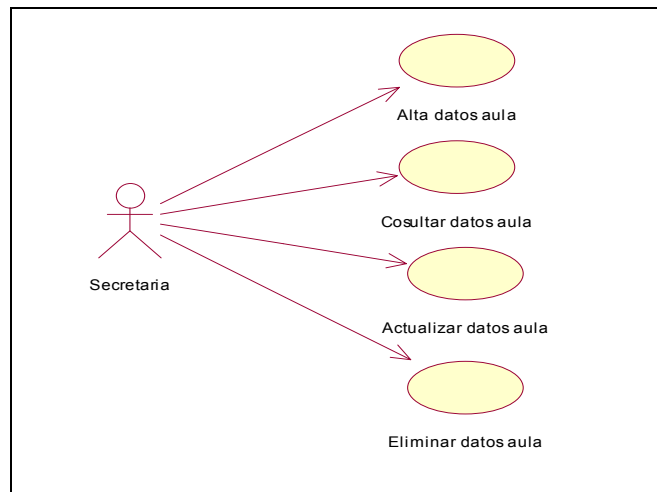


Fig. 2.5 Casos de Uso Control de Aulas

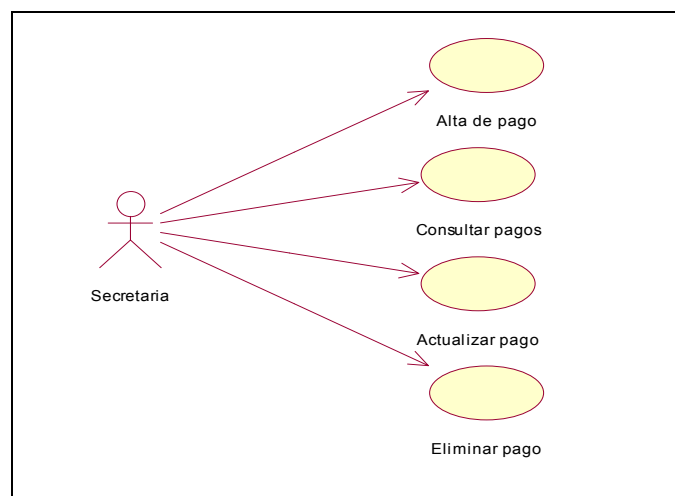


Fig. 2.6 Casos de Uso Control de Pagos

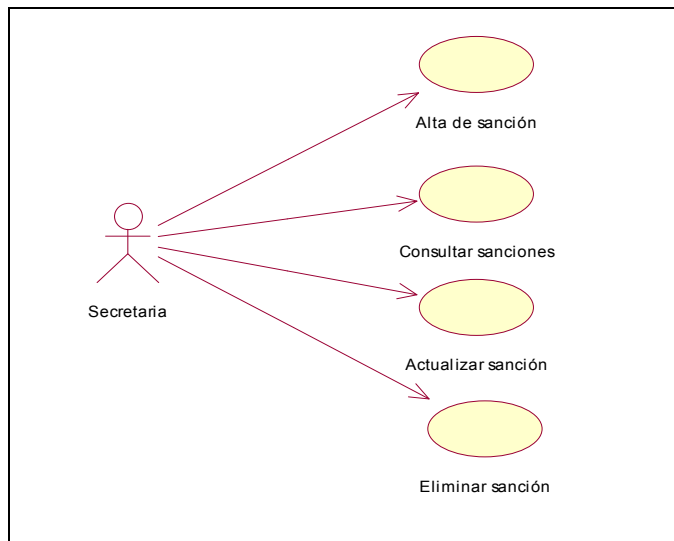


Fig. 2.7 Casos de Uso Control de Sanciones

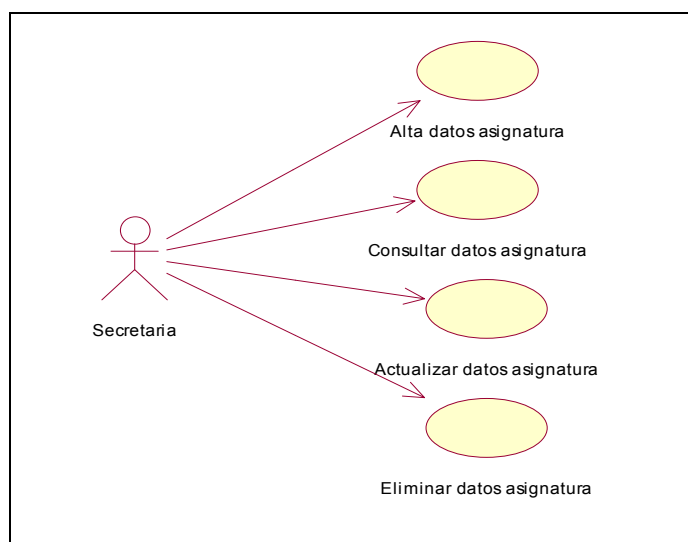


Fig. 2.8 Casos de Uso Control de Asignaturas

El siguiente caso de uso es una forma más general de los casos de uso para el actor administrador presentado en la figura 2.9.

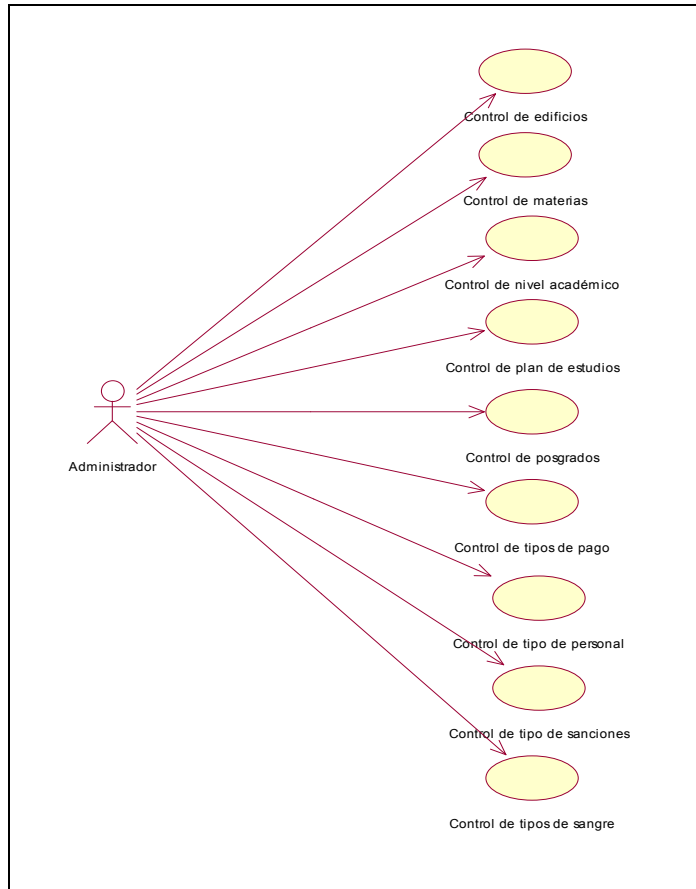


Fig. 2.9 Casos de Uso que Controla el Actor Administrador

Los siguientes casos de uso especifican en forma más detallada los casos de uso que controla el actor Administrador.

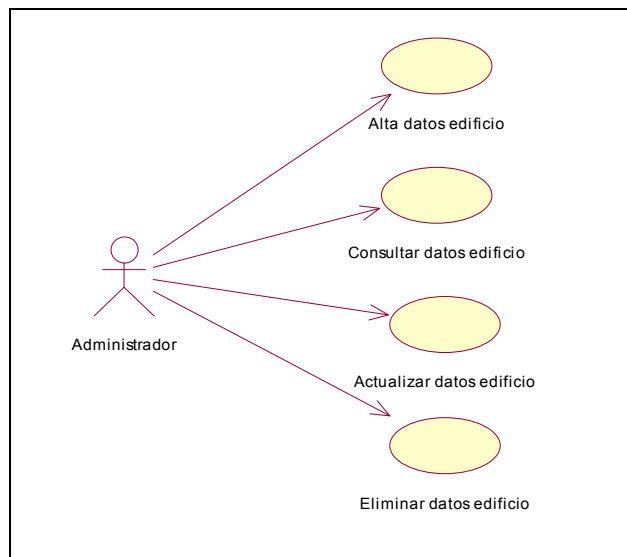


Fig. 2.10 Casos de Uso Control de Edificios

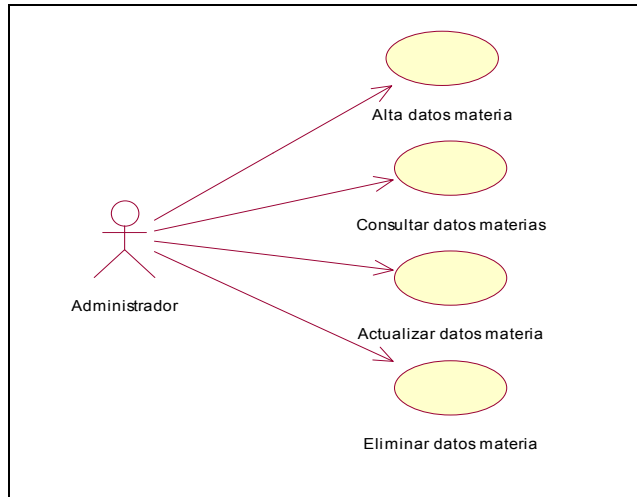


Fig. 2.11 Casos de Uso Control de Materias

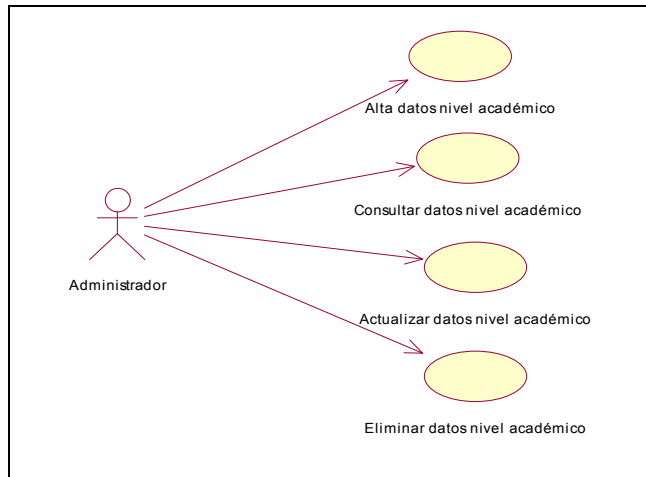


Fig. 2.12 Casos de Uso Control de Nivel Académico

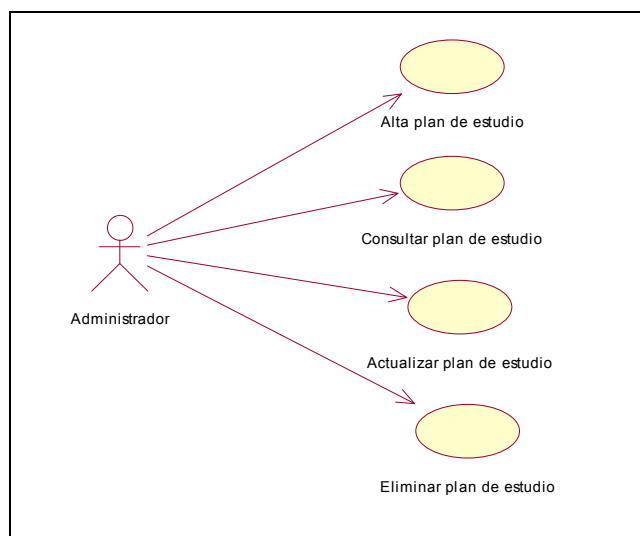


Fig. 2.13 Casos de Uso Control de Plan de Estudios

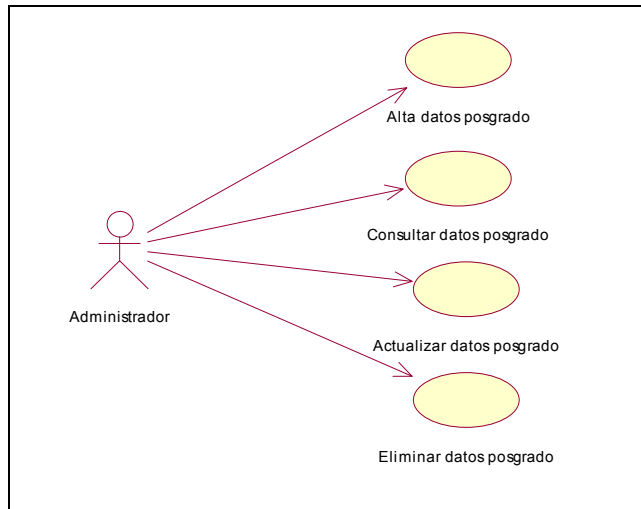


Fig. 2.14 Casos de Uso Control de Posgrados

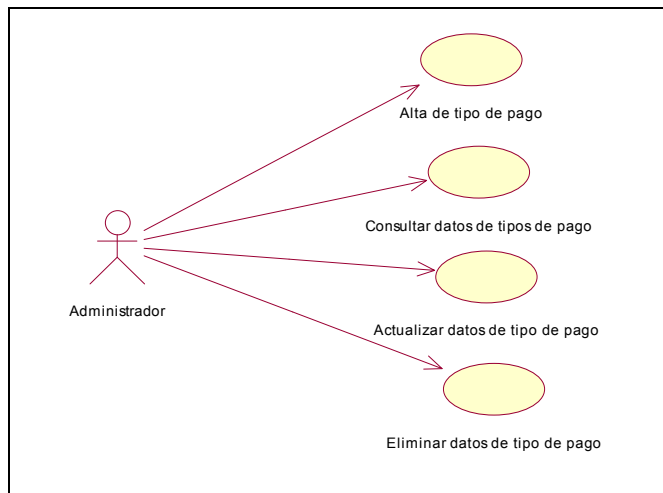


Fig. 2.15 Casos de Uso Control de Tipos de Pago

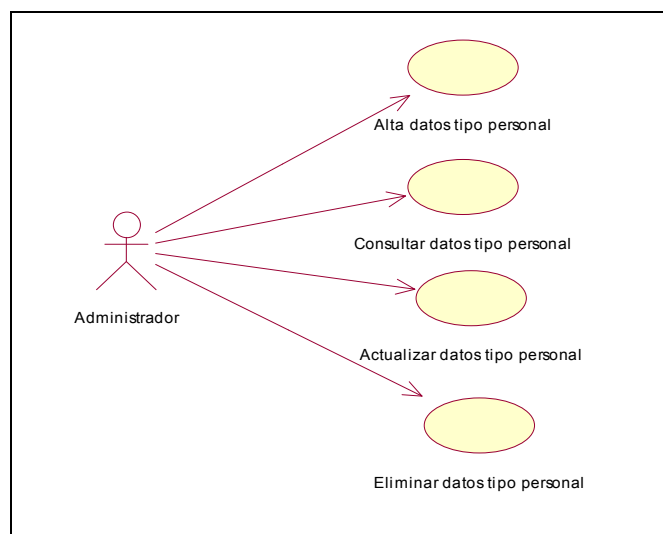


Fig. 2.16 Casos de Uso Control de Tipo de Personal

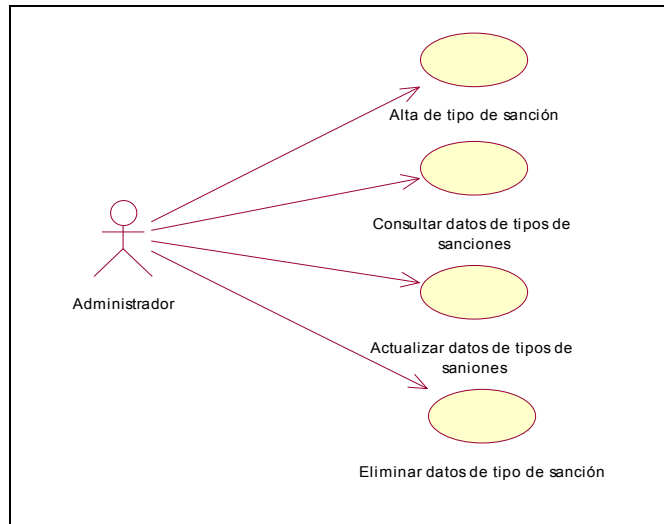


Fig. 2.17 Casos de Uso Control de Tipos de Sanciones

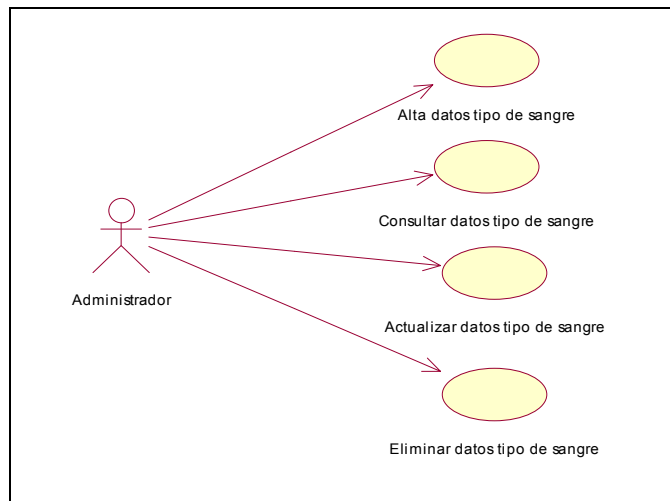


Fig. 2.18 Casos de Uso Control de Tipo de Sangre

El siguiente caso de uso es la forma general de los casos de uso para el actor director presentado en la figura 2.19.

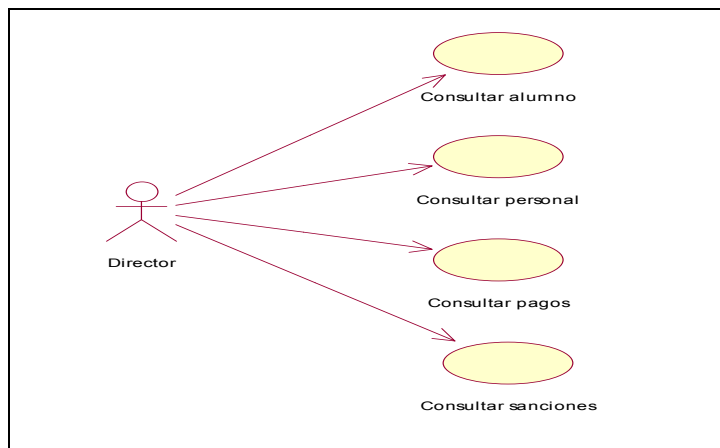


Fig. 2.19 Casos de Uso Director

2.5.1.1 Especificaciones de Casos de Uso

La siguiente especificación de caso de uso lo realizan todos los actores que intervienen en el sistema. Ya que es necesario que cada actor se registre para entrar en los casos de uso específicos para cada actor.

1. Nombre del Caso de Uso: Autenticar.

Breve descripción: El usuario (administrador, secretaria, director) inicia el caso de uso. El sistema le muestra una pantalla donde debe ingresar nombre de usuario y contraseña.

2. Flujo de eventos.

2.1 Flujo Básico

1. El sistema desplegará una pantalla en donde se visualizará un formulario con los datos que deben de ingresarse.
2. El usuario ingresa nombre de usuario y contraseña y posteriormente pulsa el botón de aceptar.
 - 2.1 Si alguno de los datos son incorrectos o si los datos no se encuentran registrados, se desplegará una pantalla con el mensaje de usuario no registrado.
 - 2.2 Si el usuario introduce correctamente los datos, se mostrará en pantalla el menú correspondiente al tipo de usuario.

2.2 Flujos Alternativos

No aplica.

3. Requerimientos especiales

No aplica.

4. Precondiciones

No aplica.

5. Excepciones

1. Si el usuario introduce un nombre de usuario o contraseña incorrecta el sistema despliega un letrero que dice que el usuario no se encuentra registrado. El usuario tiene la opción de regresar al menú principal para volver a intentar autenticarse de nuevo.

1. **Nombre del Caso de Uso:** Alta datos alumno.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide datos como matrícula, nombre, apellidos, sexo, etc. correspondientes al alumno.
 2. El actor llena los campos requeridos de acuerdo a los datos del alumno y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos del alumno en la base de datos y regresa al menú principal de alumnos.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos alumno.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los alumnos que se encuentran registrados, con un campo en el cual se pide la matrícula del alumno que se desea actualizar.
 2. El actor introduce la matrícula del alumno que desea consultar.
 3. El sistema despliega los datos correspondientes a dicho alumno.
 - 3.1 El actor corrige los datos correspondientes y envía el formato, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de alumnos para una nueva actualización.
 - 3.2 El actor cancela la actualización y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos alumno.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de un alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los alumnos que se encuentran registrados, con un campo en el cual se pide la matrícula del alumno que se desea consultar.
 2. El actor introduce la matrícula del alumno que desea consultar.
 3. El sistema despliega los datos correspondientes a dicho alumno.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos alumno.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los alumnos que se encuentran registrados, con un campo en el cual se pide la matrícula del alumno que se desea eliminar.
 2. El actor introduce la matrícula del alumno que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho alumno.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de alumnos actualizada.
 - 3.2 El actor cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta datos personal.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de personal del instituto en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide datos como nombre, apellidos, sexo, etc. correspondientes al trabajador.
 2. El actor llena los campos requeridos de acuerdo a los datos del trabajador y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos del trabajador en la base de datos y regresa al menú principal de personal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos personal.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un trabajador en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los trabajadores que se encuentran registrados, con un campo en el cual se pide la clave del trabajador que se desea actualizar.
 2. El actor introduce la clave del trabajador que desea consultar.
 3. El sistema despliega los datos correspondientes a dicho trabajador.
 - 3.1 El actor corrige los datos correspondientes y envía el formato, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista del personal para una nueva actualización.
 - 3.2 El actor cancela la actualización y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos personal.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de un trabajador en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista del personal que se encuentran registrado, con un campo en el cual se pide la clave del trabajador que se desea consultar.
 2. El actor introduce la clave del trabajador que desea consultar.
 3. El sistema despliega los datos correspondientes a dicho trabajador.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos personal.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un trabajador en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los trabajadores que se encuentran registrados, con un campo en el cual se pide la clave del trabajador que se desea eliminar.
 2. El actor introduce la clave del trabajador que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho trabajador.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de trabajadores actualizada.
 - 3.2 El actor cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta pago.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea registrar un pago de algún alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide datos como nombre del alumno, el tipo de pago que realiza, la fecha, el monto y el saldo.
 2. El actor llena los campos requeridos de acuerdo a los datos del pago y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos del pago en la base de datos y regresa al menú principal de pagos.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar pago.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un pago en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los pagos que se encuentran registrados, con un campo en el cual se pide el id del pago que se desea actualizar.
 2. El actor introduce el id del pago que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicho pago.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de los pagos.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar pagos.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos del pago de un alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los pagos que se encuentran registrados, con un campo en el cual se pide el id del pago que se desea consultar.
 2. El actor introduce la id del pago que desea consultar.
 3. El sistema despliega los datos correspondientes a dicho pago.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar pago.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un pago en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los pagos que se encuentran registrados, con un campo en el cual se pide el id de pago que se desea eliminar.
 2. El actor introduce el id del pago que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho pago.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de pagos.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta datos aula.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un aula en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide datos como nombre del aula, el edificio donde se encuentra, el piso, la capacidad que tiene y algunas observaciones correspondientes a dicha aula.
 2. El actor llena los campos requeridos de acuerdo a los datos del aula y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos del aula en la base de datos y regresa al menú principal de aulas.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentre disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos aula.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un aula en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las aulas que se encuentran registradas, con un campo en el cual se pide el id del aula que se desea actualizar.
 2. El actor introduce el id del aula que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicha aula.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de las aulas.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos aula.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de un aula en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las aulas que se encuentran registradas, con un campo en el cual se pide el id del aula que se desea consultar.
 2. El actor introduce la id del aula que desea consultar.
 3. El sistema despliega los datos correspondientes a dicha aula.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos aula.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un aula en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las aulas que se encuentran registradas, con un campo en el cual se pide el id de aula que se desea eliminar.
 2. El actor introduce el id del aula que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicha aula.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de aulas.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta de sanción.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de la sanción de un alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide datos como nombre del alumno, el tipo de sanción, la fecha y especificar si dicha sanción se encuentra activa.
 2. El actor llena los campos requeridos de acuerdo a los datos de la sanción y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicha sanción en la base de datos y regresa al menú principal de aulas.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar sanción.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de una sanción en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las sanciones que se encuentran registradas, con un campo en el cual se pide el id de la sanción que se desea actualizar.
 2. El actor introduce el id de la sanción que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicha sanción.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de las sanciones.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar sanciones.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de una sanción en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las sanciones que se encuentran registradas, con un campo en el cual se pide el id de la sanción que se desea consultar.
 2. El actor introduce la id de la sanción que desea consultar.
 3. El sistema despliega los datos correspondientes a dicha sanción.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar sanción.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de una sanción en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las sanciones que se encuentran registradas, con un campo en el cual se pide el id de sanción que se desea eliminar.
 2. El actor introduce el id de sanción que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicha sanción.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de sanciones.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta datos asignatura.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de la asignación de materias de un alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide datos como nombre del alumno, el nombre del profesor, la materia, el aula y horarios.
 2. El actor llena los campos requeridos de acuerdo a los datos de la asignatura y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicha asignatura en la base de datos y regresa al menú principal de asignaturas.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos asignatura.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de una asignatura en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las asignaturas que se encuentran registradas, con un campo en el cual se pide el id de la asignatura que se desea actualizar.
 2. El actor introduce el id de la asignatura que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicha asignatura.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de las asignaturas.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos asignatura.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de un alumno con las asignaturas que toma dicho alumno en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las asignaturas y alumnos que se encuentran registradas, con un campo en el cual se pide el id de la asignatura que se desea consultar.
 2. El actor introduce la id de la asignatura que desea consultar.
 3. El sistema despliega los datos correspondientes a dicha asignatura.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos asignatura.
Breve descripción: La secretaria inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de una asignatura en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las asignaturas que se encuentran registradas, con un campo en el cual se pide el id de asignatura que se desea eliminar.
 2. El actor introduce el id de asignatura que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicha asignatura.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de asignaturas.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta datos edificio.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un edificio en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide datos como nombre del edificio, dirección, colonia y código postal.
 2. El actor llena los campos requeridos de acuerdo a los datos del edificio y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicho edificio en la base de datos y regresa al menú principal de edificios.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos edificio.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un edificio en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los edificios que se encuentran registrados, con un campo en el cual se pide el id del edificio que se desea actualizar.
 2. El actor introduce el id del edificio que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicho edificio.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de los edificios.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos edificios.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de un edificio en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los edificios que se encuentran registrados, con un campo en el cual se pide el id del edificio que se desea consultar.
 2. El actor introduce la id del edificio que desea consultar.
 3. El sistema despliega los datos correspondientes a dicho edificio.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos edificio.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un edificio en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los edificios que se encuentran registrados, con un campo en el cual se pide el id de edificio que se desea eliminar.
 2. El actor introduce el id de edificio que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho edificio.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de edificios.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta datos materia.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de una materia en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide datos como nombre de la materia, periodo y valor en créditos.
 2. El actor llena los campos requeridos de acuerdo a los datos de la materia y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicha materia en la base de datos y regresa al menú principal de materias.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos materia.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de una materia en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las materias que se encuentran registradas, con un campo en el cual se pide el id de la materia que se desea actualizar.
 2. El actor introduce el id de la materia que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicha materia.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de las materias.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos materias.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de una materia en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las materias que se encuentran registrados, con un campo en el cual se pide el id de la materia que se desea consultar.
 2. El actor introduce la id de la materia que desea consultar.
 3. El sistema despliega los datos correspondientes a dicha materia.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos materia.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de una materia en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de las materias que se encuentran registradas, con un campo en el cual se pide el id de la materia que se desea eliminar.
 2. El actor introduce el id de materia que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicha materia.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de materias.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta datos posgrado.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un posgrado en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide el nombre del posgrado y el tipo (maestría o doctorado).
 2. El actor llena los campos requeridos de acuerdo a los datos del posgrado y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicho posgrado en la base de datos y regresa al menú principal de posgrados.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos posgrado.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un posgrado en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los posgrados que se encuentran registrados, con un campo en el cual se pide el id del posgrado que se desea actualizar.
 2. El actor introduce el id del posgrado que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicho posgrado.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de posgrados.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos posgrado.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de un posgrado en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los posgrados que se encuentran registrados, con un campo en el cual se pide el id del posgrado que se desea consultar.
 2. El actor introduce la id del posgrado que desea consultar.
 3. El sistema despliega los datos correspondientes a dicho posgrado.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos posgrado.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un posgrado en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los posgrados que se encuentran registradas, con un campo en el cual se pide el id de posgrado que se desea eliminar.
 2. El actor introduce el id de posgrado que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho posgrado.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de posgrados.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta plan estudio.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un plan de estudio en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide el nombre del posgrado y el nombre de la materia.
 2. El actor llena los campos requeridos de acuerdo a los datos del plan de estudio y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicho posgrado en la base de datos y regresa al menú principal de plan de estudio.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar plan de estudio.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un plan de estudio en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los planes de estudio que se encuentran registrados, con un campo en el cual se pide el id del plan de estudio que se desea actualizar.
 2. El actor introduce el id del plan de estudio que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicho plan de estudio.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de planes de estudio.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar plan estudio.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de un plan de estudio en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los planes de estudio que se encuentran registrados, con un campo en el cual se pide el id del plan de estudio que se desea consultar.
 2. El actor introduce el id del plan de estudio que desea consultar.
 3. El sistema despliega los datos correspondientes a dicho plan de estudio.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar plan estudio.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un plan de estudio en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los planes de estudio que se encuentran registrados, con un campo en el cual se pide el id de plan de estudio que se desea eliminar.
 2. El actor introduce el id de plan de estudio que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho plan de estudio.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de planes de estudio.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta datos nivel académico.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un nivel académico en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide el concepto del nivel académico.
 2. El actor llena los campos requeridos de acuerdo a los datos del nivel académico y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicho nivel académico en la base de datos y regresa al menú principal de nivel académico.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar nivel académico.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un nivel académico en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los niveles académicos que se encuentran registradas, con un campo en el cual se pide el id de nivel académico que se desea eliminar.
 2. El actor introduce el id de nivel académico que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho nivel académico.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de niveles académicos.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos nivel académico.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un nivel académico en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El actor elige la opción de consultar datos.
 2. El sistema despliega una lista de los niveles académicos que se encuentran registrados, con un campo en el cual se pide el id del nivel académico que se desea actualizar.
 3. El actor introduce el id del nivel académico que desea actualizar.
 4. El sistema despliega los datos correspondientes a dicho nivel académico.
 - 4.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de niveles académicos.
 - 4.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos nivel académico.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de un nivel académico en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El actor elige la opción de consultar datos.
 2. El sistema despliega una lista de los niveles académicos que se encuentran registrados.
 3. El actor visualiza los conceptos de niveles académicos existentes con la opción de regresar al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta tipo pago.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un tipo de pago en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide el concepto del tipo de pago.
 2. El actor llena los campos requeridos de acuerdo a los datos del tipo de pago y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicho tipo de pago en la base de datos y regresa al menú principal de tipos de pago.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos tipo pago.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un tipo de pago en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los tipos de pago que se encuentran registrados, con un campo en el cual se pide el id del tipo de pago que se desea actualizar.
 2. El actor introduce el id del tipo de pago que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicho tipo de pago.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de tipos de pago.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos tipo de pago.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de tipos de pago en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El actor elige la opción consultar datos.
 2. El sistema despliega una lista de los tipos de pago que se encuentran registrados.
 3. El actor visualiza los conceptos de tipos de pago existentes con la opción de regresar al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos tipo pago.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un tipo de pago en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los tipo de pago que se encuentran registrados, con un campo en el cual se pide el id de tipo de pago que se desea eliminar.
 2. El actor introduce el id de tipo de pago que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho tipo de pago.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de tipo de pagos.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta tipo personal.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un tipo de personal en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide el concepto del tipo de personal.
 2. El actor llena los campos requeridos de acuerdo a los datos del tipo de personal y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicho tipo de personal en la base de datos y regresa al menú principal de tipos de personal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos tipo personal.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un tipo de personal en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los tipos de personal que se encuentran registrados, con un campo en el cual se pide el id del tipo de personal que se desea actualizar.
 2. El actor introduce el id del tipo de personal que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicho tipo de personal.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de tipos de personal.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos tipo personal.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de tipos de personal en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El actor elige la opción consultar datos.
 2. El sistema despliega una lista de los tipos de personal que se encuentran registrados.
 3. El actor visualiza los conceptos de tipos de personal existentes con la opción de regresar al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos tipo personal.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un tipo de personal en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los tipo de personal que se encuentran registradas, con un campo en el cual se pide el id de tipo de personal que se desea eliminar.
 2. El actor introduce el id de tipo de personal que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho tipo de personal.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de tipo de personal.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta tipo sanción.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un tipo de sanción en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide el concepto del tipo de sanciones.
 2. El actor llena los campos requeridos de acuerdo a los datos del tipo de sanción y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicho tipo de sanción en la base de datos y regresa al menú principal de tipos de sanción.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos tipo sanción.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un tipo de sanción en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los tipos de sanciones que se encuentran registrados, con un campo en el cual se pide el id del tipo de sanción que se desea actualizar.
 2. El actor introduce el id del tipo de sanción que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicho tipo de sanción.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de tipos de sanciones.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos tipo sanción.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de tipos de sanción en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El actor elige la opción consultar datos.
 2. El sistema despliega una lista de los tipos de sanción que se encuentran registrados.
 3. El actor visualiza los conceptos de tipos de sanciones existentes con la opción de regresar al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos tipo sanción.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un tipo de sanción en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los tipo de sanciones que se encuentran registradas, con un campo en el cual se pide el id de tipo de sanción que se desea eliminar.
 2. El actor introduce el id de tipo de sanción que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho tipo de sanción.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de tipo de sanciones.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Alta tipo sangre.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea registrar datos de un tipo de sangre en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema muestra un formulario que pide el concepto del tipo de sangre.
 2. El actor llena los campos requeridos de acuerdo a los datos del tipo de sangre y posteriormente pulsa el botón de agregar registro.
 3. El sistema inserta los datos de dicho tipo de sangre en la base de datos y regresa al menú principal de tipos de sangre.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden insertar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Actualizar datos tipo sangre.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea actualizar los datos de un tipo de sangre en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los tipos de sangre que se encuentran registrados, con un campo en el cual se pide el id del tipo de sangre que se desea actualizar.
 2. El actor introduce el id del tipo de sangre que desea actualizar.
 3. El sistema despliega los datos correspondientes a dicho tipo de sangre.
 - 3.1 El actor corrige los datos correspondientes y pulsa el botón de actualizar datos, el sistema actualiza los datos y vuelve a la pantalla donde se muestra la lista de tipos de sangre.
 - 3.2 El actor cancela la actualización y vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden actualizar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Consultar datos tipo sangre.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea consultar los datos de tipos de sangre en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El actor elige la opción consultar datos.
 2. El sistema despliega una lista de los tipos de sangre que se encuentran registrados.
 3. El actor visualiza los conceptos de tipos de sangre existentes con la opción de regresar al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden consultar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

1. **Nombre del Caso de Uso:** Eliminar datos tipo sangre.
Breve descripción: El administrador inicia el caso de uso. Este caso se utiliza cuando se desea eliminar los datos de un tipo de sangre en la base de datos.
2. **Flujo de eventos.**
 - 2.1 **Flujo Básico**
 1. El sistema despliega una lista de los tipo de sangre que se encuentran registradas, con un campo en el cual se pide el id de tipo de sangre que se desea eliminar.
 2. El actor introduce el id de tipo de sangre que desea eliminar.
 3. El sistema despliega los datos correspondientes a dicho tipo de sangre.
 - 3.1 El actor elimina los datos y el sistema regresa a la pantalla donde muestra la lista de tipo de sangre.
 - 3.2 El usuario cancela la acción y el sistema vuelve al menú principal.
 - 2.2 **Flujos Alternativos**

No aplica.
3. **Requerimientos especiales**

No aplica.
4. **Precondiciones**

Que el actor haya realizado satisfactoriamente el caso de uso: Autenticar.
5. **Excepciones**
 1. Si los datos no se pueden eliminar el sistema manda un aviso indicando que la conexión con la base de datos es fallida y que lo intente más tarde porque el servidor no se encuentra disponible en ese momento.

CAPÍTULO 3. DISEÑO DEL SISTEMA

El diseño del sistema es el proceso de aplicar ciertas técnicas y principios con el propósito de definir el sistema, con suficientes detalles como para permitir su interpretación y realización física.

Con el presente capítulo se suministra una vista “arquitectónica” del Sistema de Control Escolar para el ICsYH. En este capítulo el principal objetivo es el proporcionar una idea completa del sistema, enfocando los dominios de datos, funcional y comportamiento desde el punto de vista de la implementación.

3.1 MODELO DE DATOS

El modelo de datos nos permite conocer a fondo y de forma abstracta la representación de los datos del sistema a realizar. En las siguientes secciones se especificarán las entidades que se identificaron, sus atributos y sus relaciones para así tener nuestro modelo entidad-relación.

3.2 DESCRIPCION DE ENTIDADES Y ATRIBUTOS

Entidad: alumno

Descripción: Contiene la información de los alumnos registrados.

Atributos:

Matrícula: número consecutivo que se asigna a cada alumno, el cual es único.

Nombre: nombre o nombres del alumno.

Apellidos: apellido paterno y materno del alumno.

Sexo: si el alumno es hombre o mujer.

Tipo de sangre: el grupo sanguíneo al cual pertenece el alumno.

Email: correo electrónico personal del alumno.

Fecha de nacimiento: fecha de nacimiento del alumno.

Lugar de nacimiento: lugar de nacimiento del alumno.

País: país del lugar de nacimiento del alumno.

Universidad de procedencia: universidad de procedencia del alumno.

Dirección: dirección del alumno.

Colonia: colonia de la dirección.

Código postal: código postal de la dirección.

Teléfono de casa: teléfono de la casa del alumno.

Teléfono de emergencia: teléfono de emergencia en caso de accidente.

CURP: clave única de registro del alumno.

Fecha de ingreso: fecha de ingreso al posgrado.

Observaciones: alguna observación respecto al alumno.

Entidad: personal

Descripción: Contiene la información del personal registrado.

Atributos:

Clave de trabajador: número consecutivo que se asigna a cada trabajador, el cual es único.

Código de nivel académico: número que corresponde al nivel académico del trabajador.

Código de tipo de personal: número que corresponde al tipo de personal del trabajador.

Nombre: nombre o nombres del trabajador.

Apellidos: apellido paterno y materno del trabajador.

Sexo: si el trabajador es hombre o mujer.

Tipo de sangre: el grupo sanguíneo al cual pertenece el trabajador.

Email: correo electrónico personal del trabajador.

Fecha de nacimiento: fecha de nacimiento del trabajador.

Lugar de nacimiento: lugar de nacimiento del trabajador.

País: país del lugar de nacimiento del trabajador.

Dirección: dirección del trabajador.

Colonia: colonia de la dirección.

Código postal: código postal de la dirección.

Teléfono: teléfono del trabajador.

RFC: registro federal de contribuyente del trabajador

Curp: curp del trabajador.

Fecha de contrato: fecha de ingreso al instituto.

Entidad: nivel_academico

Descripción: Contiene la información de los tipos de nivel académico registrados.

Atributos:

Clave nivel académico: número consecutivo que se asigna a cada tipo de nivel académico, el cual es único.

Nivel: el nombre de nivel académico.

Entidad: tipo_personal

Descripción: Contiene la información de los tipos de personal registrados.

Atributos:

Id tipo personal: número consecutivo que se asigna a cada tipo de personal, el cual es único.

Tipo: el nombre del tipo de personal.

Entidad: sanciones

Descripción: Contiene la información de las sanciones registradas.

Atributos:

Id sanción: número consecutivo que se asigna a cada sanción, el cual es único.

Matrícula alumno: la matrícula del alumno que cometió la sanción.

Código tipo de sanción: número que corresponde al tipo de sanción.

Fecha: fecha en que se registró la sanción.

Activo: si la sanción se encuentra activa o no.

Entidad: tipo_sancion

Descripción: Contiene la información de los tipos de sanciones registradas.

Atributos:

Id tipo sanción: número consecutivo que se asigna a cada tipo de sanción, el cual es único.

Concepto: el nombre del tipo de sanción.

Activo: si el tipo de sanción se encuentra activa o no.

Entidad: pagos

Descripción: Contiene la información de los pagos registrados.

Atributos:

Id pago: número consecutivo que se asigna a cada pago, el cual es único.

Matricula alumno: la matrícula del alumno que hace el pago.

Código tipo de pago: número que corresponde al tipo de pago.

Fecha: fecha en que se registró el pago.

Monto: el monto del pago.

Saldo: el saldo del pago.

Entidad: tipo_pago

Descripción: Contiene la información de los tipos de pagos registrados.

Atributos:

Id tipo pago: número consecutivo que se asigna a cada tipo de pago, el cual es único.

Concepto: el nombre del tipo de pago.

Activo: si el tipo de pago se encuentra activo o no.

Entidad: materias

Descripción: Contiene la información de las materias registradas.

Atributos:

Id materia: número consecutivo que se asigna a cada materia, el cual es único.

Nombre: nombre de la materia.

Periodo: periodo en el que se imparte la materia.

Valor en créditos: el valor en créditos que tiene la materia.

Entidad: posgrado

Descripción: Contiene la información de los posgrados registrados.

Atributos:

Código posgrado: número consecutivo que se asigna a cada posgrado, el cual es único.

Descripción: nombre del posgrado.

Tipo: se define como maestría o doctorado.

Entidad: edificios

Descripción: Contiene la información de los edificios registrados.

Atributos:

Id edificio: número consecutivo que se asigna a cada edificio, el cual es único.

Nombre: nombre del edificio.

Dirección: dirección del edificio.

Colonia: colonia donde se ubica el edificio.

Código postal: código postal de la ubicación del edificio.

Entidad: aulas

Descripción: Contiene la información de los aulas registrados.

Atributos:

Id aula: número consecutivo que se asigna a cada aula, el cual es único.

Nombre aula: nombre del aula.

Código edificio: número que corresponde al edificio en donde se encuentra ubicada el aula.

Numero de piso: el piso donde se encuentra ubicada el aula.

Capacidad: capacidad de alumnos en el aula.

Observaciones: alguna observación sobre el aula.

Entidad: asignatura

Descripción: Contiene la información registrada de las materias que toma un alumno además del profesor que imparte el curso, la materia que toma y el aula donde se ubica dicha clase.

Atributos:

Id asignatura: número consecutivo que se asigna a cada asignatura, el cual es único.

Matricula alumno: número que corresponde a la matrícula del alumno.

Código trabajador: número que corresponde al código de trabajador, es decir el profesor.

Código materia: número que corresponde a la materia.

Código aula: número que corresponde al aula.

Hora inicio: hora en que inicia la clase.

Hora fin: hora en que termina la clase.

Fecha: fecha de la inscripción.

Activo: si dicha asignatura está activa o no.

3.3 RELACIONES ENTRE ENTIDADES

Después de definir cada entidad con sus atributos se deben expresar las relaciones existentes entre ellas. Es por eso que para identificar una relación se suele hacer uso de expresiones verbales que suelen hacer de “puente” entre una entidad y otra. En la tabla 3.1 es posible ver las relaciones entre cada una de las entidades.

Tabla 3.1 Relaciones entre Entidades

Relación	Entidades	Cardinalidad
Inscripción	alumno, asignatura	1:N
Realiza	alumno, pagos	1:N
Comete	alumno, sanciones	1:N
Toma	aulas, asignatura	1:N
Ubica	edificios, aulas	1:N
Esta	materias, asignatura	1:N
Dentro_de_planestudio	materias, posgrado	M:N
Pertenece	nivel_academico, personal	1:N
Imparte	personal, asignatura	1:N
Dentro	tipo_pago, pagos	1:N
Clasifica	tipo_personal, personal	1:N
Compete	tipo_sancion, sanciones	1:N
Corresponde	tipo_sangre, alumno	1:N
Tiene	tipo_sangre, personal	1:N

3.4 DIAGRAMA ENTIDAD-RELACION

El siguiente paso es crear un modelo conceptual. El modelo más usado en base de datos es el modelo Entidad-Relación, que es el que se utilizará para realizar el sistema.

En esencia, el modelo entidad-relación, consiste en buscar las entidades que describan los objetos que intervienen en el sistema y las relaciones entre las entidades.

Todo esto se plasma en un esquema gráfico que tiene por objeto, por una parte, ayudar al programador durante la codificación y por otra, al usuario a comprender el problema y el funcionamiento del programa.

El diagrama entidad-relación presentado en la figura 3.1 describe en forma gráfica las diferentes relaciones entre cada una de las entidades encontradas.

Si bien existen muchas representaciones gráficas para el diagrama entidad-relación se ha decidido usar por comodidad los atributos incluidos dentro de cada entidad, en lugar de colocarlos en forma de óvalos. A su vez se han mapeado las llaves primarias necesarias hacia otras entidades tal y como lo marca el mapeo a tablas.

Las entidades que se presentarán a continuación son el resultado del análisis que se realizó a los requerimientos del sistema.

El diagrama entidad-relación que se muestra en la figura 3.1 se realizó con ER/Studio que es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de bases de datos que construyen y mantienen aplicaciones de bases de datos grandes y complejos.

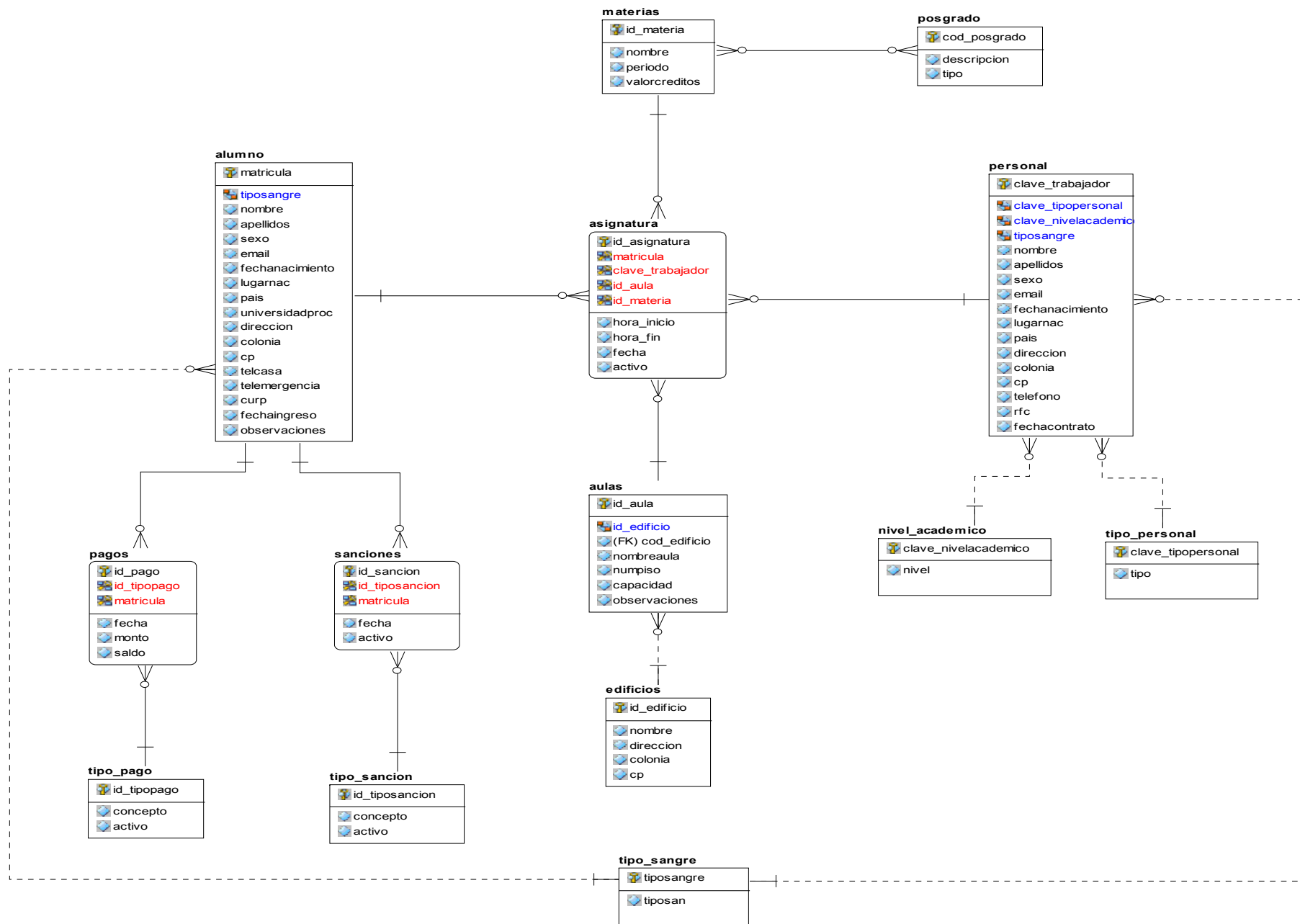


Fig. 3.1 Diagrama Entidad-Relación

3.5 MODELO RELACIONAL

La siguiente fase es convertir el modelo conceptual en un modelo lógico. Existen varios modelos lógicos, pero el más usado, el que mejor se adapta a MySQL y el que por lo tanto usaremos en el sistema, es el *modelo Relacional*.

Existen varias reglas para convertir cada uno de los elementos de los diagramas entidad-relación en tablas:

Para cada conjunto de entidades fuertes se crea una relación con una columna para cada atributo.

Para cada interrelación de cardinalidad M:N se crea una relación que contiene una columna para cada atributo correspondiente a las claves principales de las entidades interrelacionadas.

Entonces se define el siguiente modelo relacional:

alumno (matricula(PK), nombre, apellidos, sexo, tiposangre(FK), email, fechanacimiento, lugarnac, país, universidadproc, direccion, colonia, cp, telcasa, telemergencia, curp, fechaingreso, observaciones)

personal (clave_trabajador(PK), cod_nivelacademico(FK), cod_tipopersonal(FK), nombre, apellidos, sexo, tiposangre(FK), email, fechanacimiento, lugarnac, pais, direccion, colonia, cp, teléfono, rfc, fechacontrato)

tipo_sangre (tipoval(PK), tiposan)

nivel_academico (clave_nivelacademico(PK), nivel)

tipo_personal (clave_tipopersonal(PK), tipo)

pagos (id_pago(PK), matricula_alumno(FK), cod_tipopago(FK), fecha, monto, saldo)

tipo_pago (id_tipopago(PK), concepto, activo)

sanciones (id_sancion(PK), matri_alum(FK), codi_tiposan(FK), fecha, activo)

tipo_sancion (id_tiposancion(PK), concepto, activo)

edificios (id_edificio(PK), nombre, direccion, colonia, cp)

aulas (id_aula(PK), cod_edificio(FK), nombreaula, numpiso, capacidad, observaciones)

materias (id_materia(PK), nombre, periodo, valorcreditos)

posgrado (cod_posgrado(PK), descripcion, tipo)

dentro_de (id_planest(PK), codigo_mat(FK), codigo_posg(FK))

asignaturas (id_asignatura(PK), matri_alumno(FK), codigo_trab(FK), codigo_materia(FK), codigo_aula(FK), hora_inicio, hora_fin, fecha, activo)

3.6 DESCRIPCIÓN DE TABLAS

Cada tabla (procedente de una entidad o relación) debe ser documentada perfectamente para que pueda ser implementada sin ningún problema. Es necesario indicar su llave primaria y las correspondientes llaves foráneas, así como también tipo y longitud de cada atributo.

Se utilizará la siguiente simbología para describir las entidades:

- # Para el identificador único.
- (FK) Para las llaves foráneas.
- * Para los atributos obligatorios.
- o Para los atributos opcionales.

Tabla: alumno

Tabla 3.2 Tabla Alumno

	Name	Type	Length
# *	matricula	int	11
*	nombre	varchar	40
*	apellidos	varchar	40
*	Sexo	char	1
*(FK)	tiposangre	smallint	2
o	Email	varchar	50
*	fechanacimiento	date	0
o	lugarnac	varchar	50
o	Pais	varchar	50
o	universidadproc	varchar	50
o	direccion	varchar	50
o	colonia	varchar	50
o	Cp	int	5
o	telcasa	int	20
o	telemergencia	int	20
o	Curp	varchar	50
*	fechaingreso	date	0
o	observaciones	varchar	50

Tabla: nivel_academico

Tabla 3.3 Tabla Nivel Académico

	Name	Type	Length
# *	clave_nivelacademico	int	11
*	nivel	varchar	30

Tabla: personal

Tabla 3.4 Tabla Personal

	Name	Type	Length
# *	clave_trabajador	int	11
(FK) *	cod_nivelacademico	int	11
(FK) *	cod_tipopersonal	int	11
*	nombre	varchar	40
*	apellidos	varchar	40
*	Sexo	char	1
(FK) *	tiposangre	smallint	2
o	email	varchar	50
*	fechanacimiento	date	0
o	lugarnac	varchar	50
o	Pais	varchar	50
o	direccion	varchar	50
o	colonia	varchar	50
o	Cp	int	5
o	telefono	int	20
o	Rfc	varchar	50
*	fechacontrato	date	0

Tabla: tipo_personal

Tabla 3.5 Tabla Tipo de Personal

	Name	Type	Length
# *	clave_tipopersonal	int	11
*	Tipo	varchar	30

Tabla: tipo_sangre

Tabla 3.6 Tabla Tipo de Sangre

	Name	Type	Length
# *	tipoval	smallint	2
*	tiposan	varchar	20

Tabla: pagos

Tabla 3.7 Tabla Pagos

	Name	Type	Length
# *	id_pago	int	11
(FK) *	matricula_alumno	int	11
(FK) *	cod_tipopago	Int	11
*	fecha	date	0
*	monto	decimal	10
*	saldo	decimal	10

Tabla: tipo_pago

Tabla 3.8 Tabla Tipo de Pago

	Name	Type	Length
# *	id_tipopago	int	11
*	concepto	varchar	30
*	activo	char	2

Tabla: sanciones

Tabla 3.9 Tabla Sanciones

	Name	Type	Length
# *	id_sancion	int	11
(FK) *	matri_alum	int	11
(FK) *	codi_tiposan	int	11
*	fecha	date	0
*	activo	char	2

Tabla: tipo_sancion

Tabla 3.10 Tabla Tipo de Sanción

	Name	Type	Length
# *	id_tiposancion	int	11
*	concepto	varchar	30
*	activo	char	2

Tabla: aulas

Tabla 3.11 Tabla Aulas

	Name	Type	Length
# *	id_aula	int	11
*	nombreaula	varchar	40
(FK) *	cod_edificio	int	11
o	numpiso	varchar	30
o	capacidad	varchar	30
o	observaciones	varchar	50

Tabla: materias

Tabla 3.12 Tabla Materias

	Name	Type	Length
# *	id_materia	int	11
*	nombre	int	50
o	periodo	int	25
o	valorcreditos	varchar	10

Tabla: edificios

Tabla 3.13 Tabla Edificios

	Name	Type	Length
# *	id_edificio	int	11
*	nombre	varchar	50
o	dirección	varchar	50
o	colonia	varchar	50
o	cp	int	5

Tabla: posgrado

Tabla 3.14 Tabla Posgrado

	Name	Type	Length
# *	cod_posgrado	int	11
*	descripcion	varchar	40
*	tipo	char	1

Tabla: dentro_de_planestudio

Tabla 3.15 Tabla Dentro de Plan de Estudio

	Name	Type	Length
# *	id_planest	int	11
*(FK)	codigo_posg	int	11
*(FK)	codigo_mat	int	11

Tabla: asignaturas

Tabla 3.16 Tabla Asignaturas

	Name	Type	Length
# *	id_asignatura	int	11
(FK) *	matri_alumno	int	11
(FK) *	codigo_trab	int	11
(FK) *	codigo_materia	int	11
(FK) *	codigo_aula	int	11
o	hora_inicio	time	0
o	hora_fin	time	0
*	fecha	date	0
*	activo	char	2

3.7 NORMALIZACIÓN

Existe un proceso que sirve para verificar que hemos aplicado bien el modelo, y en caso contrario, corregirlo para que sea así. Este proceso se llama *normalización*.

El objetivo de normalizar una base de datos es para:

- Evitar la redundancia e inconsistencia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos

Por lo que tenemos cuatro reglas básicas de normalización:

- Primera forma normal: nos sirve para tener atributos atómicos, es decir, que son indivisibles.
- Segunda forma normal: nos sirve para no tener dependencias principales, es decir, que todos los atributos que no son parte de una clave dependen de forma completa de la clave primaria.
- Tercera forma normal: nos sirve para tener solo atributos que si no forman parte de ninguna clave, pertenecen directamente y no transitivamente de la clave primaria.
- Forma normal de Boyce-Codd: nos dice que una relación está en FNBC si cualquier atributo sólo facilita información sobre claves candidatas, y no sobre atributos que no formen parte de ninguna clave candidata.

Tabla asignatura

# <u>id asignatura</u>	
# <u>matri_alumno</u>	←
# <u>codigo_trab</u>	←
# <u>codigo_materia</u>	←
# <u>codigo_aula</u>	←
hora_inicio	←
hora_fin	←
fecha	←
activo	←

Tras observar la tabla asignatura podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla alumno

# <u>matricula</u>	
nombre	←
apellidos	←
sexo	←
# <u>tiposangre</u>	←
email	←
fechanacimiento	←
lugarnac	←
pais	←
universidadproc	←
direccion	←
colonia	←
cp	←
telcasa	←
telemergencia	←
curp	←
fechaingreso	←
observaciones	←

Tras observar la tabla alumno podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla nivel_academico

# <u>clave_nivelacademico</u>	nivel
-------------------------------	-------

A diagram showing a functional dependency. A horizontal line connects the underlined attribute '# clave_nivelacademico' to the attribute 'nivel'. An arrow points from the line to 'nivel', indicating that 'clave_nivelacademico' functionally determines 'nivel'.

Tras observar la tabla nivel_academico podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla personal

# <u>clave_trabajador</u>	
# <u>cod_nivelacademico</u>	←
# <u>cod_tipopersona</u>	←
nombre	←
apellidos	←
sexo	←
# <u>tiposangre</u>	←
email	←
fechanacimiento	←
lugarnac	←
pais	←
direccion	←
colonia	←
cp	←
telefono	←
rfc	←
fechacontrato	←

Tras observar la tabla personal podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla tipo_personal

# <u>clave_tipopersonal</u>	tipo
-----------------------------	------


A diagram showing a functional dependency from the primary key attribute '# clave_tipopersonal' to the attribute 'tipo'. A horizontal line connects the two attributes. A vertical line descends from the primary key attribute, and another vertical line ascends to the 'tipo' attribute, connected by a horizontal line at the bottom, forming a U-shape with an arrowhead pointing to 'tipo'.

Tras observar la tabla tipo_personal podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla pagos

# <u>id_pago</u>	# <u>matricula_alumno</u>	# <u>cod_tipopago</u>	fecha	monto	saldo
------------------	---------------------------	-----------------------	-------	-------	-------




Tras observar la tabla pagos podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla tipo_pago

# <u>id_tipopago</u>	concepto	activo
----------------------	----------	--------

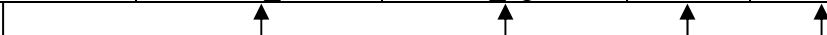


Tras observar la tabla tipo_pago podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla sanciones

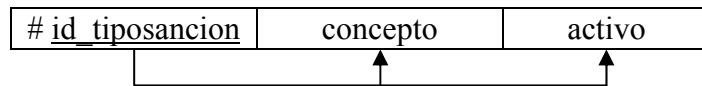
# <u>id_sancion</u>	# <u>matri_alum</u>	# <u>codi_tiposan</u>	fecha	activo
---------------------	---------------------	-----------------------	-------	--------



Tras observar la tabla sanciones podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

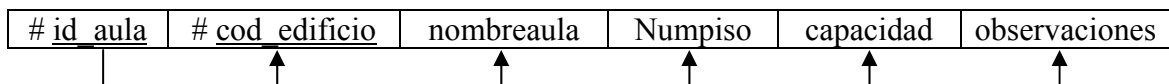
Tabla tipo_sancion



Tras observar la tabla tipo_sancion podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

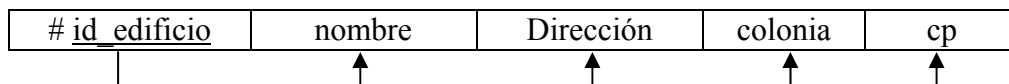
Tabla aulas



Tras observar la tabla aulas podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla edificios




Tras observar la tabla edificios podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla materias

# <u>id_materia</u>	nombre	Periodo	valorcreditos
---------------------	--------	---------	---------------




Tras observar la tabla materias podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla posgrado

# <u>cod_posgrado</u>	descripcion	tipo
-----------------------	-------------	------



Tras observar la tabla posgrado podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

Tabla dentro_de_planestudio

# <u>id_planest</u>	# <u>codigo_posg</u>	# <u>codigo_mat</u>
---------------------	----------------------	---------------------

Tras observar la tabla tipo_sangre podemos comprobar que la tabla está en:

- ✓ **1FN** porque sus atributos son atómicos y no repetidos.
- ✓ **2FN** porque está en 1FN y además los atributos que no son llave primaria dependen funcional y totalmente de la llave primaria.
- ✓ **3FN** porque está en 2FN y además no hay atributos que dependan transitivamente de la llave primaria.
- ✓ **FNBC** porque está en 3FN y además cualquier atributo facilita información sobre claves candidatas.

CAPÍTULO 4. IMPLEMENTACIÓN Y PRUEBAS

El principal objetivo de esta fase es obtener un producto de software operativo que cuente con los requisitos específicos para su uso, mencionados en la fase anterior. Mediante los apartados de esta fase se mostrarán las herramientas de software utilizadas, la implementación del sistema y algunos algoritmos de importancia.

4.1 HERRAMIENTAS UTILIZADAS

Para lograr que este sistema llegue a los usuarios desde cualquier computadora se optó por presentarlo a través de un sitio web. El uso de una herramienta para generar la base de datos antes diseñada, una plataforma para el desarrollo de la interfaz, y un servidor para el manejo de la información fueron necesarios. La utilización de Navicat, Dreamweaver y Wampserver, se considera herramientas de implementación.

A continuación se realizará una breve descripción del software utilizado para la realización de este sistema.



4.1.1 Navicat

PremiumSoft Navicat es una serie de gestión de base de datos y de software de desarrollo de gráficos para MySQL. Tiene un explorador como interfaz y apoya las conexiones de bases de datos locales.

Se trata de una aplicación que permite administrar servidores MySQL de una forma gráfica desde una aplicación de escritorio, puede conectar a servidores locales, de red o a través de Internet.

La interfaz permite sincronizar las estructuras de datos, parámetros para las consultas, asistente para crear vistas, imprimir la estructura de las tablas, importar y exportar datos, en definitiva es una buena herramienta para aquellos que trabajamos con MySQL.



4.1.2 Dreamweaver

Dreamweaver es la herramienta de diseño de páginas web más avanzada, cumple perfectamente el objetivo de diseñar páginas con aspecto profesional y soporta gran cantidad de tecnologías, además muy fáciles de usar: hojas de estilo, javascript, inserción de archivos entre otros.

Dreamweaver permite crear archivos con extensión .php, lo que facilita poder combinar código html con php, esta es otra de las razones por las que esta herramienta es una de las mejores a utilizar.



4.1.3 Wampserver

WAMP es una forma de mini-servidor que puede ejecutarse en casi cualquier sistema operativo Windows. WAMP incluye PHP 5, Apache 2 y MySQL preinstalado. Un icono en la bandeja de la barra de tareas muestra el estado de WAMP, lo que le permitirá saber si:

- a) WAMP está funcionando pero no se abren los servicios (el icono aparecerá de color rojo),
- b) WAMP está funcionando y es un servicio abierto (el icono parecerá amarillo) o
- c) WAMP está funcionando con todos los servicios abiertos (el icono aparecerá de color blanco).

4.2 IMPLEMENTACIÓN

4.2.1 Construcción de la Base de Datos en Navicat

Una vez que se tienen las tablas que debe contener la base de datos, es de suma facilidad implementar la base de datos, utilizando Navicat como herramienta para administrar mediante un navegador WEB el manejador de bases de datos MySQL.

Así que cada tabla se crea introduciendo su nombre, campos, atributos y llaves primarias y/o foráneas, como se observa en la figura 4.1.

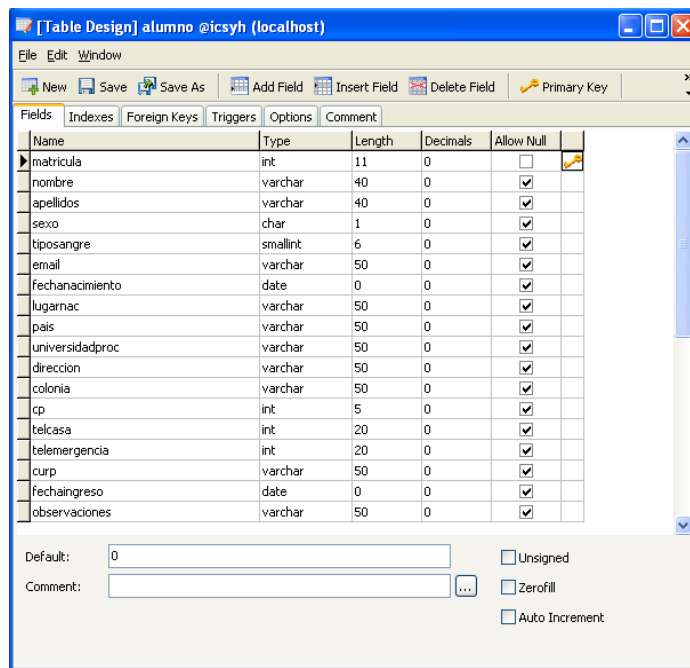


Fig. 4.1 Creación de Tablas en Navicat

La base de datos se llama icsyh y la vista general de las tablas que contiene se puede ver en la figura 4.2

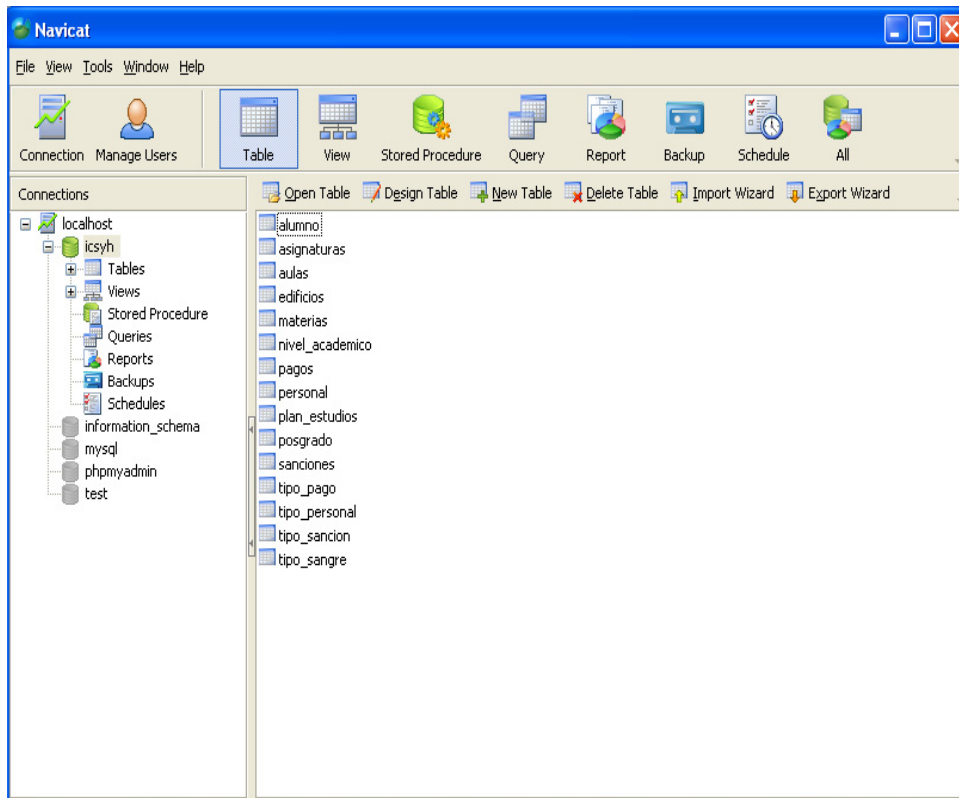


Fig. 4.2 Vista de las Tablas de la Base de Datos icsyh

Al finalizar el proceso de construcción de las tablas es posible manipularlas directamente desde algún lenguaje de programación.

4.3 CODIFICACIÓN EN PHP

Para realizar la codificación del sistema se decidió solo mostrar las partes principales: la conexión a la base de datos, inserción, consulta, modificación y eliminación de datos. Para cada una de estas áreas se construyen scripts que implementan diversas funciones, sin embargo, existen algunos que por su relevancia se presentan a continuación con el fin de mostrar la forma en la que están codificados

4.3.1 Conexión a la Base de Datos

Para conectar los scripts con la base de datos en MySQL se utilizó la siguiente función:

```
$serv = "localhost";  
$usr = "ana";  
$pass = "curso";  
$bd = "icsyh";  
  
$conx = mysql_connect($serv,$usr,$pass) or die("Conexion fallida");  
$resp = mysql_select_db($bd) or die("Error en la base de datos");
```

4.3.2 Inserción

En el caso de la inserción de datos hacia cualquier tabla de la base de datos, los scripts presentan ligeras variaciones entre cada uno, dependiendo de la funcionalidad con la que cuenten, se presenta en el siguiente script una versión para la tabla alumno.

```
include("default.php");
$nombre=$_REQUEST['nombre'];
$apellidos=$_REQUEST['apellidos'];
$sexo=$_REQUEST['sexo'];
$tiposangre=$_REQUEST['tiposangre'];
$email=$_REQUEST['email'];
$fechanacimiento=$_REQUEST['anio']."/".$_REQUEST['mes']."/".$_REQUEST['dia'];
$lugarnac=$_REQUEST['lugarnac'];
$pais=$_REQUEST['pais'];
$universidadproc=$_REQUEST['universidadproc'];
$direccion=$_REQUEST['direccion'];
$colonia=$_REQUEST['colonia'];
$cp=$_REQUEST['cp'];
$telcasa=$_REQUEST['telcasa'];
$telemergencia=$_REQUEST['telemergencia'];
$curp=$_REQUEST['curp'];
$fechaingreso=$_REQUEST['anio2']."/".$_REQUEST['mes2']."/".$_REQUEST['dia2'];
$observaciones=$_REQUEST['observaciones'];

$sql="INSERT INTO alumno (nombre, apellidos, sexo, tiposangre, email, fechanacimiento,
lugarnac, pais, universidadproc, direccion, colonia, cp, telcasa, telemergencia, curp,
fechaingreso, observaciones) VALUES ('".$nombre."', '".$apellidos."', '".$sexo."',
'".$tiposangre."', '".$email."', '".$fechanacimiento."', '".$lugarnac."', '".$pais."',
'".$universidadproc."', '".$direccion."', '".$colonia."', '".$cp."', '".$telcasa."', '".$telemergencia."',
'".$curp."', '".$fechaingreso."', '".$observaciones."' );

mysql_query($sql);
header("Location: MenuAlumno.html");
exit();
```

4.3.3 Consulta

Una de las funciones vitales del sistema es al mostrar información, por ese motivo la función de consulta es de suma importancia, a continuación se muestra una versión para la tabla alumno.

```
include("default.php");

$nombre=$_REQUEST['nombre'];
$apellidos=$_REQUEST['apellidos'];
$sexo=$_REQUEST['sexo'];
$tiposangre=$_REQUEST['tiposangre'];
$email=$_REQUEST['email'];
$fechanacimiento=$_REQUEST['fechanacimiento'];
$lugarnac=$_REQUEST['lugarnac'];
$pais=$_REQUEST['pais'];
$universidadproc=$_REQUEST['universidadproc'];
$direccion=$_REQUEST['direccion'];
$colonia=$_REQUEST['colonia'];
$cp=$_REQUEST['cp'];
$telcasa=$_REQUEST['telcasa'];
$telemergencia=$_REQUEST['telemergencia'];
$curp=$_REQUEST['curp'];
$fechaingreso=$_REQUEST['fechaingreso'];
```

```

$observaciones=$_REQUEST['observaciones'];

$sql="SELECT * alumno";

mysql_query($sql);
header("Location: ConsultarAlumno.php");
exit();

```

4.3.4 Modificación

La información de las tablas de la base de datos tiene que ser actualizada constantemente, es por eso que las funciones de modificación son importantes; aquí se presenta una versión que sufre cambios dependiendo de la tarea que ejecutan los scripts.

```

include("default.php");

$nombre=$_REQUEST['nombre'];
$apellidos=$_REQUEST['apellidos'];
$sexo=$_REQUEST['sexo'];
$tiposangre=$_REQUEST['tiposangre'];
$email=$_REQUEST['email'];
$fechanacimiento=$_REQUEST['fechanacimiento'];
$lugarnac=$_REQUEST['lugarnac'];
$pais=$_REQUEST['pais'];
$universidadproc=$_REQUEST['universidadproc'];
$direccion=$_REQUEST['direccion'];
$colonia=$_REQUEST['colonia'];
$cp=$_REQUEST['cp'];
$telcasa=$_REQUEST['telcasa'];
$telemergencia=$_REQUEST['telemergencia'];
$curp=$_REQUEST['curp'];
$fechaingreso=$_REQUEST['fechaingreso'];
$observaciones=$_REQUEST['observaciones'];

$sql="UPDATE alumno SET nombre=". $nombre. ", apellidos=". $apellidos. ",
sexo=". $sexo. ", tiposangre=". $tiposangre. ", email=". $email. ",
fechanacimiento=". $fechanacimiento. ", lugarnac=". $lugarnac. ",
pais=". $pais. ", universidadproc=". $universidadproc. ",
direccion=". $direccion. ", colonia=". $colonia. ", cp=". $cp. ",
telcasa=". $telcasa. ", telemergencia=". $telemergencia. ", curp=". $curp. ",
fechaingreso=". $fechaingreso. ", observaciones=". $observaciones. " WHERE
matricula=". $matricula;

mysql_query($sql);
header("Location: ActualizarAlumno.php");
exit();

```

4.3.5 Eliminación

La información de las tablas de la base de datos puede ser eliminada ya que tal vez sea información innecesaria y ya no se necesita en la base de datos, es por eso que las funciones de eliminación son importantes; aquí se presenta una versión que sufre cambios dependiendo de la tarea que ejecutan los scripts.

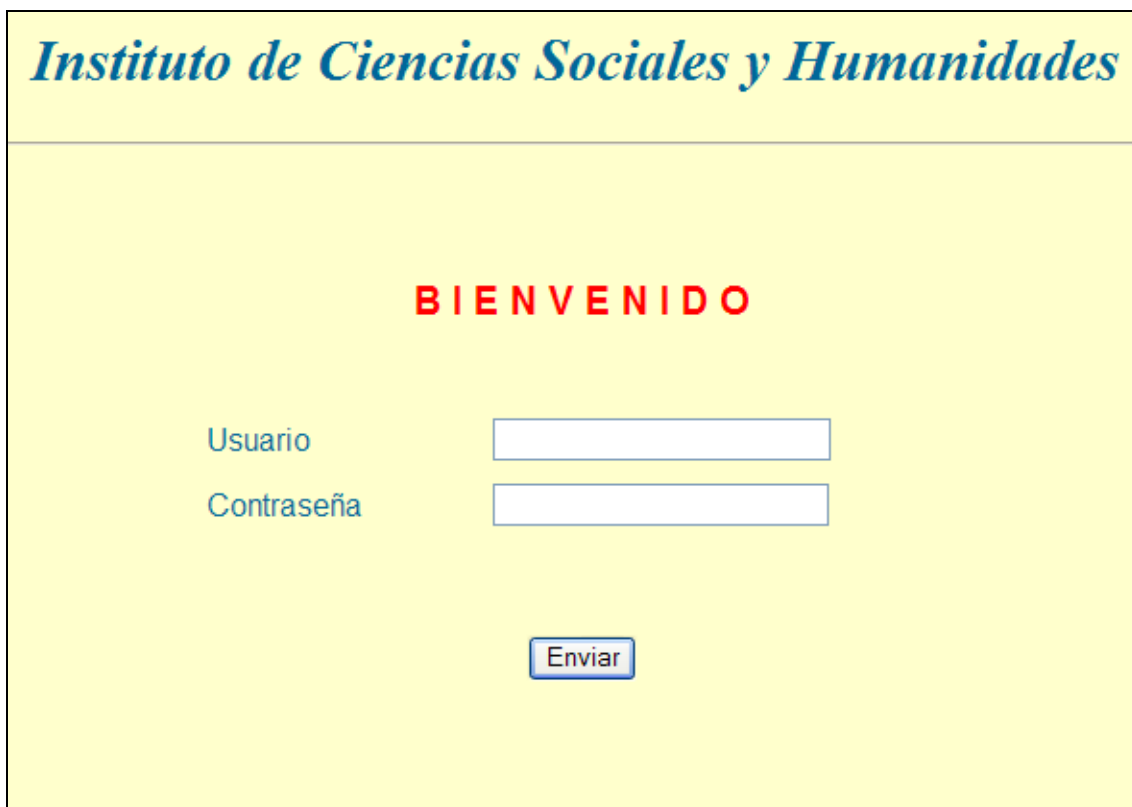
```
include("default.php");  
  
$matricula=$_REQUEST['matricula'];  
$sql="DELETE FROM alumno WHERE matricula =".$matricula;  
  
mysql_query($sql);  
header("Location: EliminarAlumno.php");  
exit();
```

4.4 PROTOTIPOS DE INTERFAZ DE USUARIO

Los prototipos de interfaz de usuario que se presentan a continuación son la realización de los casos de uso. Tal y como sucede con el punto anterior no se mostrarán todos los prototipos de interfaz realizados, sino únicamente los que se consideran principales.

4.4.1 Menú principal

El menú principal del sistema es el encargado de autenticar a los usuarios y posteriormente asignarles su sesión correspondiente. En la figura 4.3 se muestra el registro de los usuarios.



Instituto de Ciencias Sociales y Humanidades

BIENVENIDO

Usuario

Contraseña

Fig. 4.3 Menú Principal del Sistema

4.4.2 Menú Administrador

El menú del administrador es el encargado de mostrar al usuario administrador los casos de uso correspondientes, en el se muestran ligas que nos conectan con el caso de uso específico. En la figura 4.4 se muestra el menú del administrador.

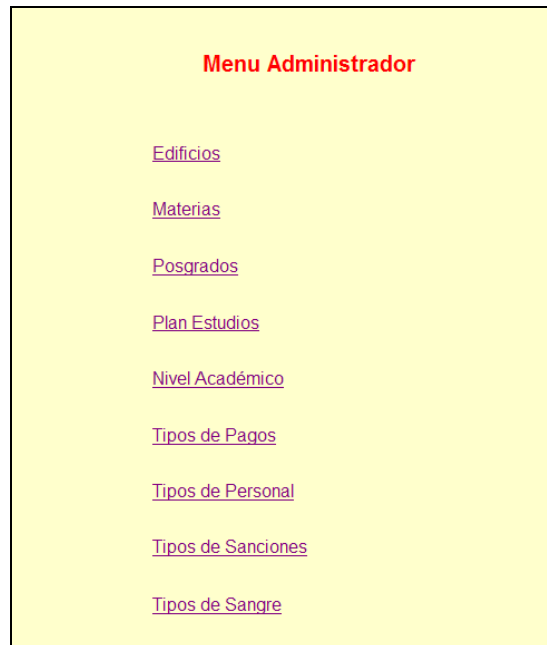


Fig. 4.4 Menú Administrador

4.4.3 Menú Secretaria

El menú del secretaria es el encargado de mostrar al usuario secretaria los casos de uso correspondientes, en el se muestran ligas que nos conectan con el caso de uso específico. En la figura 4.5 se muestra el menú del secretaria.



Fig. 4.5 Menú Secretaria.

4.4.4 Menú Director

El menú del director es el encargado de mostrar al usuario director los casos de uso correspondientes, en el se muestran ligas que nos conectan con el caso de uso específico. En la figura 4.6 se muestra el menú del director.



Fig. 4.6 Menú Director

4.4.5 Consulta de Alumno

Al ingresar al sistema se muestra una lista de los nombres y apellidos de los alumnos registrados junto con su número de matrícula, el sistema pide que el usuario introduzca el número de matrícula del alumno para poder visualizar sus datos, todo esto se muestra en la figura 4.7.

Matricula	Nombre	Apellidos
200900004	Lizbeth	Hernández Rojas
200900005	Jose	Rosete Juarez
200900006	Beatriz	Ramos Carreon
200900007	Luis Fernando	Limon Campos
200900008	Claudia	Vera Tlapa
200900009	Jesus	Ruiz Martinez
200900010	Maximo	Rojano Cienfuegos
200900011	Liliana	Lopez Cruz
200900012	Veronica	Xicontencatl Linarez
200900013	Irma	González Zavaleta
200900014	Adriana	Hernández Reyes
200900015	Luis Fernando	Reyes Hernández
200900016	Jacobo	Jimenez Muñoz
200900017	Rene	Martinez Becerril
200900018	Leticia Carmen	Becerril Pineda

Introduce el numero de matricula

Fig. 4.7 Consulta de Alumno

4.4.6 Alta de Alumno

Para que un alumno se registre en el sistema, la secretaria tiene que ingresar sus datos en el formulario de registro. Una vez que se han registrado los datos del alumno estos se guardan en la base de datos. Este formulario pide datos del alumno como son su nombre, apellidos, sexo, tipo de sangre, email, fecha de nacimiento, lugar de nacimiento, país, universidad de procedencia, dirección, colonia, código postal, teléfono de casa, teléfono de emergencia, clave única de registro de población, fecha de ingreso y observaciones, este formulario se muestra en la figura 4.8.

Nombre	<input type="text"/>
Apellidos	<input type="text"/>
Sexo	<input checked="" type="radio"/> Hombre <input type="radio"/> Mujer
Tipo Sangre	Tipo A Positivo <input type="button" value="v"/>
Email	<input type="text"/>
Fecha de nacimiento	1 <input type="button" value="v"/> Enero <input type="button" value="v"/>
Lugar de nacimiento	<input type="text"/>
País	<input type="text"/>
Universidad de procedencia	<input type="text"/>
Dirección	<input type="text"/>
Colonia	<input type="text"/>
Código postal	<input type="text"/>
Teléfono de casa	<input type="text"/>
Teléfono de emergencia	<input type="text"/>
CURP	<input type="text"/>
Fecha de ingreso	1 <input type="button" value="v"/> Enero <input type="button" value="v"/>
Observaciones	<input type="text"/>

Fig. 4.8 Alta de Alumno

4.5 PRUEBAS

Para verificar que todas las funciones implementadas en el sistema funcionen correctamente se eligieron una serie de casos de prueba.

El primer caso de prueba propuesto es el de alta de alumno, ya que es uno de los casos más importantes dentro del sistema. Para realizar la prueba se siguen los siguientes pasos:

La secretaria ingresa al sistema desde su menú y pulsa la liga de alumnos, dentro de ella pulsa la liga de alta de alumno.

Dentro del formulario la secretaria ingresa los datos correspondientes como se muestra en la figura 4.9.

Nombre	Leticia
Apellidos	Araiza Viveros
Sexo	<input type="radio"/> Hombre <input checked="" type="radio"/> Mujer
Tipo Sangre	Tipo B Positivo
Email	lety@hotmail.com
Fecha de nacimiento	11 Junio 1970
Lugar de nacimiento	Puebla
Pais	México
Universidad de procedencia	BUAP
Direccion	31 Norte 1321
Colonia	Centro
Codigo postal	72000
Telefono de casa	2345678
Telefono de emergencia	2221678905
CURP	AAVL640611MPLRVT00
Fecha de ingreso	4 Noviembre 2009
Observaciones	Ninguna

Fig. 4.9 Datos de Alta de Alumno

En el caso de la secretaria, puede verificar los datos ingresados en el caso de consulta de alumnos mostrado en la figura 4.10, pero para fines de esta prueba lo vamos a verificar directamente en la base de datos, esto se muestra en la figura 4.11.

Instituto de Ciencias Sociales y Humanidades

Consulta de alumnos

Matricula	Nombre	Apellidos
200900001	Leticia	Araiza Viveros

Introduce el numero de matricula

[Regresar](#)

Fig. 4.10 Consulta de Alumno desde el Sistema

The screenshot shows a window titled "[Table] alumno @icsyh (localhost)". The window contains a table with the following data:

nombre	apellidos	sexo	tiposangre	email	fechanacimiento	lugarna
Leticia	Araiza Viveros	M		3 lety@hotmail.com	1970-06-11	Puebla

The bottom of the window shows a SQL query: `SELECT * FROM `alumno` LIMIT 0,1000` and navigation controls. The status bar indicates "Record 1 of 1 in Page 1".

Fig. 4.11 Datos de Alumnos desde la Base de Datos

El segundo caso de prueba propuesto se refiere a actualizar los datos de un alumno, la secretaria es la encargada de realizar esta función, este caso se usa cuando hay algún error en los datos que se dieron de alta o simplemente porque los datos del alumno han cambiado, así que para realizar la prueba se siguen los siguientes pasos:

La secretaria ingresa al sistema desde su menú y pulsa la liga de alumnos, dentro de ella pulsa la liga de actualizar datos alumno.

El sistema muestra una lista con la matrícula, nombre y apellidos de todos los alumnos que están en la base de datos y pide la matrícula del alumno del cual se quieren actualizar los datos como se muestra en la figura 4.12

Instituto de Ciencias Sociales y Humanidades

Actualizar datos de alumno

Matricula	Nombre	Apellidos
200900001	Leticia	Araiza Viveros

Introduce el numero de matricula

[Regresar](#)

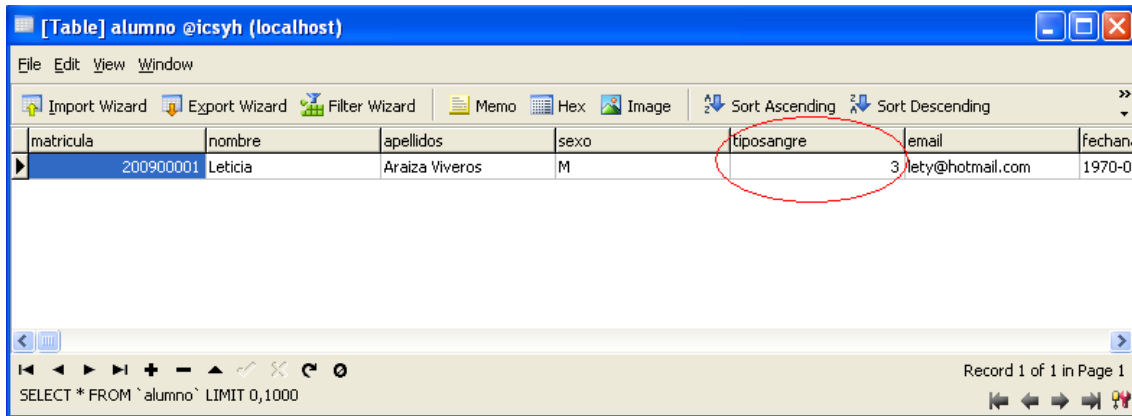
Fig. 4.12 Actualizar Datos Alumno

Dentro del formulario la secretaria actualiza los datos correspondientes, para este caso vamos a cambiar el tipo de sangre “Tipo B Positivo” a “Tipo O Positivo” como se muestra en la figura 4.13.

Matricula	<input type="text" value="200900001"/>
Nombre	<input type="text" value="Leticia"/>
Apellidos	<input type="text" value="Araiza Viveros"/>
Sexo	<input type="radio"/> Hombre <input checked="" type="radio"/> Mujer
Tipo Sangre	<input type="text" value="Tipo B Positivo"/> <input type="button" value="v"/>
Email	<input type="text"/>
Fecha de nacimiento	<input type="text"/>
Lugar de nacimiento	<input type="text"/>
Pais	<input type="text"/>
Universidad de procedencia	<input type="text" value="BUAP"/>
Direccion	<input type="text" value="31 Norte 1321"/>
Colonia	<input type="text" value="Centro"/>
Codigo postal	<input type="text" value="72000"/>
Telefono de casa	<input type="text" value="2345678"/>
Telefono de emergencia	<input type="text" value="2147483647"/>
CURP	<input type="text" value="AAVL640611MPLRVT00"/>
Fecha de ingreso	<input type="text" value="2009-11-04"/>
Observaciones	<input type="text" value="Ninguna"/>

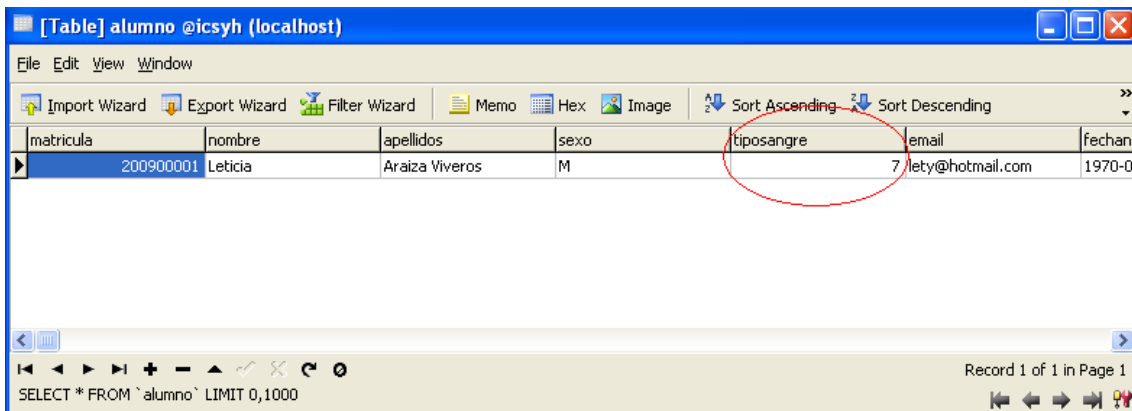
Fig. 4.13 Formulario de Actualización de Datos Alumno

Recordemos que el campo tiposangre es una llave foránea, entonces en la base de datos antes de hacer la actualización había un número 3 ya que este corresponde al tipo de sangre “Tipo B Positivo” como se muestra en la figura 4.14, al hacer la actualización tendrá que aparecer un número 7 que corresponde al tipo de sangre “Tipo O Positivo”, así que posteriormente dichos campos son modificados en la base de datos como se muestra en la figura 4.15.



matricula	nombre	apellidos	sexo	tiposangre	email	fecha
200900001	Leticia	Araiza Viveros	M	3	lety@hotmail.com	1970-0

Fig. 4.14 Datos Alumno Antes de Actualizar la Base de Datos



matricula	nombre	apellidos	sexo	tiposangre	email	fecha
200900001	Leticia	Araiza Viveros	M	7	lety@hotmail.com	1970-0

Fig. 4.15 Datos Alumno Después de Actualizar la Base de Datos

Cabe mencionar que se realizaron muchos otros casos de prueba de todo el sistema, sin embargo no todos están expresados ya que estos ejemplos son solo unas muestras de algunos casos que realiza el sistema con fin de ejemplificarlos.

CONCLUSIONES Y PERSPECTIVAS

El proyecto presentado fue finalizado cumpliendo con los objetivos planteados, el primero de ellos: crear un Sistema de Control Escolar para el Instituto de Ciencias Sociales y Humanidades.

El sistema lleva el control de alumnos y personal del instituto principalmente, además incorpora módulos como el control de pagos y sanciones, entre otras funcionalidades, cumpliendo con las expectativas del Instituto de Ciencias Sociales y Humanidades.

Para la realización de este proyecto se utilizaron herramientas de software libre, como lo son el lenguaje PHP y el manejador de bases de datos MySQL, por lo que el costo de licencias es nulo.

El sistema realizado puede modificarse en un futuro, para poder refinar ciertas características de acuerdo a requerimientos específicos del instituto, por ejemplo se puede implementar un módulo de alta de calificaciones para el caso de los profesores y otro para la consulta del historial académico de los alumnos. También es posible ampliar el sistema no sólo para uso del instituto, sino para su uso en otras instituciones educativas.

Se puede concluir que el Sistema de Control Escolar para el ICSyH es un proyecto producido bajo la metodología del Ciclo de Vida de Cascada, lo cual me permitió conocer una de las muchas formas de realizar software profesionalmente, además de llevar a cabo un proyecto que ayuda al control escolar del Instituto de Ciencias Sociales y Humanidades.

BIBLIOGRAFIA

- [1] Stephen R. Schach: *Ingeniería de Software Clásica y Orientada a Objetos*, McGraw Hill, 2006.
- [2] <http://zarza.usual.es/~fgarcia/docencia/isoftware/rincon.htm>
- [3] http://es.wikipedia.org/wiki/Desarrollo_de_software
- [4] <http://www.ati.es/novatica/2004/168/168-4.pdf>
- [5] Grady Booch, James Rumbaugh e Ivar Jacobson: *El Lenguaje Unificado de Modelado, Guía del Usuario 2/E*, Addison Wesley, 2006.
- [6] http://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso
- [7] http://es.wikipedia.org/wiki/Diagrama_de_actividades
- [8] http://es.wikipedia.org/wiki/Diagrama_de_secuencia
- [9] Pressman S. Roger: *Ingeniería de Software. Un enfoque práctico*, McGraw Hill, 2002.
- [10] http://es.wikipedia.org/wiki/Base_de_datos
- [11] Elmasri Ramez, Navathe Shamkant: *Fundamentos de sistemas de bases de datos*, 5ta. Edición, Pearson Addison-Wesley, 2007.
- [12] <http://www.monografias.com/trabajos11/basda/basda.shtml>
- [13] http://mx.geocities.com/vic_omar/db.htm
- [14] <http://www.monografias.com/trabajos12/basdat/basdat.shtml#COMPON>
- [15] Silberschatz Abraham, F. Korth Henry, Sudarshan S.: *Fundamentos de Bases de Datos*, McGraw Hill, 2002.
- [16] <http://www.mitecnologico.com/Main/ArquitecturaBaseDeDatos>
- [17] <http://www.monografias.com/trabajos72/base-datos/base-datos3.shtml>
- [18] http://sistemas.itlp.edu.mx/tutoriales/bassedat1/tema1_4.htm
- [19] <http://mysql.conclase.net/curso/index.php?cap=003>
- [20] http://es.wikipedia.org/wiki/Base_de_datos_relacional
- [21] <http://www.mysql-hispano.org/page.php?id=16>

- [22] http://es.wikipedia.org/wiki/Clave_for%C3%A1nea
- [23] Pérez César: *MySQL para Windows y Linux*, Alfaomega Ra-MA, 2008.
- [24] Sánchez Jorge: *MySQL Guía Rápida*, 2004.
- [25] http://descargar.mp3.es/lv/group/view/kl54868/Navicat_for_MySQL.htm
- [26] <http://solutions.mysql.com/solutions/item.php?id=605>
- [27] <http://www.maestrosdelweb.com/Principiantes/losdiferenteslenguajesdeprogramacionparalaweb/>
- [28] Gutiérrez Abraham, Bravo Ginés: *PHP5 a través de ejemplos*, Alfaomega Ra-Ma, 2008.
- [29] http://wapedia.mobi/es/Servidor_web
- [30] http://www.taringa.net/posts/downloads/1214036/WampServer-2_0c.html
- [31] http://es.wikipedia.org/wiki/Ciencias_de_la_computaci%C3%B3n