**PROTOTIPO DE SOFTWARE PARA EL CONTROL DE ASISTENCIA ESTUDIANTIL EN COTECNOVA**

**NATALIA AGUDELO VALDÉS**

**JUAN DAVID HOYOS RAMÍREZ**

**CORPORACIÓN DE ESTUDIOS TECNOLÓGICOS DEL NORTE DEL VALLE**

**TÉCNICO PROFESIONAL EN PROGRAMACIÓN DE APLICACIONES INFORMÁTICAS**

**CARTAGO**

**2019**

**PROTOTIPO DE SOFTWARE PARA EL CONTROL DE ASISTENCIA ESTUDIANTIL EN COTECNOVA**

**NATALIA AGUDELO VALDÉS**

**JUAN DAVID HOYOS RAMÍREZ**

**Trabajo de tesis para optar por el título de Técnico profesional en programación de aplicaciones informáticas**

**Profesor**

**Arvey Barahona Gómez**

**CORPORACIÓN DE ESTUDIOS TECNOLÓGICOS DEL NORTE DEL VALLE**

**TÉCNICO PROFESIONAL EN PROGRAMACIÓN DE APLICACIONES INFORMÁTICAS**

**CARTAGO**

**2019**

# RESUMEN

Mediante este documento se busca explicar los inconvenientes que se tienen al tomar la asistencia de forma tradicional y su organización; buscando así una alternativa más fiable para el control de los mismos en la institución COTECNOVA.

# ABSTRAC

Through this document, we seek to explain the inconveniences of taking assistance in a traditional way and his organization, looking for a more reliable alternative to control them in the COTECNOVA institution.**TABLA DE CONTENIDOS**

[RESUMEN III](#_Toc25144113)

[ABSTRAC IV](#_Toc25144114)

[INTRODUCCIÓN 1](#_Toc25144115)

[1. JUSTIFICACIÓN 2](#_Toc25144116)

[2. DIAGNÓSTICO DEL PROBLEMA 3](#_Toc25144117)

[3. OBJETIVOS 5](#_Toc25144118)

[3.1 GENERAL 5](#_Toc25144119)

[3.2 ESPECÍFICOS 5](#_Toc25144120)

[4. FUNDAMENTACIÓN TEÓRICA 6](#_Toc25144121)

[4.1 MARCO TEÓRICO 6](#_Toc25144122)

[4.1.1 Desarrollo de un sistema de base de datos 6](#_Toc25144123)

[4.1.2 Modelo entidad relación 8](#_Toc25144124)

[4.1.3 Modelo relacional 9](#_Toc25144125)

[4.1.4 Especificaciones de requerimientos 9](#_Toc25144126)

[4.1.4.1 Requisitos Funcionales 10](#_Toc25144127)

[4.1.4.2 Requisitos No Funcionales 10](#_Toc25144128)

[4.1.5 IEEE 830 11](#_Toc25144129)

[4.1.6 REM 11](#_Toc25144130)

[4.1.7 Mysql Workbench 11](#_Toc25144131)

[4.1.8 phpMyAdmin 13](#_Toc25144132)

[4.2 ANTEDECENTES 13](#_Toc25144133)

[5. RUTA METODOLÓGICA 14](#_Toc25144134)

[5.1 ENFOQUE 14](#_Toc25144135)

[5.2 MÉTODOS Y TIPO DE INVESTIGACIÓN 15](#_Toc25144136)

[5.3 TÉCNICAS DE RECOLECCIÓN DE DATOS 15](#_Toc25144137)

[5.4 POBLACIÓN Y MUESTRA 16](#_Toc25144138)

[6. DESARROLLO TEMÁTICO 17](#_Toc25144139)

[6.1 ESPECIFICACION DE REQUERIMIENTOS 17](#_Toc25144140)

[6.1.1 Participantes en el proyecto 17](#_Toc25144141)

[6.1.2 Descripción del sistema actual 18](#_Toc25144142)

[6.1.3 Objetivos del proyecto 19](#_Toc25144143)

[6.1.4 Catalogo de requisitos del sistema 21](#_Toc25144144)

[6.1.4.1 Requisitos de información 21](#_Toc25144145)

[6.1.4.2 Requisitos Funcionales 32](#_Toc25144146)

[6.1.4.2.1 Diagramas de caso de uso 34](#_Toc25144147)

[6.1.4.2.2 Definición de actores 34](#_Toc25144148)

[6.1.4.2.3 Casos de uso del sistema 35](#_Toc25144149)

[6.1.5 Requisitos no funcionales 47](#_Toc25144150)

[6.2 DISEÑO DE LA BASE DE DATOS 48](#_Toc25144151)

[6.2.1 Modelo entidad relación 48](#_Toc25144152)

[6.2.2 Modelo físico 49](#_Toc25144153)

[6.2.3 Implementación a phpmyadmin 50](#_Toc25144154)

[6.3 IMPLEMENTACIÓN DEL PROTOTIPO 55](#_Toc25144155)

[7. CONCLUSIONES 55](#_Toc25144156)

[8. CRONOGRAMA 56](#_Toc25144157)

[Bibliografía 58](#_Toc25144158)

**TABLA DE TABLAS**

Tabla 1 Organizaciones del proyecto 17

Tabla 2 Participantes del proyecto 18

Tabla 3 Objetivos del proyecto 19

Tabla 4 Requisito de estudiante 21

Tabla 5 Requisito de horario 22

Tabla 6 Requisito de tipo de usuario 23

Tabla 7 Requisito de clase 24

Tabla 8 Requisito de administrador 25

Tabla 9 Requisito de aula 26

Tabla 10 Requisito de materia 27

Tabla 11 Requisito de docente 28

Tabla 12 Requisito docente x clase 29

Tabla 13 Requisito de clase 30

Tabla 14 Requisitos funcionales 32

Tabla 15 Definición de actores 34

Tabla 16 Caso de uso 1 35

Tabla 17 Caso de uso 2 36

Tabla 18 Caso de uso 3 38

Tabla 19 Caso de uso 4 39

Tabla 20 Caso de uso 5 41

Tabla 21 Caso de uso 6 42

Tabla 22 Caso de uso 7 44

Tabla 23 Caso de uso 8 45

Tabla 24 Requisitos no funcionales 47

Tabla 25 Cronograma 56

**TABLA** **DE** **ILUSTRACIONES**

Ilustración 1 Ciclo de vida de una Base de Datos. Fuente: Universidad Autónoma del Estado de Hidalgo, plataforma virtual 7

Ilustración 2 Diagrama de casos de usos, fuente propia con ayuda de la herramienta Star UML 34

Ilustración 3 Modelo entidad relación, 48

Ilustración 4 Modelo físico 49

Ilustración 5 MySQL Workbench 50

Ilustración 6 Forward engineer 51

Ilustración 7 Forward engineer 52

Ilustración 8 Mensaje 53

Ilustración 9 phpmyadmin 53

Ilustración 10 phpmyadmin 54

# INTRODUCCIÓN

La asistencia estudiantil siempre ha sido uno de los mayores problemas en las diferentes instituciones llegando a ser incontrolable. Actualmente se han inventado varios mecanismos de control para esta, mejorando así el control de asistencia y entrega de material mediante dispositivos tecnológicos, esto ha hecho un gran auge en cuanto a las renovaciones de como se venía tomando anteriormente, dando así un impacto positivo.

# 1. JUSTIFICACIÓN

La implementación de este proyecto, y más precisamente, del software, busca mayor eficiencia y facilidad en la toma de asistencia a clases de los estudiantes, adicionalmente, permitirá el envío de material visto en el aula, así mismo, mejorará el tiempo en la cátedra al no disponer de minutos para la verificación/llamado a lista teniendo un mejor control y participación en clase de sus estudiantes.

Por su parte, los estudiantes recibirán los contenidos vistos y podrán constatar sus ausencias para una posterior descarga o justificación de la misma.

De esta forma, el resultado general del proyecto permitirá mostrar alternativas de solución y agilización acorde con las necesidades de la institución, permitiendo ser una guía para futuras investigaciones que puedan presentar el mismo problema.

# 2. DIAGNÓSTICO DEL PROBLEMA

La tecnología, a lo largo del tiempo, ha evolucionado exponencialmente llegando a ser de gran ayuda para el ser humano y ha facilitado más el trabajo para el mismo en diferentes áreas, y una de ellas son los controles de seguridad y acceso.

En Colombia, el uso de las tecnologías en cuanto la toma de asistencia aún es muy básico ya que en las instituciones se sigue usando el registro de asistencia de forma tradicional, formatos impresos, oral o lista de chequeo, cabe destacar que algunas ya han optado en la modernización de este proceso mediante los dispositivos tecnológicos e informáticos, como es el caso de la (Universidad de Antioquia, s.f), donde proponen la Tarjeta Integrada Personal (TIP) como un medio de identificación y control que permitirá el acceso a los diferentes servicios y beneficios de la comunidad académica (estudiantes, profesores y egresados), así como a los empleados, jubilados y contratistas.

En la corporación de Estudios Tecnológicos del Norte del Valle, fundada en 1971, el docente lleva un control de asistencia estudiantil de forma tradicional (llamados de lista en forma oral, lista de chequeos o simplemente se hace entrega de una hoja en la cual los estudiantes escriben sus nombres); el uso de este mecanismo puede ser poco eficaz a la hora de realizar el registro estudiantil a grupos numerosos, ya que gasta demasiado tiempo en el llamado, igualmente, el manejo de listados en papel puede resultar tedioso e incluso contaminante sino se hace un buen uso del mismo (reciclaje de papeles en los cuales puede usarse más de una vez); terminada la clase, el docente debe dejar un reporte de los estudiantes que no asistieron a la misma.

Por lo anterior, surge la pregunta:

¿Cómo identificar la asistencia estudiantil a las clases en COTECNOVA de tal manera que se utilice el menor tiempo posible, sin importar el tamaño del grupo?

# 3. OBJETIVOS

## 3.1 GENERAL

* Implementar un prototipo de software que permita la administración de la asistencia estudiantil en la Corporación de Estudios Tecnológicos del Norte del Valle.

## 3.2 ESPECÍFICOS

* Identificar los requerimientos del sistema para el control de asistencia estudiantil.
* Diseñar una base de datos que permita modelar el sistema de información de asistencia académica de COTECNOVA.
* Implementar el prototipo del software según las especificaciones.

# 4. FUNDAMENTACIÓN TEÓRICA

## 4.1 MARCO TEÓRICO

### 4.1.1 Desarrollo de un sistema de base de datos

A lo largo del tiempo, el ser humano ha ido evolucionando su forma de llevar la información de una manera mucho más cómoda, compacta y sencilla, teniendo a ella un acceso más fácil. A esto se le llama las Bases de Datos (en ocasiones abreviada BB.DD o BD), que “es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso” (Xunta de Galicia, 2015). Las bases de datos constituyen un sistema de proceso de datos cuyo objetivo básico es el de conservar información y mantenerla disponible para su acceso de forma eficiente. El interés de los usuarios por la información contenida en una base de datos es debido, normalmente, a su significación en los procesos de toma de decisiones (Martinez, s.f).

Las bases de datos tienen un sistema de desarrollo el cual se divide en varios procesos, como lo plantea (Delgado, s.f):

* Análisis.
* Diseño del modelo entidad / relación.
* Diseño del modelo relacional.
* Lenguaje SQL y base de datos final.

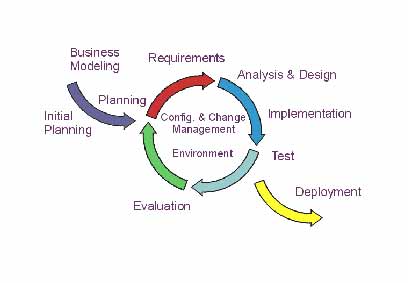


Ilustración 1 Ciclo de vida de una Base de Datos

Fuente: Universidad Autónoma del Estado de Hidalgo, plataforma virtual

* **Análisis:**

En este proceso se identifican todos los requisitos a fondo de lo que se quiere representar en la base de datos.

* **Diseño del Modelo Entidad / Relación:**

Cuando se empieza a hacer el diseño de la base de datos un problema común es que el encargado del diseño de la misma lo haga directamente en la máquina. Esto hace que en un alto grado se pierda la conceptualidad del problema quedando tal diseño faltante de detalles.

* **Modelo relacional:**

En este proceso, el diseñador ya puede implementar el modelo Entidad - Relación al modelo Relacional, teniendo en cuenta la teoría de la normalización.

* **Lenguaje sql y base de datos final:**

Este es el último paso, consiste en codificar en lenguaje SQL el modelo relacional hecho anteriormente.

### 4.1.2 Modelo entidad relación

El concepto de Modelo Entidad - Relación, presentado y creado por Peter P. Chen, menciona que este se centra más en la estructura lógica y abstracta de la información, siendo capaz de representar toda la semántica del mundo real por medio de entidades y relaciones.

Como se menciona en el libro de Chen, The Entity Relationship Model – Toward a Unified View of Data. El modelo de red proporciona una visión más natural de los datos al separar las entidades y las relaciones (hasta cierto punto), pero su capacidad para lograr la independencia de los datos ha sido un desafío. El modelo relacional se basa en la teoría relacional y puede lograr un alto grado de independencia de los datos, pero puede perder información semántica importante sobre el mundo real. El modelo de conjunto de entidades, que se basa en la teoría de conjuntos, también logra un alto grado de independencia de datos, pero su visualización de valores como "3" o "rojo" puede no ser natural para algunas personas.

El modelo de entidad-relación adopta la visión más natural de que el mundo real consiste en entidades y relaciones. Incorpora parte de la información semántica importante sobre el mundo real (Chen, 2002).

### 4.1.3 Modelo relacional

El modelo relacional, presentado por el Dr. Edgar Frank Codd, definió las bases del mismo a finales de los 60. En 1970 publicó el documento “A Relational Model of data for Large Shared Data Banks” (“Un modelo relacional de datos para grandes bancos de datos compartidos”), este modelo fue revolucionario puesto que consigue la independencia de las aplicaciones respecto a los datos.

Codd se apoya en los trabajos de los matemáticos Cantor y Childs (cuya teoría de conjuntos es la verdadera base del modelo relacional). Según Codd, los datos se agrupan en relaciones (actualmente llamadas tablas), las cuales son una estructura que aglutina datos referidos a una misma entidad de forma organizada. Las relaciones, además, estructuran los datos de forma independiente respecto a su almacenamiento real en la computadora. Es decir, es un elemento conceptual, no físico. Lo que Codd intentaba fundamentalmente es evitar que los usuarios de la base de datos tuvieran que verse obligadas a aprender los entresijos internos del sistema. Esto es lo que ocurría con el modelo en red, dominante cuando Codd diseñó el modelo relacional, que era bastante físico. Su enfoque fue revolucionario al evitar conceptos del mundo de la computación en su modelo (Sanchez, s.f).

### 4.1.4 Especificaciones de requerimientos

La Especificación de Requisitos de Software (ERS) es una de las fases más importante del proceso de desarrollo de software y es en esta fase donde se obtiene la descripción completa del comportamiento del futuro software o producto que se va a desarrollar. Por tanto, es en dicha fase que se crea el Documento Especificación de Requisito de Software que mencione previamente (Grupo Carricay, 2018).

Entre estos destacamos a los requisitos funcionales y no funcionales, como lo plantea (Grupo Carricay, 2018):

#### 4.1.4.1 Requisitos Funcionales

Los requisitos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reaccionará a determinados insumos. Cuando hablamos de las entradas, no necesariamente hablamos sólo de las entradas de los usuarios. Pueden ser interacciones con otros sistemas, respuestas automáticas, procesos predefinidos. En algunos casos, los requisitos funcionales de los sistemas también establecen explícitamente lo que el sistema no debe hacer. Es importante recordar esto: un RF puede ser también una declaración negativa. Siempre y cuando el resultado de su comportamiento sea una respuesta funcional al usuario o a otro sistema, es correcto. Y más aún, no sólo es correcto, sino que es necesario definirlo. Y eso nos lleva al siguiente punto.

#### 4.1.4.2 Requisitos No Funcionales

Se trata de requisitos que no se refieren directamente a las funciones específicas suministradas por el sistema (características de usuario), sino a las propiedades del sistema: rendimiento, seguridad, disponibilidad. En palabras más sencillas, no hablan de “lo que” hace el sistema, sino de “cómo” lo hace. Alternativamente, definen restricciones del sistema tales como la capacidad de los dispositivos de entrada/salida y la representación de los datos utilizados en la interfaz del sistema.

Los requisitos no funcionales se originan en la necesidad del usuario, debido a restricciones presupuestarias, políticas organizacionales, la necesidad de interoperabilidad con otros sistemas de software o hardware, o factores externos tales como regulaciones de seguridad, políticas de privacidad, entre otros.

### 4.1.5 IEEE 830

El estándar IEEE 830-1998 para el SRS(en inglés) o ERS (Especificación de requerimientos de software) según (Universidad ICESI, 2010), “Es un conjunto de recomendaciones para la especificación de los requerimiento o requisitos de software el cual tiene como producto final la documentación de los acuerdos entre el cliente y el grupo de desarrollo para así cumplir con la totalidad de exigencias estipuladas”.

### 4.1.6 REM

REM (Requirements Management) es una herramienta experimental gratuita de Gestión de Requisitos diseñada para soportar la fase de Ingeniería de Requisitos de un proyecto de desarrollo software de acuerdo con la metodología definida en la Tesis Doctoral "Un Entorno Metodológico de Ingeniería de Requisitos para Sistemas de Información", presentada por Amador Durán en septiembre de 2000 (Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla, 2004).

### 4.1.7 Mysql Workbench

MySQL Workbench, según su página oficial, es una herramienta gráfica para trabajar con servidores y bases de datos MySQL, este cubre cinco temas principales los cuales son:

* **Desarrollo SQL:** Permite crear y administrar conexiones a servidores de bases de datos. Además de permitirle configurar los parámetros de conexión, MySQL Workbench proporciona la capacidad de ejecutar consultas SQL en las conexiones de la base de datos utilizando el Editor SQL incorporado.
* **Modelado de datos (diseño):** Permite crear modelos de su esquema de base de datos gráficamente, realizar ingeniería inversa y directa entre un esquema y una base de datos en vivo, y editar todos los aspectos de su base de datos utilizando el completo Editor de tablas. El editor de tablas proporciona funciones fáciles de usar para editar tablas, columnas, índices, activadores, particiones, opciones, inserciones y privilegios, rutinas y vistas.
* **Administración del servidor:** Permite administrar instancias del servidor MySQL al administrar usuarios, realizar copias de seguridad y recuperación, inspeccionar datos de auditoría, ver el estado de la base de datos y monitorear el rendimiento del servidor MySQL.
* **Migración de datos:** Permite migrar desde Microsoft SQL Server, Microsoft Access, Sybase ASE, SQLite, SQL Anywhere, PostreSQL y otras tablas, objetos y datos RDBMS a MySQL. La migración también admite la migración de versiones anteriores de MySQL a las últimas versiones.
* **Soporte de MySQL Enterprise:** Soporte para productos Enterprise como MySQL Enterprise Backup, MySQL Firewall y MySQL Audit.

### 4.1.8 phpMyAdmin

phpMyAdmin es un software de código abierto, diseñado para manejar la administración y gestión de bases de datos MySQL a través de una interfaz gráfica de usuario. PhpMyAdmin cada vez más tiene la mayor lista de características; soporta todas las operaciones de uso común tales como la navegación, creación, modificación de las bases de datos MySQL, las tablas, campos e índices. Además, phpMyAdmin permite administrar los usuarios MySQL y sus privilegios. Otra característica común es su función de importación. Con phpMyAdmin, importar base de datos MySQL de copia de seguridad es fácil y se puede importar un volcado SQL o CSV con unos clics del ratón. También, puede exportarse en formatos como CSV, SQL,XML, Excel, entre otros (Carlos, s.f).

## 4.2 ANTEDECENTES

La Universidad Loyola o Universidad Plena presentó un proyecto en el cual se trabaja, actualmente, sobre un sistema de control personal conformado por una serie de estudiantes donde se habla de una base de datos, esta empieza su proceso primero registrando al empleado por medio de un intermediario (jefe de área), luego realizan una serie de consultas de dichos empleados y una configuración de cada uno de sus aspectos, como por ejemplo, los permisos establecidos para cada uno de ellos o la modificación de uno de sus datos personales, por último, por medio del dispositivo a implementarse se recoge la información que será de gran importancia como la hora de entrada y de salida del empleado y de esa forma tener un mejor control sobre sus empleados.

Por otro lado, la compañía Samsung ha impulsado un proyecto junto a la Universidad de Salamarca para controlar la asistencia de sus alumnos por medio del dispositivo móvil de los estudiantes y otro dispositivo NFC localizado dentro del aula de clase, el estudiante deberá tener la aplicación descargada en su respectivo móvil ya que este estará enlazada con la base de datos la cual tendrá la información del estudiante y un registro de las clases a las cuales ha asistido, lo que se busca con la idea planteada es que el estudiante “pase su smartphone cerca al dispositivo localizado dentro del aula, seguido esto recibirá una confirmación en su dispositivo móvil sobre su registro en la clase” (Hernandez, 2012), en cuanto al docente, este también tendrá acceso a la aplicación y podrá ver la información almacenada en la base de datos la cual tendrá el  registro de los alumnos registrados.

Por última, en la universidad Autónoma de Guerrero, la unidad académica de ingeniería llevó a cabo un prototipo de sistema para el control de asistencia a docentes ya que una de sus principales problemáticas era que el administrador perdía mucho tiempo al generar el reporte de faltas o retardo para descuento de nómina. Para la realización de la base de datos se primero debían registrar a todo docente por medio de un tercero (administrador), seguido a esto el docente registrado deberá registrar la asistencia el cual tendrá que registrar la entrada y salida para así hacer el respectivo reporte.

# 5. RUTA METODOLÓGICA

## 5.1 ENFOQUE

El tipo de enfoque que ayudará en este proyecto de investigación será el enfoque cualitativo, este consiste en el estudio de métodos de recolección de datos de tipo descriptivo y de observaciones para descubrir de manera discursiva categorías conceptuales (Sanjahingu, 2014).

## 5.2 MÉTODOS Y TIPO DE INVESTIGACIÓN

Los métodos que se emplearán serán: El método analítico, el cual consiste en el estudio de las partes que conforman un todo, estableciendo sus relaciones de naturaleza, causa y efecto (Sanjahingu, 2014), y el tipo de investigación será el de investigación aplicada que, según (DuocUC, s.f), se refiere al tipo de estudios científicos orientados a resolver problemas de la vida cotidiana y a controlar situaciones prácticas. De este modo, la Investigación Aplicada se centra en la resolución de problemas en un contexto determinado, es decir, busca la aplicación o utilización de conocimientos, desde una o varias áreas especializadas, con el propósito de implementarlos de forma práctica para satisfacer necesidades concretas, proporcionando una solución a problemas del sector social o productivo.

## 5.3 TÉCNICAS DE RECOLECCIÓN DE DATOS

Las técnicas de recolección de información a utilizar serán:

* La observación: Este permite estudiar los aspectos más significativos de los objetos, hechos, situaciones sociales o personas en el contexto donde se desarrollan normalmente; permitiendo la comprensión de la verdadera realidad del fenómeno (Liberato, 2013).
* La entrevista: Permite recolectar información mediante el contacto directo con las personas a través de una conversación interpersonal, preparada bajo una dinámica de preguntas y respuestas, donde se dialoga sobre un tópico relacionado con la problemática de investigación (Liberato, 2013).

## 5.4 POBLACIÓN Y MUESTRA

La población a ser estudiada se tomará como un grupo focal, el cual será un grupo de estudiantes del programa Técnico Profesional en Programación de Aplicaciones Informáticas de la jornada diurna.

La muestra en primer lugar son los estudiantes de cuarto semestre de la jornada diurna en el programa de ingeniería en sistemas en COTECNOVA, pero en realidad, en un producto final, es para toda la comunidad académica.

# 6. DESARROLLO TEMÁTICO

## 6.1 ESPECIFICACION DE REQUERIMIENTOS

Según los conceptos planteados anteriormente, los requisitos de este proyecto planean desarrollarse en la normativa IEEE 830 junto con el programa REM y su estructura.

La asistencia estudiantil siempre ha sido uno de los mayores problemas en las diferentes instituciones llegando a ser incontrolable. Actualmente se han inventado varios mecanismos de control para esta, mejorando así el control de la misma mediante dispositivos tecnológicos, esto ha hecho un gran auge en cuanto a las renovaciones de como se venía tomando anteriormente, dando así un impacto positivo.

El sistema es un sistema requerido por la Corporación de Estudios Tecnológicos del Norte del Valle el cual planea ayudar en el control de la asistencia estudiantil para, posteriormente, llevar un mejor manejo de la misma.

### 6.1.1 Participantes en el proyecto

Tabla 1 Organizaciones del proyecto

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Organización** | **Dirección** | **Teléfono** | **Fax** | **Comentarios** |
| **Cotecnova** | Cl. 10 #3-95, Cartago, Valle del Cauca | 2111804 | 213 44 21. Cartago, Valle del Cauca | Ninguno |
| **Grupo** | No aplica | No aplica | No aplica | Ninguno |

*Fuente: Fuente propia, con ayuda de la herramienta REM*

Tabla 2 Participantes del proyecto

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Participante** | **Organización** | **Rol** | **Desarrollador** | **Cliente** | **Usuario** | **Comentarios** |
| **Natalia Agudelo Valdés** | Grupo | Desarrollador principal | Sí | No | No | Ninguno |
| **Juan David Hoyos** | Grupo | Desarrollador principal | Sí | No | No | Ninguno |
| **Arvey Barahona** | Grupo | Profesor | No | Sí | Sí | Ninguno |

*Fuente: Fuente propia, con ayuda de la herramienta REM*

### 6.1.2 Descripción del sistema actual

En la corporación de Estudios Tecnológicos del Norte del Valle, fundada en 1971, el docente lleva un control de asistencia estudiantil de forma tradicional (documentos impresos como listas de chequeo, llamado de forma oral, entre otros); el uso de este mecanismo puede ser poco eficaz a la hora de realizar registro estudiantil a grupos numerosos, ya que gasta demasiado tiempo en el llamado a lista; terminada la clase, el docente debe dejar un reporte de los estudiantes que no asistieron a la misma.

### 6.1.3 Objetivos del proyecto

Tabla 3 Objetivos del proyecto

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Objetivos** | **Descripción** | **Subobjetivos** | **Importancia** | **Urgencia** | **Estado** | **Estabilidad** | **Comentarios** |
| **Datos estudiantiles y docentes** | El sistema deberá *recolectar todos los datos acerca de los estudiantes y docentes* | Ninguno | vital | inmediatamente | en construcción | alta | Ninguno |
| **Control de estudiantes y docentes** | El sistema deberá *apuntar al control de estudiantes y docentes que han sido registrados y asignados a sus clases* | Ninguno | vital | inmediatamente | en construcción | alta | Ninguno |
| **Datos clases** | El sistema deberá *permitir almacenar toda la información respectiva a las clases* | Ninguno | importante | hay presión | en construcción | alta | Ninguno |
| **Datos horarios** | El sistema deberá *permitir ingresar y modificar los respectivos datos de los horarios que fueron asignados a los estudiantes y docentes* | Ninguno | importante | hay presión | en construcción | alta | Ninguno |
| **Datos administrador** | El sistema deberá *almacenar los datos de los administrador* | Ninguno | vital | inmediatamente | en construcción | alta | Ninguno |
| **Datos aulas** | El sistema deberá *almacenar los datos de las distintas aulas de clase* | Ninguno | importante | hay presión | en construcción | alta | Ninguno |
| **Datos Materias** | El sistema deberá *almacenar los datos de las distintas materias que se usarán para los horarios* | Ninguno | vital | inmediatamente | en construcción | alta | Ninguno |
| **Datos carreras** | El sistema deberá *almacenar la información de todas las carreras disponibles* | Ninguno | vital | inmediatamente | en construcción | alta | Ninguno |

Fuente: Fuente propia, con ayuda de la herramienta REM

### 6.1.4 Catalogo de requisitos del sistema

#### 6.1.4.1 Requisitos de información

Tabla 4 Requisito de estudiante

|  |  |  |
| --- | --- | --- |
| **IRQ-0001** | **Estudiante** | |
| **Dependencias** |  [OBJ-0001] Datos estudiantiles y docentes   [OBJ-0002] Control de estudiantes y docentes | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a los estudiantes. En concreto: | |
| **Datos específicos** | * Id ( Identificador del estudiante) * Documento ( Numero de T.I o C.C) * Nombres ( Nombres completos de la persona) * Apellidos ( Apellidos completos de la persona) * Jornada ( Diurna, Nocturna, Sabatina) * Semestre ( I, II, III...) * Tipo de usuario\_id ( Id que identifica que tipo de usuario es) * Horario\_id ( Id del horario correspondiente) * Carrera\_id ( Id de la carrera correspondiente) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 500 año(s) | 999 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | media | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 5 Requisito de horario

|  |  |  |
| --- | --- | --- |
| **IRQ-0002** | **Horario** | |
| **Dependencias** |  [OBJ-0008] Datos Materias   [OBJ-0005] Datos horarios   [OBJ-0001] Datos estudiantiles y docentes   [OBJ-0007] Datos aulas | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a el horario del respectivo tipo de usuario. En concreto: | |
| **Datos específicos** | * Id ( Identificador del horario) * Hora ( Hora a la que se ve la respectiva materia) * Aulas\_id ( Aula en la que se dará la respectiva materia) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 500 año(s) | 999 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | media | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 6 Requisito de tipo de usuario

|  |  |  |
| --- | --- | --- |
| **IRQ-0003** | **Tipo de usuario** | |
| **Versión** | 1.0 ( 23/10/2019 ) | |
| **Dependencias** |  [OBJ-0006] Datos administrador   [OBJ-0001] Datos estudiantiles y docentes | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a los tipos de usuarios. En concreto: | |
| **Datos específicos** | * Id\_usuario ( 1, 2 o 3) * Nombre del usuario ( Estudiante, Administrador o Docente) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 500 año(s) | 999 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | media | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 7 Requisito de clase

|  |  |  |
| --- | --- | --- |
| **IRQ-0004** | **Clase** | |
| **Dependencias** |  [OBJ-0004] Datos clases   [OBJ-0001] Datos estudiantiles y docentes   [OBJ-0007] Datos aulas | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a las clases. En concreto: | |
| **Datos específicos** | * Id\_clase ( Identificador de la clase) * Dia ( Dia de la clase) * horario\_id ( Id del horario) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 500 año(s) | 999 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | hay presión | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 8 Requisito de administrador

|  |  |  |
| --- | --- | --- |
| **IRQ-0005** | **Administrador** | |
| **Dependencias** |  [OBJ-0006] Datos administrador | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a el administrador. En concreto: | |
| **Datos específicos** | * Id administrador ( Identificador del administrador) * Documento ( Numero del documento) * Nombres ( Nombres completos del administrador) * Apellidos ( Apellidos completos del administrador) * Tipo de usuario\_id ( Id del tipo) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 500 año(s) | 999 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 9 Requisito de aula

|  |  |  |
| --- | --- | --- |
| **IRQ-0006** | **Aula** | |
| **Dependencias** |  [OBJ-0007] Datos aulas | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a las aulas. En concreto: | |
| **Datos específicos** | * Id\_aula ( Del aula) * Nombre ( Del aula) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 999 año(s) | 500 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | importante | |
| **Urgencia** | hay presión | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 10 Requisito de materia

|  |  |  |
| --- | --- | --- |
| **IRQ-0007** | **Materia** | |
| **Dependencias** |  [OBJ-0008] Datos Materias | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a las materias. En concreto: | |
| **Datos específicos** | * Id\_materia ( De la materia) * Nombre ( De la materia) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 999 año(s) | 500 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 11 Requisito de docente

|  |  |  |
| --- | --- | --- |
| **IRQ-0008** | **Docente** | |
| **Dependencias** |  [OBJ-0001] Datos estudiantiles y docentes   [OBJ-0002] Control de estudiantes y docentes | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a los docentes. En concreto: | |
| **Datos específicos** | * Id ( Identificador del docente) * Documento ( Numero de C.C) * Nombres ( Nombres completos de la persona) * Apellidos ( Apellidos completos de la persona) * Tipo de usuario\_id ( Id que identifica que tipo de usuario es) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 500 año(s) | 999 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | media | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 12 Requisito docente x clase

|  |  |  |
| --- | --- | --- |
| **IRQ-0009** | **Docente x clase** | |
| **Dependencias** |  [OBJ-0004] Datos clases   [OBJ-0001] Datos estudiantiles y docentes | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a a la clase, el docente que se presenta y los estudiantes. En concreto: | |
| **Datos específicos** | * Clase\_id ( Identificador de la clase) * Docente\_id ( Identificador del docente) * Estudiante\_id ( Identificador del estudiante) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 500 año(s) | 999 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 13 Requisito de clase

|  |  |  |
| --- | --- | --- |
| **IRQ-0010** | **Carrera** | |
| **Dependencias** |  [OBJ-0009] Datos carreras | |
| **Descripción** | El sistema deberá almacenar la información correspondiente a las carreras. En concreto: | |
| **Datos específicos** | * Id\_carrera ( Identificador de la carrera) * Nombre ( Nombre de la carrera) | |
| **Tiempo de vida** | **Medio** | **Máximo** |
| 500 año(s) | 999 año(s) |
| **Ocurrencias simultáneas** | **Medio** | **Máximo** |
| 500 | 999 |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

#### 6.1.4.2 Requisitos Funcionales

Tabla 14 Requisitos funcionales

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Requisitos funcionales** | **Dependencias** | **Descripción** | **Importancia** | **Urgencia** | **Estado** | **Estabilidad** | **Comentarios** |
| **Validación** | Ninguno | El sistema deberá *Validar automáticamente al estudiante y/o docente en la gestión de usuarios y docentes.* | importante | inmediatamente | en construcción | baja | Ninguno |
| **Descripción** | Ninguno | El sistema deberá *proporcionar descripciones de los usuarios al ser consultados en la base de datos.* | vital | inmediatamente | en construcción | baja | Ninguno |
| **Agregar Usuarios** | Ninguno | El sistema deberá *permitir la creación de nuevos usuarios (Docentes y/o estudiantes) en caso de que lo requiera* | importante | inmediatamente | en construcción | baja | Ninguno |
| **Asignación** | Ninguno | El sistema deberá *asignar a cada estudiante un identificador único para luego ser utilizado en todos los procesos que se realicen sobre este.* | vital | inmediatamente | en construcción | media | Ninguno |
| **Administración** | Ninguno | El sistema deberá *permitir tener varios o un administrador* | vital | inmediatamente | en construcción | alta | Ninguno |
| **Horarios** | Ninguno | El sistema deberá *permitir a los docentes y/o estudiantes consultar sus respectivos horarios* | vital | inmediatamente | en construcción | media | Ninguno |
| **Envío automatizado** | Ninguno | El sistema deberá *permitir el envío automatizado del registro de usuarios que estuvieron en la clase a una base de datos superior o global* | vital | inmediatamente | en construcción | media | Ninguno |
| **Aulas** | Ninguno | El sistema deberá *permitir crear, ingresar y/o modificar los datos de las aulas* | importante | hay presión | en construcción | alta | Ninguno |
| **Materias** | Ninguno | El sistema deberá *permitir la creación y/o modificación de las materias en caso de que se requiera* | vital | inmediatamente | en construcción | baja | Ninguno |
| **Carrera** | Ninguno | El sistema deberá *proporcionar y/o almacenar información sobre las carreras existentes* | vital | inmediatamente | en construcción | alta | Ninguno |

Fuente: Fuente propia, con ayuda de la herramienta REM

##### 6.1.4.2.1 Diagramas de caso de uso

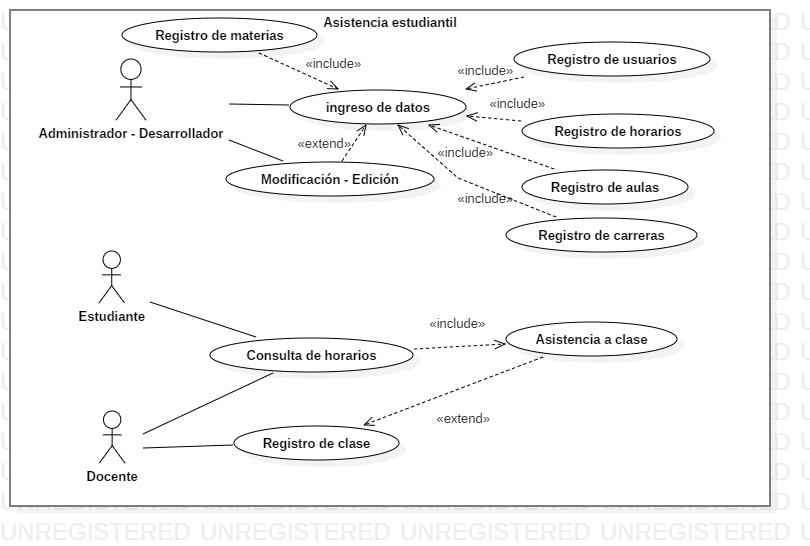


Ilustración 2 Diagrama de casos de usos

Fuente: Fuente propia con ayuda de la herramienta Star UML

##### 6.1.4.2.2 Definición de actores

Tabla 15 Definición de actores

|  |  |  |
| --- | --- | --- |
| **Actores** | **Descripción** | **Comentarios** |
| **Desarrollador/Administrador** | Este actor representa *a la persona que se encarga de administrar el sistema* | Ninguno |
| **Docente** | Este actor representa *al docente como tal, el cual maneja su propio horario* | Ninguno |
| **Estudiante** | Este actor representa *al estudiante como tal para la asistencia a clases* | Ninguno |

Fuente: Fuente propia, con ayuda de la herramienta REM

##### 6.1.4.2.3 Casos de uso del sistema

Tabla 16 Caso de uso 1

|  |  |  |
| --- | --- | --- |
| **UC-0001** | **Envío de información a la BD** | |
| **Dependencias** | Ninguno | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el mismo sistema envíe los datos | |
| **Precondición** | Debe de existir una BD para enviar los datos  Debe de haber un administrador registrado o existente | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El actor Desarrollador/Administrador (ACT-0002) ingresa al sistema |
| 2 | El actor Desarrollador/Administrador (ACT-0002) hace su respectiva función de ingreso de datos |
| 3 | El actor Desarrollador/Administrador (ACT-0002) envía los datos |
| 4 | El sistema el sistema valida que todos los campos estén llenos |
| 5 | El sistema el sistema rectifica que la BD esté conectada |
| 6 | El sistema el sistema envía los datos a la BD |
| **Postcondición** | Los datos quedan guardados en la BD | |
| **Excepciones** | **Paso** | **Acción** |
| - | - |
| **Rendimiento** | **Paso** | **Tiempo máximo** |
| - | - |
| **Frecuencia esperada** | 999 veces por año(s) | |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | registro de un nuevo estudiante | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 17 Caso de uso 2

|  |  |  |
| --- | --- | --- |
| **UC-0002** | **Validación Clase** | |
| **Dependencias** |  [IRQ-0004] Clase   [IRQ-0009] Docente x clase | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un docente ingresa al sistema | |
| **Precondición** | debe de estar registrado | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El actor Docente (ACT-0003) ingresa a la plataforma |
| 2 | El actor Docente (ACT-0003) ingresa sesión como docente |
| 3 | El sistema valida la existencia del docente |
| 4 | El actor Docente (ACT-0003) procede al módulo de clases |
| 5 | El actor Docente (ACT-0003) inicia/registra su clase |
| **Postcondición** | el docente puede registrar su clase | |
| **Excepciones** | **Paso** | **Acción** |
| 3 | Si el docente no existe, se realiza el caso de uso Registrar Docentes (UC-0006) , a continuación este caso de uso continúa |
| **Rendimiento** | **Paso** | **Tiempo máximo** |
| - | - |
| **Frecuencia esperada** | 999 veces por año(s) | |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Registro de clase | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 18 Caso de uso 3

|  |  |  |
| --- | --- | --- |
| **UC-0003** | **Registrar Docentes** | |
| **Dependencias** |  [IRQ-0005] Administrador   [IRQ-0003] Tipo de usuario   [IRQ-0009] Docente x clase   [IRQ-0008] Docente | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el desarrollador/administrador entre al sistema o durante la realización de los siguientes casos de uso: [UC-0004] Validación Clase | |
| **Precondición** | Debe existir un Administrador/Desarrollador registrado | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El actor Desarrollador/Administrador (ACT-0002) entra al sistema |
| 2 | El actor Desarrollador/Administrador (ACT-0002) entra al módulo de maestros |
| 3 | El actor Desarrollador/Administrador (ACT-0002) ingresa los datos del docente en el sistema |
| 4 | El sistema el sistema envía los datos del nuevo docente a la BD |
| 5 | El actor Docente (ACT-0003) queda registrado en la BD |
| **Postcondición** | Maestro registrado | |
| **Excepciones** | **Paso** | **Acción** |
| 3 | Si el docente ya está registrado, el sistema debe permitir modificar los datos, a continuación este caso de uso continúa |
| **Rendimiento** | **Paso** | **Tiempo máximo** |
| - | - |
| **Frecuencia esperada** | 999 veces por año(s) | |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Registro de un docente en el sistema | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 19 Caso de uso 4

|  |  |  |
| --- | --- | --- |
| **UC-0004** | **Registrar Estudiantes** | |
| **Dependencias** |  [IRQ-0001] Estudiante   [IRQ-0003] Tipo de usuario   [IRQ-0005] Administrador   [IRQ-0009] Docente x clase | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el desarrollador/administrador entre al sistema | |
| **Precondición** | Debe existir un Administrador/Desarrollador registrado | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El actor Desarrollador/Administrador (ACT-0002) entra al sistema |
| 2 | El actor Desarrollador/Administrador (ACT-0002) entra al módulo de estudiantes |
| 3 | El actor Desarrollador/Administrador (ACT-0002) ingresa los datos del estudiante en el sistema |
| 4 | El sistema el sistema envía los datos del nuevo estudiante a la BD |
| 5 | El actor Estudiante (ACT-0004) el estudiante queda registrado en la BD |
| **Postcondición** | Estudiante registrado | |
| **Excepciones** | **Paso** | **Acción** |
| 3 | Si el estudiante ya está registrado, el sistema debe permitir modificar los datos, a continuación este caso de uso continúa |
| **Rendimiento** | **Paso** | **Tiempo máximo** |
| - | - |
| **Frecuencia esperada** | 999 veces por año(s) | |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Registro de un estudiante en el sistema | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 20 Caso de uso 5

|  |  |  |
| --- | --- | --- |
| **UC-0005** | **Registro de horarios** | |
| **Dependencias** |  [IRQ-0005] Administrador   [IRQ-0002] Horario | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el desarrollador/administrador ingrese a administrar los horarios | |
| **Precondición** | Deben existir estudiantes registrados en el sistema  Deben existir docentes registrados en el sistema | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El actor Desarrollador/Administrador (ACT-0002) ingresa al sistema |
| 2 | El actor Desarrollador/Administrador (ACT-0002) ingresa al módulo de horarios |
| 3 | El actor Desarrollador/Administrador (ACT-0002) crea el horario e ingresa los datos correspondientes |
| 4 | El sistema almacena el horario en la BD |
| **Postcondición** | Horario registrado | |
| **Excepciones** | **Paso** | **Acción** |
| 3 | Si el horario ya existe, el sistema debe permitir modificar los datos, a continuación este caso de uso continúa |
| **Rendimiento** | **Paso** | **Tiempo máximo** |
| - | - |
| **Frecuencia esperada** | 999 veces por año(s) | |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 21 Caso de uso 6

|  |  |  |
| --- | --- | --- |
| **UC-0006** | **Registro de aulas** | |
| **Dependencias** |  [IRQ-0005] Administrador   [IRQ-0006] Aula | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el desarrollador/administrador desee ingresar nuevas aulas o modificarlas | |
| **Precondición** | Debe existir un Administrador/Desarrollador registrado | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El actor Desarrollador/Administrador (ACT-0002) ingresa al sistema |
| 2 | El actor Desarrollador/Administrador (ACT-0002) ingresa al módulo de aulas |
| 3 | El actor Desarrollador/Administrador (ACT-0002) crea el aula e ingresa los datos correspondientes |
| 4 | El sistema almacena el aula en la BD |
| **Postcondición** | Aulas registradas | |
| **Excepciones** | **Paso** | **Acción** |
| 3 | Si el aula ya existe, el sistema debe permitir modificar, a continuación este caso de uso continúa |
| **Rendimiento** | **Paso** | **Tiempo máximo** |
| - | - |
| **Frecuencia esperada** | 999 veces por año(s) | |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 22 Caso de uso 7

|  |  |  |
| --- | --- | --- |
| **UC-0007** | **Registro de materias** | |
| **Dependencias** |  [IRQ-0005] Administrador   [IRQ-0007] Materia | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el desarrollador/administrador desee ingresar nuevas materias o modificarlas | |
| **Precondición** | Debe existir un Administrador/Desarrollador registrado | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El actor Desarrollador/Administrador (ACT-0002) ingresa al sistema |
| 2 | El actor Desarrollador/Administrador (ACT-0002) ingresa al módulo de materias |
| 3 | El actor Desarrollador/Administrador (ACT-0002) crea la materia e ingresa los datos correspondientes |
| 4 | El sistema almacena la materia/as en la BD |
| **Postcondición** | Materias registradas | |
| **Excepciones** | **Paso** | **Acción** |
| 3 | Si la materia ya existe, el sistema debe permitir modificar los datos, a continuación este caso de uso continúa |
| **Rendimiento** | **Paso** | **Tiempo máximo** |
| - | - |
| **Frecuencia esperada** | 999 veces por año(s) | |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

Tabla 23 Caso de uso 8

|  |  |  |
| --- | --- | --- |
| **UC-0008** | **Registro de carrera** | |
| **Dependencias** |  [IRQ-0010] Carrera   [IRQ-0005] Administrador | |
| **Descripción** | El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el desarrollador/administrador desee ingresar nuevas carreras o modificarlas | |
| **Precondición** | Debe existir un Administrador/Desarrollador registrado | |
| **Secuencia normal** | **Paso** | **Acción** |
| 1 | El actor Desarrollador/Administrador (ACT-0002) ingresa al sistema |
| 2 | El actor Desarrollador/Administrador (ACT-0002) ingresa al módulo de carreras |
| 3 | El actor Desarrollador/Administrador (ACT-0002) crea la carrera e ingresa los datos correspondientes |
| 4 | El sistema almacena la carrera/as en la BD |
| **Postcondición** | Carreras registradas | |
| **Excepciones** | **Paso** | **Acción** |
| 3 | Si la carrera ya existe, el sistema debe permitir modificar los datos, a continuación este caso de uso continúa |
| **Rendimiento** | **Paso** | **Tiempo máximo** |
| - | - |
| **Frecuencia esperada** | 999 veces por año(s) | |
| **Importancia** | vital | |
| **Urgencia** | inmediatamente | |
| **Estado** | en construcción | |
| **Estabilidad** | alta | |
| **Comentarios** | Ninguno | |

Fuente: Fuente propia, con ayuda de la herramienta REM

### 6.1.5 Requisitos no funcionales

Tabla 24 Requisitos no funcionales

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Requisitos no funcionales** | **Dependencias** | **Descripción** | **Importancia** | **Urgencia** | **Estado** | **Estabilidad** | **Comentarios** |
| **Procesamiento de datos** | Ninguno | El sistema deberá *procesar x cantidad de datos* | vital | inmediatamente | en construcción | alta | Ninguno |
| **Respaldo** | Ninguno | El sistema deberá *respaldarse cada 24 horas* | vital | inmediatamente | en construcción | alta | Ninguno |
| **Modificación de la Base de datos** | Ninguno | El sistema deberá *permitir que los datos modificados en la base de datos sean actualizados para todos los usuarios que acceden en menos de 2 segundos* | vital | hay presión | en construcción | media | Ninguno |
| **Advertencias** | Ninguno | El sistema deberá *proporcionar mensajes de error o sonidos de alerta en caso de que sean necesarios* | importante | inmediatamente | en construcción | media | Ninguno |
| **Comunicación** | Ninguno | El sistema deberá *permitir comunicación externas entre los servidores de datos, la aplicación y el cliente del sistema deben estar cifradas en algún algoritmo para garantizar seguridad.* | vital | hay presión | en construcción | alta | Ninguno |

Fuente: Fuente propia, con ayuda de la herramienta REM

## 6.2 DISEÑO DE LA BASE DE DATOS

Según los requisitos de información planteados anteriormente en la especificación de requerimientos, nace el siguiente modelo entidad-relación, donde se muestran las entidades con sus respectivos atributos.

### 6.2.1 Modelo entidad relación

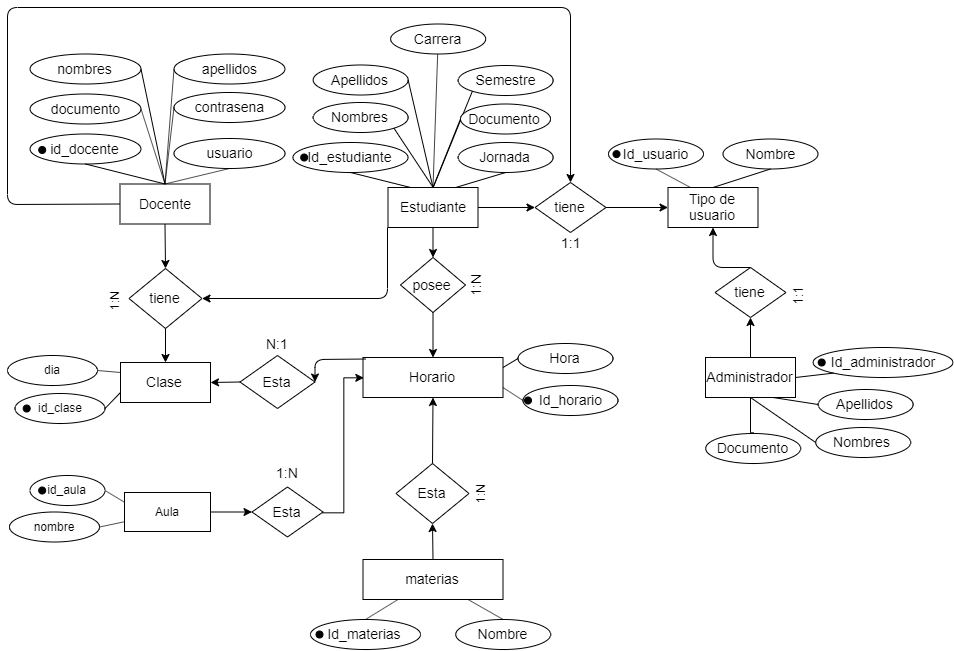


Ilustración 3 Modelo entidad relación,

Fuente: Fuente propia con ayuda de la herramienta Draw.io

En este diagrama entidad relación podemos observar 8 entidades las cuales son: Docente, Estudiante, Tipo de usuario, Clase, Horario, Aula, Materias y Administrador.

Este modelo no está normalizado por lo que aún no se encuentra optimizado al 100%, más adelante se enseñará con su forma normalizada.

### 6.2.2 Modelo físico

Para este modelo se utilizó el programa MySql Workbench el cual es una herramienta visual de diseño de bases de datos que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL.

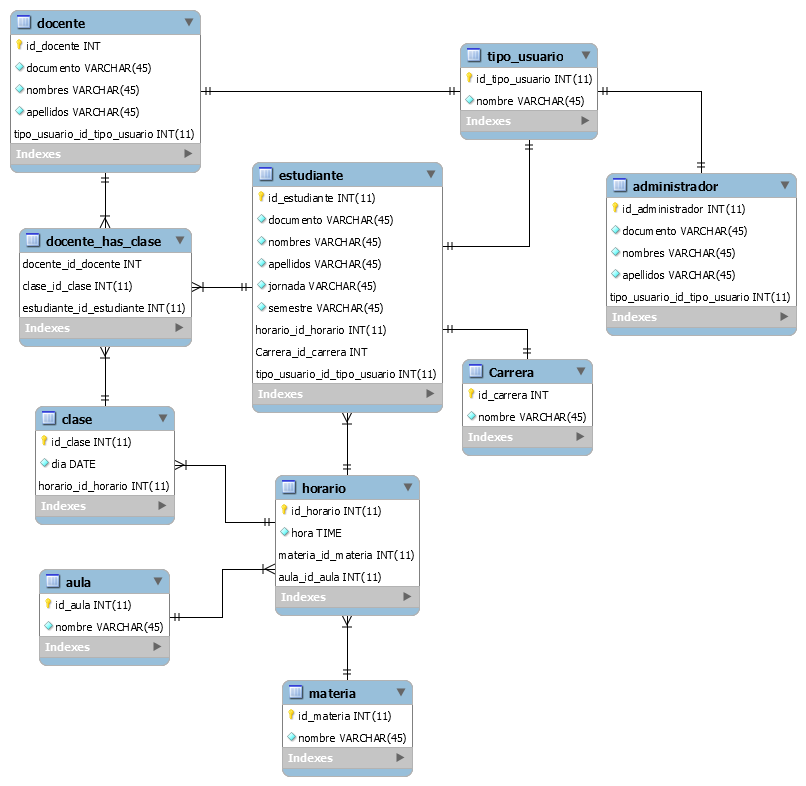


Ilustración 4 Modelo físico

Fuente: Fuente propia con ayuda de la herramienta MySQL Workbench

En esta etapa se normaliza el modelo entidad relación expuesto anteriormente, dando así como resultado nuevas entidades como se observa en la imagen de una forma más “organizada”.

### 6.2.3 Implementación a phpmyadmin

Cuando se finaliza con el modelo físico, es momento de comprobar si el modelo hecho está formado correctamente para su posterior subida a un servidor MySQL, para ello simplemente debemos dirigirnos a la pestaña database y, a continuación, forward engineer

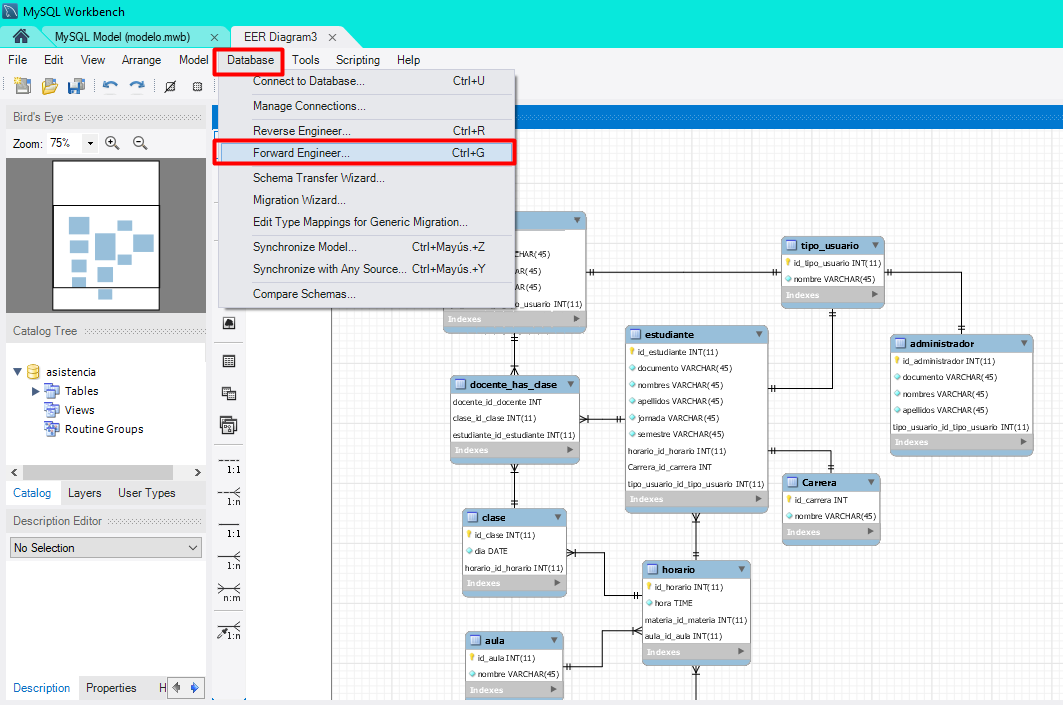


Ilustración 5 MySQL Workbench

Fuente: Fuente propia con ayuda de la herramienta Mysql Workbench

Al darle nos abrirá una pestaña como en la que se muestra en la siguiente imagen donde simplemente nos pide alguna conexión que debe crearse inicialmente en la pantalla principal del programa; a continuación daremos next hasta ubicarnos en la siguiente pestaña.

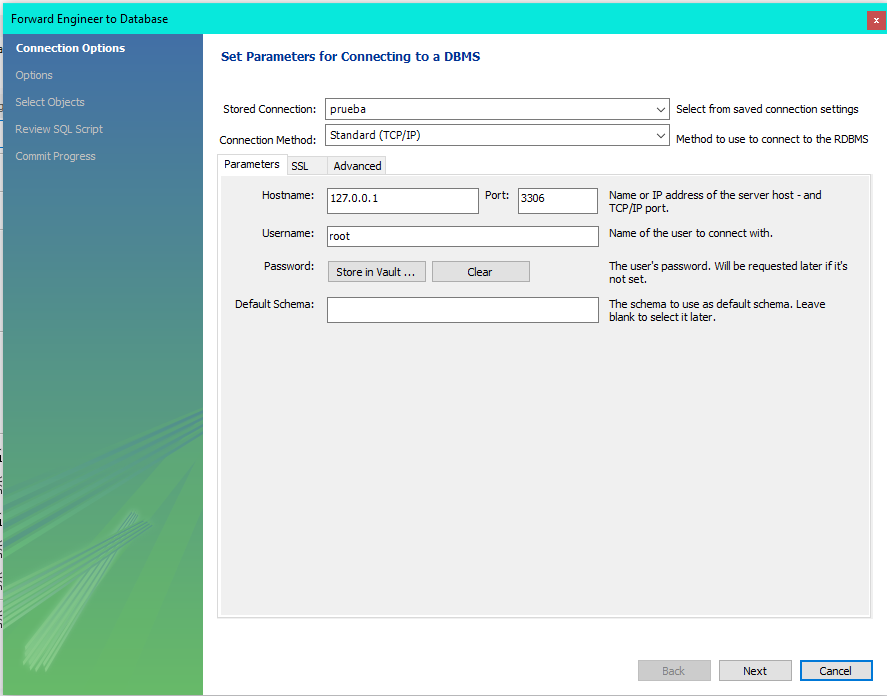


Ilustración 6 Forward engineer

Fuente: Fuente propia con ayuda de la herramienta MySQL Workbench

En esta imagen se aprecia el código de todo el esquema del modelo que se creó en un inicio, aquí es donde el programa se encarga de verificar que todo este en orden, si es así, simplemente saldrá un mensaje de que todo está bien y que el esquema ha sido exportado exitosamente

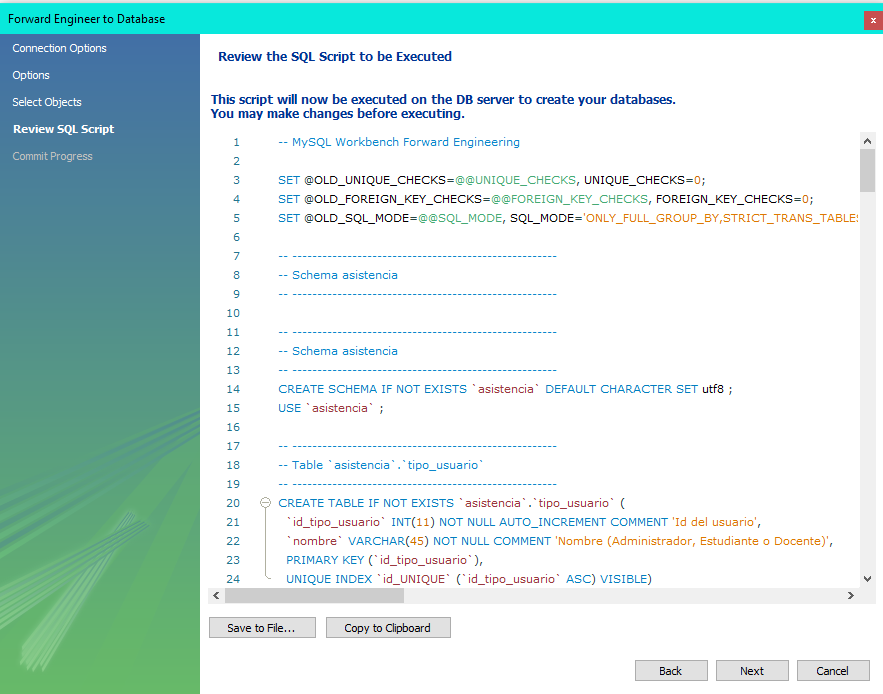


Ilustración 7 Forward engineer

Fuente: Fuente propia con ayuda de la herramienta MySQL Workbench

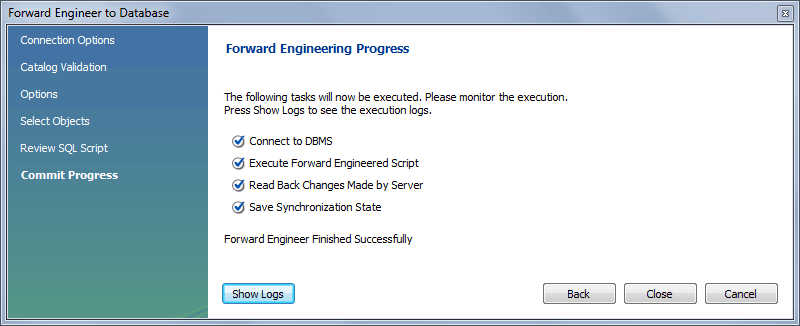


Ilustración 8 Mensaje

Fuente: Sitio oficial de MySQL Workbench

Después de haber echo lo mostrado anteriormente en las imágenes se pasa a comprobar en algún navegador la dirección localhost/phpmyadmin (para esto debemos tener funcionando un programa externo, en este caso se usó el xampp).

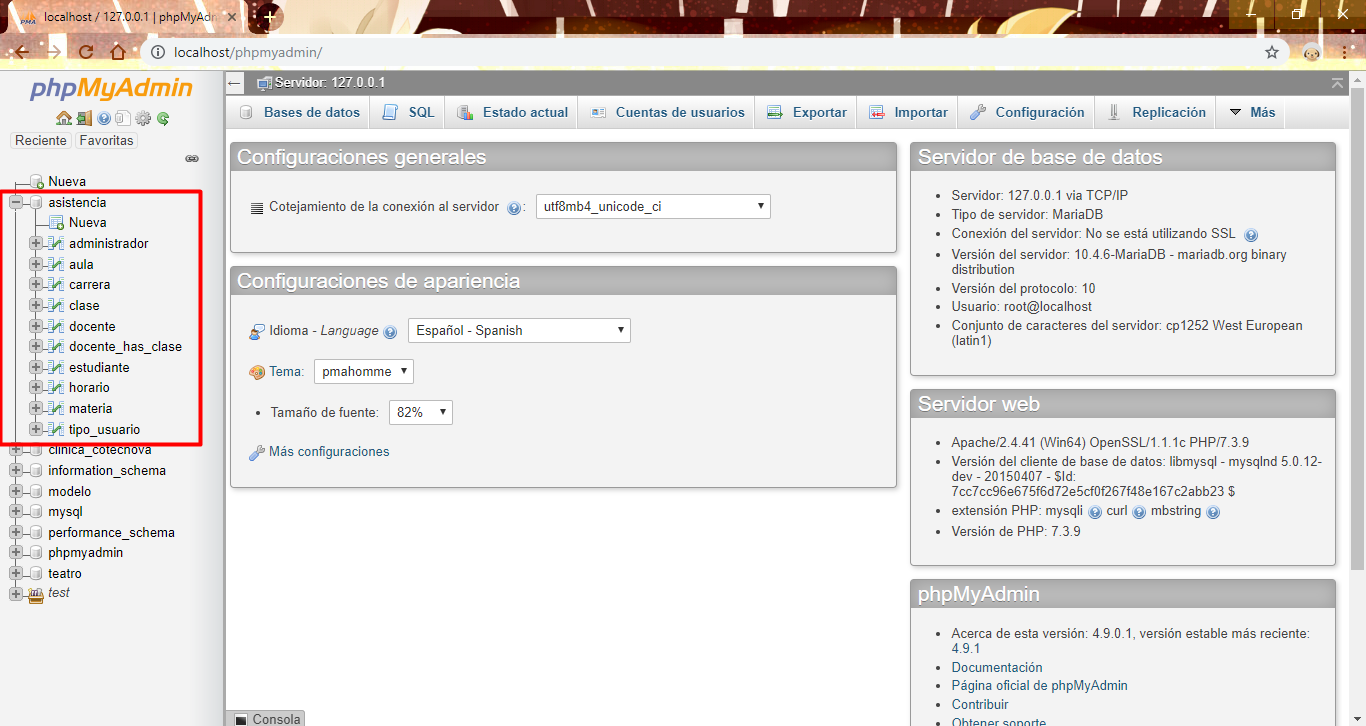


Ilustración 9 phpmyadmin

Fuente: Fuente propia

En la imagen puede observarse que en la parte izquierda del navegador se encuentra el esquema cargado que hicimos anteriormente en el programa, este ahora está transformado en una base de datos la cual se encuentra vacía en un principio, pero con la ayuda de phpmyadmin puede llenarse de una forma más ágil y rápida.

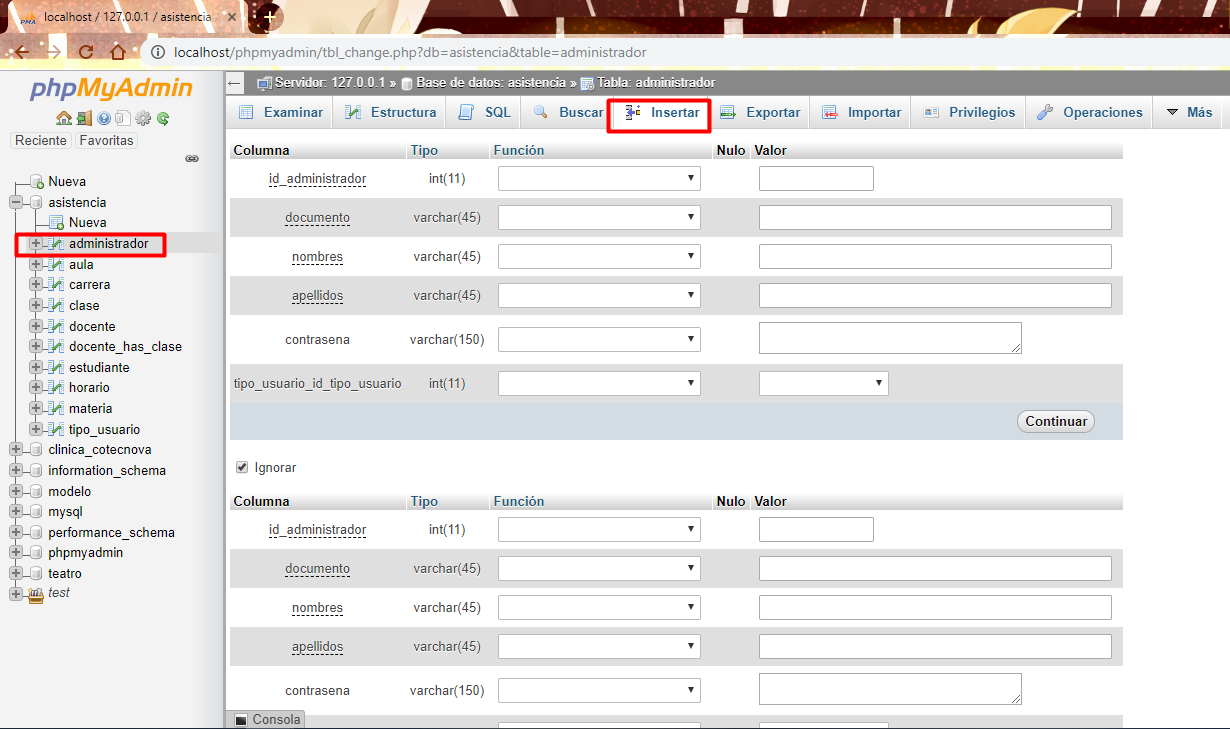


Ilustración 10 phpmyadmin

Fuente: Fuente propia

Luego de completar el proceso de llenar cada tabla con sus respectivos datos, esta lista para ser integrada y/o usada con una parte gráfica.

## 6.3 IMPLEMENTACIÓN DEL PROTOTIPO

El prototipo se desarrolló en el programa Sublime Text y Xampp, a continuación se mostrara como se desarrolló:

Primero que todo se importara la Base de datos con ayuda del programa Xampp cabe destacar que al utilizar la herramienta Xampp el prototipo quedara de forma local, para empezar a importar la base de datos tendremos que crear una nueva dentro del programa como se muestra a continuación.



Ilustración 10 Creación de la base de datos en la herramienta XAMPP

Fuente: Fuente propia con ayuda de la herramienta phpMyAdmin

Una vez que ya está creada la base de datos en XAMPP procedemos a dar en la opción importar ubicada en la parte superior.



Ilustración 11 Botón de importación de XAMPP

Fuente: Fuente propia con ayuda de la herramienta phpMyAdmin

Después de dar clic en el botón de importar, seleccionaremos la opción seleccionar archivos, esta nos abrirá una ventana emergente en la cual buscaremos el archivo SQL de nuestra base de datos ya esportada y daremos clic en la opción abrir.

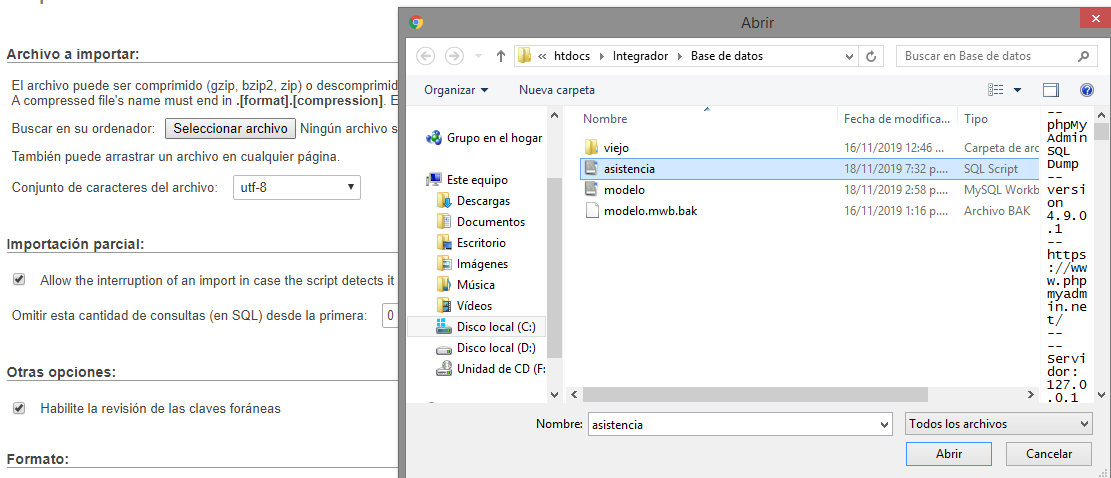


Ilustración 12 Importación de la BD

Fuente: Fuente propia con ayuda de la herramienta phpMyAdmin

Una vez que esta se haya importado, automáticamente aparecerán las diferentes tablas de esta base de datos, las cuales usaremos más adelante en la herramienta Sublime Text.

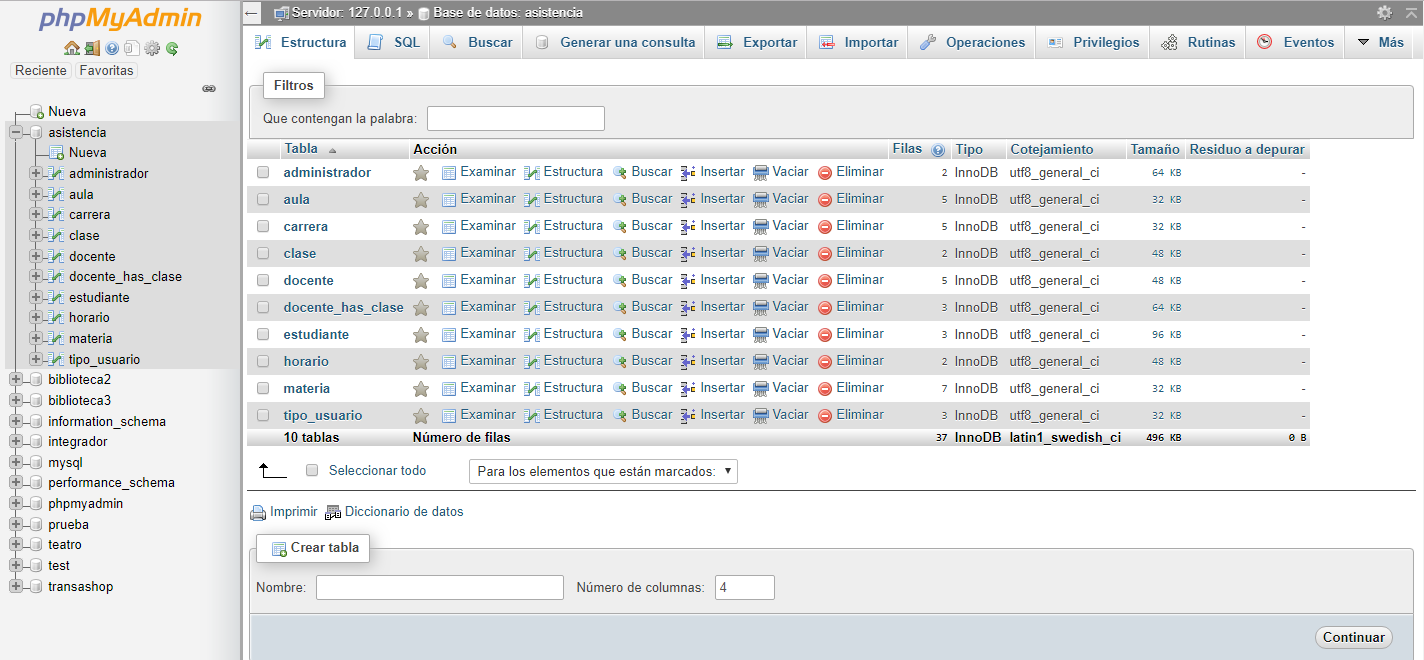


Ilustración 13 Vista global de las tablas almacenadas en la base de datos

Fuente: Fuente propia con ayuda de la herramienta phpMyAdmin

Una vez que se hayan hecho los pasos anteriores procedemos a abrir nuestro proyecto en la herramienta Sublime Text, para abrirla seleccionaremos en la esquina superior izquierda la opción file, luego la opción open folder, se abrirá una ventana emergente en la cual buscaremos la ubicación del proyecto, ya localizada daremos en la opción abrir tal como se muestra a continuación.

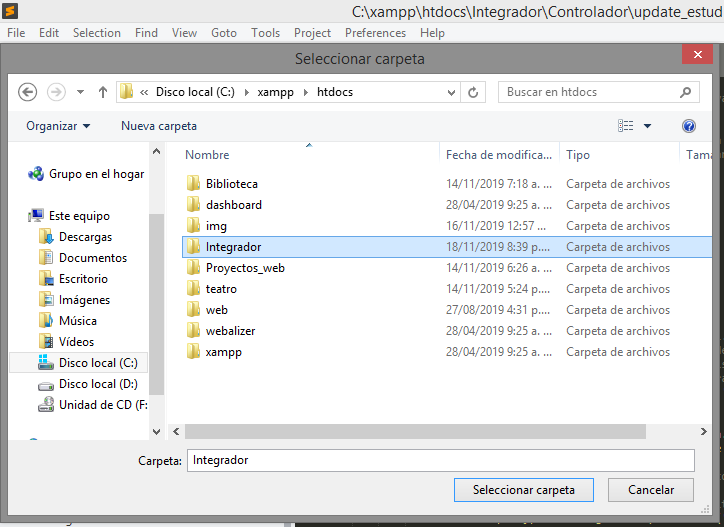


Ilustración 14 Apertura del prototipo en la herramienta Sublime Text

Fuente: Fuente propia

Una vez hecho lo anterior veremos que al lado izquierdo del programa saldrán las diferentes carpetas del proyecto.

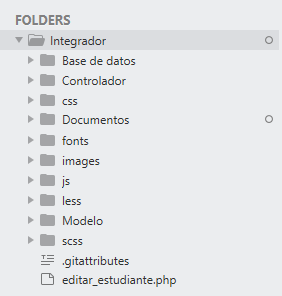


Ilustración 15 Carpetas desde Sublime Text

Fuente: Fuente propia con ayuda de la herramienta sublime text

Una vez hecho los pasos anteriores procedemos entrar a la carpeta Modelos dentro de sublime, allí encontraremos un archivo llamado MySQL.php el cual abriremos y dentro de este haremos la conexión a nuestra base de datos de forma local como se muestra a continuación.

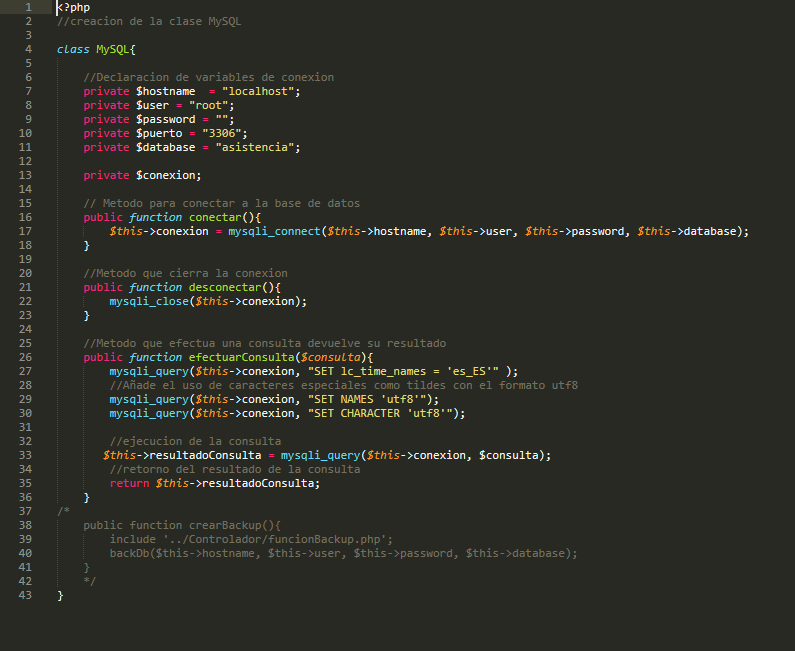


Ilustración 16 Conexión a la base de datos en forma local

Fuente: Fuente propia con ayuda de la herramienta sublime text

Una vez hecho la conexión procederemos a crear la interfaz gráfica del login, el cual dará inicio al prototipo, este se creara a partir de etiquetas HTML, una vez que esté listo iremos al navegador y escribiéremos en la URL lo siguiente: localhost/nombre\_carpeta/login.php.

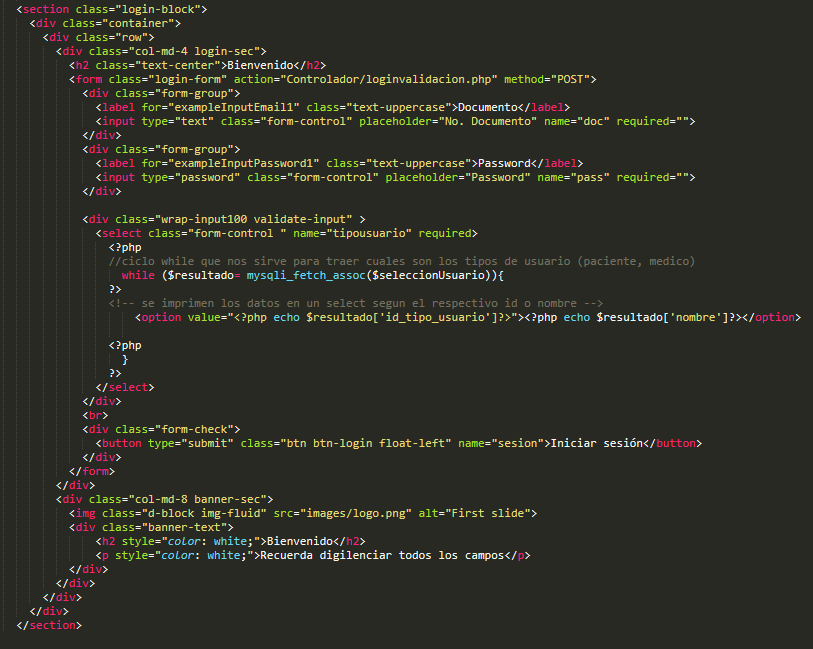


Ilustración 17 etiquetas HTML del login

Fuente: Fuente propia con ayuda de la herramienta sublime text



Ilustración 18 Vista grafica desde el navegador

Fuente: Fuente Propia

El login es controlado por medio de un controlador alojado en la carpeta Controladores, dentro de esta carpeta se encuentra el archivo loginValidacion.php el cual está conformado por las diferentes conexiones y consultas de la base de datos, el resultado de estas consultas se validara con la información presentada en los inputs del formulario, si son iguales se iniciara la sesión del usuario, el usuarios se selecciona en el select del formulario, actualmente son tres tipos de usuarios (Administrador, Docente, estudiante).

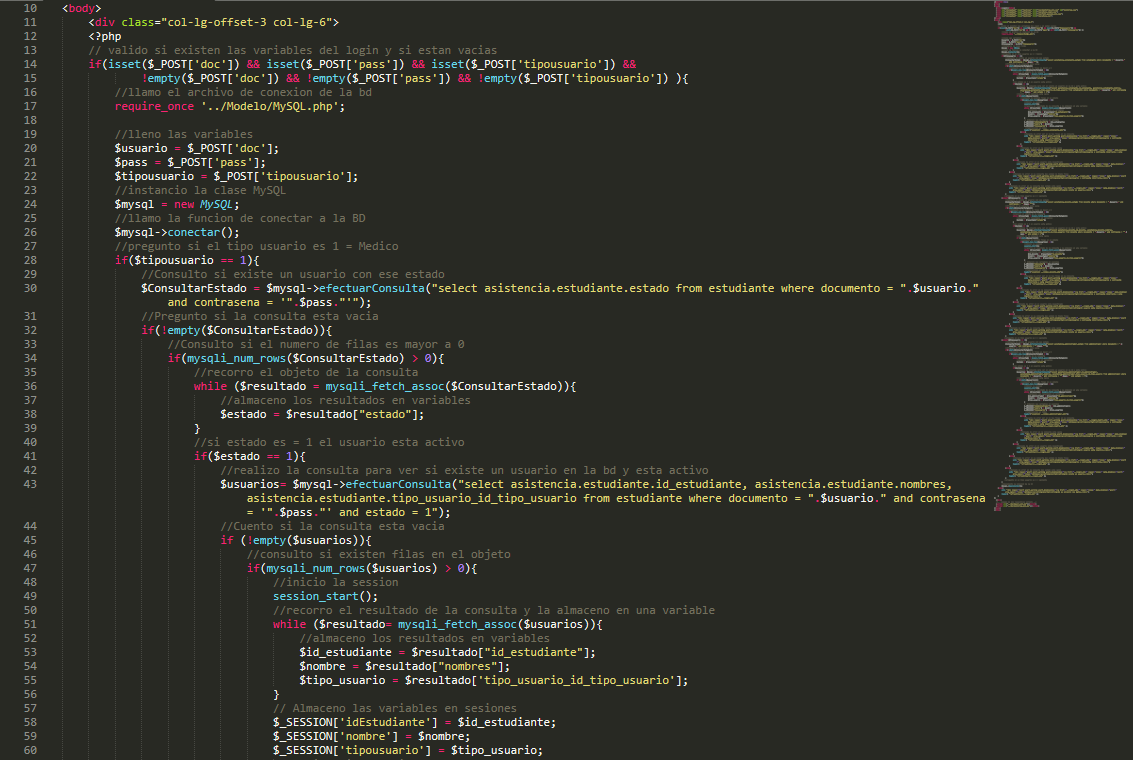


Ilustración 19 Validación del login.php

Fuente: Fuente propia con ayuda de la herramienta sublime text

Una vez que el usuario haya iniciado sesión se enviara a su respectivo index según el tipo de usuario, en el caso de administrador este será re direccionado al index\_administrador.php creado de la misma manera que el login.



Ilustración 20 Index administrador

Fuente: Fuente propia

Una vez que el administrador está dentro del index tendrá varias opciones a ejecutar en la parte izquierda del navegador, las cuales son registrar tanto estudiantes, docentes, clases, horarios, materias, etc.

Todas las opciones están creadas por medio de las etiquetas HTML que proporciona Sublime Text, a continuación mostraremos como fue creada la sección del estudiante.

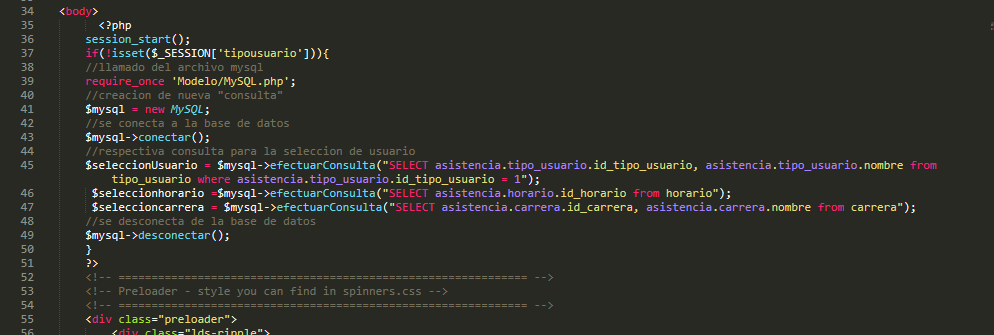


Ilustración 21 Consultas generadas para mostrar información en los formulario

Fuente: Fuente propia

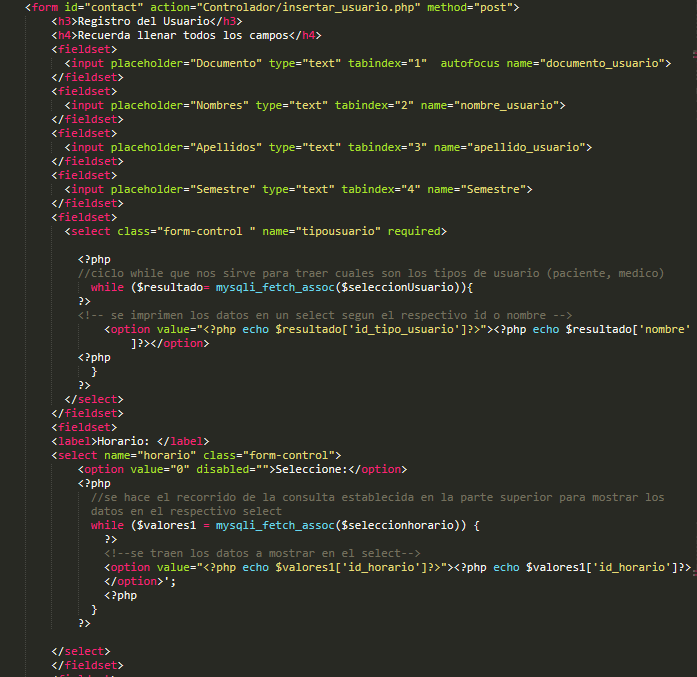


Ilustración 22 Creación del formulario de registro del estudiante

Fuente: Fuente Propia

Una vez que tengamos listo las etiquetas procedemos a ejecutarlo en el navegador de la misma forma que se ejecutó el login.

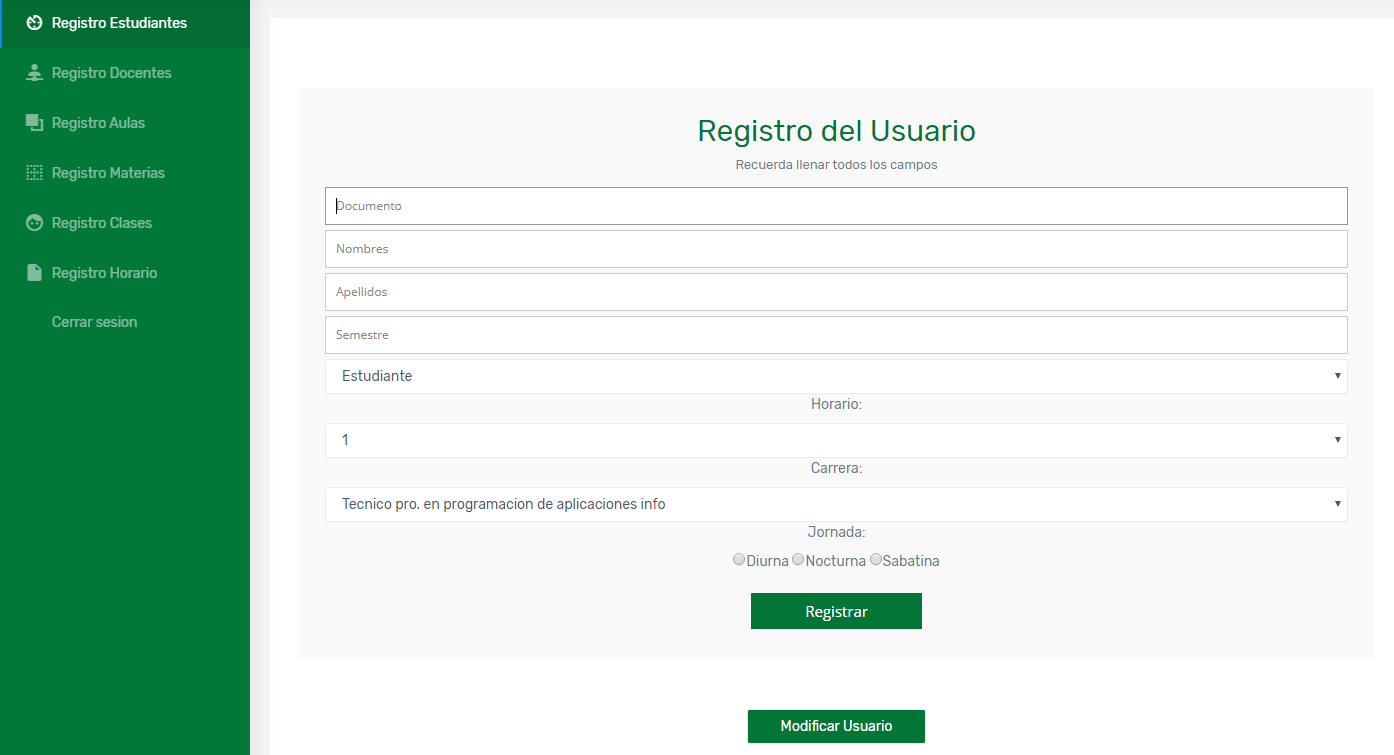


Ilustración 23 Vista grafica del formulario de registro del estudiante

Fuente: Fuente propia

En la imagen anterior el administrador empieza a llenar los diferentes campos del formulario, una vez diligenciados el administrador procede a dar en el botón de registrar, este enviara los datos de forma automática a la base de datos.

Dentro de este formulario el usuario puede modificar a los estudiantes registrados en la base de datos, para esto procede a dar clic en el botón modificar usuario, este lo enviara al siguiente formulario creado por medio de las etiquetas HTML.

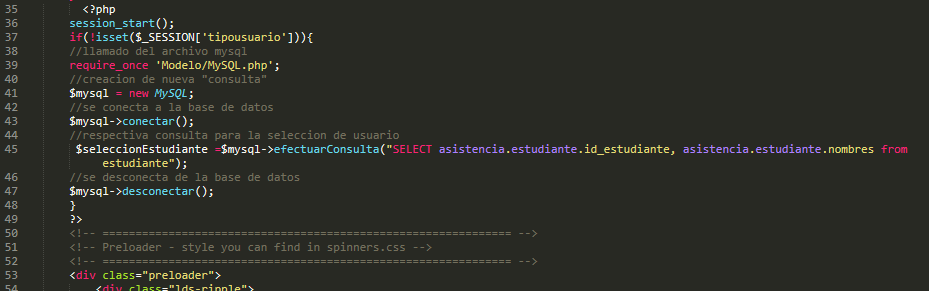


Ilustración 24 Conexión del formulario con la base de datos con su respectiva consulta

Fuente: Fuente propia



Ilustración 25 Creación del formulario

Fuente: Fuente propia

En esta imagen se puede observar que la acción que está realizando el formulario es enviar por medio de una select la información del estudiante, en este select se puede ver a los diferentes estudiantes que se encuentran en el sistema.



Ilustración 26 Selección del estudiante

Fuente: Fuente propia

Una vez que el administrador ha seleccionado al estudiante, los datos se enviaran a la siguiente vista o ventana del prototipo.

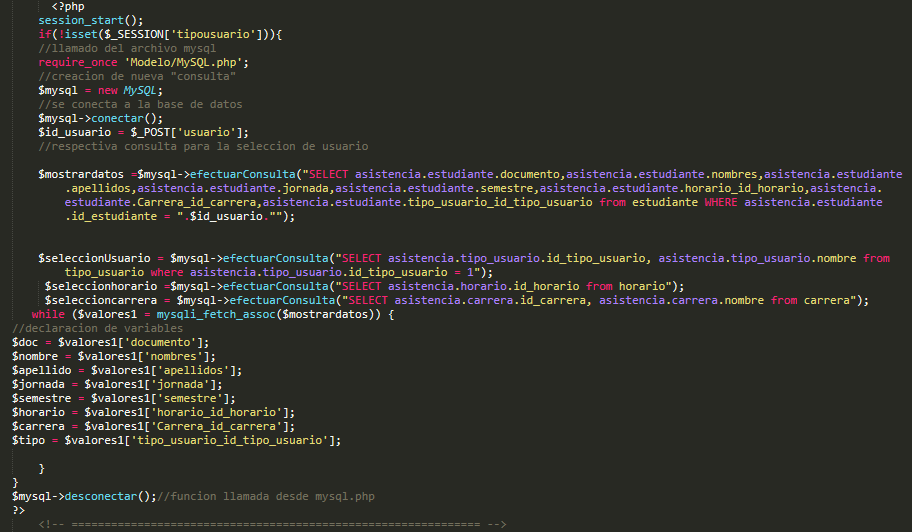


Ilustración 27 Recolección de datos del estudiante por medio de la consulta generada del select

Fuente: Fuente propia con ayuda de la herramienta sublime text

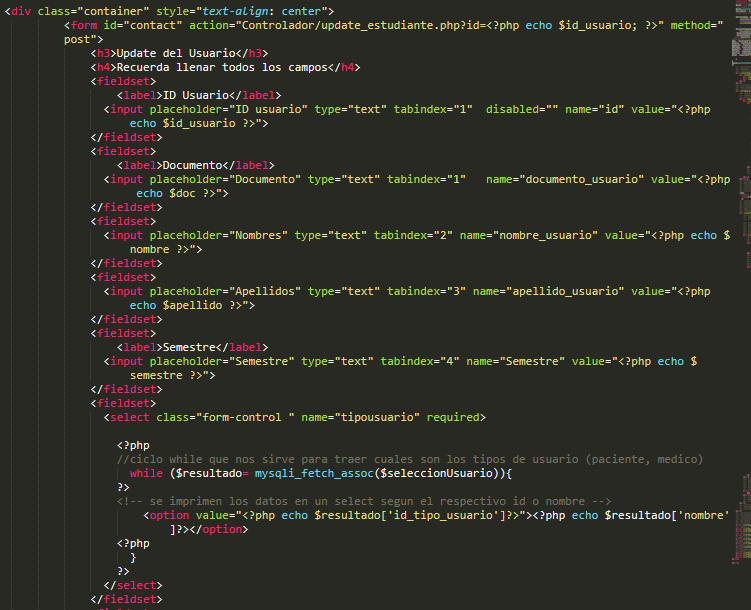


Ilustración 28 recolección de la información en el select

Fuente: Fuente propia con ayuda de la herramienta sublime text

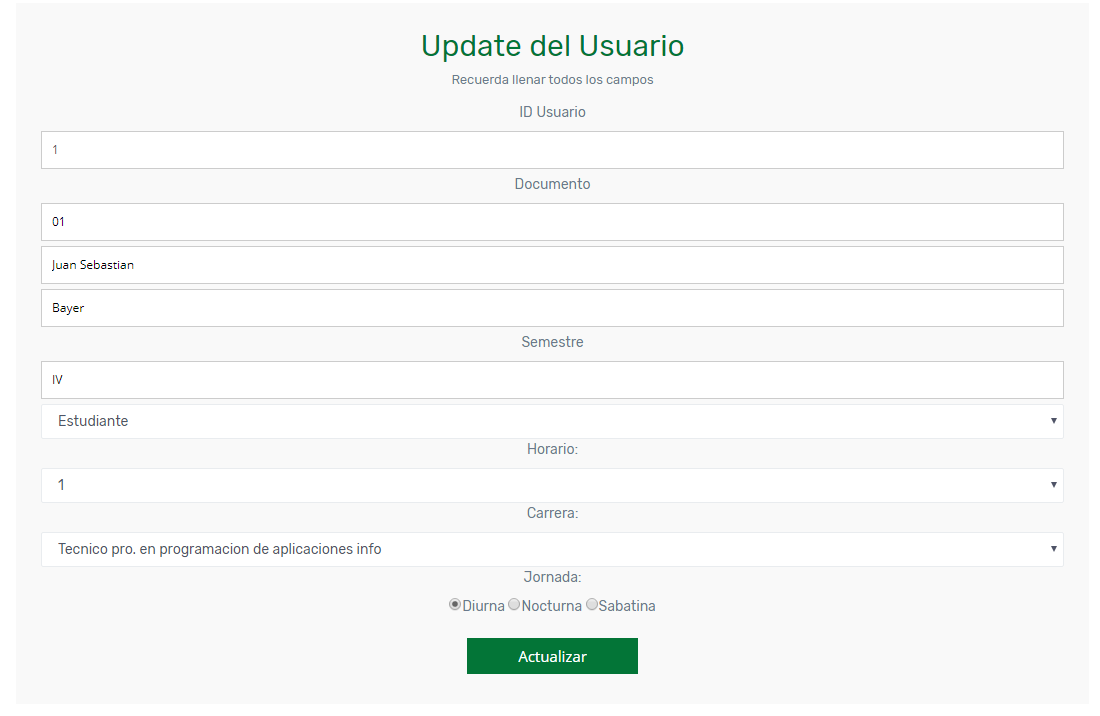


Ilustración 29 Vista grafica de la recolección de datos

Fuente: Fuente Propia

El administrador al dar clic en el botón de actualizar enviara los datos al controlador del formulario alojado en la carpeta Controladores/ update\_usuario.php, aquí se encargara de validar los datos modificados y agregarlos a la base de datos.



Ilustración 30 Validación de los campos modificados del formulario

Fuente: Fuente propia

Una vez que la validación ha finalizado re direccionara al administrador a las diferentes páginas, es decir, si la validación de datos es exitosa lo enviara al index\_adminstrador.php donde podrá hacer otra operación diferente, de lo contrario lo enviara de nuevo al menú de modificación.

Cabe aclarar que todos los procesos mencionados anteriormente se repiten para los diferentes procesos que ejecuta el administrador.

# 7. CONCLUSIONES

• Actualmente existen muchas instituciones en las cuales aún se maneja la asistencia de forma tradicional, se espera que este sistema pueda cambiarse a medida del tiempo trayendo así más fluidez en el mismo.

• Con la implementación de este prototipo se espera llegar a una gran mejoría en cuanto a la problemática planteada, la facilitación de la tecnología y la forma en la que puede manejarse podría sentar bases para futuros proyectos.

• El desarrollo de las bases de datos agiliza de una forma mucho más rápida y eficiente la forma en la que viene llevándose cualquier sistema manual y/o tradicional, siendo así, un gran avance para las comunidades académicas.

# 8. CRONOGRAMA

Tabla 25 Cronograma

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Actividad | Mes | 1 | | | | 2 | | | | 3 | | | | 4 | | | | 5 | | | | 6 | | | |
| Semana | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | Detección de requerimientos del sistema | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Búsqueda y análisis de la información | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Elaboración de los casos de uso | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | Elaboración de los requisitos funcionales | 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Elaboración de los requisitos no funcionales | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Modelado de la base de datos | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Diseño del sistema a implementar | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Programación | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Evaluación y prueba del prototipo | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Prototipo final | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | Implementación del prototipo | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Fuente: Fuente propia, con ayuda de la herramienta REM

# Bibliografía

Chen, P.-S. (2002). *The Entity Relationship Model – Toward a Unified View of Data.* Obtenido de Extra Materials: http://extras.springer.com/2002/978-3-642-63970-8/DVD3/rom/pdf/Chen\_hist.pdf

Delgado, E. (s.f). *Proceso de Desarrollo de la Base de Datos*. Obtenido de Universidad Autónoma del Estado de Hidalgo: http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro14/15\_proceso\_de\_desarrollo\_de\_la\_base\_de\_datos.html

Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla. (25 de Noviembre de 2004). *Herramienta REM*. Obtenido de Departamento de Lenguajes y Sistemas Informáticos - Universidad de Sevilla: http://www.lsi.us.es/descargas/descarga\_programas.php?id=3

DuocUC. (s.f). *Definición y propósito de la Investigación Aplicada*. Obtenido de Portal Duoc: http://www.duoc.cl/biblioteca/crai/definicion-y-proposito-de-la-investigacion-aplicada

Grupo Carricay. (25 de Junio de 2018). *¿Cómo escribir un buen documento de especificación de requisitos de software?* Obtenido de Medium: https://medium.com/grupo-carricay/c%C3%B3mo-escribir-un-buen-documento-de-especificaci%C3%B3n-de-requisitos-de-software-fd8bb3b5a39a

Grupo Carricay. (20 de Abril de 2018). *Requerimientos Funcionales y No Funcionales, ejemplos y tips*. Obtenido de Medium: https://medium.com/@requeridosblog/requerimientos-funcionales-y-no-funcionales-ejemplos-y-tips-aa31cb59b22a

Hernandez, E. (30 de Enero de 2012). *España: Tecnología NFC para controlar la asistencia en la Universidad de Salamanca*. Obtenido de FayerWayer: https://www.fayerwayer.com/2012/01/espana-tecnologia-nfc-para-controlar-la-asistencia-en-la-universidad-de-salamanca/

Liberato, G. (2013). *TECNICAS DE RECOLECCIÓN DE DATOS*. Recuperado el 04 de 09 de 2019, de TECNICAS DE RECOLECCIÓN DE DATOS: https://gabriellebet.files.wordpress.com/2013/01/tecnicas-de-recoleccic3b3n4.pdf#targetText=La%20recolecci%C3%B3n%20de%20datos%20se,y%20el%20diccionario%20de%20datos.

Martinez, M. A. (s.f). *Conocimiento y Bases de Datos: Una propuesta de integrador inteligente.* Obtenido de Tesis Doctorals en Xarxa: https://www.tesisenred.net/bitstream/handle/10803/31767/3de3.MAMcap5\_conclusiones\_bibliograf%C3%ADa.pdf?sequence=4&isAllowed=y

Sanchez, J. (s.f). *Modelo Relacional*. Obtenido de JorgeSanchez.net: https://jorgesanchez.net/manuales/gbd/modelo-relacional.html

Sanjahingu. (28 de 07 de 2014). *Métodos, Tipos y Enfoques de Investigación*. Obtenido de Investigación: http://sanjahingu.blogspot.com/2014/01/metodos-tipos-y-enfoques-de.html

Universidad de Antioquia. (s.f). *Tarjeta Integrada personal*. Obtenido de Universidad de Antioquia: http://www.udea.edu.co/wps/portal/udea/web/inicio/institucional/tarjeta-integrada-personal

Universidad ICESI. (15 de Octubre de 2010). *SRS / ERS Especificación de requerimientos de software*. Obtenido de Universidad ICESI: http://www.icesi.edu.co/departamentos/tecnologias\_informacion\_comunicaciones/proyectos/lisa/home/analisis/srs/srs

Xunta de Galicia. (2 de Julio de 2015). *Base de Datos.Database*. Obtenido de Centro Autonómico de Formación e Innovación - Aula virtual: https://www.edu.xunta.gal/centros/cafi/aulavirtual2/mod/page/view.php?id=25004