

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA PROTOTIPO PARA LA ADMINISTRACIÓN ESTUDIANTIL DE UNA UNIDAD EDUCATIVA

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TECNOLOGÍAS DE LA INFORMACIÓN**

ÁNGEL DAVID CUSHICONDOR COLLAGUAZO

angel.cushicondor@epn.edu.ec

DIRECTOR: ING. FRANKLIN LEONEL SÁNCHEZ CATOTA, MSC.

franklin.sanchez@epn.edu.ec

DMQ, octubre 2022

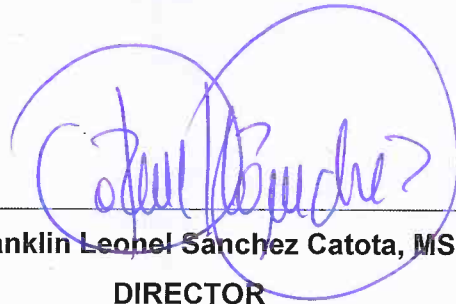
CERTIFICACIONES

Yo, Angel David Cushicondor Collaguazo declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Angel David Cushicondor Collaguazo

Certifico que el presente trabajo de integración curricular fue desarrollado por Angel David Cushicondor Collaguazo, bajo mi supervisión.



Ing. Franklin Leonel Sanchez Catota, MSc.

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Angel David Cushicondor Collaguazo

Franklin Leonel Sánchez Catota

DEDICATORIA

Dedico el presente trabajo de integración curricular a todas las personas que me han apoyado durante mi etapa universitaria, amigos, compañeros en especial a mi familia, que con sus consejos me han ayudado a seguir adelante y no rendirme para conseguir mis objetivos.

AGRADECIMIENTO

Agradezco a mi familia por el apoyo brindado, no solo en mi etapa universitarias sino durante toda mi vida, por sus palabras, sus consejos y su cariño hacia mi persona.

Agradezco también a toda la comunidad politécnica, Institución, autoridades, profesores y compañeros ya que me han ayudado a crecer como persona y en mi formación profesional.

Agradezco a mi director MSc. Franklin Sánchez, quien estuvo pendiente del desarrollo del presente trabajo respondiendo dudas con amabilidad y profesionalismo.

ÍNDICE DE CONTENIDO

| | |
|--|------|
| CERTIFICACIONES..... | I |
| DECLARACIÓN DE AUTORÍA..... | II |
| DEDICATORIA..... | III |
| AGRADECIMIENTO..... | IV |
| ÍNDICE DE CONTENIDO..... | V |
| ÍNDICE DE FIGURAS | VIII |
| ÍNDICE DE TABLAS | X |
| RESUMEN | XI |
| ABSTRACT | XII |
| 1 INTRODUCCIÓN | 1 |
| 1.1 Objetivo general | 2 |
| 1.2 Objetivos específicos | 2 |
| 1.3 Alcance | 2 |
| 1.4 Marco teórico | 3 |
| 1.4.1 Aplicaciones web..... | 3 |
| 1.4.2 Arquitectura cliente-servidor | 4 |
| 1.4.3 Servicio web | 5 |
| 1.4.3.1 Protocolos y estándares de servicios web..... | 5 |
| 1.4.3.2 RestFul..... | 6 |
| 1.4.4 Metodología Kanban..... | 6 |
| 1.4.4.1 Definición..... | 6 |
| 1.4.4.2 Implementación de la metodología Kanban | 7 |
| 1.4.4.2.1 Empezar con un tablero en blanco..... | 7 |
| 1.4.4.2.2 Creación de columnas para representar el proyecto..... | 7 |
| 1.4.4.2.3 Añadir tareas para representar el proyecto | 8 |
| 1.4.4.2.4 Hacer que el trabajo avance por etapas..... | 8 |
| 1.4.5 Herramientas y tecnologías | 8 |
| 1.4.5.1 HTML..... | 8 |
| 1.4.5.1.1 Definición..... | 8 |
| 1.4.5.1.2 Utilidad | 8 |
| 1.4.5.2 JavaScript..... | 9 |
| 1.4.5.2.1 Definición..... | 9 |

| | |
|--|----|
| 1.4.5.2.2 Características | 9 |
| 1.4.5.3 JQuery | 9 |
| 1.4.5.3.1 Definición..... | 9 |
| 1.4.5.3.2 Características | 9 |
| 1.4.5.4 CSS | 10 |
| 1.4.5.4.1 Definición..... | 10 |
| 1.4.5.4.2 Características | 10 |
| 1.4.5.5 Bootstrap..... | 10 |
| 1.4.5.6 PHP | 11 |
| 1.4.5.6.1 Definición..... | 11 |
| 1.4.5.6.2 Características | 11 |
| 1.4.5.7 MySQL | 11 |
| 1.4.5.7.1 Definición..... | 11 |
| 1.4.5.7.2 Características | 11 |
| 1.4.5.8 WampServer | 12 |
| 1.4.5.8.1 Definición..... | 12 |
| 1.4.5.8.2 Características | 12 |
| 1.4.5.9 Apache | 13 |
| 1.4.5.9.1 Definición..... | 13 |
| 1.4.5.9.2 Ventajas | 13 |
| 1.4.5.9.3 Funcionamiento..... | 13 |
| 1.4.5.10 phpMyAdmin | 13 |
| 1.4.5.10.1 Definición..... | 13 |
| 1.4.5.10.2 Características..... | 14 |
| 1.4.5.11 SublimeText | 14 |
| 1.4.5.11.1 Definición..... | 14 |
| 1.4.5.11.2 Características..... | 15 |
| 2 METODOLOGÍA | 16 |
| 2.1 DISEÑO | 16 |
| 2.1.1 Tablero Kanban | 16 |
| 2.1.2 Recolección de requerimientos..... | 17 |
| 2.1.2.1 Historias de usuario..... | 17 |
| 2.1.3 Requerimientos funcionales y no funcionales..... | 19 |
| 2.1.3.1 Requerimientos funcionales | 19 |

| | | |
|---------|--|----|
| 2.1.3.2 | Requerimientos no funcionales | 20 |
| 2.1.4 | Diagrama de casos de uso | 20 |
| 2.1.5 | Diagrama Relacional | 23 |
| 2.1.6 | Diseño de Interfaces..... | 25 |
| 2.1.7 | Diagrama de clases | 27 |
| 2.2 | IMPLEMENTACIÓN | 28 |
| 2.2.1 | Tablero Kanban actualizado | 28 |
| 2.2.2 | Instalación de las herramientas tecnológicas | 29 |
| 2.2.3 | Implementación del modelo relacional..... | 29 |
| 2.2.4 | Implementación de las interfaces web..... | 31 |
| 2.2.5 | Implementación de la lógica de negocio..... | 36 |
| 3 | RESULTADOS, CONCLUSIONES Y RECOMENDACIONES..... | 42 |
| 3.1 | Tablero Kanban actualizado..... | 42 |
| 3.2 | Resultados | 42 |
| 3.3 | Correcciones Realizadas | 54 |
| 3.4 | Conclusiones..... | 54 |
| 3.5 | Recomendaciones..... | 55 |
| 4 | REFERENCIAS BIBLIOGRÁFICAS | 56 |
| 5 | ANEXOS..... | 59 |

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1.1 Arquitectura Cliente-Servidor | 4 |
| Figura 1.2 Servicios web | 5 |
| Figura 1.3 Tablero Kanban..... | 8 |
| Figura 1.4 Interfaz web de phpMyAdmin..... | 14 |
| Figura 1.5 SublimeText IDE | 14 |
| Figura 2.1 Diagrama de Casos de Uso para Administrador | 21 |
| Figura 2.2 Diagrama de Casos de Uso para Profesor..... | 22 |
| Figura 2.3 Diagrama de Casos de Uso para Tutor..... | 22 |
| Figura 2.4 Diagrama de Casos de Uso para Estudiante | 23 |
| Figura 2.5 Diagrama de Casos de Uso para Representante..... | 23 |
| Figura 2.6 Diagrama relacional de la base de datos | 24 |
| Figura 2.7 Diseño de interfaz: Acceso al sistema..... | 25 |
| Figura 2.8 Diseño de interfaz: Administración de profesores (Administrador)..... | 26 |
| Figura 2.9 Diseño de interfaz: formulario para crear el usuario Profesor | 26 |
| Figura 2.10 Diagrama de clases del sistema prototipo..... | 27 |
| Figura 2.11 Código para crear la base de datos | 29 |
| Figura 2.12 Código para crear la tabla administradores..... | 30 |
| Figura 2.13 Código para el registro del usuario Administrador..... | 30 |
| Figura 2.14 Código para crear la tabla profesores | 30 |
| Figura 2.15 Código para crear la tabla notas | 31 |
| Figura 2.16 Código HTML base | 32 |
| Figura 2.17 Login: acceso al sistema | 32 |
| Figura 2.18 Interfaz web para la administración de profesores..... | 33 |
| Figura 2.19 Código HTML de la ventana modal para crear profesores..... | 33 |
| Figura 2.20 Ventana modal: crear alumno | 34 |
| Figura 2.21 Interfaz web: asignar profesores a asignaturas..... | 34 |
| Figura 2.22 Interfaz web para la administración de observaciones académicas particulares..... | 35 |
| Figura 2.23 Interfaz web: Consulta de notas del estudiante..... | 35 |
| Figura 2.24 Código en el cliente: Acceso al sistema | 36 |
| Figura 2.25 Código en el Servidor: archivo index.php..... | 37 |
| Figura 2.26 Código en el servidor: Validar usuario..... | 38 |

| | |
|--|----|
| Figura 2.27 Código en el Servidor: Conexión a la base de datos..... | 38 |
| Figura 2.28 Código en el cliente. Insertar profesor..... | 39 |
| Figura 2.29 Código en el servidor: crear usuario profesor | 40 |
| Figura 2.30 Código Cabecera del certificado de promoción..... | 40 |
| Figura 2.31 Código para generar el certificado de promoción..... | 41 |
| Figura 3.1 Login | 43 |
| Figura 3.2 Alerta de Bienvenida al sistema | 43 |
| Figura 3.3 Inicio (Administrador) | 43 |
| Figura 3.4 Crear Profesor..... | 44 |
| Figura 3.5 Mensaje de crear profesor | 44 |
| Figura 3.6 Editar Profesor | 45 |
| Figura 3.7 Mensaje de actualización del profesor | 45 |
| Figura 3.8 Lista de profesores..... | 46 |
| Figura 3.9 Mensaje de eliminación del profesor | 46 |
| Figura 3.10 Lista de profesores después de eliminar un profesor..... | 46 |
| Figura 3.11 Búsqueda de profesores | 46 |
| Figura 3.12 Crear fechas de habilitación..... | 47 |
| Figura 3.13 Mensaje de creación de fechas de habilitación..... | 47 |
| Figura 3.14 Ingresar notas | 48 |
| Figura 3.15 Formulario Ingresar nota | 48 |
| Figura 3.16 Mensaje de registro de notas | 48 |
| Figura 3.17 Notas del estudiante..... | 49 |
| Figura 3.18 Crear observación académica..... | 49 |
| Figura 3.19 Mensaje de creación de la observación académica | 50 |
| Figura 3.20 Editar observación académica general | 50 |
| Figura 3.21 Mensaje de actualización de la observación general | 50 |
| Figura 3.22 Lista de observaciones académicas generales..... | 51 |
| Figura 3.23 Mensaje de eliminación de observaciones generales | 51 |
| Figura 3.24 Lista de observaciones académicas generales después de la eliminación | 51 |
| Figura 3.25 Búsqueda de observaciones académicas generales | 52 |
| Figura 3.26 Generar certificados académicos | 52 |
| Figura 3.27 Cuadro de calificaciones del primer quimestre..... | 53 |
| Figura 3.28 Certificado de promoción | 53 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 2.1 Tablero Kanban para la fase de Diseño | 16 |
| Tabla 2.2 Historia de Usuario | 17 |
| Tabla 2.3 Historia de Usuario HU001 | 18 |
| Tabla 2.4 Historia de Usuario HU027 | 18 |
| Tabla 2.5 Historia de Usuario HU030 | 19 |
| Tabla 2.6 Historia de Usuario HU036..... | 19 |
| Tabla 2.7 Requerimientos funcionales | 19 |
| Tabla 2.8 Requerimientos no funcionales | 20 |
| Tabla 2.9 Tablero Kanban para la fase de Implementación | 28 |
| Tabla 3.1 Tablero Kanban para pruebas de funcionamiento y resultados | 42 |

RESUMEN

El objetivo del presente trabajo de integración curricular fue el desarrollo de un sistema prototipo (aplicación web) para la gestión de la información académica de una unidad educativa, el cual permitirá gestionar los datos de los diferentes usuarios que integran esta institución (estudiantes, profesores y representantes), las notas de los estudiantes y los certificados académicos. También, permitirá a los representantes conocer el rendimiento y comportamiento de sus representados a través de las observaciones académicas.

En el capítulo I se muestran todos los conceptos básicos y necesarios, así como las herramientas tecnológicas utilizadas para la realización del presente trabajo de integración curricular.

En el capítulo II se tiene el diseño y la implementación del sistema prototipo. En la parte del diseño se encuentra la metodología Kanban que fue utilizada para el desarrollo del sistema prototipo, así como: recolección de requerimientos, diseño de la base de datos, diagrama de casos de uso, bosquejo de las interfaces y diagrama de clases. En la parte de la implementación se muestran las herramientas tecnológicas utilizadas para el desarrollo del sistema prototipo, así se tiene: PHP (Hypertext Preprocessor) como lenguaje de programación, como gestor de base de datos se utilizó MySQL, complementado con JavaScript para las validaciones, Bootstrap para el diseño CSS, Ajax de JQuery para el consumo del servicio web RestFul. Se escogieron estos componentes por ser software libre, es de decir de uso libre o gratuito y por la facilidad de aprendizaje al tener una curva de aprendizaje muy baja.

Para el desarrollo del sistema prototipo se aplicó el modelo Cliente-Servidor para lo cual se utilizó Restful, siendo este un servicio web basado en la arquitectura REST y que emplea el formato de texto JSON para el intercambio de datos.

Finalmente, en el capítulo III se muestran los resultados obtenidos de las pruebas de funcionamiento que se les aplicó a cuatro personas, quienes asumieron los diferentes roles del sistema prototipo.

PALABRAS CLAVE: unidad educativa, aplicación web, gestión académica, cliente-servidor, servicio web, software libre.

ABSTRACT

The objective of this curricular integration work was the development of a prototype system (web application) for the management of the academic information of an educational unit, which will allow managing the data of the different users that make up this institution (students, teachers, and representatives), student grades and academic certificates. Also, it will allow the representatives to know the performance and behavior of their representatives through academic observations.

Chapter I shows all the basic and necessary concepts, as well as the technological tools used to carry out this curricular integration work.

Chapter II has the design and implementation of the prototype system. In the design part is the Kanban methodology that was used for the development of the prototype system, as well as: requirements gathering, database design, use case diagram, interface sketch and class diagram. In the implementation part, the technological tools used for the development of the prototype system are shown, as follows: PHP (Hypertext Preprocessor) as a programming language, MySQL was used as a database manager, complemented with JavaScript for validations, Bootstrap for CSS layout, JQuery Ajax for RestFul web service consumption. These components were chosen because they are free software, that is, free or free to use, and because of their ease of learning, as they have a very low learning curve.

For the development of the prototype system, the Client-Server model was applied, for which Restful was used, this being a web service based on the REST architecture and that uses the JSON text format for data exchange.

Finally, chapter III shows the results obtained from the performance tests that were applied to four people, who assumed the different roles of the prototype system.

KEYWORDS: educational unit, web application, academic management, client-server, web service, free software.

1 INTRODUCCIÓN

En el Ecuador, las unidades educativas privadas, tienen la obligación de llevar una adecuada gestión de la información estudiantil, ya sea de forma manual o automática, lo que implica la realización de actividades académicas como: registro de las calificaciones, obtención de promedios finales, generación de certificados de promoción entre otras diligencias.

Existen instituciones educativas que no cuentan con un sistema de gestión de información de aprendizaje computarizado, lo que conlleva una gestión manual de la información o el uso de hojas de cálculo para ingresar las calificaciones y generar los cuadros de calificaciones. Para generar los certificados de promoción de cada estudiante, los empleados de estas instituciones transcriben las calificaciones de la hoja de cálculo a un editor de texto, de la misma manera para dar a conocer las calificaciones de los alumnos a sus representantes legales se transcriben dichas calificaciones de la hoja de cálculo a un correo electrónico, y el proceso se repite para cada estudiante. En cuanto a las observaciones académicas, estas instituciones educativas todavía utilizan hojas impresas para informar a los representantes sobre el desempeño académico y comportamiento de sus representados o a su vez son los representantes quienes deben solicitar información a la institución para conocer dichos aspectos.

Mediante este prototipo se pretende ofrecer un sistema automatizado para la administración académica que ayude en los problemas que actualmente tienen ciertas instituciones educativas.

Actualmente existen varias aplicaciones para la gestión escolar. Una de ellas se puede encontrar en www.agora-erp.com/es/features, este software es de paga, el cual da la opción de acceder a una versión de prueba durante un periodo de 15 días. Este software es de escritorio el cual contiene gestión de estudiantes y profesores, gestión y planificación de cursos, gestión de asistencia, calificaciones y entrega de boletines de notas a sus alumnos, entre otras cosas [1]. El presente Trabajo de Integración Curricular propone una aplicación con interfaz web, en la cual los usuarios podrán acceder al sistema desde cualquier parte del mundo a través del internet. Además, los módulos de administración del sistema prototipo propuesto serán creados en función de las necesidades de una unidad educativa en específico, lo que lo diferenciará del software anteriormente mencionado.

1.1 Objetivo general

Desarrollar un sistema prototipo de administración estudiantil para una Unidad Educativa.

1.2 Objetivos específicos

Los objetivos específicos del presente Trabajo de Integración Curricular son.

1. Analizar los elementos teóricos indispensables para el desarrollo del prototipo.
2. Diseñar los diferentes módulos del sistema prototipo propuesto.
3. Analizar los resultados de las pruebas de funcionamiento del sistema prototipo.

1.3 Alcance

En el presente Trabajo de Integración Curricular se usarán datos proporcionados por la unidad educativa particular “Sociedad Educativa Futuro”, la cual ofrece educación básica y bachillerato y está ubicada en San Rafael, Valle de los Chillos [2], con esta información se desarrollará un sistema prototipo de administración estudiantil que permita la gestión de usuarios, notas, observaciones académicas y generación de certificados académicos.

El desarrollo del sistema prototipo estará basado en la metodología ágil Kanban ya que dicho método busca conseguir un proceso productivo, organizado y eficiente, también permite ser más resolutivo y reducir los tiempos de desarrollo [3].

Los requerimientos del sistema prototipo serán agrupados mediante módulos de administración los cuales separan dichos requerimientos, los módulos a implementarse son:

- Módulo de administración de usuarios.
- Módulo de administración de notas.
- Módulo de administración de observaciones académicas.
- Módulo de generación de certificados académicos.

Administración de usuarios:

- El administrador del sistema podrá crear, actualizar y borrar usuarios de tipo estudiante, profesor y representante.
- El administrador podrá realizar la búsqueda de usuarios según los parámetros que se recojan en la fase de requerimientos.

Administración de notas:

- El estudiante podrá revisar sus notas.
- El representante podrá revisar las notas de sus representados.
- El profesor podrá cargar las notas de los estudiantes.
- El administrador habilitará y deshabilitará la carga de notas al sistema.

Administración de observaciones académicas:

- El administrador podrá crear, leer, actualizar y borrar observaciones académicas generales.
- El profesor podrá crear, leer, actualizar y borrar observaciones académicas particulares.
- El representante podrá leer tanto las observaciones académicas generales como particulares de sus representados.

Generación de certificados académicos:

El administrador podrá buscar, leer e imprimir certificados académicos de los estudiantes tales como:

- Cuadro de calificaciones del primer quimestre.
- Cuadro de calificaciones del segundo quimestre.
- Cuadro de calificaciones finales.
- Certificado de promoción.

1.4 Marco teórico

En esta sección se muestran la base teórica y las tecnologías utilizadas para el desarrollo del presente Proyecto de Integración Curricular.

1.4.1 Aplicaciones web

Las aplicaciones web son aplicaciones en las cuales el cliente realiza peticiones a una aplicación remota a través de un navegador, donde puede ser accedida utilizando una red externa, como es internet, o una red interna, como es una intranet, y posteriormente recibe

una respuesta la cual se imprime en el navegador. Estas aplicaciones web usualmente son utilizadas en las arquitecturas cliente-servidor de tres niveles como son: un nivel superior que es el cliente web (Navegador), un nivel intermedio el cual se encarga de procesar los datos (Servidor web) y un nivel inferior que proporciona los datos (Base de datos) [4].

1.4.2 Arquitectura cliente-servidor

La arquitectura cliente-servidor centraliza los datos y realiza el procesamiento de datos en el servidor. En esta arquitectura la parte de cliente se encarga de la administración de la interfaz del usuario final, entrada de datos y la presentación de estas, en cambio el servidor se encarga de procesar los datos, búsqueda, inserción, actualización y eliminación [5].

Servidor web

El servidor web está a la espera de solicitudes de algún cliente (Navegador web). Cuando el servidor recibe dicha petición, este lo procesa y responde a través de una página web que mostrará el resultado en un navegador o bien mostrará un mensaje de error [6].

Cliente web

En el lado del cliente una vez que haya recibido la respuesta por parte del servidor este lo presentará por pantalla, en la cual el cliente es responsable de interpretar código HTML¹, es decir, mostrar fuentes, colores, datos que recibe del servidor entre otros. El cliente web son aplicaciones de tipo JavaScript, el servidor proporciona los datos al navegador (Cliente) donde son ejecutados, por lo que es necesario que el navegador pueda ejecutar scripts sin embargo puede añadirse otros lenguajes mediante el uso de *plugins* [6].

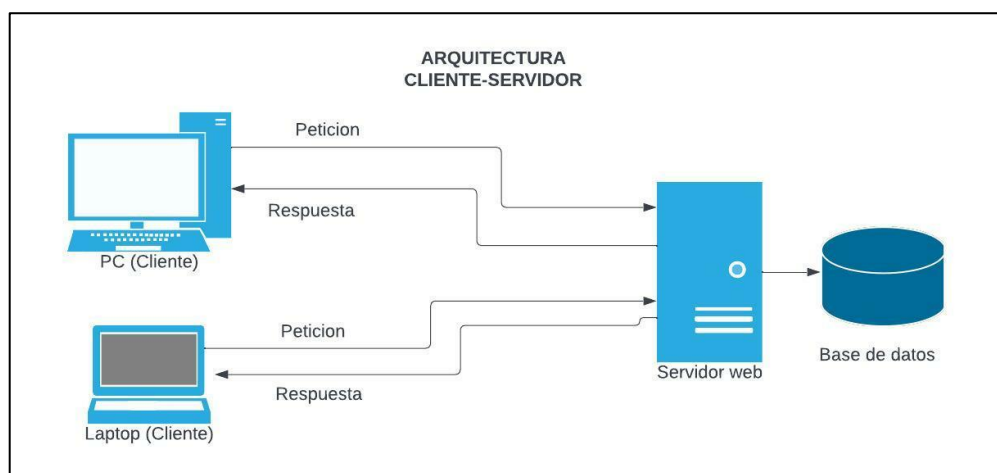


Figura 1.1 Arquitectura Cliente-Servidor

¹ HTML: HiperText Markup Lenguaje

1.4.3 Servicio web

Los servicios web permiten a los sistemas heterogéneos comunicarse con otros por medio de mensajes, Ya que los sistemas pueden ser heterogéneos, surge la necesidad de la interoperabilidad; por tal razón XML² es utilizado con frecuencia para dar formato a los mensajes, ya que XML está en formato de texto, la mayoría de sistemas pueden entender los mensajes y trabajar con ellos, estos mensajes son utilizados cuando la comunicación se produce entre dos aplicaciones que se ejecutan en diferentes maquinas [7].

Un servicio web es una aplicación que se ejecuta en un servidor remoto y puede ser accedida a través de protocolos de comunicación, independientes de las tecnologías que se utilizan, o las plataformas en las que se ejecutan, tanto en el cliente como en el servidor en el cual está alojado el servicio [8].

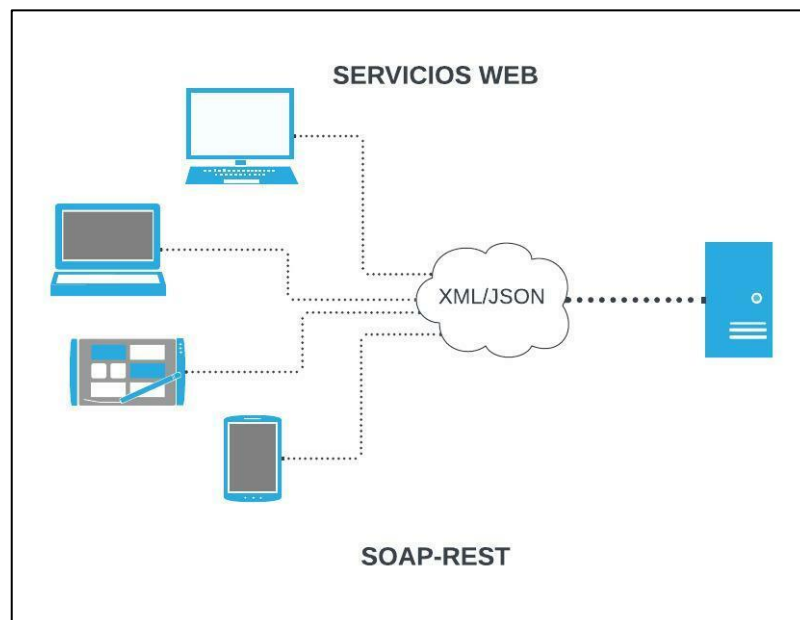


Figura 1.2 Servicios web

1.4.3.1 Protocolos y estándares de servicios web

Para que exista comunicación con un servicio web se necesita un lenguaje específico con una estructura definida; razón por la cual, se definen protocolos y estándares que describen la manera en cómo se comunican los servicios web, así como la estructura que se debe utilizar para el envío y recepción de los datos, entre otras cosas. A continuación, se muestra algunos de los protocolos y estándares básicos que puede utilizar un servicio web [9].

² XML: Extensible Markup Language

- XML (eXtensible Markup Language): Se define como un lenguaje de marcado que contiene un conjunto de reglas para la codificación de archivos. Es considerado un estándar, que a su vez posee otros estándares como DTD³ o XSD⁴ para definir el lenguaje y XSL-FO⁵ para la presentación [9].
- JSON (JavaScript Object Notation): Se define como un lenguaje simple de leer y escribir, el cual es utilizado para el intercambio de datos, [9].
- SOAP (Simple Object Access Protocol): Es un protocolo que utiliza XML, el cual describe un formato de los archivos para el intercambio de información entre el Cliente y el Servidor, comúnmente utilizado en servicios web [9].
- REST (Representation State Transfer): Para la comunicación entre el Cliente y el Servidor utiliza el protocolo HTTP⁶, los mensajes intercambiados pueden estar en XML o JSON [10].

1.4.3.2 RestFul

RestFul es un servicio web, el cual se encarga de proveer interoperabilidad entre sistemas de computadoras en internet, los servicios web compatibles con REST utilizan un conjunto de operaciones sin estado para ingresar y gestionar representaciones textuales del recurso web [11].

Un servicio web RestFul proveerá datos a través de una API⁷ sobre una red utilizado HTTP, posiblemente interactuando con otros servicios web y con bases de datos. Al probar una RestFul API se plantea desafíos como las entradas y salidas son secuencias de solicitudes y respuestas HTTP a un servidor remoto [12].

1.4.4 Metodología Kanban

1.4.4.1 Definición

Se describe como un sistema de producción eficiente, forma parte de las metodologías ágiles y su objetivo es administrar la ejecución de un proyecto desde su inicio hasta su culminación. La palabra kan viene de las palabras japonesas kan que significa visual y ban que se refiere a tarjeta, por lo tanto, Kanban significa tarjetas visuales. Esta metodología es simple de adoptar por un equipo e incluso la puede utilizar una sola persona, Kanban al

³ DTD: Document Type Definition.

⁴ XSD: XML Schema Definition.

⁵ XSL-FO: eXtensible Stylesheet language Formatting Objects.

⁶ HTTP: HyperText Transfer Protocol.

⁷ API: Application Programming Interface.

ser una metodología visual se puede apreciar el estado de los proyectos con solo una mirada y así poder asignar nuevas tareas de un modo efectivo. Para aplicar esta metodología es necesario un tablero de tareas que mejorará el trabajo [13].

1.4.4.2 Implementación de la metodología Kanban

Se puede crear un tablero Kanban con una pizarra, hoja de cálculo, herramientas de gestión de trabajo, entre otros. Independientemente de cómo este creado el tablero Kanban se recomienda seguir los siguientes pasos [14].

1.4.4.2.1 Empezar con un tablero en blanco

Obviamente el tablero Kanban estará compuesto por todas las actividades o tareas que tendrá un proyecto, sin embargo, es recomendable iniciar con un tablero sin nada encima, es decir en blanco, incluso esta recomendación se aplica a las herramientas de gestión de trabajos [14].

1.4.4.2.2 Creación de columnas para representar el proyecto

En general las columnas del tablero Kanban están vinculadas a las diferentes etapas del proyecto a desarrollarse, a continuación, se muestran algunas de las columnas más comunes [14].

- **Por hacer:** Esta columna es donde se agregarán todas las tareas, previas a ser distribuido a cada miembro de un equipo[14].
- **En progreso:** En esta columna se encuentran las tareas que se están ejecutando actualmente. Esta columna se podría dividir en otras columnas, dependiendo del proyecto que se esté realizando y del equipo que lo esté ejecutando, por ejemplo, en un equipo de ingeniería se puede tener tres columnas adicionales, una para el desarrollo, otra para las pruebas y una última para la implementación[14].
- **Trabajo terminado (Finalizado):** Las tareas que ya se finalizaron deberían aparecer en esta columna y así continuar con las tareas que se encuentran en las otras columnas[14].

TABLERO KANBAN



Figura 1.3 Tablero Kanban

1.4.4.2.3 Añadir tareas para representar el proyecto

En el tablero Kanban las tareas estarán representadas por tarjetas en las cuales el título de las tareas deberá ser concreto y que empiece con un verbo para dar a conocer a los miembros del grupo lo que deben hacer [14].

1.4.4.2.4 Hacer que el trabajo avance por etapas

El principal objetivo de los tableros Kanban es hacer que el trabajo vaya avanzando por cada una de las etapas, se lo puede hacer manualmente es decir moviendo las tareas entre las columnas o con la ayuda de una herramienta digital de gestión del trabajo [14].

1.4.5 Herramientas y tecnologías

1.4.5.1 HTML

1.4.5.1.1 Definición

Existen personas que piensan que es un lenguaje de programación, pero no lo es, HTML es un lenguaje de marcado o marcación de hipertexto y es fundamental que se utilice con algún lenguaje de programación para el desarrollo de aplicaciones web [15].

1.4.5.1.2 Utilidad

HTML es un lenguaje abstracto que utilizan ciertas aplicaciones para representar documentos (páginas web) y que puede ser transmitido a través de una red sea esta internet o una intranet. Los navegadores son los encargados de procesar e interpretar

archivos escritos en HTML. HTML es soportados por varios navegadores, entre los más populares tenemos a Chrome, Firefox, Edge, Internet Explorer, Safari entre otros [15].

1.4.5.2 JavaScript

1.4.5.2.1 Definición

JavaScript es un lenguaje de programación usualmente utilizado para el desarrollo de páginas web, estas páginas web son consideradas estáticas ya que se limitan a mostrar información establecido por el desarrollador de la página web y no tienen la capacidad de interactuar con el usuario, sin embargo, JavaScript incorpora cierta capacidad de interacción entre el usuario y las páginas web, es decir que dichas páginas se convierten en dinámicas. Es importante tener en cuenta que JavaScript se ejecuta en el Navegador (Cliente) [16].

1.4.5.2.2 Características

- JavaScript es un lenguaje interpretado, es decir que las instrucciones se ejecutan secuencialmente y no necesita de un compilador.
- JavaScript es un lenguaje simple.
- El código JavaScript se ejecuta en el Navegador (Cliente).
- JavaScript al estar escrito con un editor de texto plano y embebido en HTML el navegador es capaz de ejecutarlo e interpretarlo sin procesos intermedios.
- Es más flexible ya que no requiere la declaración de variables ni de tipos, aunque se recomienda hacerlo.

1.4.5.3 JQuery

1.4.5.3.1 Definición

JQuery es una biblioteca de JavaScript desarrollada para que los scripts que interactúan con HTML sean más simples, JQuery es una biblioteca que pertenece a los softwares de código abierto y tiene una licencia dual, siendo una de las más populares de la biblioteca de JavaScript [17].

1.4.5.3.2 Características

- Crea animaciones y manipula eventos.

- Trabaja con Ajax⁸ y DOM⁹.
- Reduce el código de JavaScript.
- Utiliza una gran cantidad de plugins creados por otros desarrolladores.

1.4.5.4 CSS

1.4.5.4.1 Definición

CSS (Cascading Style sheet/Hoja de Estilos en Cascada) es un lenguaje de diseño gráfico que se encarga de formatear la información que entrega HTML ya sea imágenes, texto, videos, audios o cualquier elemento creado. Por ejemplo, se puede manipular el audio, controlando el volumen, el tipo de voz o por cual parlante se va a sacar el audio. Con CSS se puede formatear algunas características básicas de los elementos como: colores, fondos, márgenes, posiciones, fuente entre otros, por tal motivo CSS se ha convertido en una herramienta indispensable para los desarrolladores en el diseño web [17].

1.4.5.4.2 Características

- Se encarga de dar estilos a los elementos HTML.
- Maneja una sintaxis simple.
- Puedes ser utilizado en todos los navegadores.

1.4.5.5 Bootstrap

Bootstrap es un Framework que cuenta con una gran cantidad de herramientas potentes para el diseño web en la parte de front-end la cual ayudará a diseñar páginas web en muy poco tiempo [18].

Bootstrap fue creado por Twitter para el desarrollo de interfaces web con la ayuda de CSS y JavaScript, ayuda al diseño adaptativo “responsive design” es decir que la interfaz web se adapta al tamaño del dispositivo, sea este una computadora, una tableta o un teléfono inteligente, convirtiéndose en una herramienta muy demandada en estos últimos años. Este Framework ofrece una gran cantidad de elementos con estilos predefinidos y de fácil configuración como formularios, botones, menús, tablas, ventanas modales, barras de navegación y muchos más [19].

⁸ AJAX: Asynchronous JavaScript And XML.

⁹ DOM: Document Object Model.

1.4.5.6 PHP

1.4.5.6.1 Definición

PHP (Hypertext PreProcessor) es muy conocido entre los lenguajes de programación, el cual es utilizado por los desarrolladores de software para crear aplicaciones web y puede ser incrustado en HTML, para lo cual utiliza etiquetas propias de inicio y fin que permite al desarrollador entrar y salir del modo PHP [20].

PHP se diferencia de otros lenguajes ya que el código es ejecutado en el lado del servidor el cual genera un HTML y se lo envía al cliente, siendo este el que ejecuta el script. Se puede configurar el servidor web para que PHP procese archivos HTML sin dar a conocer a los usuarios sobre estos procesos [20].

1.4.5.6.2 Características

- Se utiliza para crear aplicaciones web.
- El código es ejecutado en el lado del Servidor.
- Puede ser empleado en varios sistemas operativos conocidos como: Linux, Windows, MacOS, entre otros.
- Puede generar imágenes, ficheros PDFs inclusive películas flash.
- Tiene la posibilidad de incorporar programación orientado a objetos (POO).
- Soporta una gran cantidad de base de datos.
- Cuenta con gran cantidad de protocolos para la comunicación con otros servicios.

1.4.5.7 MySQL

1.4.5.7.1 Definición

MySQL (My Structured Query Language) es un sistema de código abierto muy popular por su rendimiento y fiabilidad, es utilizado para gestionar bases de datos. Anteriormente, MySQL se dedicaba a las aplicaciones personales y de gama baja, sin embargo, ha ido creciendo y en la actualidad la mayoría de los sitios que gestionan gran cantidad de datos, como redes sociales, portales comunitarios han alcanzado su rendimiento mediante este gestor de base de datos [21].

1.4.5.7.2 Características

- MySQL fue escrito en C y C++.

- Funciona sobre muchos sistemas operativos.
- Soporta API como C, C++, PHP, Perl, Python, Java entre otros.
- Tiene un sistema de administración de memoria muy potente.
- Es compatible con el lenguaje SQL
- Soporta columnas de gran tamaño como texto y binario.
- Gestiona bases de datos que manejan gran cantidad de información [22].

1.4.5.8 WampServer

1.4.5.8.1 Definición

WAMP (Windows Apache MySQL PHP) lo que quiere decir que es una herramienta para el sistema operativo Windows el cual posee un conjunto de soluciones de software, siendo estas: Apache, MySQL y PHP, estos softwares son instalados en conjunto por una buena razón. WAMP se deriva de LAMP y se diferencian en que WAMP es para Windows y LAMP es para LINUX [23].

Representación de cada letra en WAMP.

- W significa el sistema operativo soportado, por ejemplo, LAMP es para Linux y MAMP para Mac.
- A significa Apache, el cual es un servidor web.
- M significa MySQL el cual trabaja como un sistema de gestor de datos para el servidor.
- P significa PHP el cual es un lenguaje de programación el cual trabaja en conjunto con Apache y MySQL.

1.4.5.8.2 Características

- Actúa como un servidor web, pero de manera local por lo que se tiene acceso a todas las funciones de un servidor web sin la necesidad de conectarse a internet.
- Facilita la creación de copias de seguridad.
- Acelera el proceso de trabajo para los desarrolladores.
- Ahorro de tiempo ya que no se tendrá que cargar archivos en un sitio.

1.4.5.9 Apache

1.4.5.9.1 Definición

Apache Server Project fue quien creó el Servidor Apache, el cual es un servidor web gratuito, confiable y eficiente, al ser un software de código abierto los usuarios pueden agregar características adicionales. Existe un conjunto de voluntarios conocido como grupo Apache los cuales se encargan del desarrollo y la documentación del servidor web [24].

1.4.5.9.2 Ventajas

- El Servidor Apache puede ser utilizado de forma comercial y no comercial ya que posee una licencia de código abierto.
- Tiene a su disposición un conjunto de desarrolladores que adopta un proceso libre de desarrollo.
- Los usuarios de Apache pueden añadir funcionalidades extras.
- Apache puede trabajar en las versiones actuales de Windows, BeOs, mainframes, Unix y Linux.

1.4.5.9.3 Funcionamiento

Apache Web Server es una máquina donde se ejecuta el programa *daemon http, httpd*, este recibe solicitudes enviadas desde un cliente web cualquiera que sea este y se encarga de responder con el recurso pedido. Para la configuración de la red, Apache debe iniciarse con permisos de super usuario (root) y conectarse a un puerto, usualmente el puerto 80 para escuchar solicitudes y aceptar conexiones [24].

1.4.5.10 phpMyAdmin

1.4.5.10.1 Definición

phpMyAdmin es una herramienta de código abierto escrito en PHP el cual maneja la administración de MySQL. phpMyAdmin es una aplicación que utiliza XHTML, CSS y JavaScript en el lado del cliente, esta aplicación ofrece una interfaz web completa para gestionar las bases de datos, como es la de creación, actualización y eliminación de tablas, administración de los registros en la base de datos entre otros, por lo que es bastante reconocida en la actualidad [25].

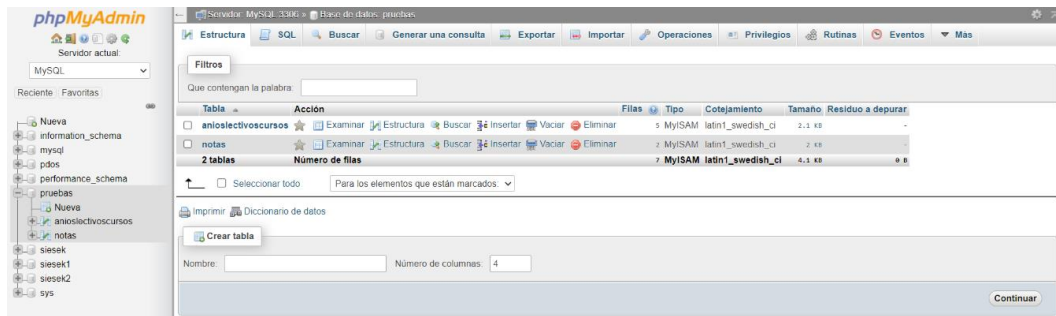


Figura 1.4 Interfaz web de phpMyAdmin

1.4.5.10.2 Características

- Interfaz web fácil de manejar.
- La aplicación es de código abierto.
- Cuenta con el apoyo de un conjunto de desarrolladores al ser un software de código abierto.
- Ofrece una gestión completa de servidores y datos MySQL.

1.4.5.11 SublimeText

1.4.5.11.1 Definición

SublimeText es un editor de texto diseñado para los usuarios que pasan una gran cantidad de tiempo escribiendo código el cual soporta muchos formatos de archivos. Ofrece un entorno de desarrollo (IDE) oscuro para resaltar las líneas de código y así mantener la concentración del usuario. Por lo tanto, SublimeText es un editor de texto ligero lo que lo hace más rápido y fácil de utilizar [26].

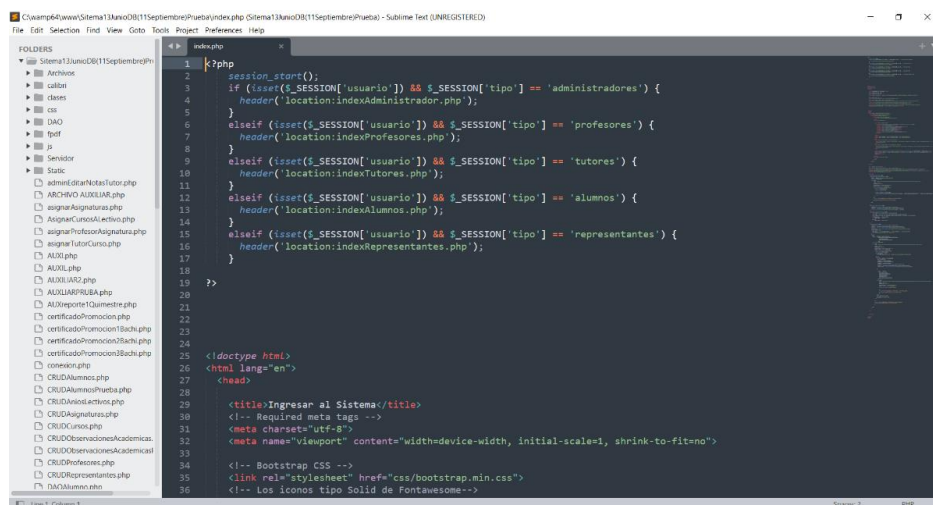


Figura 1.5 SublimeText IDE

1.4.5.11.2 Características

- La simplicidad de SublimeText ya que posee una interfaz sencilla de manejar, no muestra barras de herramientas ni diálogos de configuración.
- Posee una herramienta llamada "Minimapa" el cual permite al usuario ver en qué parte del archivo se encuentra en ese momento.
- Se puede modificar los colores del código, dependiendo del lenguaje que se esté utilizando, así mismo SublimeText contiene una gran cantidad de temas y colores.
- Soporta una gran cantidad de lenguajes como: C, C++, C#, Groovy, Haskell, HTML, Java, JavaScript, Matlab, PHP, Python, Ruby, etc.
- Dispone de una herramienta de autocompletado lo que lo hace más sencillo y rápido al momento de escribir código.

2 METODOLOGÍA

El presente capítulo se divide en dos fases, una fase de diseño y otra de implementación. En la fase de diseño se recopila toda la información necesaria para definir el funcionamiento del sistema prototipo, mientras que en la fase de implementación se encuentra la codificación basada en la fase de diseño. Tanto en la fase de diseño como la de implementación se utiliza la metodología Kanban.

2.1 DISEÑO

Basado en la metodología ágil Kanban se elabora una lista de tareas a desarrollarse la cual permite organizar el flujo de trabajo. En esta fase la primera tarea a realizar es la obtención de requerimientos funcionales y no funcionales a través de entrevistas a los diferentes funcionarios de la unidad educativa particular “Sociedad Educativa Futuro”. Se revisan los requerimientos y se los organizan en diagramas de casos de uso, el cual, relaciona directamente dichos requerimientos con los actores identificados, esto permite definir módulos del sistema prototipo. Se crea el diagrama relacional, que ayuda con el modelado de los datos para la gestión de la base de datos. Se elabora los bosquejos de las interfaces de cada uno de los módulos del sistema prototipo. Por último, se desarrolla el diagrama de clases, el cual describe la estructura del sistema prototipo.

2.1.1 Tablero Kanban

En el tablero Kanban se encuentran todas las tareas respecto a la fase de diseño las cuales están organizadas en tres columnas: por hacer, en progreso y finalizado como se puede apreciar en la Tabla 2.1.

Tabla 2.1 Tablero Kanban para la fase de Diseño

| POR HACER | EN PROGRESO | FINALIZADO |
|---|---|--|
| Implementación del diagrama relacional. | Recolección de los requerimientos a los diferentes funcionarios de la unidad educativa a través de entrevistas. | Estudio de la base teórica necesaria para el desarrollo del sistema prototipo. |
| Implementación de las interfaces web. | Clasificación de los requerimientos en funcionales y no funcionales. | |
| Implementación de la lógica de negocio. | Elaboración del diagrama de casos de uso | |
| | Elaboración del diagrama relacional | |
| | Diseño de interfaces web. | |
| | Elaboración del diagrama de clases | |

2.1.2 Recolección de requerimientos

Para la recolección de requerimientos se realizaron entrevistas a los diferentes funcionarios de la unidad educativa “Sociedad Educativa Futuro”, donde se identificaron los diferentes usuarios que intervienen en la gestión académica, como se muestra a continuación.

- **Administrador:** Se encargará de toda la gestión del sistema, como la de ingreso, actualización, eliminación, asignación, entre otros, de cada elemento que pertenece al sistema. En este caso la secretaria es quien toma el rol de administrador.
- **Profesor:** Se encargará del ingreso de las notas del estudiante, de las diferentes materias que imparte.
- **Tutor:** Se encargará de ingresar notas de comportamiento y proyectos escolares según sea el caso, adicionalmente se encargará de la administración de las observaciones académicas particulares.
- **Estudiante:** Podrá consultar sus notas de todas las asignaturas que toma.
- **Representante:** Podrá consultar las notas de sus representados, así como observaciones académicas.

2.1.2.1 Historias de usuario

Para la recolección de los requerimientos funcionales y no funcionales se utilizaron las Historias de Usuario (HU) por la facilidad de organizar los datos.

Las Historias de Usuario son descripciones sencillas de la aplicación explicadas desde el punto de vista de la persona que desea dicha aplicación, por lo general el usuario final o cliente [27].

En la Tabla 2.2 se puede apreciar la estructura de la Historia de Usuario.

Tabla 2.2 Historia de Usuario

| Historia de Usuario | |
|---------------------|-------------------------|
| ID: | Nombre Historia: |
| Usuario: | |
| Descripción: | |

Donde:

ID: Es el identificador de la Historia de Usuario.

Nombre Historia: Es un título que describe la Historia de Usuario.

Usuario: Es la personal que va a utilizar la funcionalidad descrita en la Historia de Usuario.

Descripción: Es una descripción clara y precisa de la Historia de Usuario.

A continuación, se muestran algunos ejemplos de las Historias de Usuario, realizados en base a los requerimientos recolectados en las entrevistas. Toda la información respecto a las Historias de Usuario se encuentra en el **Anexo I**.

En la Tabla 2.3 se tiene la Historia de Usuario para el acceso al sistema.

Tabla 2.3 Historia de Usuario HU001

| Historia de Usuario | |
|---|--|
| ID: HU001 | Nombre Historia: Acceso al sistema. |
| Usuario: Administrador, Profesor, Tutor, Estudiante, Representante. | |
| Descripción: Cada uno de los usuarios usaran un nombre de usuario y una contraseña para ingresar al sistema. | |

Para la administración del usuario Profesor se muestran como ejemplo. las siguientes Historias de usuario. En la Tabla 2.4 se muestra la Historia de Usuario para el registro del usuario Profesor.

Tabla 2.4 Historia de Usuario HU027

| Historia de Usuario | |
|---|---|
| ID: HU027 | Nombre Historia: Registro de profesores. |
| Usuario: Administrador | |
| Descripción: Para el registro de profesores se tomarán los siguientes datos: apellidos, nombres, número de cédula, dirección, número de teléfono, correo electrónico, usuario y contraseña. El usuario y contraseña serán utilizados para ingresar al sistema. | |

En la Tabla 2.5 se muestra la Historia de Usuario para la eliminación del usuario Profesor.

Tabla 2.5 Historia de Usuario HU030

| Historia de Usuario | |
|--|--|
| ID: HU030 | Nombre Historia: Eliminación de profesores. |
| Usuario: Administrador | |
| Descripción: Se eliminarán todos los datos del profesor seleccionado siempre y cuando no tengan relación con otros elementos del sistema. | |

En la Tabla 2.6 se muestra la Historia de Usuario para asignar profesores a signaturas.

Tabla 2.6 Historia de Usuario HU036

| Historia de Usuario | |
|--|---|
| ID: HU036 | Nombre Historia: Asignación de profesores a asignaturas. |
| Usuario: Administrador | |
| Descripción: El usuario podrá asignar un profesor a una asignatura previamente seleccionados o cancelar dicha asignación. | |

2.1.3 Requerimientos funcionales y no funcionales

2.1.3.1 Requerimientos funcionales

Una vez realizada las Historias de Usuario con la información recolectada en las entrevistas se procede a generar una lista de los requerimientos funcionales, como se muestra en Tabla 2.7.

Tabla 2.7 Requerimientos funcionales

| Requerimientos funcionales | |
|----------------------------|---|
| 1 | Acceso al sistema, los usuarios utilizarán un nombre de usuario y contraseña para ingresar al sistema. |
| 2 | El administrador puede realizar la gestión de años lectivos (creación, actualización, activación y búsqueda). |
| 3 | El administrador puede realizar operaciones CRUD de cursos. |
| 4 | El administrador puede buscar y asignar cursos a años lectivos. |
| 5 | El administrador puede ver, buscar y asignar asignaturas a cursos. |
| 6 | El administrador puede realizar operaciones CRUD de los diferentes usuarios del sistema (profesores, estudiantes y representantes). |
| 7 | El administrador puede buscar y asignar profesores a asignaturas. |
| 8 | El administrador puede buscar y asignar estudiantes a cursos. |

| | |
|----|--|
| 9 | El administrador puede buscar y asignar profesores (tutores) a cursos. |
| 10 | El administrador puede realizar operaciones CRUD de las fechas de habilitación para la carga de notas. |
| 11 | El administrador puede editar las notas que ingresó el profesor, así como las notas del comportamiento que ingresó el tutor. |
| 12 | El administrador puede realizar operaciones CRUD de representantes. |
| 13 | El administrador puede realizar operaciones CRUD de las observaciones académicas generales. |
| 14 | El administrador puede generar los cuadros de calificaciones del primer y segundo quimestre, cuadro de calificaciones finales y certificados de promoción. |
| 15 | Los usuarios (Profesores, estudiantes y representantes) pueden ver su información personal |
| 16 | El profesor puede ver lista de asignaturas que imparte. |
| 17 | El profesor puede ver lista de estudiantes con sus respectivas notas, de las materias que imparte. |
| 18 | El profesor puede ingresar las notas de los estudiantes, de las materias que imparte. |
| 19 | El tutor puede ver lista de los estudiantes que tiene a su cargo. |
| 17 | El tutor puede realizar operaciones CRUD de las observaciones académicas particulares. |
| 18 | El tutor puede ingresar las notas del comportamiento de los estudiantes, que están a su cargo. |
| 19 | El estudiante puede ver las notas de las materias que toma. |
| 20 | El representante puede ver las notas de sus representados. |
| 21 | El representante puede ver tanto las observaciones académicas generales como las observaciones académicas particulares. |

2.1.3.2 Requerimientos no funcionales

Al igual que para los requerimientos funcionales, los requerimientos no funcionales se generan a partir de las Historias de Usuario, como se muestra en la Tabla 2.8.

Tabla 2.8 *Requerimientos no funcionales*

| Requerimientos no funcionales | |
|--------------------------------------|--|
| 1 | Disponibilidad: La aplicación debe estar disponible en un 99% y ser accedido desde cualquier parte del mundo a través de internet. |
| 2 | Simultaneidad: La aplicación debe soportar el ingreso de al menos treinta usuarios simultáneamente. |
| 3 | Presentación: La aplicación debe ofrecer una interfaz amigable con el usuario. |

2.1.4 Diagrama de casos de uso

Los diagramas de casos de uso describen de forma gráfica la interacción entre el sistema y los diferentes usuarios finales, es decir, identifica quienes son los usuarios que van a interactuar con el sistema, y como los usuarios esperan interactuar con dicho sistema [28].

Para el presente sistema prototipo se tienen cinco diagramas de casos de uso, uno por cada usuario identificado en el sistema.

En la Figura 2.1 se muestra el diagrama de Casos de Uso para el usuario administrador.

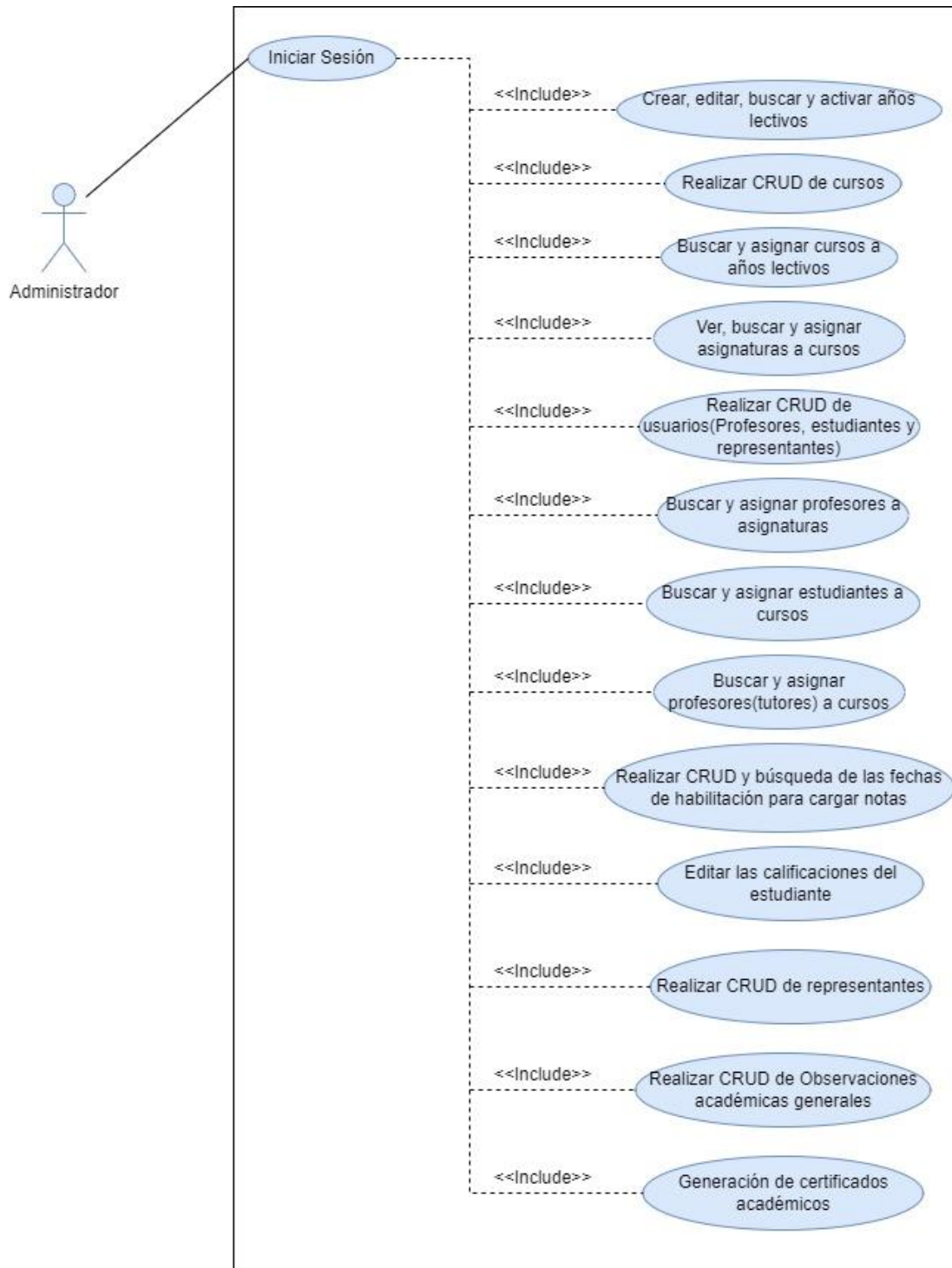


Figura 2.1 Diagrama de Casos de Uso para Administrador

En la Figura 2.2 se muestra el diagrama de Casos de Uso para el usuario Profesor.

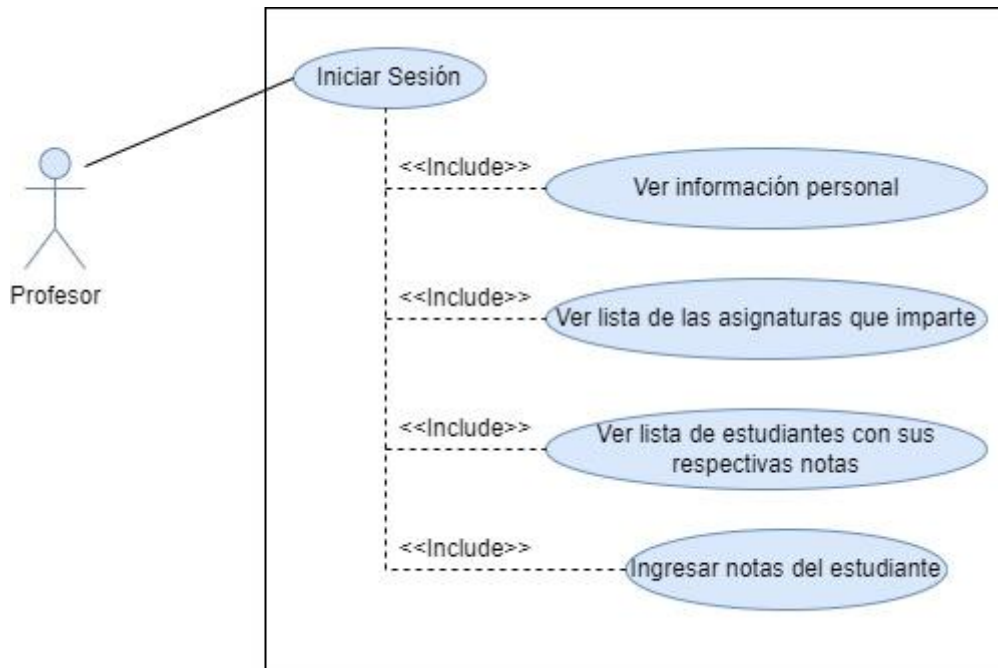


Figura 2.2 Diagrama de Casos de Uso para Profesor

En la Figura 2.3 se muestra el diagrama de Casos de Uso para el usuario Tutor.

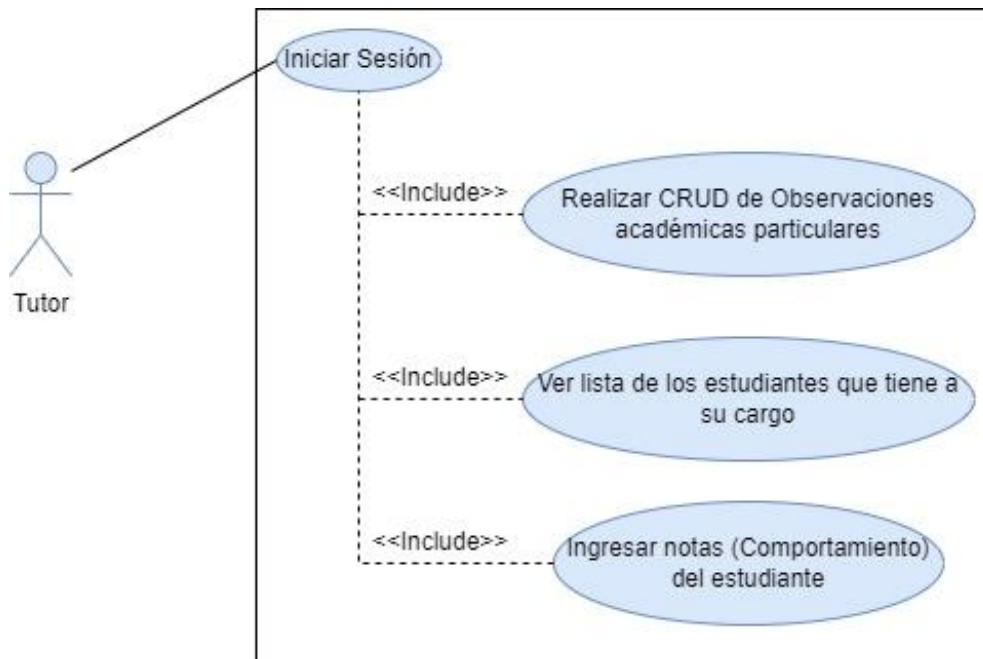


Figura 2.3 Diagrama de Casos de Uso para Tutor

En la Figura 2.4 se muestra el diagrama de Casos de uso para el Usuario Estudiante.

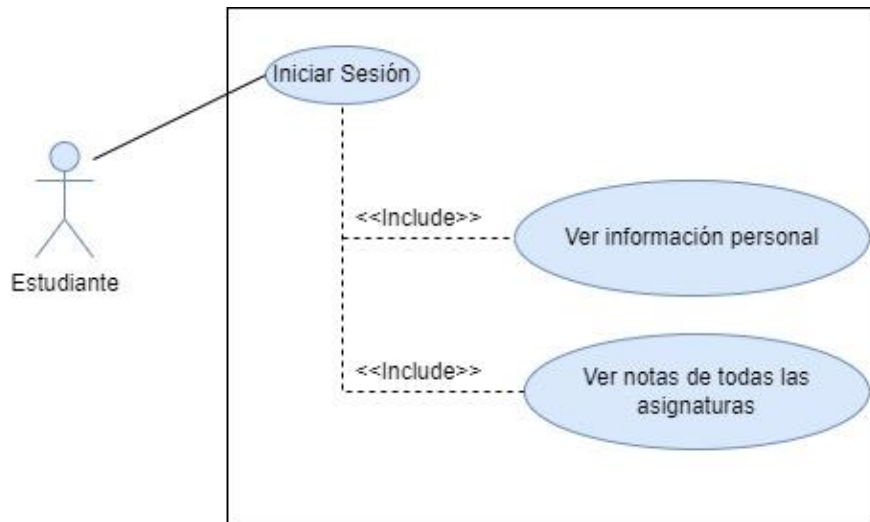


Figura 2.4 Diagrama de Casos de Uso para Estudiante

En la Figura 2.5 se muestra el diagrama de Casos de Uso para el usuario Representante.

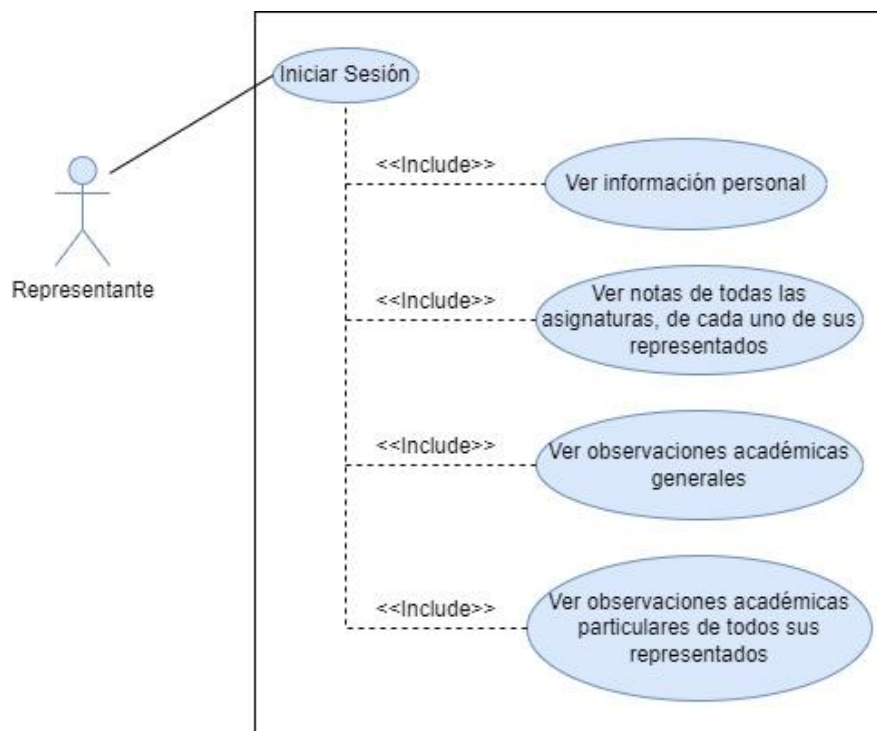


Figura 2.5 Diagrama de Casos de Uso para Representante

2.1.5 Diagrama Relacional

El diagrama relacional fue creado a partir del sistema prototipo, el cual representan un conjunto de tablas y las relaciones que existen entre ellas. Este diagrama ayuda al modelado de los datos para la gestión de la base de datos. En la Figura 2.6 se ve el diagrama relacional del presente sistema prototipo.

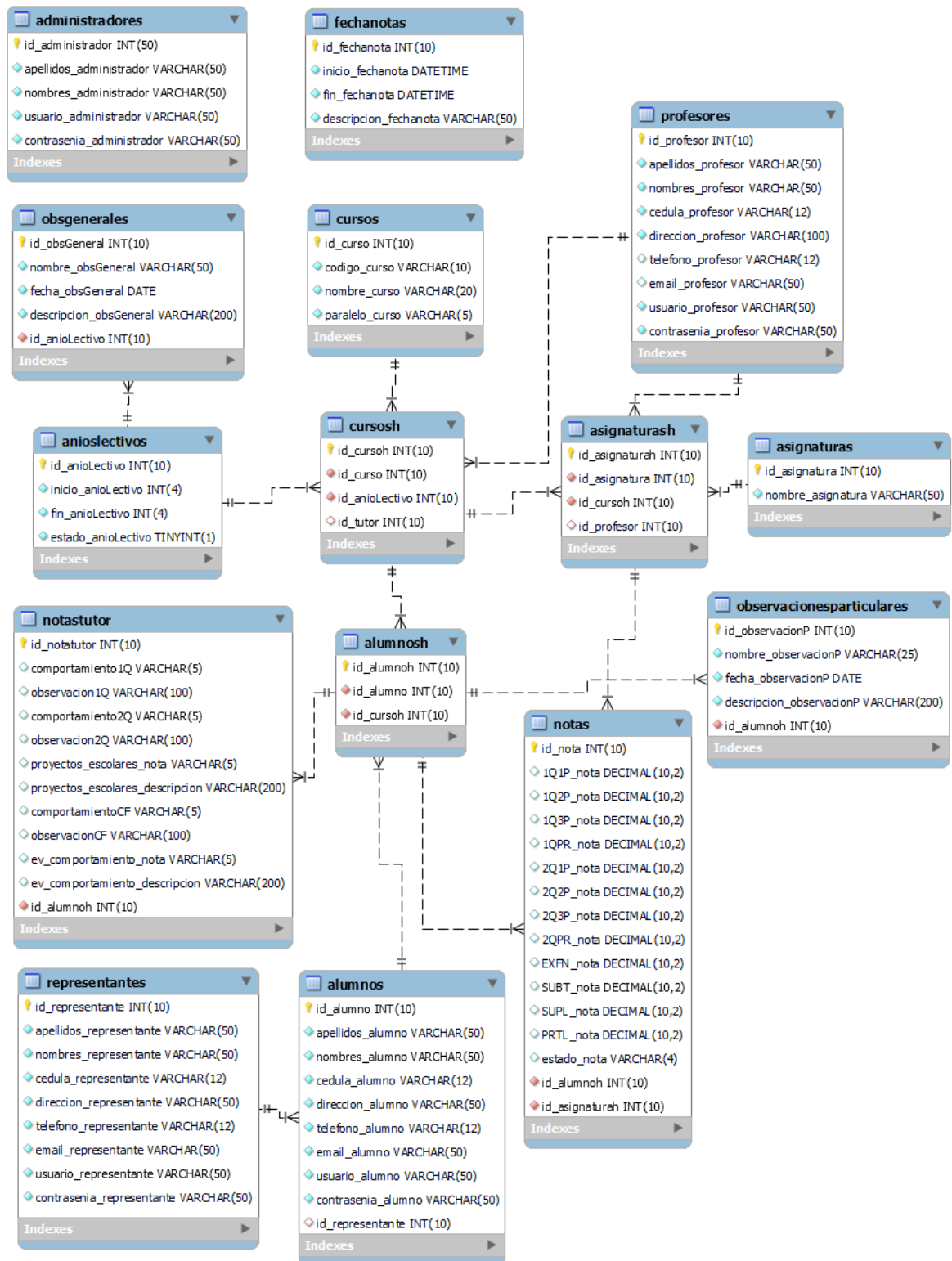


Figura 2.6 Diagrama relacional de la base de datos

2.1.6 Diseño de Interfaces

Para el diseño de las interfaces se utiliza la herramienta online <https://app.diagrams.net/> el cual permite realizar diferentes tipos de diagramas.

A continuación, se presenta algunas de las interfaces diseñadas para el sistema prototipo. mientras que el resto se lo puede encontrar en el **Anexo III**.

En la Figura 2.7 se muestra la interfaz para acceder al sistema, el cual está formado por: el perfil de usuario, nombre de usuario, contraseña (password) de usuario, un elemento para poder ver la contraseña y un botón para ingresar al sistema.

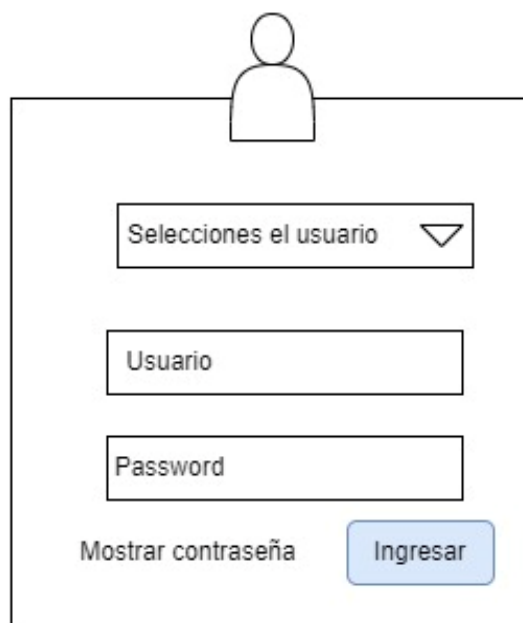


Figura 2.7 Diseño de interfaz: Acceso al sistema

En la Figura 2.8 se muestra la interfaz de inicio del usuario Administrador, en la cual se puede observar en la parte superior un logo, el año lectivo activo o vigente, el usuario que inicio sesión, la fecha actual, un botón para cerrar la sesión y una barra de navegación, donde se encuentran los diferentes módulos del sistema. Esta cabecera es fija para todas las interfaces del usuario Administrador. Adicionalmente, se tiene la administración del usuario Profesor, en donde se ve una sección para buscar el usuario, un botón para crear profesor y una lista de profesores ya creados, en esta lista se encuentran las opciones de editar y eliminar el profesor.

| | | | | | | | |
|--|------------|-------------|------------|---|----------------|--------------------------|--------------|
| | LOGO | Año lectivo | Usuario | Fecha | Cerrar sesion | | |
| Años lectivos | Cursos | Asignaturas | Profesores | Alumnos | Representantes | Observaciones academicas | Certificados |
| Administrar profesores PROFESORES Buscar <input type="text"/> <input type="button" value="Buscar"/> Lista de profesores <input type="button" value="Crear profesor"/> | | | | | | | |
| ID | Apellidos | Nombres | Cedula | Acciones | | | |
| 1 | xxxx xxxxx | xxxx xxxxx | xxxxxx | Editar Eliminar | | | |
| | | | | | | | |

Figura 2.8 Diseño de interfaz: Administración de profesores (Administrador)

En la Figura 2.9 se muestra el diseño del formulario para crear el usuario profesor, este formulario será una ventana modal, la cual está conformada de: un botón en la parte superior para cerrar la ventana, un formulario que permite ingresar toda la información del usuario Profesor como son: apellidos, nombres, cédula, dirección, teléfono, email, nombre de usuario y contraseña. Adicionalmente se tienen dos botones en la parte inferior, una para guardar el registro y otra para cerrar la ventana.

Crear profesor ✕

ID

Apellidos

Nombres

Cedula

Direccion

Telefono

E-mail

Usuario

Contraseña

Figura 2.9 Diseño de interfaz: formulario para crear el usuario Profesor

2.1.7 Diagrama de clases

El diagrama de clases es utilizado para representar la lógica del negocio, donde se muestran las diferentes clases con sus respectivas relaciones de un sistema. En la Figura 2.10 se muestra el diagrama de clases para el presente sistema prototipo.

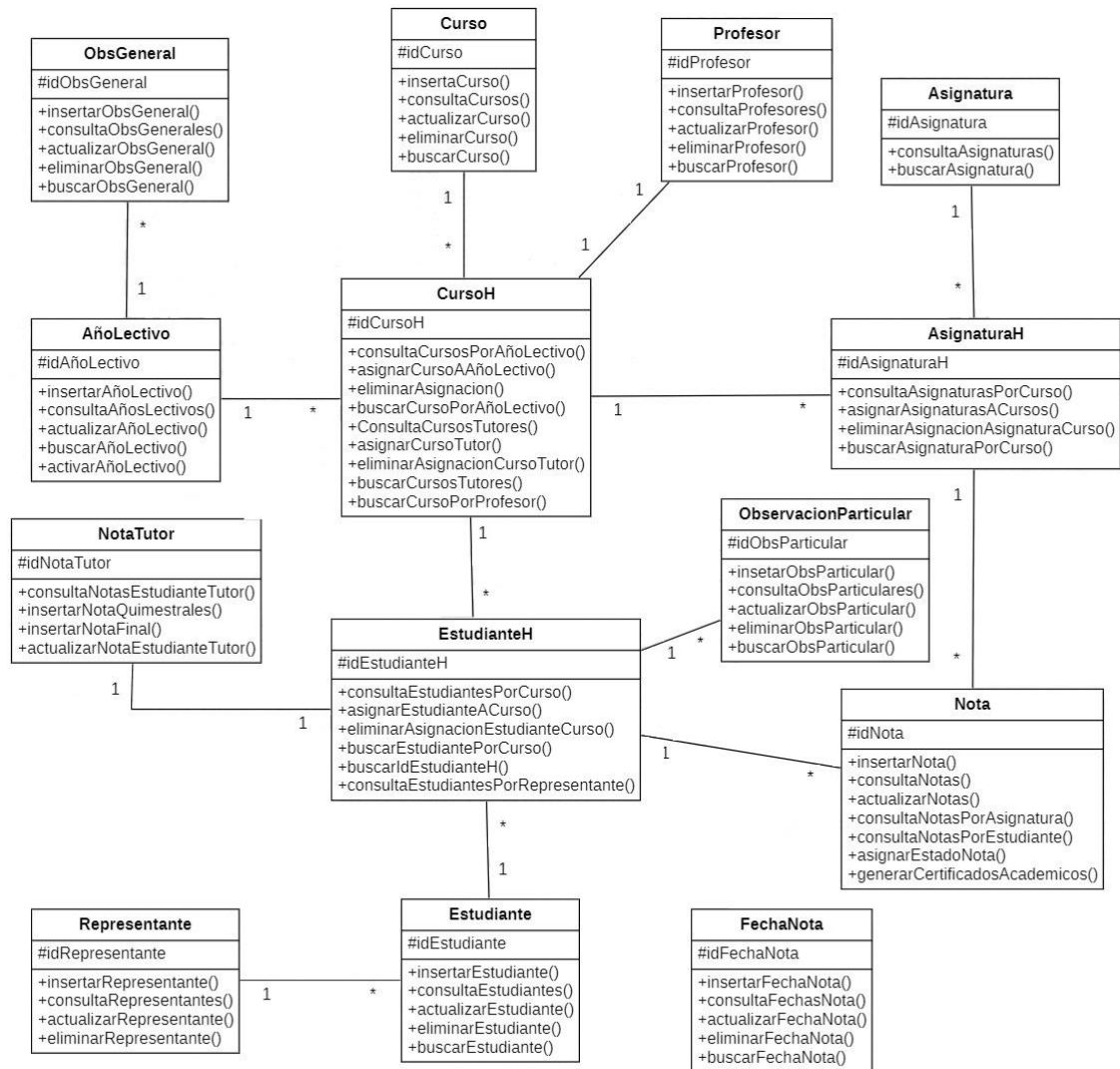


Figura 2.10 Diagrama de clases del sistema prototipo

2.2 IMPLEMENTACIÓN

Una vez comprendida completamente las funcionalidades y requerimientos del sistema prototipo, se procede con la actualización del tablero Kanban para la implementación de dicho sistema. Se empieza con la implementación del modelo relacional, el cual consiste en la creación de la base de datos tomando en cuenta el diagrama relacional definido en la fase de diseño. Después se realiza la implementación de las interfaces web, donde se crean las interfaces de los diferentes módulos que integran la aplicación, tomando como referencia los diseños realizados anteriormente. Finalmente, con el diagrama de clases se procede con la implementación de la lógica de negocio en la cual se programan o codifican los distintos módulos del sistema prototipo.

2.2.1 Tablero Kanban actualizado

Una vez finalizada la fase de diseño se procede a actualizar el tablero Kanban, donde se encuentran todas las tareas correspondientes a la fase de implementación. De la misma manera que en la fase de diseño, el tablero Kanban consta de tres columnas: por hacer, en progreso y finalizado.

En la Tabla 2.9 se muestra el tablero Kanban actualizado para la fase de implementación.

Tabla 2.9 Tablero Kanban para la fase de Implementación

| POR HACER | EN PROGRESO | FINALIZADO |
|--|---|--|
| Pruebas de funcionamiento. | Implementación del diagrama relacional. | Estudio de la base teórica necesaria para el desarrollo del sistema prototipo. |
| Presentación de los resultados obtenidos en las pruebas de funcionamiento. | Implementación de las interfaces web. | Recolección de los requerimientos a los diferentes funcionarios de la unidad educativa a través de entrevistas |
| | Implementación de la lógica de negocio. | Clasificación de los requerimientos en funcionales y no funcionales. |
| | | Elaboración del diagrama de casos de uso |
| | | Elaboración del diagrama relacional |
| | | Diseño de interfaces web. |
| | | Elaboración del diagrama de clases |

2.2.2 Instalación de las herramientas tecnológicas

Para la fase de implementación se instaló el software WampServer, el cual contiene un conjunto de soluciones de software tales como: Apache Web Server, PHP y MySQL, todos estos trabajando en un dispositivo local. Para el caso de MySQL se utilizó la aplicación web phpMyAdmin, la cual es utilizada para la gestión de base de datos del sistema prototipo.

Para la implementación de las interfaces web se instaló la aplicación SublimeText, la cual es un editor de texto que soporta una gran cantidad de formatos de archivos, siendo HTML el lenguaje de marcado utilizado para la implementación de dichas interfaces. Adicionalmente, SublimeText fue utilizado para la implementación de la lógica de negocios, donde PHP es el lenguaje de programación utilizado para la codificación del sistema prototipo.

2.2.3 Implementación del modelo relacional

Para la implementación del modelo relacional se utilizó la aplicación web phpMyAdmin, donde se crea la base de datos con las tablas necesarias para la correcta gestión de la información del sistema prototipo.

A continuación, se muestran algunos ejemplos de los códigos usados para la implementación del modelo relacional.

En la Figura 2.11 en la línea 20, se muestra el código para la creación de la base de datos y en la línea 21 se selecciona la base de datos con la que se va a trabajar, en este caso para crear sus respectivas tablas.

```
20 • CREATE DATABASE sistemaprototipo;  
21 • USE sistemaprototipo;
```

Figura 2.11 Código para crear la base de datos

Una vez creada la base de datos para el sistema prototipo se procede a la creación de las tablas, tomando en cuenta el diseño del modelo relacional propuesto en la fase de diseño. En la Figura 2.12 se muestra el código para la creación de la tabla administradores con sus respectivos atributos, donde cada atributo tiene asociado un tipo de dato, en esta tabla se guardarán los usuarios de tipo administrador, entre sus atributos principales están el nombre de usuario y su respectiva contraseña, las cuales le ayudarán a acceder al sistema para poder administrarlo.

```

30 • DROP TABLE IF EXISTS `administradores`;
31 • CREATE TABLE IF NOT EXISTS `administradores` (
32     `id_administrador` int(50) NOT NULL AUTO_INCREMENT COMMENT 'id de administrador',
33     `apellidos_administrador` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
34     `nombres_administrador` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
35     `usuario_administrador` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
36     `contrasenia_administrador` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
37     PRIMARY KEY (`id_administrador`)
38 ) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

```

Figura 2.12 Código para crear la tabla administradores

Enseguida se crea el usuario administrador, el cual será el encargado de la gestión de varios módulos del sistema prototipo, entre ellos la administración de los usuarios de tipo profesor, estudiante (alumno) y representante, generación de certificados académicos, entre otros, lo que es de gran importancia ya que se podrá ir comprobando el funcionamiento de los diferentes módulos.

En la Figura 2.13 se muestra la creación del usuario Administrador con sus respectivos atributos, entre ellos el nombre de usuario y contraseña que serán utilizados para acceder al sistema.

```

44 • INSERT INTO `administradores` (`id_administrador`, `apellidos_administrador`,
45     `nombres_administrador`, `usuario_administrador`, `contrasenia_administrador`) VALUES
46     (1, 'APELLIDOS', 'NOMBRES', 'admin', 'admin');

```

Figura 2.13 Código para el registro del usuario Administrador

En la Figura 2.14 se muestra la creación de la tabla profesores con sus respectivos atributos, donde se guardará la información de los usuarios de tipo profesor.

```

134 • DROP TABLE IF EXISTS `profesores`;
135 • CREATE TABLE IF NOT EXISTS `profesores` (
136     `id_profesor` int(10) NOT NULL AUTO_INCREMENT,
137     `apellidos_profesor` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
138     `nombres_profesor` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
139     `cedula_profesor` varchar(12) COLLATE latin1_spanish_ci NOT NULL,
140     `direccion_profesor` varchar(100) COLLATE latin1_spanish_ci NOT NULL,
141     `telefono_profesor` varchar(12) COLLATE latin1_spanish_ci DEFAULT NULL,
142     `email_profesor` varchar(50) COLLATE latin1_spanish_ci DEFAULT NULL,
143     `usuario_profesor` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
144     `contrasenia_profesor` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
145     PRIMARY KEY (`id_profesor`)
146 ) ENGINE=InnoDB AUTO_INCREMENT=41 DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

```

Figura 2.14 Código para crear la tabla profesores

Como último ejemplo, como se muestra en la Figura 2.15, se tiene la creación de la tabla notas, donde se guardarán las notas de todas las asignaturas de cada uno de los estudiantes, correspondientes a los distintos parciales de los dos quimestres, promedios quimestrales, examen final, supletorio y promedios finales.

```
195 • DROP TABLE IF EXISTS `notas`;
196 • CREATE TABLE IF NOT EXISTS `notas` (
197     `id_nota` int(10) NOT NULL AUTO_INCREMENT,
198     `1Q1P_nota` decimal(10,2) DEFAULT NULL,
199     `1Q2P_nota` decimal(10,2) DEFAULT NULL,
200     `1Q3P_nota` decimal(10,2) DEFAULT NULL,
201     `1QPR_nota` decimal(10,2) DEFAULT NULL,
202     `2Q1P_nota` decimal(10,2) DEFAULT NULL,
203     `2Q2P_nota` decimal(10,2) DEFAULT NULL,
204     `2Q3P_nota` decimal(10,2) DEFAULT NULL,
205     `2QPR_nota` decimal(10,2) DEFAULT NULL,
206     `EXFN_nota` decimal(10,2) DEFAULT NULL,
207     `SUBT_nota` decimal(10,2) DEFAULT NULL,
208     `SUPL_nota` decimal(10,2) DEFAULT NULL,
209     `PRTL_nota` decimal(10,2) DEFAULT NULL,
210     `estado_nota` varchar(4) COLLATE latin1_spanish_ci DEFAULT NULL,
211     `id_alumno` int(10) NOT NULL,
212     `id_asignatura` int(10) NOT NULL,
213     PRIMARY KEY (`id_nota`),
214     KEY `fk_notas_alumnos1` (`id_alumno`),
215     KEY `fk_notas_asignaturas1` (`id_asignatura`)
216 ) ENGINE=InnoDB AUTO_INCREMENT=154 DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;
```

Figura 2.15 Código para crear la tabla notas

El código completo relacionado con la implementación de la base de datos se encuentra en el **ANEXO III**.

2.2.4 Implementación de las interfaces web

Para la implementación de las interfaces web se utilizó HTML para estructurar la interfaz web, CSS para dar estilos a los elementos en HTML y Bootstrap el cual es un framework de gran utilidad para la creación dichas interfaces.

En la Figura 2.16 se muestra el código HTML base para todos los archivos de las interfaces web que conforman el sistema prototipo, en las líneas 9, 15 y 16 se muestra el código para agregar las referencias necesarias y principales para el uso del framework Bootstrap. Adicionalmente, en las líneas 8 y 14 se muestran el código para incluir la biblioteca de JQuery, la cual es utilizada para realizar llamadas AJAX.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title></title>
7
8   <script src="https://code.jquery.com/jquery-3.3.1.js" integrity="sha256-2Kok7Mb0yxpGUvVak/
9   HJ2jig0SYS2auK4Pfz7uH60=" crossorigin="anonymous"></script>
10  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet
11  " integrity="sha384-0evHe/X+R7YkIZDRvuzKMRqM+OrBnVFB6D0itfPri4tjfHxaWutUpFmBp4vmVor" crossorigin="
12  anonymous">
13 </head>
14 <body>
15   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="
16   sha384-wHAiFfrR1MFy6i5SRaxvf0CifBUQy1xHdJ/yoi7FRNXMRBu5WHdZyu1hA6Z0blgut" crossorigin="anonymous"></script>
17   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/js/bootstrap.bundle.min.js" integrity
18   ="sha384-pprn3073KE6tl6bjs2QrFaJGz5/SUsLqktiwsUTF55Jfv3qYSDhgCecCxMW52nD2" crossorigin="anonymous"></
19   script>
20   <script src="js/bootstrap.js"></script>
21 </body>
22 </html>

```

Figura 2.16 Código HTML base

A continuación, se muestran algunas interfaces web del sistema prototipo.

En la Figura 2.17 se muestra la interfaz web de login para el acceso al sistema prototipo, donde se puede apreciar un combo box para seleccionar el tipo de usuario, dos cajas de texto para ingresar el nombre de usuario y contraseña del usuario seleccionado y un botón para ingresar al sistema.

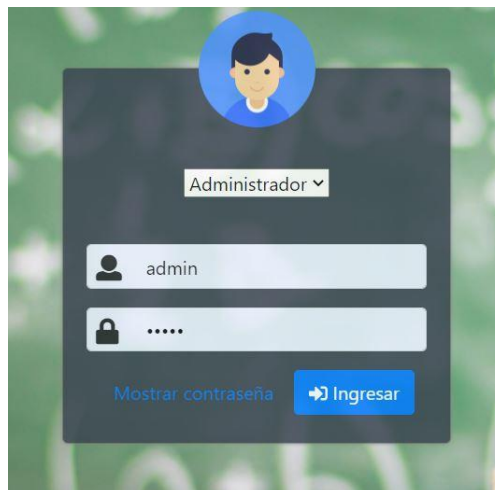


Figura 2.17 Login: acceso al sistema

En la Figura 2.18 se muestra la interfaz web para administrar profesores, con su respectiva cabecera, la cual presenta información del sistema, un botón para cerrar sesión y una barra de navegación con los diferentes módulos del sistema. Esta cabecera se repite para todas las interfaces web del usuario Administrador. Adicionalmente, se puede ver una lista de profesores, opciones de buscar, crear, editar y eliminar el usuario de tipo profesor.

Figura 2.18 Interfaz web para la administración de profesores

En la Figura 2.19 se muestra parte del código para crear la ventana modal del formulario para crear profesores.

```

89 <div class="modal fade" id="modalProfesor" tabindex="-1" aria-labelledby="exampleModallabel" aria-hidden
90 =true">
91 <div class="modal-dialog modal-lg">
92 <div class="modal-content">
93 <div class="modal-header text-center">
94 <h5 class="modal-title" id="exampleModallabel"></h5>
95 <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
96 </div>
97 <div class="modal-body">
98 <form>
99 <div class="mb-1">
100 <div class="row">
101 <div class="col-3-ya">
102 <label for="txtIdProfesor" class="col-form-label">ID:</label>
103 <input type="text" class="form-control" readonly="readonly" id="txtIdProfesor" name="
104 txtIdProfesor">
105 </div>
106 <div class="col-9-ya">
107 </div>
108 </div>
109 <div class="mb-1">
110 <div class="row">
111 <div class="col-6-ya">
112 <label for="txtApellidoProfesor" class="col-form-label">Apellidos:</label>
113 <input type="text" class="form-control" id="txtApellidoProfesor" name="
114 txtApellidoProfesor" placeholder="Ingrese apellidos completos" style="text-transform:
115 uppercase;" onkeypress="return soloLetras(event);">
116 </div>
117 <div class="col-6-ya">
118 <label for="txtNombreProfesor" class="col-form-label">Nombres:</label>
119 <input type="text" class="form-control" id="txtNombreProfesor" name="txtNombreProfesor"
120 placeholder="Ingrese nombres completos" style="text-transform:uppercase;" onkeypress="
121 return soloLetras(event);">
122 </div>
123 </div>
124 </div>
125 </div>
126 </div>
127 </div>

```

Figura 2.19 Código HTML de la ventana modal para crear profesores

En la Figura 2.20 se muestra la ventana modal para la creación de los usuarios profesores donde se puede apreciar un formulario para ingresar la información de los usuarios, un botón para cerrar la ventana modal y un botón para guardar dicha información.

Crear Profesor ✕

ID:

Apellidos: Nombres:

Cédula: Dirección:

Teléfono: E-mail:

Usuario: Contraseña: [Mostrar contraseña](#)

La contraseña debe contener min 8 caracteres, max 15 caracteres, al menos una letra mayúscula, al menos una letra minúscula, al menos un dígito, al menos un caracter especial

Figura 2.20 Ventana modal: crear alumno

En la Figura 2.21 se muestra la interfaz web para asignar los profesores a las asignaturas, donde se puede ver una lista de profesores a la cual se puede buscar y seleccionar al profesor, una lista de asignaturas a la cual se la puede buscar y seleccionar una asignatura, un botón para asignar el profesor seleccionado a la asignatura seleccionada y un botón para cancelar dicha asignación.

Año Lectivo:
2020-2021

Usuario: (Administrador)
PEREZ GURRERO JOSE LUIS

Quito, 11 de agosto de 2022

ASIGNAR PROFESOR A ASIGNATURA

Nota: Debe seleccionar el profesor y la asignatura que va a impartir

Profesor Seleccionado:
 Asignatura Seleccionada:

Buscar:
 Buscar:

Lista de profesores

| Apellidos | Nombres | Cedulas | Acciones |
|-------------------|------------------|------------|-----------------------------|
| ANDRADE FARINANGO | MATIAS FERNANDO | 1716151223 | Seleccionar |
| AREVALO DIAZ | CRISTIAN FABIAN | 1716452332 | Seleccionar |
| BENAVIDES CRUZ | ELIANA ESTEFANIA | 1234222331 | Seleccionar |

Lista de asignaturas

| ID | Curso | Nombre | Acciones |
|-----|-------|------------------------------|-----------------------------|
| 120 | M03A | EDUCACION FISICA | Seleccionar |
| 119 | M03A | INGLES | Seleccionar |
| 67 | M12A | EDUCACION PARA LA CIUDADANIA | Seleccionar |

Figura 2.21 Interfaz web: asignar profesores a asignaturas.

Para el perfil del tutor se tiene la interfaz web para la administración de las observaciones académicas particulares, donde se puede observar una barra de navegación, una lista de

observaciones académicas y opciones de búsqueda, creación, edición y eliminación de las observaciones, como se muestra en la Figura 2.22.

| ID | Nombre | Fecha | Accion |
|----|----------------|------------|---|
| 15 | reunión 1 | 2022-08-07 | Editar Eliminar |
| 16 | Comportamiento | 2022-08-16 | Editar Eliminar |

Figura 2.22 Interfaz web para la administración de observaciones académicas particulares

Finalmente, en la Figura 2.23 se muestra para el perfil del estudiante (alumno) la interfaz web para consultar o ver las notas de las asignaturas que toma el estudiante, donde se muestra en la parte superior su respectiva cabecera y en la parte inferior la lista de las asignaturas que toma el estudiante con sus respectivas notas, adicionalmente, se tiene la opción de seleccionar años lectivos anteriores para poder ver las notas de esos años lectivos.

| Materias | 1Q 1er Parcial | 1Q 2do Parcial | 1Q 3er Parcial | 1er Quimestre | 2Q 1er Parcial | 2Q 2do Parcial | 2Q 3er Parcial | 2do Quimestre | Examen final | Promedio Quimestral | Supletorio | Nota final | Estado |
|--------------------------------|----------------|----------------|----------------|---------------|----------------|----------------|----------------|---------------|--------------|---------------------|------------|------------|--------|
| MATEMATICA | 9.00 | 5.00 | 8.00 | 7.33 | 7.00 | 9.00 | 7.00 | 7.67 | 10.00 | 8.00 | | 8.00 | AP |
| LENGUA Y LITERATURA | 6.00 | 7.00 | 9.00 | 7.33 | 9.00 | 10.00 | 9.00 | 9.33 | 8.00 | 8.26 | | 8.26 | AP |
| ESTUDIOS SOCIALES | 8.78 | 8.00 | 10.00 | 8.93 | 7.50 | 9.00 | 9.00 | 8.50 | 8.00 | 8.57 | | 8.57 | AP |
| CIENCIAS NATURALES | 5.00 | 7.00 | 9.00 | 7.00 | 9.00 | 10.00 | 8.00 | 9.00 | 9.00 | 8.20 | | 8.20 | AP |
| EDUCACION CULTURAL Y ARTISTICA | 7.00 | 6.00 | 8.70 | 7.23 | 6.00 | 10.00 | 7.00 | 7.67 | 10.00 | 7.96 | | 7.96 | AP |
| EDUCACION FISICA | 6.00 | 10.00 | 4.00 | 6.67 | 10.00 | 9.00 | 6.80 | 8.60 | 8.00 | 7.71 | | 7.71 | AP |

Figura 2.23 Interfaz web: Consulta de notas del estudiante

Las interfaces completas relacionado con la implementación de las interfaces web se encuentra en el **ANEXO IV**.

2.2.5 Implementación de la lógica de negocio

Para la implementación de la lógica de negocio se utilizó la aplicación SublimeText. Donde se acogió a PHP como el lenguaje de programación para el desarrollo de la presente aplicación, JavaScript para las validaciones y manejar los eventos de los elementos en HTML, AJAX de JQuery para la comunicación cliente-servidor y consumir el servicio web Restful.

A continuación, se muestran imágenes con partes del código utilizado en la implementación de la lógica de negocio, tanto en la parte del Cliente como la del Servidor para el desarrollo del presente sistema prototipo.

En la Figura 2.24 se muestra parte del código en el cliente para el acceso al sistema, donde se muestra la función o método en JavaScript que se encarga de enviar las credenciales del usuario al servidor, adicionalmente, se realiza la validación para verificar que los campos estén llenos y el uso del método AJAX para la comunicación con el servidor.

```
132 function validar(){
133     //se obtiene los valores del formulario para ingresar al sistema
134     opciones=document.getElementById("listaIndex");
135     mostrar = opciones.options[opciones.selectedIndex].value;
136     usuario = document.getElementById("txtNombreUsuario").value;
137     contrasenia = document.getElementById("txtContraseniaUsuario").value;
138     //Se valida si los campos estan vacios
139     if (usuario == '' || contrasenia == '' || mostrar == 'Seleccione') {
140         alert('Debe llenar todos los campos');
141     }
142     else{
143         //se crea un array con los datos para ingresar al sistema
144         datos = {tipo:"validarUsuario",
145                 datosUsuario:{tablaUsuario:mostrar,
146                             usuario:usuario,
147                             contrasenia:contrasenia,
148                             }};
149         console.log(datos);
150         //El metodo AJAX se utiliza para el intercambio de datos entre el cliente y el servidor
151         // y consumir el servicio web Restful
152         $.ajax({
153             //Se especifica la URL para la peticion
154             url: "http://localhost:8080/Sistema13JunioDB(11Septiembre)Prueba/Servidor/index.php",
155             //Se especifica que se utiliza el metodo POST para la peticion
156             type: "post",
157             //Especifica el tipo de informacion que se espera de respuesta, en este caso JSON
158             dataType: 'json',
159             //Envía el encabezado http json al navegador para informarle qué tipo de datos se espera
160             contentType: 'application/json',
161             //Convierte el objeto datos de JavaScript a una cadena de texto JSON para ser enviados
162             data: JSON.stringify(datos),
163             //Se ejecuta la funcion si la peticion es satisfactoria
164             success: function (response) {
165                 console.log(response);
166                 //Si el servidor responde con un 0 quiere decir los valores estaban incorrectos
167                 if (response == 0) {
168                     alert("Mensaje de error \nUsuario o contraseña incorrectas");
169                 }
170             }
171             //Si el servidor responde con un valor distinto de 0 se precesa los datos segun convenga
172             else{
173                 for (let valor of response){
174                     id = valor.id;
175                     usuario = valor.usuario;
176                     apellidos = valor.apellidos;
177                     nombres = valor.nombres;
```

Figura 2.24 Código en el cliente: Acceso al sistema

En la Figura 2.25 se muestra parte del código en el servidor del archivo *index.php*, en donde se reciben los datos que envía el cliente para ser decodificados ya que vienen en formato JSON, posterior a eso se identifica la tarea que se desea realizar (tipo) y el método utilizado, en este caso POST. Adicionalmente se muestran todas las funciones o métodos que se tienen en el servidor. Dependiendo del tipo que se recibe este buscará y ejecutará la función deseada.

```
2 //Incluye el archivo consultas.php a este archivo
3 require('consultas.php');
4 //Se decodifica los datos enviados desde el cliente
5 $datos = json_decode(file_get_contents('php://input'), true);
6 //Se recupera y guarda el tipo de metodo que se utilizó
7 $metodo = $_SERVER['REQUEST_METHOD'];
8 //Se recupera el tipo, el cual se refiere a la función que se desea realizar
9 $tipo = $datos['tipo'];
10 //Se ejecuta el código si el método es POST
11 if($metodo == 'POST'){
12     //Se crea un indice para todos los metodos que se encuentran en el archivo consulta.php
13     //Donde se buscara el metodo que se desea ejecutar
14     switch ($tipo){
15
16         case "validarUsuario":
17             consultas::validarUsuario($datos['datosUsuario']);
18             break;
19
20         case "consultaAniosLectivos":
21             consultas::consultaAniosLectivos();
22             break;
23         case "insertarAnioLectivo":
24             consultas::insertarAnioLectivo($datos['datosAnioLectivo']);
25             break;
26         case "actualizarAnioLectivo":
27             consultas::actualizarAnioLectivo($datos['datosAnioLectivo']);
28             break;
29         case "eliminarAnioLectivo":
30             consultas::eliminarAnioLectivo($datos['idAnioLectivo']);
31             break;
32         case "buscarAnioLectivo":
33             consultas::buscarAnioLectivo($datos['anioInicio']);
34             break;
35         case "activarAnioLectivo":
36             consultas::activarAnioLectivo($datos['id_anioLectivo']);
37             break;
```

Figura 2.25 Código en el Servidor: archivo *index.php*

En la Figura 2.26 se muestra parte del código en el servidor, donde se validan las credenciales de los usuarios para acceder al sistema, en la cual se muestra la función o método en donde se realiza la conexión a la base de datos, se recuperan los datos recibidos, y por medio de una consulta a la base de datos se comprueba si estas credenciales son iguales a las de la base de datos, si coinciden se envía información del usuario para acceder al sistema, caso contrario se envía un valor de 0 para indicar que las credenciales son incorrectas.

```

9 public static function validarUsuarioDAO($datosUsuario){
10 //Se realiza la conexion a la base de datos
11 $db = new ConexionAux();
12 $conexion = $db->conectar();
13 //Se recuperan los datos enviados desde el cliente
14 $tablasUsuario = $datosUsuario["tablaUsuario"];
15 $usuario = $datosUsuario["usuario"];
16 $contrasenia = $datosUsuario["contrasenia"];
17 //Si el perfil es de tipo administrador se ejecuta el siguiente codigo
18 if ($tablasUsuario == "administradores") {
19 //Se realiza la consulta para comparar si la credenciales enviadas desde el cliente son iguales a
20 //los de la base de datos
21 $consulta = $conexion->prepare("SELECT * FROM $tablasUsuario WHERE usuario_administrador = ? AND
22 //contrasenia_administrador = ? ");
23 $resultado_aux = array();
24 $resultado = $consulta->execute(array($usuario, $contrasenia));
25 //Si las credenciales son iguales se envia la informacion del usuario
26 if ($consulta->rowCount() > 0) {
27 $resultado2 = $consulta->fetch(PDO::FETCH_ASSOC);
28 $administrador = new Usuario();
29 $administrador->setId($resultado2["id_administrador"]);
30 $administrador->setApellidos($resultado2["apellidos_administrador"]);
31 $administrador->setNombres($resultado2["nombres_administrador"]);
32 $administrador->setUser($resultado2["usuario_administrador"]);
33 array_push($resultado_aux, $administrador);
34 return $resultado_aux;
35 }
36 //Si las credenciales son distintas se envia un valor de 0
37 else{
38 $rsp = 0;
39 return $rsp;
40 }
41 $conexion = null;
42 }

```

Figura 2.26 Código en el servidor: Validar usuario

En la Figura 2.27 se muestra el código en el lado del servidor para la conexión a la base de datos. Se puede apreciar un conjunto de variables que serán necesaria para la conexión. Adicionalmente se utiliza PDO (Objetos de Datos de PHP), la cual es una extensión de PHP de mucha utilidad, ya que ofrece una capa de abstracción para el acceso a los datos independientemente de la base de datos que se utilice [29].

```

5 //Se crean las variables necesarias para la conexion a la base de datos
6 private $hostname = "localhost";
7 private $database = "sistemaprototipo";
8 private $username = "root";
9 private $password = "";
10 private $charset = "utf8";
11 //Se crea el metodo conectar para relaizar la coexion a la base de datos
12 public function conectar(){
13 try {
14 //Se crea un arreglo con algunas variables anteriormente creados
15 $conexion = "mysql:host=".$this->hostname.";dbname=".$this->database.";charset=".$this->charset;
16 //Se crea al arreglo option, donde estan algunas opciones que posee PDO
17 $option = [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION,
18 PDO::ATTR_EMULATE_PREPARES => false,];
19 //Se crea un instancia con las variables anteriores para la conexion
20 $pdo = new PDO($conexion, $this->username, $this->password, $option);
21 //Se retorna la conexion
22 return $pdo;
23 }
24 //Muestra un mensaje de error, si se produce algún error
25 catch (PDOException $e) {
26 echo 'Error conexion'. $e->getMessage();
27 exit;
28 }
29 }

```

Figura 2.27 Código en el Servidor: Conexión a la base de datos

En la Figura 2.28 se muestra parte del código en el lado del cliente para crear o insertar un usuario de tipo profesor en la base de datos, en la cual se puede ver la estructura para enviar la información del usuario Profesor al servidor utilizando en método AJAX de JQuery, una vez que el Servidor procese los datos responderá con 1, que quiere decir que el dato de la cédula ya existe, en el caso que el valor sea 0, significará que el nombre de usuario ya existe y en el caso que el valor sea 2, quiere decir que el profesor fue insertado correctamente en la base de datos.

```

266     datos = {tipo:"insertarProfesor",
267             datosProfesor:{
268                 apellidos_profesor:apellidos,
269                 nombres_profesor:nombres,
270                 cedula_profesor:cedula,
271                 direccion_profesor:direccion,
272                 telefono_profesor:telefono,
273                 email_profesor:email,
274                 usuario_profesor:usuario,
275                 contrasenia_profesor:contrasenia,
276             }};
277     console.Log(datos);
278
279     $.ajax({
280         url: "http://localhost:8080/Sistema13JunioDB(11Septiembre)Prueba/Servidor/index.php",
281         type: "post",
282         dataType: 'json',
283         contentType: 'application/json',
284         data: JSON.stringify(datos),
285         success: function (response) {
286             console.Log(response);
287             if (response.confirmacion == 1) {
288                 alert("El usuario con cedula: "+cedula+" ya existe");
289             }
290             else if(response.confirmacion == 0){
291                 alert("El nombre de usuario ya existe elija otro");
292             }
293             else{
294                 alert(response.confirmacion);
295                 cargarlista();
296                 limpiarCampo();
297             }
298         },
299         error: function(jqXHR, textStatus, errorThrown) {
300             //console.log(textStatus, errorThrown);
301         }
302     });
303

```

Figura 2.28 Código en el cliente. Insertar profesor

En la Figura 2.29 se muestra el código en el lado del servidor para insertar el usuario Profesor a la base de datos, en primer lugar, se puede ver el código para la conexión a la base de datos, a continuación, se tienen dos consultas para verificar que la cédula y el nombre de usuario del profesor que se recibe sean distintos a los de los profesores que ya existen en la base de datos y una sentencia para ingresar la información del profesor a la base de datos, dependiendo de los resultado que se obtiene de las consultas o la sentencia este responderá con algún valor el cual será interpretado por el cliente (Navegador).

```

23 public function insertarProfesorDAO($datosProfesor){
24     //Se conecta a la base de datos
25     $db = new conexionAux();
26     $conexion = $db->conectar();
27     //Se crea el objeto profesor
28     $profesor = new Profesor($datosProfesor["apellidos_profesor"], $datosProfesor["nombres_profesor"], $
        datosProfesor["cedula_profesor"], $datosProfesor["direccion_profesor"], $datosProfesor["
        telefono_profesor"], $datosProfesor["email_profesor"], $datosProfesor["usuario_profesor"], $
        datosProfesor["contrasenia_profesor"]);
29     $cedulaProfe = $profesor->getCedula();
30     $usuarioProfe = $profesor->getUser();
31     //Se realiza una consulta a la BD para comprobar si ya existe un profesor con en N° de cedula ingresado
32     $consulta = $conexion->prepare("SELECT * FROM profesores WHERE cedula_profesor = '$cedulaProfe'");
33     $resultado = $consulta->execute();
34     if ($consulta->rowCount() > 0) {
35         $respuesta=1;
36         return $respuesta;
37     }
38     else{
39         //Se realiza una consulta a la BD para comprobar si ya existe un profesor con el nombre de usuario
        ingresado.
40         $consulta2 = $conexion->prepare("SELECT * FROM profesores WHERE usuario_profesor = '$usuarioProfe'");
41         $consulta2->execute();
42         if ($consulta2->rowCount() > 0) {
43             $respuesta=0;
44             return $respuesta;
45         }
46         else{
47             //Se prepara la consulta para ingresar la información del profesor a la base de datos
48             $sentencia = $conexion->prepare("INSERT INTO profesores (apellidos_profesor, nombres_profesor,
        cedula_profesor, direccion_profesor, telefono_profesor, email_profesor, usuario_profesor,
        contrasenia_profesor) VALUES (:apellidos, :nombres, :cedula, :direccion, :telefono, :email,
        :usuario, :contrasenia)");
49             //Se ejecuta la consulta anterior y se ingresa la inforacion del profesor
50             $resultado2 = $sentencia->execute(array('apellidos'=>$profesor->getApellidos(), 'nombres'=>$
        profesor->getNombres(), 'cedula'=>$profesor->getCedula(), 'direccion'=>$profesor->
        getDireccion(), 'telefono'=>$profesor->getTelefono(), 'email'=>$profesor->getEmail(), '
        usuario'=>$profesor->getUser(), 'contrasenia'=>$profesor->getContrasenia()));
51             //Si se ejecuta la consulta el servidor envia un valor de 2 al cliente
52             if ($resultado2) {
53                 $respuesta=2;
54                 return $respuesta;
55             }
56         }
57     }
58 }

```

Figura 2.29 Código en el servidor: crear usuario profesor

En la Figura 2.30 se muestra el código de la cabecera para la generación del certificado de promoción, donde se puede observar el código para insertar imágenes, cambiar el tipo de letra e insertar párrafos. Para la generación de certificados en PDF se utiliza la biblioteca FPDF escrita en PHP, para lo cual se crea una clase PDF y se hereda de la clase FPDF para utilizar sus métodos.

```

114 function Header()
115 {
116     // Se inserta imagenes, donde se utiliza la ruta y las posiciones
117     $this->Image('Static/Imagenes/escudo.jpg',25,18,30);
118     $this->Image('Static/Imagenes/MinEducacion.jpg',160,8,33,38);
119     //Se cambian el tipo de letra Arial bold 15
120     $this->AddFont('CALIBRIB');
121     $this->SetFont('CALIBRIB', '',12);
122     //Se inserta parrafos en cada celda (Cell) con su respetivo salto de linea (Ln)
123     $this->Cell(190,8,utf8_decode('COORDINACIÓN ZONAL 2'),0,0,'C');
124     $this->Ln(6);
125     $this->Cell(190,8,utf8_decode('DISTRITO EDUCATIVO 17D11'),0,0,'C');
126     $this->Ln(12);
127     $this->Cell(190,8,utf8_decode('UNIDAD EDUCATIVA "SOCIEDAD EDUCATIVA FUTURO"'),0,0,'C');
128     $this->Ln(6);
129     $this->Cell(190,8,utf8_decode('CERTIFICADO DE PROMOCIÓN'),0,1,'C');
130 }

```

Figura 2.30 Código Cabecera del certificado de promoción

En la Figura 2.31 se muestra parte del código para la generación del certificado de promoción, donde se crea el objeto PDF para utilizar los métodos de la clase FPDF, también se traen los datos de los alumnos desde la base de datos y se crea o agrega una página en la cual se inserta la información necesaria para la generación del certificado.

```

222 //Se crea el objeto PDF que se extiende o hereda de FPDF
223 $pdf = new PDF();
224
225 $pdf->AliasNbPages();
226 //Se recupera los datos de los alumnos con sus notas desde la base de datos
227 $datos = consultasReportes::certificadoPromocion($id_curso);
228 foreach ($datos as $row) {
229 //Se agrega o se crea una página
230 $pdf->AddPage();
231 //Se cambia el tipo de letra
232 $pdf->AddFont('Calibri');
233 $pdf->AddFont('CALIBRIB');
234 //Se setea el margen izquierdo
235 $pdf->SetLeftMargin(15);
236 $pdf->SetFont('CALIBRIB','',12);
237 //Se inserta parrafos a la página
238 $pdf->Cell(190,6,utf8_decode('AÑO LECTIVO '.$aniolectivo),0,1,'C');
239 $pdf->Cell(190,5,utf8_decode('JORNADA MATUNINA'),0,1,'C');
240 // $pdf->Cell(10);
241 $pdf->SetFont('Calibri','',10);
242 $pdf->Ln(3);
243 $pdf->SetFont('Arial','',9);
244 $pdf->WriteHTML(utf8_decode("De conformidad con lo prescrito en el Art. 197 del Reglamento
    General a la Ley Orgánica de Educación Intercultural y ademas normativas vigentes, certifica
    que el/la estudiante <b>".$row['apellidos_alumno']." ".$row['nombres_alumno']."</b> del <b>".$
    letra_curso."</b> obtuvo las siguientes calificaciones durante el presente año lectivo:");
245
246 //-----Tabla (Títulos)-----
247
248 $pdf->Ln(9);
249 //Se crea la tabla para mostrar la informacion de las notas del alumno
250 $pdf->SetFont('CALIBRIB','',10);
251 $pdf->Cell(55,14,'AREAS',1,0,'L');
252 $pdf->Cell(55,14,'ASIGNATURAS',1,0,'L');
253 $pdf->Cell(70,7,'CALIFICACIONES',1,1,'C');
254 $pdf->Cell(110);
255 // $pdf->setY(-5);
256 $pdf->Cell(20,7,utf8_decode('NÚMERO'),1,0,'C');
257 $pdf->Cell(50,7,utf8_decode('LETRAS'),1,1,'C');
258 $pdf->SetFont('Calibri','',10);
259 $pdf->Cell(55,7,'LENGUA Y LITERATURA',1,0,'L');
260 $pdf->Cell(55,7,'LENGUA Y LITERATURA',1,0,'L');
261 $pdf->Cell(20,7,utf8_decode($row['PRTLLENG']),1,0,'C');
262 $pdf->Cell(50,7,utf8_decode($row['LETRLENG']),1,1,'L');

```

Figura 2.31 Código para generar el certificado de promoción

El código completo relacionado con la implementación de la lógica de negocio se encuentra en el **ANEXO V**.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentan los resultados obtenidos de las pruebas de funcionamiento realizadas al sistema prototipo, así como la conclusiones y recomendaciones para el presente Trabajo de Integración Curricular.

3.1 Tablero Kanban actualizado

Una vez finalizada la fase de implementación se procede a la actualización del tablero Kanban, donde se encuentran las tareas finales del presente Trabajo de Integración Curricular. En la Tabla 3.1 se muestra el tablero Kanban actualizado para las pruebas de funcionamiento y resultados.

Tabla 3.1 Tablero Kanban para pruebas de funcionamiento y resultados

| POR HACER | EN PROGRESO | FINALIZADO |
|------------------|--|--|
| | Pruebas de funcionamiento. | Estudio de la base teórica necesaria para el desarrollo del sistema prototipo. |
| | Presentación de los resultados obtenidos en las pruebas de funcionamiento. | Recolección de los requerimientos a los diferentes funcionarios de la unidad educativa a través de entrevistas |
| | | Clasificación de los requerimientos en funcionales y no funcionales. |
| | | Elaboración del diagrama de casos de uso |
| | | Elaboración del diagrama relacional |
| | | Diseño de interfaces web. |
| | | Elaboración del diagrama de clases |
| | | Implementación del diagrama relacional. |
| | | Implementación de las interfaces web. |
| | | Implementación de la lógica de negocio. |

3.2 Resultados

Para la presentación de los resultados se realizaron pruebas de funcionamiento con cuatro personas particulares, las cuales tomaron el rol de cada uno de los usuarios del sistema prototipo y probaron cada uno de los módulos de dicho sistema. A continuación, se presentan algunas de las pruebas de funcionamiento realizadas.

Para las pruebas de funcionamiento se tomaron como referencia las Historias de Usuario que se encuentran en el **Anexo I**.

La Historia de Usuario HU001 describe que los usuarios: administrador, profesor, tutor, estudiante y representante pueden acceder al sistema a través de un nombre de usuario y una contraseña. En la Figura 3.1 se muestra el formulario para ingresar al sistema en la cual los usuarios deben escoger el tipo de usuario e ingresar sus credenciales y dar clic en el botón Ingresar. En la Figura 3.2 se muestra un mensaje que dice “Bienvenido” el cual indica que sus credenciales con correctos.

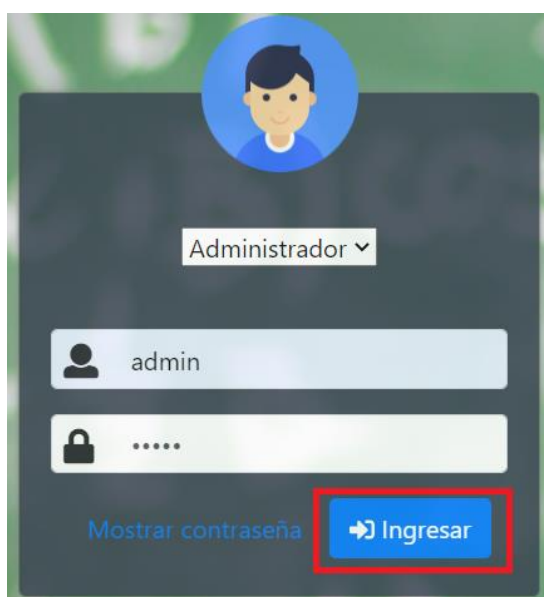


Figura 3.1 Login

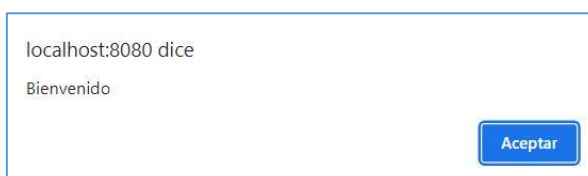


Figura 3.2 Alerta de Bienvenida al sistema

En la Figura 3.3 muestra la página principal a la que fue redirigido el usuario Administrador después de haber ingresado sus credenciales de manera correcta.

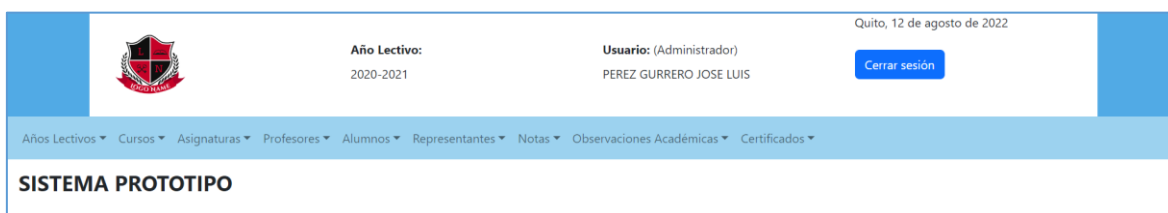


Figura 3.3 Inicio (Administrador)

La Historia de Usuario HU027 describe que el usuario Administrador puede registrar profesores. En la Figura 3.4 se muestra el formulario para crear nuevos profesores, en donde el usuario Administrador debe ingresar la información del usuario Profesor, tales

como: apellidos, nombres, número de cédula, dirección, número de teléfono, correo electrónico, nombre de usuario y contraseña, finalmente, presionar el botón “Guardar”.

Crear Profesor

ID:

Apellidos: AREVALO DIAZ

Nombres: CRISTIAN FABIAN

Cédula: 111452333

Dirección: Carapungo, calle xx

Teléfono: 0987654562

E-mail: arevalo@gmail.com

Usuario: arevalo

Contraseña: [Mostrar contraseña](#)
arevaloA1

La contraseña debe contener min 8 caracteres, max 15 caracteres, al menos una letra mayúscula, al menos una letra minúscula, al menos un dígito, al menos un caracter especial

Cerrar Guardar

Figura 3.4 Crear Profesor

En la Figura 3.5 se muestra un mensaje que indica que el usuario Profesor fue insertado correctamente; siempre y cuando el nombre de usuario y el número de cedula sean diferentes a los profesores que ya están registrados en la base de datos. Adicionalmente se debe presionar el botón “Aceptar” para cerrar la ventana del mensaje.

localhost:8080 dice

Profesor insertado correctamente

Aceptar

Figura 3.5 Mensaje de crear profesor

La Historia de Usuario HU029 describe que el usuario Administrador puede editar la información de los profesores.

En la Figura 3.6 se muestra el formulario para editar la información del usuario Profesor, en donde el usuario Administrador puede editar o actualizar la información de dicho usuario tales como: apellidos, nombres, número de cédula, dirección, número de teléfono, correo

electrónico, nombre de usuario y contraseña, finalmente se debe presionar el botón “Guardar” para guardar los cambios efectuados.

Editar Profesor [X]

ID: 45

Apellidos: AREVALO DÍAZ Nombres: CRISTIAN FABIÁN

Cédula: 171E42332 Dirección: Carapungo, calle xx

Teléfono: 0987654562 E-mail: arevalo85@gmail.com

Usuario: arevalo Contraseña: arevaloA1 [Mostrar contraseña](#)

La contraseña debe contener min 8 caracteres, max 15 caracteres, al menos una letra mayúscula, al menos una letra minúscula, al menos un dígito, al menos un caracter especial

Cerrar **Guardar**

Figura 3.6 Editar Profesor

En la Figura 3.7 se muestra un mensaje que indica que el usuario Profesor fue actualizado correctamente, al igual que para crear el usuario Profesor, el nombre de usuario y el número de cédula deben ser diferentes a los de la base de datos. Finalmente, se debe presionar el botón “Aceptar” para cerrar la ventana del mensaje.

localhost:8080 dice

Profesor actualizado correctamente

Aceptar

Figura 3.7 Mensaje de actualización del profesor

La Historia de Usuario HU030 describe que el usuario Administrador puede eliminar la información de los profesores. En la Figura 3.8 se muestra una lista de profesores con opciones de “Editar” y “Eliminar”, para este caso se da clic en la opción de “Eliminar” para eliminar el registro del usuario Profesor con toda su información.

| Id | Apellidos | Nombres | Cedula | Accion |
|----|-------------------|------------------|------------|---|
| 36 | ANDRADE FARINANGO | MATIAS FERNANDO | 1718151233 | Editar Eliminar |
| 41 | AREVALO DÍAZ | CRISTIAN FABIÁN | 1715143232 | Editar Eliminar |
| 40 | BENAVIDES CRUZ | ELIANA ESTEFANIA | 1234222111 | Editar Eliminar |

Figura 3.8 Lista de profesores

En la Figura 3.9 se muestra un mensaje que indica que el usuario Profesor fue eliminado correctamente.

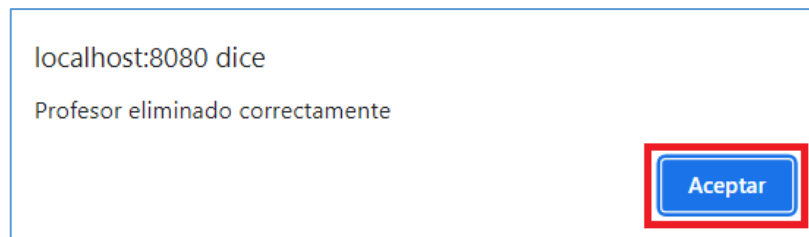


Figura 3.9 Mensaje de eliminación del profesor

En la Figura 3.10 se muestra la lista de profesores después de haber eliminado el usuario Profesor con ID 41.

| Id | Apellidos | Nombres | Cedula | Accion |
|----|-------------------|------------------|------------|---|
| 36 | ANDRADE FARINANGO | MATIAS FERNANDO | 1718151233 | Editar Eliminar |
| 40 | BENAVIDES CRUZ | ELIANA ESTEFANIA | 1234222111 | Editar Eliminar |

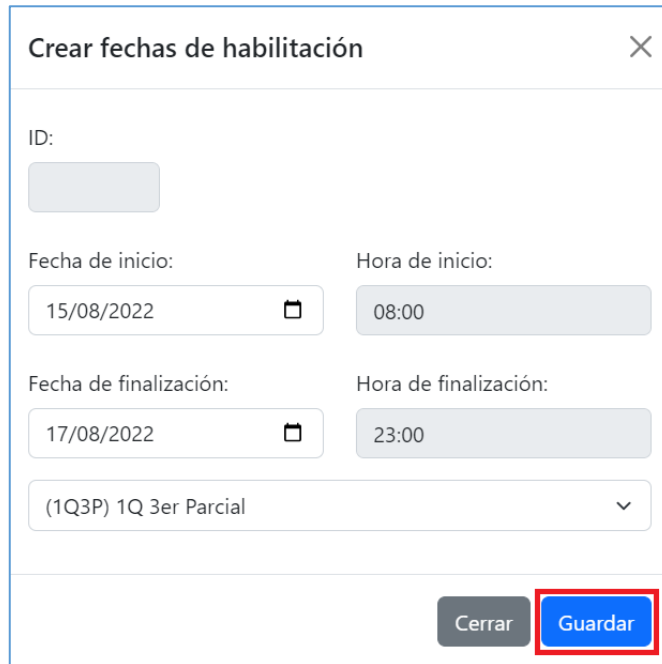
Figura 3.10 Lista de profesores después de eliminar un profesor

La Historia de Usuario HU031 escribe que el usuario Administrador puede buscar a un profesor utilizando los apellidos de este.

En la Figura 3.11 se muestra la búsqueda realizada por apellidos, para lo cual se ingresa a la caja de texto los apellidos del profesor y se presiona el botón de buscar.

Figura 3.11 Búsqueda de profesores

La Historia de Usuario HU070 describe que el usuario Administrador puede registrar fechas de habilitación para la carga de notas. En la Figura 3.12 se muestra el formulario para crear fechas de habilitación, en donde el usuario Administrador debe ingresar la información de estas, tales como: fecha de inicio, fecha de finalización, hora de inicio, hora de finalización y una descripción, finalmente se presiona el botón “Guardar”.



Crear fechas de habilitación

ID:

Fecha de inicio: 15/08/2022

Hora de inicio: 08:00

Fecha de finalización: 17/08/2022

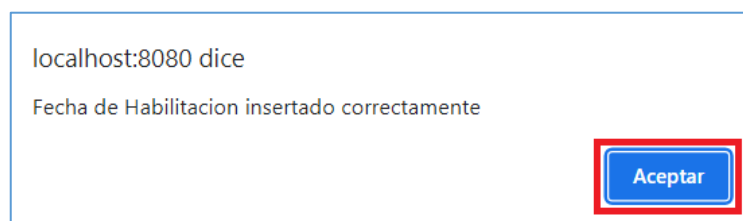
Hora de finalización: 23:00

(1Q3P) 1Q 3er Parcial

Cerrar Guardar

Figura 3.12 Crear fechas de habilitación

En la Figura 3.13 se muestra un mensaje que indica que las fechas de habilitación fueron insertadas correctamente.



localhost:8080 dice

Fecha de Habilitacion insertado correctamente

Aceptar

Figura 3.13 Mensaje de creación de fechas de habilitación

La Historia de Usuario HU099 describe que el usuario Profesor puede registrar las notas de los estudiantes de las diferentes asignaturas que imparte.

En la Figura 3.14 se muestra la interfaz principal para el registro de notas en la cual se muestra, las fechas en que se pueden cargar las notas, la asignatura, la descripción de la nota que se debe ingresar y una lista de los estudiantes que toman la asignatura mencionada anteriormente con la opción de “Ingresar nota”.

NOTAS

Usted puede ingresar las notas del: 1Q1P desde: 2022-8-21 8:0:00 hasta: 2022-8-23 23:0:00

Asignatura:

Ingrese las notas del:

Buscar:

Lista de Alumnos
ESTADO (AP = Aprobado, PA = Pierde el año.)

| Apellidos | Nombres | 1Q1P | 1Q2P | 1Q3P | 1Q | 2Q1P | 2Q2P | 2Q3P | 2Q | EXF | PQ | SU | PF | EN | Acciones |
|----------------|----------------|-------|-------|------|------|------|------|------|------|-------|------|-------|------|----|-------------------------------|
| ALUMNO PRIMERO | BASICA TRES | 10.00 | 10.00 | 7.00 | 9.00 | 5.00 | 4.00 | 6.00 | 5.00 | 10.00 | 7.60 | 0.00 | 7.60 | AP | Ingresar nota |
| ALUMNO PRIMERO | BASICA DOS | 2.00 | 4.00 | 8.00 | 4.67 | 7.00 | 6.00 | 7.00 | 6.67 | 9.00 | 6.34 | 10.00 | 8.17 | AP | Ingresar nota |
| ALUMNO PRIMERO | BASICA UNO | 9.00 | 5.00 | 8.00 | 7.33 | 7.00 | 9.00 | 7.00 | 7.67 | 10.00 | 8.00 | | 8.00 | AP | Ingresar nota |
| DIAZ MORENO | GUSTAVO ANDRES | | | | | | | | | | | | | | Ingresar nota |

Figura 3.14 Ingresar notas

En la Figura 3.15 se muestra un formulario para ingresar la nota del alumno, en la cual el usuario Profesor puede ver el nombre del alumno e ingresar la nota de este, finalmente se presiona el botón “Guardar” para cargar la nota.

X

Ingrese la nota

Alumno seleccionado:

Ingrese la calificación:

Figura 3.15 Formulario Ingresar nota

En la Figura 3.16 se muestra un mensaje donde se indica que la nota fue insertada correctamente.

localhost:8080 dice

Nota insertada correctamente

Figura 3.16 Mensaje de registro de notas

La Historia de Usuario HU114 describe que el usuario Estudiante puede ver sus notas de todas las asignaturas que toma.

En la Figura 3.17 se muestra una lista de todas las asignaturas con sus respectivas notas de cada parcial con sus promedios, las cuales pertenecen al usuario Estudiante.

| CALIFICACIONES | | | | | | | | | | | | | |
|--|----------------|----------------|----------------|---------------|----------------|----------------|----------------|---------------|--------------|---------------------|------------|------------|--------|
| ESTADO (AP = Aprobado, PA = Pierde el año.) | | | | | | | | | | | | | |
| Escoja un año lectivo: 2013-2014 | | | | | | | | | | | | | |
| Materias | 1Q 1er Parcial | 1Q 2do Parcial | 1Q 3er Parcial | 1er Quimestre | 2Q 1er Parcial | 2Q 2do Parcial | 2Q 3er Parcial | 2do Quimestre | Examen final | Promedio Quimestral | Supletorio | Nota final | Estado |
| MATEMATICA | 9.00 | 5.00 | 8.00 | 7.33 | 7.00 | 9.00 | 7.00 | 7.67 | 10.00 | 8.00 | | 8.00 | AP |
| LENGUA Y LITERATURA | 6.00 | 7.00 | 9.00 | 7.33 | 9.00 | 10.00 | 9.00 | 9.33 | 8.00 | 8.26 | | 8.26 | AP |
| ESTUDIOS SOCIALES | 8.78 | 8.00 | 10.00 | 8.93 | 7.50 | 9.00 | 9.00 | 8.50 | 8.00 | 8.57 | | 8.57 | AP |
| CIENCIAS NATURALES | 5.00 | 7.00 | 9.00 | 7.00 | 9.00 | 10.00 | 8.00 | 9.00 | 9.00 | 8.20 | | 8.20 | AP |
| EDUCACION CULTURAL Y ARTISTICA | 7.00 | 6.00 | 8.70 | 7.23 | 6.00 | 10.00 | 7.00 | 7.67 | 10.00 | 7.96 | | 7.96 | AP |
| EDUCACION FISICA | 6.00 | 10.00 | 4.00 | 6.67 | 10.00 | 9.00 | 6.80 | 8.60 | 8.00 | 7.71 | | 7.71 | AP |
| INGLES | 8.70 | 10.00 | 9.00 | 9.23 | 8.00 | 7.50 | 8.00 | 7.83 | 6.00 | 8.02 | | 8.02 | AP |

Figura 3.17 Notas del estudiante

La Historia de Usuario HU087 describe que el usuario Administrador puede registrar observaciones académicas generales.

En la Figura 3.18 se muestra el formulario para crear nuevas observaciones académicas, en donde el usuario Administrador debe ingresar información, tales como: nombre de la observación y una descripción, adicionalmente, se presiona el botón “Guardar”.

Crear Observación Académica
✕

ID:

Nombre: Fecha:

Descripción:

Cerrar
Guardar

Figura 3.18 Crear observación académica

En la Figura 3.19 se muestra un mensaje que indica que la observación académica general fue insertada correctamente.

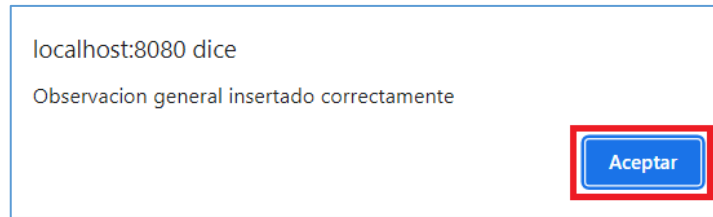


Figura 3.19 Mensaje de creación de la observación académica

La Historia de Usuario HU088 describe que el usuario Administrador puede editar o actualizar la información de las observaciones académicas generales.

En la Figura 3.20 se muestra el formulario para editar la información de las observaciones académicas, en donde el usuario Administrador puede editar la información de las observaciones, tales como: nombre de la observación y descripción, adicionalmente se presiona el botón “Guardar” para guardar los cambios efectuados.

Figura 3.20 Editar observación académica general

En la Figura 3.21 se muestra un mensaje que indica que la observación académica general fue actualizada correctamente

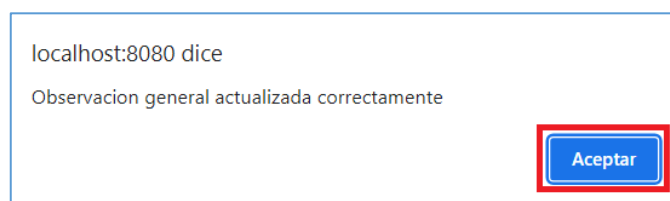


Figura 3.21 Mensaje de actualización de la observación general

La Historia de Usuario HU089 describe que el usuario Administrador puede eliminar la información de las observaciones académicas generales.

En la Figura 3.22 se muestra una lista de observaciones académicas generales con opciones de “Editar” y “Eliminar”, para este caso se da clic en la opción de “Eliminar” para eliminar el registro de la observación académica.

| IDs | Nombre | Fecha | Acción |
|-----|------------------------|------------|---|
| 19 | Suspensión de clases | 2022-08-15 | Editar Eliminar |
| 20 | Reunión representantes | 2022-08-15 | Editar Eliminar |
| 8 | Reunion | 2022-07-06 | Editar Eliminar |

Figura 3.22 Lista de observaciones académicas generales

En la Figura 3.23 se muestra un mensaje que indica que la observación académica general fue eliminada correctamente.

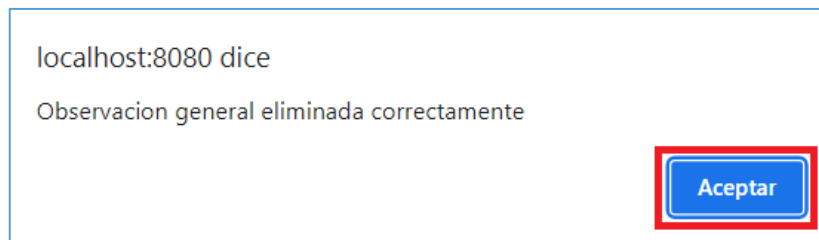


Figura 3.23 Mensaje de eliminación de observaciones generales

En la Figura 3.24 se muestra la lista de las observaciones académicas generales después de haber eliminado la observación con ID 22.

| IDs | Nombre | Fecha | Acción |
|-----|----------------------|------------|---|
| 19 | Suspensión de clases | 2022-08-15 | Editar Eliminar |
| 8 | Reunion | 2022-07-06 | Editar Eliminar |

Figura 3.24 Lista de observaciones académicas generales después de la eliminación

La Historia de Usuario HU090 describe que el usuario Administrador puede buscar una observación académica general utilizando la fecha de la observación.

En la Figura 3.25 se muestra la búsqueda realizada por fechas, para lo cual se ingresa a la caja de texto la fecha de la observación y se presiona el botón de buscar.

Buscar:

Lista de observaciones generales

| IDs | Nombre | Fecha | Acción |
|-----|---------|------------|---|
| 8 | Reunion | 2022-07-06 | Editar Eliminar |

Figura 3.25 Búsqueda de observaciones académicas generales

La Historia de Usuario HU093 describe que el usuario Administrador puede generar cuadro de calificaciones del primer quimestre, segundo quimestre, finales y certificados de promoción.

En la Figura 3.26 se muestra la página principal donde se generan los diferentes certificados académicos para lo cual primero se debe seleccionar el tipo de certificado en la barra de navegación, después se selecciona un curso y por último se presiona el botón de “Generar certificado”.

Quito, 21 de agosto de 2022

Año Lectivo: 2020-2021 Usuario: (Administrador) PEREZ GURRERO JOSE LUIS

Años Lectivos ▾ Cursos ▾ Asignaturas ▾ Profesores ▾ Alumnos ▾ Representantes ▾ Notas ▾ Observaciones Académicas ▾ **Certificados ▾**

CUADRO DE CALIFICACIONES PRIMER QUI

Usted a escogido el curso: Buscar:

-
-
-
-

Lista de cursos

| IDs | Codigo | Curso | Paralelo | Acciones |
|-----|--------|---------------|----------|--|
| 30 | M01A | 1RO DE BASICA | A | <input type="button" value="Seleccionar"/> |
| 31 | M02A | 2DO DE BASICA | A | Seleccionar |
| 54 | M03A | 3RO DE BÁSICA | A | Seleccionar |
| 33 | M04A | 4TO DE BASICA | A | Seleccionar |
| 34 | M05A | 5TO DE BASICA | A | Seleccionar |

Figura 3.26 Generar certificados académicos

En la Figura 3.27 se muestra el cuadro de calificaciones del primer quimestre, que pertenecen al primero de básica.

| UNIDAD EDUCATIVA "SOCIEDAD EDUCATIVA FUTURO" | | | | | | | | | | |
|---|----------------------------|---------------------|------------|--------------------|-------------------|--------------------------------|------------------|--------|----------------|---------------|
| CUADRO DE CALIFICACIONES DEL PRIMER QUIMESTRE | | | | | | | | | | |
| AÑO LECTIVO 2020-2021 | | | | | | | | | | |
| PRIMER AÑO DE EDUCACIÓN GENERAL BÁSICA | | | | | | | | | | |
| JORNADA MATUTINA | | | | | | | | | | |
| ORD. | APPELLIDOS Y NOMBRES | LENGUA Y LITERATURA | MATEMÁTICA | CIENCIAS NATURALES | ESTUDIOS SOCIALES | EDUCACIÓN CULTURAL Y ARTÍSTICA | EDUCACIÓN FÍSICA | INGLÉS | COMPORTAMIENTO | OBSERVACIONES |
| 1 | ALUMNO PRIMERO BASICA DOS | 5.00 | 4.67 | 6.67 | 7.89 | 7.93 | 9.00 | 8.23 | | |
| 2 | ALUMNO PRIMERO BASICA TRES | 6.33 | 9.00 | 6.33 | 8.55 | 7.29 | 8.67 | 5.67 | | |
| 3 | ALUMNO PRIMERO BASICA UNO | 7.33 | 7.33 | 7.00 | 8.93 | 7.23 | 6.67 | 9.23 | A | |
| 4 | DÍAZ MORENO GUSTAVO ANDRES | | 4.80 | 4.43 | 3.33 | 3.33 | | | A | algo |

LCDA. TATIANA COSTA M
RECTORA

SRA. DORYS GORDILLO V.
SECRETARIA

Figura 3.27 Cuadro de calificaciones del primer quimestre

En la Figura 3.28 se muestra el certificado de promoción, que pertenece a un estudiante de primero de básica.

| COORDINACIÓN ZONAL 2 DISTRITO EDUCATIVO 17D11 | | | |
|---|--------------------------------|----------------|--------------------------------|
| UNIDAD EDUCATIVA "SOCIEDAD EDUCATIVA FUTURO" | | | |
| CERTIFICADO DE PROMOCIÓN | | | |
| AÑO LECTIVO 2020-2021 | | | |
| JORNADA MATUTINA | | | |
| <p>De conformidad con lo prescrito en el Art. 197 del Reglamento General a la Ley Orgánica de Educación Intercultural y además normativas vigentes, certifica que el/la estudiante ALUMNO PRIMERO BASICA DOS del PRIMER GRADO DE EDUCACIÓN GENERAL BÁSICA obtuvo las siguientes calificaciones durante el presente año lectivo:</p> | | | |
| AREAS | ASIGNATURAS | CALIFICACIONES | |
| | | NÚMERO | LETRAS |
| LENGUA Y LITERATURA | LENGUA Y LITERATURA | 7,80 | Siete coma ochenta |
| MATEMÁTICA | MATEMÁTICA | 8,17 | Ocho coma diecisiete |
| CIENCIAS NATURALES | CIENCIAS NATURALES | 7,94 | Siete coma noventa y cuatro |
| CIENCIAS SOCIALES | ESTUDIOS SOCIALES | 8,48 | Ocho coma cuarenta y ocho |
| EDUCACIÓN CULTURAL Y ARTÍSTICA | EDUCACIÓN CULTURAL Y ARTÍSTICA | 7,77 | Siete coma setenta y siete |
| EDUCACIÓN FÍSICA | EDUCACIÓN FÍSICA | 8,80 | Ocho coma ochenta |
| LENGUA EXTRANJERA | INGLÉS | 8,68 | Ocho coma sesenta y ocho |
| PROMEDIO GENERAL | | 8,23 | Ocho coma veinte y tres |
| EVALUACIÓN DE PROYECTOS EDUCATIVOS | | | |
| EVALUACIÓN DEL COMPORTAMIENTO | | | |

Por lo tanto es promovido/a al **SEGUNDO GRADO DE EDUCACIÓN GENERAL BÁSICA** Para constancia suscriben en unidad de acto el/ la Rector/a con el/ la Secretario/a General del Plantel que certifica.

LCDA. TATIANA COSTA M
RECTORA

SRA. DORYS GORDILLO V.
SECRETARIA

Figura 3.28 Certificado de promoción

3.3 Correcciones Realizadas

Antes de realizar las pruebas de funcionamiento con los cuatro usuarios particulares hubieron ciertos problemas con el sistema prototipo, entre ellos se encontraron:

- En la parte de registro de notas, si el estudiante obtuviese un promedio total mayor a siete en los dos quimestres no permitía ingresar la nota del examen final, sin embargo, este punto ya fue solucionado.
- En la parte de la interfaz gráfica, al cambiar la resolución de la pantalla, la barra de navegación se distorsionaba y no permitía ver ciertas partes de las interfaces, sin embargo, este problema ya fue solucionado con la ayuda de Bootstrap.

Durante las pruebas de funcionamiento que se realizaron a las personas que tomaron los roles del sistema prototipo, se encontraron con algunos errores en las interfaces web, entre ellos se tienen:

- El nombre de algunas listas no coincidía con la lista, por ejemplo, en una lista de alumnos se tenía como título “lista de cursos”, estos errores ya fueron corregidos.
- Los mensajes de confirmación de algunas tareas no coincidían con las tareas que se realizaban, así mismo estos errores ya fueron corregidos.

3.4 Conclusiones

- Al culminar con el presente Trabajo de Integración Curricular se logró desarrollar el sistema prototipo, la cual es una aplicación web para la administración estudiantil de una unidad educativa, basado en los datos proporcionados por funcionarios de la unidad educativa “Sociedad Educativa Futuro”.
- Para el desarrollo del sistema prototipo se realizaron entrevistas a los diferentes funcionarios de la unidad educativa, los cuales ayudaron a entender cómo funciona un centro educativo e identificar a los usuarios que intervienen en la gestión académica, los usuarios identificados fueron: Administrador, Profesor, Tutor, Estudiante y Representante.
- Se adquirieron nuevos conocimientos sobre aplicaciones web, servicios web, metodologías ágiles entre otros, gracias a la investigación teóricas que se realizó para el presente Trabajo de Integración Curricular. Así también se adquirieron

nuevas habilidades para el diseño y desarrollo de software con la ayuda de ciertas tecnologías como: HTML, Bootstrap, PHP, MySQL entre otros.

- Los requerimientos funcionales permitieron definir cada uno de los módulos que conforman el sistema prototipo y a partir de esto se diseñaron, la base de datos, las interfaces web y la lógica de negocio.
- Para el desarrollo del sistema prototipo se utilizó la metodología Kanban en las fases de diseño e implementación, demostrando que es una herramienta de mucha utilidad ya que permite organizar de una manera eficiente y sencilla cada tarea para cumplir con el objetivo establecido.
- Las pruebas de funcionamiento que se realizaron a cuatro personas demostraron que los módulos del sistema prototipo funcionan de manera correcta y cumplen con los requerimientos obtenidos en la fase de diseño, es decir que se cumplió con el objetivo establecido.
- El sistema prototipo ayudará a mejorar la gestión académica ya que disminuye el tiempo en el proceso de notificar a los representantes sobre el desempeño académico de su o sus representados, así como, para generar los diferentes certificados académicos, también para el ingreso de las notas, ya que el profesor no tiene que realizar cálculos de forma manual sino solo debe ingresar la notas de los diferentes parciales.

3.5 Recomendaciones

- Es recomendable realizar una investigación previa de las herramientas tecnológicas con la finalidad de escoger las más apropiadas para el desarrollo del sistema prototipo en base a los requerimientos obtenidos.
- Se recomienda para un futuro aumentar funcionalidades como: matriculación, gestión de tareas, gestión de seguimientos y registro de asistencia, para tener una aplicación completa para la gestión académica de una unidad educativa.
- Se recomienda para un futuro crear una versión móvil para ciertas funcionalidades como la de notificar al representante el desempeño de sus representados, entre otros, y así cubrir las necesidades de los usuarios que poseen dispositivos móviles.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] “Software de gestion para centros de formacion y academias con cientos de herramientas y funciones - aGora ERP.” [En línea]. Disponible: <https://www.agora-erp.com/es/features> [Accedido: 30 Julio 2022].
- [2] “SOCIEDAD EDUCATIVA FUTURO en RUMIÑAHUI - Opiniones e información.” [En línea]. Disponible: <https://www.infoescuelas.com/ecuador/pichincha/sociedad-educativa-futuro-en-ruminahui/> [Accedido: 30 Julio 2022].
- [3] “¿Cómo Aplicar el Método Kanban? Características y Ventajas.” [En línea]. Disponible: <https://apiumhub.com/es/tech-blog-barcelona/metodo-kanban-ventajas/> [Accedido: 30 Julio 2022].
- [4] S. Lujan Mora, Programación de aplicaciones web: Historia, principios básicos y clientes web.
- [5] B. Roberto, DISEÑO E DESARROLLO WEB con CodeIgniter 3. [En línea]. Disponible: <https://books.google.com.ec/books?id=RZGWDwAAQBAJ&pg=PA9&dq=Desarrollo+web&hl=es&sa=X&ved=2ahUKEwjA1f7CmKT5AhW8TDABHUOCCs4Q6AF6BAgGEAI#v=onepage&q=Desarrollo%20web&f=false>. [Accedido: 30 Julio 2022]
- [6] “Arquitectura de Aplicaciones Web M2 | PDF.” [En línea]. Disponible: <https://es.scribd.com/doc/233561635/Arquitectura-de-Aplicaciones-Web-M2> [Accedido: 31 Julio 2022].
- [7] Samisa Abeysinghe, RESTful PHP Web Services. [En línea]. Disponible: https://books.google.com.ec/books?hl=es&lr=&id=jkIKNnLO104C&oi=fnd&pg=PT8&dq=restful+api+en+PHP&ots=g_vtJuJk-H&sig=s-GnTxd_Z5Ldh0ZS86P2J0leImU#v=onepage&q=restful%20api%20en%20PHP&f=false. [Accedido: 31 Julio 2022].
- [8] J. Alberto Guzmán-, E. Juan David Meza González, and E. Paola Andrea Galeano, “Implementación de la arquitectura PELEA por medio de servicios web semánticos para la manipulación remota de robots,” QUID: Investigación, Ciencia y Tecnología, ISSN-e 2462-9006, ISSN 1692-343X, N°. 18, 2012, págs. 25-33, vol. 18, no. 18, pp. 25–33, 2012. [En línea]. Disponible: <https://dialnet.unirioja.es/servlet/articulo?codigo=5235886&info=resumen&idioma=SPA>. [Accedido 31 Julio 2022]
- [9] I. Gabriel and E. Duarte Vega, “Arquitectura Propuesta para un Servicio Web Completo: Metodología de Desarrollo e Implementación,” 2016.
- [10] F. J. Cevallos Sierra, JAVA. Interfaces gráficas y aplicaciones para Internet. 4ª Edición, Editorial RA-MA. [En línea]. Disponible: <https://books.google.com.ec/books?id=A7E-DwAAQBAJ&printsec=frontcover&dq=Fco.+Javier+Ceballos+Sierra,+JAVA.+Interfases+gr%C3%A1ficas+y+aplicaciones+para+Internet.,+4ta+Edici%C3%B3n.&hl=es&sa=X&ved=2ahUKEwiigfWP7fP5AhVSnIQIHVG8Bt8Q6AF6BAgLEAI#v=onepage&q&f=false>. [Accedido: 31 Julio 2022]

- [11] X. Chen, Z. Ji, Y. Fan, and Y. Zhan, "Restful API Architecture Based on Laravel Framework," in *Journal of Physics: Conference Series*, Nov. 2017, vol. 910, no. 1. doi: 10.1088/1742-6596/910/1/012016.
- [12] A. Arcuri, "RESTful API Automated Test Case Generation," Jan. 2019, doi: 10.1109/QRS.2017.11.
- [13] "Metodología Kanban: en qué consiste y cómo utilizarla | APD." [En línea]. Disponible: <https://www.apd.es/metodologia-kanban/> [Accedido: 31 Julio 2022].
- [14] "¿Qué es la metodología Kanban y cómo funciona? • Asana." [En línea]. Disponible: <https://asana.com/es/resources/what-is-kanban> [Accedido: 31 Julio 2022].
- [15] Jsjs Productions, HTML DESDE 0 - ¿Como Funciona HTML? - Aprender HTML Facil Y Rapido. [En línea]. Disponible: https://books.google.com.ec/books?id=YBf0DwAAQBAJ&printsec=frontcover&dq=html&hl=es&sa=X&redir_esc=y#v=onepage&q=html&f=false. [Accedido: 31 Julio 2022].
- [16] J. Mohedano, J. M. Saiz, and P. S. Román, *Iniciación a Javascript*. [En línea]. Disponible: https://books.google.com.ec/books?id=iy8bAgAAQBAJ&printsec=frontcover&dq=Mohedano+Jorge,+Saiz+Jos%C3%A9+Miguel,+and+Salazar+Rom%C3%A1n+Pedro,+Iniciaci%C3%B3n+a+Javascript.+Espa%C3%B1a.&hl=es&sa=X&redir_esc=y#v=onepage&q&f=false. [Accedido: 31 Julio 2022].
- [17] M. A. Arias, *Guía de HTML5, CSS3, y JavaScript. La Web 2.0: 2ª Edición*. [En línea]. Disponible: https://books.google.com.ec/books?id=kdXECgAAQBAJ&printsec=frontcover&dq=M.+A.+Arias,+Gui%CC%81a+de+HTML5,+CSS3+y+Javascript.+La+Web+2.0.&hl=es&sa=X&redir_esc=y#v=onepage&q=M.%20A.%20Arias%2C%20Gui%CC%81a%20de%20HTML5%2C%20CSS3%20y%20Javascript.%20La%20Web%202.0.&f=false. [Accedido: 31 Julio 2022].
- [18] "Get started with Bootstrap · Bootstrap v5.2." [En línea]. Disponible: <https://getbootstrap.com/docs/5.2/getting-started/introduction/> [Accedido: 31 Julio 2022].
- [19] "¿Qué es Bootstrap y cómo funciona en el diseño web? - Blog, ARWEB Agencia Digital Costa Rica." [En línea]. Disponible: <https://www.arweb.com/blog/%C2%BFque-es-bootstrap-y-como-funciona-en-el-diseno-web/>. [Accedido: 31 Julio 2022].
- [20] "PHP: ¿Qué es PHP? - Manual." [En línea]. Disponible: <https://www.php.net/manual/es/intro-what-is.php>. [Accedido: 01 Agosto 2022].
- [21] S. Combaudon, *MySQL 5.7: administración y optimización*. [En línea]. Disponible: https://books.google.com.ec/books?id=QpYLonKflesC&pg=PT2&dq=St%C3%A9phane+Combaudon,+MySQL+5.7:+administraci%C3%B3n+y+optimizaci%C3%B3n.&hl=es&sa=X&ved=2ahUKEwiP0oOf_PP5AhWFtoQIHReXANcQ6AF6BAgCEAI#v=onepage&q=St%C3%A9phane%20Combaudon%2C%20MySQL%205.7%3A%20administraci%C3%B3n%20y%20optimizaci%C3%B3n.&f=false. [Accedido: 01 Agosto 2022].

- [22] C. Thibaud, Recursos Informáticos MYSQL 5. [En línea]. Disponible: https://books.google.com.ec/books?id=wY0bHPmW-NUC&printsec=frontcover&dq=Cyril+Thibaud,+Recursos+Inform%C3%A1ticos+MySQL+5.&hl=es&sa=X&redir_esc=y#v=onepage&q=Cyril%20Thibaud%2C%20Recursos%20Inform%C3%A1ticos%20MYSQL%205.&f=false. [Accedido: 01 Agosto 2022].
- [23] “Qué es WampServer (Guía Sencilla para Principiantes).” [En línea]. Disponible: <https://www.hostinger.es/tutoriales/que-es-wamp>. [Accedido: 01 Agosto 2022].
- [24] J. Márquez Díaz, L. Sampedro, and F. Vargas, “Instalación y configuración de Apache, un servidor Web gratis.”
- [25] Marc. Delisle, Mastering phpMyAdmin 3.3.x for effective MySQL management : a complete guide to getting started with phpMyAdmin 3.3 and mastering its features. Packt Pub, 2010.
- [26] “Sublime Text: qué es y para qué sirve - Code donostia.” [En línea]. Disponible: <https://www.codedonostia.com/sublime-text-que-es-y-para-que-sirve/>. [Accedido: 01 Agosto 2022].
- [27] M. P. Izaurrealde, “Caracterización de Especificación de Requerimientos en entornos Ágiles: Historias de Usuario.”
- [28] V. Fernández Alarcón, “Desarrollo de Sistemas de Información una Metodología Basada en el Modelado.” [En línea]. Disponible: https://books.google.com.ec/books?id=Sqm7jNZS_L0C&pg=PA132&dq=casos+de+uso&hl=es&sa=X&ved=2ahUKEwjHreSbi7r5AhV8QzABHT1DD2sQ6AF6BAgJEA#v=onepage&q=casos%20de%20uso&f=false. [Accedido: 06 Agosto 2022].
- [29] “PHP: Introducción - Manual.” [En línea]. Disponible: <https://www.php.net/manual/es/intro.pdo.php>. [Accedido: 13 Agosto 2022].

5 ANEXOS

Los anexos pueden ser accedidos desde el siguiente enlace <https://tinyurl.com/48nd6k78>

ANEXO I. Historias de Usuario

ANEXO II. Bosquejos de las interfaces web

ANEXO III. Implementación de la base de datos

ANEXO IV. Implementación de las interfaces web

ANEXO V. Implementación de la lógica de negocios

ANEXO VI. Pruebas de funcionamiento