

TECNOLÓGICO UNIVERSITARIO

VIDA NUEVA

SEDE MATRIZ



TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE

TEMA

**DESARROLLO DEL MÓDULO DE GESTIÓN ACADÉMICA DEL PERSONAL DOCENTE
EN LA INSTITUCIÓN EDUCATIVA FISCAL SUCRE**

PRESENTADO POR

CHICAIZA CORDOVA ANDY JOEL

GALARZA RAMOS SAMUEL EDUARDO

TUTOR

LCDO. JARAMILLO CAICEDO RUBÉN PATRICIO MG

FECHA

ENERO 2024

QUITO – ECUADOR

Certificación del Tutor

En mi calidad de Tutor del Proyecto de Aplicación Práctica con el tema: “Desarrollo del Módulo de Gestión Académica del personal Docente en la Institución Educativa Fiscal Sucre”, presentado por los ciudadanos Chicaiza Cordova Andy Joel y Galarza Ramos Samuel Eduardo para optar por el título de Tecnólogo Superior en Desarrollo de Software, certifico que dicho proyecto ha sido revisado en todas sus partes y considero que reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del tribunal examinador que se designe.

En la ciudad de Quito, del mes de enero de 2024

Tutor: Lcdo. Jaramillo Caicedo Rubén Patricio Mg

C.I.: 060403599

Aprobación del Tribunal

Los miembros del tribunal aprueban el Proyecto de Aplicación Práctica, con el tema: “Desarrollo del Módulo de Gestión Académica del personal Docente en la Institución Educativa Fiscal Sucre”, presentado por los ciudadanos Chicaiza Cordova Andy Joel, Galarza Ramos Samuel Eduardo, facultados en la carrera Tecnología Superior en Desarrollo de Software.

Para constancia firman:

C.I.:
DOCENTE TUVN

C.I.:
DOCENTE TUVN

C.I.:
DOCENTE TUVN

Cesión de Derechos de Autor

Yo, Chicaiza Cordova Andy Joel portador de la cédula de ciudadanía 1753728029 y Galarza Ramos Samuel Eduardo portador de la cédula de ciudadanía 1722644968, facultados en la carrera Tecnología Superior en Desarrollo de Software, autores de esta obra, certifico y proveo al Tecnológico Universitario Vida Nueva usar plenamente el contenido de este Proyecto de Aplicación Práctica con el tema “Desarrollo del Módulo de Gestión Académica del personal Docente en la Institución Educativa Fiscal Sucre.” con el objeto de aportar y promover la cultura investigativa, autorizando la publicación de nuestro proyecto en la colección digital del repositorio institucional, bajo la licencia Creative Commons: Atribución-NoComercial-SinDerivadas.

En la ciudad de Quito, del mes de enero de 2024.

Chicaiza Cordova Andy Joel

C.I.: 1753728029

Galarza Ramos Samuel Eduardo

C.I.: 1722644968

Dedicatoria

A mis amados padres por su ayuda, sacrificio y apoyo constante en todo el transcurso de mi vida cotidiana. Su guía ha sido la brújula que me permitió abrirme paso en los constantes retos académicos durante este desafiante viaje académico.

A mis queridos hermanos, quienes con sus consejos y ejemplos me han llenado de gran valor para superarme día a día. Juntos, hemos superado obstáculos, creando recuerdos que atesoro.

Gracias por esa compañía, esas palabras y, sobre todo, por ser mis hermanos.

A mi sincera amiga, cuya amistad ha sido un faro en las tormentas, ofreciéndome apoyo en los momentos difíciles. Agradezco tu presencia en este trayecto. Gracias por darme tus palabras de nunca rendirme, por estar siempre pendiente de mi progreso y por confiar en mis habilidades.

Samuel Eduardo Galarza Ramos

Dedicatoria especial a mis padres, hermanos y mascotas debido al apoyo brindado durante los años en los que no solo eh comenzado la carrera, sino que también eh crecido como persona, por tratar de comprender mi forma de ver el mundo y permitirme seguir avanzando hacia el futuro manteniendo comunicación y su apoyo conmigo. Agradezco el continuo apoyo que me han brindado y el tiempo que mi mascota ah pasado conmigo durante los momentos complicados de desesperación o de falencias, que con su compañía y con la de mis familiares he logrado avanzar y seguiré avanzando, puliendo mis habilidades y teniendo un desarrollo personal eficiente eficaz y recto.

Chicaiza Cordova Andy Joel

Agradecimiento

Agradezco a Dios por ser mi guía, dándome la fortaleza para alcanzar mi sueño profesional. A mis queridos padres y hermanos, su apoyo incondicional ha sido mi ancla en este viaje, gracias por ser la luz que ilumina mi camino académico y por ser mi inspiración y recordarme la importancia de la familia.

A mi sincera amiga, tu amistad ha sido mi refugio en las tormentas; aprecio tus palabras alentadoras y tu confianza inquebrantable.

A mi Tutor del proyecto de Titulación, agradezco tu orientación y apoyo a lo largo de este proceso académico. Tus enseñanzas han sido fundamentales para el éxito de este proyecto.

Samuel Eduardo Galarza Ramos

Enfatizo el agradecimiento a mis padres por el esfuerzo que realizan para poder brindarme la oportunidad de continuar mis estudios y obtener un título profesional.

Agradezco a mis hermanos y mascotas que durante los años me han aportado con su carácter e inspiración para continuar y avanzar en la vida. Finalmente agradezco a mis amistades por abrirme las puertas a que en el mundo aún existen personas agradables y amables.

Chicaiza Cordova Andy Joel

Tabla de Contenido

Resumen	11
Abstract	12
Introducción	13
Planteamiento del Problema	14
Descripción de la Situación Problemática	14
Formulación del Problema	15
Objetivos	16
Objetivo General	16
Objetivos Específicos	16
Justificación	17
Antecedentes	18
Marco Teórico	19
Aplicación Web	19
Historia	19
Lenguaje de Programación	20
Definición	20
Características	21
Constantes y Variables	21
Tipo de Lenguajes de Programación	23
PHP	23
Java	24
JavaScript	24

	8
HTML 5	25
JQuery	26
CSS	26
Bootstrap	27
Base de Datos	28
Definición y Características	28
Gestor de Base de Datos	29
MySQL	30
PostgreSQL	30
Modelo de Base de Datos	31
Modelo Racional	31
Modelo Entidad-Relación	31
Modelo Relacional de Objetos	32
Servidores	32
Definición y Características	32
Función de un Servidor Web	33
Xampp	34
CPanel	34
Dominio Web	34
Definición y Características	34
Registrador de Dominios	35
Framework	35
Codeigniter 4	36

	9
Arquitectura MVC	36
Modelo	37
Vista	37
Controladores	38
Editores de Texto	38
Visual Studio Code	38
Metodología y Desarrollo del Proyecto	39
Diseño Metodológico	39
Variables y Definición Operacional	39
Variable Independiente: Modulo Docente	39
Variable Dependiente: Eficiencia y Eficacia del Modulo	40
Diseño Muestral	40
Técnicas de Recolección de Datos	41
Técnicas Estadísticas para el Procesamiento de la Información	41
Fase de Planeación	43
Fase de Historia de Usuarios	43
Historia de Usuario	43
Fase de Diseño	48
Desarrollo	48
Codificación	50
Fase de Prueba	55
Técnica de Pruebas de Caja Negra: Requerimiento Funcional / Historias de Usuarios	55
Propuesta	57

	10
Conclusiones	69
Recomendaciones	70
Referencias	71
Anexos	73

Resumen

En la actualidad la tecnología ha permitido a las personas y entidades llevar su información de una manera más eficiente y organizada mediante diversos softwares y aplicaciones creados para dicho objetivo. Las aplicaciones web son usadas para la administración de gran cantidad de información en las diversas instituciones de educación en las cuales el uso de estas aplicaciones facilita tanto la recolección y la tabulación de la información de los estudiantes de la Institución.

Por ello cada institución enfocada al ámbito educativo consta de un servicio web el cual le permite realizar las acciones anteriormente mencionadas de forma eficaz y eficiente, por esta razón el presente proyecto enfocado al módulo de gestión académica del personal docente busca implementar un módulo al sistema el cual permita al docente y a las autoridades de la institución acceder, visualizar, crear y manipular la información de los estudiantes y sus puntajes a lo largo del periodo Académico.

Remediando así la problemática y necesidad de la institución, identificando los procesos administrativos, normas y necesidades las cuales permitirán proveer una solución eficaz de la problemática actual además de evidenciar el uso de herramientas para el desarrollo de un sistema web como son las Bases de datos, Frameworks y librerías acoplándonos a la metodología de Programación Modular para el desarrollo de dicho modulo y realizar la etapa de pruebas con base en los requerimientos solicitados por el personal de la Institución, en donde se identificara que el software supero toda la fase de pruebas y permitiendo así la implementación de esta en la institución para su posterior uso.

Palabras Clave: FRAMEWORKS, METODOLOGÍA DE PROGRAMACIÓN MODULAR, APLICACIÓN WEB, BASE DE DATOS, FASE DE PRUEBAS.

Abstract

Currently, technology has allowed individuals and entities to manage their information more efficiently and organized through various software and applications created for this purpose. Web applications are used for managing large amounts of information in various educational institutions, where the use of these applications facilitates the collection and tabulation of student information within the institution.

Therefore, every institution focused on education includes a web service that enables the aforementioned actions to be carried out effectively and efficiently. For this reason, the present project, focused on the academic management module for teaching staff, seeks to implement a module within the system that allows teachers and institutional authorities to access, view, create, and manipulate student information and their scores throughout the academic period.

This project addresses the institution's problems and needs by identifying the administrative processes, regulations, and requirements that will provide an effective solution to the current issues. It also demonstrates the use of tools for developing a web system, such as databases, frameworks, and libraries, adhering to the Modular Programming methodology for the development of this module and conducting the testing phase based on the requirements requested by the institution's staff. It will be verified that the software has successfully passed the testing phase, allowing its subsequent implementation in the institution for future use.

Keywords: FRAMEWORKS, MODULAR PROGRAMMING METHODOLOGY, WEB APPLICATION, DATABASE, TESTING PHASE.

Introducción

Debido a la situación nacional en las unidades educativas, la Institución Educativa Fiscal Sucre enfrenta un desafío crucial generado por la reciente eliminación de partidas administrativas, como las secretarías. Esta medida ha sumido a la institución en una compleja situación afectando directamente el desarrollo de actividades educativas cotidianas.

En este contexto, el presente proyecto de titulación se centra en el desarrollo de un módulo de gestión académica destinado al personal docente de la Institución Educativa Fiscal Sucre. Este módulo, diseñado con el framework CodeIgniter 4 y aplicando la arquitectura Modelo-Vista-Controlador (MVC), tiene como objetivo optimizar los procesos administrativos y académicos, abordando la falta de personal administrativo y mejorando la eficiencia en la gestión académica.

La estructura del proyecto de titulación se divide en tres capítulos fundamentales. El Capítulo 1: Marco Teórico explorará conceptos clave sobre aplicaciones web, lenguajes de programación, bases de datos, servidores, y el uso de CodeIgniter 4 como framework. En el Capítulo 2: Metodología y Desarrollo del Proyecto, se detallará la investigación de requerimientos, el desarrollo del módulo con CodeIgniter 4, y su implementación y capacitación para mejorar la eficiencia y precisión en procesos administrativos y académicos. Finalmente, en el Capítulo 3: Propuesta y Conclusiones, se presentará la solución propuesta, su impacto esperado en la Institución Educativa Fiscal Sucre, y las conclusiones finales del proyecto.

Esta investigación busca no solo abordar la problemática actual de la institución educativa, sino también ofrecer una solución tecnológica eficiente y adaptada a sus necesidades específicas, contribuyendo así al fortalecimiento de la gestión académica y al mantenimiento de altos estándares en la calidad educativa.

Planteamiento del Problema

Descripción de la Situación Problemática

En respuesta a una reestructuración a nivel nacional en el ámbito de las unidades educativas, la eliminación reciente de partidas administrativas, en particular la supresión de roles como el de las secretarías, ha sumido a instituciones educativas como la Institución Educativa Fiscal Sucre en una distintiva situación respecto a los departamentos administrativos. Esta medida ha generado una falta de control en el manejo de la información académica que involucra a toda la comunidad educativa: docentes, autoridades y estudiantes, afectando directamente el desarrollo de actividades institucionales académicas.

En consecuencia, la Institución Educativa Fiscal Sucre se ve ante la carencia de un sistema administrativo que pueda organizar y proporcionar información de manera oportuna y personalizada a docentes, representantes y estudiantes. Esta ausencia de roles administrativos ha tenido un impacto directo en la gestión de datos y procesos fundamentales, debilitando la capacidad de la institución para ofrecer información relevante en tiempo real.

Como solución a esta situación, surge la necesidad de implementar un sistema eficiente que organice la información específica de cada curso y docente en la institución. Este sistema debe abordar aspectos importantes como la gestión de notas, el control de asistencias y el proceso de tutoría. La interfaz propuesta debe ser intuitiva y personalizada para satisfacer las necesidades individuales de cada docente, contribuyendo así a la mejora integral de la gestión académica.

La Institución Educativa Fiscal Sucre, consciente de la necesidad de mantener estándares elevados en la calidad educativa, propone la creación de un módulo de gestión académica virtual. Este módulo no solo aspira a llenar el vacío dejado por la falta de personal administrativo, sino

también a mejorar la eficiencia en la gestión académica. Este proyecto, busca sentar las bases para una administración académica más efectiva y moderna en la institución, posicionándola como referente en la excelencia educativa.

Formulación del Problema

¿Cómo implementar módulos de gestión académica virtual que permitan cubrir la carencia de roles administrativos en la Institución Educativa Fiscal Sucre?

Objetivos

Objetivo General

Desarrollar un módulo de gestión de información del personal docente en el sistema de gestión académica de la Institución Educativa Fiscal Sucre, proporcionando una solución eficiente y automatizada que mejore los procesos administrativos y académicos.

Objetivos Específicos

- Identificar los requerimientos funcionales y datos necesarios para el módulo de gestión docente mediante entrevistas con los usuarios, para determinar las necesidades que se debe incorporar dentro del módulo docente.
- Desarrollar el módulo para la gestión académica de docentes con CodeIgniter 4, implementando la separación de la lógica, los datos y la interfaz con arquitectura MVC, para la automatización efectiva de los procesos de administración.
- Implementar y capacitar el módulo de gestión docente con el objetivo de mejorar sustancialmente la eficiencia y precisión en los procesos administrativos y académicos.

Justificación

La presente investigación se enfoca en el desarrollo de un módulo de gestión académica destinado al personal docente de la Institución Educativa Fiscal Sucre. Su objetivo central consiste en abordar la complejidad surgida a raíz de la carencia de roles administrativos, proponiendo la implementación de un sistema de gestión académica virtual.

Este proyecto tiene como objetivo mejorar el diseño y desarrollo de un módulo de gestión académica utilizando la plataforma CodeIgniter 4. Al aplicar esta tecnología junto con la metodología SCRUM y a su vez aplicar Modelo-Vista-Controlador (MVC), buscamos aportar conocimientos prácticos y eficientes para gestionar datos académicos en entornos educativos. La elección de CodeIgniter 4 y la implementación del MVC se centran en ofrecer una base técnica sólida y adaptable a las necesidades específicas de la Institución Educativa Fiscal Sucre.

El impacto del proyecto de desarrollo del módulo de gestión académica se sentirá de manera directa en la comunidad educativa de la Institución Educativa Fiscal Sucre. Los principales beneficiarios de esta iniciativa serán los docentes, las autoridades educativas, los estudiantes y sus representantes. La automatización de los procesos administrativos facilitará el acceso desde dispositivos móviles y ordenadores de escritorio, promoviendo así la comunicación eficiente entre todas las partes involucradas en el entorno educativo.

El proyecto de desarrollo del módulo de gestión académica no solo busca mejorar la eficiencia administrativa, sino también generar beneficios sociales como una gestión académica más efectiva y acceso oportuno a información relevante. Esta implementación crucial contribuirá a mantener estándares elevados en la calidad educativa, abordando la carencia de roles administrativos y promoviendo el desarrollo educativo y el bienestar de la comunidad.

Antecedentes

La Institución Educativa Fiscal Sucre se encuentra inmersa en un desafío de relevancia en el marco de la reestructuración nacional de las unidades educativas. La eliminación reciente de partidas administrativas, en particular la supresión de roles como los de las secretarías, ha generado una situación compleja que impacta directamente en la gestión de la información académica. Esta medida ha sumido a la institución en una carencia crítica de roles administrativos, especialmente en la esfera de la gestión de información académica, afectando significativamente el desarrollo normal de las actividades educativas cotidianas.

La ausencia de roles administrativos ha creado un vacío sustancial en la Institución Educativa Fiscal Sucre, particularmente en la gestión de datos específicos de cada curso y docente. La carencia de un servidor administrativo ha debilitado la capacidad de la institución para proporcionar información relevante en tiempo real.

Como consecuencia directa de esta problemática, surge la presente necesidad de implementar un sistema enfocado en que aborde de manera integral la gestión de notas, el control de asistencias y los procesos de tutoría. La Institución Educativa Fiscal Sucre se encuentra en la búsqueda de una solución que no solo subsane la falta de roles administrativos, sino que también mejore la eficiencia en la gestión académica, contribuyendo así al desarrollo educativo de la institución.

La investigación no solo aspira a abordar la problemática actual de la Institución Educativa Fiscal Sucre, sino también a ofrecer una respuesta tecnológica que fortalezca la gestión académica y contribuya al mantenimiento de altos estándares en la calidad educativa.

Marco Teórico

Aplicación Web

Las aplicaciones web se definen de manera integral como el conjunto de códigos de programa compilados y ejecutados dentro de diversos navegadores web. “Estos navegadores interpretan diferentes lenguajes de programación, tales como PHP, HTML, CSS, entre otros, que se utilizan para construir la interfaz de la aplicación, la cual se despliega en nuestras computadoras” (Luján, 2002, p. 45).

Un aspecto fundamental a tomar en cuenta de las aplicaciones dirigidas a la web es su capacidad para comunicarse con un servidor y la base de datos correspondiente. Esta interacción posibilita la obtención de información del sistema y el intercambio de datos esenciales para la ejecución de los programas.

Según el libro de Luján (2002) “Las aplicaciones web permiten la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o el desarrollo del comercio electrónico” (p. 3).

Historia

La historia de las aplicaciones web comienza cuando internet empezó a expandirse en las últimas décadas del siglo pasado. “Al principio, estas aplicaciones eran simples y solo mostraban información básica en páginas estáticas. A medida que la población obtuvo acceso nuevas tecnologías e internet, las aplicaciones web” (Aristizábal, 2012, p. 25).

Un gran cambio ocurrió con la introducción de tecnologías que permitían actualizar la información en tiempo real sin tener que recargar toda la página. Luego, con la llegada de frameworks se facilitó la creación de aplicaciones web más avanzadas y fáciles de usar.

Lenguaje de Programación

Cuando nos referimos al concepto de lenguaje de programación, estamos hablando acerca del conjunto específico de herramientas y reglas que se emplean con el propósito de crear programas de computadora. Básicamente, “se trata de un grupo predeterminado de símbolos y estructuras que distingue a cada lenguaje de programación de manera única” (Martín, 2021, p. 9).

La aplicación de un lenguaje de programación se enfoca en describir de manera precisa las instrucciones que la computadora debe ejecutar, utilizando un lenguaje que sea comprensible tanto para el usuario como para la máquina.

Definición

De manera general un lenguaje de programación se describe como un conjunto de órdenes o comandos que trazan de manera general el proceso que una aplicación debe llevar a cabo. Estas instrucciones se formulan mediante diversos símbolos.

Figura 1

Tipos de lenguajes de programación



Nota. Representación gráfica de tipos de lenguajes de programación con sus logos distintivos.

Reproducido de lenguaje de programación y cuáles son los tipos, por Ecdisis Estudio, 2021

(<https://ecdisis.com/>).

Según Duran (2008) “en general, cada tipo de lenguaje de programación ofrece una serie de mecanismos que lo hacen más (o menos) apropiado al tipo de problema que tratamos de resolver y al tipo de sistema que pretendemos desarrollar” (p. 15).

Características

Sintaxis: Involucra al grupo de símbolos y reglas que describen la estructura y las reglas adecuadas para formar sentencias en el código.

Semántica: Se refiere a las reglas que permiten convertir las sentencias en instrucciones lógicas entendibles para la máquina.

Pragmática: Trata de utilizar de manera efectiva las construcciones particulares del lenguaje.

Constantes y Variables

Un elemento de datos que conserva su valor constante a lo largo del programa se denomina constante, identificado por un nombre específico. En cambio, “una variable, también identificada por un nombre, es un elemento de datos cuyo valor puede alterarse durante la ejecución del programa. Estas variables, categorizadas como globales o locales, determinan su alcance en el código” (Catherine, 2009, pp. 21-22).

Las constantes permiten definir valores fijos que no deben cambiar, proporcionando estabilidad y previsibilidad en el comportamiento del programa. Por ejemplo, “las constantes son ideales para definir parámetros que rigen la configuración del sistema o valores matemáticos que se utilizan repetidamente, como π (pi) en cálculos geométricos” (Catherine, 2009, p. 140).

Las variables son elementos de datos que pueden almacenar diversos tipos de información, como números enteros, cadenas de texto y valores booleanos, adaptándose a las necesidades dinámicas del programa mientras se ejecuta.

Tabla 1

Tipos de Datos Comunes en Desarrollo de Software

Tipo de Datos	Descripción
int	Representa números enteros, sin decimales ni fracciones. Es ideal para contar elementos o expresar cantidades enteras.
float o double	Se emplea para números decimales, permitiendo mayor precisión en cálculos matemáticos que involucran fracciones.
Boolean	Este tipo de dato es como un interruptor digital, almacenando valores verdaderos o falsos. Resulta esencial para tomar decisiones condicionales en el código.
char	Sirve para almacenar un único carácter, como letras o símbolos. Útil cuando se requiere representar datos simples, como iniciales o códigos de identificación.
varchar(n)	Un tipo de dato flexible para almacenar diferentes textos con una longitud variable, con una capacidad máxima de 'n' caracteres. Es útil para nombres, direcciones u otra información variable en longitud.
date	Reservado para almacenar información relacionada con fechas. Facilita la gestión de eventos temporales y programación de tareas basadas en el tiempo.
time	Destinado a representar horas del día, desde minutos hasta segundos. Fundamental en situaciones donde la precisión temporal es esencial.
datetime	Combina la información de fechas y horarios en un solo tipo de formato de dato, permitiendo el almacenamiento de eventos específicos en un punto temporal exacto.
enumeration	Un tipo de dato numérico que limita las opciones posibles hacia un conjunto fijo de valores. Resulta útil para definir categorías predefinidas, como estados o tipos específicos.
array	Ideal para gestionar colecciones de datos del mismo tipo, permitiendo un fácil acceso y manipulación de múltiples elementos a través de índices.

Nota. Muestra una referencia rápida de tipos de datos esenciales utilizados en el desarrollo de software.

Tipo de Lenguajes de Programación

PHP

El lenguaje de programación PHP se distingue como un conjunto diverso de herramientas que simplifica el desarrollo de aplicaciones con una estructura y programación clásica. Este lenguaje está disponible para el uso de todas las personas ya que es open source. Un aspecto fundamental a destacar es que, al ejecutarse en el servidor, PHP no requiere compatibilidad por parte del navegador, lo cual confiere una independencia respecto al entorno visual del navegador. Aristizábal (2012) describe a PHP de la siguiente manera:

El lenguaje de programación PHP goza en la actualidad de muy buena aceptación en la comunidad de programadores para la Web, por su versatilidad, facilidad para manipularlo, está en el servidor, porque es libre, fácil en su aprendizaje, de libre distribución, tiene acceso a bases de datos y otras funcionalidades orientadas a la red, porque precisamente en la red, se encuentra una cantidad considerable de comunidades de programadores de apoyo y soporte, variedad de librerías de código que se pueden utilizar, y sobre todo, es orientado a objetos. (p. 15)

Figura 2

Logo representativo de lenguaje de programación Hypertext Preprocessor (PHP)



Nota. Logo oficial del lenguaje de programación PHP. Reproducido de Iconos de lenguaje de programación PHP, por Illustration, 2019 (<https://es.vexels.com>).

Java

Java considerado entre los lenguajes de programación más populares para desarrollar sistemas empresariales y aplicaciones web. Se destaca por su capacidad de ser utilizado en diferentes plataformas, hablando de portabilidad, y su capacidad para adaptarse y crecer según las necesidades del proyecto, hablando de su escalabilidad.

Arroyo (2019) señala que:

Java es independiente de la plataforma porque tiene una máquina virtual para cada sistema; esta máquina funciona como puente entre el sistema operativo y el programa en Java, y posibilita que este último se lea y se ejecute a la perfección. (p. 5)

Figura 3

Logo representativo del lenguaje de programación Java



Nota. Lenguaje de programación Java-Logo el cual es una base fundamental para las aplicaciones. Reproducido de ¿Qué es JAVA?, por Rosa, 2019 (<https://rosamarfil.es/>).

JavaScript

JavaScript, ampliamente empleado en páginas web junto con HTML o XHTML, es un lenguaje de programación del lado del cliente que se ejecuta de forma directa en el navegador del usuario.

Proporciona funcionalidades interactivas y mejoras visuales, como mostrar/ocultar texto, deslizar imágenes y crear animaciones.

“JS ha logrado un importante posicionamiento entre los desarrolladores por las incontables bondades en el desarrollo de páginas web, por ejemplo, páginas dinámicas para el envío y validación de formularios, interactividad, animación, seguimiento de actividades de usuario, etcétera”. (Arroyo, 2019, p. 11)

Figura 4

Logo representativo del Lenguaje de programación JavaScript una extensión de Java



Nota. Lenguaje de programación JavaScript-Logo el cual es usado en diversos lenguajes.

Reproducido de El mejor CMS JavaScript, por Sanity, 2020 (<https://www.sanity.io/>).

HTML 5

HTML5, tiene la función principal de establecer el diseño, el contenido y la estructura de dichas páginas. Opera como un lenguaje que indica al navegador cómo mostrar la información al usuario.

De acuerdo con Herrera (2012) “HTML5 nos proporcionará una manera de hacer un código más limpio, más fácil de leer y escribir, cubriendo al mismo tiempo y de mejor manera la cada vez mayor demanda de funcionalidades por parte de programadores, diseñadores y usuarios” (p. 6).

JQuery

JQuery se presenta como un framework JavaScript de gran relevancia, es considerado como una librería de código que simplifica procesos y rutinas listas para su implementación, “permitiendo a los programadores evitar el desarrollo manual de tareas básicas al aprovechar las implementaciones ya probadas y funcionales dentro del propio framework” (Comesaña, 2013, p. 15).

Figura 5

Logo representativo del lenguaje JQuery aplicado en el funcionamiento



Nota. Lenguaje de programación JQuery-Logo lenguaje usado para el funcionamiento interactivo y estético de diversas funciones en las aplicaciones. Reproducido de JQuery Hello World, por Mkyong, 2024 (<https://mkyong.com/>).

CSS

CSS son Hojas de Estilo en Cascada, se refiere esencialmente a la presentación visual de un documento web. “Se encarga de definir aspectos como el tamaño, tipo de fuente, espaciado y otros atributos visuales de cada elemento, permitiendo al desarrollador ajustar la disposición de los elementos según sus preferencias” (Luján, 2002, p. 34).

Los estilos CSS pueden ser aplicados de diversas maneras, ya sea directamente en el documento como CSS interno, de forma externa en archivos independientes como CSS externo, o incluso de manera dinámica y remota como CSS en línea.

Herrera (2012) explica que el programador usa CSS para especificar el formato para un tipo de etiqueta o para una etiqueta en particular. La idea principal detrás de todo esto es separar el código de estructura del código de presentación, de manera que cuando se modifica el archivo con código CSS, éste inmediatamente afecta a todas las páginas HTML que hacen referencia a él, evitando así editar cada página HTML de forma individual. (p. 8)

Figura 6

Logo de Cascading Style Sheets usado para la estructura del programa y sus elementos visibles



Nota. Logo del Lenguaje de programación CSS-Logo aplicado en diversas aplicaciones web.

Reproducido de CSS Logo, por Rudloff, 2022 (<https://1000marcas.net/>).

Bootstrap

Bastidas (2019) señala: "Es una biblioteca que es de uso de varias plataformas o conjunto de herramientas de código abierto para diseño de sitios web y aplicaciones web. Es un Framework que fue usado por Twitter para su negocio" (p. 278).

Bootstrap, se presenta como una herramienta gratuita diseñada para simplificar el procedimiento en creación de sitios web responsivos y adaptados a dispositivos móviles. Facilita a los desarrolladores la construcción rápida de sitios al ofrecer una serie de sintaxis para diseñar plantillas. En términos sencillos, este framework permite a los programadores crear páginas en la web más eficientemente. Este framework incorpora scripts fundamentados en JS, CSS y HTML que abarcan varias funciones y elementos vinculados al diseño en línea.

Base de Datos

Una base de datos es un sistema organizado de información relacionada, representando aspectos característicos que definan de manera concreta un objeto real. Almacenada electrónicamente, puede contener información diversa como números, imágenes y archivos, organizada mediante un sistema de bases de datos representado con las siglas DBMS.

En una base de datos estándar, la información se organiza de manera que facilite su acceso, permitiendo tanto consultas individuales como la generación de informes extensos.

Definición y Características

“Una base de datos es una colección estructurada de registros o datos almacenados en un sistema informático y organizado de tal manera que se pueda buscar rápidamente la información y se pueda recuperar también rápidamente” (Nixon, 2019, p. 161).

Es un sistema estructurado que permite almacenar, administrar y recuperar información eficientemente, basándose en un modelo de datos que describe la estructura y las conexiones entre la información. Los sistemas de administración de bases de datos (SGBD) controlan y administran estas bases, ofreciendo lenguaje y herramientas que permitan consultar e interactuar con los datos que existe dentro de la base de datos, facilitando consultas, informes y mantenimiento de la totalidad de la información guardada dentro de la base de datos.

Gestor de Base de Datos

Un administrador de bases de datos (DBMS) es una aplicación informática que simplifica la creación, administración y manipulación de bases de datos, centrándose principalmente en la organización de los datos. Proporcionando un entorno eficiente y estructurado para almacenar, recuperar y modificar la información. Los DBMS permiten a los usuarios interactuar con las bases de datos mediante consultas y operaciones, garantizando la integridad de los datos y proporcionando herramientas para gestionar la estructura y relaciones de la información almacenada.

“Todos los accesos a la base de datos están controlados mediante un sofisticado paquete de software llamado sistema de gestión de base de datos (DBMS, por sus siglas en inglés)”
(Catherine, 2009, p. 6).

Figura 7

Gestores de Bases de Datos Comunes



Nota. Presentan los gestores de bases de datos utilizados en proyectos de desarrollo de software cumpliendo con el rol de ser el intermediario entre el programa y la gestión de la base de datos. Reproducido de Tipos de bases de datos, por J. Larque, 2023 (<https://www.hiberus.com/>).

MySQL

MySQL es un gestor de base de datos que se destaca por su amplio uso en aplicaciones web. Originado por la empresa MySQL AB de origen sueco, se encuentra disponible bajo la Licencia de Software Libre. Además, para aquellas empresas que buscan utilizar este software de manera privada, existe una opción comercial que requiere la adquisición de una licencia para su utilización.

“El SQL en MySQL significa Structured Query Language. Este lenguaje tiene su base en el inglés y también se utiliza en otras bases de datos como Oracle y Microsoft SQL Server” (Nixon, 2019, p. 162).

Figura 8

Logo del Gestor de Base de Datos MySQL comúnmente usado en la mayoría de programas web



Nota. Presentación visual del distintivo logo del gestor de base de datos MySQL actualmente usado en la aplicación para el correcto funcionamiento de las consultas SQL. Reproducido de Conexiones asíncronas a MySQL, por J. Guerra, 2022 (<https://javguerra.github.io>).

PostgreSQL

PostgreSQL, es considerado al igual que MySQL como un sistema de administración de bases de datos relacional ampliamente empleado en aplicaciones web. PostgreSQL proporciona una robusta plataforma para la dirección eficiente de bases de datos. Además, al igual que

MySQL, ofrece una opción comercial para empresas que deseen utilizar el software de manera privada, requiriendo la adquisición de una licencia correspondiente

“PostgreSQL es un gestor de bases de datos orientadas a objetos (SGBDOO o ORDBMS en sus siglas en inglés) muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99” (Gibert & Pérez, 2020, p. 5).

Modelo de Base de Datos

Explican la forma en que se diseña la organización lógica de una base de datos. Son conceptos esenciales para introducir la idea de abstracción en una base de datos. Los modelos de datos establecen la manera en que los datos se relacionan entre sí y cómo se manipulan y guardan en el sistema. “Es crucial comprender la definición de "bases de datos" al tener en cuenta la distinción entre datos e información, ya que estos aspectos son esenciales para el desarrollo de las bases de datos” (Juárez, 2006, p. 45).

Modelo Racional

Los datos se encuentran en una base de datos relacional en forma de filas y columnas, que se combinan para formar una tabla. Por lo general, los datos se organizan en varias tablas que se pueden conectar usando una clave principal o externa. “Estos identificadores exclusivos representan las numerosas conexiones que existen entre las tablas, y una variedad de modelos de datos suelen ser utilizados para representar estas relaciones” (Akiko, 2010, p. 20).

Modelo Entidad-Relación

El modelo entidad-relación es una herramienta que simplifica la representación de los elementos involucrados en un proceso empresarial y cómo se relacionan entre sí. El diagrama entidad-relación es la representación gráfica de este modelo, donde las entidades se muestran mediante rectángulos, los atributos se representan con círculos o elipses, y las relaciones se

visualizan como líneas que conectan las entidades que tienen alguna conexión. “Otra forma común de presentar este tipo de diagramas es listar los atributos de una entidad en filas dentro del rectángulo que la representa” (Esic, 2018, p. 1).

Modelo Relacional de Objetos

Un objeto-relacional se emplea para caracterizar una base de datos que ha progresado desde el modelo relacional hacia una base de datos híbrida que integra tanto tecnologías relacionales como de objetos. “Una de las razones fundamentales del rápido éxito de las bases de datos relacionales fue su habilidad para generar consultas ad hoc” (Valencia, 2009, p. 2).

Servidores

Un servidor hardware es una máquina física que está integrada en una red informática y donde funcionan uno o más servidores software además del sistema operativo. Un servidor hardware también se conoce como "host", que en inglés significa "anfitrión". En esencia, cualquier computadora puede actuar como "host" mediante el uso del software adecuado para servidores.

Por otro lado, un servidor basado en software es un programa que proporciona un servicio especial que otros programas, llamados clientes, pueden utilizar localmente o a través de una red. El tipo de servicio ofrecido depende del software del servidor. “La comunicación se establece mediante el modelo cliente-servidor, y en lo que respecta al intercambio de datos, entran en juego los protocolos de transmisión específicos del servicio” (IONOS, 2013, p. 1).

Definición y Características

Dado que los servidores son computadoras responsables de satisfacer las solicitudes de los clientes en la red, una de sus características clave es la necesidad de mantener sus recursos disponibles de forma continua para garantizar la operatividad constante de la red. En otras

palabras, deben permanecer encendidos y siempre listos. Esta condición explica por qué en ocasiones una página web o sus recursos pueden no cargarse: “si un servidor colapsa o experimenta una falla, la porción correspondiente de la red a su cargo dejará de estar accesible para los usuarios” (Ivar, 2000, pp. 48-50).

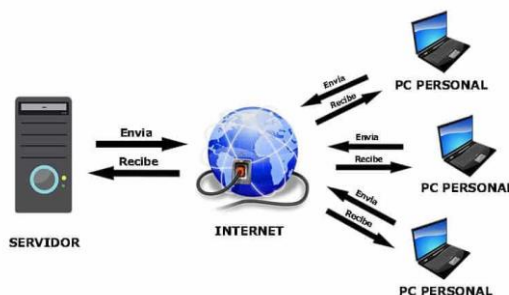
Además, los servidores suelen ser equipos de alto rendimiento con suficiente capacidad de procesamiento para manejar numerosas solicitudes de clientes con un tiempo mínimo de demora. Esto implica la necesidad de contar con alimentación eléctrica constante y otros recursos físicos para asegurar su funcionamiento continuo. En muchos casos, “los servidores se almacenan en compartimentos especialmente acondicionados, incluso con control climático, manteniendo un entorno suficientemente fresco y libre de interferencias” (IONOS, 2023, p. 1).

Función de un Servidor Web

El servidor web tiene la responsabilidad de proporcionar información o recursos a usuarios u otras máquinas que lo requieran. “Esta información puede manifestarse en diversas formas o servicios, abarcando desde archivos de texto, imágenes o vídeos hasta desempeñar funciones como las de un servidor de impresión, correo o web” (Roblenado, 2019, p. 2).

Figura 9

Estructura del funcionamiento de un servidor web y sus relaciones con los diferentes



Nota. Representa el funcionamiento y los componentes. Reproducido de ¿Qué es un servidor web?, por B. Rodríguez, 2019 (<https://www.webebre.net/>).

Xampp

Xampp es uno de los servidores web multiplataforma más empleados, facilitando a los desarrolladores la creación y prueba de sus programas en un servidor web local. Fue creado por Apache Friends y su código fuente original puede ser examinado o modificado por la comunidad. “Se encuentra disponible en 11 idiomas y es compatible con diversas plataformas, incluyendo Windows, macOS y Linux” (Mundobytes, 2013, p. 1).

CPanel

CPanel es un panel de control ampliamente reconocido que facilita la gestión de su servidor de alojamiento web a través de una interfaz basada en la web, diseñada para ser amigable para los usuarios. “Es particularmente destacado en entornos de alojamiento compartido, siendo la solución ofrecida por usuarios del hosting asequibles” (Kinsta, 2023, p. 1).

Dominio Web

Definición y Características

Cada página web consta principalmente de dos elementos esenciales: un nombre de dominio y un servidor de alojamiento web. El nombre de dominio está vinculado a una dirección IP, y al ingresar un nombre de dominio en un navegador, el servidor web correspondiente es localizado a través del Sistema de Nombres de Dominio (DNS), que es una red global de servidores.

Los servidores DNS encuentran la dirección IP relacionada con el nombre de dominio y la envían al navegador web. Posteriormente, el navegador solicita al servidor de alojamiento del dominio información sobre el sitio.

Una vez que el servidor de alojamiento proporciona los datos solicitados, el navegador web los convierte en una página web que los usuarios pueden visualizar. Es así como “la

conexión entre el nombre de dominio, la dirección IP y el servidor de alojamiento colabora en la presentación de la página web” (Gustavo, 2023, p. 1).

Figura 10

Representación de la Estructura de un Dominio en la web



Nota. Análisis rápido de la estructura del dominio con sus componentes clave: 'google' (nombre), 'com' (extensión), y 'mx' (código de país). Reproducido de Qué es un dominio web, por V. Fetter, 2019 (<https://www.hostgator.mx/>).

Registrador de Dominios

Los registradores de dominios son entidades que simplifican el proceso de añadir y distribuir nombres de dominio a una escala amplia. En su función, actúan como intermediarios entre los registros de dominios que supervisan los dominios de nivel superior (.es, .com o .net) y los usuarios que buscan obtener un dominio. En términos simples, “aquellos que desean adquirir un nombre de dominio deben comunicarse con un registrador” (IONOS, 2010, p. 2).

Framework

Un framework, también conocido como marco de trabajo, constituye una estructura que facilita el desarrollo de software de forma más ágil y eficiente. En este contexto, solo necesitas escribir el código que gestione la lógica específica de tu aplicación.

La ventaja radica en que no es necesario reinventar conceptos básicos del desarrollo, como la creación de clases, el manejo de objetos y la definición de funciones comunes. En

esencia, “un framework establece la estructura para tu proyecto futuro y provee las herramientas esenciales que puedes utilizar como componentes fundamentales” (EBAC, 2023, p. 2).

Codeigniter 4

CodeIgniter 4 es un marco de trabajo PHP liviano que posibilita a los desarrolladores la creación eficiente y rápida de aplicaciones web. “Su reconocida facilidad de uso y su arquitectura flexible lo convierten en una excelente elección para el desarrollo de aplicaciones en línea, independientemente de su tamaño o escala” (Codeigniter, 2023, p. 3).

Figura 11

Representación de logo de Codeigniter



Nota. Representación visual del distintivo logo de Codeigniter equivalente a un entorno de trabajo usado con el fin de una mejor organización y funcionamiento en las aplicaciones.

Reproducido de CodeIgniter, por Coria, 2016 (<https://www.coriaweb.hosting/>).

Arquitectura MVC

MVC es una propuesta arquitectónica del software que se emplea para fragmentar el código según sus diversas responsabilidades, estableciendo capas independientes encargadas de ejecutar tareas específicas, lo que resulta en diversos beneficios.

Inicialmente, se utiliza MVC en sistemas que requieren interfaces de usuario, pero en la práctica, este patrón de arquitectura puede aplicarse a diferentes tipos de aplicaciones. Surgió como respuesta a la necesidad de crear software más resistente con un ciclo de vida más adecuado, “priorizando la facilidad de mantenimiento, la reutilización del código y la clara separación de conceptos” (Ángel, 2023, p. 11).

Modelo

Representa los datos y la lógica de negocio de la aplicación. Es responsable de gestionar el acceso a los datos, realizar cálculos y procesamiento, y definir las reglas de negocio.

Esta capa se enfoca en la manipulación de datos, por lo tanto, incorporará métodos para acceder a la información y para actualizar su estado.

Generalmente, los datos estarán almacenados en una base de datos, por lo que en “los modelos se incluirán todas las funciones que interactúan con las tablas y ejecutan operaciones como selects, updates, inserts, entre otros” (Ángel, 2023, p. 12).

Vista

Las vistas albergan el código de la aplicación encargado de generar la presentación de las interfaces o vistas de usuario, es decir, el código necesario para renderizar la aplicación en formato HTML. “En las vistas, únicamente encontramos el código HTML y PHP que posibilita la visualización de la salida” (Ángel, 2023, p. 12).

Es la interfaz de usuario que presenta los datos al usuario final de manera visual. Se encarga de mostrar la información de manera adecuada y de interactuar con el usuario para recoger datos de entrada.

Controladores

“Incluye la codificación esencial para manejar las diferentes solicitudes en la aplicación, tales como mostrar un elemento, llevar a cabo una compra, realizar una búsqueda de información, entre otras” (Ángel, 2023, p. 12).

Editores de Texto

Se trata de una aplicación informática diseñada para editar o crear archivos digitales compuestos exclusivamente por texto sin formato, es decir, “archivos que carecen de un formato de texto específico y que comúnmente se conocen como archivos de texto o texto plano” (Susana, 2022, p. 2).

Actúa como intermediario entre el modelo y la vista. Gestiona las solicitudes del usuario y actualiza el modelo en función de las acciones realizadas en la vista. Además, actualiza la vista cuando el modelo cambia.

Visual Studio Code

Es un editor de código fuente gratuito y de código abierto desarrollado por Microsoft para Windows, Linux y macOS. “Ofrece funciones como depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código” (Corporation, 2018, p. 1).

Metodología y Desarrollo del Proyecto

Diseño Metodológico

El desarrollo del proyecto del módulo de gestión académica del personal docente se basa en un tipo de enfoque cuantitativo y experimental debido a diversos factores los cuales se centran en la recolección de datos exactos, medidas exactas de rendimiento de estudiantes, control de asistencias, tutorías y a su vez diversas medidas que se irán implementando rigiéndose a las necesidades de la Institución y el ministerio de educación todas estas permitirán la automatización de diversas actividades.

La investigación realizada para el proyecto conlleva una investigación descriptiva permitiendo describir variables tanto dependientes como las independientes, además de dar uso del tipo de investigación experimental en la obtención de información sobre las causas y efectos que ocurren tras la carencia del módulo y su implementación. “Esta información recolectada se le dará un seguimiento con el cual se corrige las falencias y novedades encontradas a través de la experimentación” (Hernández, 2006, p. 140).

Variables y Definición Operacional

Variable Independiente: Modulo Docente

El módulo docente es la sección del sistema integrado el cual permitirá a los docentes monitorizar, asignar y controlar las asistencias notas de los estudiantes asignados a cada docente, rigiéndose según los estándares del Ministerio de Educación y su reglamento.

Los sistemas web son considerados programas y aplicaciones los cuales ejecutan un conjunto de funciones programadas previamente con la finalidad de almacenar, modificar y obtener información. El módulo tendrá las siguientes dimensiones:

- Control de asistencia

- Gestión de puntajes
- Revisión de Tutorías
- Personalización de Perfil
- Control de Rendimiento Estudiantil

Variable Dependiente: Eficiencia y Eficacia del Modulo

Tanto la eficiencia como la eficacia del módulo y del sistema en si depende del correcto uso y creación de diferentes secciones tanto por parte de los programadores como por parte de las autoridades es decir, así como la información mostrada dependerá de que los docentes y autoridades ingresen la información de forma correcta y a la vez información de manera específica, controlada y coordinada, la estructura de la Base de Datos también debe ser lo más simple y eficaz al momento de almacenar, modificar y obtener la información en las tablas.

“La investigación cuantitativa y cualitativa permitiendo medir y describir toda esta información de manera rápida, precisa y sencilla” (Carles, 2004, p. 50).

Las dimensiones serán sustentadas en:

- Optimización en ejecución de acciones.
- Tabla de almacenamiento de Usuarios
- Tabla de almacenamiento de Puntajes
- Tabla de almacenamiento de cursos
- Tabla de almacenamiento de Estudiantes

Diseño Muestral

La población se basa en el conjunto de personas, situaciones, objetos o acontecimientos en el cual se investigará refiriéndose al contexto de investigación del proyecto vigente. La Institución Educativa Fiscal Sucre consta de diferentes departamentos y secciones sin embargo el

presente modulo está enfocado en el personal docente los cuales podrán gozaran de diversas acciones que facilitaran la asignación de puntuación a sus estudiantes designados y a su vez el control de estos entre estas acciones serán almacenamiento de puntajes estudiantiles segmentados por materias y cursos y control de tutoría. Estos estudiantes y docentes serán basadas en una tabla y una lista de cotejo con docentes y estudiantes razón por la cual la se dará uso ala mayor parte de la población total del colegio.

Técnicas de Recolección de Datos

La técnica a usar para la recolección de datos y almacenar dicha información recolectada para el desarrollo del módulo de Docente con las diversas herramientas necesarias es la de observación y formularios de contacto con la razón de dar uso a una retroalimentación a la Base de datos por parte de la misma población mediante de los formularios de contacto y una observación del comportamiento que los usuarios portan ante dicho sistema y formularios de esta forma se podrá tanto “alimentar la Base de datos con información legitima y a su vez permitirá seleccionar las herramientas más factibles para el desarrollo del presente modulo” (Gallardo, 2000, p. 78).

Técnicas Estadísticas para el Procesamiento de la Información

Para la identificación de requerimientos por parte del Institución Educativa Fiscal Sucre serán tomado en cuenta las diversas formas de recolección de datos además de enfocarse en su mayoría en las necesidades e ideas que proporcionen las autoridades de dicha institución, pretendiendo así acoplarse a su punto de vista y necesidad, con estos requerimientos se buscara usar las diferentes herramientas como son Frameworks, “motores de bases de datos y servidor de alojamiento en las cuales nos permitan realizar la automatización y a su vez optimización de diversas actividades que el personal docente requiere realizar diariamente” (Wong, 2017, p. 52).

El uso de los diferentes recursos y el establecimiento de las herramientas a usar serán de distribución libre para elegir un lenguaje de desarrollo el cual se acople a los conocimientos y familiarización de los programadores evitando muchas discrepancias por cualquier problema relacionado a la falta de licencia o financiamiento de las herramientas.

Las pruebas para la aceptación del módulo de docentes serán presentadas y basadas tanto en las opiniones tras la observación del comportamiento de este en su aplicación de prueba como las opiniones por parte de las autoridades de esta forma podremos evidenciar “el correcto funcionamiento de este y permitiendo a los programadores enfocarse en la funcionalidad, rendimiento y legibilidad de toda la información en este módulo” (Wong, 2017, pp. 15-16).

La metodología Scrum para el desarrollo de software ágiles permite asegurar el control del proyecto y una adecuada implementación en donde la misma sea concreta y eficaz. Esta metodología definirá el tiempo, calidad y alcance de las variables anteriormente mencionadas del presente proyecto de software, tras ellas pueden establecerse arbitrariamente actores externos como son usuarios y jefes de proyecto mientras que el valor del sobrante se fijara por el equipo de desarrollo.

Finalmente, el ciclo de vida del proyecto se basará en el entendimiento de lo que el usuario requiere, apreciamiento del esfuerzo y el desarrollo del software concediendo así un producto final a la institución por lo que el ciclo será dinámico en donde se especificara los requerimientos uno tras otro al comienzo, intermedio y final del proyecto pasando por diversas fases como son

Fase de Planeación

De acuerdo con el criterio de Joskowicz (2008) la fase se puede definir como el alcance general del proyecto por lo que el cliente delimita lo que necesita a través de la redacción de sencillas historias de usuarios. (p. 43)

Los programadores valoraran los tiempos de desarrollo según sus criterios en base a la información recibida y los requerimientos que sean solicitados. Se procederá a representar las historias de usuarios en base a los requerimientos de las autoridades.

Fase de Historia de Usuarios

Historia de Usuario

Son conocidas como documentos escritos en el cual cada uno de los requisitos son mencionados usando un lenguaje sencillo por parte de la persona que requiere el software. A continuación, se describirá las diferentes historias de usuarios y cada uno de los campos.

Tabla 2

Historia de usuario 01, registro de usuario docente

HISTORIA DE USUARIO	
Numero: 1	Usuario: Inspector General
Denominación de la Historia: Registro de Usuarios Docentes	
HISTORIA DE USUARIO	
Prioridad en negocio: Alta	Amenaza de desarrollo: Intermedia
Desarrollador encargado: Chicaiza Andy	
Descripción: Permitirá al Inspector general crear usuarios con sus respectivas contraseñas para docentes los cuales serán registrados por este mismo y a su vez estos podrán actualizar su información en cualquier momento.	
Observaciones: Dependerá del Inspector General.	

Nota. Historia de usuario solicitada para el registro de usuarios y almacenamiento de estos.

Tabla 3*Historia de usuario 02, asignación de estudiantes*

HISTORIA DE USUARIO	
Numero: 2	Usuario: Inspector General
Denominación de la Historia: Asignación de Estudiantes	
HISTORIA DE USUARIO	
Prioridad en negocio: Alta	Amenaza de desarrollo: Intermedia
Desarrollador encargado: Chicaiza Andy	
Descripción: Permitirá al Inspector General asignar estudiantes a los diferentes cursos anteriormente creados con sus respectivas materias y a estos asignarles un docente.	
Observaciones: El correcto funcionamiento dependerá de listas de cotejo para la asignación de estudiantes.	

Nota. Historia de usuario solicitada para la asignación de estudiantes y almacenamiento de estos.**Tabla 4***Historia de usuario 03, actualización de información*

HISTORIA DE USUARIO	
Numero: 3	Usuario: Docente
Denominación de la Historia: Actualización de información	
HISTORIA DE USUARIO	
Prioridad en negocio: Alta	Amenaza de desarrollo: Intermedia
Desarrollador encargado: Chicaiza Andy	
Descripción: Permitirá al Usuario vigente en el módulo de docente modificar y actualizar cualquier información errónea almacenada sobre su perfil de usuario.	
Observaciones: Ninguna.	

Nota. Historia de Usuario solicitada para la actualización de la información del docente.

Tabla 5*Historia de usuario 04, asignación de puntuaciones*

HISTORIA DE USUARIO	
Numero: 4	Usuario: Docente
Denominación de la Historia: Asignación de Puntuaciones	
HISTORIA DE USUARIO	
Prioridad en negocio: Alta	Amenaza de desarrollo: Intermedia
Desarrollador encargado: Chicaiza Andy	
Descripción: Permitirá al Usuario vigente en el módulo de docente modificar, analizar y actualizar las notas asignadas a cada docente en la tabla tabulada por el ministerio de educación y las autoridades.	
Observaciones: Ninguna.	

Nota. Historia de usuario solicitada para la actualización, inserción y eliminación de puntajes.**Tabla 6***Historia de usuario 05, automatización de cálculos*

HISTORIA DE USUARIO	
Numero: 5	Usuario: Docente
Denominación de la Historia: Automatización de Cálculos	
HISTORIA DE USUARIO	
Prioridad en negocio: Alta	Amenaza de desarrollo: Leve
Desarrollador encargado: Galarza Samuel	
Descripción: Permitirá a los usuarios docentes gozar de una automatización de cálculos automáticos para el puntaje final, quimestral o parcial de cada estudiante.	
Observaciones: Ninguna.	

Nota. Historia de usuario solicitada para la optimización de tiempo en el ingreso de puntajes.

Tabla 7*Historia de usuario 06, Automatización de rendimiento estudiantil*

HISTORIA DE USUARIO	
Numero: 6	Usuario: Docente
Denominación de la Historia: Automatización de rendimiento estudiantil	
HISTORIA DE USUARIO	
Prioridad en negocio: Media	Amenaza de desarrollo: Leve
Desarrollador encargado: Galarza Samuel	
Descripción: Permitirá a los usuarios docentes gozar de una visualización resumida y específica de cada estudiante resaltando su rendimiento general diferenciada en tablas de alcanza el aprendizaje, no alcanza el aprendizaje, entre otras.	
Observaciones: Ninguna.	

Nota. Historia de usuario solicitada para el control de rendimiento estudiantil.**Tabla 8***Historia de usuario 07, Control de Asistencia*

HISTORIA DE USUARIO	
Numero: 7	Usuario: Docente
Denominación de la Historia: Control de Asistencia	
HISTORIA DE USUARIO	
Prioridad en negocio: Alta	Amenaza de desarrollo: Leve
Desarrollador encargado: Chicaiza Andy	
Descripción: Permitirá a los usuarios docentes el registro de asistencia en su clase, con día y fecha automáticamente permitiendo así de una forma simple solo registrar estudiantes faltantes.	
Observaciones: Ninguna.	

Nota. Historia de usuario solicitada para la optimización de tiempo en el ingreso de puntajes.

Tabla 9*Historia de usuario 08, gestión y control tutorial*

HISTORIA DE USUARIO	
----------------------------	--

Numero: 8	Usuario: Docente
------------------	-------------------------

Denominación de la Historia: Gestión y Control Tutorial

HISTORIA DE USUARIO

Prioridad en negocio: Alta	Amenaza de desarrollo: Leve
-----------------------------------	------------------------------------

Desarrollador encargado: Galarza Samuel

Descripción: Permite a los docentes recibir cursos completos con sus respectivas materias para la revisión de notas de los estudiantes de dicho curso y la asistencia de estos.

Observaciones: Ninguna.

Nota. Historia de usuario solicitada para gestión y control tutorial.**Tabla 10***Historia de usuario 09, interfaz y control de comportamiento*

HISTORIA DE USUARIO	
----------------------------	--

Numero: 9	Usuario: Docente
------------------	-------------------------

Denominación de la Historia: Interfaz y control de Comportamiento

HISTORIA DE USUARIO

Prioridad en negocio: Alta	Amenaza de desarrollo: Leve
-----------------------------------	------------------------------------

Desarrollador encargado: Chicaiza Andy

Descripción: Permitir al usuario docente comprender de una forma amigable y sencilla las interacciones y acciones que esté realizando mediante formas de sistemas ya conocidos anteriormente por el usuario.

Observaciones: Ninguna.

Nota. Historia de usuario solicitada para la familiarización y fácil comprensión.

Tabla 11*Historia de usuario 10, interfaz y comportamiento familiar*

HISTORIA DE USUARIO	
Numero: 10	Usuario: Docente
Denominación de la Historia: Interfaz y administración de clases	
HISTORIA DE USUARIO	
Prioridad en negocio: Alta	Amenaza de desarrollo: Leve
Desarrollador encargado: Galarza Samuel	
Descripción: Permitir al usuario docente configurar, observar y revisar tanto los cursos y materias como los estudiantes asignados a sus diferentes asignaturas.	
Observaciones: Ninguna.	

Nota. Historia de usuario solicitada para la familiarización y fácil comprensión.

Fase de Diseño

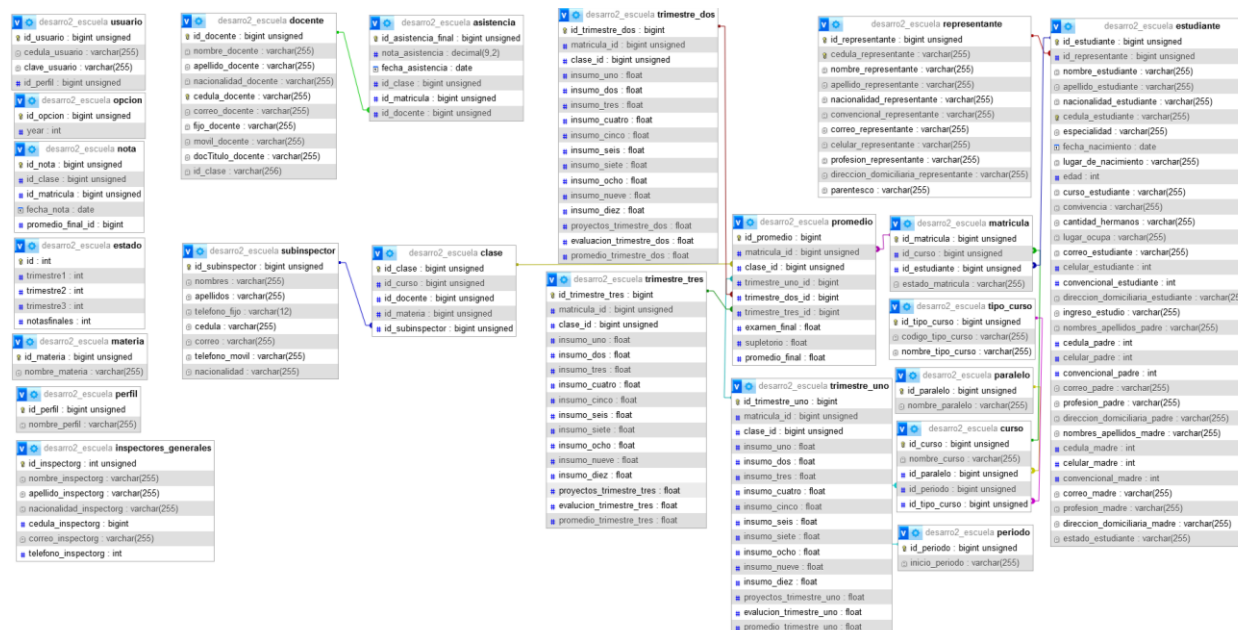
En la presente fase del ciclo se realizarán los diversos sectores visuales los cuales serán presentados hacia el usuario, estos posteriormente serán de referencia para el avance y actualización del sistema. La metodología escogida sugiere realizar varios diseños simples pero eficientes.

Desarrollo

En nuestro proyecto de titulación sobre el módulo de gestión académica del personal docente, usamos un diagrama de base de datos creado con MySQL. Este mapa nos ayuda a entender cómo se relacionan las diferentes partes de la aplicación.

Figura 12

Diagrama y estructura de la base de datos utilizada en el sistema de gestión académica



Nota. Diagrama modelo entidad-relación específico de MySQL, detallando la estructura de la base de datos y las relaciones entre entidades.

A través de este diagrama, se pueden visualizar todas las tablas presentes en nuestra base de datos, como se mostró en el diagrama anterior. Detallando las características esenciales de cada tabla, incluyendo su tipo, número de filas, cotejamiento y tamaño. Este análisis minucioso nos proporciona una visión detallada de la estructura interna de la base de datos, fundamentando así la solidez de nuestro módulo de gestión académica del personal docente.

Figura 13

Análisis de la Estructura de la Base de Datos en el Módulo de Gestión Académica

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
asistencia	Examinar	13	InnoDB	utf8mb4_general_ci	64.0 KB	-
clase	Examinar	23	InnoDB	utf8mb4_general_ci	80.0 KB	-
curso	Examinar	51	InnoDB	utf8mb4_general_ci	64.0 KB	-
docente	Examinar	43	InnoDB	utf8mb4_general_ci	32.0 KB	-
estado	Examinar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
estudiante	Examinar	79	InnoDB	utf8mb4_general_ci	48.0 KB	-
inspectores_generales	Examinar	1	InnoDB	utf8mb4_general_ci	16.0 KB	-
materia	Examinar	36	InnoDB	utf8mb4_general_ci	16.0 KB	-
matricula	Examinar	49	InnoDB	utf8mb4_general_ci	48.0 KB	-
nota	Examinar	0	InnoDB	utf8mb4_general_ci	64.0 KB	-
opcion	Examinar	15	InnoDB	utf8mb4_general_ci	16.0 KB	-
paralelo	Examinar	9	InnoDB	utf8mb4_general_ci	16.0 KB	-
perfil	Examinar	6	InnoDB	utf8mb4_general_ci	16.0 KB	-
periodo	Examinar	3	InnoDB	utf8mb4_general_ci	16.0 KB	-
promedio	Examinar	31	InnoDB	utf8mb4_general_ci	96.0 KB	-
representante	Examinar	91	InnoDB	utf8mb4_general_ci	32.0 KB	-
subinspector	Examinar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
tipo_curso	Examinar	5	InnoDB	utf8mb4_general_ci	16.0 KB	-
trimestre_dos	Examinar	31	InnoDB	utf8mb4_general_ci	16.0 KB	-
trimestre_tres	Examinar	31	InnoDB	utf8mb4_general_ci	16.0 KB	-
trimestre_uno	Examinar	31	InnoDB	utf8mb4_general_ci	16.0 KB	-
usuario	Examinar	138	InnoDB	utf8mb4_general_ci	16.0 KB	-
22 tablas	Número de filas	690	InnoDB	utf8mb4_0900_al_ci	736.0 KB	0 B

Nota. Representación visual detallada de las tablas, tipos, filas, cotejamientos y tamaños de la base de datos de MySQL.

Codificación

Figura 14

Gráfico de codificación de funciones para el control de las acciones en el perfil Docente

```

public function nota()
{
    $request = \Config\Services::request();
    $id = $request->getPost('id');
    $mat = $request->getPost('mat');
    $cla = $request->getPost('cla');
    $cal = $request->getPost('cal');
    $cal1 = $request->getPost('cal1');
    $cal2 = $request->getPost('cal2');
    $cal3 = $request->getPost('cal3');
    $cal4 = $request->getPost('cal4');
    $cal5 = $request->getPost('cal5');
    $cal6 = $request->getPost('cal6');
    $cal7 = $request->getPost('cal7');
    $cal8 = $request->getPost('cal8');
    $cal9 = $request->getPost('cal9');
    $evaluacion = $request->getPost('evaluacion'); //evaluacion
    $proyectos = $request->getPost('proyectos');
    $promedio = $request->getPost('call1'); // promedio trimestre
    $hoy = date("Y-m-d");
    $contador = count($mat);

    for ($i = 0; $i < $contador; $i++) {
        $data = array(
            'matricula_id' => $mat[$i],
            'clase_id' => $cla[$i],
            'insumo_uno' => $cal[$i],
            'insumo_dos' => $cal1[$i],
            'insumo_tres' => $cal2[$i],

```

Nota. Muestra la codificación y la estructura del archivo de control para el módulo de docentes.

Figura 15

Gráfico de estructura de la tabla y campos del modelo Docente

```

DocenteModel.php (PHP script, ASCII text, with CRLF line terminators)

<?php
namespace App\Models;

use CodeIgniter\Model;

class DocenteModel extends Model
{
    protected $table      = 'docente';
    protected $primaryKey = 'id_docente';

    protected $useAutoIncrement = true;

    protected $returnType     = 'array';
    protected $useSoftDeletes = false;

    protected $allowedFields = ['nombre_docente', 'apellido_docente', 'nacionalidad_docente',
    'cedula_docente', 'fijo_docente', 'correo_docente', 'movil_docente', 'doclitulo_docente',
    'id_clase'];

    protected $useTimestamps = false;
    protected $createdField   = 'created_at';
    protected $updatedField   = 'updated_at';
    protected $deletedField   = 'deleted_at';

    protected $validationRules = [];
    protected $validationMessages = [];
    protected $skipValidation   = false;
}

```

Nota. Muestra la codificación de una clase modelo para la interacción entre el programa y la BD.

Figura 16

Gráfico de la estructura del modelo Trimestre Uno con su tabla campos y registros

```

Trimestre1Model.php (PHP script, ASCII text, with CRLF line terminators)

<?php
namespace App\Models;

use CodeIgniter\Model;

class Trimestre1Model extends Model
{
    protected $table      = 'trimestre_uno';
    protected $primaryKey = 'id_trimestre_uno';

    protected $useAutoIncrement = true;

    protected $returnType     = 'array';
    protected $useSoftDeletes = false;

    protected $allowedFields = [
        'matricula_id',
        'clase_id',
        'insumo_uno',
        'insumo_dos',
        'insumo_tres',
        'insumo_cuatro',
        'insumo_cinco',
        'insumo_seis',
        'insumo_siete',
        'insumo_ocho',
        'insumo_nueve',
        'insumo_diez',
        'proyectos_trimestre_uno',
        'evaluacion_trimestre_uno',
        'promedio_trimestre_uno'
    ];

    protected $useTimestamps = false;
}

```

Nota. Muestra la codificación de una clase modelo para la interacción entre el programa y la BD.

Figura 17

Gráfico de codificación de la estructura del modelo Promedio con sus campos

```

PromedioModel.php (PHP script, ASCII text, with CRLF line terminators)

<?php

namespace App\Models;

use CodeIgniter\Model;

class PromedioModel extends Model
{
    protected $table      = 'promedio';
    protected $primaryKey = 'id_promedio';

    protected $useAutoIncrement = true;

    protected $returnType     = 'array';
    protected $useSoftDeletes = false;

    protected $allowedFields = ['matricula_id','clase_id','trimestre_uno_id','trimestre_dos_id','trimestre_tres_id',
    'examen_final','supletorio','promedio_final'];

    protected $useTimestamps = false;
    protected $createdField   = 'created_at';
    protected $updatedField   = 'updated_at';
    protected $deletedField   = 'deleted_at';

    protected $validationRules   = [];
    protected $validationMessages = [];
    protected $skipValidation    = false;
}

```

Nota. Muestra la codificación de una clase modelo para la interacción entre el programa y la BD.

Figura 18

Gráfica de codificación de la vista de docente, estructura en lenguaje HTML y PHP

```

<form action="php echo base_url(); ?&gt;docentesA" method="post"&gt;
  &lt;?php
  echo '&lt;center&gt;'.date("Y-m-d").'&lt;/center&gt;';
  ?&gt;
  &lt;table&gt;
    &lt;caption&gt;Asistencia &lt;/caption&gt;

    &lt;thead&gt;
      &lt;tr&gt;
        &lt;th&gt;Apellido&lt;/th&gt;
        &lt;th&gt;Nombre&lt;/th&gt;
        &lt;th&gt;Asistencia&lt;/th&gt;
      &lt;/tr&gt;
    &lt;/thead&gt;
    &lt;tbody&gt;
      &lt;?php
      foreach ($matricula as $mat) {
        foreach ($estudiante as $value) {
          if ($value['id_estudiante'] == $mat['id_estudiante'] &amp;&amp; $mat['id_curso'] == $clase['id_curso']) {
            &lt;tr&gt;
              &lt;td&gt;?php echo $value['apellido_estudiante'] ?&gt;&lt;/td&gt;
              &lt;td&gt;?php echo $value['nombre_estudiante'] ?&gt;&lt;/td&gt;
              &lt;td&gt;
                &lt;input type="hidden" name="cla[]" value="<?php echo $clase['id_clase'] ?&gt;" required&gt;
                &lt;input type="hidden" name="mat[]" value="<?php echo $mat['id_matricula'] ?&gt;" required&gt;
                &lt;input type="number" name="ast[]" value="0" required&gt;
              &lt;/td&gt;
            &lt;/tr&gt;
          &lt;?php
          }
        }
      }
    &lt;/tbody&gt;
  &lt;/table&gt;
  &lt;br&gt;
  &lt;center&gt;&lt;button style="background-color: green;color: white;border-radius: 18px;cursor: pointer;overflow: hidden;font-size: 16px;height: 35px;
font-weight: bold;border-color: white; width:100px;" type="submit"&gt;Enviar&lt;/button&gt;&lt;/center&gt;
&lt;/form&gt;
</pre

```

Nota. Muestra la codificación de la vista de Docente el cual se mostrará al usuario.

Figura 19

Gráfica de codificación de la vista de clases, cursos y materias en lenguaje HTML y PHP

```

listaEA.php (HTML document, ASCII text, with CRLF line terminators)
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="<?php echo base_url('css/tabla.css'); ?>">
  <title>Document</title>
</head>
<body>
  <form action="<?php echo base_url(); ?>docenteSA" method="post">
  <?php
  echo 'center'.date("Y-m-d").'center';
  ?>
  <table>
    <caption>Asistencia </caption>
    <thead>
      <tr>
        <th>Apellido</th>
        <th>Nombre</th>
        <th>Asistencia</th>
      </tr>
    </thead>
    <tbody>
      <?php
      foreach ($matricula as $mat) {
        foreach ($estudiante as $value) {
          if ($value['id_estudiante'] == $mat['id_estudiante'] && $mat['id_curso'] == $class['id_curso']) {
            <tr>
              <td><?php echo $value['apellido_estudiante']; ?></td>
              <td><?php echo $value['nombre_estudiante']; ?></td>
              <td>
                <input type="hidden" name="cla[]" value="<?php echo $class['id_class']; ?>" required>
                <input type="hidden" name="mat[]" value="<?php echo $mat['id_matricula']; ?>" required>
              </td>
            </tr>
          }
        }
      }
    </tbody>
  </table>
  </form>
  </body>
</html>

```

Nota. La imagen representa la codificación de la visualización y el formato para el agrupamiento de clases y cursos.

Figura 20

Gráfica de codificación de la página principal del módulo docente

```

index.php (HTML document, UTF-8 Unicode text, with CRLF line terminators)
<center>
<h1>Docentes</h1>
</center><br>
<div class="colpassword" style="margin-left: 50px;">
<input class="input_field" id="buscar" type="text" required />
<label for="password" class="input_label">Buscar</label>
<span class="input_icon-wrapper">
  <i class="input_icon ri-eye-off-line"></i>
</span>
</div>
<table id="tabla_datos">
  <caption><a href="<?php echo base_url(); ?>docenteN" NUEVO</a></caption>
  <thead>
    <tr>
      <th>Nombres</th>
      <th>Apellidos</th>
      <th>Cedula</th>
      <th>Nacionalidad</th>
      <th>Correo</th>
      <th>Figjo</th>
      <th>Movil</th>
      <th>Titulo</th>
      <th>Acciones</th>
    </tr>
  </thead>
  <tbody>
    <?php
    foreach($docente as $value){
    ?>
    <tr>
      <td hidden="<?php echo $value['id_docente']; ?>"></td>
      <td style="text-transform: uppercase;"><?php echo $value['nombre_docente']; ?></td>
      <td style="text-transform: uppercase;"><?php echo $value['apellido_docente']; ?></td>
      <td><?php echo $value['cedula_docente']; ?></td>
      <td style="text-transform: uppercase;"><?php echo $value['nacionalidad_docente']; ?></td>
      <td style="word-wrap: break-word;"><?php echo $value['correo_docente']; ?></td>
      <td><?php echo $value['figjo_docente']; ?></td>
      <td><?php echo $value['movil_docente']; ?></td>
      <td><?php echo $value['docTitulo_docente']; ?></td>
      <td>
        <form action="<?php echo base_url(); ?>docenteE" method="post">
        <input type="hidden" name="id" id="id" value="<?php echo $value['id_docente']; ?>">
        <button class="btn-secondary custom-button" type="submit" >EDITAR</button>
      </td>
    </tr>
  </tbody>
</table>

```

Nota. Muestra la codificación principal para el módulo de docentes esta se mostrará primero.

Figura 21

Gráfica de codificación de visualización del perfil de docente

```

perfil.php (HTML document, ASCII text, with CRLF line terminators)

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title></title>
  <style>
    .text-h1{
      text-transform: uppercase;
    }
  </style>
</head>
<body >

  <?php
  foreach($docente as $doc){
  echo "<center>";
  if(session('cedula')===$doc['cedula_docente']){
  $nombre= $doc['nombre_docente'];
  $apellidos=$doc['apellidos_docente'];
  $nacionalidad=$doc['nacionalidad_docente'];
  $correo=$doc['correo_docente'];
  $cedula=$doc['cedula_docente'];
  $celular=$doc['movil_docente'];
  $telefonofijo=$doc['fijo_docente'];
  $cedula=$doc['cedula_docente'];
  echo "<div class='text-h1'>";
  echo "<h1 text-transform:uppercase >BIENVENIDO ". $apellidos." ". $nombre."</h1>";
  echo "<img src='usuario.png' width='15%' height='15%'>";
  ";
  echo "</div>";

  echo "<table border='2' id='tabla' align='center' background:'green'> ";
  echo "<thead>
    <tr>
      <th bgcolor='blue'>
        APELLIDOS Y NOMBRES DE DOCENTE
      </th>
  </thead>
  </table>
  </body>
  </html>
  
```

Nota. Muestra la codificación del archivo perfil para traer datos.

Figura 22

Archivo máster de rutas y redirecciones para el enrutamiento de información

```

Routes.php (PHP script, UTF-8 Unicode text)

$routes->post('/seguidientos', 'RegistroController::registros', ['filter' => 'authGuard']);
$routes->post('/registro', 'RegistroController::registraEstudiante', ['filter' => 'authGuard']);
$routes->post('/registro', 'RegistroController::registraRepresentante');

$routes->get('/asistencia', 'AsistenciaController::index', ['filter' => 'authGuard']);
$routes->get('/asistencia', 'AsistenciaController::new', ['filter' => 'authGuard']);
$routes->post('/asistencia', 'AsistenciaController::save', ['filter' => 'authGuard']);
$routes->post('/asistencia', 'AsistenciaController::edit', ['filter' => 'authGuard']);
$routes->post('/asistencia', 'AsistenciaController::update', ['filter' => 'authGuard']);
$routes->post('/asistencia', 'AsistenciaController::delete', ['filter' => 'authGuard']);

$routes->get('/clase', 'ClaseController::index', ['filter' => 'authGuard']);
$routes->get('/clase', 'ClaseController::new', ['filter' => 'authGuard']);
$routes->post('/clase', 'ClaseController::save', ['filter' => 'authGuard']);
$routes->post('/clase', 'ClaseController::edit', ['filter' => 'authGuard']);
$routes->post('/clase', 'ClaseController::update', ['filter' => 'authGuard']);
$routes->post('/clase', 'ClaseController::delete', ['filter' => 'authGuard']);

$routes->get('/curso', 'CursoController::index', ['filter' => 'authGuard']);
$routes->get('/curso', 'CursoController::new', ['filter' => 'authGuard']);
$routes->post('/curso', 'CursoController::save', ['filter' => 'authGuard']);
$routes->post('/curso', 'CursoController::edit', ['filter' => 'authGuard']);
$routes->post('/curso', 'CursoController::update', ['filter' => 'authGuard']);
$routes->post('/curso', 'CursoController::delete', ['filter' => 'authGuard']);

$routes->get('/docente', 'DocenteController::index', ['filter' => 'authGuard']);
$routes->get('/docente', 'DocenteController::new', ['filter' => 'authGuard']);
$routes->post('/docente', 'DocenteController::save', ['filter' => 'authGuard']);
$routes->post('/docente', 'DocenteController::edit', ['filter' => 'authGuard']);
$routes->post('/docente', 'DocenteController::update', ['filter' => 'authGuard']);
$routes->post('/docente', 'DocenteController::delete', ['filter' => 'authGuard']);
$routes->get('/docente', 'DocenteController::lista', ['filter' => 'authGuard']);

$routes->get('/docentePer', 'DocenteController::verperfil', ['filter' => 'authGuard']);
$routes->post('/docenteP1', 'DocenteController::trasmite_ano', ['filter' => 'authGuard']);
$routes->post('/docenteP2', 'DocenteController::trasmite_mes', ['filter' => 'authGuard']);
$routes->post('/docenteP3', 'DocenteController::trasmite_tres', ['filter' => 'authGuard']);
$routes->post('/docenteP4', 'DocenteController::notas_finales', ['filter' => 'authGuard']);
$routes->post('/docenteP5', 'DocenteController::notas_finales_guardar', ['filter' => 'authGuard']);
$routes->post('/docenteP6', 'DocenteController::menu_asistencia', ['filter' => 'authGuard']);
$routes->post('/docenteP7', 'DocenteController::asistencia_visualizar', ['filter' => 'authGuard']);
$routes->post('/docenteP8', 'DocenteController::cargar_asistencia', ['filter' => 'authGuard']);
$routes->post('/docenteP9', 'DocenteController::nota', ['filter' => 'authGuard']);
$routes->get('/inicio', 'Home::inicio', ['filter' => 'authGuard']);
$routes->get('/nav', 'Home::nav', ['filter' => 'authGuard']);
$routes->get('/cambio', 'Home::cambiocontraseña', ['filter' => 'authGuard']);
$routes->post('/changePass', 'Home::actualizacionContraseña', ['filter' => 'authGuard']);
  
```

Nota. Muestra las redirecciones y el DNS de cada función permitiendo saber el camino de la información.

Fase de Prueba

La siguiente fase se basa en la iteración de entrega de módulos funcionales y sin errores, deseando así al cliente con un avance simbólico el cual seguirá mejorando y proveyendo información para posibles mejoras futuras, en actualizaciones.

Técnica de Pruebas de Caja Negra: Requerimiento Funcional / Historias de Usuarios

- **Historia de usuario 01**, Registro de Usuarios Docentes: Resultado esperado (funcional): El sistema registra eficiente y correctamente los docentes colocándoles credenciales de ingreso válidas.
- **Historia de usuario 02**, Asignación de Estudiantes: Resultado esperado (funcional): El sistema asigna eficazmente los estudiantes a los docentes y clases indicadas, apareciendo estos en estas clases y los perfiles de docentes.
- **Historia de usuario 03**, Actualización de información: Resultado esperado(funcional): El sistema envía peticiones de actualización, eliminación e inserción tanto de notas como de estudiantes reflejándose en la BD y en la interfaz en tiempo real.
- **Historia de usuario 04**, Asignación de Puntuaciones: Resultado esperado (funcional): El sistema asigna las notas correctamente a estudiantes indicados y en la materia indicada sin perjudicar al resto de información y actualizando la BD.
- **Historia de usuario 05**, Automatización de Cálculos: Resultado esperado (funcional): El sistema proporciona los cálculos correspondientes de manera rápida y eficiente mostrándole al usuario en tiempo real el puntaje y las demás calificaciones estandarizadas.

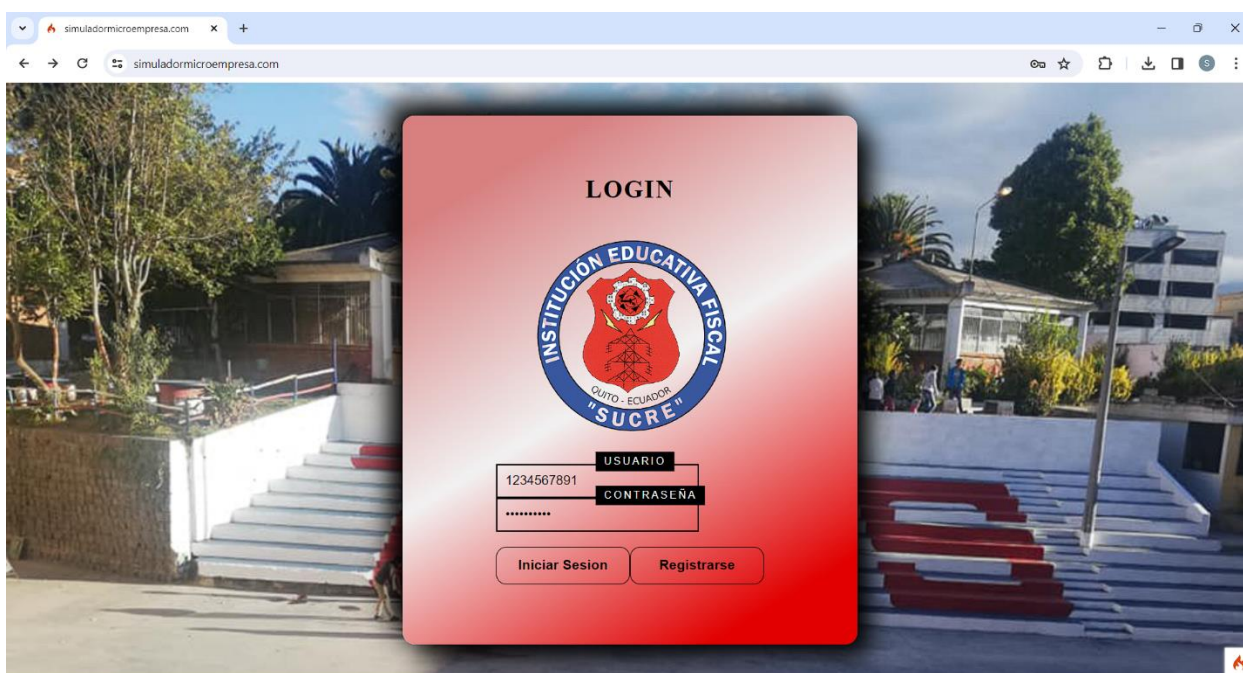
- **Historia de usuario 06**, Automatización de rendimiento estudiantil: Resultado Esperado (funcional): El sistema permite que el usuario docente visualice el rendimiento general del curso y materia, permitiendo un fácil control de rendimiento escolar.
- **Historia de usuario 07**, Control de Asistencia: Resultado Esperado (funcional): El sistema permite el control de asistencia mediante el ingreso diario de asistencias a la Base de datos y finalmente mostrando un resumen.
- **Historia de usuario 08**, Gestión y Control Tutorial: Resultado esperado(funcional): El sistema permite controlar y visualizar las notas y asistencias de clases que sean asignadas a un docente tutor.
- **Historia de usuario 09**, Interfaz y Control de Comportamiento: Resultado esperado (funcional): El sistema es familiar con los usuarios mostrándoles una interfaz para la visualización y control de comportamiento amigable para el usuario docente.
- **Historia de usuario 10**, Interfaz y administración de clases: Resultado esperado (funcional): El sistema permite la eficiente visualización de un menú el cual permite la administración de las clases que el docente tiene permitiendo así saber en todo momento que clases y cursos están a su cargo con la materia asignada.

Propuesta

La propuesta del proyecto incluye la presentación de la interfaz de la aplicación web que ha sido cargada en el servidor CPanel. En esta interfaz, se resalta la presencia de la página de inicio, `index.php`, la cual posibilita la iniciación de sesión para el usuario, el docente y la secretaria dentro de la misma interfaz distinguiendo a cada uno a partir de la información registrada dentro de la base de datos.

Figura 23

Interfaz de Inicio de Sesión y Registro



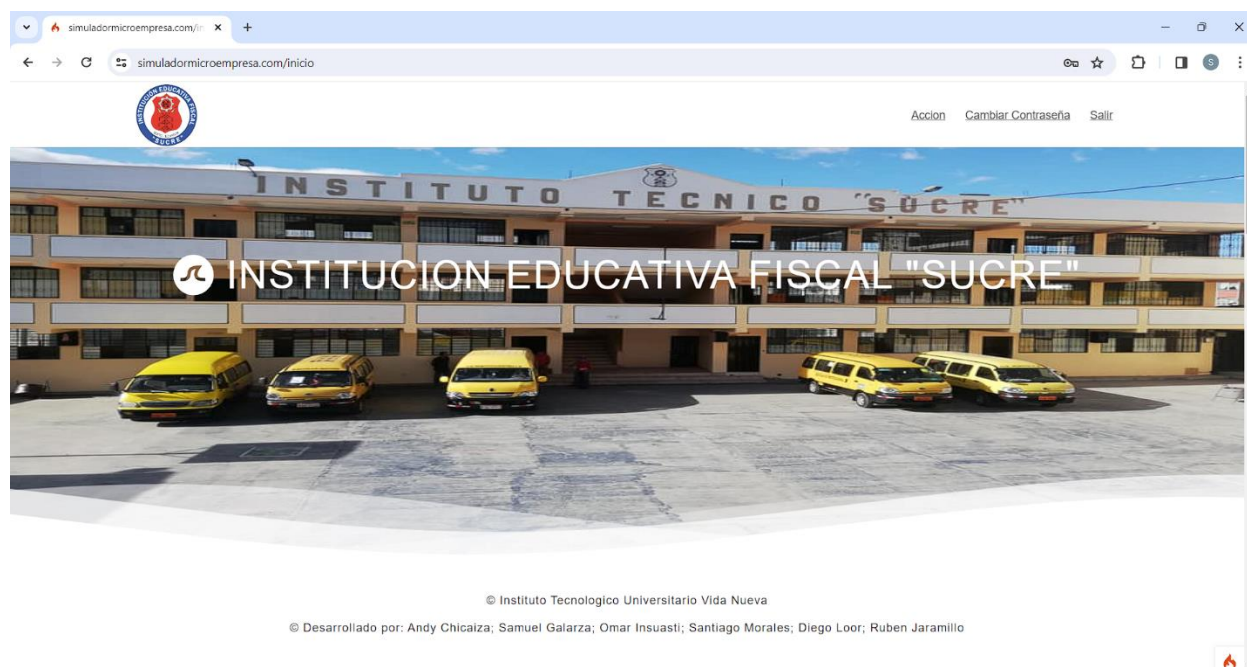
Nota. Una vista que muestra la página de inicio diseñada para que los usuarios inicien sesión en el sistema.

Dentro del sistema, el acceso se realiza mediante las credenciales asignadas al docente, lo que resulta en una redirección automática a la página principal del módulo de gestión académica del personal docente. En esta interfaz intuitiva, se presentan las opciones de "Acciones", "Cambiar Contraseña" y "Salir". La elección de "Acciones" conduce al panel encargado de

gestionar las notas, asistencias y perfil del docente, mientras que "Cambiar Contraseña" lleva a una interfaz destinada al cambio de las credenciales del docente en cuestión. Por último, la opción "Salir" facilita la salida del interfaz del docente, redirigiendo al usuario a la página de inicio de sesión. Este diseño pretende ofrecer una experiencia de usuario eficiente y amigable.

Figura 24

Interfaz Inicial del Docente en el Módulo de Gestión Académica

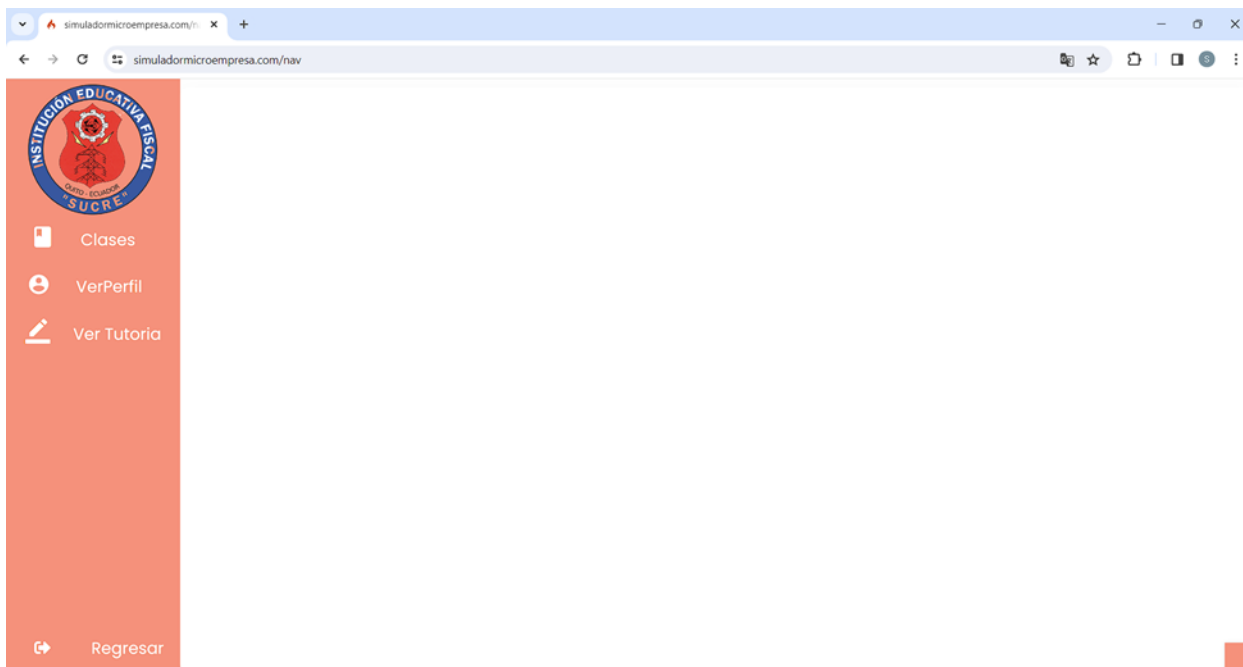


Nota. Interfaz inicial tras el acceso con las credenciales del docente.

Al seleccionar la opción "Acciones" desde la interfaz inicial, se accede a la interfaz de gestión específica diseñada para el docente. En este espacio, se presenta un menú desplegable que exhibe las clases, perfil y tutorías asociadas al docente en cuestión. Este enfoque proporciona al docente un acceso directo y organizado a la información esencial, facilitando la gestión de clases, revisión del perfil y coordinación de tutorías de manera eficaz y centralizada.

Figura 25

Interfaz 'Acciones' para la Gestión Docente



Nota. Interfaz 'Acciones' que se despliega tras ingresar al sistema como docente.

Dentro del menú desplegable, se encuentra la opción "Clases", la cual desencadena una interfaz detallada que enumera las clases asignadas al docente. En esta sección, se encuentran dos botones de "Acciones": uno que guía hacia la interfaz de notas, identificado como "Notas", y otro que conduce a la interfaz de asistencia denominado "Asistencia".

Al seleccionar la opción "Notas", se despliega un nuevo menú con las diferentes secciones correspondientes a los trimestres académicos: Trimestre 1, Trimestre 2, Trimestre 3 y, finalmente, Notas Finales. En cada interfaz de trimestre, se presentan dos cuadros informativos. En el primero, se detalla la información correspondiente de estudiantes matriculados en la materia del docente, distinguiendo cada trimestre de manera individual. En el segundo cuadro, se presenta una tabla dinámica que realiza cálculos automáticos, evaluando el número de estudiantes que se encuentran en cada categoría de rendimiento académico, como Dominio de

Aprendizajes (DA), Alcanza los Aprendizajes (AA), Próximo a Alcanzar los Aprendizajes (PA) y No Alcanza los Aprendizajes (NA). Para ofrecer mayor claridad, se incluye un encabezado de tabla que especifica la equivalencia, rangos, cualitativo, número de estudiantes y porcentaje para cada categoría.

En lo que respecta a la interfaz de "Notas Finales", se presenta un cuadro similar a los trimestres, pero en este caso, calcula los promedios generados en los trimestres anteriores. La tabla dinámica de equivalencias se mantiene, ofreciendo una evaluación holística del desempeño académico de los estudiantes. Vale la pena destacar que todos estos procesos de cálculo se ejecutan de manera automática y cohesionada dentro de la misma interfaz, garantizando una evaluación académica eficiente y precisa.

Figura 26

Interfaz 'Trimestre 1' para la Gestión de Notas del Docente

The screenshot displays the 'Trimestre 1' interface for grade management. It features a sidebar with navigation options: Trimestre 1, Trimestre 2, Trimestre 3, and Notas Finales. The main content area is titled 'TRIMESTRE 1' and 'Trimestre No 1'. It contains a table of student grades and a summary table of equivalencies.

Nómina de Estudiantes	Actividades disciplinares o interdisciplinares- Individuales										Actividades disciplinares o interdisciplinares-Grupales					Evaluación Trimestral				Cualitativo	
	Insumos					Promedio	Insumos					Promedio	Formativos y portafolio Total Evaluaciones 80%	Proyecto	Proyecto 5%	Estructurada	Estructurada 5%	Promedio final Trimestre 1			
	INSUMO 1	INSUMO 2	INSUMO 3	INSUMO 4	INSUMO 5		INSUMO 1	INSUMO 2	INSUMO 3	INSUMO 4	INSUMO 5										
Álvarez Nunez Arturo Jesus	10	10	5	10	10	9.00	4.05	5	6	5	5	4	5.00	2.25	6.30	10	0.50	8	0.40	7.20	AA
Garcia Martinez Diego Miguel	1	5	6	8	9	5.80	2.61	5	6	2	6	2	4.20	1.89	4.50	10	0.50	5	0.25	5.25	PA
Martinez Sánchez Valentina Adriana	8	6	7	8	9	7.60	3.42	4	10	9	6	10	7.80	3.51	6.93	8	0.40	9	0.45	7.78	AA

Below the table is a green button labeled 'GUARDAR NOTAS'.

Equivalencia	Rangos	Cualitativo	No. Est.	Porcentaje
Domina los aprendizajes (DA).	9,00 – 10,00	DA	0	0.00%
Alcanza los aprendizajes (AA).	7,00 – 8,99	AA	2	66.67%

Nota. Interfaz 'Trimestre 1' que se despliega tras ingresar a la interfaz notas, mostrando la tabla de notas como la tabla dinámica de Equivalencias.

Figura 27

Interfaz 'Trimestre 2' para la Gestión de Notas del Docente

TRIMESTRE 2

Trimestre No 2

Nómina de Estudiantes	Actividades disciplinarias o interdisciplinarias- Individuales					Actividades disciplinarias o interdisciplinarias- Grupales					Evaluación Trimestral				Promedio final Trimestre 2	Cualitativo					
	Insumos					Insumos					Formativas o portafolio Total Evaluaciones 10%	Proyecto	Proyecto 5%	Estructurada			Estructurada 5%				
	INSUMO 1	INSUMO 2	INSUMO 3	INSUMO 4	INSUMO 5	Promedio	Promedio 45%	INSUMO 1	INSUMO 2	INSUMO 3								INSUMO 4	INSUMO 5	Promedio	Promedio 45%
Álvarez Nunez Arturo Jesus	10	5	5	5	10	7.00	3.15	6	6	10	10	10	8.40	3.78	6.93	10	0.50	0	0.00	7.43	AA
García Martínez Diego Miguel	6	6	5	5	5	5.40	2.43	10	10	10	10	5	9.00	4.05	6.48	5	0.25	0	0.00	6.73	PA
Martínez Sánchez Valentina Adriana	8	9	7	5	9	7.60	3.42	8	9	9	9	9	8.80	3.96	7.38	8	0.40	0	0.00	7.78	AA

GUARDAR NOTAS

Equivalencia	Rangos	Cualitativo	No. Est.	Porcentaje
Domina los aprendizajes (DA).	9,00 – 10,00	DA	0	0.00%
Alcanza los aprendizajes (AA).	7,00 – 8,99	AA	2	66.67%

Nota. Interfaz 'Trimestre 2' que se despliega tras ingresar a la interfaz notas, mostrando la tabla de notas como la tabla dinámica de Equivalencias.

Figura 28

Interfaz 'Trimestre 3' para la Gestión de Notas del Docente

TRIMESTRE 3

Trimestre No 3

Nómina de Estudiantes	Actividades disciplinarias o interdisciplinarias- Individuales					Actividades disciplinarias o interdisciplinarias- Grupales					Evaluación Trimestral				Promedio final Trimestre 3	Cualitativo					
	Insumos					Insumos					Formativas o portafolio Total Evaluaciones 10%	Proyecto	Proyecto 5%	Estructurada			Estructurada 5%				
	INSUMO 1	INSUMO 2	INSUMO 3	INSUMO 4	INSUMO 5	Promedio	Promedio 45%	INSUMO 1	INSUMO 2	INSUMO 3								INSUMO 4	INSUMO 5	Promedio	Promedio 45%
Álvarez Nunez Arturo Jesus	10	5	2	5	5	5.40	2.43	5	6	6	6	6	5.80	2.61	5.04	6	0.30	5	0.25	5.59	PA
García Martínez Diego Miguel	10	5	6	6	6	6.60	2.97	5	6	2	2	2	3.40	1.53	4.50	2	0.10	0	0.00	4.60	PA
Martínez Sánchez Valentina Adriana	7	8	9	9	3	7.20	3.24	9	9	9	9	6	8.40	3.78	7.02	9	0.45	9	0.45	7.92	AA

GUARDAR NOTAS

Equivalencia	Rangos	Cualitativo	No. Est.	Porcentaje
Domina los aprendizajes (DA).	9,00 – 10,00	DA	0	0.00%
Alcanza los aprendizajes (AA).	7,00 – 8,99	AA	1	33.33%

Nota. Interfaz 'Trimestre 2' que se despliega tras ingresar a la interfaz notas, mostrando la tabla de notas como la tabla dinámica de Equivalencias.

Figura 29

Interfaz 'Notas Finales' para la Gestión de Notas del Docente

NOTAS FINALES

Nomina de Estudiantes	Trimestre 1						Trimestre 2						Trimestre 3						Promedio Final Anual									
	Promedio Trimestre		Promedio Ponderado		Promedio Trimestre		Promedio Ponderado		Promedio Trimestre		Promedio Ponderado		Total promedio ponderado		Evaluacion Final		Evaluacion Final 10%		Promedio		Supletorio		Promedio Final		Estado		Cualitativo	
Álvarez Nunez Arturo Jesus	7.2	2.16	7.43	2.23	5.59	1.68	6.07	10	1.00	7.07	10	8.00	Aprue	AA														
Garcia Martinez Diego Miguel	5.25	1.57	6.73	2.02	4.6	1.38	4.97	10	1.00	5.97	8.5	7.75	Aprue	AA														
Martinez Sanchez Valentina Adriana	7.78	2.33	8.23	2.47	7.92	2.38	7.18	9	0.90	8.08	8.5	7.75	Aprue	AA														

GUARDAR NOTAS

Equivalencia	Rangos	Cualitativo	No. Est.	Porcentaje
Domina los aprendizajes (DA).	9,00 – 10,00	DA	0	0.00%
Alcanza los aprendizajes (AA).	7,00 – 8,99	AA	3	100.00%

Nota. Interfaz 'Notas Finales' que se despliega tras ingresar a la interfaz notas, mostrando la tabla de notas finales como la tabla dinámica de Equivalencias.

Dentro de las interfaces previas, como Trimestre 1, Trimestre 2, Trimestre 3 y Notas Finales, se incluye un botón específico que facilita la tarea de guardar las notas ingresadas en nuestra base de datos, en las respectivas tablas relacionadas a cada estudiante. Este proceso de almacenamiento se lleva a cabo mediante un controlador y un modelo que validan la información, asegurándose de que los campos requeridos estén completos y actualizando las notas si es necesario.

Dentro del menú desplegable del usuario docente se incorpora la interfaz del "Perfil Docente". Esta interfaz ha sido diseñada con la finalidad de presentar de manera global la información de la docente almacenada en nuestra base de datos. Se exhiben datos como el nombre completo del docente, su nacionalidad, correo electrónico, teléfono fijo, teléfono móvil y

número de cédula. Esta disposición no solo ofrece una visión integral del perfil del docente, sino que también sirve como un mecanismo de validación para garantizar la precisión y actualización de la información en nuestra base de datos.

Figura 30

Interfaz 'Ver Perfil' para la Gestión de Docente



Nota. Interfaz 'Perfil Usuario' que muestra la información relevante del docente.

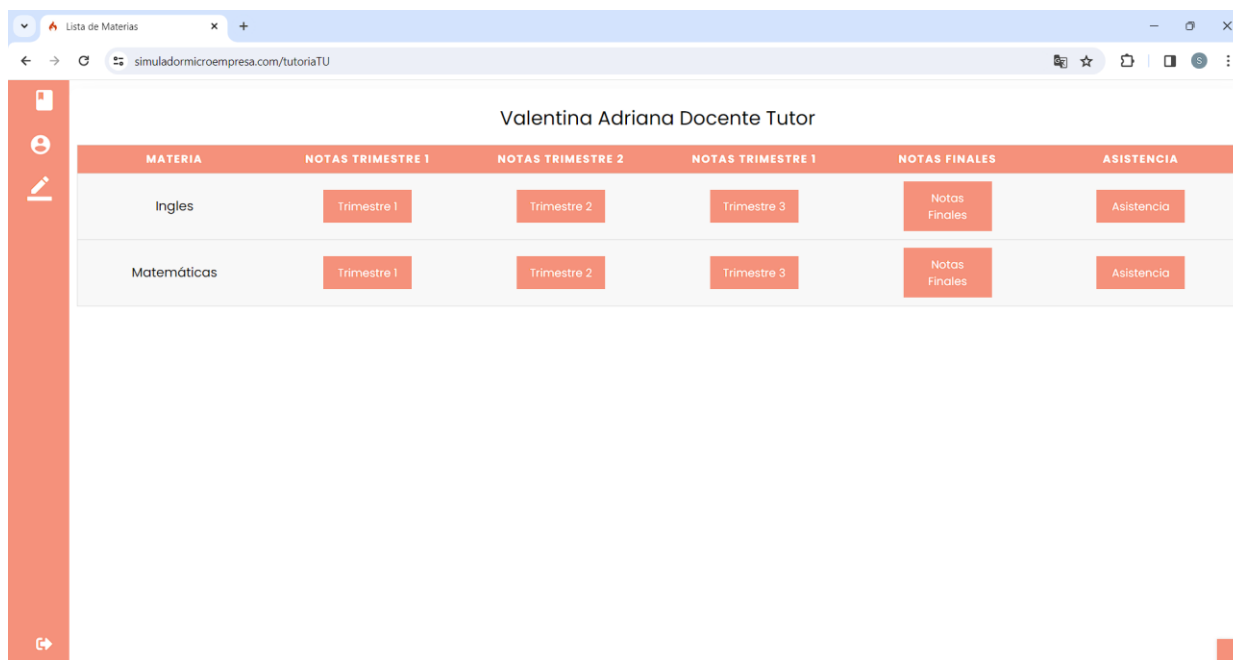
Dentro del menú desplegable del usuario docente, se incluye la interfaz denominada "Ver Tutoría", la cual presenta una tabla detallada del curso al que el docente ha sido asignado como tutor. Esta herramienta permite monitorear de manera constante el estado académico de los estudiantes bajo su tutela, mostrando las materias asociadas a todo el curso en el que el docente funge como tutor. En esta tabla, se incorporan botones que enlazan directamente a las secciones correspondientes a cada trimestre (1, 2 y 3), así como a las notas finales y asistencia, todas diferenciadas por materia.

En cuanto a los trimestres y las notas finales, se presenta una tabla de notas con la particularidad de que los campos no son editables en este apartado. Esta característica se aplica a los trimestres 1, 2 y 3, resaltando también el nombre del docente a cargo de dicha tutoría.

En el apartado de asistencia, al seleccionar esta opción, se redirige a una interfaz que verifica y muestra las asistencias registradas en la base de datos, indicando la fecha de asistencia y la materia correspondiente. Este enfoque busca facilitar a los docentes la supervisión general de la asistencia de sus representados, garantizando un seguimiento efectivo para asegurar la asistencia puntual a las clases.

Figura 31

Interfaz 'Ver Tutoría' para la Gestión de Docente



Valentina Adriana Docente Tutor					
MATERIA	NOTAS TRIMESTRE 1	NOTAS TRIMESTRE 2	NOTAS TRIMESTRE 3	NOTAS FINALES	ASISTENCIA
Inglés	Trimestre 1	Trimestre 2	Trimestre 3	Notas Finales	Asistencia
Matemáticas	Trimestre 1	Trimestre 2	Trimestre 3	Notas Finales	Asistencia

Nota. Interfaz 'Ver Perfil' que muestra una tabla con las materias y notas que tiene asociado el docente tutor a su cargo.

Figura 32

Interfaz 'Ver Tutoría-Trimestre 1' para el docente tutor

Inglés Decimo A
Valentina Adriana Martínez Samuel

TRIMESTRE 1

Nómina de Estudiantes	Trimestre No 1										Evaluación Trimestral					Promedio final Trimestre 1	Cualitativo				
	Actividades disciplinares o interdisciplinares- Individuales					Actividades disciplinares o interdisciplinares- Grupales					Formativas o portafolio Total Evaluaciones 80%	Proyecto	Proyecto 5%	Estructurada	Estructurada 5%						
	INSUMO 1	INSUMO 2	INSUMO 3	INSUMO 4	INSUMO 5	Promedio	Promedio 45%	INSUMO 1	INSUMO 2	INSUMO 3								INSUMO 4	INSUMO 5	Promedio	Promedio 45%
Álvarez Nunez Arturo Jesus	10	10	5	10	10	9.00	4.05	5	6	5	5	4	5.00	2.25	6.30	10	0.50	8	0.40	7.20	AA
García Martínez Diego Miguel	1	5	6	8	9	5.80	2.61	5	6	2	6	2	4.20	1.89	4.50	10	0.50	5	0.25	5.25	PA
Martínez Sánchez Valentina Adriana	8	6	7	8	9	7.60	3.42	4	10	9	6	10	7.80	3.51	6.93	8	0.40	9	0.45	7.78	AA

Nota. Interfaz 'Ver Tutoría-Trimestre 1' que muestra la información de todos los estudiantes matriculados en esta clase, el nombre del docente que dicta la materia y las notas.

Figura 33

Interfaz 'Ver Tutoría-Trimestre 2' para el docente tutor

Inglés Decimo A
Valentina Adriana Martínez Samuel

TRIMESTRE 2

Nómina de Estudiantes	Trimestre No 2										Evaluación Trimestral					Promedio final Trimestre 2	Cualitativo				
	Actividades disciplinares o interdisciplinares- Individuales					Actividades disciplinares o interdisciplinares- Grupales					Formativas o portafolio Total Evaluaciones 80%	Proyecto	Proyecto 5%	Estructurada	Estructurada 5%						
	INSUMO 1	INSUMO 2	INSUMO 3	INSUMO 4	INSUMO 5	Promedio	Promedio 45%	INSUMO 1	INSUMO 2	INSUMO 3								INSUMO 4	INSUMO 5	Promedio	Promedio 45%
Álvarez Nunez Arturo Jesus	10	5	5	5	10	7.00	3.15	6	6	10	10	10	8.40	3.78	6.93	10	0.50	0	0.00	7.43	AA
García Martínez Diego Miguel	6	6	5	5	5	5.40	2.43	10	10	10	10	5	9.00	4.05	6.48	5	0.25	0	0.00	6.73	PA
Martínez Sánchez Valentina Adriana	8	9	7	5	9	7.60	3.42	8	9	9	9	9	8.80	3.96	7.38	8	0.40	0	0.00	7.78	AA

Nota. Interfaz 'Ver Tutoría-Trimestre 2' que muestra la información de todos los estudiantes matriculados en esta clase, el nombre del docente que dicta la materia y las notas.

Figura 34

Interfaz 'Ver Tutoría-Trimestre 3' para el docente tutor

Inglés Decimo A
Valentina Adriana Martínez Samuel
TRIMESTRE 3

Nómina de Estudiantes	Actividades disciplinares o interdisciplinares- Individuales										Actividades disciplinares o interdisciplinares- Grupales					Evaluación Trimestral			Promedio final Trimestre 1	Calificativo	
	Insumos					Insumos					Formativas o portafolio Total Evaluaciones 80%	Proyecto	Proyecto 5%	Estructurada							
	INSUMO 1	INSUMO 2	INSUMO 3	INSUMO 4	INSUMO 5	Promedio	Promedio 45%	INSUMO 1	INSUMO 2	INSUMO 3					INSUMO 4	INSUMO 5	Promedio	Promedio 45%			
Álvarez Nunez Arturo Jesus	10	5	2	5	5	5.40	2.43	5	6	6	6	6	5.80	2.61	5.04	6	0.30	5	0.25	5.59	PA
García Martínez Diego Miguel	10	5	6	6	6	6.60	2.97	5	6	2	2	2	3.40	1.53	4.50	2	0.10	0	0.00	4.60	PA
Martínez Sánchez Valentina Adriana	7	8	9	9	3	7.20	3.24	9	9	9	9	6	8.40	3.78	7.02	9	0.45	9	0.45	7.92	AA

Nota. Interfaz 'Ver Tutoría-Trimestre 3' que muestra la información de todos los estudiantes matriculados en esta clase, el nombre del docente que dicta la materia y las notas.

Figura 35

Interfaz 'Ver Tutoría-Notas Finales' para el docente tutor

Matemáticas Decimo A
Juan Stuart Rodríguez Martínez
NOTAS FINALES

Nómina de Estudiantes	Trimestre 1		Trimestre 2		Trimestre 3		Total promedio ponderado	Evaluación Final	Evaluación Final 10%	Promedio Final Anual				
	Promedio Trimestre	Promedio Ponderado	Promedio Trimestre	Promedio Ponderado	Promedio Trimestre	Promedio Ponderado				Promedio	Supletorio	Promedio Final	Estado	Cualitativo
Álvarez Nunez Arturo Jesus	7.8	2.34	0	0.00	0	0.00	2.34	0	0.00	2.34	0	0.00	No Ap	NA
García Martínez Diego Miguel	6.49	1.95	0	0.00	0	0.00	1.95	0	0.00	1.95	0	0.00	No Ap	NA
Martínez Sánchez Valentina Adriana	7.24	2.17	8.05	2.42	8.24	2.47	7.06	10	1.00	8.06	0	0.00	No Ap	NA

Nota. Interfaz 'Ver Tutoría-Notas Finales' que muestra la información de todos los estudiantes matriculados en esta clase, el nombre del docente que dicta la materia y las notas.

Figura 36

Interfaz 'Asistencia' para el docente tutor

Matemáticas Decimo A
Juan Stuart Rodríguez Martínez

Asistencia

Valores: 1=ASISTENCIA 0= FALTA

Fecha de la novedad	Valor	Materia
2023-11-27	1.00	Matemáticas
2023-11-27	1.00	Matemáticas
2023-11-27	1.00	Matemáticas

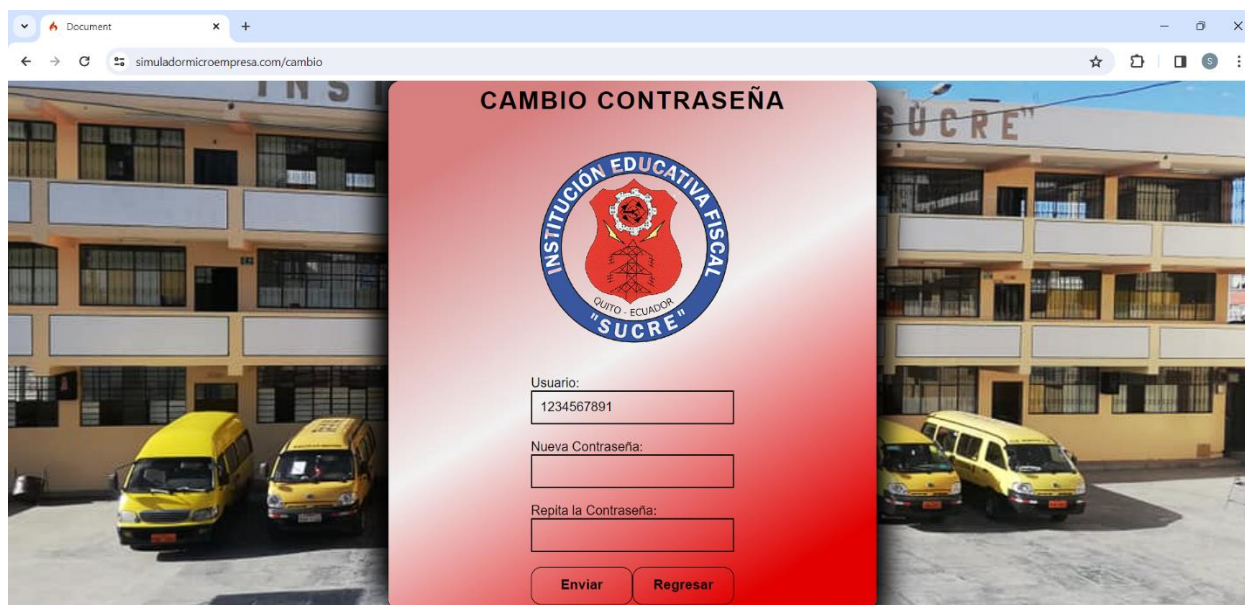
Materia	Asistencias	Faltas
Matemáticas	3	0

Nota. Interfaz 'Ver Tutoría-Asistencia' muestra la asistencia de todos los estudiantes de la materia elegida.

Finalmente, se encuentra la interfaz de cambio de contraseña, accesible desde la interfaz principal del docente. En esta sección, se presenta el número de cédula del docente que ha iniciado sesión, acompañado de las opciones de "Enviar" para confirmar el cambio de contraseña y "Regresar" para retornar a la página principal del usuario. Cabe destacar que esta interfaz únicamente solicita la contraseña nueva y su confirmación, asegurando un proceso simple y seguro para verificar y validar el cambio de contraseña, garantizando así que la nueva contraseña cumpla con los estándares de seguridad establecidos.

Figura 37

Interfaz 'Cambio de Contraseña' para el Docente



Document x +

simuladormicroempresa.com/cambio

CAMBIO CONTRASEÑA

INSTITUCIÓN EDUCATIVA FISCAL
QUITO, ECUADOR
"SUCRE"

Usuario:

Nueva Contraseña:

Repita la Contraseña:

Nota. Formulario que permite el cambio de contraseña al docente.

Conclusiones

A través de entrevistas con las autoridades, se identificaron detalladamente los requerimientos funcionales y no funcionales necesarios para el módulo de gestión docente. Este enfoque permitió un avance rápido y sin complicaciones en el proyecto, proporcionando un enfoque claro y preciso en la codificación para resolver las problemáticas planteadas. La identificación precisa de las necesidades tanto del personal docente como de los programadores facilitó el desarrollo orientado a soluciones efectivas.

El módulo de gestión académica de docentes fue desarrollado utilizando CodeIgniter 4, con una estructura basada en la arquitectura MVC. Asegurando la separación de la lógica, los datos y la interfaz, garantizando seguridad y confiabilidad tanto en la codificación como en la base de datos.

El módulo de gestión docente fue implementado en el sistema mediante un servidor web proporcionado por el Instituto Universitario “Vida Nueva”, que apoyó en la fase de pruebas. Esto permitió recibir retroalimentaciones valiosas y confirmar la funcionalidad y cumplimiento de los requerimientos tanto funcionales como no funcionales del módulo.

Recomendaciones

Se recomienda enfocarse en la implementación de medidas de seguridad sólidas, especialmente en la utilización de cifrados adecuados. Es esencial asegurarse de que las transacciones de datos, incluyendo contraseñas y notas de docentes y estudiantes, estén protegidas contra posibles amenazas, garantizando así la integridad y confidencialidad de la información sensible.

Dada la importancia de la participación de los usuarios en el desarrollo del módulo de gestión académica. La retroalimentación y la participación activa de los usuarios son esenciales para asegurar que el sistema se adapte a sus necesidades.

Para garantizar la obtención de información precisa y fiable en nuestra base de datos, se recomienda implementar un registro eficiente de datos proporcionado por los representantes legales. Este enfoque fortalecerá la veracidad de la información almacenada, respaldando así la toma de decisiones y la continuidad operativa del sistema.

Referencias

- Aristizábal Martínez, D. A. (2012). *PHP Básico y Practico*. Medellín: Instituto Tecnológico Metropolitano de Medellín.
- Arroyo Diaz, C. (2019). *Programacion de Java*. Argentina: USERS.
- Bastidas Logroño, D. J. (2019). *Implementación del Bootstrap como una Metodología ágil en la Web*. Santo Domingo.
- Carles, M. (2004). *Desarrollo de Aplicaciones Web*. Barcelona: Universitat Oberta de Catalunya - Uoc.
- Catherine, R. (2009). *Base De Datos*. México: Mcgraw Hill.
- Codeigniter. (25 de Junio de 2023). *Welcome to CodeIgniter4*. Obtenido de CodeIgniter:
https://www.codeigniter.com/user_guide/intro/index.html
- Comesaña, J. L. (2013). *Manual de JQuery en español*. Obtenido de Academia de software libre:
https://mundosica.github.io/tutorial_hispano_jQuery/
- Corporation, M. (2018). *Conceptos sobre bases de datos*. Obtenido de Microsoft Support.:
<https://support.microsoft.com/es-es/topic/conceptos-b%C3%A1sicos-sobre-bases-de-datos-a849ac16-07c7-4a31-9948-3c8c94a7c204>
- Duran Muñoz, F. (2008). *Desarrollo de Software Dirigido por Modelos*. Universidad Oberta de Catalunya.
- Gallardo, Y. (2000). *Aprender a Invetigar*. Colombia: ICFES.
- Gibert Ginesta, M., & Pérez Mora, O. (2020). *Base de Datos en PostgreSQL*. España: Univesitat Oberta de Catalunya.
- Hernández, R. (2006). *Metodología de la Investigación*. México: McGraw Hill.
- Herrera Ríos, E. (2012). *Arrancar con HTML5*. México: Alfaomega.

- IONOS, G. (01 de Marzo de 2023). *Digital Guide*. Obtenido de Servidores que es un servidor conceptos y definiciones: <https://www.ionos.es/digitalguide/servidores/know-how/que-es-un-servidor-un-concepto-dos-definiciones/>
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El Proceso Unificado de Desarrollo de Software*. Madrid: Pearson.
- Luján Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Club Universitario.
- Mannino, M. (2007). *Administracion de Base de Datos. Diseño y Desarrollo de Aplicaciones*. México: Mcgraw Hill.
- Martín Villalba, C. (2021). *Lenguaje de Programación*. UNED.
- Martínez Villalobos, G., Camacho Sánchez, G. D., & Biancha Gutiérrez, D. A. (2010). *Diseño de Framework Web para el Desarrollo Dinámico de Aplicaciones*. Colombia: Scientia Et Technica.
- Nixon, R. (2019). *Aprender Php, Mysql y Javascript con JQuery, Css y Html5*. Marcombo.
- Pavón Mestras, J. (Septiembre de 2008). *MVC-Modelo Vista Controlador*. Obtenido de Dep. Ingeniería del Software e Inteligencia Artificial Universidad Complutense Madrid.: <https://www.fdi.ucm.es/profesor/jpavon/poo/2.14.mvc.pdf>
- Proeduca. (22 de Agosto de 2022). *Framework: Qué es, para qué sirve y algunos ejemplos*. Obtenido de Unir: <https://unirfp.unir.net/revista/ingenieria-y-tecnologia/framework/>
- Wong, S. (2017). *Análisis y Requerimientos de Software*. Perú: Universidad Continental.

Anexos

Anexo 1

Presentación avances 1



Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 1 en la comunidad educativa.

Anexo 2

Presentación avances 2



Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 1 en la comunidad educativa.

Anexo 3

Presentación avances 3



Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 2 en la comunidad educativa.

Anexo 4

Presentación avances 4



Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 3 en la comunidad educativa.

Anexo 5*Presentación avances 5*

Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 4 en la comunidad educativa.

Anexo 6*Presentación avances 6*

Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 5 en la comunidad educativa.

Anexo 7*Presentación avances 7*

Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 5 en la comunidad educativa.

Anexo 8*Presentación avances 8*

Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 6 en la comunidad educativa.

Anexo 9*Presentación avances 9*

Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 7 en la comunidad educativa.

Anexo 10*Presentación avances 10*

Nota. La imagen captura el momento en que los docentes, el rector, los vicerrectores y los autores del proyecto presentando el avance número 7 en la comunidad educativa.