Universidad Tecnológica de Durango

Tecnologías de la Información

Aplicaciones Web

Reporte parcial 2

*“Desarrollo de Front end.”*

Proyecto

*“Gestor de constancias y planeaciones.”*

Alumnos:

* González Espino Marco Antonio
* Rivas García Gregorio
* Rosales García Matías Manuel
* Walkup Nevárez Cristopher Fano

3°B

Docente:

* Ing. Dagoberto Fiscal Gurrola, M.T.I.

Victoria de Durango, 03 de Octubre de 2024

*Tabla de Contenido*

[Tabla de ilustraciones 4](#_Toc181535797)

[Objetivos 5](#_Toc181535798)

[Objetivo general: 5](#_Toc181535799)

[Objetivo específico: 5](#_Toc181535800)

[Actividades de reporte 6](#_Toc181535801)

[Interfaces del proyecto integrador 6](#_Toc181535802)

[Interfaz de Login 6](#_Toc181535803)

[Interfaz de Inicio 6](#_Toc181535804)

[Interfaz de Gestión de Constancias para Docentes 7](#_Toc181535805)

[Interfaz de Gestión de Planeaciones para Docentes 7](#_Toc181535806)

[Interfaz del Panel de Administración de Constancias 8](#_Toc181535807)

[Interfaz del Historial de Constancias 8](#_Toc181535808)

[Interfaz del Panel de Solicitud de Constancias 9](#_Toc181535809)

[Interfaz del Menú para Subir Planeación 9](#_Toc181535810)

[Interfaz para Ver las Planeaciones de los Docentes 10](#_Toc181535811)

[Interfaz para Editar Planeación 10](#_Toc181535812)

[Liga del hospedaje del proyecto integrador 11](#_Toc181535813)

[Diagramas de la BD 12](#_Toc181535814)

[Entidad Relación 12](#_Toc181535815)

[Relacional 14](#_Toc181535816)

[Conclusiones: 16](#_Toc181535817)

[Bibliografía: 18](#_Toc181535818)

## Tabla de ilustraciones

[Ilustración 1 Interfaz 1 6](#_Toc181535819)

[Ilustración 2 Interfaz 2 6](#_Toc181535820)

[Ilustración 3 Interfaz 3 7](#_Toc181535821)

[Ilustración 4 Interfaz 4 7](#_Toc181535822)

[Ilustración 5 Interfaz 5 8](#_Toc181535823)

[Ilustración 6 Interfaz 6 8](#_Toc181535824)

[Ilustración 7 Interfaz 7 9](#_Toc181535825)

[Ilustración 8 Interfaz 8 9](#_Toc181535826)

[Ilustración 9 Interfaz 9 10](#_Toc181535827)

[Ilustración 10 Interfaz 10 10](#_Toc181535828)

[Ilustración 11 Entidad Relación 12](#_Toc181535829)

[Ilustración 12 Relacional 14](#_Toc181535830)

# Objetivos

## Objetivo general:

* Desarrollar aplicaciones Web con acceso a base de datos a través de lenguajes y herramientas especializadas para distribuirlas en internet

## Objetivo específico:

* Desarrollar aplicaciones Web dinámicas utilizando un lenguaje e implementar estándares de diseño Web para crear interfaces responsivas.

# Actividades de reporte

## Interfaces del proyecto integrador

## Interfaz de Login



Ilustración Interfaz 1

## Interfaz de Inicio

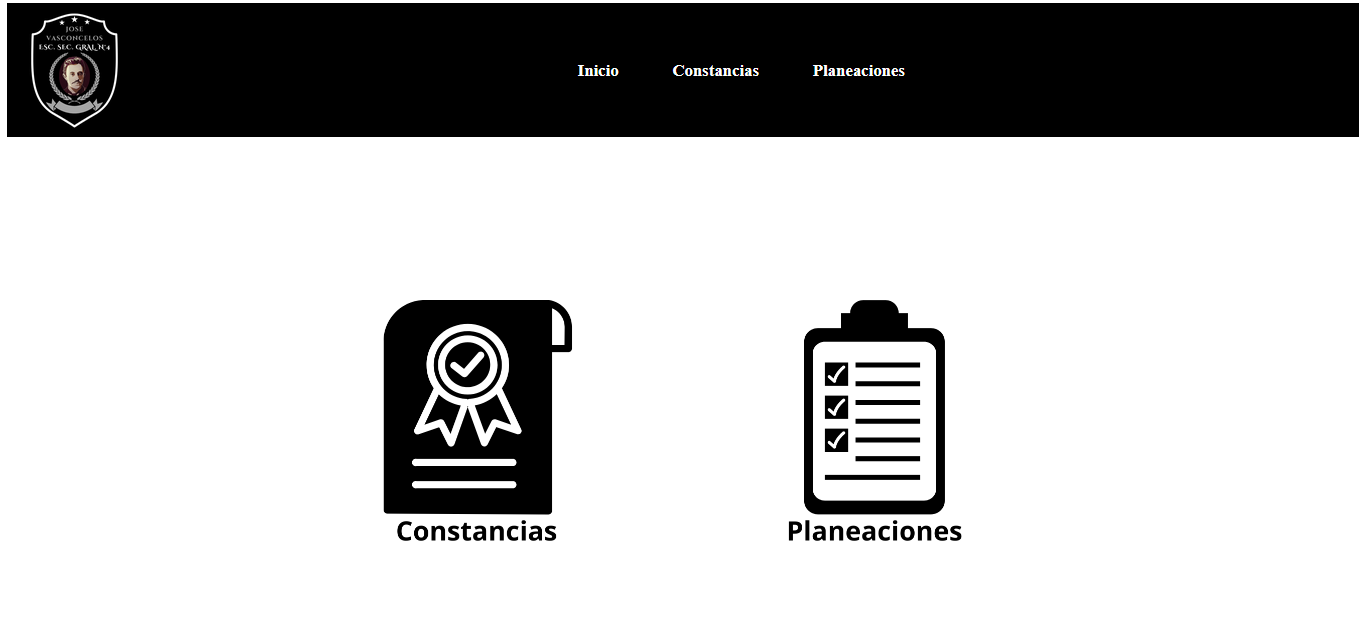


Ilustración Interfaz 2

## Interfaz de Gestión de Constancias para Docentes



Ilustración Interfaz 3

## Interfaz de Gestión de Planeaciones para Docentes

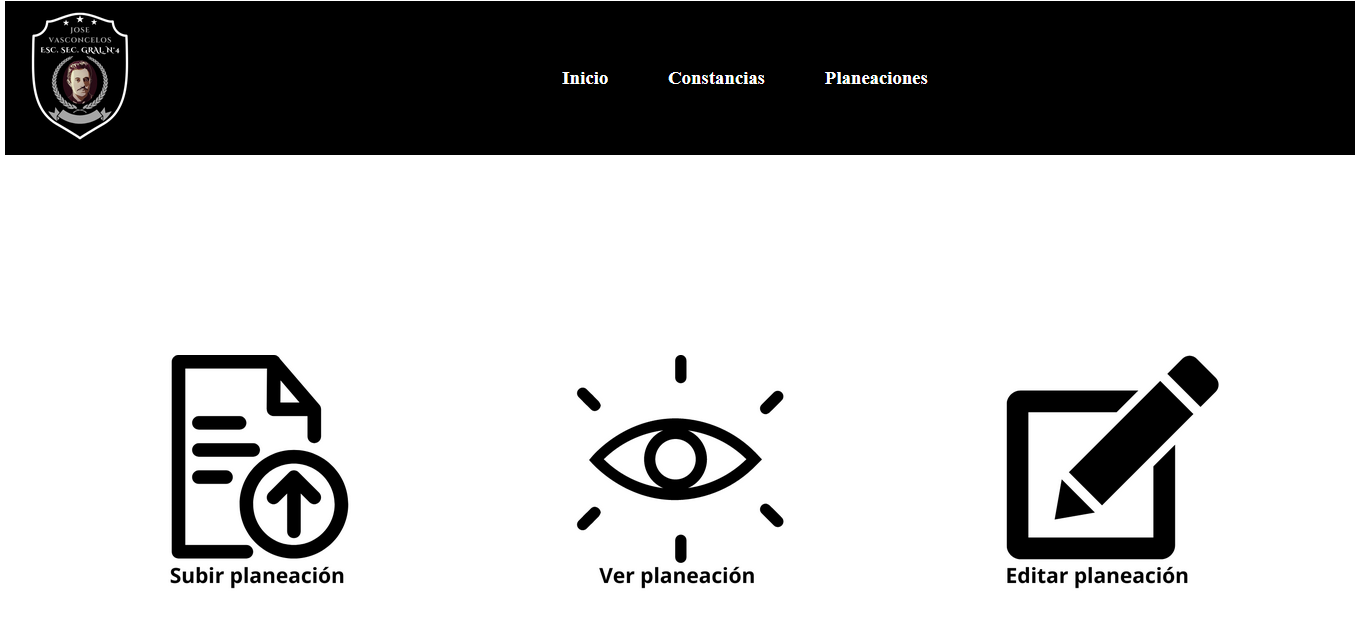


Ilustración Interfaz 4

## Interfaz del Panel de Administración de Constancias

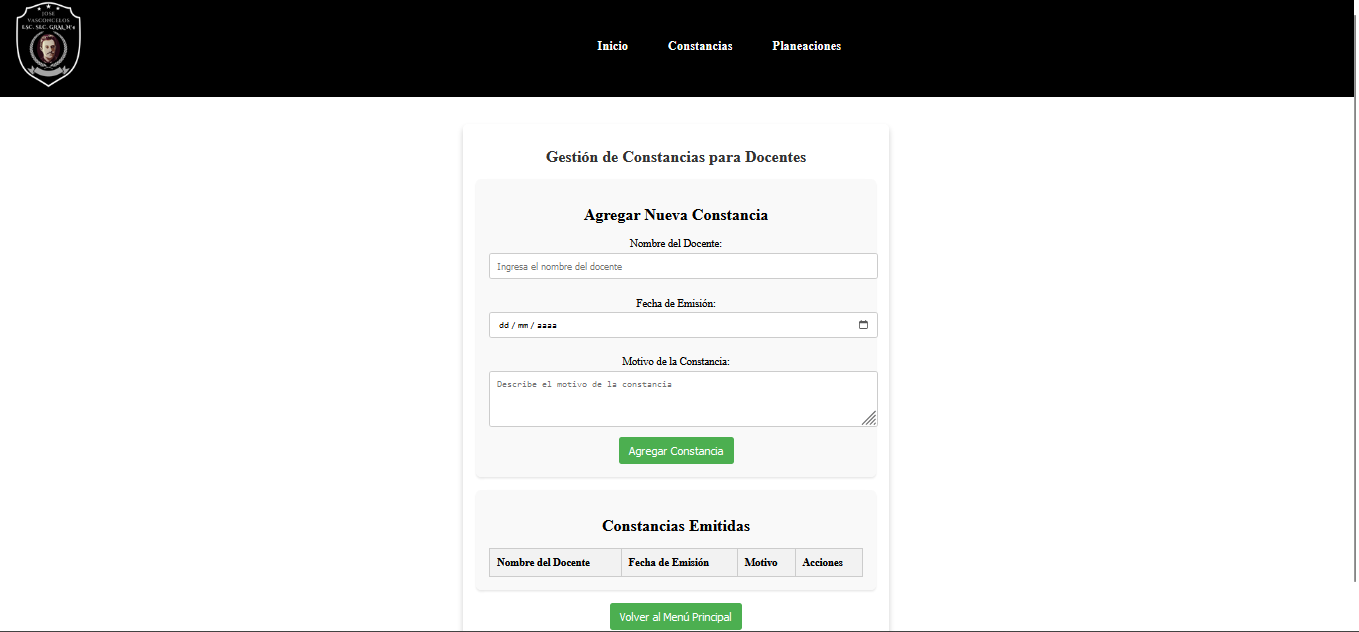


Ilustración Interfaz 5

## Interfaz del Historial de Constancias

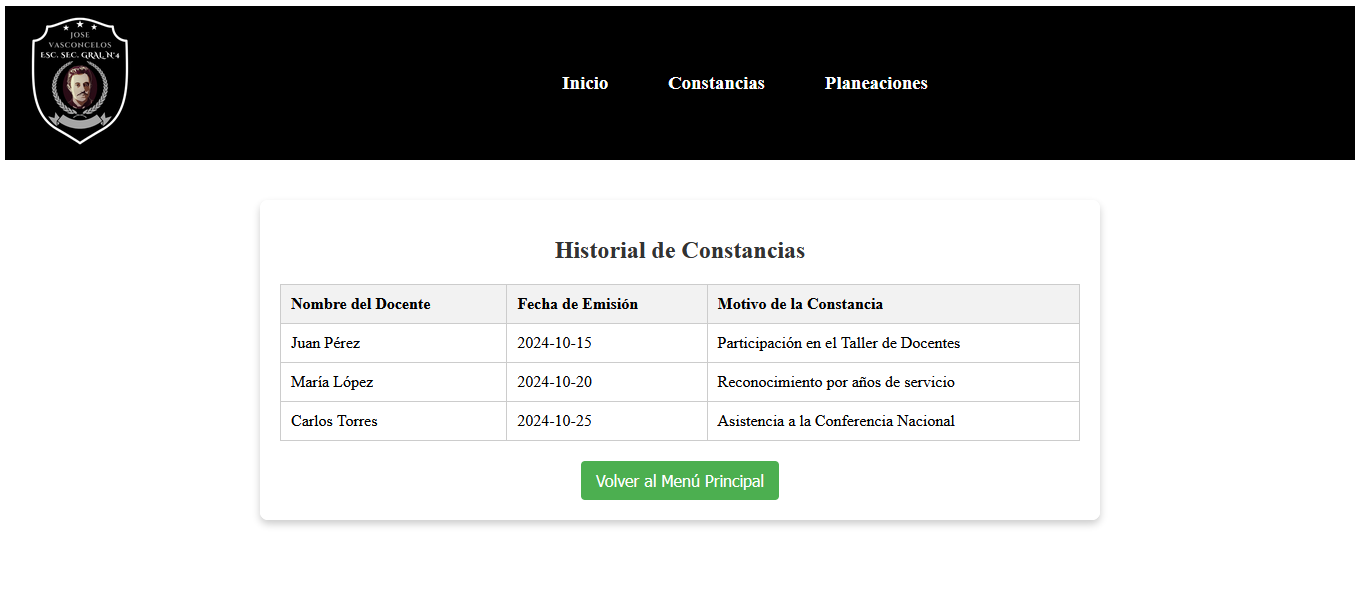


Ilustración Interfaz 6

## Interfaz del Panel de Solicitud de Constancias

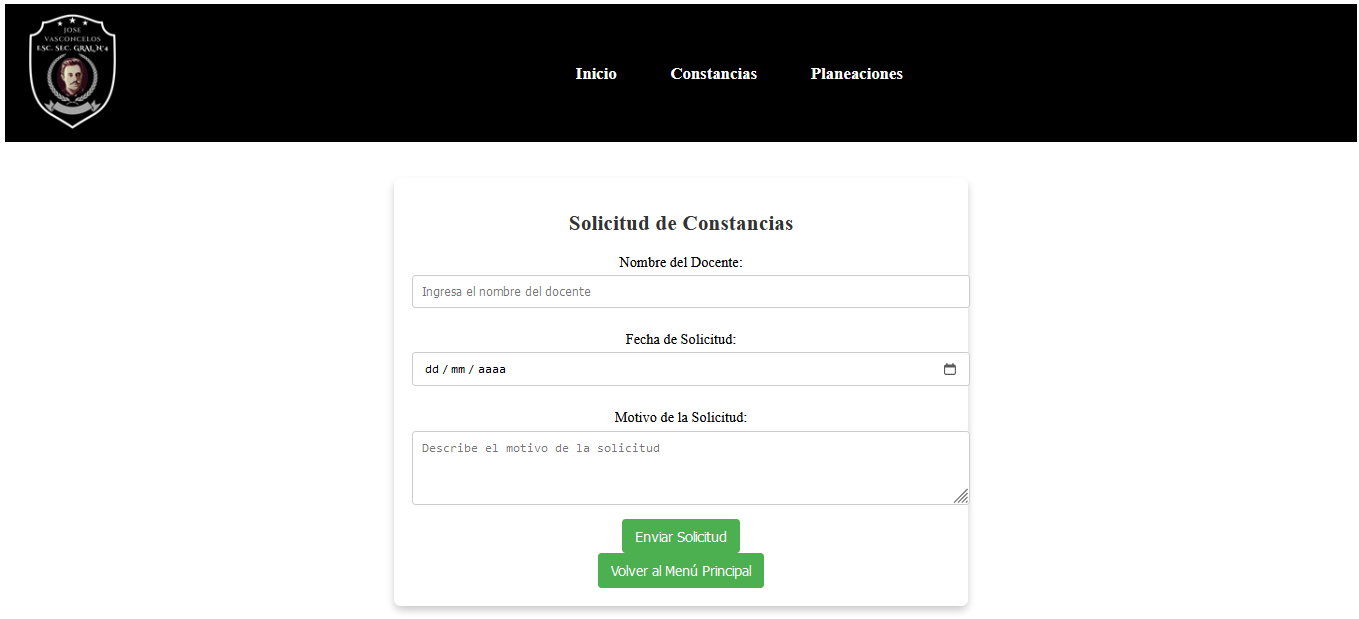


Ilustración Interfaz 7

## Interfaz del Menú para Subir Planeación

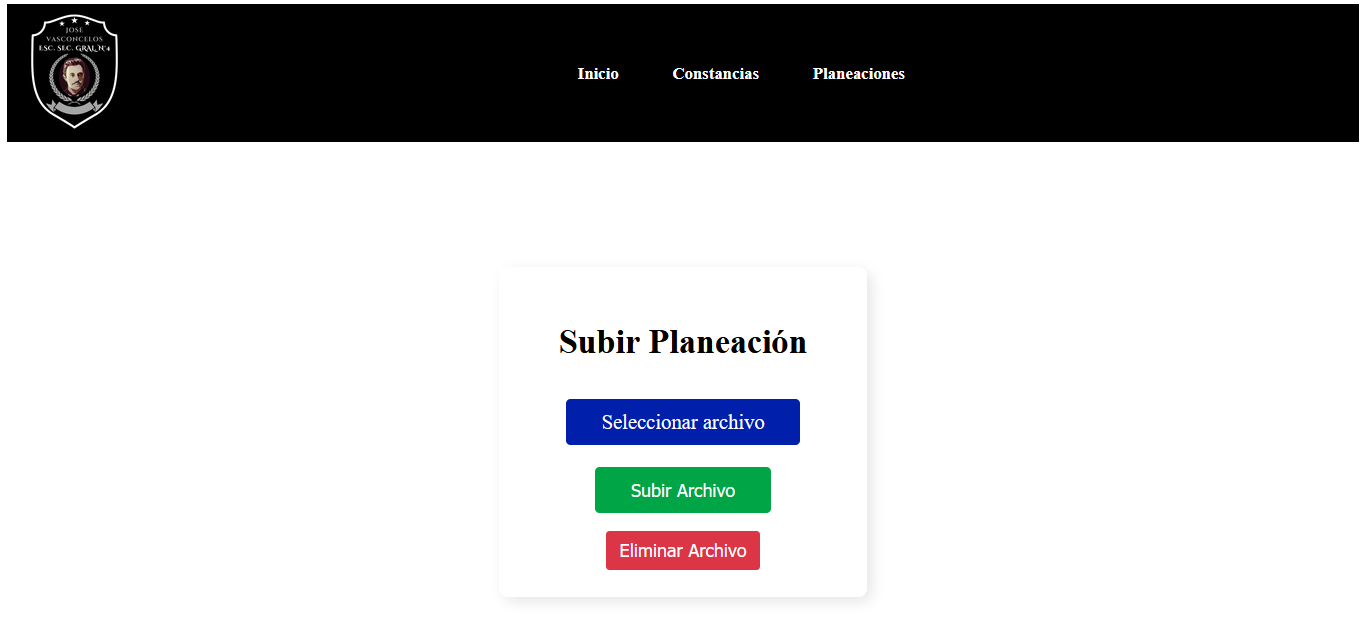


Ilustración Interfaz 8

## Interfaz para Ver las Planeaciones de los Docentes

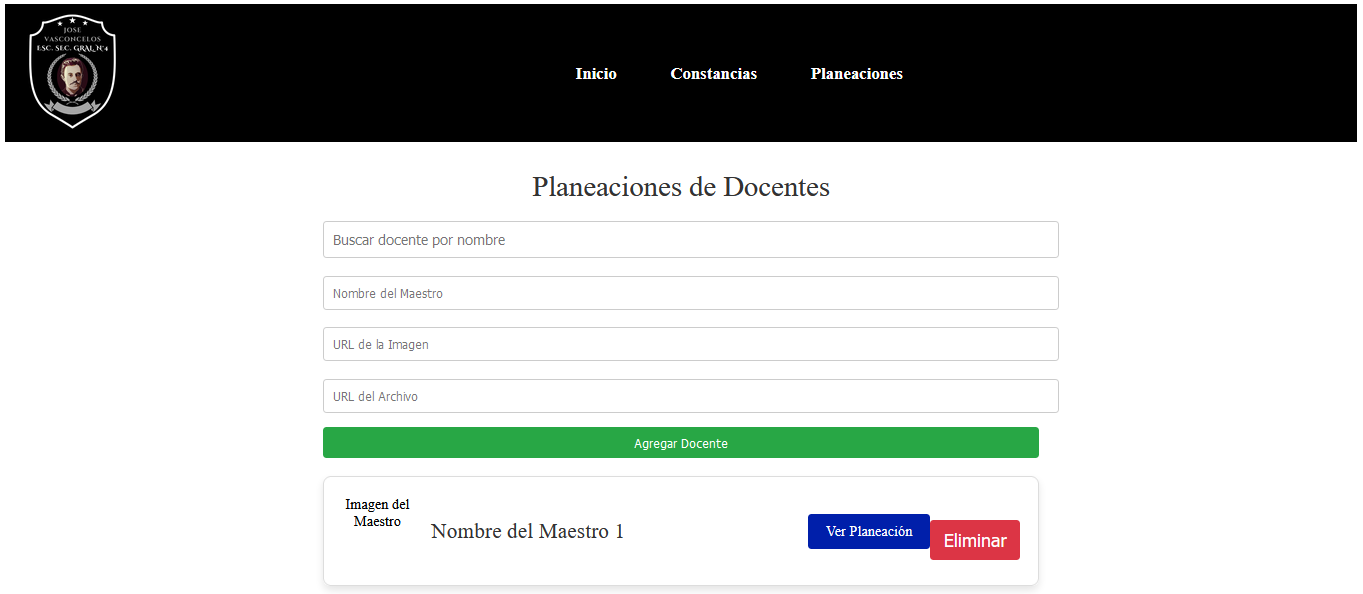


Ilustración Interfaz 9

## Interfaz para Editar Planeación

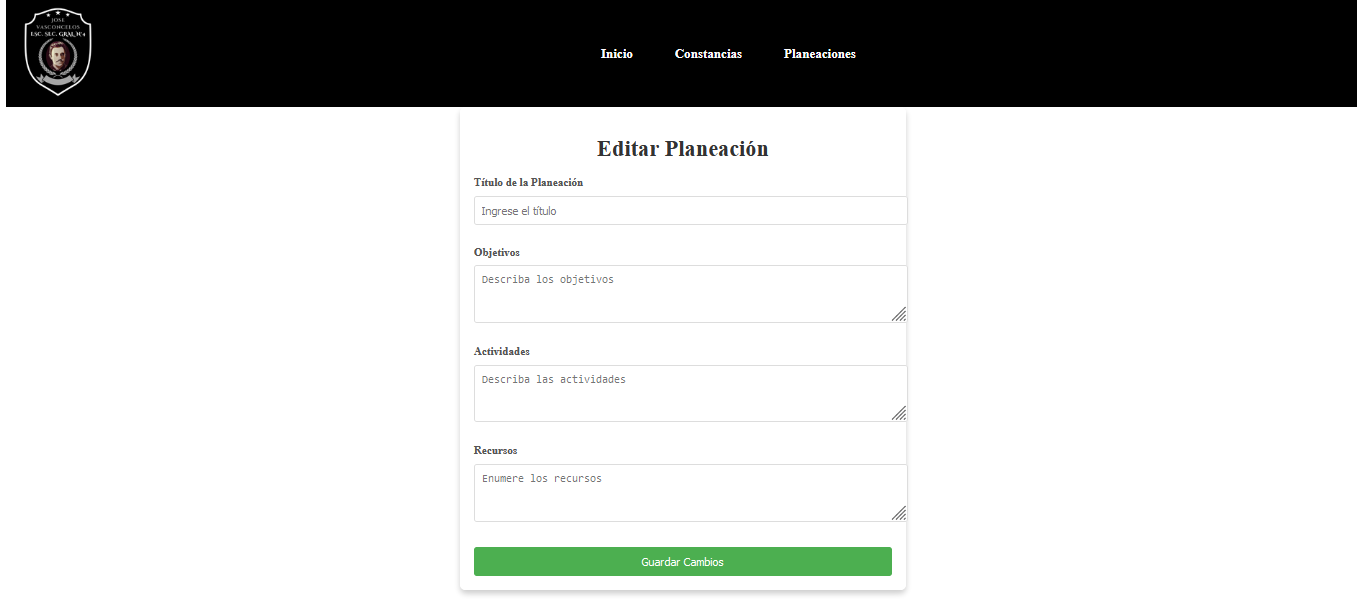


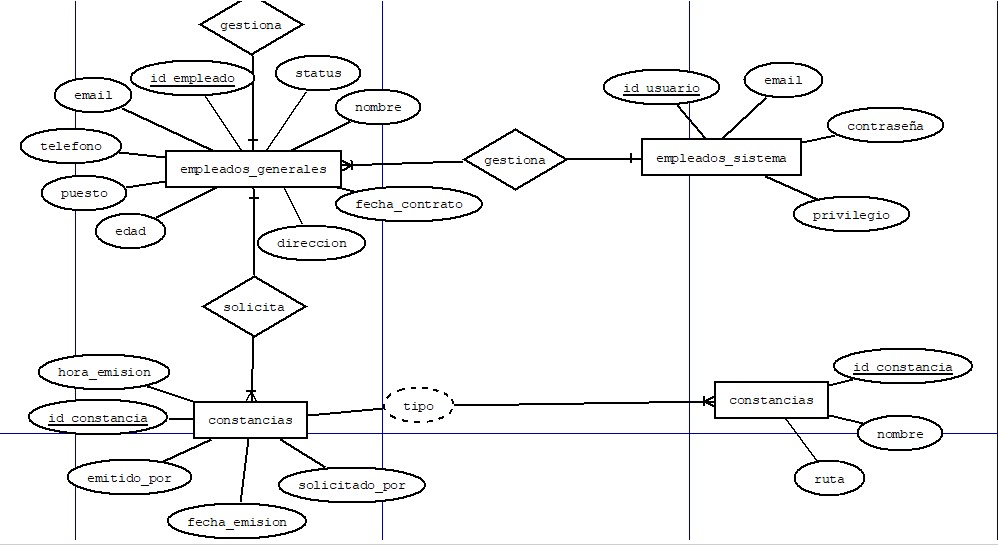
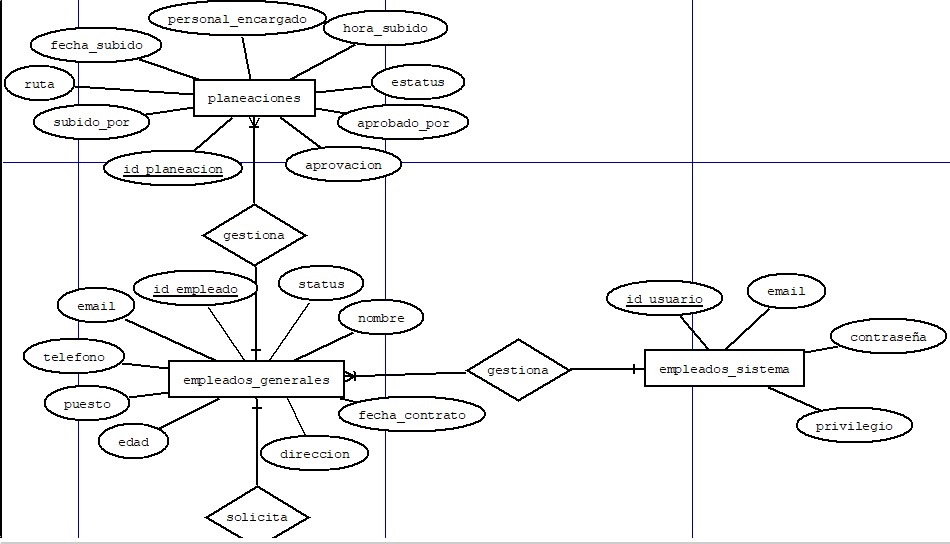
Ilustración Interfaz 10

## Liga del hospedaje del proyecto integrador

## Diagramas de la BD

## Entidad Relación

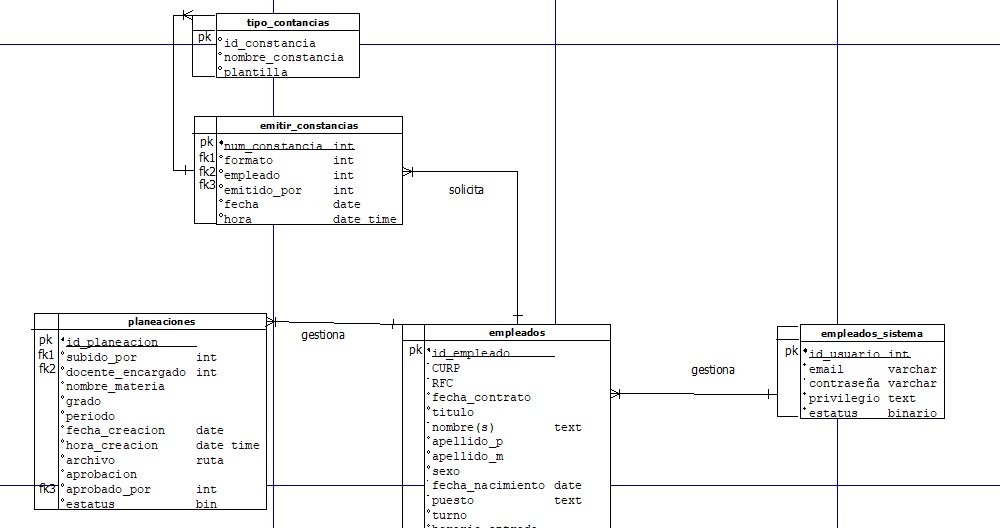
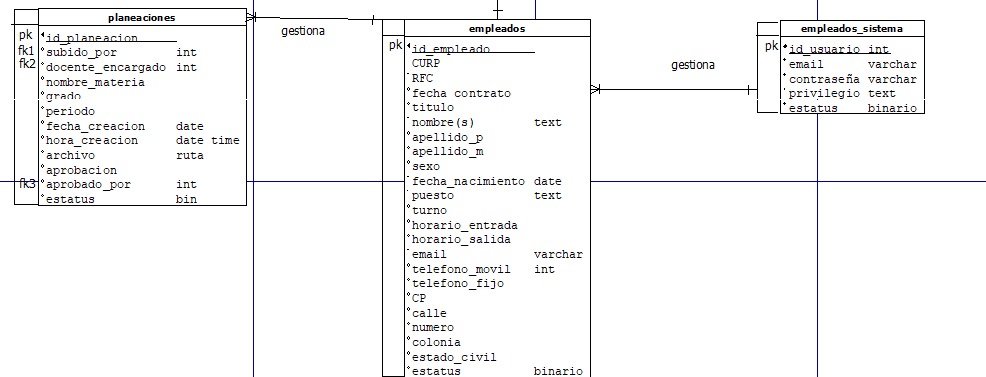
Ilustración Entidad Relación



En la imagen anterior se muestra el modelo Entidad Relación de la Base de Datos en el que se observan 4 entidades, las principales son: “planeaciones” y “constancias” que tienen una relación de muchos a uno con los empleados que son quienes pueden subir planeaciones y solicitar constancias, sin embargo, se observa otra entidad que son los administradores de la institución y son quienes gestionan a los empleados.

## Relacional

Ilustración Relacional



En la imagen se muestra el modelo lógico-relacional de la base de datos, compuesto por cuatro tablas principales: planeaciones, constancias, empleados, y empleados\_sistema. Estas tablas están relacionadas entre sí para gestionar el flujo de información dentro de la organización. La tabla “planeaciones” registra las planeaciones subidas por los empleados, con detalles sobre quién la subió, el docente encargado y la aprobación. Está vinculada a la tabla empleados a través de claves foráneas. La tabla “constancias” almacena las constancias solicitadas y emitidas por los empleados, relacionando tanto al solicitante como al emisor con la tabla empleados. La tabla “empleados” contiene los datos personales y roles de los empleados, quienes interactúan con las planeaciones y constancias, siendo la entidad central de gestión. Y la tabla “empleados\_sistema” administra los usuarios del sistema y sus privilegios, vinculándolos con sus respectivos registros en la tabla empleados.

# Conclusiones:

El proceso de configuración de nuestro ambiente de desarrollo web en Visual Studio Code ha sido exitoso y ha cubierto todas las herramientas esenciales para un desarrollo eficiente. Comenzamos con la instalación de Visual Studio Code, seguido por la instalación de Node.js y npm para ejecutar JavaScript en el servidor, y Git para el control de versiones. La elección de Git nos permite llevar un seguimiento detallado de los cambios en nuestro código, facilitando la colaboración y el manejo de versiones.

Además, agregamos extensiones clave como Live Server, Prettier y ESLint, que son fundamentales para optimizar nuestro flujo de trabajo. Live Server nos permite lanzar un servidor local y previsualizar nuestros cambios en tiempo real, mientras que Prettier se encarga de formatear nuestro código de manera automática, asegurando una presentación consistente y limpia. ESLint, por su parte, proporciona un análisis estático del código para identificar y corregir errores y malas prácticas, lo que resulta en un desarrollo más ágil y con menos errores.

La configuración incluyó la creación de un entorno de trabajo organizado, con una estructura de carpetas que facilita la navegación y el manejo de los archivos del proyecto. También personalizamos la interfaz de Visual Studio Code mediante la configuración de temas, lo que mejora la experiencia visual y la comodidad durante las largas sesiones de programación. Estas personalizaciones hacen que el entorno sea no solo funcional, sino también agradable de usar.

Realizamos pruebas de funcionamiento para verificar que toda la configuración funcionaba correctamente. Esto incluyó la ejecución de un servidor básico en Node.js y la visualización de una página HTML mediante Live Server, así como la creación de archivos de prueba en diferentes lenguajes de programación, como Python y C++. Estas pruebas nos permitieron asegurarnos de que las extensiones y configuraciones estaban correctamente implementadas, y que podíamos cambiar sin problemas entre distintos lenguajes y contextos de programación.

En paralelo, trabajamos en la creación de diagramas entidad-relación y lógico-relacional para nuestra base de datos. Los diagramas entidad-relación nos permitieron visualizar las relaciones entre las diferentes entidades que gestionamos, como constancias, planeaciones y empleados, facilitando así el diseño estructural de la base de datos. A través de esta representación gráfica, pudimos identificar las claves primarias y foráneas necesarias para garantizar la integridad de los datos.

Por otro lado, los diagramas lógico-relacionales nos ayudaron a traducir nuestro diseño conceptual a un modelo que pudiera ser implementado en un sistema de gestión de bases de datos. Esto incluyó definir las tablas, los atributos y las relaciones entre ellas, asegurando que la estructura de la base de datos se ajustara a las necesidades del proyecto. La correcta creación de estos diagramas es fundamental para establecer una base de datos eficiente que soporte nuestras aplicaciones web.

Gracias a esta configuración, ahora disponemos de un ambiente robusto y flexible que nos permitirá desarrollar aplicaciones web de manera eficiente y profesional. Las herramientas que instalamos garantizan un código limpio y nos ayudan a seguir buenas prácticas de desarrollo. La capacidad de trabajar con múltiples lenguajes, como HTML, Python y C++, en un solo entorno mejora nuestra productividad y facilita el aprendizaje continuo. Además, también aprendimos a mejorar la creación de bocetos utilizando Balsamiq, lo que nos será de gran utilidad para futuros proyectos.

-Walkup Nevárez Cristopher Fano.

-Rivas García Gregorio.

-González Espino Marco Antonio.

-Rosales García Matías Manuel.

Principio del formulario

Final del formulario

# Bibliografía:

* *Diagrama UML: Qué es, cómo hacerlo y ejemplos | Miro*. (s. f.). https://miro.com/. <https://miro.com/es/diagrama/que-es-diagrama-uml/>
* *Balsamiq: Fast, focused wireframing for teams and individuals | Balsamiq*. (s. f.). <https://balsamiq.com/>
* *Visual Studio Code - Code editing. Redefined*. (2021, 3 noviembre). <https://code.visualstudio.com/>
* Admin. (2024, 22 enero). *Instalación de Visual Studio Code: Guía Paso a Paso*. Informatica Tutoriales. <https://informaticatutoriales.com/instalacion-visual-studio-code/>
* Cristalab, T. (s. f.). *Prototipos, bocetos y wireframes con Balsamiq Mockups*. Cristalab. <https://www.cristalab.com/blog/prototipos-bocetos-y-wireframes-con-balsamiq-mockups-c61456l/>
* *Planeación S.E.N.L.* (s. f.). <http://planeacioneducativa.uienl.edu.mx:8044/SeguridadGlobal/Publico/Logon.aspx>
* *Download Python*. (s. f.). Python.org. <https://www.python.org/downloads/>
* *Git - Downloading Package*. (s. f.). <https://git-scm.com/downloads/win>