

Universidad de Carabobo  
Facultad Experimental de Ciencias y Tecnología  
Departamento de Computación  
Asignatura Desarrollo de Aplicaciones Web  
Profesor Javier Escobar

**Proyecto Final - Desarrollo de Aplicaciones Web**

**Integrantes:**

**Yhonka Machado, C.I: 27.718.080**

**Kevin Breto, C.I: 28.351.815**

13/04/2025

## Introducción

El presente informe detalla el proceso de desarrollo y los resultados obtenidos en el proyecto de creación de una aplicación web orientada a la gestión académica universitaria. Este proyecto se ha llevado a cabo utilizando un conjunto de tecnologías robustas y ampliamente adoptadas en la industria del desarrollo front-end y back-end.

En el lado del cliente (front-end), se ha empleado Vue.js, un framework progresivo de JavaScript conocido por su flexibilidad, rendimiento y facilidad de integración. Vue.js ha permitido la construcción de una interfaz de usuario dinámica y reactiva, facilitando la creación de componentes reutilizables y la gestión eficiente del estado de la aplicación.

Para la gestión de la base de datos y la lógica del lado del servidor (back-end), se ha optado por Firebase, una plataforma de desarrollo de aplicaciones móvil y web respaldada por Google. Firebase ofrece una amplia gama de servicios, incluyendo una base de datos NoSQL en tiempo real (Firestore), autenticación de usuarios y almacenamiento de archivos, lo que ha simplificado significativamente el desarrollo del back-end y ha permitido una rápida iteración.

Finalmente, para la creación de una interfaz de usuario se ha integrado Vuetify, un framework de componentes de Material Design para Vue.js. Vuetify proporciona una lista de componentes pre-construidos, personalizables y responsivos.

Este informe explorará en detalle la arquitectura de la aplicación desarrollada, las decisiones técnicas clave tomadas durante el proceso, los desafíos encontrados y las soluciones implementadas, así como los resultados finales.

## **Objetivos del Trabajo**

### **Objetivo principal**

El objetivo principal de este trabajo es la creación de una aplicación web orientada a la gestión académica universitaria, que permitirá gestionar estudiantes, asignaturas, matrículas y calificaciones.

### **Subobjetivos**

Inicialmente definimos siguientes objetivos/tareas que pretendemos cumplir durante en desarrollo:

- Analizar, diseñar e implementar la base de datos no relacional que dará soporte a todos los componentes de la aplicación y su integración.
- Analizar, diseñar e implementar el componente de registro y autenticación e los usuarios de la aplicación.
- Analizar, diseñar e implementar componentes que permitan añadir, editar y eliminar información de la base de datos.

## Planificación del Trabajo

### Tecnologías y recursos

Se utilizo las siguientes tecnologías:

**JavaScript:** (JS) lenguaje de programación en que se basara la solución del proyecto. Es un lenguaje de programación de alto nivel, interpretado y dinámico, fundamental para el desarrollo web moderno. Originalmente diseñado para añadir interactividad a las páginas web en los navegadores, JavaScript permite hacer que las páginas web sean más que simples documentos estáticos. Con JS, se pueden crear efectos visuales, manejar eventos del usuario (clics, movimientos del ratón, etc.), interactuar con el servidor para enviar y recibir datos, manipular el contenido de la página (DOM), y construir interfaces de usuario complejas y dinámicas.

**Vue:** Vue.js (a menudo abreviado como Vue) es un framework progresivo de JavaScript utilizado para construir interfaces de usuario interactivas y aplicaciones web de una sola página (SPA). En esencia, Vue se centra en la capa de la vista de una aplicación, lo que significa que se encarga principalmente de la interacción y presentación de los datos al usuario. Sin embargo, también es lo suficientemente flexible como para integrarse con otras bibliotecas y herramientas, lo que lo convierte en una solución viable para construir aplicaciones completas.

**Vuetify:** es un framework de componentes de interfaz de usuario (UI) para Vue.js que sigue las especificaciones de Material Design, te proporciona una amplia colección de componentes preconstruidos, personalizables y listos para usar (como botones, tarjetas, formularios, barras de navegación, etc.) que te permiten construir interfaces de usuario atractivas, responsivas y coherentes de manera rápida y eficiente.

**Firebase:** es una plataforma de desarrollo de aplicaciones móvil y web integral, respaldada por Google, que ofrece una amplia gama de herramientas y servicios para ayudar a los desarrolladores a construir, lanzar y hacer crecer sus aplicaciones de manera rápida y eficiente. Firebase simplifica el desarrollo del back-end, proporcionando soluciones listas para usar para muchas funcionalidades comunes que las aplicaciones necesitan. Esto permite a los desarrolladores centrarse más en la experiencia del usuario y la lógica del front-end, en lugar de preocuparse por la infraestructura del servidor.

**GitHub:** Plataforma web que utilizamos para el almacenamiento, gestión de las versiones del código del proyecto, gestión de configuración del proyecto y documentación. GitHub es una plataforma web basada en el sistema de control de versiones Git y desarrollado en Ruby on Rails y Erlang por GitHub, inc. La plataforma es

gratuita para proyectos de código abierto y desde 2019 para pequeños proyectos privados.

## **Análisis de Requisitos**

### **Usuarios**

A continuación, se identifican los tipos de usuarios que interactúan con el sistema.

**Usuario no registrado:** un usuario que está interesado en la posibilidad de poder registrarse en el sistema para acceder a sus funcionalidades.

**Usuario registrado:** un usuario del sistema que puede acceder directamente a todas las funcionalidades relacionadas con añadir, editar y eliminar información de la base de datos.

### **Requisitos funcionales del sistema**

Se identifican una serie de requisitos funcionales. Para cada uno de los requisitos se especifican los usuarios que hacen uso de estas funcionalidades: Usuario registrado (**UR**), usuario no registrado (**UNR**).

- Inicial sección (**UR**)
- Registrarse como usuario (**UNR**).
- Página principal (**UNR**) (**UR**)
- Registrar estudiante (**UR**)
- Consultar estudiante (**UR**)
- Modificar estudiante (**UR**)
- Eliminar estudiante (**UR**)
- Registrar asignatura (**UR**)
- Consultar asignatura (**UR**)
- Modificar asignatura (**UR**)
- Eliminar asignatura (**UR**)
- Matricular estudiante (**UR**)
- Desmatricular estudiante (**UR**)
- Registrar calificación (**UR**)
- Consultar calificación (**UR**)
- Modificar calificación (**UR**)

- Eliminar calificación **(UR)**
- Generar reporte **(UR)**

### **Requisitos no funcionales del sistema**

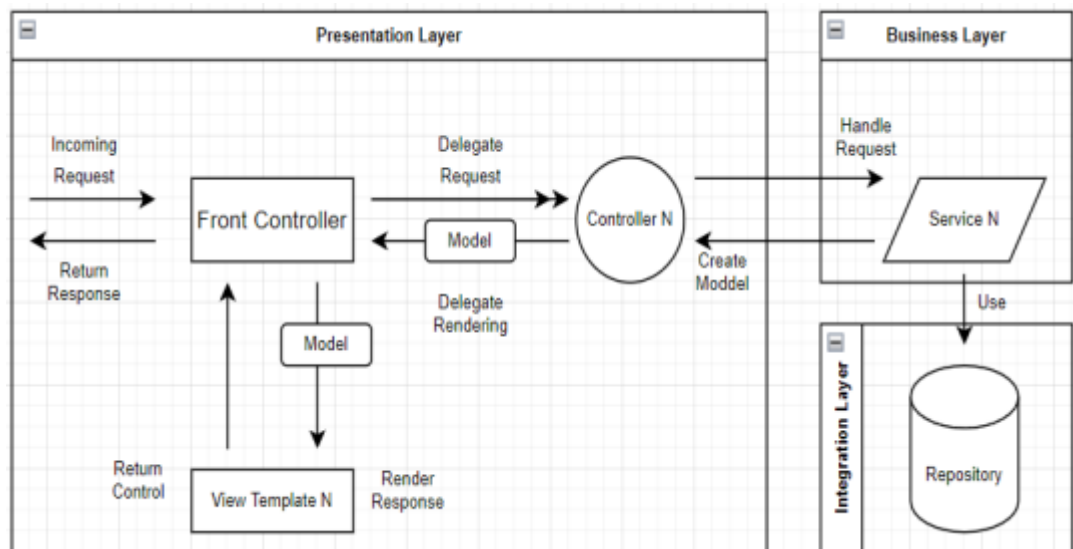
- **Usabilidad:** Interfaz intuitiva, clara y sencilla.
- **Seguridad:** Protección de datos personales, acceso restringido mediante autenticación.
- **Rendimiento:** Capacidad del sistema para funcionar sin bloqueos bajo condiciones de alta demanda.
- **Escalabilidad:** Arquitectura que permita adaptarse al crecimiento de usuarios y datos.

### **Diseño del Proyecto**

En nuestro proyecto optaremos por la arquitectura de tres capas, es una arquitectura cliente-servidor y que separa las funciones de presentación, procesamiento y almacenamiento de datos.

En esta arquitectura se define las siguientes capas/niveles en la aplicación:

- Capa de presentación (Presentation Layer) es la capa encargada de generar la interfaz de usuario en función de las acciones llevadas a cabo por el mismo.
- Capa de Negocio (Business Layer) es la capa con la lógica que modela los procesos para atender las peticiones del usuario.
- Capa de integración de Datos (Integration Layer) es la capa encargada de hacer persistente toda la información, así como de suministrar y almacenar la información.

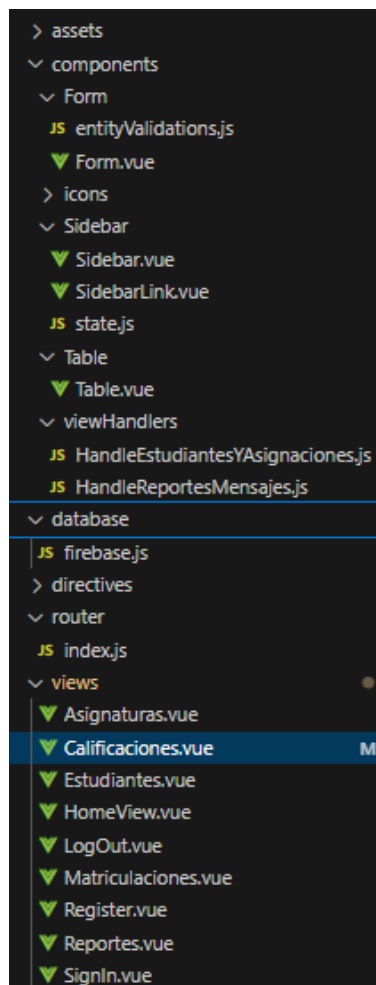


*Estructura de patrón MVC y flujo de un request dentro de la capa de presentación.*

## Estructura de ficheros de las vistas

Según los requisitos y la arquitectura a aplicada decidimos organizar la aplicación y desarrollarla en 6 carpetas principales:

- **Assets:** Contiene la base de los estilos aplicados.
- **Components:** Contiene los componentes Form, las validaciones de los formularios, iconos, sidebar y sus links a las vistas, tablas y vista de los reportes.
- **Database:** Contiene la inicialización y configuración de la base de datos firebase.
- **Router:** Contiene la creación de todas las rutas a las páginas.
- **Views:** Contiene todas las vistas que conforman el proyecto.

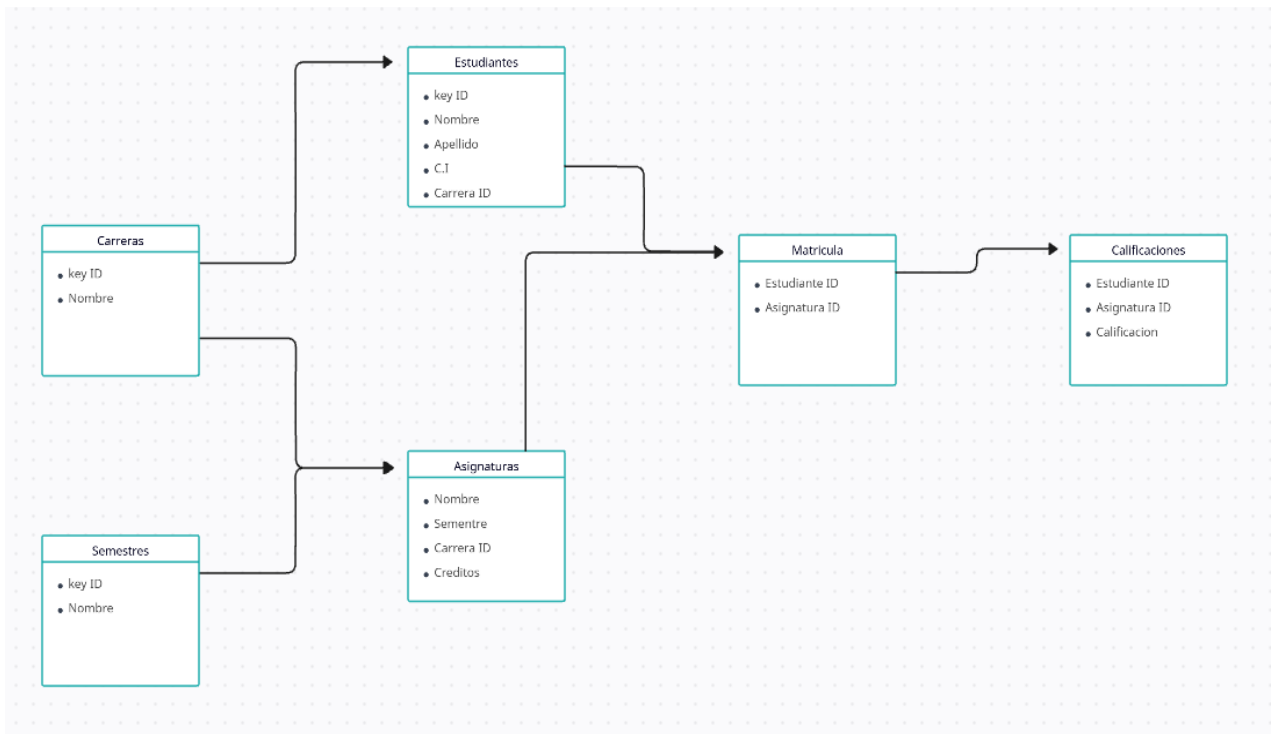




## Estructura de la Base de Datos

La base de datos que elegimos fue de tipo no relacional (también conocida como NoSQL) y fue firebase. Es un tipo de base de datos que no utiliza el modelo tabular de filas y columnas que se encuentra en las bases de datos relacionales tradicionales (SQL). En cambio, las bases de datos NoSQL emplean una variedad de modelos de datos optimizados para los requisitos específicos del tipo de datos que se están almacenando y consultando.

En el apartado siguiente se presenta el modelo de datos de la aplicación mediante un diagrama.

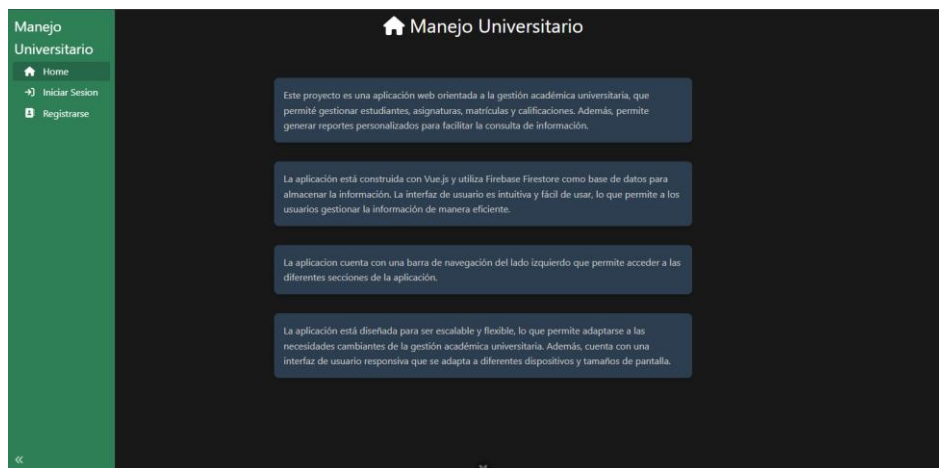


*Diagrama de la Base de Datos (Cada recuadro es una colección en la base de datos).*

## Desarrollo de los Componentes

Nuestro sitio web presenta información y contenido sobre objetos. Todas las validaciones de todos los campos la hacemos aquí y no en el controlador. Por ejemplo, la validación de los inputs, ya sea para controlar el formato, la dimensión o simplemente que no esté vacío lo hacemos en los componentes.

- **Form.vue:** Es el encargado de mostrar los inputs necesarios para cada página en la que se necesite un formulario, estos los recibe como props el componente.
- **EntityValidations.js:** Se encarga de validar los campos de los formularios.
- **Slidebar.vue:** Muestra la lista de navegación a las páginas y está ubicado a la izquierda de la pantalla.

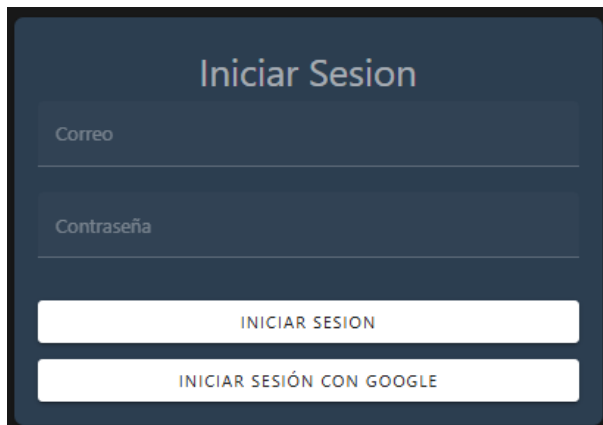


*Sin inicial sesión.*



*Con una sesión activa.*

- **SlidevarLink.vue:** Se encarga de redireccionar a la página que usuario dio click.
- **Table.vue:** Se utiliza para mostrar la información ya agregada a la base de datos, los datos que muestra y su headers es recibido como props, por cada linea de la tabla tiene dos botones de acciones para editar y eliminar.
- **SignIn.vue:** Muestra el login y sus botones para inicial sesión con sus credenciales o directamente con la cuenta de correo, se encarga de autenticar y validez si los datos agregados por el usuario están registrados en la base de datos.



**Iniciar Sesión**

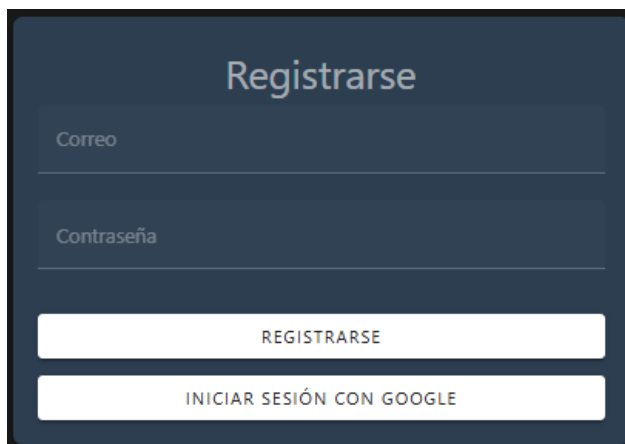
Correo

Contraseña

INICIAR SESION

INICIAR SESIÓN CON GOOGLE

- **Register.vue:** Muestra el formulario para registrar el usuario al sistema.



**Registrarse**

Correo


Contraseña

REGISTRARSE





INICIAR SESIÓN CON GOOGLE



- **Estudiantes.vue:** Muestra el formulario para agregar asignaturas usando el componente Form.vue y la tabla con la información ya agregada usando el componente Tabla.vue. Se valida que los campos nombre, apellido, cedula y carrera no estén vacíos y que el número de cedula de no excitan en la base de datos.

 Manejo de Estudiantes

AGREGAR

Nombre	Apellido	Cedula	Carrera	Acciones
alan	turin	1	Computacion	 
Yhonka	Machado	123456	Computacion	 

Items per page: 10
 1-2 of 2
 |< < > >|

- **Matriculaciones.vue:** Muestra el formulario para agregar asignaturas usando el componente Form.vue y la tabla con la información ya agregada usando el componente Tabla.vue. Se valida que los campos estudiante y asignatura no estén vacíos y que el estudiante junto a su asignatura no excita en la base de datos.

 Manejo de Matriculaciones

AGREGAR

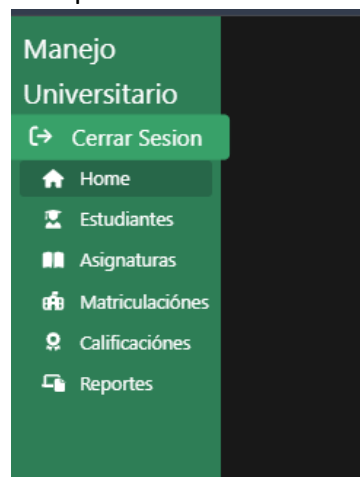
Estudiante	Asignaturas	Acciones
123456	Fundamentos de Algoritmos	 

Items per page: 10
 1-1 of 1
 |< < > >|

- **HomeView.vue:** Página principal que muestra información detallada de lo que es el proyecto.



- **LogOut.vue:** Muestra el botón para salir del sistema.



- **Reports.vue:** Muestra los porcentajes y tabla de información agregada de los estudiantes, asignaturas, carreras, matriculaciones y calificaciones.



## Matriculaciones (1)

Fundamentos de Algoritmos: cuenta con 1 estudiantes matriculados para este semestre.

Estudiante	Asignaturas
123456	Fundamentos de Algoritmos
Items per page: 10 1-1 of 1  < < > >	

## Calificaciones (1)

Fundamentos de Algoritmos: tiene un promedio de calificaciones de 18.

Estudiante	Asignatura	Calificacion
123456	Fundamentos de Algoritmos	18
Items per page: 10 1-1 of 1  < < > >		

## Asignaturas (5)

Computacion: cuenta con 18 creditos en total.

Biologia: cuenta con 8 creditos en total.

Matematicas: cuenta con 8 creditos en total.

Fisica: cuenta con 38 creditos en total.

Quimica: cuenta con 14 creditos en total.

Nombre	Semestre	Creditos	Carreras
Fundamentos de Algoritmos	2do semestre	10	Computacion
Fisica Cuantica 2	9no semestre	20	Fisica
Calculo 1	1er semestre	8	Biologia, Computacion, Fisica, Matematicas, Quimica
Fisica 1	1er semestre	4	Fisica
Quimica 1	1er semestre	6	Fisica, Quimica
Items per page: 10 1-5 of 5  < < > >			

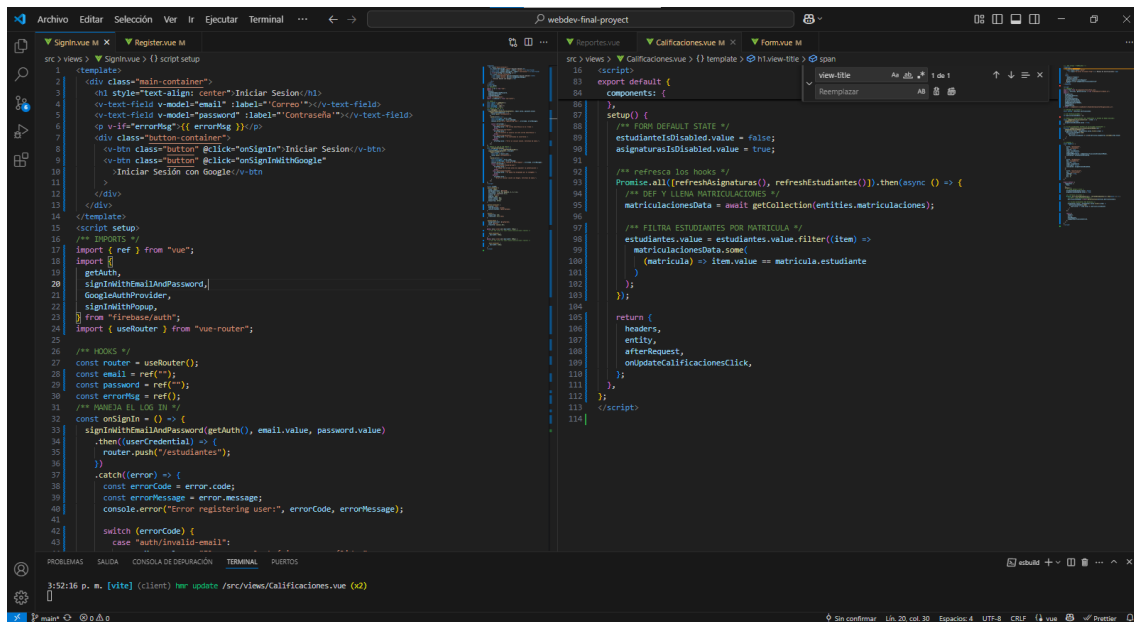
## Preparación del Entorno de Desarrollo

Para el desarrollo hemos utilizado principalmente Visual Studio Code (VS Code) es un editor de código fuente gratuito, desarrollado por Microsoft. Se ha convertido en una de las herramientas más populares entre los desarrolladores de software gracias a su versatilidad, rendimiento y extensibilidad.

VS Code es una aplicación que te permite escribir, editar y depurar código en una amplia variedad de lenguajes de programación. Va más allá de un simple editor de texto, ofreciendo una gran cantidad de características y funcionalidades que facilitan el flujo de trabajo de los desarrolladores.

Características clave de Visual Studio Code:

- Soporte para múltiples lenguajes: Funciona con una gran variedad de lenguajes de programación como JavaScript, TypeScript, Python, Java, C++, C#, Go, PHP, y muchos más. Ofrece resaltado de sintaxis, autocompletado (IntelliSense), linting y depuración específica para cada lenguaje.





## **Bibliografía**

<https://cli.vuejs.org/guide/creating-a-project.html#vue-create>

<https://firebase.google.com/>

<https://vuetifyjs.com/en/components/buttons/#usage>