

Instituto Tecnológico de Costa Rica

Escuela de Computación

Bases de Datos II GR 1

Proyecto 1 Bases de Datos II

Estudiantes:

Carlos Eduardo Leiva Medaglia / 2021032973

Kenneth Palacios Molina / 2020035407

Kendall Fabian Guzman Ramirez / 2019076561

Ivan Solis Avila / 2018209698

Jose Pablo Quesada / 2020211670

Profesor:

Gerardo Nereo Campos Araya

I Semestre 2023

Explicación de Diagrama de Base de Datos

Cursos: Esta es la tabla con más conexiones porque, al igual que en una universidad, se centra en estos. Esta posee el nombre, cantidad de créditos, horas por semana, descripción y dos FK (Foreign Keys) que apuntan a las tablas Tipos y Escuelas. El resto de las conexiones se verán en las tablas que poseen las FK.

Tipos: Esta es una tabla catálogo que tiene los tipos de Periodos Lectivos y Cursos. Esta solo contiene 5 entradas: Anual, Semestral, Trimestral, Cuatrimestral y Bimestral.

Escuelas: Esta es una tabla que contiene las escuelas en la universidad.

PeriodosLectivos: Esta tabla contiene los períodos lectivos en la universidad. Como atributos tiene: Inicio, Final y Nombre Amigable. Inicio y final hace referencia a las fechas en las cuales inició y finalizó el periodo lectivo. Además de esto, tiene dos FK: IdEstadoPeriodo e IdTipo; las cuales hacen referencia a las tablas Tipos y EstadosPeriodo respectivamente.

EstadosPeriodo: Esta es una tabla catálogo que tiene los estados de los períodos, estos se dividen en creado, en progreso y en finalizado.

PeriodosLectivos_Cursos: Esta es una tabla intermediaria para relacionar PeriodosLectivos con Cursos, solo tiene las FK de ambas tablas.

PeriodosMatricula: Esta es una tabla que guarda los periodos de matrícula, está compuesta por FechaInicio y FechaFinal, ambas siendo datetime para tener la hora también. Además, posee una FK a PeriodosLectivos.

Grupos: Esta tabla tiene el registro de grupos de un curso, como atributos posee la cantidad máxima de estudiantes y FK a Periodos, Cursos y Profesores.

Horarios: Esta tabla posee los horarios de los cursos, está compuesta por el día, la hora y las horas que dura.

Grupos_Horarios: Esta es la tabla intermediaria que relaciona a los grupos con su horario.

Profesores: Esta es la tabla que tiene el registro de todos los profesores de la universidad, esta solo está compuesta por el nombre del profesor y una FK a la escuela asociada.

Escuelas: Esta es la tabla que tiene los registros de las escuelas de la institución. Solo tiene el nombre.

Carreras: Esta tabla mantiene el registro de todas las carreras de la universidad, está compuesta por el nombre, una descripción y una FK a la escuela a la que pertenece.

Planes: Esta es la tabla que tiene todos los planes de estudio, esta está compuesta por el número de plan, la fecha de creación, la fecha de activación y la fecha de activación, además de dos FK a EstadosPlan y Carreras.

EstadosPlan: Esta es una tabla catálogo que tiene los estados de los planes, solamente tiene 3 entradas: creación, activo o cerrado.

Planes_Cursos: Esta es una tabla intermediaria entre planes y cursos. **PlanesCursos_Requisitos:** Esta es una tabla que relaciona, en un plan de curso a un curso con su requisito.

Estudiantes: Esta tabla tiene a todos los estudiantes de la universidad, con su carné, contraseña, nombre, apellidos y una FK al plan al que pertenecen. **Asistentes_Profesores:** Esta tabla relaciona a un profesor con su

estudiante asistente.

Grupos_Estudiantes: Esta tabla relaciona a un estudiante con un grupo, esto, por ejemplo, cuando se matricula.

Archivos: Esta tabla tiene un registro de todos los archivos del sistema, está compuesta por el Id del usuario que lo subió, la fecha de creación, fecha de su última modificación, nombre del archivo, descripción, versión y si es latest (la última versión) o no. Además de esto, tiene 3 FK, a TiposArchivo, Periodos y Cursos.

TiposArchivo: Esta es una tabla catálogo de los tipos que pueden estar asociados a los archivos, solo tiene su nombre.

Actividades: Esta es una tabla de las actividades que se hagan en los diferentes cursos y está compuesta por la fecha de entrega, el porcentaje y una FK al rubro asociado.

Rubros: Esta es una tabla de los rubros que se usan para los diferentes cursos, está compuesta por el nombre, porcentaje y un FK que apunta al grupo al que pertenece.

Evaluaciones_Comentarios: Esta tabla tiene registro de todos los comentarios que se hacen a una evaluación, está compuesta por el comentario (de máximo 255 caracteres) y una Fk a la evaluación.

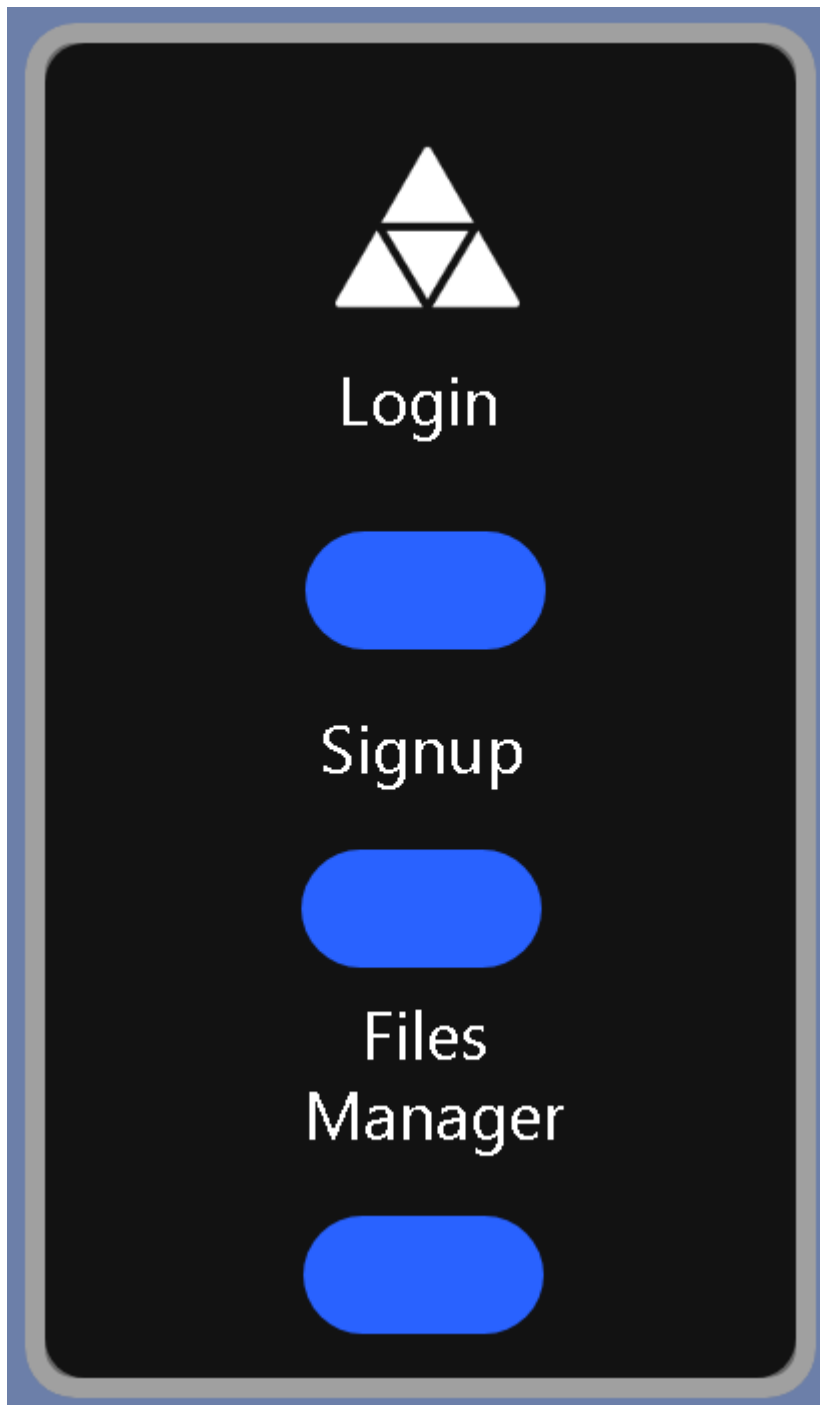
Evaluaciones_Archivos: Esta es una tabla que relaciona las evaluaciones con sus archivos.

Actividades_Estudiantes_Archivos: Esta tabla relaciona una actividad con un estudiante y un archivo.

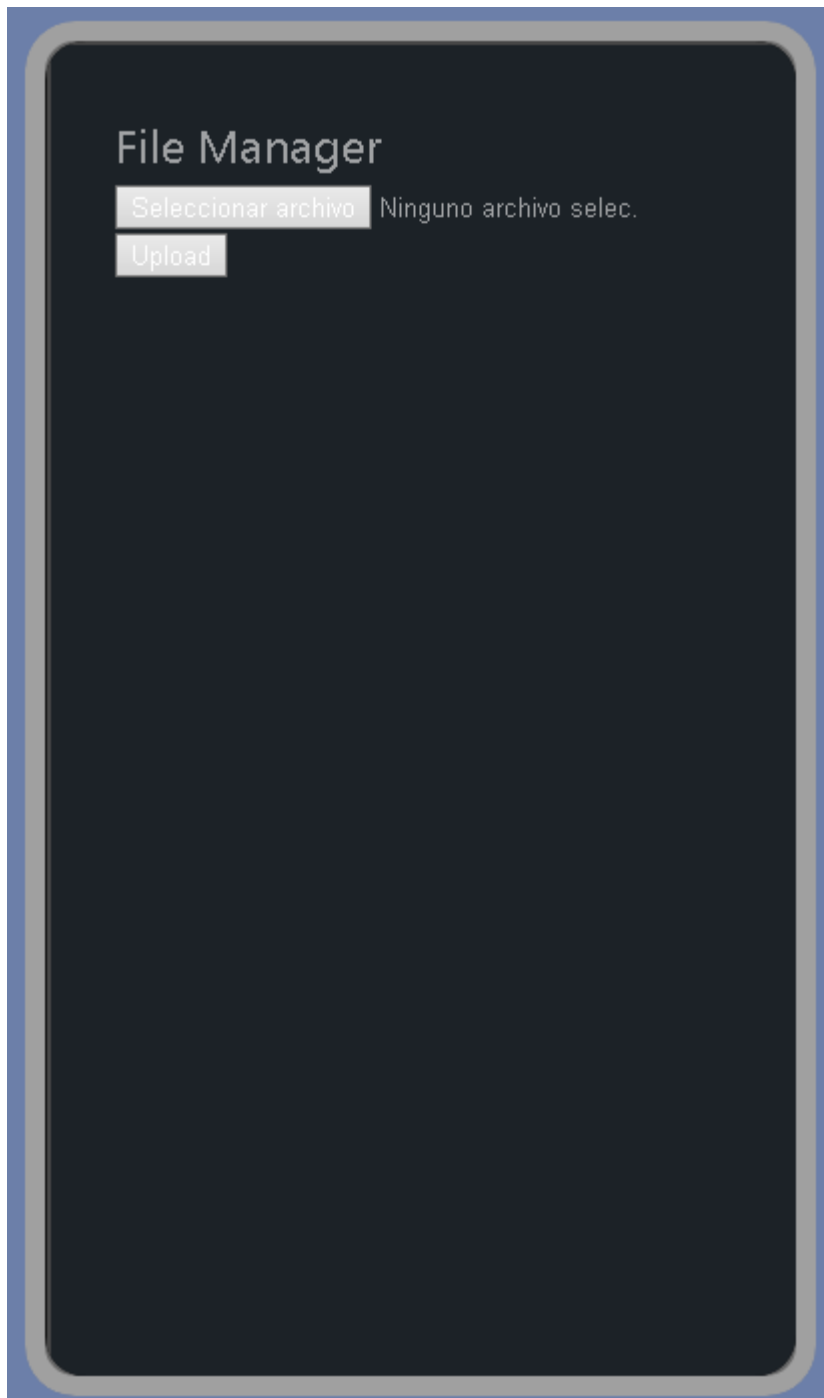
Archivos_Carreras: Esta es una tabla que relaciona un archivo con la carrera a la cual pertenece. Esto para archivos como evaluaciones de cátedra.

Explicación Funcionamiento de Thunkable

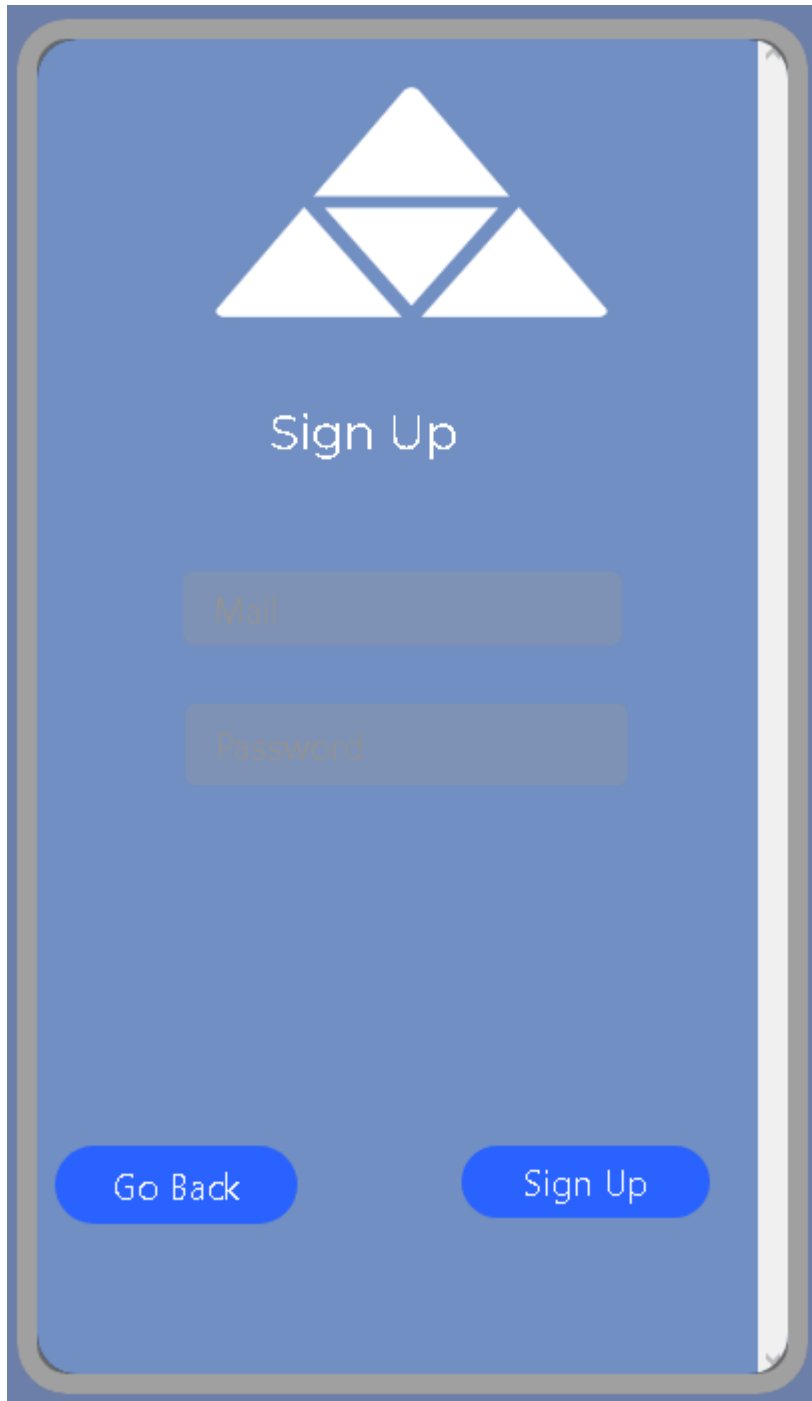
Ventana Principal: Es la ventana principal, esta se ejecuta al iniciar la aplicación. En esta se encuentran tres botones los cuales al ser presionados nos permiten realizar acciones como lo son registrarse, iniciar sesión y navegar al files manager.



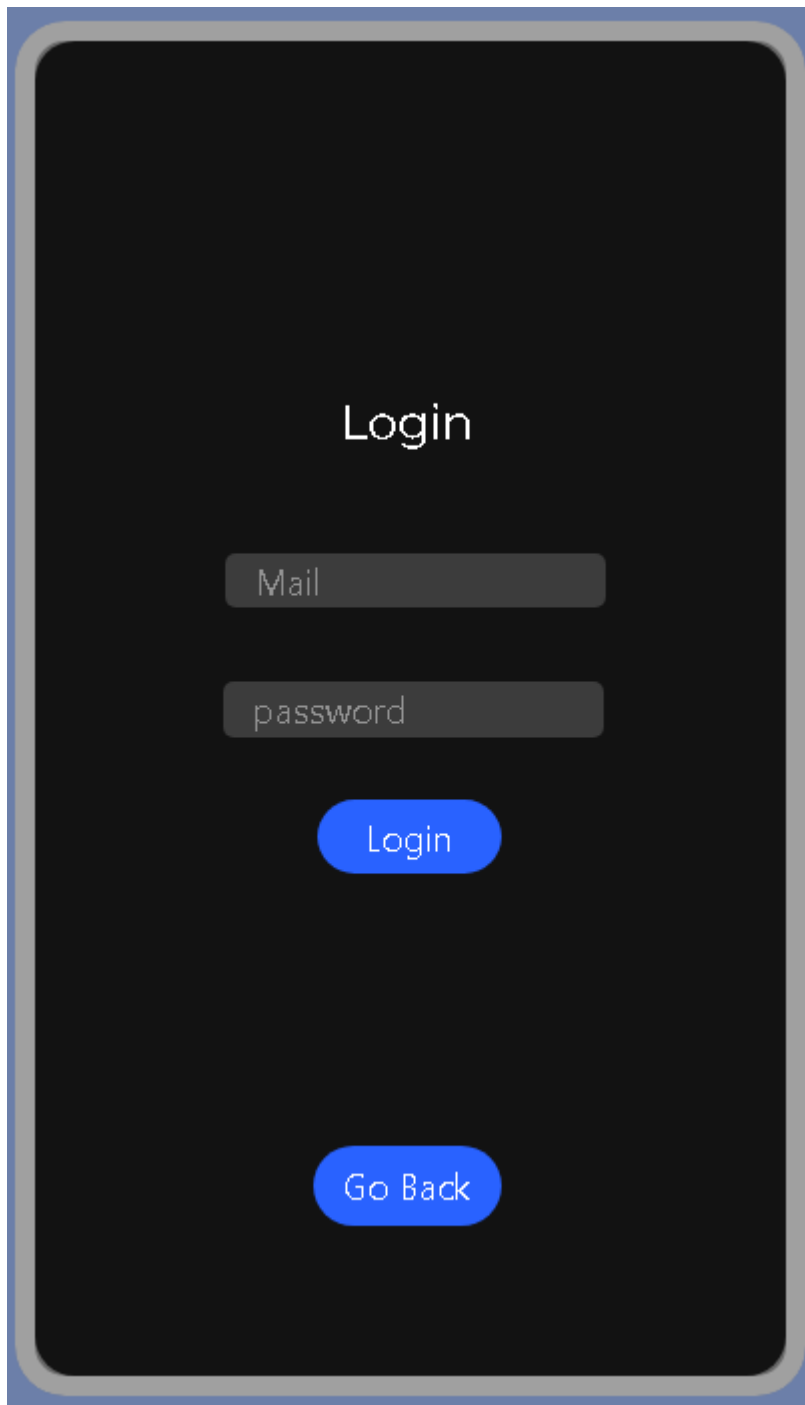
Files Manager: Esta ventana contiene la opción para poder seleccionar un archivo y subirlo.



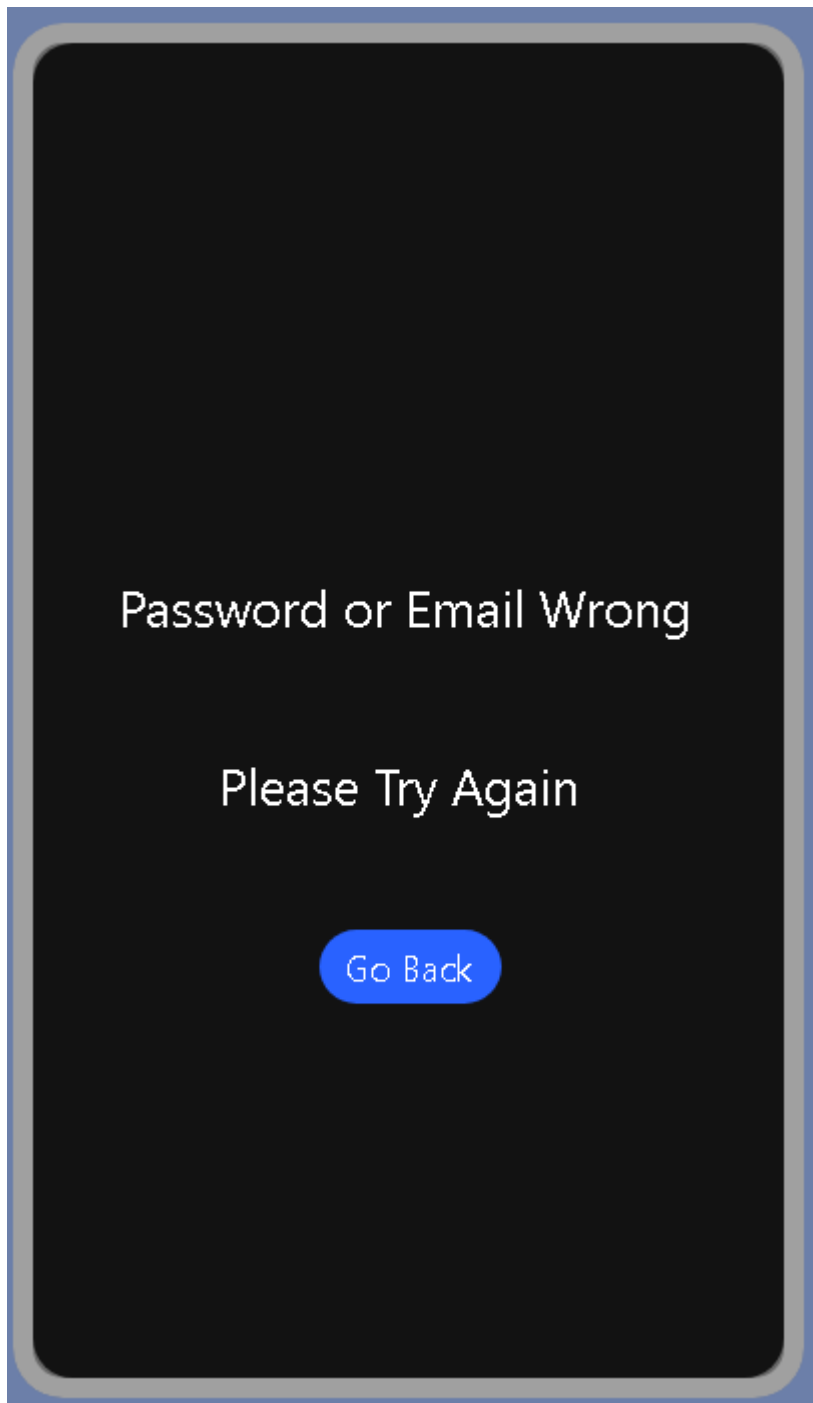
Sign Up: Esta ventana funciona para registrarse en caso de ser un nuevo usuario. Es muy sencillo simplemente se deben llenar los espacios de texto con la información que se le solicita y presionar el botón 'Sign Up', en caso de acceder a esta ventana y querer regresar a la anterior se presiona 'Go Back'.



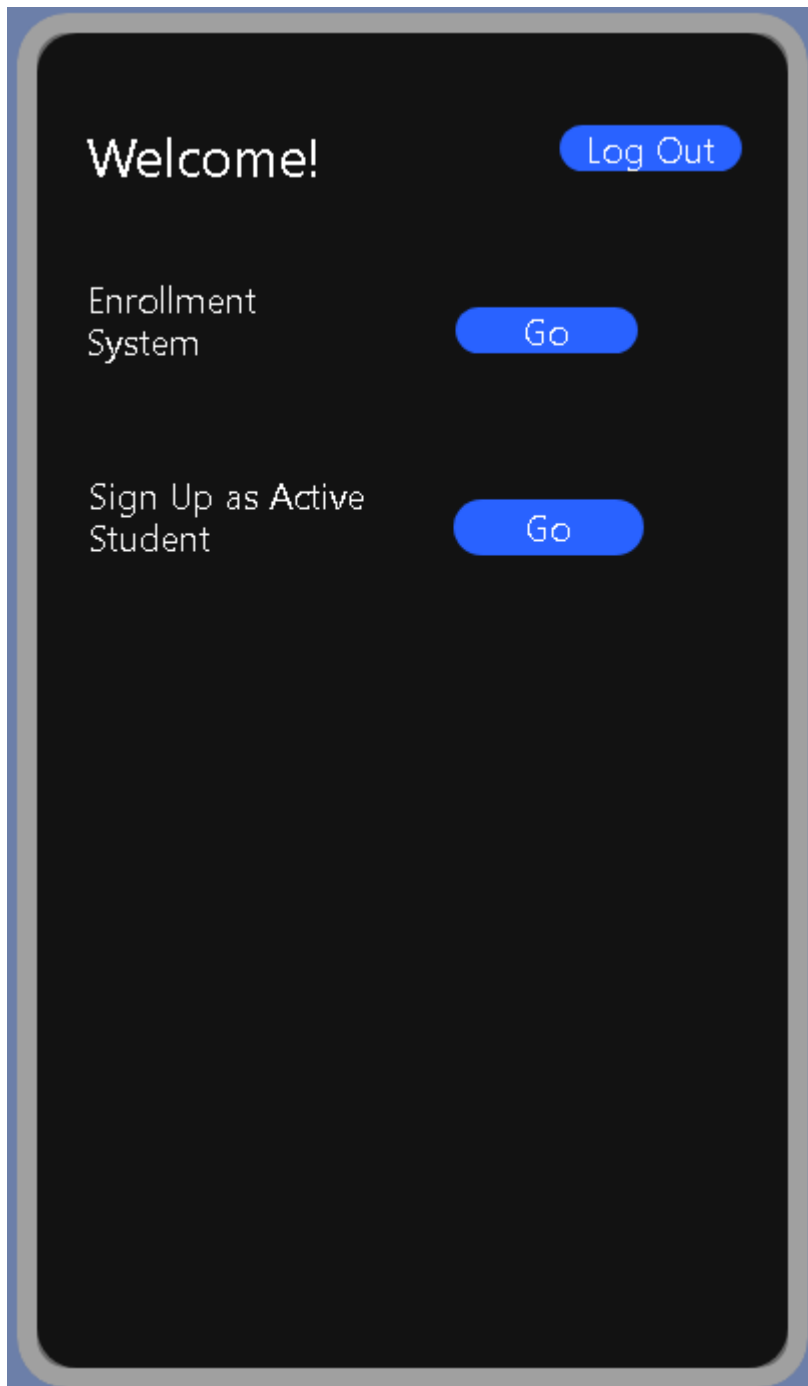
Login: Esta ventana funciona para usuarios que ya han sido registrados, en esta podrán iniciar sesión para ingresar en la aplicación. Simplemente deben llenar los espacios de texto con la información que se les solicita y con la que ya se habían registrado previamente, seguidamente se presiona 'Login' para ingresar. En caso de haber utilizado datos erróneos se le avisara y se le pedirá que vuelva a intentarlo. En caso de querer regresar a la ventana anterior se presiona 'Go Back'.



Error Login: Esta ventana unicamente se hace visible si los datos que se introdujeron a la hora de hacer login son incorrectos. Esta ventana muestra que el correo electrónico utilizado o la contraseña son incorrectas y que debe volver a intentarlo. Para volver a intentarlo simplemente se debe presionar 'Go Back' y este lo regresara a la ventana para volver a iniciar sesión.



Menu: Esta sera la ventana que se abrirá a la hora de iniciar sesión, esta nos permite acceder al sistema de matricula y registrarse como un estudiante activo. Cualquiera sea la opción seleccionada una vez al ser presionada esta nos llevara a la ventana que cumpla con dicha solicitud. En caso de querer cerrar la sesión simplemente se debe presionar 'Log Out'.



Matricula: Esta ventana se abrirá a la hora de acceder a la matricula. Esta nos permite realizar 'Submit' para enviar los datos registrados o 'Go back' para salir de la matricula y regresar al menu.



Active Student Sign Up: En esta ventana tendremos la opción de poder registrarnos como un estudiante, simplemente debemos llenar la información con los datos que se nos pidan y quedaremos registrados. También podremos ver las carreras que se encuentran disponibles, se selecciona la que se desee y quedaremos registrados para esa carrera. Haciendo 'Sign Up' quedaremos registrados, de lo contrario si se desea regresar simplemente se presiona 'Go Back'.

Ingeniería en Sistemas

Ingeniería Electrónica

Licenciatura en Administración

Licenciatura en Negocios Internacionales

Licenciatura en Derecho

Licenciatura en Criminología

Student ID

Name

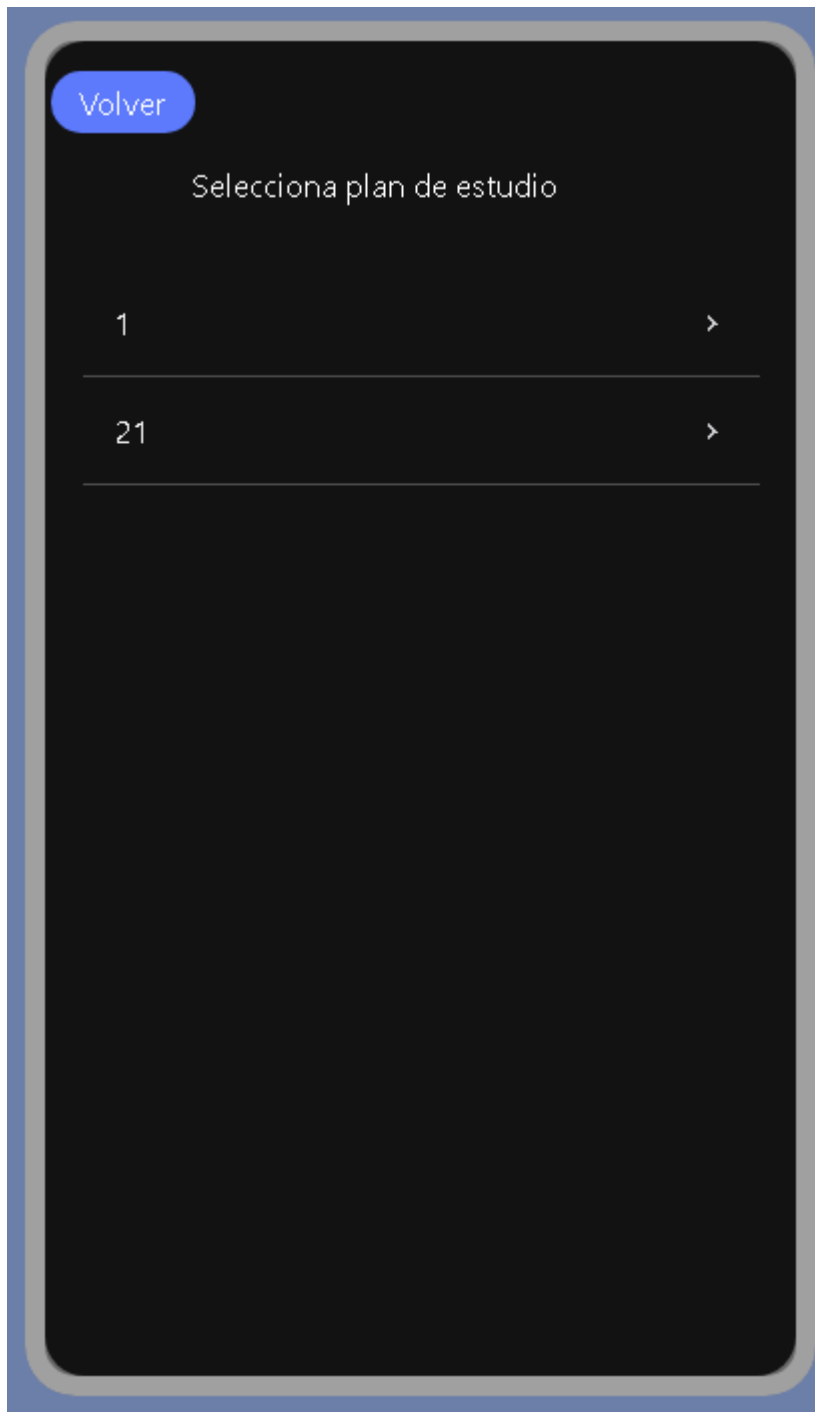
Last Name 1

Last Name 2

Go Back

Sign Up

Seleccionar Plan Estudio: Esta ventana se abrirá después de acceder a alguna de las carreras disponibles, aca se mostrar los planes de estudio que tienen disponibles estas carreras. Se puede seleccionar el plan de estudio que se desee y queda registrado, o de lo contrario si se desea devolver a la ventana anterior simplemente se presiona 'Go Back'.



Explicacion de NodeJS App

Para la app de node js decidimos usar el framework **VueJS** por su simplicidad para realizar el frontend. Ya que acá se muestran todos los documentos guardados para un usuario en la base de datos, se necesita un componente para representar el documento y otro para representar el arreglo de documentos. El primero es llamado **Item** y recibe los datos del titulo, data e isFile. El título es el nombre con el que se mostrará el archivo, data son otros datos relevantes sobre este como tamaño por ejemplo, por último isFile es un booleano que solamente se encarga de cambiar el ícono mostrado en caso de que se quiera mostrar una carpeta. Para los casos de uso actual no se debería cambiar el isFile pero se implementó por si fuese necesario a futuro. A su vez cada Item muestra unos botones para borrar o editar el item. Por otra parte existe el componente llamado **ItemList**. Como su nombre lo indica, este sirve para mostrar el listado de Items. Recibe un json con el formato:

```
items : [ { title: 'Archivo 1', data: 'Datos del archivo 1', isFile: true }, { title: 'Archivo 2', data: 'Datos del archivo 2', isFile: true } ],
```

y crea un objeto item por cada objeto de esta lista, de tal manera que si acá se traen los datos de los objetos de un usuario se pueden mostrar con mucha facilidad. Finalmente hay un área extra con un botón para agregar un nuevo archivo, que lo agregaría a la base de datos y a al itemList. Por otra parte, cabe recalcar que por la manera en la que trabaja VueJS la funcionalidad, la estructura y el estilo se encuentran en un solo archivo, por lo cual es sencillo de modularizar las partes.

Conclusiones

- Desde el inicio del proyecto se pactó una reunión la cual nos permitió estructurar todo lo que se debía hacer y dividirnos las cargas de trabajo entre los miembros del grupo. Gracias a esto cada miembro del grupo pudo aportar su parte en el proyecto y enfocarse en su respectiva tarea, de igual manera hubo mucho compañerismo a la hora de trabajar y esto nos ayudó a mantenernos en comunicación y a ayudarse unos a otros en caso de errores.
- Una ventaja que tuvimos desde el principio fue enfocarnos en modelar un diagrama de bases de datos eficiente, gracias a que esto fue una de las prioridades, una vez terminado fue mucho más sencillo continuar con el trabajo de la base de datos y todo el trabajo en esta se hizo de manera más rápida al saber de qué manera esta ya estaba modelada. Consideramos que el modelar de manera eficiente este diagrama ha sido uno de los pasos primordiales en el desarrollo.
- Docker fue una de las herramientas más útiles utilizadas en este proyecto, ya que esta nos permitía tener nuestro código en una imagen la que posteriormente podía ser configurada en Azure para que corriera desde ahí. Consideramos que Docker es una herramienta bastante importante en el desarrollo de este tipo de aplicaciones y el hecho de llevarla de la mano con un servicio en la nube como lo es Azure nos brinda mayor eficiencia al desarrollo de esta.
- Trabajar con Cassandra como base de datos NoSql fue una experiencia bastante agradable. La manera que nos permite Azure de manipular esta base de datos es muy sencilla, ya que todo lo necesario para realizar la conexión a esta es otorgado por Azure, simplemente se debe reemplazar la información con la que se nos brinda para que se haga la conexión. Una vez hecho ya se puede proceder a insertar datos en las tablas. Sin embargo, una de las dificultades que se presentaron fue el hecho de que CosmosDB necesitaba tener el ip de la persona que estaba manipulando la base, el problema está en que una vez agregado el ip si se cambia la infraestructura el ip se volvía a borrar, lo que hacía tedioso el tener que estar agregando el ip una y otra vez. Esto podía ser arreglado al agregar un bloque de código a la infraestructura, sin embargo, este paso fue realizado tarde.
- El haber utilizado Thunkable para el desarrollo de aplicaciones para dispositivos Móvil fue una experiencia interesante, ya que nunca se había trabajado con una herramienta la cual su código viniera en formato de bloques. Consideramos que Thunkable es una herramienta interesante de aprender, sin embargo, mientras se indagaba por información nos hemos dado cuenta que muchas opciones solo están disponibles en la opción de pago, lo que lo hace en algunos casos bastante limitante para trabajar, ya que había que buscar otro tipo de solución que podía resultar en agregar más bloques de código.

- El utilizamiento de VueJS como framework para la creación de componentes reutilizables en el frontend fue un gran acierto debido a que de esta manera se pueden mostrar los datos de manera simple. El hecho de que vue sea un framework minimalista y con pocas restricciones facilitó mucho su uso, a diferencia de si se hubiera elegido un framework como Angular ya que este es mucho más estructurado y complejo. Aún así la curva de aprendizaje no fue sencilla ya que posee una sintaxis un poco compleja, además que se complicó el hecho de mover información entre componentes ya que no lo encontramos muy intuitiva. Desgraciadamente no se logró conectar esta parte del proyecto con la autenticación en Firebase, por lo cual no fue posible obtener el userId para mostrar solamente los archivos de un usuario. La idea original era enviar al ItemList un JSON con los datos de todos los archivos provenientes de la BD, sin embargo al no tener userId no podemos llamar al procedimiento correspondiente.
- Es importante mencionar que la utilización de GitHub nos permitió facilitar mucho mas el trabajo en equipo, si había un cambio en la infraestructura o en el código de la api, era importante que todos pudiéramos ver estos cambios, por lo que gracias GitHub podíamos estar en constante actualización de la aplicación sin tener problemas de que tuviéramos códigos diferentes. Sin embargo, no se puede decir lo mismo de Thunkable, al trabajar con la version gratuita de este, este traía problemas a la hora de querer trabajar de manera colaborativa, ya que no lo permitía, lo que si era posible, era generar una copia del proyecto y compartirla entre todos, y de esta manera revisar que se había agregado y que no, y asi estarlo actualizando, lo que si puede considerarse algo tedioso.