UNIVERSIDAD DEL NORTE
Barranquilla - Colombia

Solución Reto Técnico #3

Sophos Solutions

Estudiante:

Jhon W. Alvarez Caldera
Código: 200093232
CC: 1007823097

Barranquilla (Atlántico, Colombia)
Octubre 29, 2022

1. Descripción

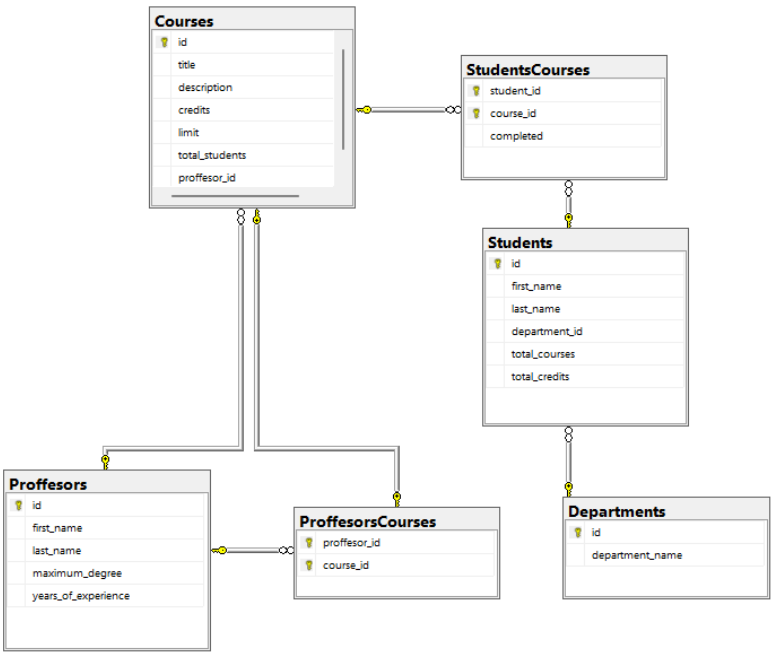
El siguiente documento muestra la forma en que se desarrolló la solución a la problemática planteada en el reto descrito en este link, con el fin de demostrar habilidades adquiridas durante una etapa de aprendizaje. En términos simples, la aplicación se basa en tener un control de los profesores, estudiantes y cursos que la Universidad Sophos tiene, así como mostrar información relevante en cada una de estas categorías. Cabe destacar que, a nivel técnico, se domina más el Frontend que el backend, así que durante esté documento, se mostrará más información según la tecnología dominante.

Tecnologías usadas:

i. Diseño de las interfaces gráficas en Figma
ii. Para el versionado de código se usó Git, montando todo el código base en Github.
iii. Tecnología para Desarrollo de Aplicación Web: Reactjs, tailwindcss y paquetes de terceros para agregar funcionalidades. Dichos paquetes se listan en la sección de Frontend.
iv. El diseño de la base de datos se realizo en SQL Server, así como la base de datos propia.
v. El framework backend usado es .NET con C#
vi. El manual para el uso del backend se realizó en Markdown, adjunto en el repositorio del código base para el backend.

2. Backend en C# con .NET

i. Diseño de la base de datos y modelo relacional: este se realizó dentro de SQL Server Management Studio, en donde se hizo la base de datos relacional también.



ii. Script de creación la base de datos. Este script se puede encontrar en el archivo adjunto, o bien, descarándolo del siguiente link LINKKKKK. Algunos de los comandos usados:

```
CREATE TABLE Students
(
    id int primary key identity,
    first_name varchar(20) not null, --required
    last_name varchar(20) not null, --required
    department_id int not null, --required
    total_courses int not null default 0, -- default value here
    total_credits int not null default 0, -- default value here
    foreign key (department_id) references Departments (id),
);
```

iii. El script para la población de la base de datos se encuentra adjunto, o bien, descargándolo del siguiente link LINKKKKKKK. Un ejemplo para insertar datos dentro de la tabla de estudiantes se muestra a continuación. Para ver el ejemplo completo se puede ver en el archivo en cuestión.

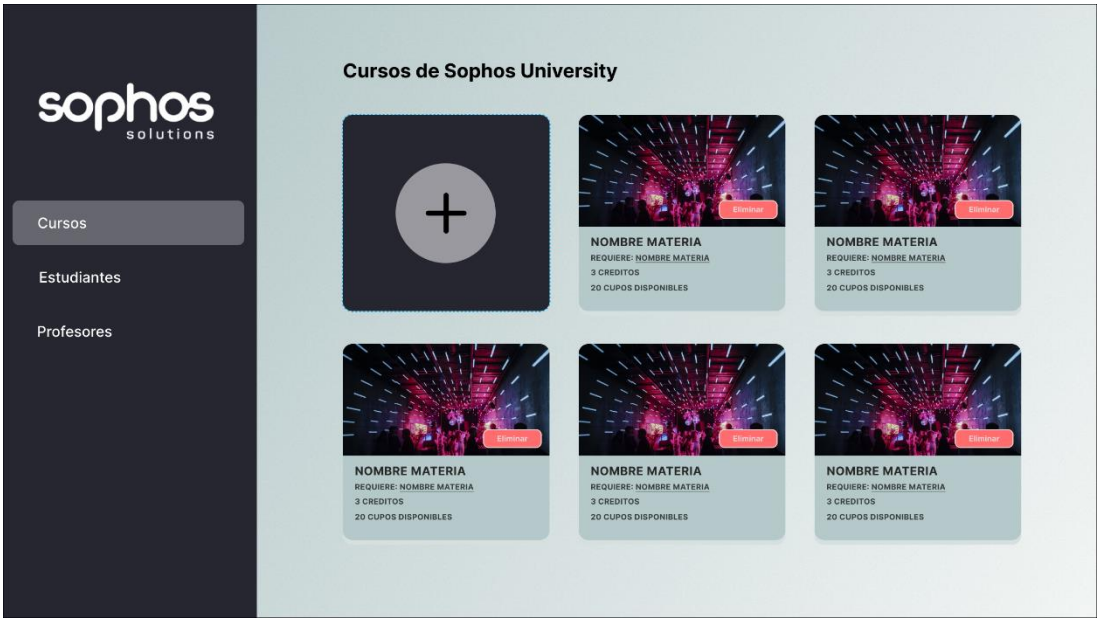
```
-- Insert into Students
INSERT INTO Students VALUES ('Eddy','Cardona',1, DEFAULT, DEFAULT);
INSERT INTO Students VALUES ('Michael','Valero', 1, DEFAULT,
DEFAULT);
INSERT INTO Students VALUES ('Helymar','Acosta', 1, DEFAULT,
DEFAULT);
INSERT INTO Students VALUES ('Joel','Borrero', 1, DEFAULT,
DEFAULT);
INSERT INTO Students VALUES ('Jose','Arteaga', 1, DEFAULT,
DEFAULT);
```

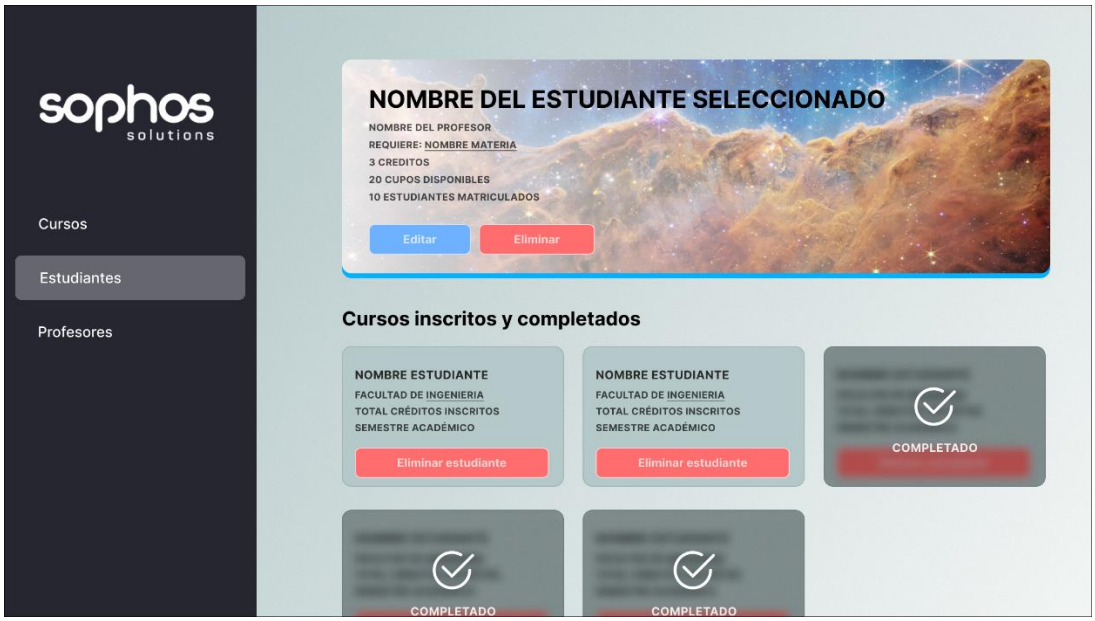
iv. Para el manual del Servicio web expuesto se usó Postman, en donde estarán los endpoints usados, así como ejemplos y anotaciones de las respuestas de estos. Para acceder se puede ingresar por medio del siguiente enlace: https://documenter.getpostman.com/view/23401702/2s84LPwXY9

3. Frontend

i. Prototipo UX/UI

Para hacer el prototipado de la aplicación se utilizó Figma, dicho diseño se puede encontrar en este link. Un vistazo rápido al diseño es el siguiente:





ii. Implementación con React.js

Para esta sección, se usará principalmente React.js junto con la librería tailwindcss para una maquetación rápida y cómoda, logrando así una vista atractiva para quien usará la aplicación web. Adicionalmente se usarán los siguientes paquetes para implementar funcionalidades extras en la aplicación:

- Axios, para hacer las peticiones al backend.
- Tailwindcss y daisyui para los estilos.
- Bootstrap icon para los íconos.
- Loaders, para mostrar durante las peticiones hhtp de forma dinámica
- Toastify, para las notificaciones. n