



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMAS DISTRIBUIDOS

LIC. ARMANDO REYES GALICIA

ARANZÚA CHÁVEZ CÉSAR OCTAVIO

BAUTISTA ORTEGA ELVIA

COLÍN BUSTAMANTE ISRAEL

GONZÁLEZ AGUILAR JULIO CÉSAR

LÓPEZ ZUGASTI CHRISTIAN

RUELAS VIURQUEZ RICARDO

TORRES OROZCO PEÑA ISRAEL ALEXANDER

GRUPO: 10

PROYECTO:

*MICROSERVICIOS (FRONTEND) PARTE I*

7 de noviembre del 2023  
Semestre: 2024 – 1



# PLATAFORMA EDUCATIVA

INTRODUCCIÓN .....	2
PÁGINA DE INICIO .....	3
MÓDULO: AUTENTICACIÓN E INICIO DE SESIÓN .....	3
MÓDULO: PERFIL DEL USUARIO .....	5
MÓDULO: CATÁLOGO DE CURSOS .....	7
MÓDULO: PÁGINA DE ADMINISTRACIÓN .....	9
REFERENCIAS .....	10

## INTRODUCCIÓN

El presente proyecto consiste en una aplicación con *microfrontends*, desarrollada con arquitectura micro *frontend* y el *framework* NextJS; los cuales consisten en módulos de:

- El catalogo de los cursos disponibles en la plataforma
- La página particular de cada curso
- La página de administración de la plataforma
- La autenticación e inicio de sesión
- La página del perfil del usuario

Inherentemente, todos los módulos están enrutados y es posible navegar entre ellos.

En concordancia con la arquitectura de un sistema distribuido, el presente proyecto implementa la aplicación en un contenedor de *Docker*, mismo del que consta su imagen como parte de los descargables de este repositorio.

Funcionalmente el sistema pretende ser una plataforma educativa, cuyos usuarios son previamente conocidos, por ello es que la página inicial es el inicio de sesión.

Una vez dentro de la plataforma, se pueden visualizar los diferentes cursos que se ofrecen en esta; y al seleccionar uno, se actualiza la vista con la información

## PÁGINA DE INICIO

Al iniciarse la aplicación, y una vez que ya se ha iniciado sesión con alguno de los métodos, aparece esta página.



# EduPlat<sup>®</sup>

The Educational Plataform of the future

## MÓDULO: AUTENTICACIÓN E INICIO DE SESIÓN

Se implementa la autenticación mediante dos métodos: un formulario que solicita correo y contraseña y mediante la api de GitHub. Si no se ha cumplido con esto, no se podrán utilizar las demás funcionalidades de la aplicación. El código respectivo se encuentra en: `'main/login/loginit/app/api/auth/[...nextauth]/route.js'`, en este se hace uso de *NextAuth.js* y los proveedores mencionados, según la documentación proporcionada por la página.

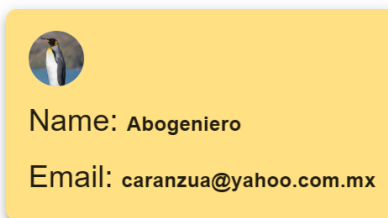
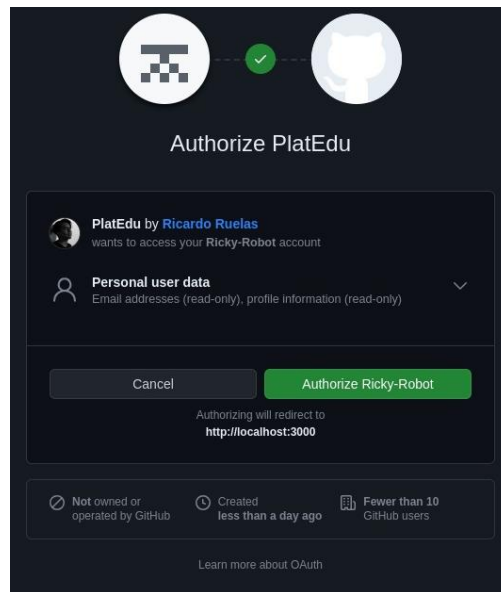
Se crea el archivo local `'/.env.local'` que contendrá las claves para la API. Este archivo debe incluirse en el `'gitignore'`

Es necesario configurar el botón en `'/components/signin-btn.jsx'` para que se pueda iniciar sesión



ACCESO

INICIAR SESIÓN CON GITHUB



La funcionalidad del inicio de sesión mediante GitHub se logra con las siguiente línea de código}

```
const authOptions = {
  providers: [
    GithubProvider({
      clientId: process.env.GITHUB_ID,
      clientSecret: process.env.GITHUB_SECRET,
    })
  ],
};


const handler = NextAuth(authOptions);
export { handler as GET, handler as POST };
```

## MÓDULO: PERFIL DEL USUARIO

Además de los datos básicos (nombre, correo y contraseña), esta vista cuenta con la funcionalidad de modificar la imagen de perfil, permitiendo cargar una nueva con la interfaz del sistema de archivos del sistema operativo, así como un botón para guardar la información.

El avatar por defecto se obtiene de la sesión generada por OAuth, así como el nombre y el correo.





UPLOAD IMAGE

Name:

Ricardo Ruelas

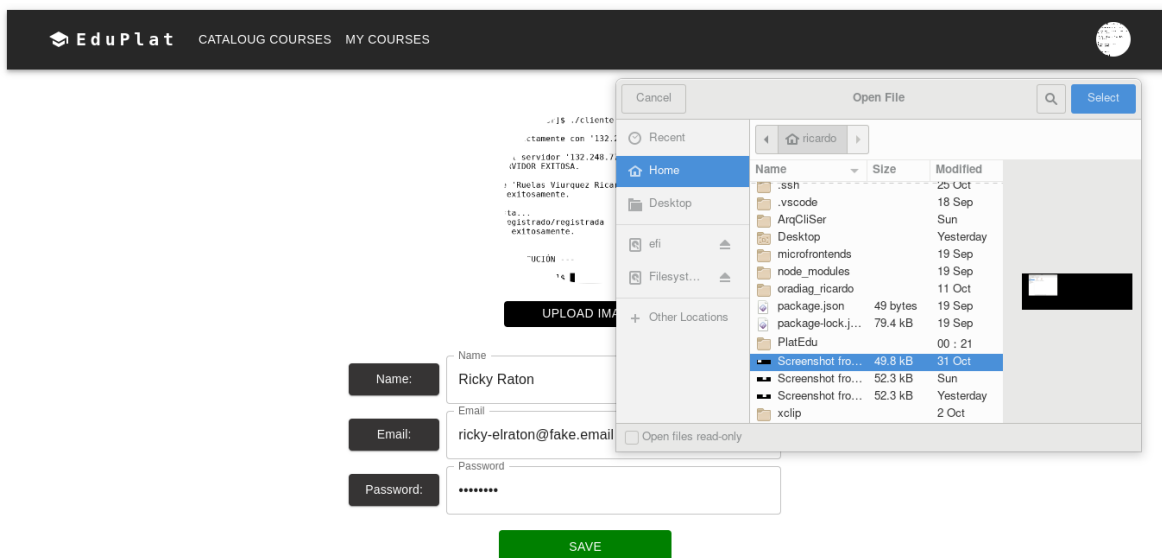
Email:

ricardo.ruelasv3@gmail.com

Password:

Password

SAVE



El campo para correo, tiene un validador pequeño.

EduPlatCATALOG COURSESMY COURSES

```
..js ./cliente 1s.  
ctamente con '132.248.77.1s.  
servidor '132.248.77.168' en p  
VIDEO EXITOSA.  
'Ruelas Virquez Ricardo'...  
exitosamente.  
Ed...  
registrado/registrado  
exitosamente.  
TUCION ...  
Tc
```

UPLOAD IMAGE

Name:

Ricky Raton

Email:

ricky-elraton@fake

Password:

\*\*\*\*\*

SAVE

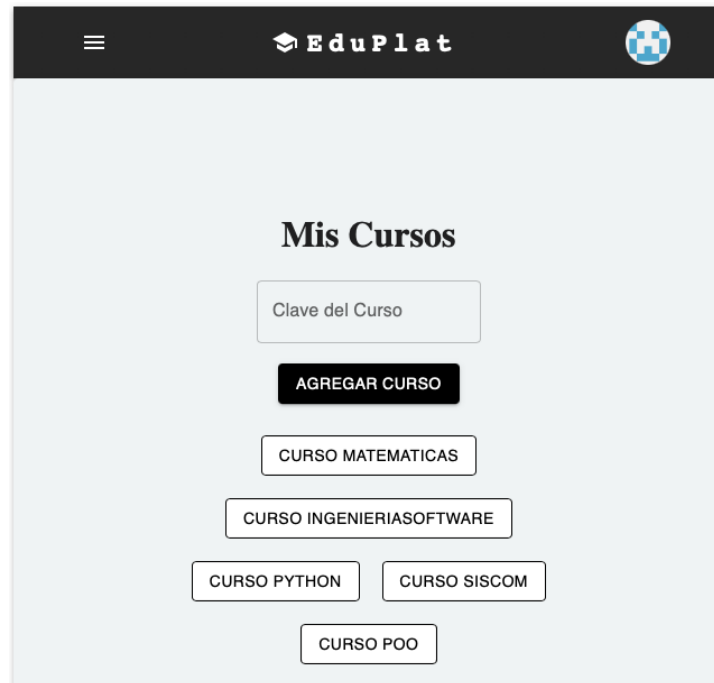
Al presionar los cambios generados en el formulario de datos se almacenarán durante el tiempo de la sesión (evitando interacciones con *backend* para enfocarnos completamente en *frontend*). Los cambios se compartirán entre todas las páginas y se mantendrán durante toda la sesión y la persistencia que permita el navegador:

EduPlatCATALOG COURSESMY COURSES

**EduPlat**<sup>®</sup>  
The Educational Plataform of the future

## MÓDULO: CATÁLOGO DE CURSOS

Esta vista se encarga de registrar las materias deseadas para cada usuario. Hay un apartado para ingresar la palabra clave de la materia. En esta plataforma las palabras claves son: Matematicas, IngenieriaSoftware, Python, SisCom y POO. En la imagen de abajo se muestra que se han agregado todas las materias.



La parte del código que permite realizar esta funcionalidad es:

```
mycourses.jsx x
pages > mycourses.jsx > MyCourses
1  import React, { useState } from "react";
2  import { useRouter } from "next/router";
3  import { Grid, Button, TextField, Paper } from '@mui/material';
4
5  const MyCourses = () => {
6    const [courses, setCourses] = useState([]);
7    const [courseKey, setCourseKey] = useState("");
8    const router = useRouter();
9
10   const handleCourseKeyChange = (event) => {
11     setCourseKey(event.target.value);
12   };
13
14   const addCourse = () => {
15     setCourses([...courses, { id: courseKey, name: `Curso ${courseKey}` }]);
16     setCourseKey("");
17   };
18
19   const viewCourseDetails = (courseId) => {
20     router.push(`/${courseId}`);
21   };
22 }
```



En donde exportan las dependencias de react, useState y las utilidades de next.js y material UI, se utiliza el estado local para almacenar la lista de cursos (courses) y la clave de curso actual (courseKey). Además, se obtiene el objeto de enrutador de Next.js.

Se define una función addCourse que se utiliza para agregar un curso a la lista de cursos.

Se define una función viewCourseDetails que se utiliza para navegar a los detalles de un curso específico.

```
19   const viewCourseDetails = (courseId) => {  
20     |   router.push(`/${courseId}`);  
21   };
```

## MÓDULO: PÁGINA DE ADMINISTRACIÓN

En esta vista se pueden visualizar los cursos disponibles y los usuarios activos en ese momento

Nombre de los cursos:	Estudiantes registrados:
Matematicas	Julio César
<input type="button" value="BORRAR"/> <input type="button" value="EDITAR"/>	<input type="button" value="STATUS"/>
Historia	Jessica Saucedo
<input type="button" value="BORRAR"/> <input type="button" value="EDITAR"/>	<input type="button" value="STATUS"/>
Python	Eric Aguilar
<input type="button" value="BORRAR"/> <input type="button" value="EDITAR"/>	<input type="button" value="STATUS"/>
EDA	Israel Lopez
<input type="button" value="BORRAR"/> <input type="button" value="EDITAR"/>	<input type="button" value="STATUS"/>
Java	Karina Madrigal
<input type="button" value="BORRAR"/> <input type="button" value="EDITAR"/>	<input type="button" value="STATUS"/>

Se tienen los datos de un arreglo que se puede ir cambiando, como solo el proyecto es *frontend*. Sin embargo, para un posible *backend*, obtener los datos sería sencillo ya que solo se tienen que poner en ese arreglo para que se despliegue y que también se puedan editar tanto los usuarios como los cursos.

```
function getRandomBoolean() {  
    return Math.random() < 0.5;  
}
```

Los usuarios aparecen con un estatus de “conectado” o “desconectado” según esta función que lo selecciona aleatoriamente. También a lo largo de la ejecución de la página se pueden ir cambiando los status para imitar el movimiento de un sitio estudiantil

## REFERENCIAS

*Deploying*, NextJS. Disponible en: [<https://nextjs.org/docs/app/building-your-application/deploying>], 29 de octubre del 2023.

*GitHub*, NextAuth.js. Disponible en: [<https://next-auth.js.org/providers/github>], 29 de octubre del 2023

*Google*, NextAuth.js. Disponible en: [<https://next-auth.js.org/providers/google>], 29 de octubre del 2023