



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas



Curso: Practicas Iniciales

Sección: F-

Docente: Herman Igor Véliz

Taller 4
Manual de Técnico

Nombres:	Carnet:
Esdras Bautista	202100301
Adriana Ojeda	202000363
Eduardo Reyes	202200091
Gary Aju	202206963
Francisco Peruch	202100639



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas



USAC APP

Se realizó el fronted en React, ya que este sirve para crear aplicaciones web o móviles donde los usuarios ven y e interactúan directamente. React es una biblioteca de JavaScript que se utiliza para construir interfaces de usuario (UI) interactivas y dinámicas.

Al desarrollar el frontend con React, se crean componentes reutilizables que representan diferentes partes de la interfaz de usuario, como botones, formularios, barras de navegación, etc. Estos componentes se pueden combinar para formar la estructura completa de la aplicación.

```
import React from 'react';
import ReactDOM from "react-dom/client";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import './index.css';
import BarraNav from './componentes/BarraNav';
import PaginaP from './componentes/PaginaP';
import Perfil from './componentes/Perfil';
import Testimonios from './componentes/Testimonios';
```

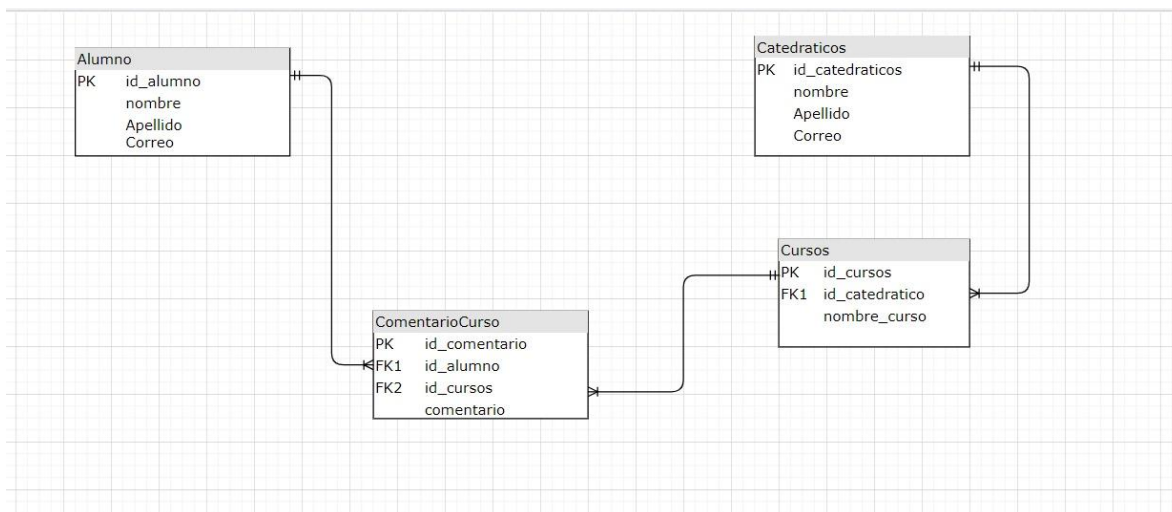
Para el backend se utilizó Node.js, ya que este entorno de ejecución de JavaScript es altamente eficiente y escalable, lo que permite manejar de manera efectiva las solicitudes y respuestas del servidor. Además, Node.js cuenta con una amplia gama de módulos y librerías que facilitan el desarrollo de aplicaciones web y de servidor robustas y rápidas. El backend se encargará de las peticiones de los clientes, también para comunicarse con la base de datos realizada con MySQL, ya que de esta manera se podrá recuperar y almacenar la información. Se utilizó NodeJs para hacer una conexión entre el fronted anteriormente mencionada, y la base de datos de MySQL. Gracias a esta parte, la aplicación funciona correctamente.



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas



A continuación, se muestra el diagrama de base de datos realizado.



Para el registro de usuarios se realizó la conexión del backend de la siguiente manera:

```
router.post('/registro', async (req, res) => {
  // Mostrar en la consola los datos recibidos desde el frontend
  console.log('Datos recibidos desde el frontend:', req.body);

  try {
    // Obtener una conexión del pool de conexiones
    const connection = await pool.getConnection();

    // Query SQL para insertar un nuevo usuario en la tabla 'usuarios'
    const query = `
    INSERT INTO usuarios (registroAcademico, nombre, apellido, contra, correo, has_commented)
    VALUES (?, ?, ?, ?, ?, false)
    `;

    // Datos del usuario recibidos desde el frontend
    const { registroAcademico, nombre, apellido, contra, correo } = req.body;
    const has_commented = false; // Por defecto, el usuario no ha comentado

    // Ejecutar la consulta SQL
    await connection.query(query, [registroAcademico, nombre, apellido, contra, correo, has_commented]);
```



Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas



Para consultar el registro académico de realizó lo siguiente:

```
router.post('/verificar_registro', async (req, res) => {
  const { registroAcademico, correo } = req.body;

  try {
    const [rows] = await pool.query('SELECT * FROM usuarios WHERE registroAcademico = ? AND correo = ?', [registroAcademico, correo]);

    if (rows.length > 0) {
      res.json({ exists: true, message: 'Usuario encontrado', usuario: rows[0] });
    } else {
      res.json({ exists: false, message: 'El usuario no está registrado' });
    }
  } catch (error) {
    console.error('Error al realizar la consulta:', error);
    res.status(500).json({ exists: false, message: 'Error interno del servidor' });
  }
});
```

Para consultar la existencia del usuario, se realizó lo siguiente:

```
router.post('/verificar', async (req, res) => {
  const { registroAcademico, contra } = req.body;

  try {
    const [rows] = await pool.query('SELECT * FROM usuarios WHERE registroAcademico = ?', [registroAcademico]);

    if (rows.length > 0) {
      const usuario = rows[0];
      if (usuario.contra === contra) {
        res.json({ exists: true, message: 'Inicio de sesión exitoso', usuario });
      } else {
        res.json({ exists: false, message: 'Contraseña incorrecta' });
      }
    } else {
      res.json({ exists: false, message: 'El usuario no está registrado' });
    }
  } catch (error) {
    console.error('Error al realizar la consulta:', error);
    res.status(500).json({ exists: false, message: 'Error interno del servidor' });
  }
});

module.exports = router;
```



Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas



Por último, si el usuario ordena restablecer su contraseña, se realizó la siguiente conexión en backend.

```
router.put('/reestablecer-contrasena', async (req, res) => {
  try {
    const { registroAcademico, correo, contra } = req.body;

    // Verificar si el usuario existe en la base de datos con el registro académico y el correo electrónico
    const connection = await pool.getConnection();
    const [rows] = await connection.query('SELECT * FROM usuarios WHERE registroAcademico = ? AND correo = ?', [registroAcademico, correo]);
    connection.release();

    if (rows.length === 0) {
      return res.status(404).json({ mensaje: 'Usuario no encontrado.' });
    }

    // Actualizar la contraseña del usuario
    const updateConnection = await pool.getConnection();
    await updateConnection.query('UPDATE usuarios SET contra = ? WHERE registroAcademico = ? AND correo = ?', [contra, registroAcademico, correo]);
    updateConnection.release();

    res.json({ mensaje: 'Contraseña reestablecida correctamente.' });
  } catch (error) {
    console.error('Error al reestablecer la contraseña:', error);
    res.status(500).json({ mensaje: 'Error al reestablecer la contraseña.' });
  }
});
```