



Universidad San Carlos de Guatemala
Facultad de ingeniería
Escuela de Ciencias y Sistemas
Prácticas iniciales F-

Manual Técnico

Taller 4

Grupo 1

Carlos Sancir 202201131
Daniel Gálvez 202203361
Jesica Colaj 201516997
Keitlyn Tunchez 202201139
Mateo Noriega 202203009

Índice

| | |
|---|----------|
| Introducción | 3 |
| Objetivos..... | 4 |
| Arquitectura | 5 |
| Front-end (Cliente) | 5 |
| Back-end (Servidor) | 6 |
| Base de Datos | 6 |
| Despliegue | 7 |
| Despliegue Local..... | 7 |
| Despliegue en la Nube | 7 |
| Contenedores Docker..... | 7 |
| Contribuciones y Mantenimiento | 8 |
| Conclusión | 9 |

Introducción

Esta aplicación web es una herramienta diseñada para los estudiantes de la Facultad de Ingeniería que buscan información y recomendaciones sobre catedráticos y cursos. Con esta aplicación, los estudiantes podrán registrarse, iniciar sesión y realizar publicaciones calificando a catedráticos o cursos específicos. Otros usuarios podrán comentar en estas publicaciones, aportando más opiniones y experiencias. De esta manera, se creará un registro completo y accesible de las valoraciones de los estudiantes.

La aplicación contará con funciones de búsqueda y filtrado, permitiendo a los usuarios encontrar fácilmente las opiniones sobre un catedrático o curso en particular. Además, los usuarios podrán ver los perfiles de otros estudiantes y la lista de cursos aprobados por cada uno.

En resumen, esta aplicación web busca brindar una solución práctica y centralizada para que los estudiantes de la Facultad de Ingeniería puedan tomar decisiones informadas al seleccionar sus cursos y catedráticos, basándose en las experiencias de otros compañeros.

Objetivos

- Proporcionar una plataforma donde los estudiantes de la Facultad de Ingeniería puedan compartir sus opiniones y experiencias sobre catedráticos y cursos.
- Facilitar la selección de catedráticos y cursos al tener acceso a calificaciones y comentarios de otros estudiantes.
- Fomentar la interacción y retroalimentación entre los estudiantes a través de publicaciones y comentarios.

Arquitectura

La aplicación se desarrolló siguiendo una arquitectura cliente-servidor, donde:

1. Cliente: Aplicación de front-end desarrollada con Angular o ReactJS.
2. Servidor: API REST desarrollada con Node.js que expone endpoints para ser consumidos por el cliente.
3. Base de Datos: Toda la información se almacena en una base de datos MySQL.
4. Tecnologías Utilizadas
5. Front-end: Angular o ReactJS
6. Back-end: Node.js
7. Base de Datos: MySQL
8. Estructura de la Aplicación

Front-end (Cliente)

El front-end de la aplicación fue desarrollado utilizando Angular o ReactJS. Esta capa es responsable de proporcionar la interfaz de usuario y manejar las interacciones con el usuario.

Algunos componentes y funcionalidades principales del front-end incluyen:

- Autenticación: Manejo de inicio de sesión, registro de usuarios y recuperación de contraseña.
- Pantalla Inicial: Componente que muestra el listado de publicaciones, permite aplicar filtros y acceder a la creación de nuevas publicaciones.
- Crear Publicación: Componente para crear nuevas publicaciones sobre un catedrático o curso específico.
- Comentarios: Componente para mostrar y agregar comentarios en una publicación.
- Perfil de Usuario: Componente que muestra la información de un usuario, sus cursos aprobados y permite editar la información personal.
- El front-end se comunica con el back-end a través de solicitudes HTTP, consumiendo los endpoints de la API REST.

Back-end (Servidor)

El back-end de la aplicación fue desarrollado utilizando Node.js y proporcionando una API REST. Esta capa es responsable de manejar la lógica de negocio y la interacción con la base de datos.

Algunos endpoints principales de la API REST incluyen:

- Autenticación: Endpoints para registro, inicio de sesión y recuperación de contraseña.
- Publicaciones: Endpoints para obtener, crear, actualizar y eliminar publicaciones.
- Comentarios: Endpoints para obtener y crear comentarios en una publicación.
- Usuarios: Endpoints para obtener información de un usuario y actualizar datos personales.
- Cursos Aprobados: Endpoints para obtener y gestionar los cursos aprobados de un usuario.
- El back-end utiliza una base de datos MySQL para almacenar y recuperar la información necesaria para la aplicación.

Base de Datos

La base de datos MySQL almacena toda la información de la aplicación, incluyendo:

- Usuarios: Tabla para almacenar los datos de los usuarios registrados.
- Publicaciones: Tabla para almacenar las publicaciones realizadas por los usuarios.
- Comentarios: Tabla para almacenar los comentarios realizados en las publicaciones.
- Cursos Aprobados: Tabla para almacenar los cursos aprobados por cada usuario.

Despliegue

La aplicación se puede desplegar de diferentes maneras, dependiendo de los recursos disponibles.

Algunas opciones son:

Despliegue Local

- El front-end se puede ejecutar localmente siguiendo las instrucciones de Angular o ReactJS.
- El back-end se puede ejecutar localmente con Node.js.
- La base de datos MySQL se puede ejecutar en una instancia local.

Despliegue en la Nube

- El front-end se puede desplegar en un servicio de hosting web como Netlify, Vercel o GitHub Pages.
- El back-end se puede desplegar en un servicio de hosting como Heroku, AWS o Google Cloud.
- La base de datos MySQL se puede alojar en un servicio de base de datos en la nube como AWS RDS, Google Cloud SQL o similar.

Contenedores Docker

Se pueden crear contenedores Docker para el front-end, back-end y base de datos. Los contenedores se pueden orquestar y desplegar utilizando herramientas como Docker Compose o Kubernetes.

Independientemente del método de despliegue elegido, es importante configurar las variables de entorno necesarias, como las credenciales de la base de datos y las URLs de los diferentes servicios.

Contribuciones y Mantenimiento

El proyecto se desarrolló de forma colaborativa utilizando un repositorio en línea como GitHub. Todos los miembros del equipo tuvieron acceso al repositorio y realizaron commits con sus respectivas contribuciones.

Para mantener y actualizar la aplicación, se recomienda seguir las mejores prácticas de desarrollo de software, como:

- Utilizar un flujo de trabajo de control de versiones adecuado (por ejemplo, Git Flow o GitHub Flow).
- Crear ramas separadas para nuevas características y correcciones de errores.
- Realizar revisiones de código antes de fusionar cambios en la rama principal.
- Mantener una documentación actualizada y pruebas unitarias.
- Implementar procesos de integración y despliegue continuos (CI/CD) para facilitar las actualizaciones.

Conclusión

Al desarrollar esta aplicación, se puso en práctica habilidades esenciales como programación con frameworks modernos, arquitectura cliente-servidor, manejo de bases de datos relacionales, control de versiones y trabajo en equipo. El resultado final es una aplicación funcional que cumple con los requerimientos establecidos y brinda una solución práctica para que los estudiantes puedan evaluar y recomendar catedráticos y cursos de una manera organizada y centralizada.