

Examen Final - Desarrollo de Aplicaciones Híbridas

Introducción

Este examen final es una combinación de los dos parciales anteriores, donde se espera que se utilice React (con estados como useContext), Node.js, MongoDB y Express para crear una aplicación compleja.

Parte 1: Desarrollo de Sitio Web y API con React y Node.js

1.1 Creación de Sitio Web en React

Desarrolla un sitio web que se ejecute en el puerto 3333 con las siguientes características:

- Interfaz de Usuario con React: Diseña una interfaz de usuario dinámica y reactiva utilizando React.
- Menú Temático: Crea un menú con al menos 5 secciones relacionadas con tu temática elegida.
- Visualización y Redirección de Proyectos: Al seleccionar una sección, muestra proyectos específicos de esa categoría.
- Detalles de Proyectos: Cada proyecto debe incluir nombre, descripción, tecnologías usadas, link al repositorio e imagen.

1.2 Creación de API con Node.js y Express

El API, desarrollado con Node.js y Express, deberá:

- Operaciones CRUD: Permitir obtener, modificar, eliminar y crear proyectos.
- Filtrado de Proyectos: Implementar filtrado por al menos dos campos.
- Entidad "Cliente": Agregar la entidad "Cliente" con nombre, foto y descripción.
- Modelado de Base de Datos en MongoDB: Asegurar relaciones entre clientes y proyectos.

1.3 Especificaciones Técnicas

- Base de Datos MongoDB: Utilizar la base de datos "AH2023" con una colección adecuada.
- Estructura del Documento en MongoDB: Incluir campos necesarios como ID, nombre, descripción, link, imagen, tecnologías y sección.
- Tecnologías y Buenas Prácticas: Asegurar el uso correcto de MongoDB, ECMAScript modules, React, Node.js y Express.

Parte 2: Integración con Proyecto Estilo Trello (o la temática que tengan)

2.1 Funcionalidades Requeridas

- Gestión de Usuarios y Proyectos: Incluir capacidades para crear cuentas, modificar perfiles, gestionar proyectos y tareas.

- Interacción entre Usuarios: Permitir invitar y eliminar usuarios de proyectos.
- Visualización de Proyectos: Mostrar proyectos asociados al usuario en la página principal.

2.2 Extras y Tecnologías Opcionales

- Restricciones de Edición: Limitar la edición de proyectos según su estado (en curso, cancelado, terminado).
- Uso Opcional de Redux, GraphQL o Sockets: Estas tecnologías pueden añadirse para mejorar la funcionalidad y eficiencia de la aplicación.

Criterios de Evaluación y Entrega

- Código en GitHub o Google Drive: Comparte tu código, sin necesidad de incluir la base de datos.
- Evaluación: Se valorará el modelado de la base de datos, la API, la interfaz de usuario con React, la lógica del backend con Node.js, y el uso correcto de Express y MongoDB.