Taller de trabajo individual en patrones arquitecturales

Requisitos de finalización

Crear un Sistema CRUD para Gestionar Propiedades

Objetivo:

Los estudiantes deben desarrollar un sistema CRUD (Crear, Leer, Actualizar, Eliminar) simple para gestionar propiedades inmobiliarias. El objetivo es construir una aplicación web básica que permita a los usuarios realizar las siguientes operaciones en listados de propiedades:

- Crear nuevos listados de propiedades.
- Leer o ver una lista de todas las propiedades y los detalles individuales de cada propiedad.
- Actualizar los detalles de propiedades existentes.
- Eliminar listados de propiedades.

Requisitos:

Frontend (HTML + JavaScript):

- Crear una interfaz de usuario simple con formularios para capturar información de propiedades (por ejemplo: dirección, precio, tamaño, descripción).
- Mostrar una lista de todas las propiedades con opciones para ver, actualizar y eliminar cada una.
- Implementar validación del lado del cliente (por ejemplo: campos obligatorios, tipos de datos válidos).
- Usar AJAX o Fetch API para comunicarse con los servicios REST del backend.

Backend (Spring Boot REST API):

Desarrollar endpoints RESTful para cada operación CRUD:

- POST para crear una nueva propiedad.
- GET para obtener todas las propiedades o una sola propiedad por ID.
- PUT para actualizar una propiedad existente.
- DELETE para eliminar una propiedad por ID.

Manejar errores como:

- Entradas inválidas.
- Solicitudes de propiedades no existentes.

Asegurarse de que cada propiedad tenga los siguientes atributos:

- Property ID (generado automáticamente)
- Address
- Price
- Size

- Description

Base de datos (MySQL):

- Crear una tabla llamada `properties` con columnas para ID, dirección, precio, tamaño y descripción.
- Usar JPA/Hibernate para mapear los objetos de propiedades a la base de datos.
- Implementar persistencia de datos para todas las operaciones CRUD.

4. Los servicios backend y la base de datos deben estar desplegados en servidores separados en AWS.

Mejoras opcionales (para crédito adicional):

- Agregar paginación a la lista de propiedades.
- Implementar búsqueda para filtrar propiedades por ubicación, precio o tamaño.
- Proveer retroalimentación al usuario en operaciones exitosas o fallidas (por ejemplo, mensajes de éxito, notificaciones de error).

Entregables

Un repositorio independiente que contenga:

- El código de las evidencias del taller de AWS-CLI.
- El código de las clases del tutorial de JPA.

Un sistema CRUD funcional que permita gestionar propiedades desplegado en AWS.

Código en un Repositorio GitHub:

- Todo el código del proyecto debe estar organizado y subido a un repositorio GitHub.
- El repositorio debe seguir buenas prácticas con estructuras claras de carpetas para frontend, backend y configuración de base de datos.

Archivo README:

Incluir un archivo README.md con:

- **Resumen del Proyecto:** Una breve explicación del sistema y su funcionalidad principal (gestión de propiedades).
- **Arquitectura del Sistema:** Descripción del frontend, backend y base de datos, y cómo interactúan.
- **Diseño de Clases:** Vista general de las clases principales con diagramas si es posible (por ejemplo, Property, PropertyService, PropertyController).
- **Instrucciones de Despliegue:** Pasos para generar y desplegar imágenes, desde la configuración hasta la ejecución de los servicios en AWS.
- **Screenshots:** Imágenes mostrando el sistema en funcionamiento, incluyendo operaciones CRUD.

Video de Despliegue:

- Un video corto demostrando el sistema en ejecución, incluyendo ejemplos de crear, leer, actualizar y eliminar.
- El video debe mostrar cómo desplegar el sistema.