

Sistema de Gestión de Tareas con Estructuras Personalizadas

Descripción General:

Proyecto desarrollado con Java y Spring Boot que gestiona tareas utilizando estructuras de datos manuales (Lista, Pila, Cola y Árbol). El sistema expone APIs REST y publica eventos a través de RabbitMQ.

Tecnologías Utilizadas:

- Java 17
- Spring Boot
- Maven
- MySQL
- RabbitMQ (mensajería de eventos)
- Postman (pruebas de API)
- JPA / Hibernate (persistencia de historial)

Estructuras Implementadas (Semana 2):

- ListaPersonalizada<T>: Lista enlazada para gestionar tareas.
- PilaAcciones<T>: Registro de acciones recientes.
- ColaProgramada<T>: Tareas en orden FIFO.
- ArbolTareas<T>: Representación jerárquica de subtareas.

Servicios (Semana 2):

- Servicios para cada estructura de datos.
- Métodos: agregar, eliminar, listar, contar, etc.

APIs REST (Semana 3):

- /api/tareas: CRUD básico sobre lista de tareas.
- /api/acciones: Apilar y desapilar acciones.
- /api/cola: Encolar, desencolar y consultar tareas programadas.
- /api/arbol: Inicializar árbol, agregar tareas hijas, mostrar jerarquía.

Todos los controladores publican eventos a RabbitMQ.

RabbitMQ (Semana 3):

- Se configuró tareas-exchange, acciones-exchange, cola-exchange, arbol-exchange.
- Las colas reciben eventos como: creación, eliminación y procesamiento de tareas.

Persistencia (Semana 4):

- Historial: Registro de todas las acciones ejecutadas.
- Guardado automático en base de datos usando Spring Data JPA.
- Entidades Usuario y Tarea están modeladas, aunque no persistidas.

Pruebas:

- Todas las rutas fueron probadas desde Postman.
- Se verificó publicación de eventos en RabbitMQ.
- Datos del historial se almacenan correctamente en MySQL.

Documentación Adicional:

- Swagger puede ser integrado para documentación automática.
- Código limpio y modular.

Video (Opcional):

- Puede grabarse una demo mostrando uso de Postman, RabbitMQ y base de datos.

Proyecto completo y funcional según el cronograma establecido.