



Proyecto: Sistema de Visualización de Estructuras de Datos y Algoritmos

Descripción General

Este proyecto tiene como objetivo desarrollar una aplicación web completa que permita visualizar gráficamente el comportamiento de diferentes estructuras de datos y algoritmos. Los estudiantes deben aplicar conceptos de programación frontend y backend, trabajando con tecnologías modernas como Angular 18, Ionic 7, Spring Boot, Java 17, Graphviz y MongoDB.

La aplicación permitirá cargar datos mediante archivos CSV, procesarlos en segundo plano, y generar representaciones visuales de cómo se comportan las estructuras de datos al insertar o eliminar elementos. Estas visualizaciones se mostrarán en un carrusel interactivo y los registros procesados podrán consultarse en cualquier momento.

Objetivos del Proyecto

1. Implementar una aplicación web completa con arquitectura frontend-backend
2. Desarrollar dos microservicios independientes
3. Integrar Angular 18 e Ionic 7 para crear una interfaz de usuario responsive y moderna
4. Utilizar Spring Boot y Java 17 para crear una API robusta y escalable
5. Implementar diferentes estructuras de datos y algoritmos de grafos
6. Generar visualizaciones dinámicas usando Graphviz
7. Persistir la información procesada en MongoDB
8. Implementar un sistema de procesamiento asíncrono en segundo plano
9. Crear interfaces de usuario intuitivas para visualizar el comportamiento de las estructuras de datos

Arquitectura del Sistema

Frontend (Angular 18 + Ionic 7)

- Aplicación responsive que funcione en navegadores de escritorio
- Interfaz para carga de archivos CSV
- Visualización de imágenes generadas en un carrusel interactivo
- Panel de consulta de registros procesados con filtros y paginación
- Comunicación con el backend a través de API REST

Backend (Spring Boot + Java 17)

- **Microservicio 1:** Gestión de estructuras de datos lineales y árboles
 - Procesamiento de archivos CSV
 - Implementación de estructuras de datos (pilas, colas, listas, arreglos, árboles)
 - Generación de visualizaciones con Graphviz
 - Persistencia en MongoDB

- **Microservicio 2:** Gestión de grafos y algoritmos de ruta
 - Procesamiento de archivos CSV con información de grafos
 - Implementación de algoritmos de búsqueda de rutas óptimas
 - Generación de visualizaciones con Graphviz
 - Persistencia en MongoDB

Funcionalidades Requeridas

Parte 1: Estructuras de Datos Lineales

- Procesamiento de archivos CSV con operaciones para pilas, colas, listas y arreglos
- Generación de imágenes por cada operación de inserción o eliminación
- Visualización del estado de la estructura después de cada operación

Parte 2: Árboles

- Procesamiento de archivos CSV con operaciones para árboles binarios y AVL
- Generación de imágenes por cada operación de inserción o eliminación
- Visualización del balance y rotaciones en árboles AVL

Parte 3: Grafos

- Procesamiento de archivos CSV con información de grafos
- Implementación de algoritmos de búsqueda de rutas óptimas (Dijkstra)
- Visualización gráfica de las rutas encontradas

Funcionalidades Transversales

- Carga y validación de archivos CSV
- Procesamiento asíncrono de datos
- Persistencia en MongoDB
- Consulta y filtrado de registros procesados
- Visualización en carrusel de las imágenes generadas

Especificaciones Técnicas

Formatos de Archivo

CSV para Estructuras Lineales

estructura,operacion,valor
pila,insertar,10
pila,insertar,20
pila,eliminar,
cola,insertar,30
lista,insertar,40
arreglo,insertar,50
pila,eliminar,
cola,insertar,100
cola,insertar,90
cola,insertar,87
cola,eliminar,

CSV para Árboles

estructura,operacion,valor
binario,insertar,10
binario,insertar,5
binario,insertar,15
avl,insertar,20
avl,insertar,15
avl,insertar,25
binario,eliminar,5

CSV para Grafos

origen,destino,peso
A,B,5
B,C,3
A,C,10
C,D,2
B,D,4

Requisitos de Implementación

Frontend (Angular 18 + Ionic)

1. **Componentes Principales:**
 - Componente de carga de archivos con validación
 - Carrusel de imágenes generadas
 - Lista paginada de registros procesados con filtros
 - Visualizador de detalles de cada estructura
2. **Servicios:**
 - Servicio de comunicación con los microservicios
 - Servicio de gestión de imágenes
 - Servicio de notificaciones

Backend (Spring Boot + Java 17)

1. **Microservicio 1 - Estructuras Lineales y Árboles:**
 - Controladores REST para la carga de archivos
 - Servicios para procesamiento de estructuras
 - Integración con Graphviz para generar visualizaciones
 - Repositorio MongoDB para persistencia
2. **Microservicio 2 - Grafos:**
 - Controladores REST para la carga de archivos
 - Servicios para procesamiento de grafos y rutas
 - Integración con Graphviz para generar visualizaciones
 - Repositorio MongoDB para persistencia
3. **Servicios Comunes:**
 - Validación de archivos CSV
 - Procesamiento asíncrono mediante Spring Batch o similar
 - Gestión de errores centralizada

Validaciones Requeridas

1. **Archivos CSV:**
 - Validación de extensión (.csv únicamente)
 - Validación de formato (separadores, encabezados)
 - Validación de contenido (valores permitidos)
 - Tamaño máximo permitido (5MB)
2. **Operaciones:**
 - Validación de tipos de estructura válidos
 - Validación de operaciones permitidas por estructura
 - Validación de valores (numéricos para estructuras de datos)
3. **Consultas:**
 - Validación de parámetros de filtrado
 - Validación de rangos de paginación

Entregables

1. Código fuente completo del proyecto en un repositorio Git (en total son 3)
2. Archivo README con instrucciones de instalación y ejecución (Se debe indicar que puertos están utilizando)
3. Ejemplos de archivos CSV para probar cada funcionalidad
4. Presentación del proyecto (10-15 minutos)

Plazo de Entrega

- **14 de Mayo de 2025**

Recomendaciones

1. Comenzar con un diseño claro de la arquitectura antes de implementar
2. Desarrollar primero una versión mínima funcional de cada componente
3. Integrar gradualmente las diferentes partes del sistema
4. Realizar pruebas regulares de cada módulo
5. Utilizar control de versiones (Git) de manera efectiva
6. Dividir el trabajo en tareas específicas asignadas a cada miembro del equipo

Este proyecto permite a los estudiantes aplicar conocimientos de programación, trabajando con tecnologías actuales en un contexto práctico y visual que facilitará la comprensión de conceptos fundamentales de estructuras de datos y algoritmos.