Instructivo de ejecución

El informe describe el progreso alcanzado en el desarrollo del aplicativo web para la selección de códigos de parroquia, cantón y provincia.

El desarrollo fue realizado utilizando \*\*Java 8\*\*, \*\*Maven\*\* y \*\*PrimeFaces\*\*, con la configuración de un servidor \*\*WildFly\*\* para la ejecución de la aplicación.

**### Instalación y Configuración**

Se instaló y configuró Java 8 junto con Maven en el IDE \*\*Eclipse\*\*, creando un entorno de desarrollo funcional para gestionar las dependencias del proyecto.

Además, se instaló WildFly, configurado correctamente para soportar la arquitectura requerida.

La creación del proyecto se realizó bajo el estándar \*\*Maven\*\*, nombrándolo `localidades`, con la estructura de directorios apropiada para los archivos fuente (`src/main/java`) y recursos (`src/main/resources`). Asimismo, se definieron las dependencias necesarias en el archivo `pom.xml`.

**### Desarrollo de Clases Java y Lógica de Negocio**

En el directorio `src/main/java`, se desarrollaron las siguientes clases que representan las entidades de negocio del sistema:

- \*\*Cantón.java\*\*

- \*\*Parroquia.java\*\*

- \*\*Provincia.java\*\*

Adicionalmente, se implementaron:

- \*\*Controladores\*\*: Gestionan las peticiones del frontend al backend.

- \*\*Servicios\*\*: Procesan y manejan los datos de las entidades.

- \*\*Managed Bean\*\*: Facilita la conexión entre la vista y la lógica del servidor.

En el directorio `src/main/resources`, se copió el archivo `provincias.json`.

**### Desarrollo del Frontend**

En el frontend, se utilizó \*\*PrimeFaces\*\* para diseñar una interfaz de prueba `.xhtml`, ubicada en el directorio `webapp`. La funcionalidad principal incluye un \*\*dropdown dinámico\*\* para la selección de información.

**### Problemas Encontrados**

Durante el desarrollo, se presentaron los siguientes inconvenientes:

1. \*\*Errores en la lectura del archivo JSON\*\*: Debido a una desalineación entre la estructura de datos del archivo `provincias.json` y las clases Java destinadas a representarlos, surgieron problemas de serialización de objetos JSON a objetos Java. Por esta razón, se crearon otros archivos JSON de prueba.

2. \*\*Limitaciones del equipo informático\*\*: La falta de memoria del equipo utilizado ralentizó significativamente la instalación, configuración de herramientas y compilación, lo que afectó el tiempo disponible para compilación, realización de pruebas y ajustes.

**### Estado del Proyecto**

Por razones de tiempo, no fue posible depurar completamente los errores ni realizar las pruebas funcionales necesarias. Aunque el código tiene una estructura básica funcional, los problemas pendientes impidieron la verificación completa de su funcionamiento.

Final del formulario