

Solución prueba SmartJPS FullStack

DESARROLLADOR: RAFAEL ANGEL RAMÍREZ SERNA

Contenido

Introducción.....	3
Herramientas utilizadas	3
Backend.....	3
Frontend.....	3
Requerimientos satisfechos	3
Diseño de la Solución.....	4
Backend.....	4
Frontend.....	4
Diagrama UML	5

Introducción

Con el propósito de cumplir con el requerimiento presentado por la empresa SmartJPS, se realizó una aplicación web que cumpliera con cuatro casos de uso básicos los cuales eran filtrar información, crear registro, modificar registro y eliminar registro. Además, dentro del planteamiento del problema y anexo a cada uno de los casos de uso, se presentaban unas restricciones de funcionalidad y comportamiento de la información.

Herramientas utilizadas

En la construcción de la solución del problema se utilizaron las siguientes herramientas:

Backend

- *Eclipse* y *Spring Tools Suite* como IDE de desarrollo
- *Java 8*, *Java Persistence API* y *Spring Boot* para gestionar el modelamiento y gestión de los datos.
- *Tomcat Server*
- *PostgreSQL* y *SQL* como administrador y lenguaje de base de datos relacional.

Frontend

- *Angular 8* framework para el desarrollo de la interfaz de usuario.
- *Bootstrap*, para diseñar estilos de la interfaz gráfica.
- Librería *Datatables* para gestionar los filtros sin necesidad de hacer consultas al backend.

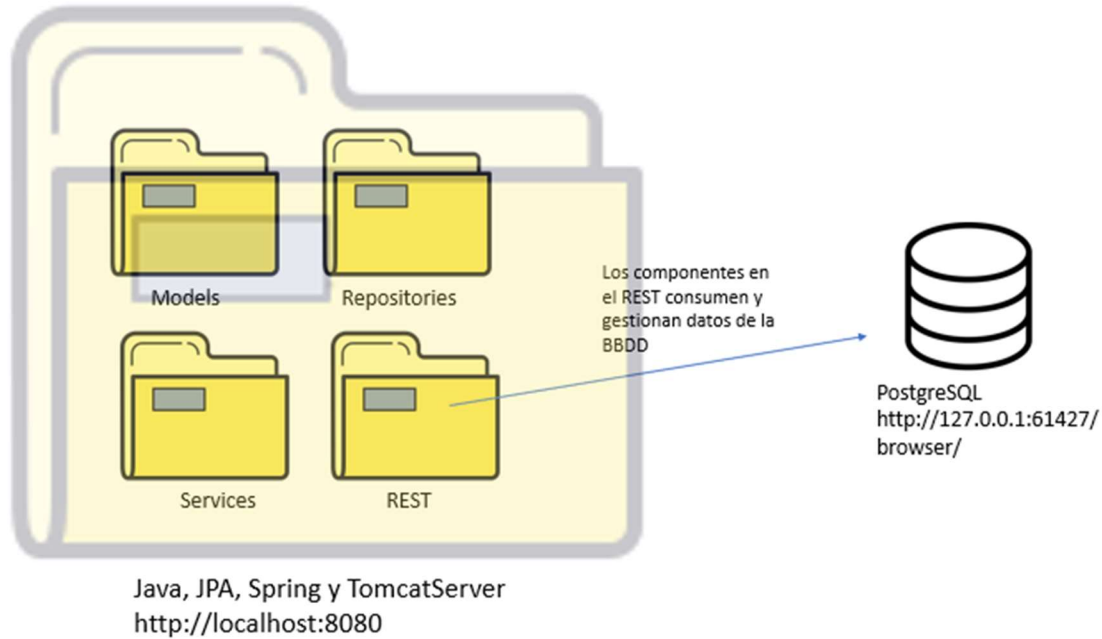
Requerimientos satisfechos

1. Flujo de evento básico al 100%
2. Implementado filtro para los registros de las tablas, 100% funcional ágil sin consultas que recarguen el servidor.
3. Creación de usuario por medio de un formulario reactivo a campos que no pueden ser indefinidos, fecha del sistema y mensaje informativo al momento de la creación del elemento con retorno a la vista de usuarios del sistema.
4. Modificación de registro 100% funcional, consume el mismo formulario de la creación, trayendo los datos del usuario seleccionado, con registro con campos estado y fecha no modificables, común formulario reactivo a campos que no pueden ser indefinidos.

5. Eliminación sencilla, mensaje de confirmación del elemento que se va a eliminar y actualización de la tabla.

Diseño de la Solución

Backend



Frontend

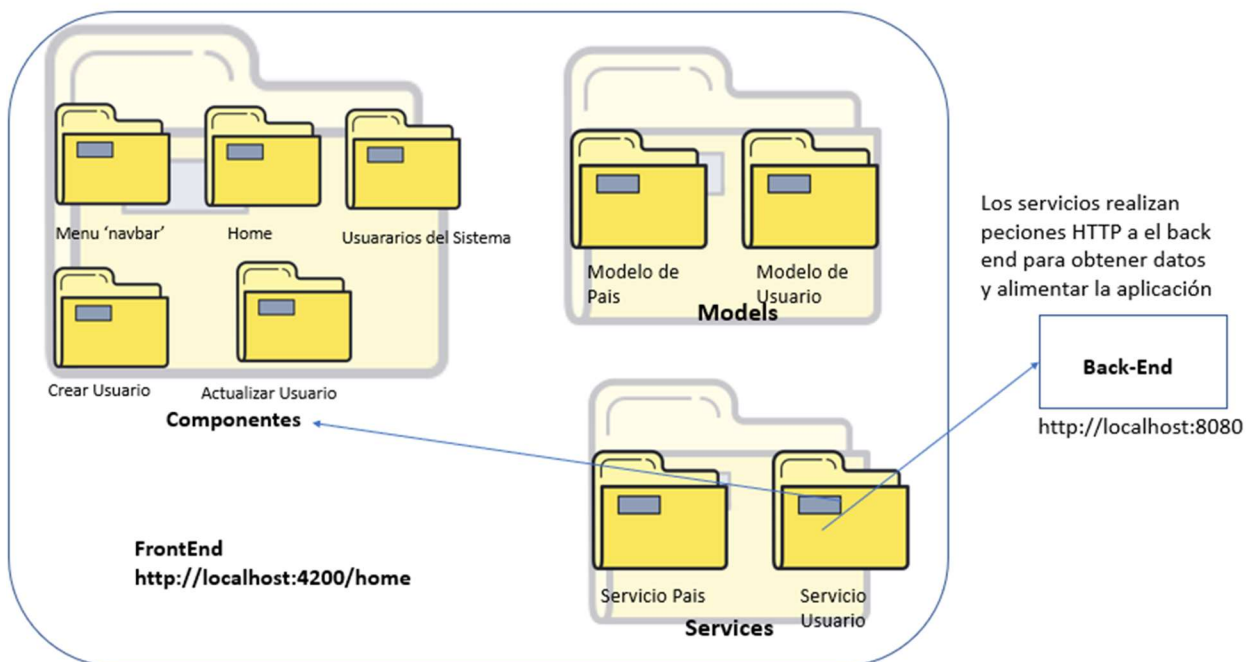
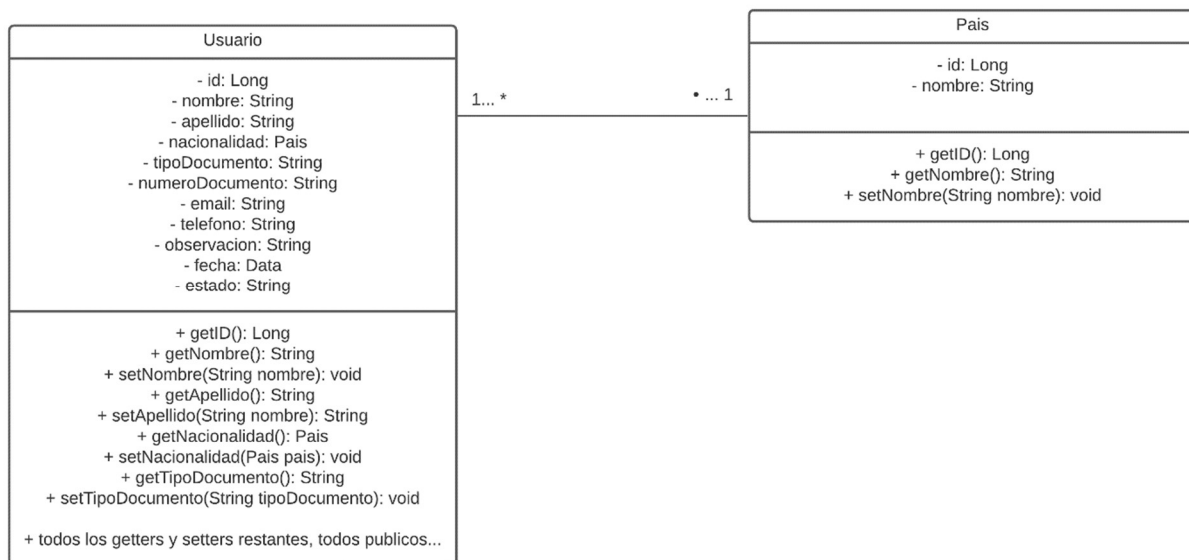


Diagrama UML

- Tanto la clase usuario como la clase país, son tablas creadas por Spring (ver archivo application.properties) y de esa misma forma son preestablecidas por medio de decoradores con la relación que se ve a continuación.



- Con todo y lo anterior, cada una de estas clases tiene una interfaz repositorio una clase que la implemente: servicio y un REST que mapea la base de datos relacional con las entidades modelo.

