**Prueba backend Sergio Perez**

Para realizar esta prueba decidí utilizar angular 11 para la parte del front end y spring boot para la parte del backend.

* Backend:
  + Recursos expuestos:

En el backend se están exponiendo 3 rutas que permiten gestionar los usuarios de la aplicación:

* + - POST mapping **/users**
      * A través de este path se realiza la creación de los usuarios en la aplicación.
    - GET mapping **/users**
      * A través de este path se obtiene la lista de los usuarios creados en la aplicación.
    - PUT mapping **/users/process**
      * A través de este path se realiza el procesamiento de los usuarios de la aplicación cambiando su estado procesado a true
  + Estructura general del proyecto
    - Modelo:
      * Se estableció una clase que representa un usuario dentro de la aplicación. Esta clase se ubica en el paquete model dentro del proyecto.
    - Repositorio
      * Interfaz que supone el puente de la aplicación con la base de datos H2. Esta clase extiende de JPA repository y permite realizar acciones sobre la base de datos como son búsquedas de elementos, modificaciones o persistencias.
    - Servicios
      * Capa intermedia que conecta el controlador que expone los recursos creados, con el repositorio que permite acceso a la data necesaria de la aplicación.
    - Controlador
      * Clase que se encarga de exponer los recursos de la aplicación. En esta clase se establecen los enpoints a los que se tendrá acceso, la información que puede ser recibida y los tipos y formatos de respuesta de consumo.
* Front end

Para el frontend decidí utilizar angular 11. Este proyecto angular consume los servicios expuestos por el backend y se encarga de “pintar” en pantalla la información.

* + Estructura del proyecto
    - Components:
      * User-create:
        + En este componente se encapsulan los elementos referentes a la creación de un usuario en la aplicación. Este componente cuenta con un elemento netamente presentacional (user-create.component.html) y un componente puente de flujo de presentación (user-create.component.ts) que se encarga de definir la forma de consumo de los servicios del backend para la creación de un usuario nuevo en la aplicación.
      * User-list:
        + En este componente se encapsulan los elementos referentes a la consulta de usuarios dentro de la aplicación al igual que el procesamiento de los usuarios seleccionados.   
          Este componente cuenta con un elemento netamente presentacional (user-list.component.html) y un componente puente de flujo de presentación (user-list.component.ts) que se encarga de definir la forma de consumo de los servicios del backend tanto para listar los usuarios ya creados , como para procesar los usuarios seleccionados.
    - Model
      * User.model.ts
        + Define las características de la entidad utilizada en la aplicación (Usuario).
    - Services
      * User.service.ts
        + Elemento que se encarga de consumir los servicios de lógica de negocio que provee el backend y suministrar de información a los componentes presentacionales antes mencionados. El consumo del backend se realiza a través de llamados HTTP.