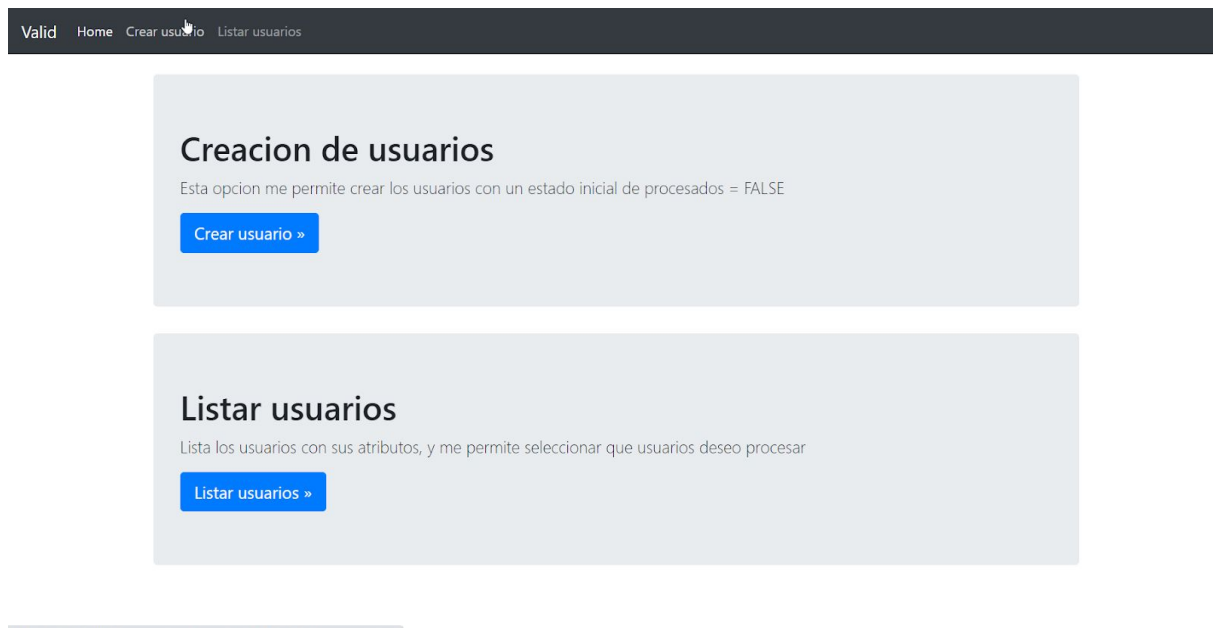


Instalación de la aplicación:

- Descargar el proyecto y abrir la carpeta users en el editor de preferencia para la parte java.
- Ejecutar el comando `./mvnw clean install` para que se ejecuten los test y se genere el JAR.
- validar que el puerto 8080 no esté ocupado ya que el JAR se despliega en el puerto 8080.
- Ejecutar el JAR creado en la Carpeta target con el comando `java -jar <nombre del jar>`
- url de la base de datos <http://localhost:8080/h2-console> para ingresar no necesita ninguna contraseña.
- Abrir la carpeta frontEnd y abrir el archivo index.html en cualquier navegador.

Print screen:





Formulario de creacion de usuario

Guardar usuario

© 2017-2020

#	Nombre	Apellido	Process	
8981d165-6e00-4a2d-a9b8-e3901dc16411	Ramiro	Contreras	true	
3792fc8f-28e5-4a31-bd31-f1d2056cf7a0	Juan	Cuadrado	false	<input type="checkbox"/>
623bc6b0-285a-4a5d-97c1-c83c009ed84d	James	Rodriguez	false	<input type="checkbox"/>

Procesar usuarios seleccionados

© Ramiro Contreras2020

Ejemplos de peticiones:

Creación de usuario:

url: POST localhost:8080/users

body:

```
{
  "name": "Andrea",
  "last_name": "alvarez"
}
```

Listar usuarios:

url: GET localhost:8080/users

Procesar usuarios:

url: POST localhost:8080/users-process

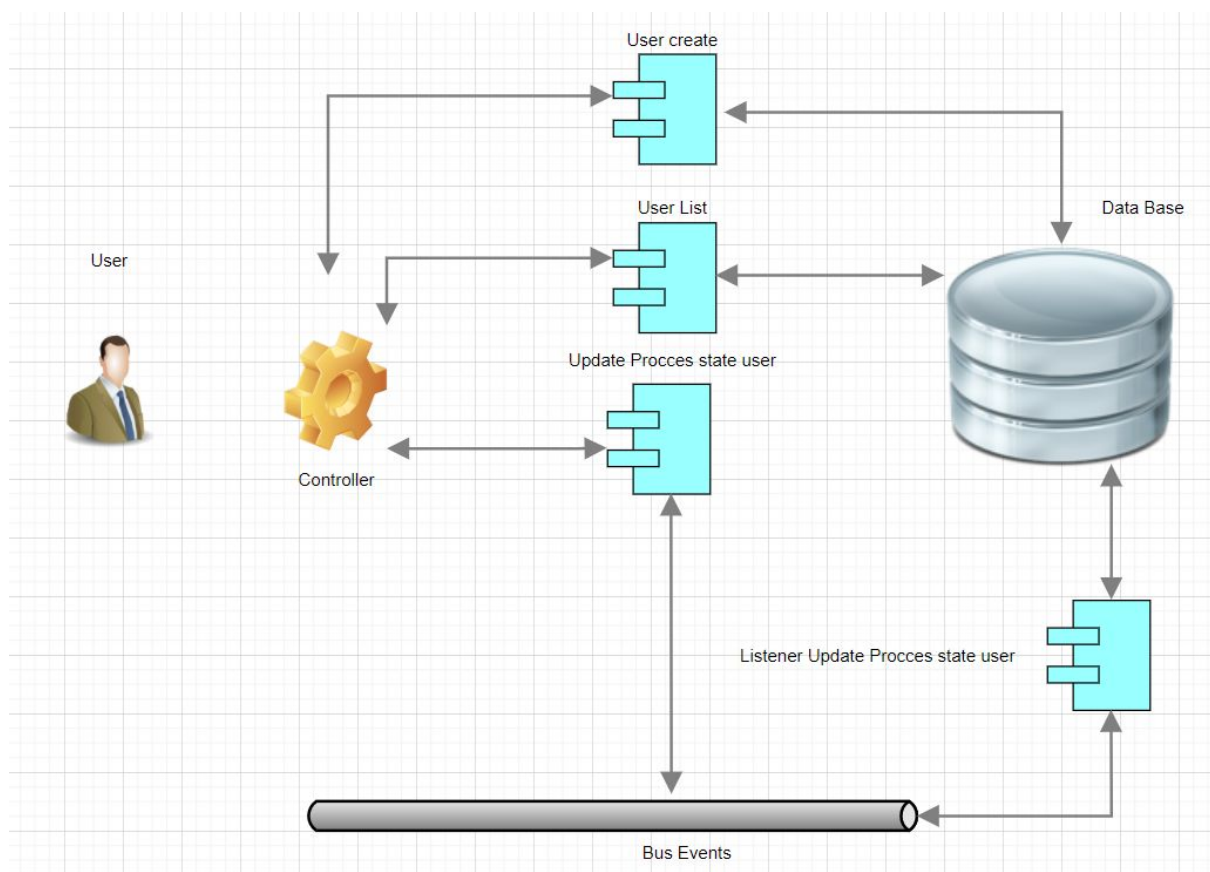
body:

```
{
  "user_ids": [
    "bcbe0da1-e705-4588-87b2-66b8fc7a74c1",
    "bcbe0da1-e705-4588-87b2-66b8fc7a74c1"
  ]
}
```

Patrones:

- Se hizo uso de la inyección de dependencias provisto por Spring, esta inyección se hizo mediante constructor.
- Se realizó un approach de ordenamiento de folders en base a la arquitectura hexagonal.
- Principios SOLID.
- Patrón Repository provisto por jpa, para la conexión a la BD.
- Patrón de UUID para los id de los usuarios.
- FrontEnd creado con Html, JavaScript, JQuery y para los estilos Bootstrap en la version 4.5.

Arquitectura componentes prueba Valid:



- **User create**: Componente que me permite crear un usuario, por defecto lo crea con un estado Proces: False.
- **User List**: Componente que lista los usuarios creados con su respectivo estado en proces.

- **Update Procces State user:** Componente que actualiza el estado de los usuarios mediante su id.
- **Events Bus:** para el procesamiento y el cambio de estado de los usuarios, se implementó un componente de eventos con spring Boot, el controller me permite crear un evento con los ids que se tienen que procesar y posteriormente un listener ejecutara el evento llamando al componente **Update Procces State user** para que se persistan en la BD.