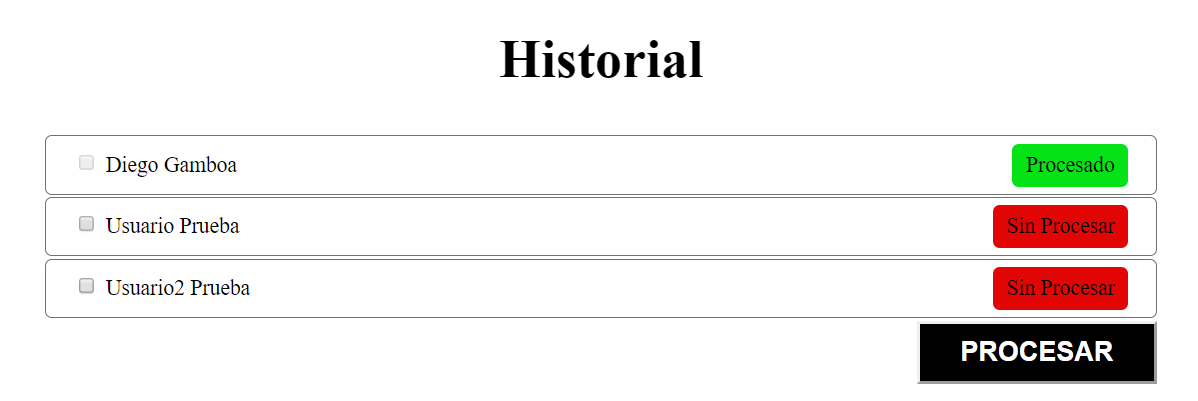
**Prueba Práctica - Backend Developer**

**Sitio Web**

* Pagina 1: Registro de un usuario  
  <http://localhost:8080/register>

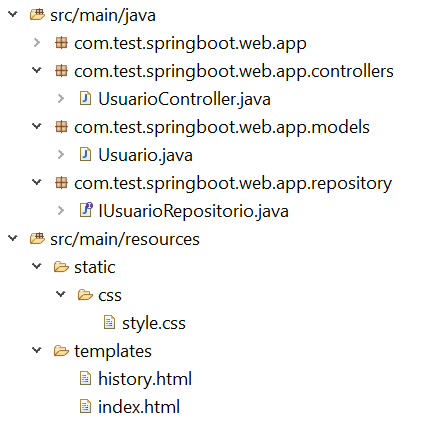


* Pagina 2: Historial de usuarios y cambio de estado en el campo “Procesado”  
  <http://localhost:8080/history>



**Patrón arquitectural**

* MVC (Modelo, Vista y Controlador): Este patrón arquitectural es altamente interactivo con el usuario, y debido a que el fin de la prueba es permitirle al usuario interactuar con la página se decidió implementarlo en la solución.
  + **Models:** Contiene la representación de los datos del usuario, adicionalmente define la estructura de la base de datos, en este caso solo contiene una sola tabla (usuarios)
  + **Controllers:** Interactúa con los modelos y las vistas para obtener la información y ejecutar las funcionalidades del sistema.
  + **Views:** No esta constituido por un paquete especifico, sino que se maneja a partir de archivos HTML y CSS los cuales se encuentran en las carpetas *static* y *templates* correspondientemente.



**Estilos arquitecturales**

* Cliente – Servidor: El cliente realiza peticiones al servidor a través de una página web (http) usando servicios REST para las operaciones de registrar usuarios, desplegarlos en una lista y actualizar sus valores.
* Capas:
  + Lógica de presentación: Corresponde a los archivos HTML y CSS que permiten la visualizar la información en el navegador. De igual forma el controlador también funciona como manejador de eventos.
  + Lógica de negocio: se puede asociar con el paquete de *models* definiendo la representación de datos a partir de la clase Usuario.
  + Lógica de Integración: Corresponde a la integración que se realiza con la base de datos H2
  + Sidecar: Correspondiente al manejo de los datos del usuario en las diferentes capas para ser procesados, editados o visualizados.