Documentación

# Prueba

## Aplicación Servidor

Descripción de Servicio:

* Web Service de tipo REST, utilizado para el acceso a la información para todas las sucursales de la tienda.
* Seguridad Implementada Basic Auth/Validación de los usuarios por medio de la clase UserDetailService, perteneciente al paquete security del servidor.
* Puerto configurado 9000
* Framework utilizado para su desarrollo Spring Boot.
* Metodología utilizada PagingAndSortingRepository de SRPING-DATA que el api de spring nos provee con funciones automáticas para la creación de un web api.
* Servicio de Base de datos utilizado MYSQL.
* IDE utilizado: el que nos brinda Spring Tools o eclipse mars.
* Version de java 1.8.0\_05
* Control de dependencias mediante Maven.
* ORM utilizado Hibernate, en conjunto de las anotaciones de JPA
* Para mayor información revisar el archivo application.properties.

### Rutas Utilizadas

Host:Puerto = <http://localhost:9000>

|  |  |  |
| --- | --- | --- |
| Método | Ruta | Resultado |
| GET | <http://localhost:9000/productos/> | Devuelve la lista de todos los productos |
| GET | <http://localhost:9000/paises/> | Devuelve los países |
| GET | <http://localhost:9000/marcas/> | Devuelve Marcas |
| GET | <http://localhost:9000/tipoInstrumento/> | Devuelve los tipos de Instrumento |
| POST | <http://localhost:9000/productos/> | Crea nuevos productos |
| GET | <http://localhost:9000/search?modelo=valor> | Filtra productos por modelo. |

Ejecución de Servicio

1er forma: Mediante la utilización del archive war dentro de servidor Pivotal tc server Developer Edition v3.1

Recomendada

2da forma: Mediante la utilización del comando mvn spring-boot:run o desde el ide al importarlo como un maven project

### Justificación

Se utilizó esta tecnología por el tipo de requerimiento solicitado y por la limitante del tiempo ya que nos provee de herramientas para la realización de aplicaciones de una forma rápida y sencilla. Permitiendo también personalizar todo lo elaborado.

## Aplicación Cliente

Descripción de aplicación:

* Tipo de cliente JEE, mediante la utilización de Spring Boot y su api MVC.
* Manejo de dependencias con Maven
* Etiquetas utilizadas para respetar el patrón mvc, Thymeleaf en lugar de etiquetas JSTL.
* Interfaz gráfica TwittwerBootstrap.
* Utilización de Spring RestTemplate para el acceso a la información.
* ViewResolver automatico mediante la utilización de Spring mvc.
* Acceso a la aplicación mediante la dirección : localhost:8085

### Justificación

Forma rápida para creación de aplicaciones clientes y forma de respetar el patrón MVC mediante Thymeleaf del lado html.

Usuarios para acceso a la información:

|  |  |
| --- | --- |
| Email | Contraseña |
| [inforetana@gmail.com](mailto:inforetana@gmail.com) | 123 |
| [jp@gmail.com](mailto:jp@gmail.com) | 321 |

Ejecución de servicio: Mediante la utilización del comando mvn spring-boot:run o desde el ide al importarlo como un maven project