

## Documento guía de Inicialización de servicios

El proyecto ha sido desarrollado con el lenguaje de programación Java, utilizando el Framework Spring Boot y como gestor de base de datos MySQL.

Puede ver el diseño del proyecto Diagrama Entidad-Relación y Diagrama arquitectónico adjuntos. A demás se ha incluido un archivo “.sql” que contiene las querys para la creación de todas las tablas y datos que las contienen.

Se debe Asegurar que el proyecto cuente con las siguientes dependencias en el archivo pom.xml:

```
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>org.apache.tomcat.embed</groupId>
        <artifactId>tomcat-embed-jasper</artifactId>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-security</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
```

```

        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>io.jsonwebtoken</groupId>
        <artifactId>jjwt</artifactId>
        <version>0.7.0</version>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <optional>true</optional>
    </dependency>
</dependencies>

```

## Servicio de Login

Primeramente, es necesario que en la tabla Usuario de nuestra base de datos “lifebank” se encuentre información, por lo que en el archivo “lifebank.sql” se ha añadido una query que inserta datos de 4 usuarios.

Para inicializar el servicio de Login es necesario ingresar a la [url: “localhost:8080/login”](http://localhost:8080/login), especificar que el método de llamada es “post” y como parámetros el nombre del usuario “user” y la contraseña “password”. A continuación, un ejemplo de cómo debe ir el cuerpo de la petición.

```

{
  "user": "life2019",
  "password": "123_vMiles-456*Hola#"
}

```

Para hacer pruebas los usuarios que están registrados en la base de datos son los siguientes:

```

{
  "user": "barapa",
  "password": "1234"
}

```

```
{  
  "user": "ricBermu",  
  "password": "hola"  
}  
  
{  
  "user": "Katy89",  
  "password": "Libro@conejo"  
}
```