

Descripción de la aplicación.

La aplicación fue desarrollada en Java 7, con una base de datos MySql y se utilizó el servidor Payara Micro

El proyecto fue creado en Maven, y para los servicios rest se utiliza la librería Jersey

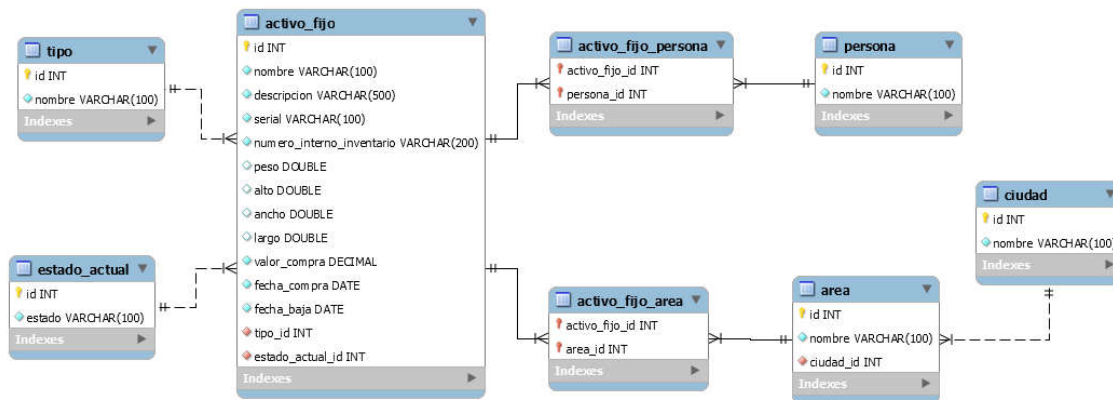
El proyecto se divide en tres capas una capa web (Clase ServicioActivos en el paquete com.asd.api.rest.activos.servicios) en donde se implementan los servicios REST, una capa de control (Clase ActivosBean en el paquete com.asd.api.ejb.activos.impl) implementada en un ejb en la cual se controlan las transacciones y alguna lógica de negocio y una capa de persistencia implementada con jpa.

Se entrega archivo “base_de_datos.sql” para la instalación de la base de datos y archivo de despliegue “asdapi-web-1.0.war”.

Para la generación del war se debe abrir el proyecto en netbeans o eclipse con sus respectivas funcionalidades maven activadas y se debe compilar el proyecto maven asdapi, luego se debe localizar el archivo desplegable en la carpeta “asdapi-web\target”

Las pruebas del servicio se entregan en el archivo “ASD_API.postman_collection.json”

Modelo de Datos



Modelo de Paquetes

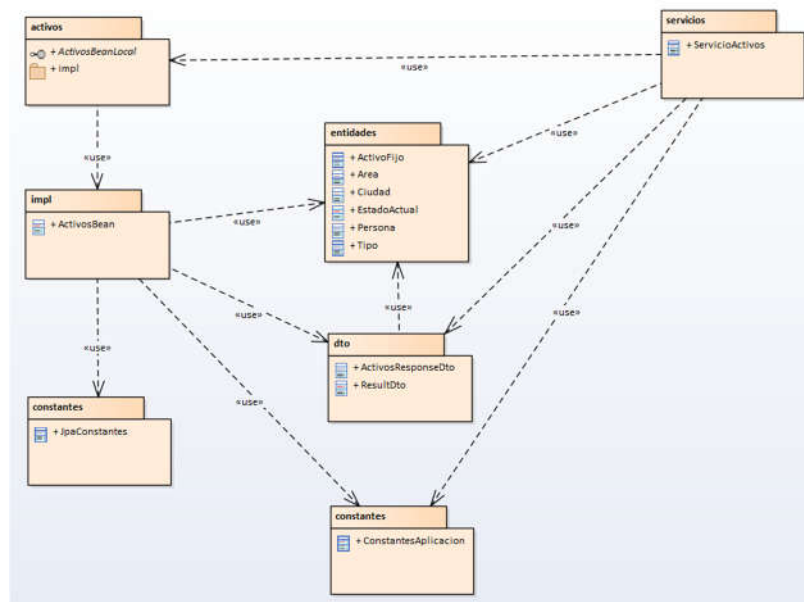
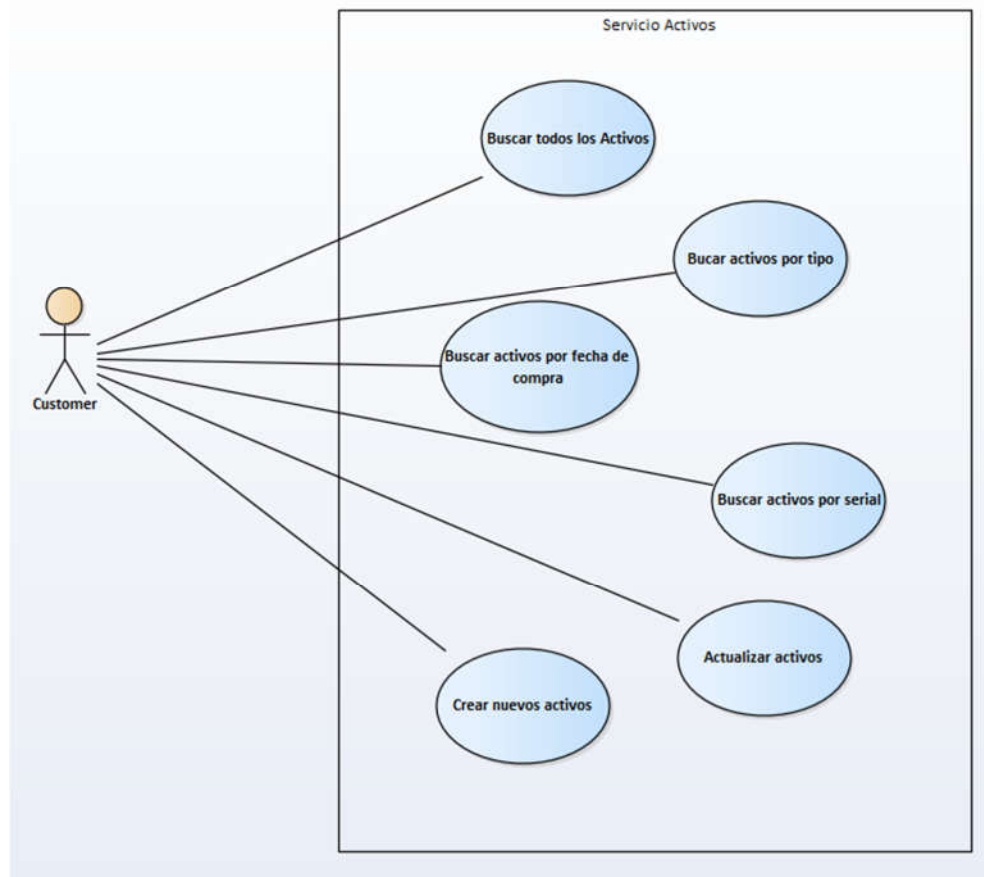


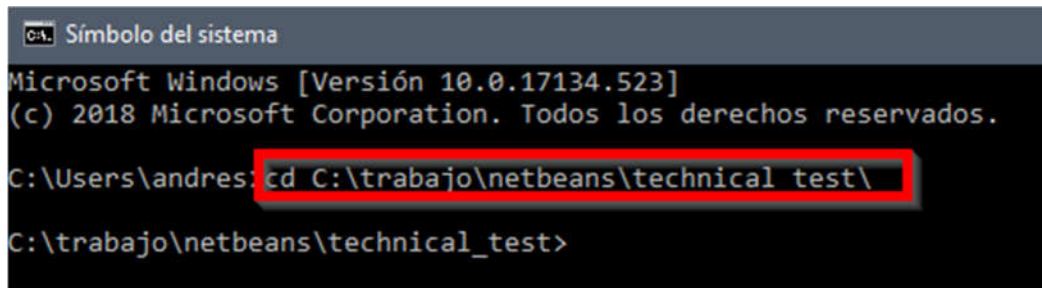
Diagrama de casos de uso



Ejecucion

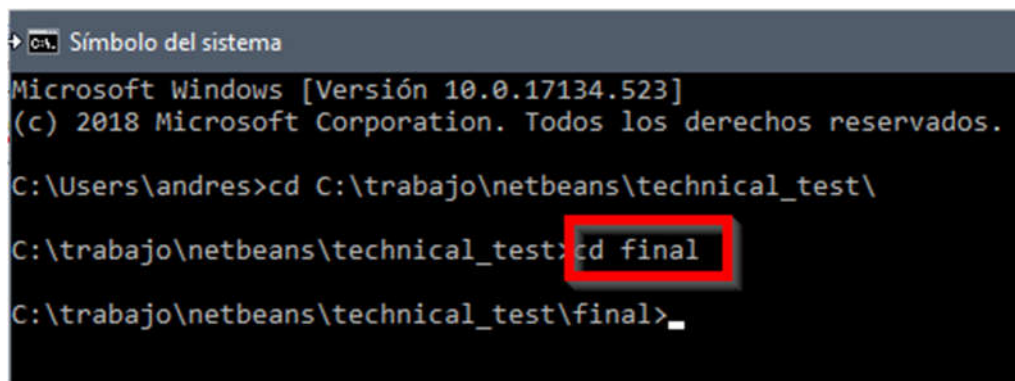
La API se ejecuta en un servidor Payara Micro como se muestra a continuación:

En un terminal ingresar a la carpeta del proyecto, en mi caso se encuentra en la ruta “C:\trabajo\netbeans\technical_test”



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17134.523]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.
C:\Users\andres>cd C:\trabajo\netbeans\technical_test\
C:\trabajo\netbeans\technical_test>
```

Luego ingresar a la carpeta final

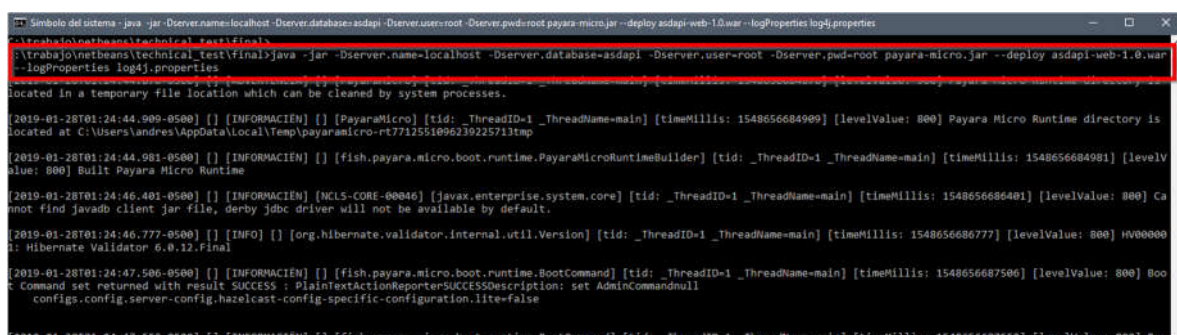


```
Símbolo del sistema
Microsoft Windows [Versión 10.0.17134.523]
(c) 2018 Microsoft Corporation. Todos los derechos reservados.
C:\Users\andres>cd C:\trabajo\netbeans\technical_test\
C:\trabajo\netbeans\technical_test>cd final
C:\trabajo\netbeans\technical_test\final>
```

Y aquí ejecutar el siguiente comando:

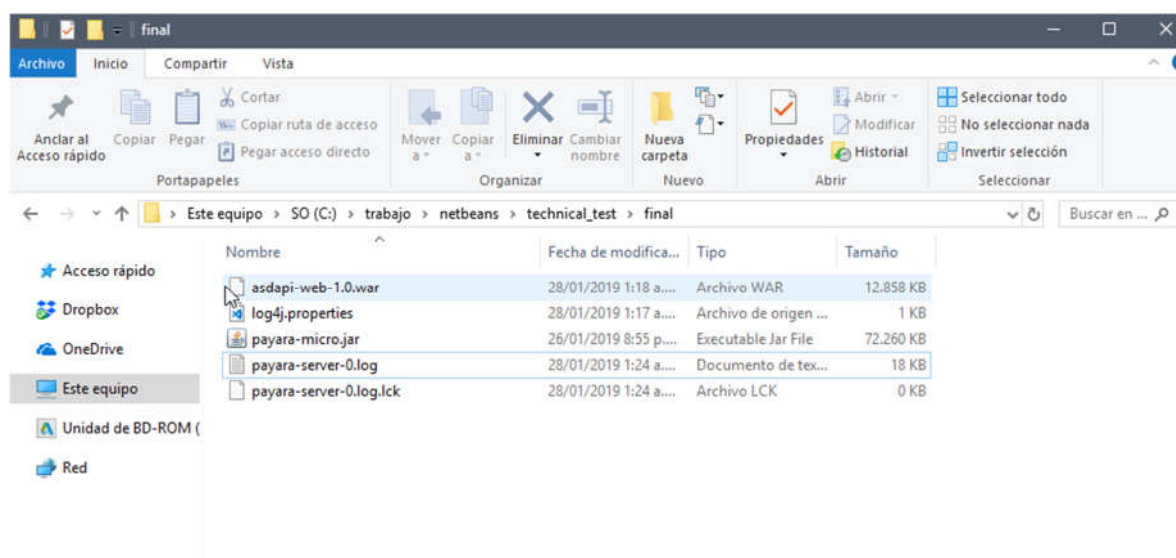
```
java -jar -Dserver.name=localhost -Dserver.database=asdapi -Dserver.user=root -Dserver.pwd=root payara-micro.jar --deploy asdapi-web-1.0.war --logProperties log4j.properties
```

Donde los parámetros “-Dserver.name=localhost -Dserver.database=asdapi -Dserver.user=root -Dserver.pwd=root” establecen la configuración de base de datos



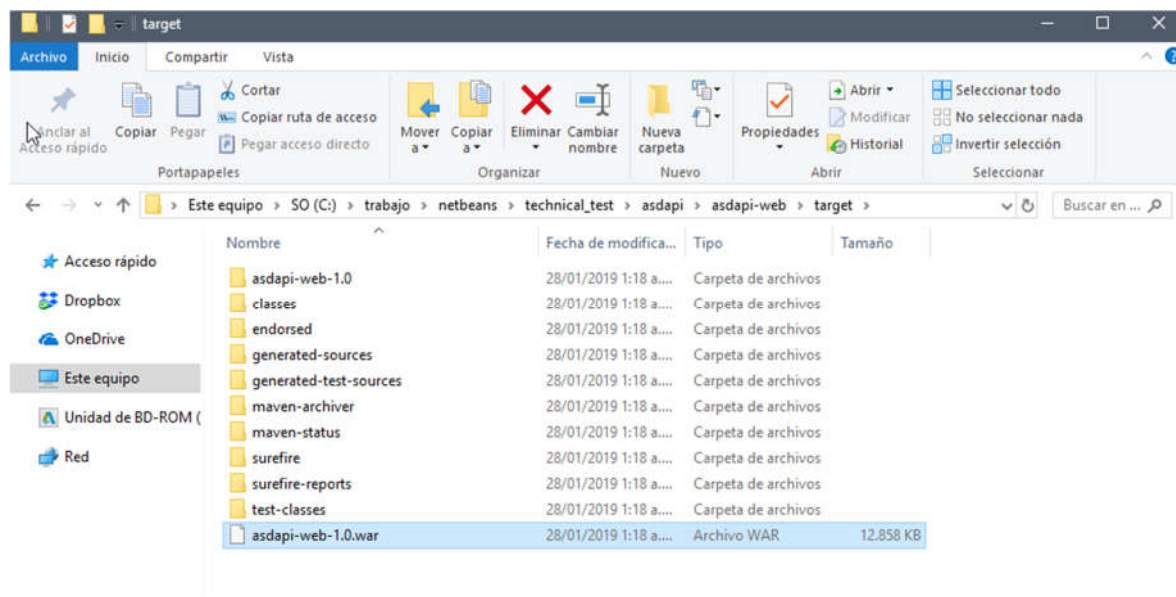
```
Símbolo del sistema - java -jar -Dserver.name=localhost -Dserver.database=asdapi -Dserver.user=root -Dserver.pwd=root payara-micro.jar --deploy asdapi-web-1.0.war --logProperties log4j.properties
C:\trabajo\netbeans\technical_test\final>java -jar -Dserver.name=localhost -Dserver.database=asdapi -Dserver.user=root -Dserver.pwd=root payara-micro.jar --deploy asdapi-web-1.0.war
--logProperties log4j.properties
[2019-01-28T01:24:44.909-0500] [] [INFORMACION] [] [PayaraMicro] [tid: _ThreadID=1 _ThreadName=main] [timeMillis: 1548656684989] [levelValue: 800] Payara Micro Runtime directory is
located at C:\Users\andres\AppData\Local\Temp\payaramicro-rt7712551096239225713tmp
[2019-01-28T01:24:44.981-0500] [] [INFORMACION] [] [fish.payara.micro.boot.runtime.PayaraMicroRuntimeBuilder] [tid: _ThreadID=1 _ThreadName=main] [timeMillis: 1548656684981] [levelV
alue: 800] Built Payara Micro Runtime
[2019-01-28T01:24:46.401-0500] [] [INFORMACION] [NCLS-CORE-00046] [javax.enterprise.system.core] [tid: _ThreadID=1 _ThreadName=main] [timeMillis: 1548656686401] [levelValue: 800] Ca
nnot find javadb client jar file, derby jdbc driver will not be available by default.
[2019-01-28T01:24:46.777-0500] [] [INFO] [] [org.hibernate.validator.internal.util.Version] [tid: _ThreadID=1 _ThreadName=main] [timeMillis: 1548656686777] [levelValue: 800] HV00000
0: Hibernate Validator 6.0.12.Final
[2019-01-28T01:24:47.506-0500] [] [INFORMACION] [] [fish.payara.micro.boot.runtime.BootCommand] [tid: _ThreadID=1 _ThreadName=main] [timeMillis: 1548656687506] [levelValue: 800] Boo
t Command set returned with result SUCCESS : PlainTextActionReporterSUCCESSDescription: set AdminCommandnull
config.config.server-config.hazelcast-config-specific-configuration.lite=false
[2019-01-28T01:24:47.569-0500] [] [INFORMACION] [] [fish.payara.micro.boot.runtime.BootCommand] [tid: _ThreadID=1 _ThreadName=main] [timeMillis: 1548656687569] [levelValue: 800] Boo
```

La carpeta final debe contener los siguientes archivos:



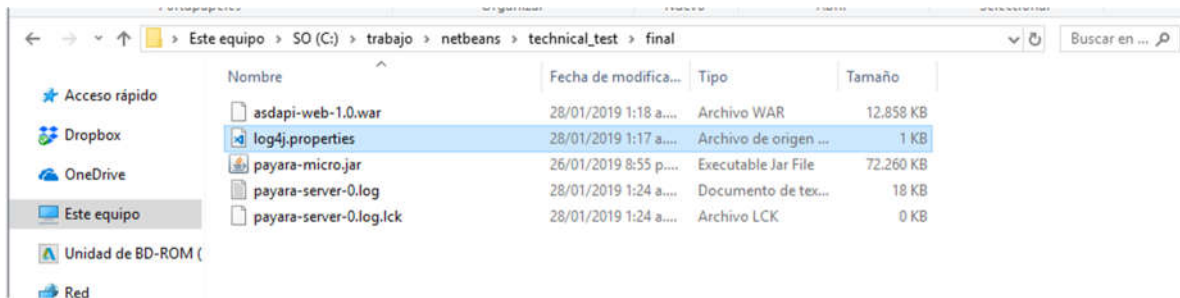
El archivo .lck se crea cuando el servidor inicia la ejecución

El archivo "asdapi-web-1.0.war" es el desplegable generado por el IDE se puede encontrar en la siguiente carpeta después de la compilación:

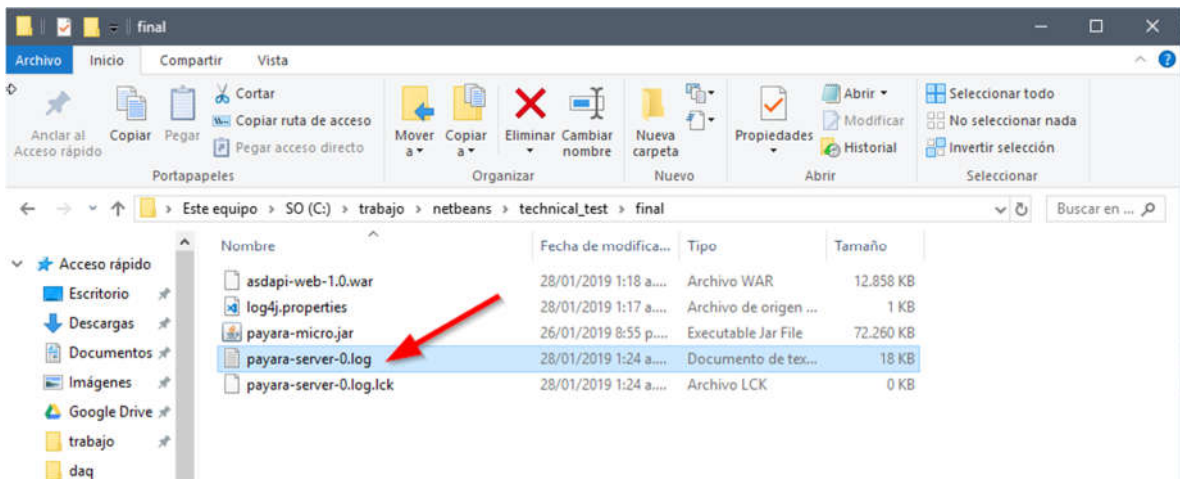


Log4j

La configuración de log4j se encuentra en el archivo "log4j.properties":



Archivo de log



Configuración Junit

Debido a que junit debe ejecutar ejb y jpa fue necesario colocar en el proyecto maven un contenedor embebido de ejb's lo único que se debe configurar para esto es la conexión con base de datos, si no se configura la conexión los casos de prueba fallaran en la compilación.

Para configurar la conexión se debe editar las siguientes propiedades del archivo domain.xml incluido en el proyecto

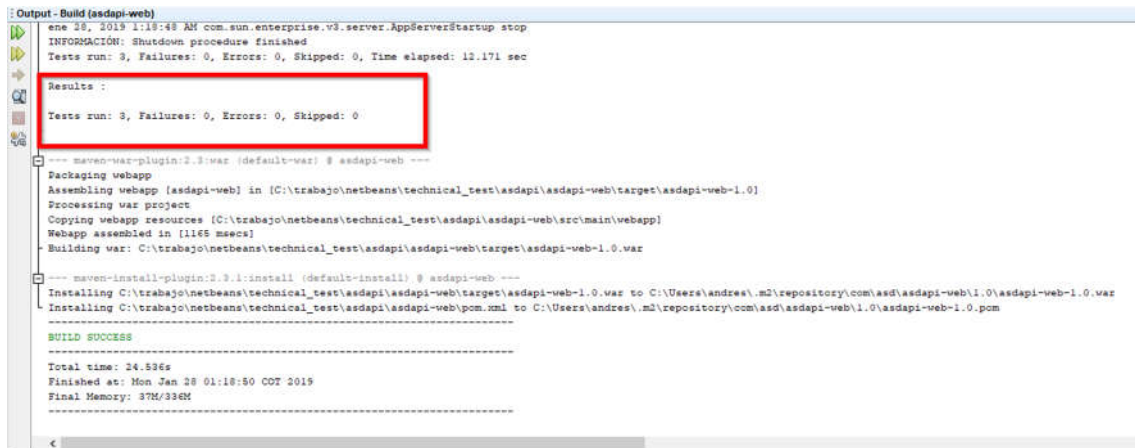
```
<property name="User" value="root"></property>
<property name="Password" value="root"></property>
<property name="PortNumber" value="3306"></property>
<property name="URL" value="jdbc:mysql://localhost:3306/asd_activos_api"></property>
<property name="Url" value="jdbc:mysql://localhost:3306/asd_activos_api"></property>
```

Por favor prestar especial atención en la configuración de **URL** y **Url** ambas deben ser iguales

El archivo domain.xml se encuentra en la siguiente ubicación:



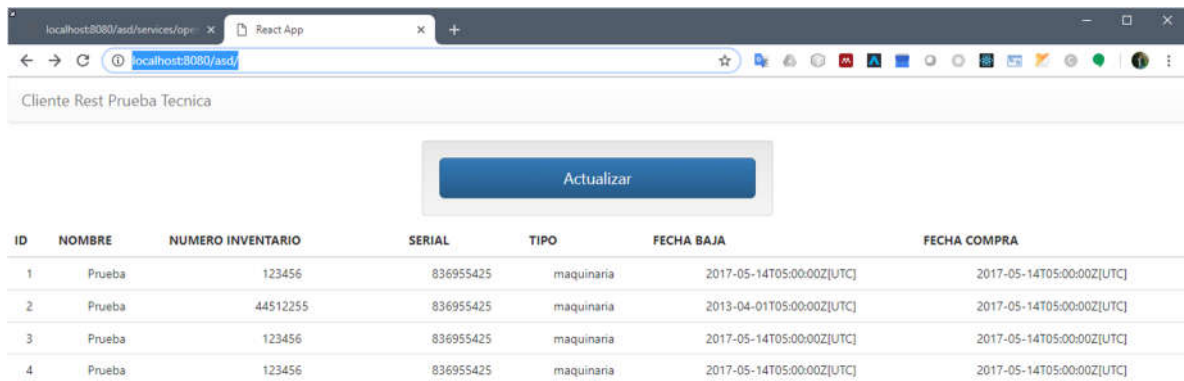
Resultado de la compilación:



Documentación Swagger

El servicio rest se encuentra documentado con swagger para ver la documentación se puede ingresar a la URL:

<http://localhost:8080/asd/services/openapi.json>



ID	NOMBRE	NUMERO INVENTARIO	SERIAL	TIPO	FECHA BAJA	FECHA COMPRA
1	Prueba	123456	836955425	maquinaria	2017-05-14T05:00:00Z[UTC]	2017-05-14T05:00:00Z[UTC]
2	Prueba	44512255	836955425	maquinaria	2013-04-01T05:00:00Z[UTC]	2017-05-14T05:00:00Z[UTC]
3	Prueba	123456	836955425	maquinaria	2017-05-14T05:00:00Z[UTC]	2017-05-14T05:00:00Z[UTC]
4	Prueba	123456	836955425	maquinaria	2017-05-14T05:00:00Z[UTC]	2017-05-14T05:00:00Z[UTC]

Los códigos fuente del cliente se encuentran en la carpeta “technical_test\reactclient”

Autenticación por token

Para los servicios “obtenerPersonas” y “actualizarActivoFijo” se programo un sistema de autenticación por token, el token esta fijado por código y este es “123456” el token debe ser enviado en la cabecera del servicio “Authorization: Bearer 123456”

Cuando se envía un token invalido a cualquiera de estas dos funciones se obtendrá la siguiente respuesta:

GET http://localhost:8080/asd/services/servicioActivos/actualizarActivoFijo

POST http://localhost:8080/asd/services/servicioActivos/actualizarActivoFijo

Send Save

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (0)

KEY	VALUE	DESCRIPTION
Authorization	Bearer 1234562	
Content-Type	application/x-www-form-urlencoded	
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 401 Unauthorized Time: 25 ms Size: 1.18 KB Save Download

Pretty Raw Preview HTML

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <title>Payara Micro #badassfish - Error report</title>
5 <style type="text/css">
6 <!-- H1 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:22px;} H2 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:16px;} H3 {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:14px;} BODY {font-family:Tahoma,Arial,sans-serif;color:black;background-color:white;} B {font-family:Tahoma,Arial,sans-serif;color:white;background-color:#525D76;font-size:12px;} A {color : black;} HR {color : #525D76;--}
7 </head>
8 <body>
9 <h1>HTTP Status 401 - Unauthorized</h1>
10
11 <p>
12 <b>type</b> Status report
13 </p>
14 <p>
15 <b>message</b>Unauthorized
16 </p>
17 <p>
18 <b>description</b>This request requires HTTP authentication.
19 </p>
20 <hr/>
21 <h3>Payara Micro #badassfish</h3>
22 </body>
23 </html>
```

Cuando el token es valido el servicio se ejecutará correctamente:

GET http://localhost:8080/asd/services/servicioActivos/actualizarActivoFijo

POST http://localhost:8080/asd/services/servicioActivos/actualizarActivoFijo

Send Save

Params Authorization Headers (2) Body Pre-request Script Tests Cookies Code Comments (0)

KEY	VALUE	DESCRIPTION
Authorization	Bearer 123456	
Content-Type	application/x-www-form-urlencoded	
Key	Value	Description

Body Cookies Headers (5) Test Results

Status: 200 OK Time: 569 ms Size: 242 B Save Download

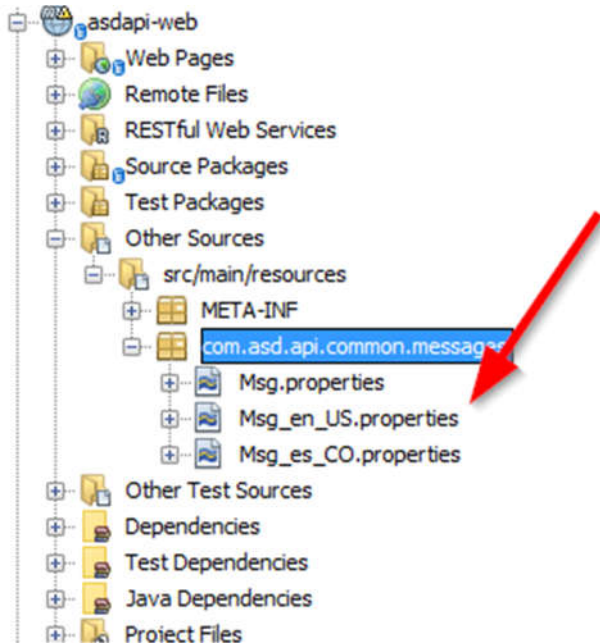
Pretty Raw Preview JSON

```
{
  "result": {
    "resultCode": 200,
    "resultMsg": "Datos actualizados correctamente"
  }
}
```

Internacionalización i18n

El servicio cuenta con internacionalización para los mensajes de respuesta

Los textos de los mensajes se encuentran en los siguientes archivos:



GIT

Todo el desarrollo de la aplicación se encuentra versionado en Git

