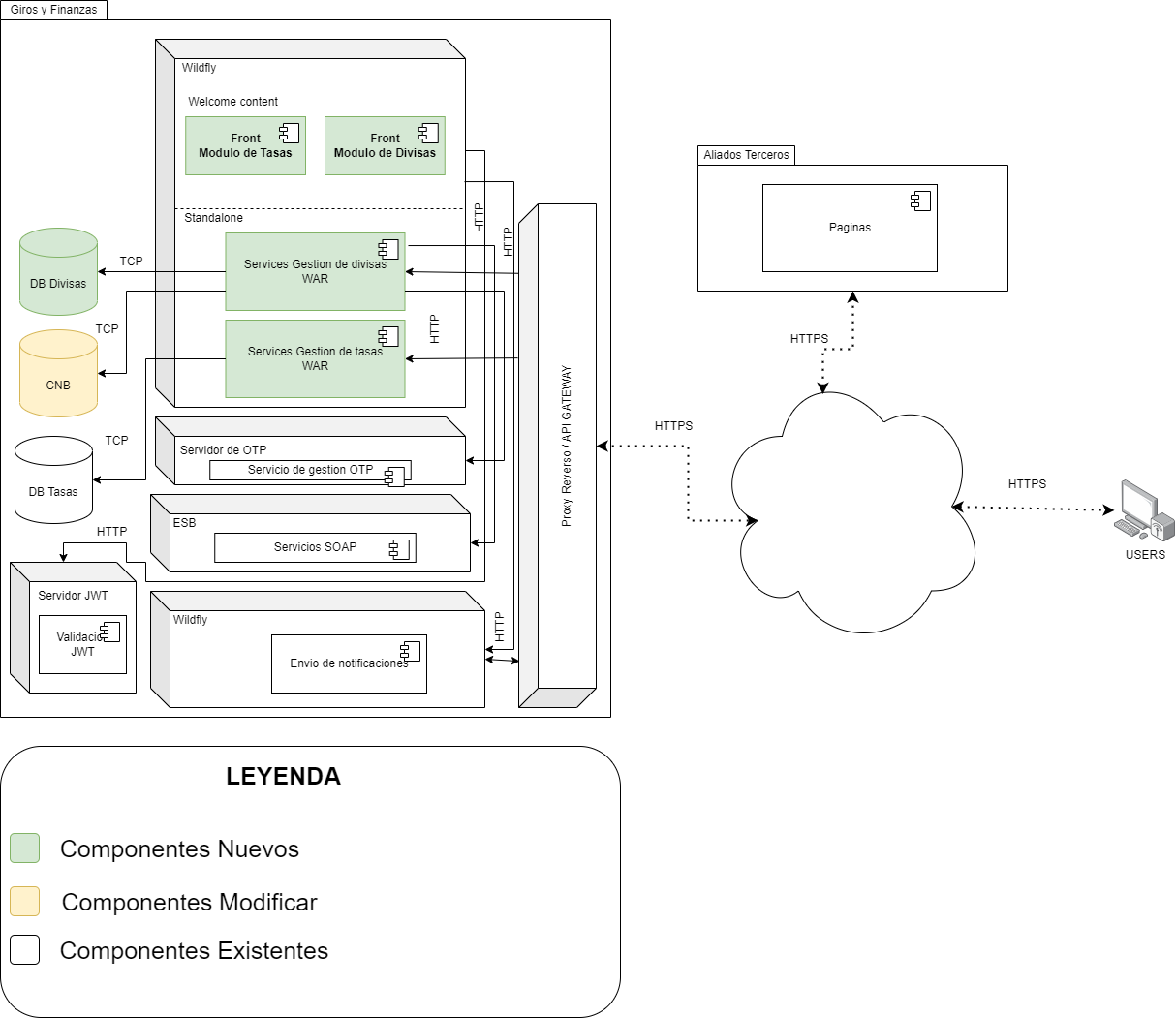
Solicitudes de compra de divisas a domicilio

Venta de divisas

**Propósito del desarrollo:** Venta de divisas se compone de dos front con dos microservicios que permite configurar los parámetros requeridos para que un cliente pueda realizar una solicitud de divisas por la web.

**Arquitectura general:**

****

**Front módulo de tasas:** Aplicación web donde el administrador puede configurar los barrios donde tendrá alcance el domicilio.

**Front módulo de divisas:** Aplicación web donde el usuario puede realizar la solicitud de las divisas.

**Services gestión de divisas war:** Backend en java desde donde se exponen los servicios encargados de realizar la solicitud de las divisas.

**Services gestión de tasas war:** Backend en java desde donde se exponen los servicios que administran los barrios donde se prestara el servicio.

**BD Divisas:** Esquema donde se almacenaran todos los parámetros de aliados, ciudades, municipios, barrios donde se prestara el servicio, también tiene las configuraciones de la aplicación y es donde se almacenaran los datos de la solicitud.

**Log a nivel de server:** Se crea un log del servicio dentro del servidor del wildfly, este log se lo pueden encontrar en dos rutas diferentes dependiendo como se inicie el servidor.

Domain: Se encuentra en la carpeta domain, se ubica en la carpeta con el dominio creado donde se alojara el compilado del backend con el nombre de server.log.

Standalone: Se encuentra en la carpeta standalone – logs – server.log

**Condiciones en las que se realiza la entrega:**

1. Se despliegan los backend en el servidor widlfy por la consola 10.122.7.10:9990.
2. El front se despliega en la ruta welcont-contect del servidor wildfly 10.122.7.10
3. Se crea las tablas y se deja el ambiente apuntando a la base de datos 172.16.16.15:1521/test.

**Importación del proyecto en desarrollo:**

1. Descargar la suit de spring boot.
2. Instalar java 1.8.
3. Instalar el jdk disponible en Oracle.
4. Abrir la suit de spring boot – Ingresar a la pestaña “File” y escoger “Import…” – abrir la carpeta “Maven” y escoger la opción “Existing Maven Projects” – Colocar la ubicación del código fuente y escoger finalizar.

**Nota:** Para este proceso se requiere tener acceso a internet para descargar las librerías de Maven.

**Parámetros requeridos:**

#CONFIGURACION DE LA BASE DE DATOS

spring.datasource.url= **Url de la base de datos, ejemplo: jdbc:oracle:thin@localhost:port:xe**

spring.datasource.username= **Usuario de la base de datos**

spring.datasource.password=  **Contraseña de la base de datos**

spring.datasource.driver-class-name= **Driver de conexión a la base de datos “oracle.jdbc.driver.OracleDriver”**

spring.main.banner-mode= **Modo de banner “off”**

spring.datasource.jndi-name**= Nombre del datasource del servidor**

spring.datasource.jndiName**= Nombre del datasource del servidor**

#CONFIGURACION DEL LOG DEL APLICAC

logging.file.name=PATH/FICHERO.LOG APP.JAR **Ruta de archivo log**

logging.pattern.console=%clr(%d{yy-MM-dd E HH:mm:ss.SSS}){blue} %clr(%-5p) %clr(%logger{0}){blue} %clr(%m){faint}%n **Estructura de cómo se expondrá la consola**

#PARAMETROS DE TOKEN

urlValidateToken=**http://172.16.16.55:8092/girosyfinanzas/oidc/userinfo**

urlGenerateToken=**http://172.16.16.55:8092/girosyfinanzas/oauth/token?grant\_type=password&**

client\_id=client\_id=**3f94c906-4a2a-408b-94c9-064a2a408b74&**

client\_secret=client\_secret=**PTKEq4H8srwhnXUElAoSLRKuLBF\_Z3AbwChM02\_3a7E&**

usernam=**username=jwt.decameron&**

password=**password=Giros123**

#PARAMETROS DE OTP

urlOTP=**http://10.122.7.7:8080/SrvIntOTPDiffuserApi/Send**

urlValidateOTP=**http://10.122.7.7:8080/SrvIntOTPDiffuserApi/Validate**

#PARAMETROS DE LISTA CUMPLIMIENTO

urlListComplice=**http://10.122.7.7:8080/SrvIntCustomerOrqApi/ValidateComplianceList**

#NOTIFICACIONES

urlNotificationEmail=**http://api.messaging-service.com/email/1/send**

#CODIGO DE TRANSACCION

transactionType=**CVDD**

**Desplegar aplicación backend:**

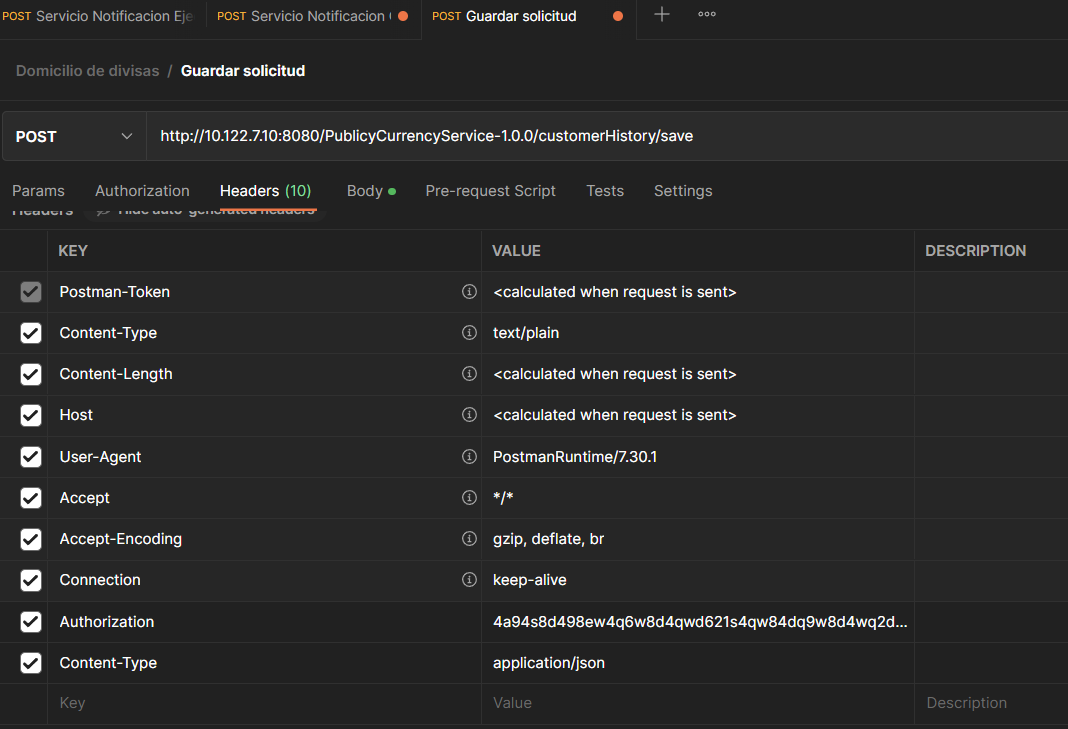
1. Se da click derecho sobre el proyecto y se escoge “Run as” – se escoge la opción “Maven build” y se escribe la palabra “package” en la sección de “goals” para finalizar el proceso.
2. Observer la consola para validar errores
3. Obtener el .warde la carpeta “Target”
4. Ubicar el .war en el servidor donde funcionara y se ejecuta mediante el wildfly.

**Desplegar aplicación backend:**

1. Se le envia el comando ng build en la consola.
2. Tomar el build y alojarlo en la carpeta welcont-contect del aplicativo

**Consumo del servicio para guardar la solicitud de venta de divisas:**

<http://10.122.7.10:8080/PublicyCurrencyService-1.0.0/customerHistory/save>



{

"idCustTransac": 2,

"idAsocCityNeighborhood": 1228131258,

"transactionType": 1,

"transactionAmountLocal": 900000,

"transactionAmountDivisa": "500",

"currencyType": 1,

"currencyValue": 1800,

"customerIdType": 1765,

"idReason": 3,

"customerId": "387084",

"firstname": "Leonardo",

"lastname": "Aedo Jimenez",

"contactNumber": "3218363843",

"contactEmail": "leonardoae798@gmail.com",

"expeditionDate": 1660107600,

"address": "Carrera 14A # 5-46",

"addressDetail": "CASA PRIMER PISO",

"deliveryDate": 1660885200,

"deliveryTime": "7AM - 8AM",

"dateCreate": 1660809283,

"idPlataform": 2

}

Además, se agrega en el entregable un archivo postman con los servicios que se usan para el proyecto administrativo

**Recomendaciones por el proveedor del servicio:**

1. Llevar los desarrollos a entornos dockers y kubernets para así darles una infraestructura independiente a cada servicio con su respectivo ambiente.
2. Se recomienda trabajar con el con el versionado semántico 1.0.0 donde el primer digito representa cambios incompatibles con el api, el segundo digito son cuando se añaden funcionalidades compatibles con versiones anteriores, y el ultimo digito son parches donde se da solución a errores o bugs.
3. El proyecto cuenta con una estructura escalable y fácil de portar dando así la maniobrabilidad de crear nuevas apis bajo un mismo estándar y agilizar los desarrollos. Este detalle técnico puede encontrarlo en los documentos del código fuente.