A white and blue background

Description automatically generated

**Memoria Técnica**

**Plataforma de Verificación de Identidad Única**

**Preparado para: AMEXTRA**

****

**martes 10, octubre de 2023**

**Versión: 6.0**

**Código:**

**CA-PGO-D-E-04-Memoria técnica**

**Preparado por:**

**Julio César Becerril Montiel**

**jbecerril@casystem.com.mx**

# Información General

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha:** | | |  |
|  | | |  |
| **Nombre del Proyecto:** | Plataforma de Verificación de Identidad Única | | |
|  | | | |
| **Fecha de inicio:** |  | **Fecha Planeada de Término:** |  |
|  | | | |
| **Complejidad:** | Alta | **ID del Proyecto:** | PGO-AMEXTRA-001-2022 |
|  | | | |
|  | | | |
| **Solicitante:** | AMEXTRA | | |
|  | | | |
| **Proveedor:** | C&A Systems | | |
|  |  | | |
| **Administrador de Proyecto solicitante:** | Rogelio Jesús Rosales Campos | | |
|  | | | |
| **Administrador de Proyecto C&A Systems:** | Arturo Díaz Alvarado | | |
|  | | | |
| **Líder Técnico:** | Julio Cesar Becerril Montiel | | |

# Control de versiones

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Versión | Fecha | Sección modificada | Descripción | Autor |
| 1.0 | 17/04/2012 | Todo | Se creo la plantilla | Pilar Selina Nájera Villavicencio |
| 2.0 | 27/03/2020 | todo | Se modificó el estándar | Calidad y procesos |
| 3.0 | 23/09/2020 | Todo | Se cambió al dueño del activo | Calidad y procesos |
| 4.0 | 05/10/2021 | Portada | Se modificó la portada y nomenclatura | Calidad y procesos |
| 5.0 | 17/11/2022 | Todo | Se cambió el diseño. | Calidad |
| 6.0 | 10/10/2023 | Portada | Se cambió el diseño. | Calidad |

# Contenido

[1. Información General 2](#_Toc152890480)

[2. Control de versiones 3](#_Toc152890481)

[3. Contenido 4](#_Toc152890482)

[4. Objetivo 5](#_Toc152890483)

[4.1. Software necesario 6](#_Toc152890484)

[5. Pasos de instalación de software 6](#_Toc152890485)

[5.1. Instalación JDK 6](#_Toc152890486)

[5.2. Instalación de tomcat 6](#_Toc152890487)

[5.3. Instalación de Postgresql 8](#_Toc152890488)

[6. Acceso remoto 9](#_Toc152890489)

[6.1. Configuración 9](#_Toc152890490)

[7. Restauración de base de datos 10](#_Toc152890491)

[7.1. Restauración 11](#_Toc152890492)

[8. Instalación de SFTP 12](#_Toc152890493)

[9. Apertura de puertos 13](#_Toc152890494)

[10. Despliegue de aplicativos 13](#_Toc152890495)

[10.1. App Web 13](#_Toc152890496)

[10.1.1. Desplegar archivo war 13](#_Toc152890497)

[10.1.2. Creación de .war con credenciales diferentes 15](#_Toc152890498)

[10.2. Despliegue de microservicios 16](#_Toc152890499)

[10.2.1. Creación de microservicios a partir de código fuente 17](#_Toc152890500)

[10.2.2. Generación de cron 18](#_Toc152890501)

[10.2.3. Verificación de servicios 18](#_Toc152890502)

[9. Firmas de Aceptación 19](#_Toc152890503)

# Objetivo

El objetivo de este documento es proporcionar una descripción detallada y estructurada de los aspectos técnicos inherentes al desarrollo y funcionamiento de las plataformas digitales correspondientes al proyecto “Plataforma de Verificación de Identidad Única”.

Este documento busca ofrecer una comprensión exhaustiva de la arquitectura, tecnologías empleadas, módulos funcionales, protocolos de comunicación, seguridad, y demás elementos esenciales que configuran la infraestructura técnica des aplicaciones tecnológicas de la solución.

La finalidad última radica en facilitar la comprensión integral de la implementación técnica, permitiendo a los desarrolladores, stakeholders y cualquier entidad relevante entender, evaluar y mantener eficazmente las plataformas en cuestión.

## Software necesario

A continuación se enlistan los componentes de software necesario:

* JDK 8 u open jdk 8
* Postgresql v.13
* Tomcat 9
* SO: Windows 10, Windows server 2012, Ubuntu 20, 18, Centos 7, Debian 9
* Archivo .war (app web)
* Archivos .jar (microservicios)

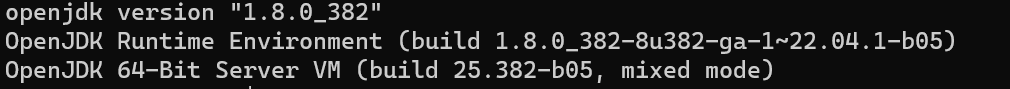
# Pasos de instalación de software

## Instalación JDK

En el sistema operativo de su preferencia instale la versión del jdk 8 u open jdk 8 según sea su preferencia, asegúrese de que la instalación se realizó con éxito invocando la versión con el comando:

java -version

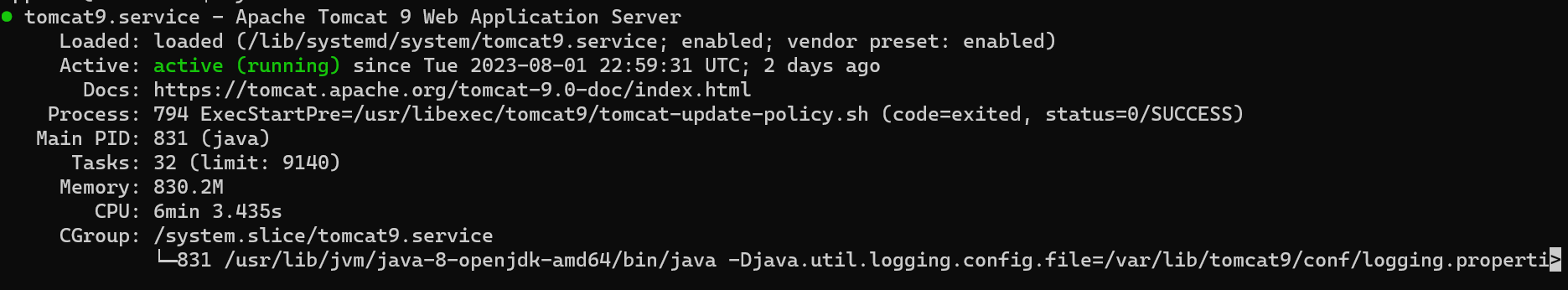
Deberá visualizar una pantalla simiar a lo siguiente:



## Instalación de tomcat

Utilice el instalador de la página oficial de tomcat, en caso de que sea Windows y siga las instrucciones o ejecute el siguiente comando indicado, según la distribución, una vez instalado, puede comprobar su ejecución con el comando:

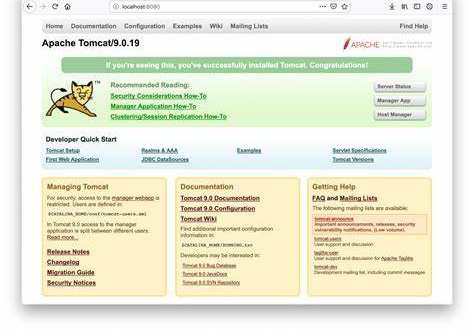
systemctl status tomcat9



En caso de Windows acceda a un navegador con la siguiente url:

localhost:8080

El cual desplegara la pantalla de inicio de tomcat



No olvide configurar el usuario administrador, para poder realizar los despliegues desde la interfaz web, para acceder a ello deberá teclear lo siguiente:

<http://localhost:8080/manager/html>

A screenshot of a computer

Description automatically generated

Se le pedirá el usuario y contraseña que se configuro para el admin-manager de tomcat.

## Instalación de Postgresql

Si es en Windows descargue el instalador oficial de la página, en su versión 13 y siga las instrucciones (le pedirá una contraseña para el usuario postgres, asegúrese de anotarla).

Si es en alguna distribución de linux, ejecute el comando correspondiente para instalar la versión 13 de postgres, así como las dependencias necesarias.

No olvide configurar el usuario postgres, así como su contraseña, para validar la instalación deberá ejecutar el siguiente comando en caso de Linux:

systemctl status postgresql

A computer screen with white text

Description automatically generated

En caso de Windows, revise que en los servicios se encuentre uno con el nombre de postgres:

A screenshot of a computer

Description automatically generated

# Acceso remoto

El acceso remoto a un servidor permite conectarse y gestionar la base de datos desde una ubicación externa al servidor, autorizando a los usuarios o aplicaciones acceder y manipular datos de la base de datos a través de una red, facilitando la administración y consulta de información sin necesidad de estar físicamente en el mismo lugar que el servidor.

Este acceso remoto es beneficioso para equipos distribuidos geográficamente, desarrolladores que trabajan fuera de la ubicación del servidor, y aplicaciones que necesitan interactuar con la base de datos desde ubicaciones remotas. Facilita la colaboración, el monitoreo y la administración eficiente de bases de datos PostgreSQL en entornos descentralizados.

## Configuración

Para configurar el acceso remoto en el servidor de base de datos, será necesario seguir las siguientes indicaciones:

En el archivo **postgresql.conf**:

1. **Localización del archivo**: el archivo postgresql.conf se encuentra en el directorio de configuración de PostgreSQL. Este directorio puede variar según la instalación, pero comúnmente se encuentra en /etc/postgresql/[version]/main/ en sistemas basados en Unix. En caso de un sistema operativo Windows, la ruta al directorio comúnmente se encuentra en la carpeta dentro del disco local bajo la ruta C:\Program Files\PostgreSQL\[versión]\data\
2. **Configuración de direcciones**: buscar la configuración de listen\_addresses en el archivo *postgresql.conf* y configurar para escuchar en todas las direcciones IP. Esto permite conexiones remotas.

listen\_addresses = '\*'

1. **Guardar Cambios**: guardar y cerrar el archivo *postgresql.conf*.

En el archivo **postgresql.conf**:

1. Localización del Archivo: encontrar y abrir el archivo pg\_hba.conf. Este archivo también se encuentra en el directorio de configuración de PostgreSQL-
2. **Añadir Regla para el Usuario "postgres" y la Base de Datos "appnous”**: agregar una regla al final del archivo pg\_hba.conf para permitir el acceso remoto al usuario "*postgres*" y a la base de datos "*appnous*" desde cualquier dirección IP:



El campo METHOD puede ser ajustado según la autenticación deseada (en este caso, "md5" para contraseña encriptada).

1. **Guardar Cambios**: guardar y cerrar el archivo *pg\_hba.conf*.
2. **Reiniciar PostgreSQL**: reiniciar el servicio de PostgreSQL para que los cambios surtan efecto.

# Restauración de base de datos

La restauración de base de datos es el proceso de recuperar y restablecer una base de datos a partir de una copia de seguridad previamente creada. Este procedimiento implica la reinstalación de los datos almacenados en la base de datos en un punto anterior en el tiempo.

La restauración de la base de datos es esencial para recuperarse de eventos no deseados, como pérdida de datos, fallos del sistema o para recrear el estado de la base de datos en un momento específico.

De igual forma, sirve como medida de seguridad crítica y permite a las organizaciones mantener la integridad de la información y la continuidad operativa en caso de eventualidades adversas.

## Restauración

Para poder restaurar una base de datos en PostgreSQL, será necesario utilizar la copia de seguridad (dump) previamente proporcionada, y seguir los siguientes pasos:

1. **Copiar el Archivo de Respaldo**: transferir el archivo de respaldo (dump) al servidor donde se restaurará la base de datos.
2. **Acceder al Servidor**: conectarse al servidor de PostgreSQL donde se realizará la restauración. Es posible conectarse a través de SSH u otro método permitido por las políticas del cliente.
3. **Detener Conexiones (opcional)**: para evitar conflictos durante la restauración, se pueden detener nuevas conexiones a la base de datos. Esto puede lograrse ejecutando el siguiente comando SQL

psql -h localhost -U usuario -d postgres -c "SELECT pg\_terminate\_backend(pg\_stat\_activity.pg\_backend\_pid) FROM pg\_stat\_activity WHERE pg\_stat\_activity.datname = 'appnous';"

1. **Eliminación de la Base de Datos Existente (Opcional):** en caso de que la base de datos ya exista y se desee sobrescribirla, se puede eliminar antes de la restauración.

dropdb -h localhost -U usuario appnous

1. **Creación de una nueva Base de Datos**: si se eliminó la base de datos existente, se debe crear una nueva con el mismo nombre, se puede utilizar el siguiente comando:

createdb -h localhost -U usuario -T template0 appnous

1. **Restauración de la Base de Datos**: se utiliza el comando *pg\_restore* para restaurar la base de datos desde el archivo de respaldo.

pg\_restore -h localhost -U usuario -d appnous *ruta/al/archivo.dump*

1. **Inicio de Conexiones (si se detuvieron en el paso 3)**: en caso de haber detenido las conexiones previamente, se pueden permitir nuevas conexiones ejecutando el siguiente comando SQL:

psql -h localhost -U usuario -d postgres -c "SELECT pg\_terminate\_backend(pg\_stat\_activity.pg\_backend\_pid) FROM pg\_stat\_activity WHERE pg\_stat\_activity.datname = 'appnous';"

1. **Verificación de la Restauración**: Se debe verificar que la restauración se haya realizado correctamente conectándose a la base de datos y revisando la integridad de los datos

psql -h localhost -U usuario -d appnous

# Instalación de SFTP

SI la instalación es en ambiente Windows instale FileZilla server y cree un usuario y contraseña en la ubicación de su preferencia, el usuario debe tener permisos de lectura y escritura, dicho usuario y password deberá ser actualizado en la siguiente tabla de la base de datos:

cat\_configuraciones\_servicios\_externos

Con la condición de id\_servicio\_validacion = 3, ya que ahí es de donde el sistema toma las credenciales:

A screen shot of a computer

Description automatically generated

En caso de Linux, instale el servicio SFTP con el comando que indique su distribución, asegurece de configurar el usuario y password correspondientes y de igual manera actualice el registro en la tabla antes mencionada.

# Apertura de puertos

Tanto para Windows como para Linux asegurece que de su firewall permita el trafico de los siguientes puertos http:

* + - 8080 (portal web)
    - 22 (ssh y sftp)
    - 8000 (Gateway de microservicios)
    - 5432 (postgres, en caso de conectar remotamente)

# Despliegue de aplicativos

## App Web

## Desplegar archivo war

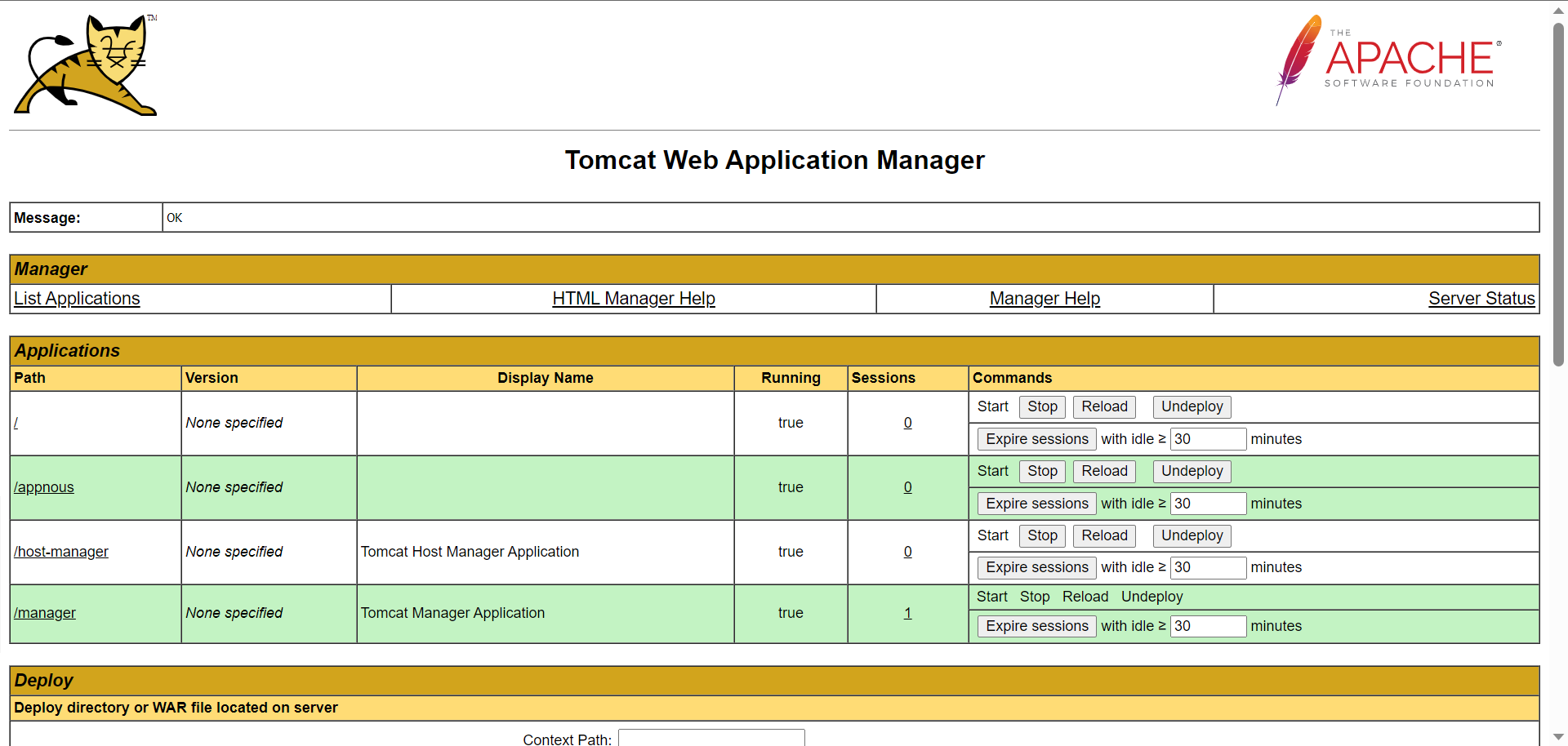
Se le proporcionará un archivo .war, el cual apunta a una base de datos determinada (preguntar cual), dicho war deberá ser cargado en el panel de despliegue de tomcat en la ruta

http://{ip\_o\_nombre\_de\_dominio\_del\_server}:8080/manager/html

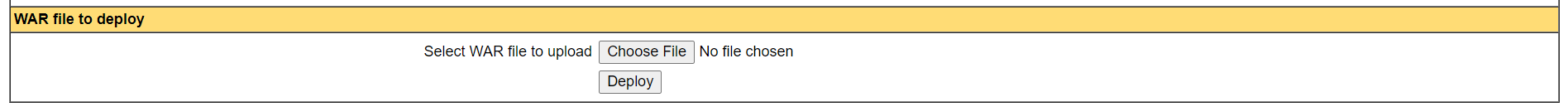
A screenshot of a computer

Description automatically generated

Al momento de cargar sus credenciales vera algo como esto:



Busque la siguiente sección y cargue el war proporcionado:



Una vez cargado, verá el nombre del war que se cargo en la lista de aplicaciones:

A screenshot of a computer

Description automatically generated

Al darle clic, será redireccionado a la url de inicio:

A screenshot of a computer

Description automatically generated

## Creación de .war con credenciales diferentes

En caso de que necesite crear su propio war con otras credenciales de base de datos, necesitará el código fuente del aplicativo (proporcionado por el proveedor), deberá abrir el código con algún editor de código y buscar el siguiente archivo:

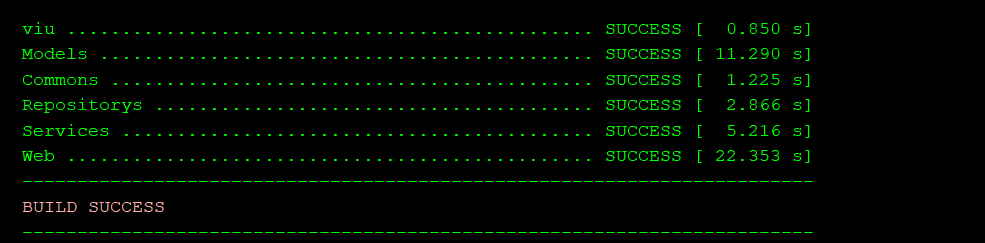
A computer screen shot of a message

Description automatically generated

Realizar los cambios correspondientes a las nuevas credenciales, posterior ejecutar la compilación de los proyectos con el perfil QA

**NOTA**: el sistema es un proyecto multi módulos Maven, el cual utiliza perfiles de compilación en el IDE de desarrollo puede elegir con que perfil compilar, o desde consola ejecutar al proyecto principal:

mvn -PQA clean install



El war resultante lo vera en la ruta del paquete WEB/Web/target.

## Despliegue de microservicios

Realice la siguiente estructura de carpetas en el lugar de su preferencia del servidor:

A green rectangle with white dots

Description automatically generated with medium confidence

Se le proporcionará la misma estructura y dentro encontrara un archivo con extensión .jar correspondiente al servicio en cuestión, en caso de estar en Linux, dar permisos de ejecución a los .jar con el comando chmod

De igual manera se le proporcionara un script para levantar de manera conjunta los servicios (caso linux):

“levantaMicros.sh”

Dar permisos de ejecución y ejecutar ”./levantaMicros.sh”

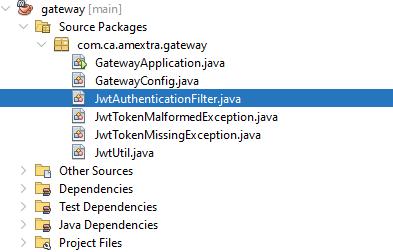
En caso de que sea Windows, deberá ejecutar uno por uno con el comando java -jar ‘nombre\_del\_servicio.jar’ , o ejecutar el archivo con extensión .bat que ejecute todo en conjunto.

**NOTA**: los microservicios proporcionados se encuentran configurados con un host, usuario y password de base de datos definido por el cliente, si desea cambiar esta configuración, realizar la siguiente sección.

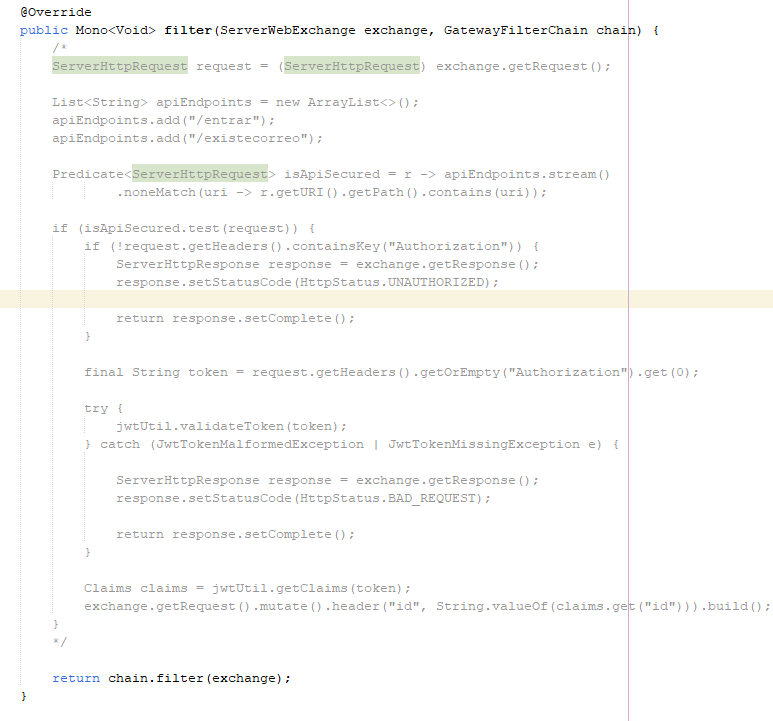
## Activación / Desactivación de JWT (JSON Web Token) para los microservicios

Para poder desactivar el token lo que se debe realizar es lo siguiente:

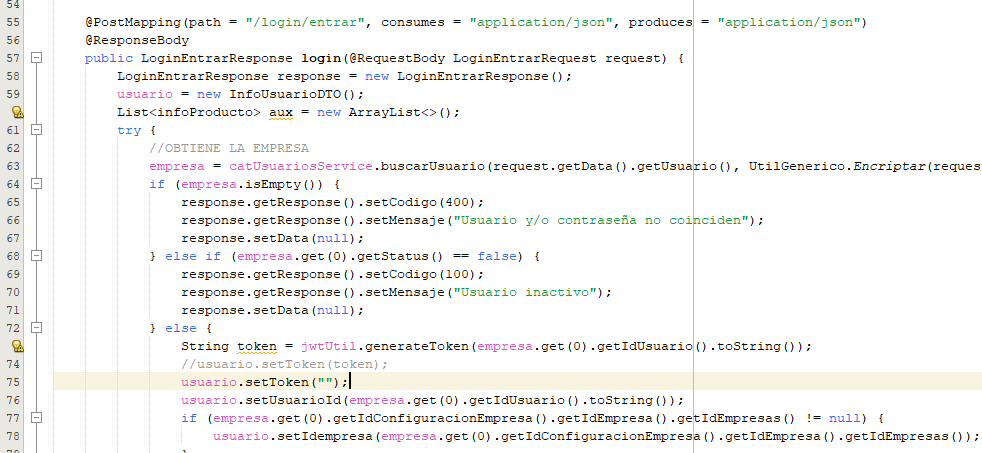
Se debe posicionar sobre el proyecto **gateway**, una vez dentro del proyecto se busca la carpeta llamada **Source Packages** y después abrimos **com.ca.amextra.gateway** dentro de estas se encuentran los archivos de configuración en específico necesitamos abrir el archivo llamado **JwtAuthenticationFilter.java**



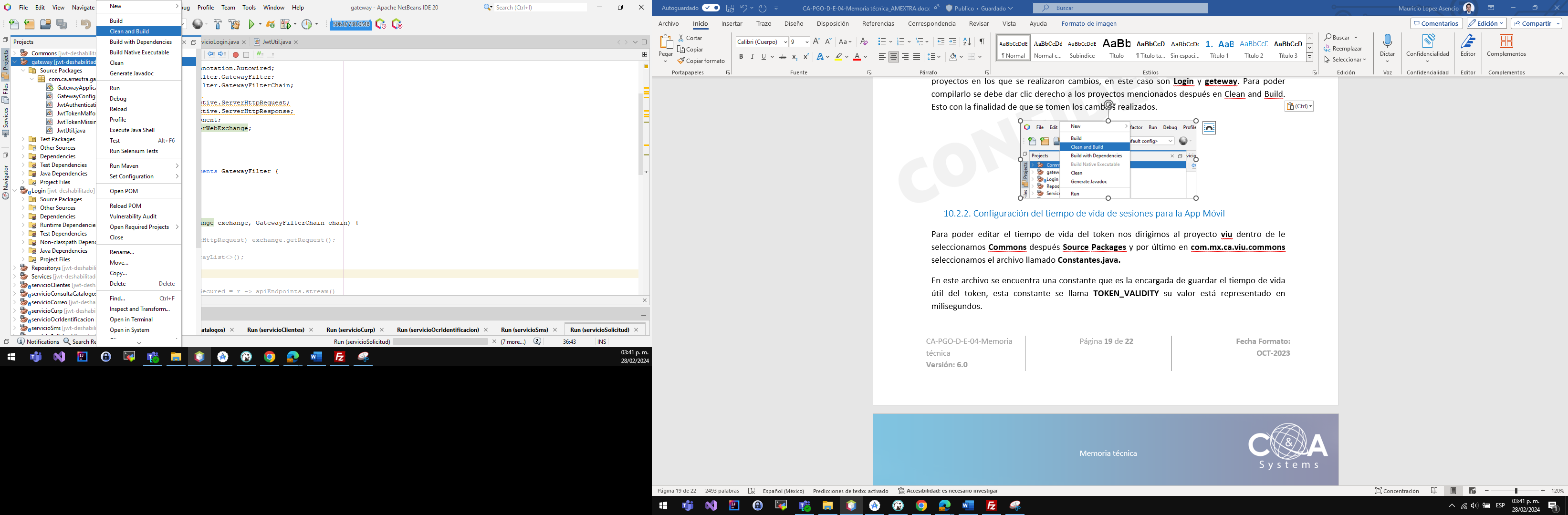
El siguiente paso después de abrir el archivo **JwtAuthenticationFilter** es posicionarnos en la primera línea después de empezar el método **filter**, se procede a comentar el contenido de ese método dejando sin comentar el **return**, dejando inhabilitado el método.



El paso siguiente es posicionarnos en el proyecto del microservicio de **login**, entrar en la carpeta **Source Packages** y seguido de eso en **com.mx.ca.viu.microservicios.login.Login** buscaremos el archivo llamado **ControllerServicioLogin.java** buscamos la línea donde se utiliza el token, la línea 74 o 75 que contiene lo siguiente **usuario.setToken(token);** la comentaremos y mandaremos el token vacío así como se muestra en la imagen.



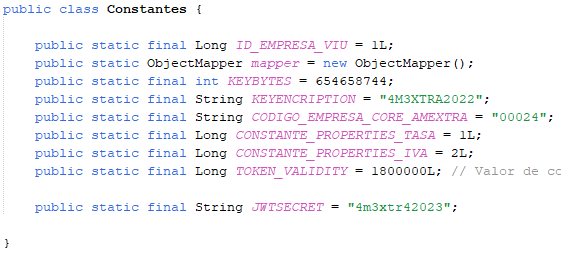
Cuando se termine de realizar las modificaciones correspondientes deben compilar los proyectos en los que se realizaron cambios, en este caso son **Login** y **gateway**. Para poder compilarlo se debe dar clic derecho a los proyectos mencionados después en **Clean and Build**. Esto con la finalidad de que se tomen los cambios realizados.



## Configuración del tiempo de vida de sesiones para la App Móvil

Para poder editar el tiempo de vida del token nos dirigimos al proyecto **viu** dentro del seleccionamos **modules** y después **Commons** después **Source Packages** y por último en **com.mx.ca.viu.commons** seleccionamos el archivo llamado **Constantes.java.**

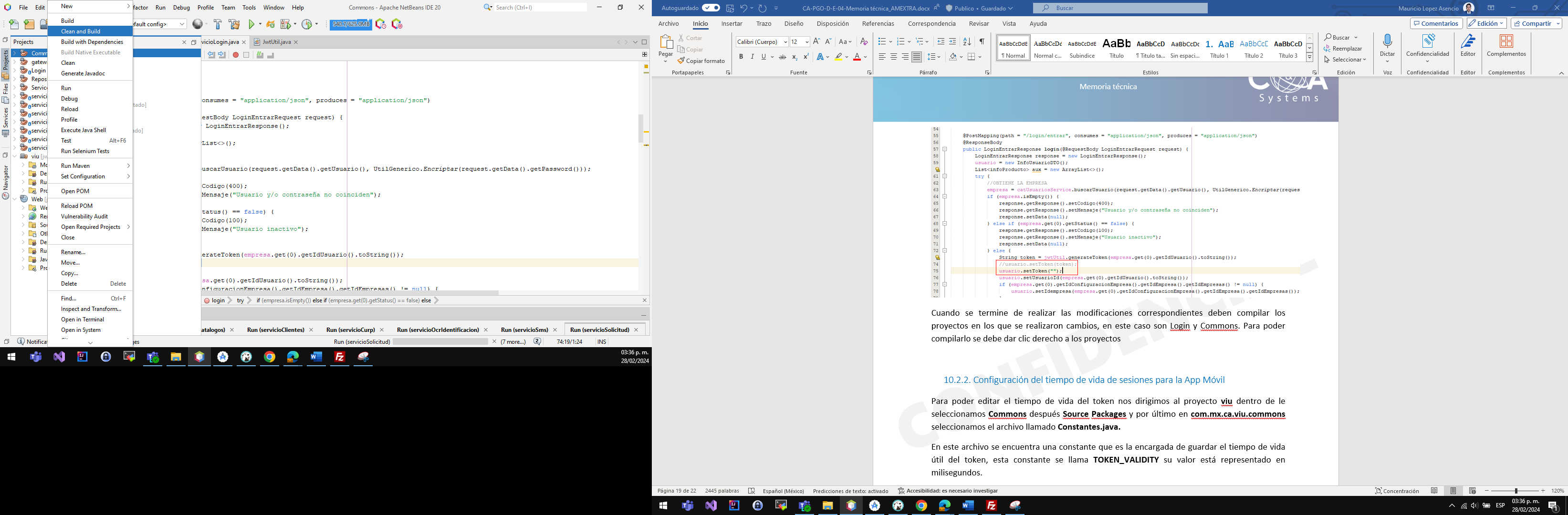
En este archivo se encuentra una constante que es la encargada de guardar el tiempo de vida útil del token, esta constante se llama **TOKEN\_VALIDITY** su valor está representado en milisegundos.



El valor actual está en 1800000L que es el equivalente a 30 minutos.

|  |  |  |
| --- | --- | --- |
| HORAS | MINUTOS | MILISEGUNDOS |
| 1 | 60 | 3600000 |
| 1.5 | 90 | 5400000 |
| 2 | 120 | 7200000 |
| 2.5 | 150 | 9000000 |
| 3 | 180 | 10800000 |

De igual manera que en el punto anterior debemos recompilar el proyecto al terminar de realizar los cambios. Dando clic derecho al proyecto de **commons** y después en **Clean and Build**



## Creación de microservicios a partir de código fuente

Con su IDE de preferencia, abrir cada proyecto de microservicio y buscar el siguiente archivo en a la siguiente ubicación:

A screenshot of a computer

Description automatically generated

Realizar los cambios necesarios y compilar el proyecto, si desea hacerlo desde consola, diríjase a la raíz del proyecto y ejecute el comando:

mvn clean install

El cual generara un archivo .jar en la ubicación: carpeta\_servicio/target:



Deberá copiar dicho archivo en su carpeta correspondiente en el servidor de ejecución y realizar los pasos mencionados previamente.

## Generación de cron

En caso de Linux deberá generar un cron tab que ejecute el archivo levantaMicros.sh de manera automática, para garantizar la ejecución de los servicios en todo momento.

En caso de Windows, deberá crear una tarea programada que ejecute un .bat que levantará los servicios en caso de reinicio del server.

**NOTA**: para el tomcat, postgresql o sftp no es necesario crear crons o tareas programadas de ejecución automática, ya que estos al momento de instalarse, las crearon por defecto.

## Verificación de servicios

Para verificar si los microservicios están corriendo, deberá ejecutar en consola el siguiente comando y buscar lo siguiente:

ps -aux

A screen shot of a computer screen

Description automatically generated

Si encuentra esos elementos, podemos corroborar que todos los microservicios se están ejecutando de forma exitosa.

# Firmas de Aceptación

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NOMBRE** | **ROL PROYECTO** | **COMPAÑÍA** | **FIRMA** | **FECHA** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

[En este apartado se deberán asentar los nombres y cargos de los responsables de la elaboración, revisión y aprobación de los casos de uso, incluyendo al administrador del proyecto; se deberán obtener las firmas autógrafas correspondientes e indicar las fechas de firma.]