# MANUAL DE DESPLIEGUE



### **Autor**

John Esteban Orrego Anaya

Jhon Harvy Vélez Rodríguez

Faculta de ingenierías

2021

### Contenido

1.	Objetivo	3
	Instalación y configuración del servidor	
	2.1 Requisitos mínimos del servidor	4
	2.2 Instalación librerías del servidor.	4
	2.2.1 Instalación Java	4
	2.2.2 Instalación y configuración NGINX	4
	2.2.3 Instalación y configuración PostgreSQL	6
3.	Instalación y configuración de la aplicación	6
	3.1 Requisitos mínimos	6
	3.2 Instalación Frontend Órbita Académica.	6
	3.3 Instalación Backend Órbita Académica.	7
	3.4 Logs de la aplicación.	8
	3.5 Comprobar despliegue aplicación.	8

## 1. Objetivo

El objetivo de este documento es facilitar la instalación y ejecución de los artefactos que hacen parte de la solución.

#### 2. Instalación y configuración del servidor

#### 2.1 Requisitos mínimos del servidor

- Sistema Operativo Windows o Linux
- Memoria RAM de 6 GB.
- Espacio en disco mínimo de 16 GB
- Acceso a Internet (10MB Subida / 10MB Descarga)

**Nota:** Para el despliegue de la solución se utilizó una máquina virtual EC2 en Amazon Web Services por lo que los comandos de instalación y configuración de librerías se ejecutan de acuerdo con la documentación de AWS.

#### 2.2 Instalación librerías del servidor.

Para la ejecución de los artefactos que conforman la Solución de Software 'Órbita Académica' es necesario la instalación de ciertos componentes:

- Maguina Virtual de Java Java 8.
- Servidor web NGINX
- Base de datos PostgreSQL.

#### 2.2.1 Instalación Java

Para la instalación de java es necesario la ejecución de los siguientes comandos, Amazon Web Services recomienda la instalación de su propia máquina virtual de java por motivos de seguridad.

- sudo amazon-linux-extras enable corretto8
- sudo yum install java-1.8.0-amazon-corretto-devel

#### 2.2.2 Instalación y configuración NGINX

Para la instalación de NGINX es necesario la ejecución de los siguientes comandos

- sudo amazon-linux-extras install nginx1
- sudo systemctl start nginx
- sudo systemctl enable nginx

Para la configuración del NGINX se realiza las siguientes operaciones

- sudo vi /etc/nginx/nginx.conf
- Reemplazar la configuración del archivo nginx.conf por la siguiente:

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;
include /usr/share/nginx/modules/*.conf;
events {
   worker_connections 1024;
http {
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                     '"$http_user_agent" "$http_x_forwarded_for"';
    access_log /var/log/nginx/access.log main;
    sendfile
   tcp_nopush
   tcp_nodelay
   keepalive_timeout 65;
    types_hash_max_size 4096;
    include
                      /etc/nginx/mime.types;
   default_type
                      application/octet-stream;
    include /etc/nginx/conf.d/*.conf;
    server {
       listen
                   80;
       listen
                   [::]:80;
       server_name _;
                    /usr/share/nginx/html/OrbitaAcademica;
       index index.html index.htm;
       try_files $uri $uri/ /index.html;
       include /etc/nginx/default.d/*.conf;
       proxy_set_header Host $remote_addr;
       location /api {
               proxy_pass http://localhost:8908;
       error_page 404 /404.html;
           location = /40x.html {
       error_page 500 502 503 504 /50x.html;
           location = /50x.html {
```

#### 2.2.3 Instalación y configuración PostgreSQL

Para la instalación de PostgreSQL es necesario la ejecución de los siguientes comandos

- sudo amazon-linux-extras | grep postgre
   sudo tee /etc/yum.repos.d/pgdg.repo<<EOF
   [pgdg12]
   name=PostgreSQL 12 for RHEL/CentOS 7 x86\_64
   baseurl=https://download.postgresql.org/pub/repos/yum/12/redhat/rhel-7-x86\_64
   enabled=1
   gpgcheck=0
   EOF</li>
- sudo yum makecache
- sudo yum install postgresql12 postgresql12-server
- sudo /usr/pgsql-12/bin/postgresql-12-setup initdb
- sudo systemctl enable --now postgresql-12

#### 3. Instalación y configuración de la aplicación

#### 3.1 Requisitos mínimos

- Frontend: OrbitaAcademica.zip
- Backend: config-server-0.0.1-SNAPSHOT.jar
- Backend: eureka-server-0.0.1-SNAPSHOT.jar
- Backend: gateway-server-0.0.1-SNAPSHOT.jar
- Backend: auth-orbita-1.0.0.jar
- Backend: evaluaciones-orbita-1.0.0.jar
- Backend: parametrizacion-orbita-1.0.0.jar
- Backend: reportes-orbita-1.0.0.jar
- Backend: tmo-orbita-1.0.0.jar
- BaseDatos: scriptIni.sql

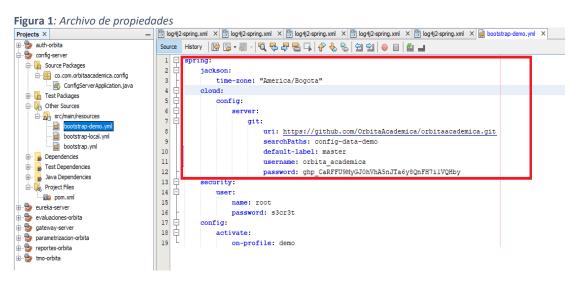
#### 3.2 Instalación Frontend Órbita Académica.

Copiar OrbitaAcademica.zip en la ubicación /usr/share/nginx/html/ y ejecutar los siguientes comandos en la consola o terminal.

- sudo OrbitaAcademica.zip
- sudo chmod -R 755 OrbitaAcademica/
- sudo systemctl restart nginx

#### 3.3 Instalación Backend Órbita Académica.

**Nota:** En el archivo de propiedades del código fuente del proyecto config-server se encuentra la configuración desde donde los demás microservicios toma la configuración de cada uno de ellos, por lo que, si el proyecto se migra a un repositorio diferente, la configuración de este archivo se debe actualizar.



Para realizar la instalación de los artefactos del Backend es necesario ejecutar los siguientes procedimientos y comandos en la consola o terminal de la máquina virtual.

- sudo mkdir /opt/microservicios/
- cd mkdir /opt/microservicios/
- Copiar todos los artefactos de extensión .jar en la ubicación anteriormente creada e iniciarlos con los siguientes comandos y orden.
- java -jar /opt/microservicios/config-server-0.0.1-SNAPSHOT.jar &
- java -jar /opt/microservicios/eureka-server-0.0.1-SNAPSHOT.jar &
- java -jar /opt/microservicios/gateway-server-0.0.1-SNAPSHOT.jar &
- java -jar /opt/microservicios/auth-orbita-1.0.0.jar &
- java -jar /opt/microservicios/evaluaciones-orbita-1.0.0.jar &
- java -jar /opt/microservicios/parametrizacion-orbita-1.0.0.jar &
- java -jar /opt/microservicios/reportes-orbita-1.0.0.jar &
- java -jar /opt/microservicios/tmo-orbita-1.0.0.jar &

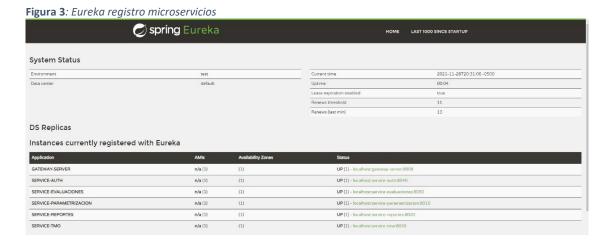
#### 3.4 Logs de la aplicación.

Cada uno de los microservicios que componen el backend escribre su propio log en la ruta /opt/microservicios/logs/ haciendo corte por hora y guardándose en una carpeta por día transcurrido.

Figura 2: Configuración Logs 🖺 log#j2-spring.xml 🗴 🖫 log#j2-spring.xml 🗴 🖫 log#j2-spring.xml 🗴 🖫 log#j2-spring.xml 🗴 🖫 log#j2-spring.xml 🗴 Source History | 🕝 🖫 - 🗐 - 💆 😓 🖶 📮 | 🌳 😓 🥦 | 🕮 💁 | 🔘 🕞 | 🤝 🗳 <?xml version="1.0" encoding="UTF-8"?> <!-- Logging Properties --> 4 -5 6 7 -8 -9 <Properties> <Property name="LOG PATTERN">[%tn,%tid] [%d{yyyy-MM-dd'T'HH:mm:ss.SSSZ}] [%p{lowerCase=false}] [%c{1.}] [%M]: %m%nproperty> <Property name="APP\_LOG\_ROOT">/opt/microservicios/logs/tmo/</Property> </Properties> <Appenders> <!-- Console Appender --> 10 <Console name="Console" target="SYSTEM OUT" follow="true"> <PatternLayout pattern="\${LOG\_PATTERN}"/> 11 12 </Console> <RollingFile name="tmoLog" fileName="\${APP\_LOG\_ROOT}/tmo.log"
filePattern="\${APP\_LOG\_ROOT}/\$d(yyyy-MM-dd)/tmo-%d(yyyy-MM-dd-HH).log.gz" > 13 14 <PatternLayout pattern="\${LOG\_PATTERN}"/> 15 16 <Policies> 17 <TimeBasedTriggeringPolicy interval="1" /> 18 </Policies> 19 <DefaultRolloverStrategy max="1"/> </RollingFile> 20 21 </Appenders>

#### 3.5 Comprobar despliegue aplicación.

Una vez ejecutados los microservicios, al ingresar a la ip:8082 de la máquina virtual donde está instala la aplicación se debería de observar la siguiente pantalla con los microservicios registrados.



Comprobado que todos los microservicios están iniciados, se puede ingresar a un explorador y en dirección colocar la ip de la máquina virtual o el dominio de esta y empezar a usar la aplicación.

