**PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU**

**FACULTAD DE CIENCIAS E INGENIERÍA**

**INGENIERÍA INFORMÁTICA**



**Guía de Instalación**

**Profesor del curso:** Abraham Eliseo Dávila Ramón

**Equipo:** 3C

**Horario:** 0981

**Integrantes:**

* Castro Garcia, Santiago Miguel 20200586
* Flores Gonzales, Jose Luis Ángel 20190202
* Huaranga Hanco, Saul Florentino 20172360
* Pinto Quiroz, Renzo Gabriel 20201491
* Veramendi Malpartida, Diego Gustavo 20200754

**Lima, 25 de Junio del 2024**

**Historial de revisión**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Responsable |
| 25/06/2024 | 1.0 | Guía de Instalación | Equipo 3C |

# Introducción

El propósito de este documento es indicar los pasos necesarios para el despliegue de los componentes de la aplicación web, los cuales son el Backend y Frontend. Se utilizó el framework de Java Spring Boot para el desarrollo del Backend, y React, para el Frontend. Además, se empleó Nginx para la configuración del servidor web.

# Instalación de Backend

Para correr el lado de backend del proyecto, se requiere la versión 17 o superior de Java (https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html).

Luego, en \backend\redex\_back, abrir Command Prompt (cmd) y ejecutar

mvnw.cmd clean install

mvnw.cmd clean package

# Instalación de Frontend

Para correr el lado de frontend del proyecto, se requiere tener la versión más reciente de NodeJS (https://nodejs.org/en).

Luego de esto, el proyecto requiere instalar sus propias dependencias. Abriendo una terminal desde el directorio del proyecto (/frontend), introducir el comando “npm install”.

Tras haber instalado estas, se puede iniciar el proyecto con “npm run dev”, y para una versión optimizada “npm run build” y “npm run start”

# Obtención de los archivos compilados

Dentro de la carpeta \backend\redex\_back, abrir Command Prompt (cmd) y ejecutar

mvnw.cmd clean package

De esta manera, se obtendrá el compilado del proyecto \backend\redex\_back\target

Luego, para compilar el front end, dirigirse a la carpeta \frontend, abrir abrir Command Prompt (cmd) y ejecutar

npm run build

De esta manera, se obtendrá el compilado del proyecto en

\frontend\out

# Conexión con la instancia

En este caso, se ha utilizado PuTTY para conectarse con la instancia por medio de SSH.

Luego de ingresar con las credenciales correspondientes, ejecutar:

$sudo apt update

$sudo apt install openjdk-11-jdk

$sudo apt install nginx

$sudo apt install nodejs

Asegurarse que se encuentra en /home/(nombre de usuario)

Por ejemplo, /home/inf226.981.3c

$cd /home/inf226.981.3c

Luego, ejecutar  
$sudo mkdir app

$sudo mkdir logs

$cd /var/www

$sudo mkdir redex

$ sudo chown -R $USER:$USER /var/www/redex

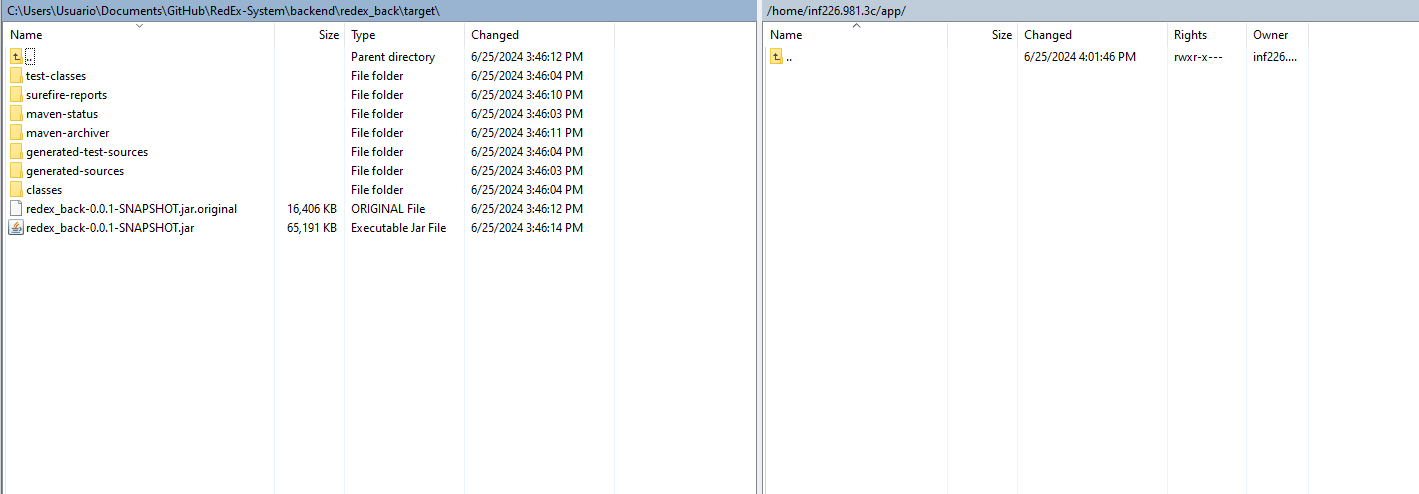
$cd /home/inf226.981.3c

Consideración:

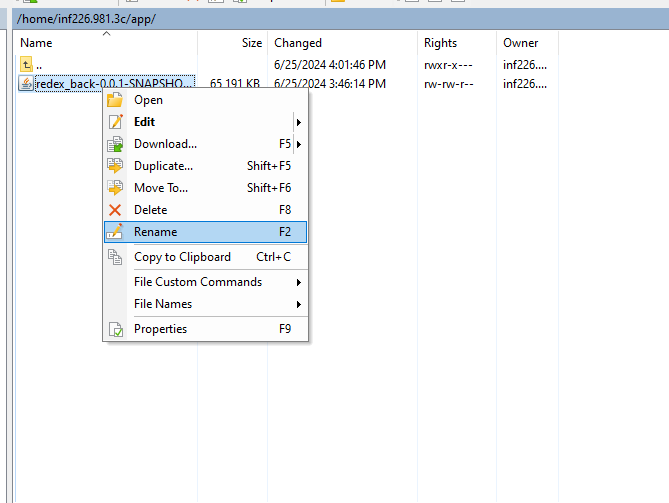
Si la ruta de home es diferente a /home/inf226.981.3c, debe cambiar las líneas del archivo \backend\redex\_back\src\main\resources\log4j2.xml para que coincida con la ruta correspondiente. Luego tendrá que volver a compilar el proyecto backend (indicado en el paso 4).

# Transferencia de archivos a la instancia

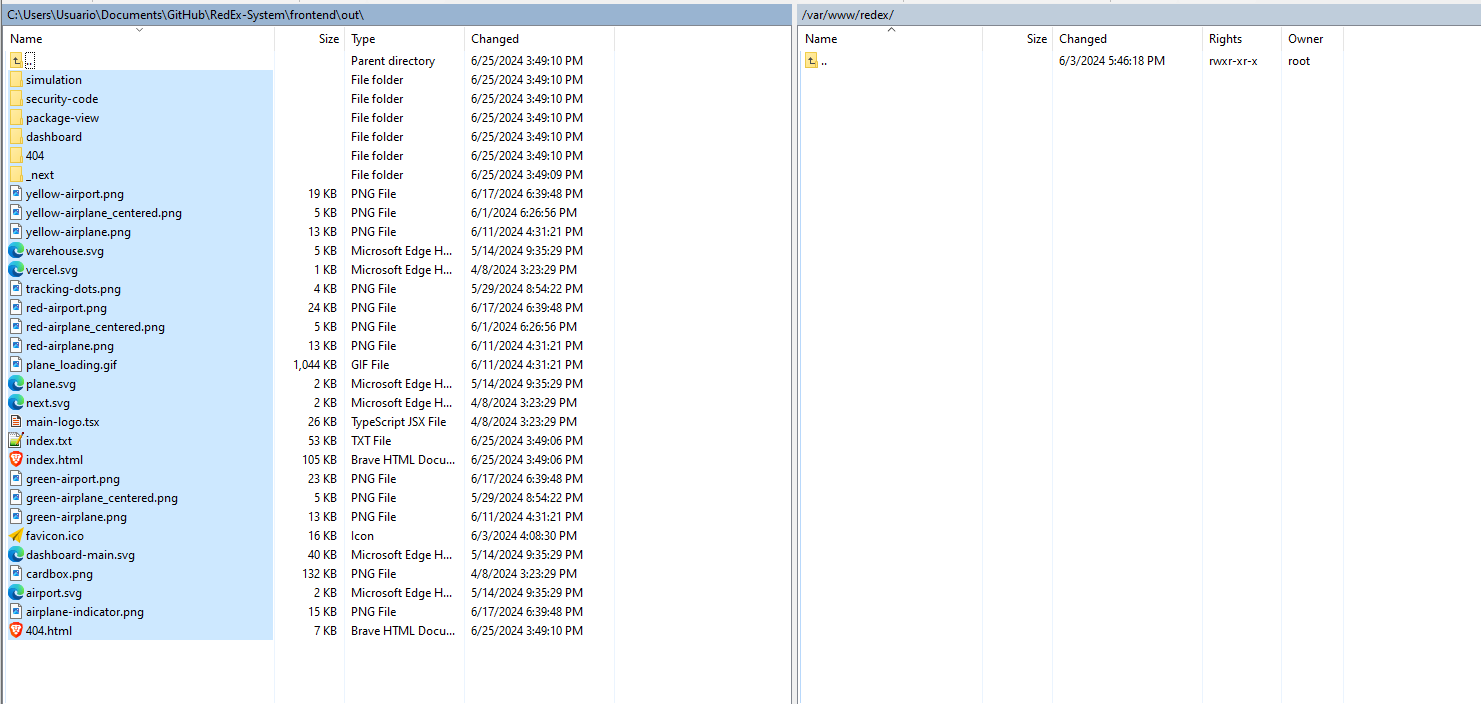
Para la transferencia de archivos, se utilizará el programa WinSCP. Primero se moverá el archivo compilado de backend, par ello es necesario el archivo correspondiente (en este caso, “redex\_back-0.0.1-SNAPSHOT.jar”) a /home/inf226.981.3c/app



Luego, se tendrá que renombrar el archivo a app.jar



De manera similar, se pasarán los archivos de \frontend\out a /var/www/redex. Se deberán arrastrar todos los archivos.



# Configuración de NGINX (Parte 1)

En la instancia, ejecutar  
  
$sudo nano /etc/nginx/sites-available/combined

Copiar y pegar el siguiente texto:

server {

listen 80;

server\_name inf226-981-3c.inf.pucp.edu.pe;

location / {

proxy\_pass http://localhost:8080;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

}

}

Luego, ejecutar:  
$sudo ln -s /etc/nginx/sites-available/combined /etc/nginx/sites-enabled/

Copiar y pegar el siguiente texto:  
[Unit]

Description=Spring Boot App

After=syslog.target

[Service]

User= inf226.981.3c

ExecStart=java -jar -Dspring.profiles.active=prod -Dserver.port=8080 /home/inf226.981.3c/app/app.jar

SuccessExitStatus=143

[Install]

WantedBy=multi-user.target

Luego, reiniciar la instancia:  
$sudo reboot now

# Instalación de Certbot

Ejecutar:

$sudo apt install certbot python3-certbot-nginx

$sudo certbot --nginx -d inf226-981-3c.inf.pucp.edu.pe

Luego, reiniciar la instancia y servicios:

$sudo reboot now

$sudo systemctl restart app.service

$sudo systemctl reload nginx

# Configuración de NGINX (Parte 2)

En la instancia, ejecutar

$sudo nano /etc/nginx/sites-available/combined

Se va a reemplazar el contenido del archivo, copiar y pegar el siguiente texto:

server {

listen 80;

listen [::]:80;

server\_name inf226-981-3c.inf.pucp.edu.pe www.inf226-981-3c.inf.pucp.edu.pe;

root /var/www/redex;

index index.html index.htm index.nginx-debian.html;

# Redirect HTTP to HTTPS

if ($host = inf226-981-3c.inf.pucp.edu.pe) {

return 301 https://$host$request\_uri;

}

location / {

try\_files $uri $uri/ /index.html;

}

# Backend proxy configuration for /back

location /back/ {

proxy\_pass http://localhost:8080;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

# WebSocket proxy configuration

location /websocket {

proxy\_pass http://localhost:8080;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_set\_header Host $host;

proxy\_buffering off;

proxy\_read\_timeout 1800;

proxy\_send\_timeout 1800;

}

}

server {

listen 443 ssl;

listen [::]:443 ssl;

server\_name inf226-981-3c.inf.pucp.edu.pe www.inf226-981-3c.inf.pucp.edu.pe;

root /var/www/redex;

index index.html index.htm index.nginx-debian.html;

ssl\_certificate /etc/letsencrypt/live/inf226-981-3c.inf.pucp.edu.pe/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/inf226-981-3c.inf.pucp.edu.pe/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

location / {

try\_files $uri $uri/ /index.html;

}

# Backend proxy configuration for /back

location /back/ {

proxy\_pass http://localhost:8080;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

# WebSocket proxy configuration

location /websocket {

proxy\_pass http://localhost:8080;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

proxy\_set\_header Host $host;

proxy\_buffering off;

proxy\_read\_timeout 1800;

proxy\_send\_timeout 1800;

}

error\_page 404 /404.html;

location = /404.html {

internal;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

internal;

}

}

Luego, ejecutar  
$sudo nano /etc/nginx/nginx.conf

Dentro del archivo, añadir las siguientes líneas dentro del bloque de “http {...}”

server\_names\_hash\_bucket\_size 64;

client\_header\_buffer\_size 64k;

large\_client\_header\_buffers 4 64k;

Finalmente, reiniciar la instancia y servicios:

$sudo reboot now

$sudo systemctl restart app.service

$sudo systemctl reload nginx

En caso el puerto 8080 se encuentre en uso, verificar los procesos que lo estén ocupando.

$sudo lsof -i :8080

$sudo kill <PID>

$sudo systemctl restart app.service

$sudo systemctl reload nginx

# Ingreso a la aplicación

Ingresar al siguiente link:

<https://inf226-981-3c.inf.pucp.edu.pe/>