

Tarea FTP - despliegue de aplicación en VPS

Trabajo realizado por Manuel Macho Calvín

Tarea FTP - despliegue de aplicación en VPS

Objetivo

Descripción de la tarea

VirtualBack

VirtualFRONT

Objetivo

Desplegar una aplicación web `Full Stack` en un **VPS**, siguiendo las indicaciones de la `playlist` facilitada en el aula virtual (https://www.youtube.com/watch?v=ibln4qlniv8&list=PL4bT56Uw3S4zIHpaNjz4OVAmXGkYgLpjo&ab_channel=LuigiCode). Como Servidor Privado Virtual (VPS) se utilizará el servidor remoto instalado en la nube privada del aula

Descripción de la tarea

VirtualBack

Lo primero que necesitamos es instalar una base de datos de `mysql`

```
sudo apt-get install mysql-server mysql-client
```

```
tutorial@tutorial-vps:~$ sudo apt-get install mysql-server mysql-client
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libtimedate-perl liburi-perl mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0
  mysql-common mysql-server-8.0 mysql-server-core-8.0
Paquetes sugeridos:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
Se instalarán los siguientes paquetes NUEVOS:
  libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libevent-core-2.1-7
  libevent-pthreads-2.1-7 libfcgi-perl libhtml-parser-perl libhtml-tagset-perl
  libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl
  liblwp-mediatypes-perl libmecab2 libtimedate-perl liburi-perl mecab-ipadic
  mecab-ipadic-utf8 mecab-utils mysql-client mysql-client-8.0
  mysql-client-core-8.0 mysql-common mysql-server mysql-server-8.0
  mysql-server-core-8.0
0 actualizados, 26 nuevos se instalarán, 0 para eliminar y 230 no actualizados.
Se necesita descargar 31,5 MB de archivos.
Se utilizarán 262 MB de espacio de disco adicional después de esta operación.
```

Una vez descargado todos los paquetes probamos a entrar como `root`

```
sudo mysql
```

```
tutorial@tutorial-vps:~$ mysql
ERROR 1045 (28000): Access denied for user 'tutorial'@'localhost' (using password: NO)
tutorial@tutorial-vps:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.28-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema      |
| mysql                   |
| performance_schema      |
| sys                     |
+-----+
4 rows in set (0,01 sec)

mysql>
```

Realizamos la instalación segura:

```
sudo mysql_secure_installation
```

A continuación hacemos cambios para poder entrar con una contraseña propia:

```
alter user 'root'@'localhost' identified with mysql_native_password by 'tutorial';
```

Seguimos para crear la base de datos con el nombre `'test_virtual'` coincidiendo con la que aparece en el proyecto `virtualBACK`:

```
mysql -u root -p
create database test_virtual;
show databases;
```

```
mysql> create database test_virtual;
Query OK, 1 row affected (0,01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test_virtual |
+-----+
5 rows in set (0,00 sec)
```

Creamos un usuario al que llamaremos `user` para poder tener acceso:

```
create user 'user'@'localhost' identified by 'user';
```

```
mysql> create database test_virtual;
Query OK, 1 row affected (0,01 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| test_virtual |
+-----+
5 rows in set (0,00 sec)
```

Y le damos privilegios a `user`:

```
grant all privileges on test_virtual.* to 'user'@'localhost' with grant option;
```

```
mysql> grant all privileges on test_virtual.* to 'user'@'localhost' with grant option;
Query OK, 0 rows affected (0,00 sec)
```

Seguidamente tenemos que mirar que versión de `java` y de `javac` tenemos:

```
java -version
javac -version
```

Al no tener versión de `jdk`, la instalaremos y seguidamente el `openjdk-8`:

```
sudo apt-get install maven
sudo apt-get install openjdk-8-jdk
```

```
tutorial@tutorial-vps:~$ sudo apt-get install maven
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
ca-certificates-java default-jre-headless java-common libaopalliance-java
libapache-pom-java libatinject-jsr330-api-java libcdi-api-java
libcommons-cli-java libcommons-io-java libcommons-lang2-java
```

```
tutorial@tutorial-vps:~$ sudo apt-get install openjdk-8-jdk
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

Comprobamos la version:

```
java -version
javac -version
```

```
tutorial@tutorial-vps:~$ java -version
openjdk version "11.0.13" 2021-10-19
OpenJDK Runtime Environment (build 11.0.13+8-Ubuntu-0ubuntu1.20.04)
OpenJDK 64-Bit Server VM (build 11.0.13+8-Ubuntu-0ubuntu1.20.04, mixed mode)
tutorial@tutorial-vps:~$ javac -version
javac 11.0.13
```

A continuación instalamos un servidor ftp, en nuestro caso utilizaremos **vsftpd** :

```
sudo apt-get install vsftpd
```

```
tutorial@tutorial-vps:~$ sudo apt-get install vsftpd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
```

Lo habilitamos para escritura:

```
sudo nano /etc/vsftpd.conf
```

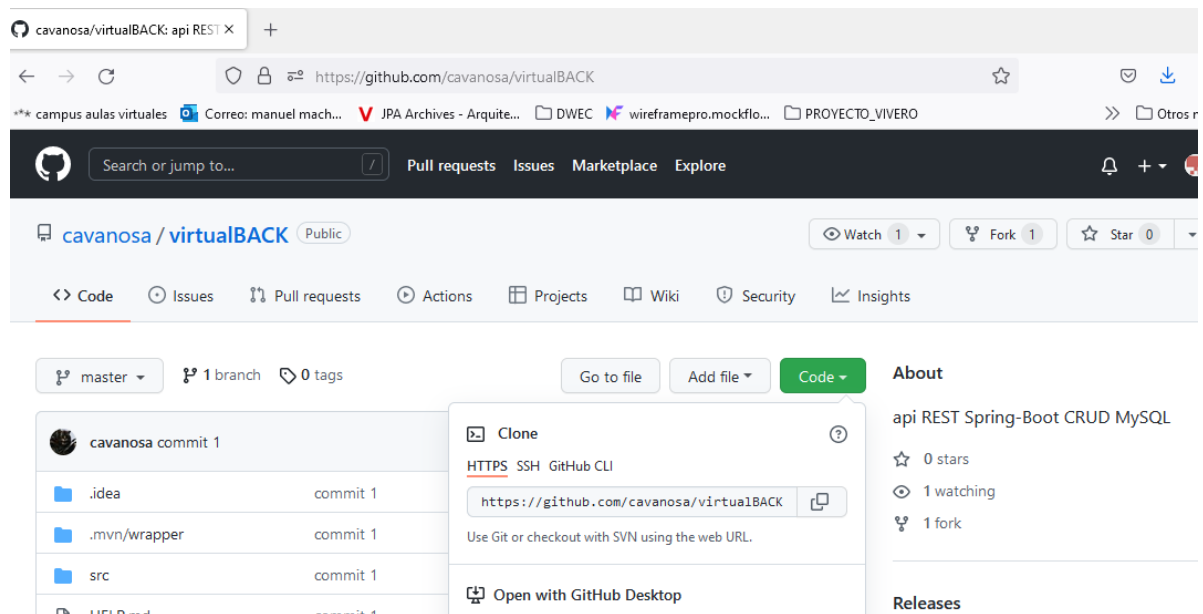
```
GNU nano 4.8 /etc/vsftpd.conf Modificado
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone? vsftpd can run either from an inetd or as a standalone
# daemon started from an init script.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
```

Reiniciamos el servidor `vsftpd` y comprobamos si está habilitado:

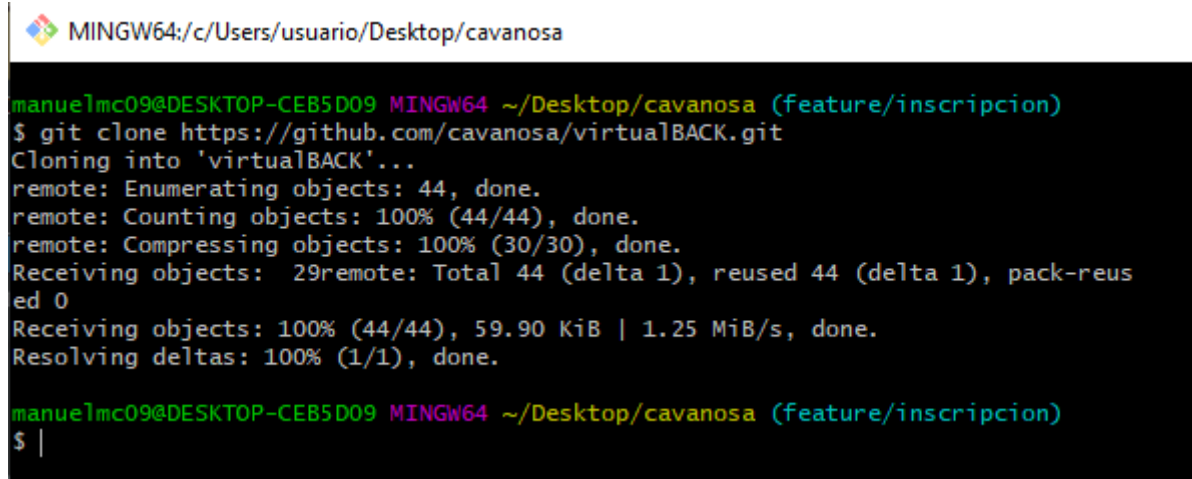
```
sudo systemctl restart vsftpd.service
sudo systemctl is-enabled vsftpd.service
```

```
tutorial@tutorial-vps:~$ sudo systemctl restart vsftpd.service
tutorial@tutorial-vps:~$ sudo systemctl is-enabled vsftpd.service
enabled
```

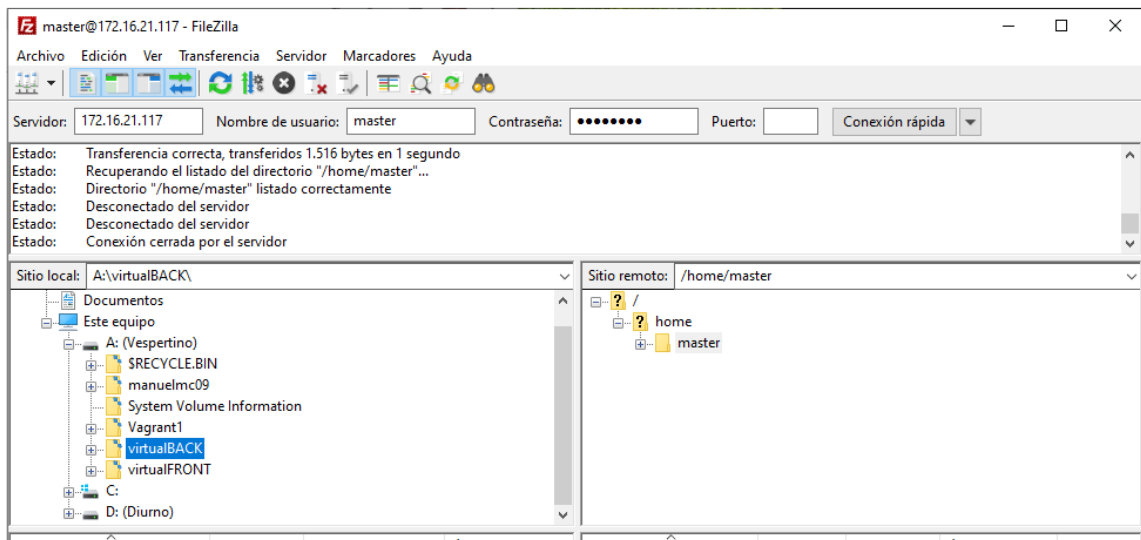
En este punto desde el repositorio que nos referencia clonamos el proyecto `virtualBACK` a una carpeta de nuestro escritorio.



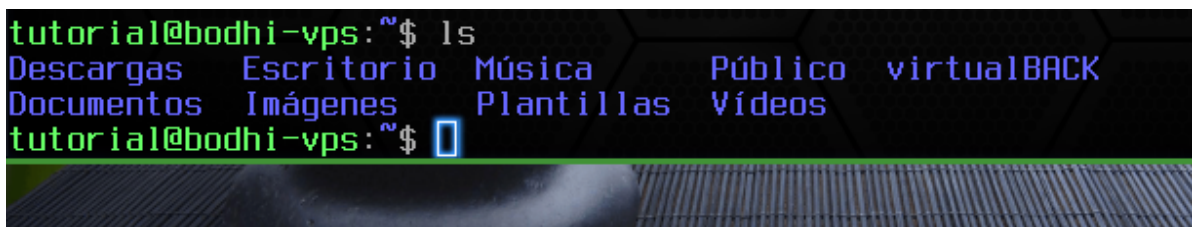
```
git clone https://github.com/cavanosa/virtualBACK.git
```



Una vez clonado el proyecto desde el **FileZilla** cliente subimos el proyecto al servidor ftp, a nuestra carpeta `/home/master`.

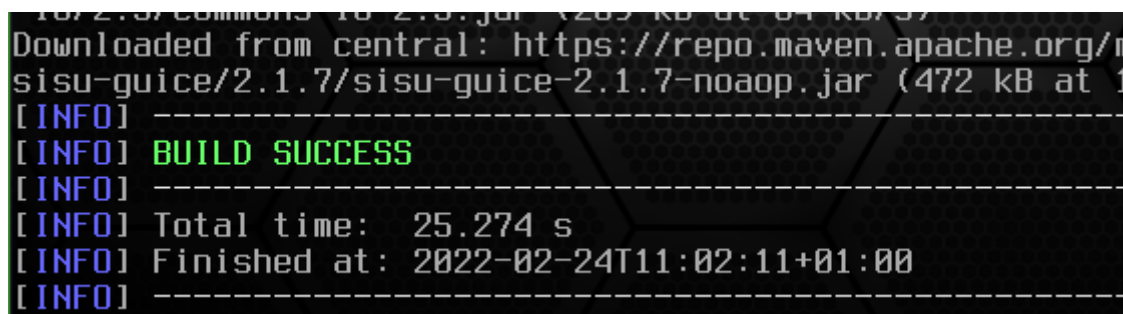
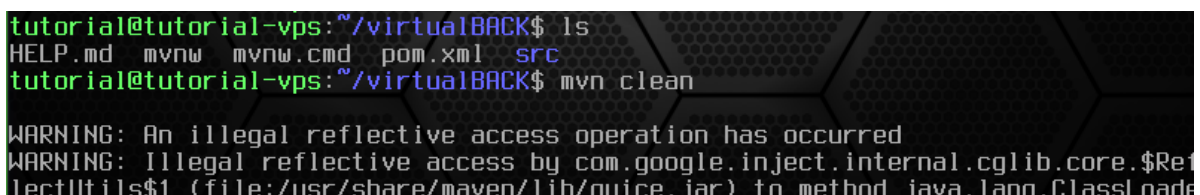


Comprobamos efectivamente que la transferencia mediante el ftp se ha realizado correctamente:



A continuación nos dirigimos a la carpeta `virtualBACK/` creada y ejecutamos para eliminar el target en el caso de que lo hubiera:

```
mvn clean
```



Editamos el pom.xml:

```
sudo nano pom.xml
```



```
<build>
  <plugins>
    <plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
      <configuration>
        <executable>>true</executable>
        <finalName>virtual</finalName>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Generamos el `.jar`

[illegible]

Ejecutamos el `.jar` creado:

```
java -jar target/virtual.jar
```

```
tutorial@bodhi-vps: ~/virtualBACK
(( ( ) \__ | ' _ | ' _ | ' _ \ / | \ \ \ \
 \ \ \ \ | | ) | | | | | | | | | | ) ) ) )
 ' _ | _ | _ | _ | _ | _ | _ | _ | _ | _ |
=====|_|=====|_|_/=/_/_/_/
:: Spring Boot :: (v2.2.1.RELEASE)

2022-02-15 21:08:34.245 INFO 7796 --- [          main] com.cavanosa.virtual.VirtualApplication : Starting VirtualApplication v0.0.1-SNAPSHOT on bodhi-vps with PID 7796 (/home/tutorial/virtualBACK/target/virtual.jar started by tutorial in /home/tutorial/virtualBACK)
2022-02-15 21:08:34.265 INFO 7796 --- [          main] com.cavanosa.virtual.VirtualApplication : No active profile set, falling back to default profiles: default
2022-02-15 21:08:38.590 INFO 7796 --- [          main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data repositories in DEFAULT mode.
2022-02-15 21:08:38.957 INFO 7796 --- [          main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 305ms. Found 1 repository interfaces.
2022-02-15 21:08:41.447 INFO 7796 --- [          main] trationDelegate$BeanPostProcessorChecker : Bean 'org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration' of type [org.springframework.transaction.annotation.ProxyTransactionManagementConfiguration] is not eligible for getting processed by all BeanPostProcessors (for example: not eligible for auto-proxying)
```

La aplicación se desplegará en el puerto `8080` . Podemos comprobar que se está ejecutando el

`spring`



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Wed Feb 23 15:41:08 UTC 2022

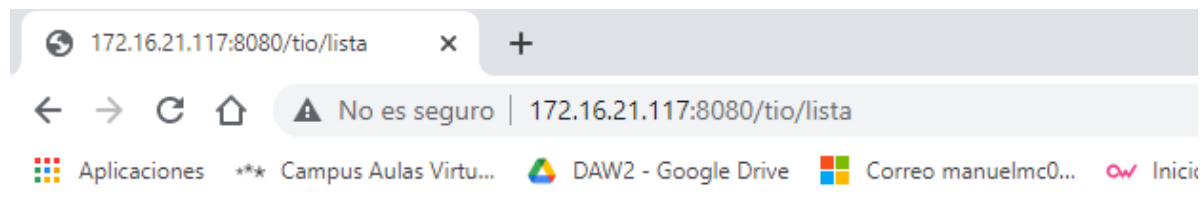
There was an unexpected error (type=Not Found, status=404).

No message available

```
master@daw-117: ~/virtualBACK
2022-02-23 15:38:15.215 INFO 16459 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.0.Final}
2022-02-23 15:38:15.371 INFO 16459 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-02-23 15:38:15.650 INFO 16459 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-02-23 15:38:15.660 INFO 16459 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL5InnoDBDialect
2022-02-23 15:38:16.811 INFO 16459 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
2022-02-23 15:38:16.817 INFO 16459 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2022-02-23 15:38:17.395 WARN 16459 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2022-02-23 15:38:17.586 INFO 16459 --- [main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2022-02-23 15:38:17.953 INFO 16459 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2022-02-23 15:38:17.954 INFO 16459 --- [main] com.cavanosa.virtual.VirtualApplication : Started VirtualApplication in 6.492 seconds (JVM running for 7.262)

2022-02-23 15:41:08.696 INFO 16459 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-02-23 15:41:08.696 INFO 16459 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2022-02-23 15:41:08.704 INFO 16459 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet : Completed initialization in 8 ms
```

Hacemos comprobaciones y miramos que nos devuelva la lista, en este caso al estar vacía, nos devolverá un array vacío:



[]


```

2022-02-23 15:41:08.696 INFO 16459 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
tion : Started VirtualApplication in 6.492 seconds (JVM running for 7.262)

2022-02-23 15:41:08.696 INFO 16459 --- [nio-8080-exec-1] o.a.c.c.C.[Tomcat].[localhost].[/]
: Initializing Spring DispatcherServlet 'dispatcherServlet'
2022-02-23 15:41:08.696 INFO 16459 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
: Initializing Servlet 'dispatcherServlet'
2022-02-23 15:41:08.704 INFO 16459 --- [nio-8080-exec-1] o.s.web.servlet.DispatcherServlet
: Completed initialization in 8 ms
Hibernate: select tio0_.id as id1_0_, tio0_.email as email2_0_, tio0_.nombre as nombre3_0_ fr
om tio tio0_

```

A continuación lo que haremos es convertirlo en ejecutable. Para que nada más arrancar el S.O. se ejecute el `.jar`

Para ello hay que crear el servicio y lo haremos creando un archivo para tal fin:

```
sudo nano /etc/systemd/system/spring.service
```

A continuación lo habilitamos:

```

master@daw-117:~/virtualBACK/target$ sudo systemctl enable spring.service
Created symlink /etc/systemd/system/multi-user.target.wants/spring.service -> /etc/systemd/sy
stem/spring.service.
master@daw-117:~/virtualBACK/target$ 

```

Lo iniciamos y comprobamos que efectivamente está activo

```
sudo systemctl is-enabled spring.service
sudo systemctl is-active spring.service
```

```

master@daw-117:~/virtualBACK/target$ sudo systemctl is-enable spring.service
Unknown operation is-enable.
master@daw-117:~/virtualBACK/target$ sudo systemctl is-enabled spring.service
enabled
master@daw-117:~/virtualBACK/target$ sudo systemctl is-active spring.service
inactive
master@daw-117:~/virtualBACK/target$ 

```

```

master@daw-117:~/virtualBACK/target$ sudo systemctl start spring.service
[sudo] password for master:
master@daw-117:~/virtualBACK/target$ sudo systemctl is-active spring.service
active
master@daw-117:~/virtualBACK/target$ 

```

Ya tenemos activa la aplicación. Con esto ya estaría configurado `spring` correctamente

Finalmente aplicamos la actualización correspondiente:

```
sudo apt-get update
```

```
master@daw-117:~/virtualBACK/target$ sudo apt-get update
Hit:1 http://archive.ubuntu.com/ubuntu bionic InRelease
Get:2 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Fetched 177 kB in 10s (17.1 kB/s)
Reading package lists... Done
master@daw-117:~/virtualBACK/target$ sudo apt-get apache2
E: Invalid operation apache2
master@daw-117:~/virtualBACK/target$ sudo apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
apache2 is already the newest version (2.4.29-1ubuntu4.21).
0 upgraded, 0 newly installed, 0 to remove and 201 not upgraded.
master@daw-117:~/virtualBACK/target$
```

VirtualFRONT

Primeramente instalaremos el Apache

```
sudo apt-get install apache2
```