

DESPLIEGUE DE APLICACIÓN EN SPRING BOOT



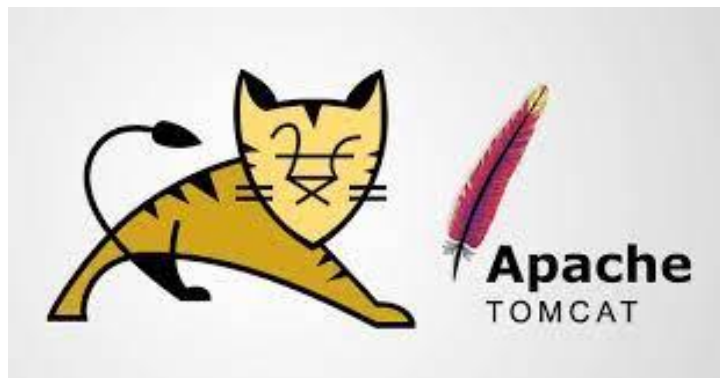
Víctor Márquez
Angel Pereira

Índice

Plataforma de despliegue	3
Ajustes en el proyecto de Spring	3
Generar archivo .war	7
Ajustes Tomcat	10
Despliegue de aplicación	11
Conexión desde otro dispositivo	14

Plataforma de despliegue

En nuestro caso hemos elegido desplegar la aplicación de forma local, subiendo la aplicación a un **servidor local**, en nuestro caso **Tomcat**.

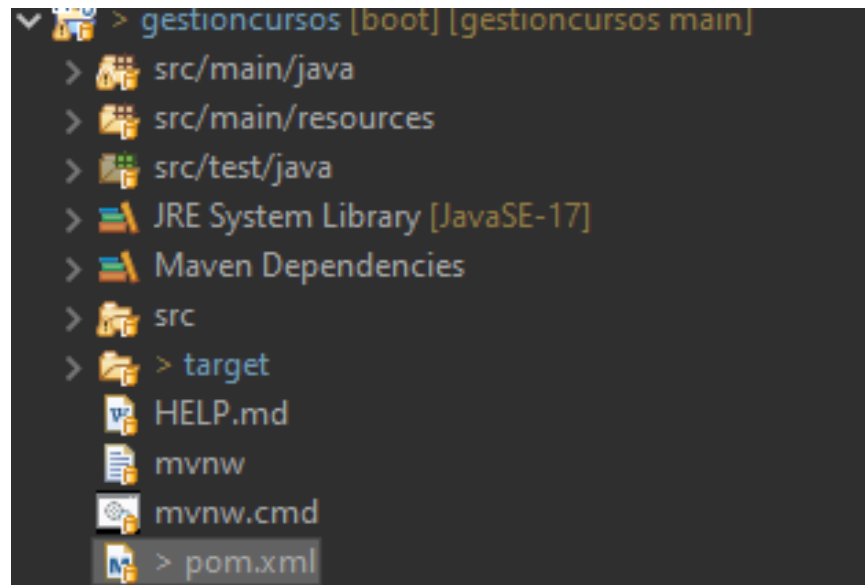


Ajustes en el proyecto de Spring

Para poder realizar nuestro despliegue de aplicación primero deberemos editar algunos **ajustes** en nuestro **proyecto de Spring**.

Primero accederemos a nuestro archivo **pom.xml**.

Despliegue de Aplicación en Spring boot



Una vez dentro debajo de la etiqueta version:

```
<parent>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-parent</artifactId>
  <version>2.7.5</version>
  <relativePath /> <!-- lookup parent from repository -->
</parent>

<groupId>com.example</groupId>
<artifactId>GestionCursos</artifactId>
<version>0.0.1</version>

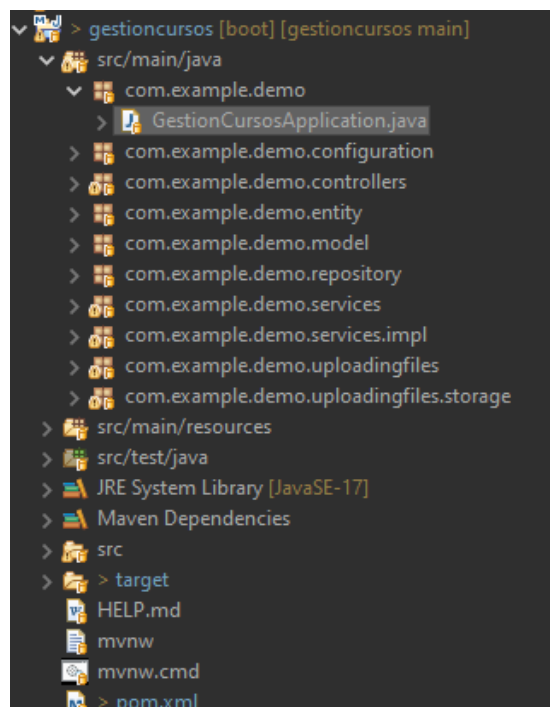
<name>GestionCursos</name>
<description>Mi primer proyecto Spring</description>
<properties>
  <java.version>17</java.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.thymeleaf.extras</groupId>
    <artifactId>thymeleaf-extras-springsecurity5</artifactId>
  </dependency>
  <dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-security</artifactId>
  </dependency>
  <dependency>
    <groupId>org.modelmapper</groupId>
    <artifactId>modelmapper</artifactId>
    <version>2.1.1</version>
  </dependency>
  <dependency>
    <groupId>org.webjars</groupId>
    <artifactId>bootstrap</artifactId>
    <version>4.5.3</version>
  </dependency>
</dependencies>
```

Aquí introducimos esta línea de código

```
<packaging>war</packaging>
```

para el siguiente cambio que debemos realizar en nuestro proyecto debemos acceder a nuestro archivo, en este caso **GestionCursosApplication**



Una vez en nuestro archivo **GestionCursosApplication** deberemos de extender nuestra clase de la clase **SpringBootTestInitializer** que nos permitirá poder usar nuestra aplicación en un servidor y añadimos el siguiente trozo de código:

```
@Override
public void onStartup(ServletContext servletContext) throws ServletException{
    super.onStartup(servletContext);
}
```

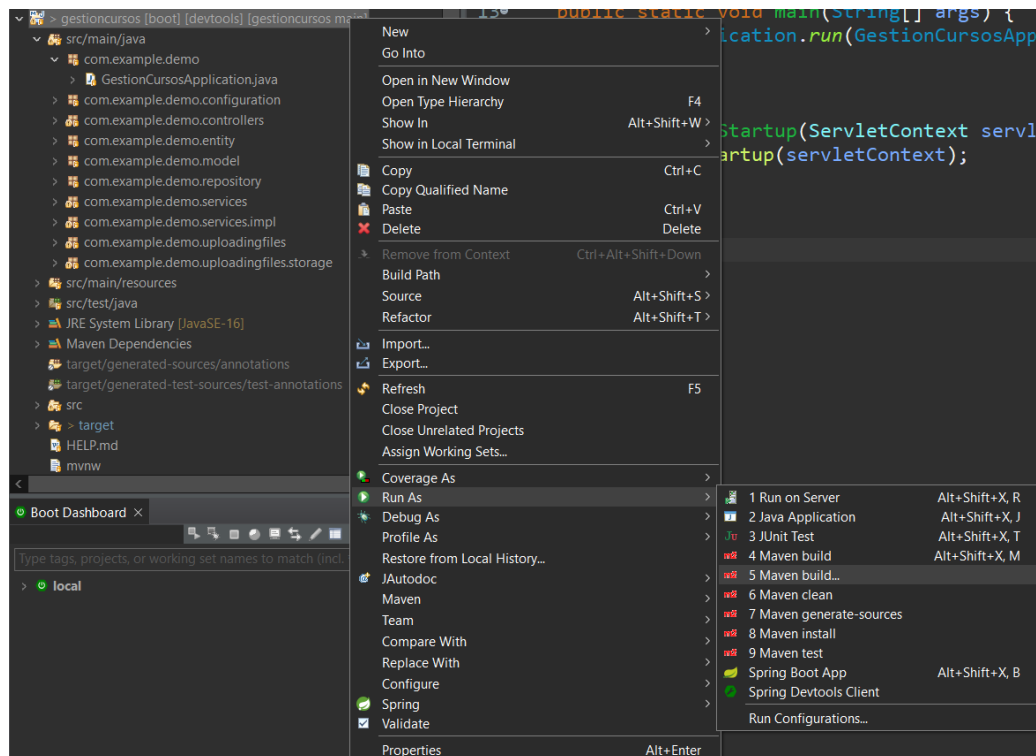
De la forma que nuestra clase quedará de la siguiente manera:

```
1 package com.example.demo;
2
3 import javax.servlet.ServletContext;
4 import javax.servlet.ServletException;
5
6 import org.springframework.boot.SpringApplication;
7 import org.springframework.boot.autoconfigure.SpringBootApplication;
8 import org.springframework.boot.web.servlet.support.SpringBootTestInitializer;
9
10 @SpringBootApplication
11 public class GestionCursosApplication extends SpringBootTestInitializer {
12
13     public static void main(String[] args) {
14         SpringApplication.run(GestionCursosApplication.class, args);
15     }
16
17     @Override
18     public void onStartup(ServletContext servletContext) throws ServletException{
19         super.onStartup(servletContext);
20     }
21
22 }
```

Generar archivo .war

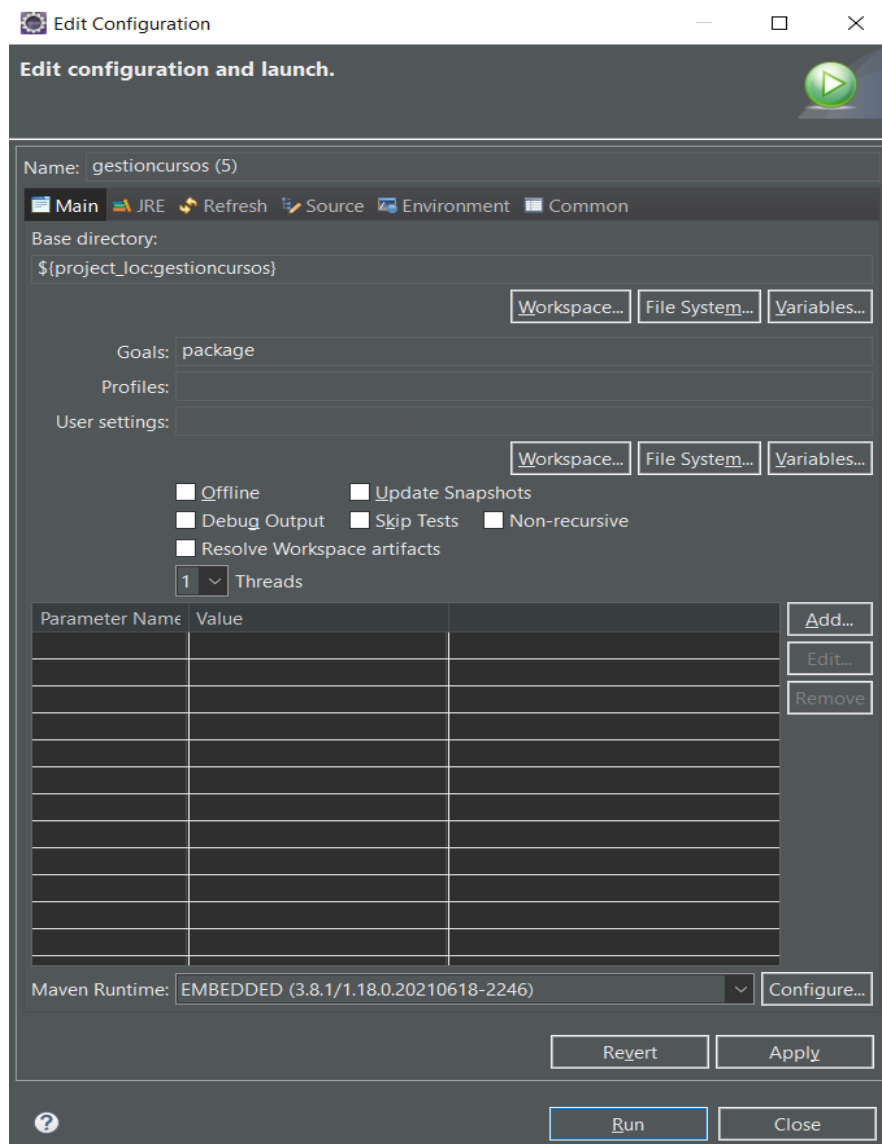
Para poder subir nuestro proyecto a un servidor local necesitaremos nuestro proyecto en un formato de **archivo .war** para ello seguiremos los siguientes pasos.

Accederemos desde eclipse a nuestro proyecto y con click derecho seleccionamos **run as y maven build**



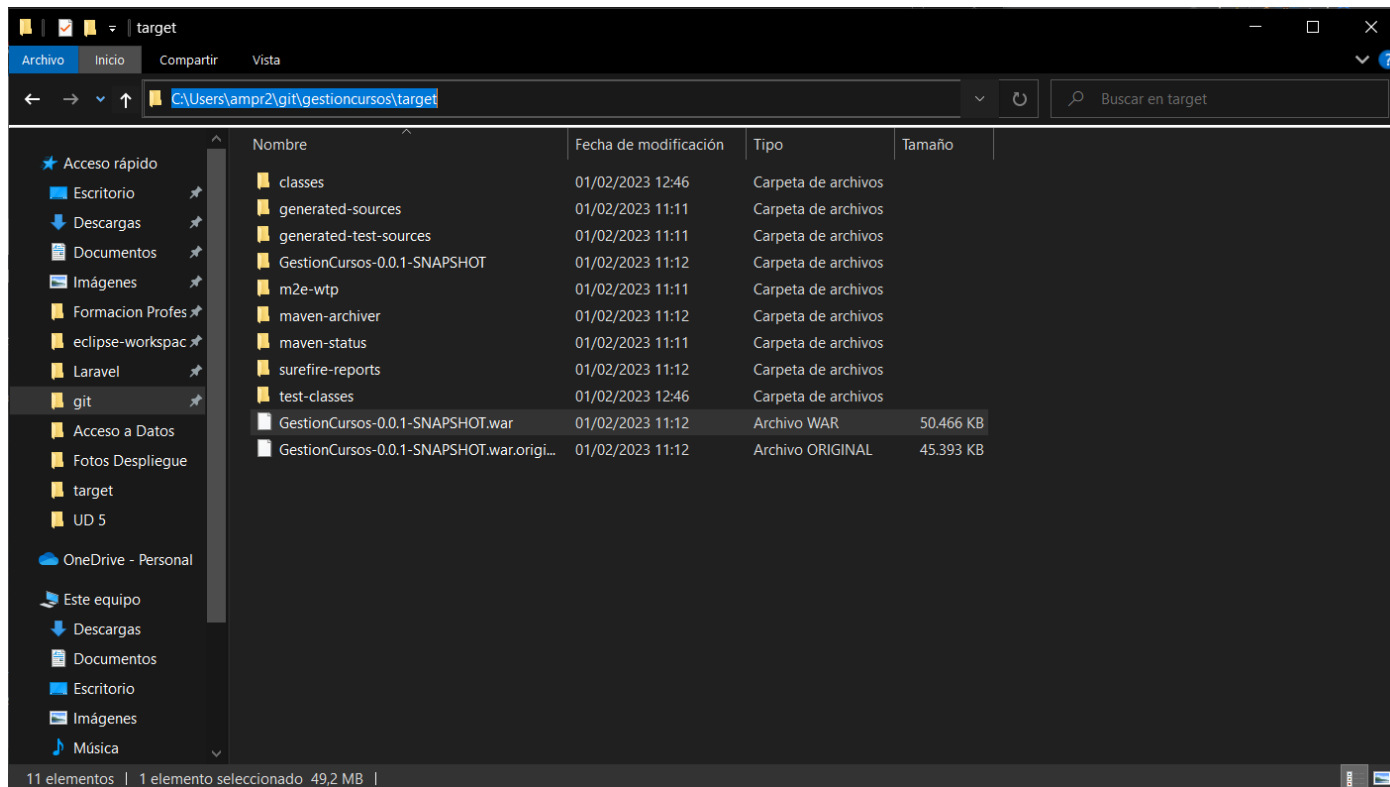
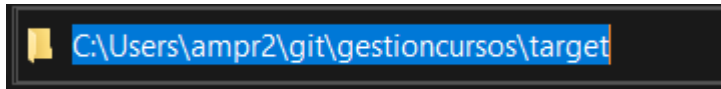
Despliegue de Aplicación en Spring boot

Seguidamente se nos abrirá la siguiente ventana y en el campo goals **escribiremos package** y pulsamos apply y run



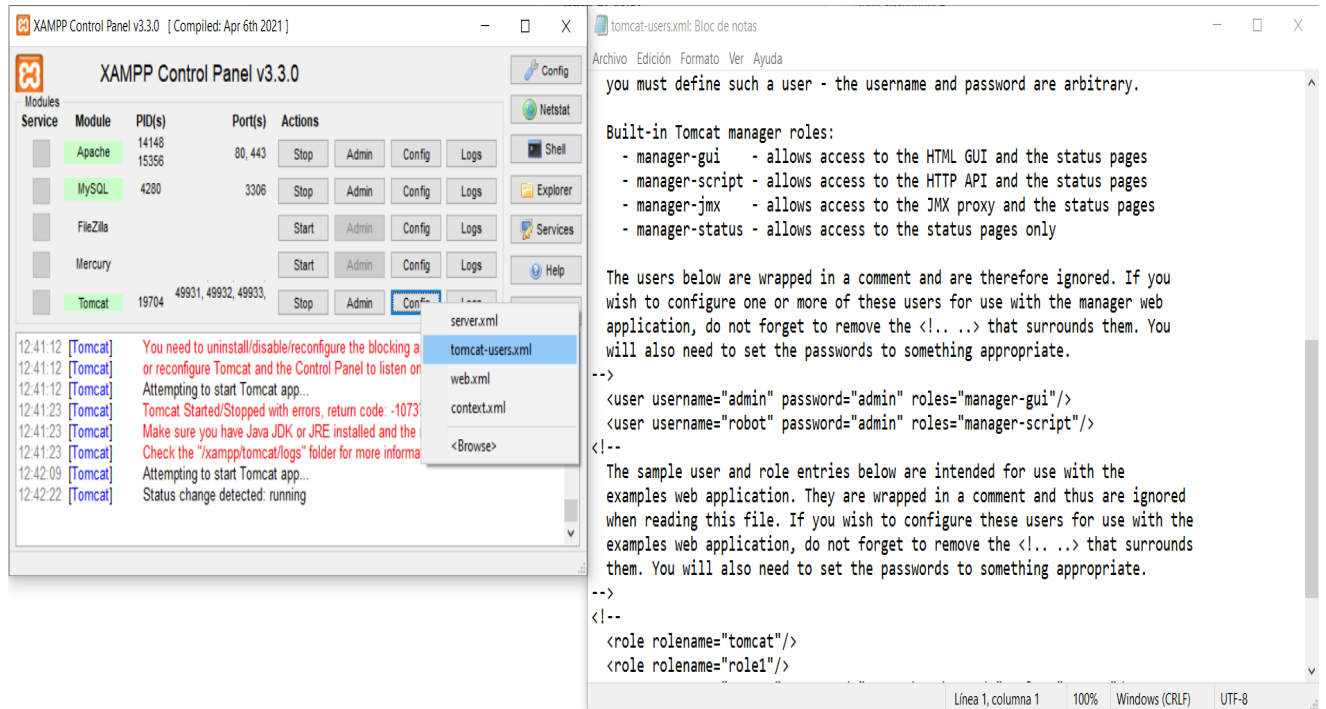
Despliegue de Aplicación en Spring boot

Una vez ejecutado el proyecto nos generará un archivo .war en la carpeta interna de nuestro proyecto, para verlo deberemos acceder desde el explorador de archivos en nuestro caso esta en la siguiente ruta:



Ajustes Tomcat

A Continuación debemos realizar un pequeño cambio en el tomcat, por lo cual **abrimos xampp** y accedemos al siguiente archivo:



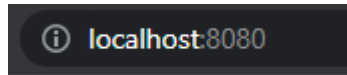
donde descomentamos y cambiamos estas dos líneas por las siguientes

```
34 <!--  
35 <user username="admin" password="<must-be-changed>" roles="manager-gui"/>  
36 <user username="robot" password="<must-be-changed>" roles="manager-script"/>  
37 -->
```

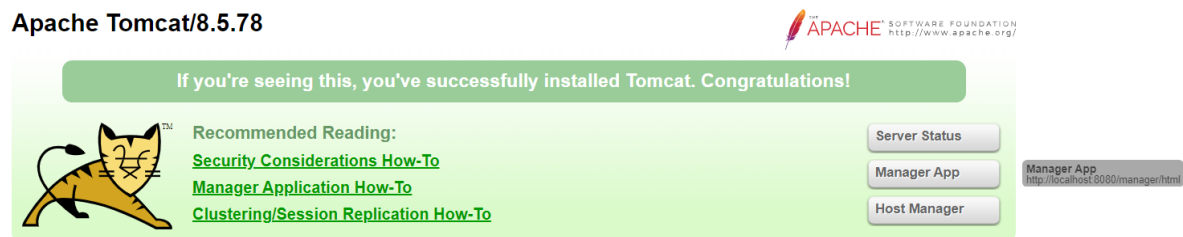
```
<user username="admin" password="admin" roles="manager-gui"/>  
<user username="robot" password="admin" roles="manager-script"/>
```

Despliegue de aplicación

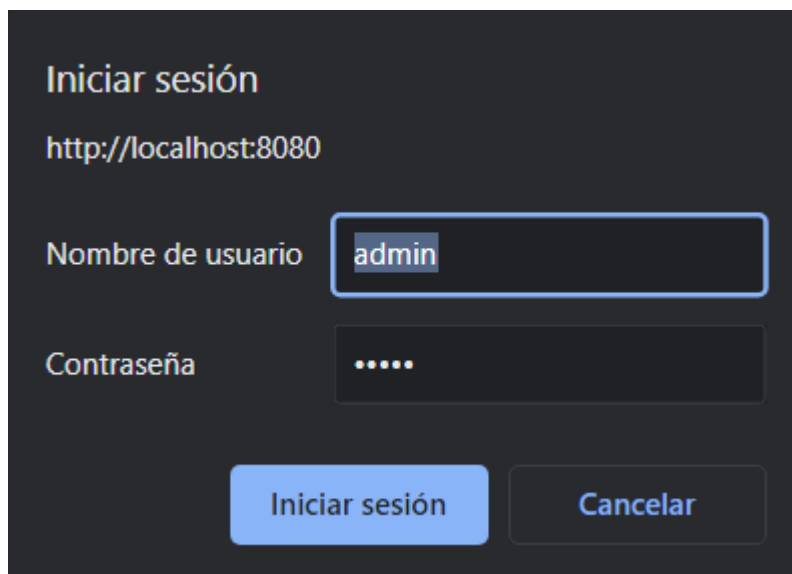
A continuación **iniciamos tomcat** y en nuestro navegador escribimos la siguiente ruta



esta ruta nos llevará la página siguiente, una vez en ella seleccionamos **manager app**



Nos pedirá **iniciar sesión**, las credenciales son las mismas escritas en nuestra configuración de tomcat que hemos editado anteriormente

A screenshot of the login form for the Tomcat Manager App. The title is 'Iniciar sesión'. Below it is the URL 'http://localhost:8080'. There are two input fields: 'Nombre de usuario' with the value 'admin' and 'Contraseña' with masked characters. At the bottom, there are two buttons: 'Iniciar sesión' and 'Cancelar'.

Despliegue de Aplicación en Spring boot

En la página de a continuación debemos de seleccionar nuestro **archivo .war** generado anteriormente, para ello pulsaremos en Seleccionar archivo

Desplegar

Desplegar directorio o archivo WAR localizado en servidor

Trayectoria de Contexto (opcional):

URL de archivo de Configuración XML:

URL de WAR o Directorio:

Desplegar

Archivo WAR a desplegar

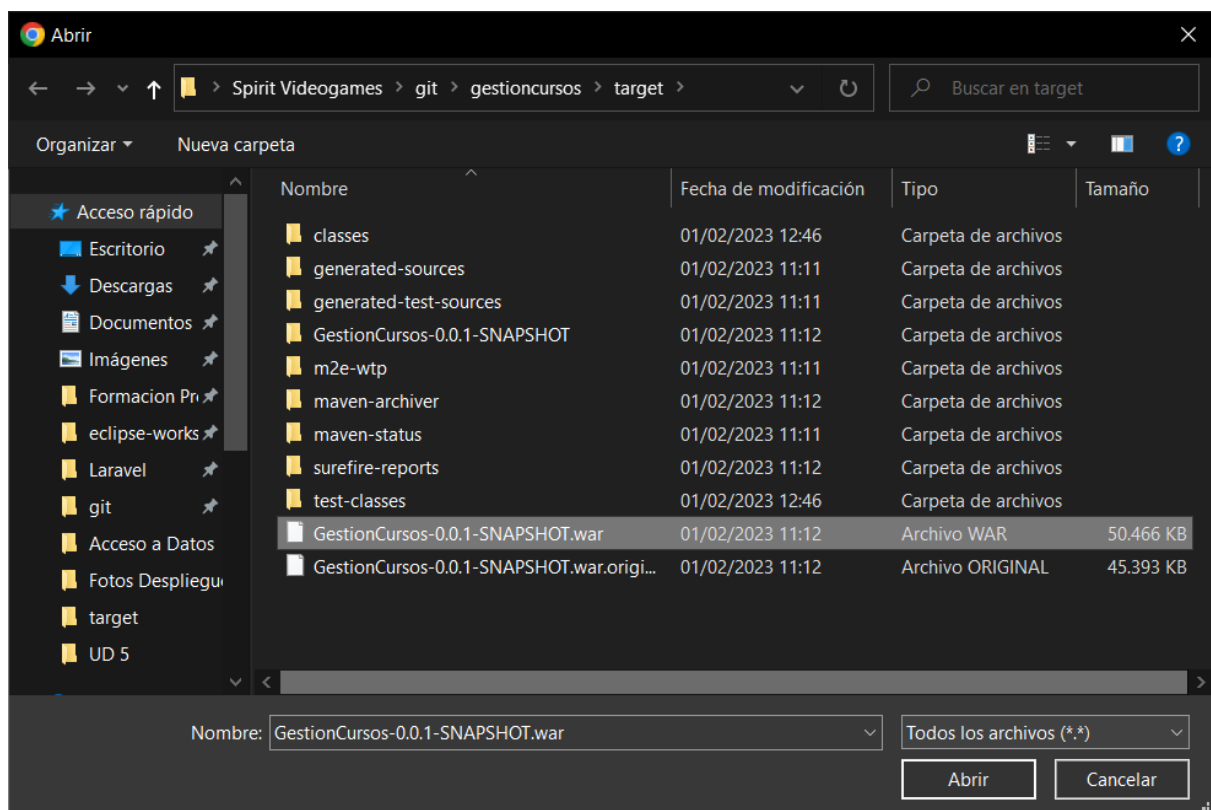
Seleccione archivo WAR a cargar

Seleccionar archivo

Ninguno archivo selec.

Desplegar

Buscamos en la ruta de nuestro proyecto que ya la vimos en pasos anteriores



Despliegue de Aplicación en Spring boot

Ahora podemos ver todas las rutas actuales en tomcat, para ver nuestro proyecto debemos de pulsar en **gestión cursos**

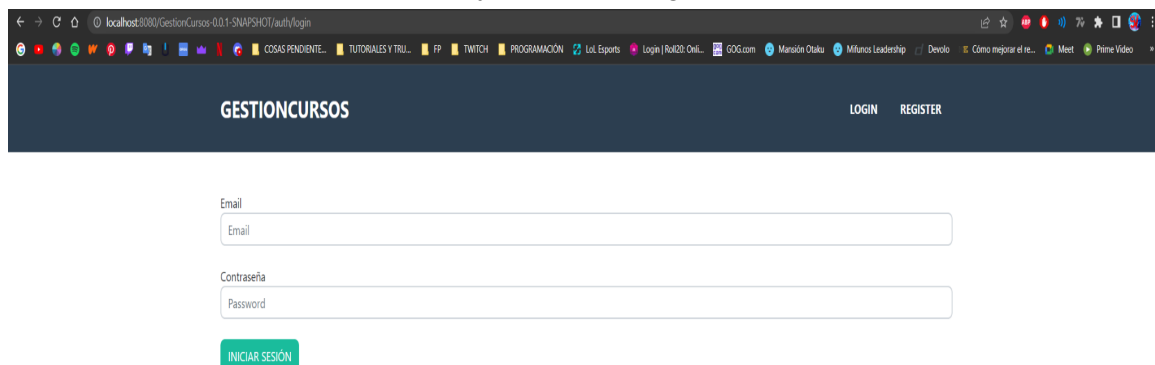
/GestionCursos-0.0.1-SNAPSHOT	Ninguno especificado
---	----------------------

Aplicaciones					
Ruta	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 2 30 minutos
/GestionCursos-0.0.1-SNAPSHOT	Ninguno especificado		true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 2 30 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 2 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 2 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 2 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar 2 30 minutos

Que nos mostrará la siguiente **ruta con un error 404** si seguida de ella ponemos **auth/login** nos llevará al login de nuestra aplicación

`localhost:8080/GestionCursos-0.0.1-SNAPSHOT/`

Aquí podemos ver que la aplicación ya está desplegada en un servidor local



localhost:8080/GestionCursos-0.0.1-SNAPSHOT/auth/login

GESTIONCURSOS LOGIN REGISTER

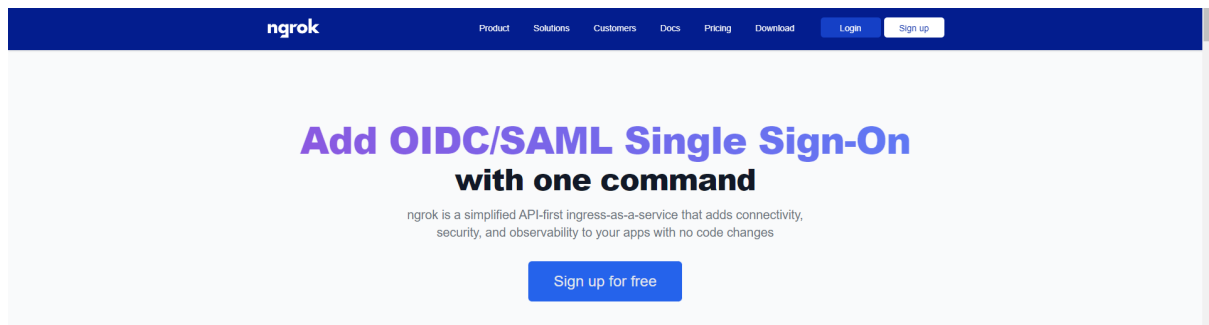
Email

Contraseña

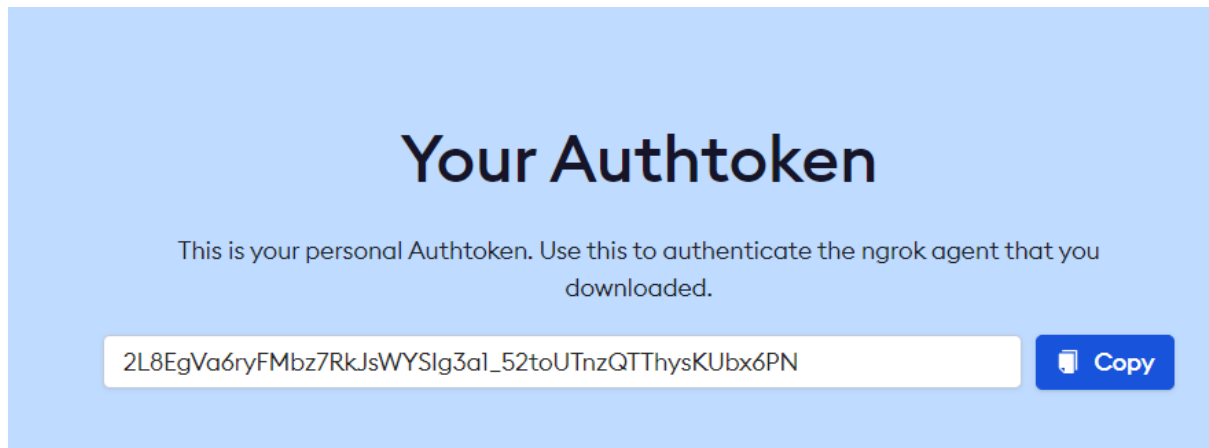
INICIAR SESIÓN

Conexión desde otro dispositivo

En nuestro caso hemos tenido problemas con el dispositivo que hostea el servidor ya que no conseguía abrir el puerto con el cual enlaza su conexión, para ello hemos encontrado la solución de descargarnos un **programa externo** que permite abrir el puerto. El programa se llama **Ngrok**



Cuando descarguemos el programa, obtendremos un **token de autenticación**



Una vez tengamos el programa lo ejecutamos y nos abrirá una consola de comandos donde insertamos la siguiente línea

```
C:\Users\ampr2\Downloads>ngrok config add-authtoken 2L8EgVa6ryFMbz7RkJsWYSIg3a1_52toUTnzQTThysKUbx6PN
```

Despliegue de Aplicación en Spring boot

La cual se forma de **ngrok config add-authtoken** y seguidamente el token obtenido

```
The ngrok agent gets you online in one line
ngrok tls --hostname=foo.com 443 # TLS traffic for foo.com to port 443
ngrok start foo bar baz # start tunnels from the configuration file

COMMANDS:
api          use ngrok agent as an api client
completion  generates shell completion code for bash or zsh
config       update or migrate ngrok's configuration file
credits      prints author and licensing information
diagnose     diagnose connection issues
help         Help about any command
http        start an HTTP tunnel
service      run and control an ngrok service on a target operating system
start        start tunnels by name from the configuration file
tcp         start a TCP tunnel
tls          start a TLS tunnel
tunnel       start a tunnel for use with a tunnel-group backend
update       update ngrok to the latest version
version      print the version string

OPTIONS:
--config strings  path to config files; they are merged if multiple
-h, --help        help for ngrok
-v, --version     version for ngrok

ngrok is a command line application, try typing 'ngrok.exe http 80'
at this terminal prompt to expose port 80.
C:\Users\ampr2\Downloads>ngrok config add-authtoken 2L8EgVa6ryFMbz7RkJswYSIg3a1_52toUTnzQTThysKUbx6PN
Authtoken saved to configuration file: C:\Users\ampr2\AppData\Local\ngrok\ngrok.yml
C:\Users\ampr2\Downloads>
```

Una vez hecho esto habremos iniciado sesión en la aplicación y para poder **abrir el puerto** y comenzar a hostear insertamos la siguiente linea de comando

```
C:\Users\ampr2\Downloads>ngrok http 8080
```

Y al realizarlo aparecerá lo siguiente

```
ngrok
Try our new native Go library: https://github.com/ngrok/ngrok-go

Session Status      online
Account             ampr2003@gmail.com (Plan: Free)
Version             3.1.1
Region              Europe (eu)
Latency              -
Web Interface       http://127.0.0.1:4040
Forwarding           https://2046-84-78-244-82.eu.ngrok.io -> http://localhost:8080

Connections         ttl    opn    rt1    rt5    p50    p90
                   0      0      0.00   0.00   0.00   0.00
```

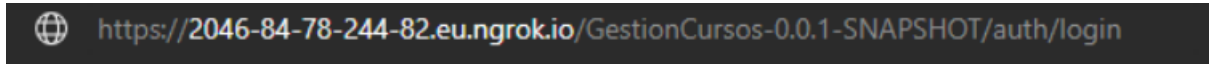
Esto significa que se ha conseguido abrir el puerto y nos generará una url que sustituirá a <http://localhost:8080>

```
https://2046-84-78-244-82.eu.ngrok.io -> http://localhost:8080
```

Despliegue de Aplicación en Spring boot

Una vez hecho esto podremos proceder a probar a **conectarnos desde otro dispositivo**, en nuestro caso hemos utilizado una máquina virtual

Por lo cual vamos a nuestro buscador e introducimos la url generada



Y como podemos ver nos podemos conectar adecuadamente a nuestra aplicación web

