



Playit

Documentación del proyecto

Proyecto Desarrollo de
Aplicaciones Web

*David César Fernández-
Aliseda*

Índice

| | |
|---|----|
| 1. Definición del proyecto..... | 1 |
| 1.1 Propuesta del proyecto..... | 1 |
| 1.2 Descripción detallada del proyecto..... | 1 |
| 1.3 Ámbito del proyecto..... | 4 |
| 1.4 ¿Adaptación o creación?..... | 4 |
| 1.5 Conceptos básicos del software..... | 4 |
| 2. Planificación del proyecto web..... | 5 |
| 2.1 Definición de tareas..... | 5 |
| 3. Análisis del proyecto web..... | 6 |
| 3.1 Requisitos técnicos web..... | 6 |
| 3.2 Requisitos técnicos web..... | 7 |
| 4. Ingeniería del proyecto web..... | 7 |
| 4.1 Diseño del software..... | 7 |
| 4.2 Diseño de datos..... | 8 |
| 4.3 Diseño del sistema..... | 10 |
| 4.4 Diseño de la interfaz del usuario..... | 10 |
| 5. Despliegue..... | 11 |
| 6. Posibles errores..... | 13 |

1. Definición del proyecto.

1.1 Propuesta del proyecto.

La idea inicial surge debido a la existencia de una multitud de plataformas de streaming que provocan que el contenido audiovisual se encuentre dividido, dificultando el acceso y consumo de este contenido. Además, a esto se le añade el problema de que el contenido puede cambiar de plataforma, pudiendo perder así la noción de si el usuario ha visto o no una película, o por cual capítulo de su serie favorita iba.

Una vez definida la idea principal, y el problema que se quería resolver, se contemplaron las posibles tecnologías para su desarrollo. Finalmente, se decidió optar por las tecnologías empleadas durante el curso.

1.2 Descripción detallada del proyecto.

Playit, es una aplicación web donde poder consultar información sobre películas y series, actores y directores que participan en ella. Además de las plataformas de streaming en las que se pueden visionar (Netflix, HBO, PrimeVideo, etc.), el número de temporadas o ver el tráiler de dicha película o serie. También se podrá llevar un registro sobre todo lo que el usuario ve, como por ejemplo, por qué capítulo va en una serie o las películas que ha visto y se podrá añadir películas y series a unas listas predefinidas: vistas, pendientes, favoritas.

Existirán dos tipos de usuarios, el usuario administrador y el usuario estándar:

- El usuario administrador podrá crear las fichas de todo el contenido existente (películas y series), así como editarlo y eliminarlo. También podrá editar y eliminar la información del resto de usuarios. Este tipo de usuarios tendrá acceso al mismo contenido que el usuario estándar excepto a los diferentes tipos de listas que se comentarán a continuación.
- El usuario estándar podrá acceder a todas las películas y series, así como a su información. Además, podrá añadir este contenido a diferentes listas, las cuales son:
 - Para películas: pendientes, favoritas y vistas.
 - Para series: pendientes, favoritas, vistas y siguiendo.

Este tipo de usuario no podrá editar ni crear ningún tipo de información.

Ambos usuarios necesitarán crear una cuenta, que contará de los siguientes datos: nombre, apellidos, nombre de usuario, contraseña y e-mail. Tanto el e-mail como el nombre de usuario serán únicos, es decir, no existirán dos usuarios con estos datos repetidos.

La creación del usuario administrador podrá hacerla otro administrador, como también podrá dar permisos de administrador a otro usuario ya existente.

En cuanto a la creación del contenido, a la que, como se ha mencionado anteriormente, solo tendrá acceso el usuario administrador, será independiente para películas y capítulos. El contenido inicial en ambas es el mismo: título, año de estreno, duración, tráiler, país de origen, sinopsis y ruta de la portada. Además, para películas añadiremos la valoración y para series el número de temporadas que la forman.

Para series tendremos una particularidad. Una vez que la tengamos creada podremos añadir capítulos, los cuales tendremos que crear con anterioridad. Estos constarán de: nombre y número del capítulo y número de temporada en el que se encuentra.

Existirán dos formas para crear las fichas de películas y series:

- A mano: es decir toda la información la introducirá el administrador.
- Con la ayuda de APIs: el administrador obtendrá una serie de información, la cual podrá editar antes de cargar en la base de datos.

Contaremos también con otros tres campos, con los cuales el administrador podrá crear y modificar contenido:

- Plataforma: plataforma de streaming donde se encuentra el contenido, de la cual guardaremos el nombre, logo y el enlace de la película o serie.
- Género: el género al que pertenece el contenido, este constará únicamente del nombre del género.
- Persona: este campo podrá ser de tres tipos: actor, director y ambos. En cualquier caso, se compone de nombre, primer y segundo apellido y fecha de nacimiento.

1.3 Ámbito del proyecto.

El proyecto busca la creación de una aplicación que sirva para centralizar y facilitar la obtención de información relativa a las películas y series, de forma que se comprima información que de otra forma se encuentra completamente dispersa por diferentes páginas webs y plataformas, simplificando la forma en la que un usuario pueda acceder a su contenido.

Por tanto, nuestro ámbito se puede asemejar a los que realizan plataformas de streaming, centrándonos en lo que se refiere a contener todas las películas y series en un mismo lugar.

1.4 ¿Adaptación o creación?

El proyecto es una adaptación que funciona como puente entre plataformas de streaming y aplicaciones que antes de la existencia de estas plataformas eran muy utilizadas para tener acceso a este contenido, aunque de forma gratuita. La principal diferencia con estas aplicaciones como *Megadede* o *Series.ly* es que en *Playit* no puedes visualizar el contenido. Y la diferencia con respecto a las plataformas es que no contamos con la limitación de las licencias de series y películas ya que no son accesibles en la propia página para el usuario.

1.5 Conceptos básicos del proyecto software.

Para la correcta comprensión del documento conviene conocer ciertos conceptos básicos que se explican a continuación.

- Base de datos Relacional: es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- ORM (Mapeo Objeto-Relacional): Es una técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.
- Framework: Es una estructura conceptual y tecnológica de soporte definido, normalmente con módulos de software concretos, que pueden servir de base para la organización y desarrollo de software.
- Spring Framework: se trata de un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.

2. Planificación del proyecto web.

2.1 Definición de tareas.

Motivación Inicial: la necesidad de crear un proyecto provoca la idea inicial que implicó una búsqueda de una opción de creación de una aplicación viable.

Planificación Inicial: proceso en el que se llevó a cabo una base sobre la que cimentar los primeros pasos del desarrollo del aplicativo. Estos pasos son los siguientes:

- Descripción del proyecto. Se trata de la primera descripción de la idea inicial y sobre la cual se desarrolla la aplicación.
- Alcance del proyecto. Se fija el ámbito sobre el cual se va a trabajar.
- Stack tecnológico. Basándonos en la situación actual de los lenguajes/frameworks del mercado se fijaron los lenguajes a emplear en el proyecto.
- Especificación de requisitos. Se crean los requisitos que va a tener que cumplir el proyecto, tanto los funcionales como los no funcionales.
- Mapa del sitio. Se realiza una descripción de las pantallas necesarias, así como el flujo existente entre estas pantallas.
- Modelado de datos. Creación del modelo inicial, empleando una base de datos relacional, fijando las tablas con sus correspondientes campos y las relaciones necesarias que se establecen entre las distintas tablas.
- Casos de uso. Se crea un documento registrando los casos de usos más relevantes de la aplicación, fijando los diagramas pertinentes con los actores y el proceso de los casos tratados.
- Diagramas de actividad. Con los casos de uso establecidos, se realizan los diagramas de actividad los cuales desarrollan los distintos usos de forma más detallada.
- Realización de maqueta. Empleando la aplicación web *Figma* se realiza un MockUp con el aspecto que se desea que tenga la aplicación.

Fase de desarrollo del aplicativo.

- Realización del prototipo. Basándonos en el MockUp, se realiza el prototipo empleando *HTML*, *CSS* y el framework *Materialize* estableciendo la base estética del proyecto y la navegación entre las distintas interfaces de la aplicación.
- Estructura inicial de Back-end. Realización de una estructura inicial empleando Spring.
 - Creación del modelo de datos empleando un ORM.
 - Creación de los repositorios necesarios empleando Spring.
 - Creación de los servicios necesarios para comenzar la aplicación.

- Creación de los controladores que emplean estos servicios y mapean el acceso que se realizará desde el Front-End.
- Implementación de la seguridad en la aplicación.
 - Empleo de Spring Security.
- Estructura inicial de Front-end. Se realiza la primera aproximación a la tecnología elegida para el Front.
 - Creación de las vistas, sin lógica aún, que se desarrollaran.
 - Creación de los distintos servicios, sin lógica aún.
 - Creación de los distintos componentes
- Creación de datos en la base de datos relacional empleada. Se crean datos necesarios para llevar a cabo las pruebas en Front-End.
- Se comienza la realización de la codificación necesaria para integrar front y back.
 - Realización de los servicios que se consumirán.
 - Construcción de las distintas vistas empleando los servicios.
 - Adición de comprobaciones en formularios.
 - Construcción de diversos componentes empleando Javascript.
 - Implementación de la navegabilidad entre las distintas vistas.
- Pruebas manuales de la aplicación. Revisiones manuales de las distintas funcionalidades del aplicativo, detectando bugs y corrigiéndolos.
- Creación de la documentación necesaria referente al desarrollo de la aplicación, a la instalación/despliegue y a los manuales de uso.

3. Análisis del proyecto web.

3.1 Requisitos técnicos web.

Desarrollo de Back-End.

Para el desarrollo de la parte del back-end se empleará como entorno de desarrollo *Eclipse* y el lenguaje empleado será *Java*.

Se empleará el uso del framework *Spring* por medio de *Spring Tool Suite*. Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control de código abierto para la plataforma Java. Spring Tool Suite es un IDE basado en la versión Java EE de Eclipse pero customizado para trabajar con Spring Framework.

Desarrollo de Front-End.

En primer lugar, los diseños serán realizado mediante *Figma*, el cual es un editor de gráficos vectorial y una herramienta de generación de prototipos.

El desarrollo se realizara empleando como entorno *Visual Studio Code*. En el diseño se empleará *HTML* y *CSS* además de *Materialize*. Materialize es una herramienta de código abierto para el diseño de aplicaciones y sitios web.

También se usará *Javascript* para el diseño de algunas funcionalidades, así como para el uso de determinadas APIs.

Base de datos

Se empleará una base de datos relacional SQL, usando como gestor de datos *MySQL* y *MySQL Workbench* como herramienta visual de base de datos.

Gestor de versiones

El control de versiones se hará con el uso de *Git* mediante *GitHub Desktop*.

Las pruebas se han realizado de forma manual mediante el navegador Mozilla Firefox.

3.2 Diseño gráfico.

Para el diseño, se ha buscado que sean interfaces claras e intuitivas, se han empleado el contraste de colores claros sobre fondos oscuros, evitando así que el fondo deslumbre al usuario.

Los colores adoptados son el negro, el blanco y el azul, empleados tanto en las distintas páginas de la aplicación como en el logotipo realizado, son colores que transmiten limpieza, paz en el caso del blanco y cielo, agua o tranquilidad en el caso del azul.

Tanto el tipo de letra como el tamaño de la letra buscan que sea de fácil y cómoda lectura, en el caso del tamaño de la letra, varía en función de la resolución de la pantalla, permitiendo lecturas en móviles y tablets.

Para la fuente de la letra se ha empleado Google Font.

4. Ingeniería del proyecto web.

4.1 Diseño del software.

Para el diseño del aplicativo se ha escogido una arquitectura de tres capas (presentación, aplicación y persistencia). La utilización de esta arquitectura se debe a que los distintos niveles son independientes unos de otros de manera que se puede cambiar fácilmente el comportamiento de las clases en el nivel de la aplicación sin que ello influya en las otras capas. En los siguientes apartados vamos a explicar las distintas capas del software.

4.2 Diseño de datos.

En este apartado hablamos de la capa de persistencia, que se corresponde con la base de datos de la aplicación y las distintas tablas que la conforman.

Estas tablas son:

- Usuario
- Genre
- Platform
- Person
- Content
- Film
- Serie
- Chapter

A continuación, se muestra el diagrama *Entidad-Relación* utilizado para la implementación de la base de datos del aplicativo.

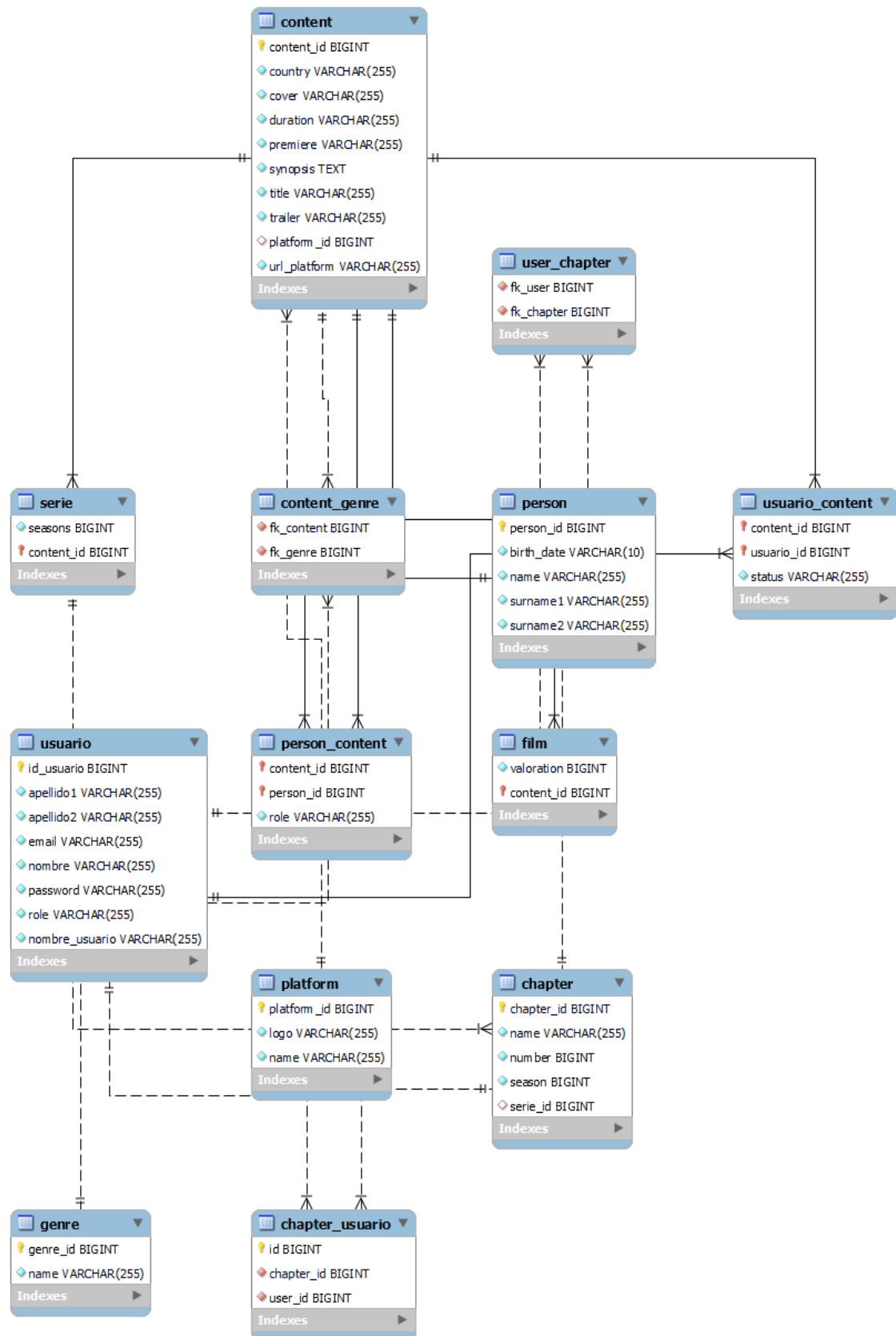


Figura 1. Diagrama Entidad-Relación

4.3 Diseño del sistema.

En este apartado nos centramos en la capa de lógica de la aplicación, es decir, el conjunto de componentes que implementa la funcionalidad de la aplicación web.

Esta capa sirve de enlace entre los niveles de presentación y de persistencia, ya que la capa de presentación no accede a la base de datos directamente, si no que se comunica con la capa de la aplicación para demandarle el servicio deseado y es la capa que se comunica con la capa de persistencia para recuperar los datos necesarios.

El diseño se ha realizado empleando el lenguaje Java y un framework de Java (Spring) en Spring Tool Suite una modificación de Eclipse IDE diseñada para trabajar con Spring.

Nuestra capa de presentación presenta la siguiente paquetería.

- Controller. Gestiona todos los accesos a los servicios desde la capa de presentación.
- DTO. Sirven para crear objetos planos que contienen información de múltiples fuentes que se concentran en una sola clase para evitar que sean necesarias varias invocaciones.
- Models. Recoge los modelos de las distintas clases que se recogen en el proyecto.
- Repository. Nos lo provee Spring y nos permite acceder a servicios de forma más cómoda.
- Service. Sirve como intermediario entre los repositorios y los controladores.
- Utils. Paquete que almacena las clases que proporciona métodos útiles pero que no pertenecen a la aplicación como tal.
- Carpeta Resources. Almacena los datos de configuración de la aplicación, de la base de datos y los archivos necesarios para la interfaz de usuario (HTML, CSS, JS, img...).

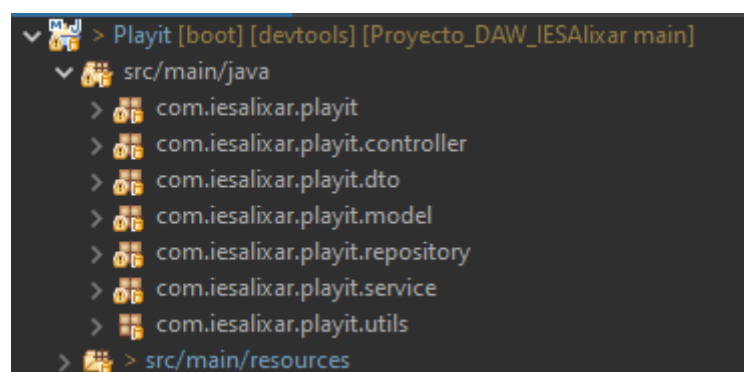


Figura 2. Paquetería del proyecto.

4.4 Diseño de la interfaz de usuario.

Esta parte se corresponde con la capa de presentación, que es el conjunto de componentes que implementan la interacción con los usuarios a través de una representación visual de la aplicación. A partir de la interfaz gráfica el usuario podrá navegar por las distintas páginas para poder obtener toda la información que desee, o aportarla en caso de ser necesario.

La gestión de la información se realiza mediante el framework Spring y Thymeleaf.

La distribución que se ha adoptado es la siguiente.

- Img. Imágenes de fondos.
- CSS. Hojas de estilos.
- JS. Javascript necesarios.
- Admin. HTML de la interfaz de administrador.
- User. HTML de la interfaz de usuario estandar.
- parts. Partes de HTML que se reutilizan en diferentes páginas.

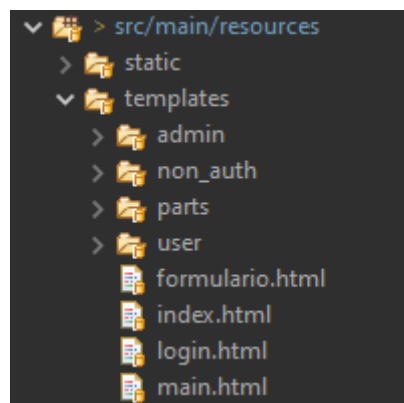


Figura 3. Paquetería del proyecto.

En el documento “Mapa de navegación”, existe una detallada descripción de la navegación en la aplicación web.

5. Despliegue.

El despliegue se realizará en el propio Spring Tool Suite. Para ello solo necesitaremos cualquier gestor de base de datos MySQL y hacer uso de Docker.

Una vez hemos instalado Docker en nuestro sistema haremos uso de la siguiente instrucción:

```
docker run --name bbdd -p 3336:3306 -e MYSQL_ROOT_PASSWORD=root mysql
```

Esto nos creará un contenedor mysql con el que desde nuestro gestor de base de datos podremos acceder a la base de datos por medio del puerto 3336 y la

contraseña root. Tras poner en marcha la base de datos lo único que tendremos que hacer es abrir el proyecto en Spring Tool Suite, hacer click derecho sobre el proyecto, seleccionar Run as... y la opción de Spring Boot App. Como podemos ver en la Figura 4.

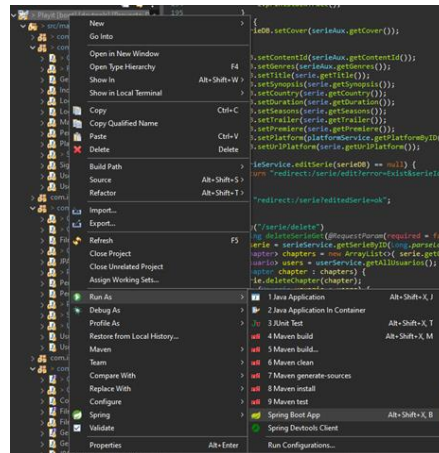


Figura 4. Despliegue proyecto.

En la pestaña de Boot Dashboard veremos que se ha desplegado el proyecto (Figura 5).

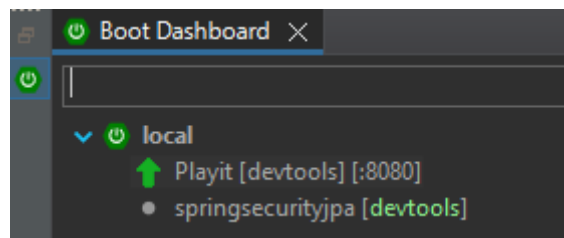


Figura 5. Proyecto desplegado.

Por ultimo hacemos click derecho sobre el proyecto en la pestaña Boot Dashboard y seleccionamos Open Web Browser (figura 6) para que se nos abra la pestaña inicial de la aplicación web.

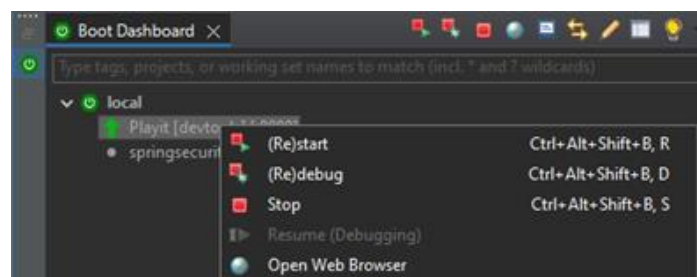


Figura 6. Iniciar aplicación..

6. Posibles errores.

Entre los errores más probables ante posibles fallos están:

- La información del contenido no ha sido añadido en su totalidad (podría faltar añadir género o reparto).
- Pérdida de distribución o del responsive debido a que las librerías ya no estén disponibles.