

COOLSOUND



12/03/2021

Desarrollo de Aplicaciones Web

Fernando Manuel Rabanal Márquez

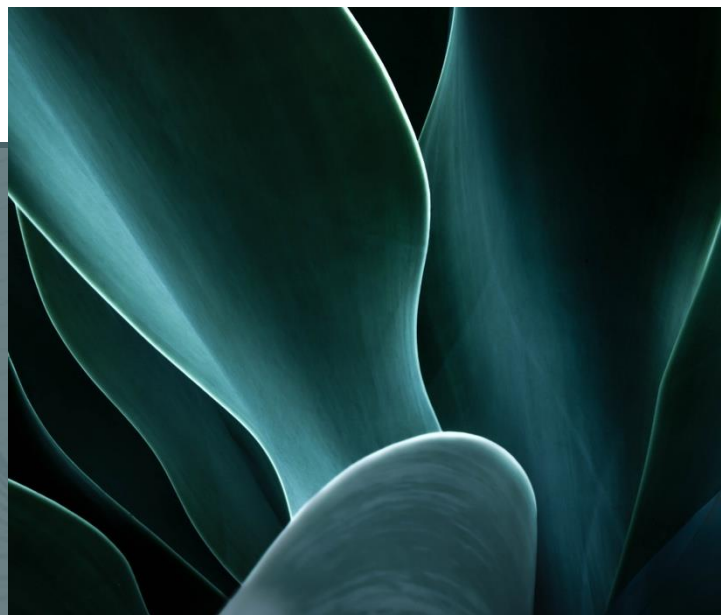
Agradecimientos

Me gustaría dedicar este proyecto a muchas personas por la paciencia, ayuda o tiempo que no les he podido prestar.

“A mis Padres eternamente, por los principios, valores y consejos que me han inculcado siempre para seguir progresando en el duro camino de la vida.”

“Al profesorado de este curso por su confianza, apoyo e infinita paciencia en mi formación para conseguir los objetivos marcados. Gracias de corazón.”

“A los demás, mis amigos y mi hermano, gracias por haberme ayudado a hacer esto posible. “



INDICE

| | |
|--|----|
| COOLSOUND | 1 |
| 1. PRESENTACION DEL PROYECTO | 4 |
| 1.1 INTRODUCCIÓN | 4 |
| 1.2 PLANTEAMIENTO DE LA TAREA | 5 |
| 1.3 OBJETIVOS Y DESARROLLO | 5 |
| 1.3 TECNOLOGÍA Y HERRAMIENTAS EMPLEADAS | 6 |
| 2. DISEÑO DEL SISTEMA | 13 |
| 2.1 ARQUITECTURA DEL SISTEMA | 13 |
| 2.2 REQUISITOS DEL SISTEMA | 18 |
| 2.3 DEFINICIÓN DE INTERFACES DE USUARIO | 19 |
| 2.4 CASOS DE USO | 20 |
| 3. DESARROLLO DEL SISTEMA | 22 |
| 3.1 PLANIFICACIÓN DE LAS ACTIVIDADES DE DESARROLLO | 22 |
| 3.2 DESCRIPCIÓN DEL ENTORNO DE DESARROLLO | 24 |
| 3.3 DOCUMENTACION | 25 |
| 4. DESPLIEGUE E IMPLANTACIÓN | 26 |
| 4.1 Implantación del sistema y pruebas | 26 |
| 4.1.1 Instalación de servidores y aplicaciones. | 26 |
| 4.1.2 Mockups | 30 |
| SOBRE NOSOTROS | 34 |
| 5. MANTENIMIENTO | 39 |
| 6. CONCLUSIONES | 40 |
| 6.1 Conclusiones científicas | 40 |
| 6.2 Conclusiones personales | 40 |
| 7. LINEAS FUTURAS | 41 |
| 8. BIBLIOGRAFIA | 42 |

1. PRESENTACION DEL PROYECTO

1.1 INTRODUCCIÓN

Este es el documento de memoria para el proyecto final del grado superior Desarrollo de Aplicaciones Web realizado en el IES JACARANDÁ, Brenes. En él encontraremos diversas secciones donde se detalla desde la fase de planificación hasta la fase de implementación y puesta en marcha.

La idea del proyecto final es una web con temática musical.

La idea principal del proyecto es una web donde usuarios comunes puedan acceder y puedan escuchar y ver videoclips de otros artistas, puedan configurar sus propias playlist para escuchar las que ellos mismos elijan. También ofrecerá un sistema donde podrán subir sus propios videoclips. Dentro de la web podrán valorar videoclips y tendrá un foro donde podrán crear hilos de discusión y comentar en diferentes hilos.

La aplicación tendrá una serie de usuarios, que pueden ser de dos tipos: administrador y usuario normal.

El usuario normal podrá ver en la pantalla principal los videoclips más escuchados y podrá loguearse o registrarse en la web. Ya logueado podrá acceder a las diferentes pantallas donde puede buscar videoclips, podrá valorar, escuchar y comentar dichos videoclips. También podrán añadirlos a playlists que estos mismos usuarios hayan creado anteriormente en la pantalla de playlist.

Dichos usuarios podrán escuchar las playlist que hayan creado ellos mismos. También podrán modificar sus datos. También podrán acceder al foro donde podrán crear nuevos hilos e incluso entrar en hilos creados y comentar.

Otra acción que tendrán disponible es acceder a la pantalla Sobre Nosotros donde podrán mandar una duda o sugerencia a los administradores.

El otro tipo de usuario que habrá en la aplicación es el usuario administrador, este usuario podrá hacer todo lo que un usuario normal puede hacer en la web.

Aparte de todo lo anterior, podrá modificar la información del menú principal, podrá borrar usuarios de la web, podrá administrar los videoclips, dentro del foro podrá moderar los mensajes escritos o hilos creados de tal forma que no haya contenido inapropiado dentro de la web.

1.2 PLANTEAMIENTO DE LA TAREA

En el planeamiento para el desarrollo de la aplicación se ha seguido una guía confeccionada por el profesorado del curso, que a su vez ha sido consensuada con el alumno mediante un documento llamado “Historia del Usuario”.

Estos documentos han sentado las bases para el desarrollo completo de nuestra página web, denominada “CoolSound”.

Para cada fase de implementación de la página, se ha ido convocando reuniones al profesor a cargo del seguimiento del proyecto para una supervisión del mismo.

1.3 OBJETIVOS Y DESARROLLO

El presente documento está dividido en una serie de apartados que corresponden, esencialmente, a las distintas etapas que conforman el proceso de desarrollo del proyecto. Estas etapas han sido:

Especificación de requisitos

Se redactó de una manera global una primera visión del proyecto donde señalamos los requisitos que debía cumplir. La finalidad de esta etapa era plasmar el acuerdo entre el desarrollador y el cliente acerca de las funcionalidades del proyecto.

Análisis

Se realizó el modelado conceptual de la futura solución mediante el uso de diagramas (diagrama de E/R y diagramas de casos de uso) que veremos más adelante. Los modelos ayudan a visualizar como es el sistema, proporcionando plantillas que sirven de guía en la construcción de la aplicación. En esta etapa se especifica qué puede hacer la aplicación, pero no cómo debe hacerlo.

Diseño

Se utilizaron los elementos y modelos obtenidos durante el análisis para transformarlos en mecanismos que puedan ser utilizados en un entorno web con las características y condiciones que establecen este tipo de entornos. Se diseñaron todos los niveles de los que consta la aplicación (nivel de presentación, nivel lógico y nivel de persistencia). Tanto la etapa del análisis como la del diseño están desprovistas de código. Un buen análisis y un buen diseño son la mejor forma de llegar a producir software de calidad.

Implementación

Se utilizaron los elementos obtenidos en el diseño para permitir la elaboración del producto o prototipo funcional, es decir, que puede ser puesto en marcha y sometido a 6 pruebas. Para ello se consideraron las diversas tecnologías que han intervenido en la elaboración de dicho producto. Todo lo desarrollado en las etapas del análisis y del diseño, se trasladó a código.

1.3 TECNOLOGÍA Y HERRAMIENTAS EMPLEADAS

Propósito

El propósito de la especificación de requisitos es definir cuáles son los requerimientos que debe tener la aplicación que se va a desarrollar y describir la funcionalidad que tendrán los usuarios en todo su entorno.

Ámbito

El desarrollo del sitio web está orientado a ofrecer contenidos y funcionalidades que ayuden a poder obtener información sobre las reuniones efectuadas en la aplicación.

Definiciones, siglas y abreviaturas:

Sitio web:

Conjunto de archivos electrónicos y páginas web referentes a un tema en particular que incluye una página inicial de bienvenida, con un nombre de dominio y una dirección en Internet específicos.

Interfaz:

Parte del programa informático que permite el flujo de información entre varias aplicaciones o entre el propio programa y el usuario.

Navegador:

Permite al usuario recuperar y visualizar páginas web a través de Internet.

Servidor web:

Se trata de un programa que implementa el protocolo HTTPS (HyperText Transfer Protocol Secure). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, vínculos, imágenes, formularios, botones y objetos incrustados, etc.

Tecnologías Utilizadas

El desarrollo de la aplicación lo dividiremos en dos partes, en el frontend y en el backend:

-Para la parte del Backend, usaremos el lenguaje de programación Java, usando Spring Boot como framework para montar el backend como un api rest que mandara la información al frontend, implementare Hibernate y JPA como conector a base de datos.

- Para el frontend, usaremos Angular, donde usaremos HTML y Javascript/Typescript. Además, utilizaremos como añadidos Bootstrap y Angular Material para el diseño del front.

La base de datos que usaremos para la aplicación será una base de datos MySQL.

Para el entorno de producción montaremos en Docker los siguientes contenedores donde desplegaremos las diferentes partes que componen la aplicación.

- Un contenedor mysql que deberá montar un volumen para almacenar la base de datos.
- Un contenedor wildfly que deberá montar un volumen para almacenar el backend.
- Un contenedor apache que deberá montar un volumen para almacenar el frontend.
- Un contenedor jenkins que deberá desplegar el frontend y el backend de forma automática.

Hemos elegido estos lenguajes, tecnologías y FrameWorks porque son los que, a nuestro parecer, los más adecuados a la hora de realizar nuestro proyecto, ya que:

- Son gratuitos.
- Lleva muy bien a la práctica el concepto Modelo-Vista-Controlador, lo cual es muy recomendable para el mantenimiento de aplicaciones.
- Buena comunicación entre el Backend y el Frontend.
- Son fáciles de aprender y manejar.
- Son multiplataforma.
- Orientado a objetos.
- Estructurado de carpetas intuitivo y cómodo.

HTML

HTML5 es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML.1 2 Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y , pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web) y <footer>. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>.

Hay muchos atributos y etiquetas nuevas los cuales son utilizadas constantemente en desarrollo de mi proyecto de curso. Ha sido muy importante la interacción de esta tecnología con Bootstrap.

```
src > app > components > nuevacancion > nuevacancion.component.html > div.bordes.container.formRegister > form > div.form-group > label
1 <br>
2 <hr>
3 <br>
4 <div class="bordes container formRegister">
5   <br>
6   <h3 style="text-align: center;">Añadir Nueva Canción</h3>
7   <form [formGroup]="registerForm" (ngSubmit)="addNewCancion(registerForm)" method="POST">
8     <div class="form-group">
9       <label for="titulo">Título:</label>
10      <input type="text" class="form-control" formControlName="titulo" id="titulo" placeholder="Título...">
11      <div class="alert alert-danger" *ngIf="(titulo.touched) && (titulo.invalid)" style="margin-top: 5px;">
12        <span *ngIf="titulo.errors.required">El título es un campo requerido</span>
13        <span *ngIf="titulo.errors.minlength">
14          Longitud mínima del título: {{nombre.errors.minlength.requiredLength}}
15        </span>
16      </div>
17    </div>
18    <div class="form-group">
19      <label for="genero">Género</label>
20      <br>
21      <select name="genero" id="genero" class="sidebar-box">
22        <option value="POP">POP</option>
23        <option value="HIP HOP">HIP HOP</option>
24        <option value="RAP">RAP</option>
25        <option value="ELECTRÓNICA">ELECTRÓNICA</option>
26        <option value="HOUSE">HOUSE</option>
27        <option value="ROCK">ROCK</option>
28        <option value="INDIE">INDIE</option>
29        <option value="REGGAETON">REGGAETON</option>
30      </select>
31    </div>
32    <div class="form-group">
33      <label for="cancion">URL de Youtube del Videoclip</label>
34      <input type="text" class="form-control" formControlName="cancion" id="cancion" placeholder="Canción...">
35      <div class="alert alert-danger" *ngIf="(cancion.touched) && (cancion.invalid)" style="margin-top: 5px;">
36        <span *ngIf="cancion.errors.required">La canción es requerida</span>
37      </div>
38    </div>
39    <br>
40    <button [disabled]="registerForm.valid" type="submit" class="btn btn-success">Añadir</button>
41    <button type="button" class="btn btn-light" [routerLink]="['/mimusica']">Cancelar</button>
42    <br><br>
43  </form>
44 </div>
45 <br>
```

Esta parte del código hace referencia a la ventana de creación de nuevos Videoclips de los usuarios del sistema. En ella se pueden apreciar diferentes etiquetas HTML.

CSS/SCSS

CSS u Hojas de Estilo en Cascada (Cascading Style Sheets) es el lenguaje de hojas de estilo utilizado para describir el aspecto y el formato de mi página web. El proyecto cuenta con bastantes archivos de aspecto diferentes, cada uno con configuraciones únicas para cada sección de la página.

El uso de varias hojas de estilo me ha permitido poder editar la página web de una manera dinámica, sin tener que especificar en cada hoja qué tipo de estilo quiero que tenga cada parte de ésta.

Prácticamente todo el código se mueve por “clases”, para así poder aplicar un mismo estilo a varios elementos a la vez dentro de la página web.

```
src > app > components > hilo > hilo.component.scss > .bordes
1  .imgRedonda {
2      width:60px;
3      height:300px;
4      border-radius:150px;
5      border:3px solid rgb(155, 154, 154);
6      height: auto;
7  }
8
9  .bordes {
10     border-radius: 33px 33px 33px 33px;
11     -moz-border-radius: 33px 33px 33px 33px;
12     -webkit-border-radius: 33px 33px 33px 33px;
13     border: 2px solid #000000;
14 }
```

En esta captura se pueden apreciar diferentes tipos de estilos propios de CSS3. Por ejemplo, aquí es llamada con “imgRedonda” delante, que sirven para dar estilos a las imágenes, y así poder hacerlos más agradables e intuitivos de cara al usuario.

Javascript

La utilización de Javascript (abreviado js) ha sido una buena elección a la hora de mejorar el contenido de la web. Sus funciones en mi proyecto han tenido una gran importancia a la hora de mejorar la funcionalidad de la página web.

Se utiliza el parte del front para conectarse con el back y así traer información hacia el front, así como para la creación de métodos que ayudan en la funcionalidad de la web.

```
src > app > components > musicadetalle > TS musicadetalle.component.ts > MusicadetalleComponent > ngOnInit
38  ngOnInit() {
39    this.id = this._route.snapshot.paramMap.get('idCancion');
40    console.log(this.id);
41    this.valoracion = this._route.snapshot.paramMap.get('valoracion');
42    console.log(this.valoracion);
43    this.vecesValorada = Number(this._route.snapshot.paramMap.get('vecesValorada'));
44    console.log(this.vecesValorada);
45
46    this.cs.vecesEscuchada(this.id).subscribe();
47
48    let usuario = JSON.parse(sessionStorage.getItem('usuario'));
49    this.rol = usuario.rol;
50    this.idUsuario = usuario.idUsuario;
51
52    this.cs.unaCancion(this.id).subscribe(data => {
53      this.canciones = data;
54    })
55
56    this.ccs.comentario(this.id).subscribe( data => {
57      this.comentarios = data;
58    })
59
60    this.ps.playlistByUser(this.idUsuario).subscribe( data => {
61      this.playlists = data;
62    })
63
64  }
65
66  valorar(){
67    var valorar = (<HTMLInputElement>document.getElementById('valorar')).value;
68    var vecesValorada2: number;
69
70    vecesValorada2 = this.vecesValorada + 1;
71    console.log("VECES VALORADA" + vecesValorada2);
72    this.valoracion = (Number(valorar) + Number(this.valoracion)) / vecesValorada2;
73    console.log("MEDIA" + this.valoracion);
74
75    this.cs.changeValoracion(this.valoracion, vecesValorada2, this.id).subscribe(data=> {
76      Swal.fire({
77        icon: 'success',
78        title: 'Valoración añadida con éxito.',
79        showConfirmButton: false,
80        timer: 1500
81      });
82    });
83  };
```

La mayoría de las funciones se han tenido que crear usando esta tecnología. Las librerías de javascript que he utilizado en este proyecto son bastante complejas en temas de funcionamiento, ya que son muchísimas las funciones diferentes que hacen que funcione todo correctamente. Llevó un tiempo poder leer toda la documentación y decidir qué utilizar o qué no utilizar, pero creo que el resultado final ha sido bueno.

Bootstrap

Bootstrap es un framework front-end para un desarrollo responsivo, el cual es muy interesante a parte por todas las funcionalidades que ofrece del tipo de css para el estilo de la página, su función más importante es que prioriza el contenido que se va a mostrar en el caso de que la página sea vista desde un dispositivo móvil.

Actualmente, la industria del teléfono móvil ha cobrado mucho poder con respecto a sus anteriores años, sobretodo ahora con los nuevos “smartphone” y los sistemas operativos tales como android o WindowsPhone, que llevan el uso del teléfono a un nuevo nivel.

Muchas de las páginas web de hoy en día son vistas por personas con sus dispositivos móviles, o sus “tablets” por el hecho de que sean personas sin mucho tiempo o demás, con lo cual desarrollar las páginas web pensando en que va a ser útil y visible desde un Smartphone es muy importante. Bootstrap nos ofrece la posibilidad de hacer este tipo de páginas utilizando sus etiquetas y sus hojas de estilo.

MySQL

MySQL es un sistema de gestión de bases de datos. Existen varias interfaces de programación de aplicaciones que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, C#, Pascal, Delphi (vía dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby, Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una interfaz de programación de aplicaciones específica.

Para mi proyecto me he decantado por usar PhpMyAdmin, que es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet.

Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, etc.

Java

El back esta completamente desarrollado en Java con su framework Spring Boot.

Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. Java no es lo mismo que javascript, que se trata de una tecnología sencilla que se usa para crear páginas web y solamente se ejecuta en el explorador.

Java sirve para **crear aplicaciones y procesos** en una gran diversidad de dispositivos. Se base en programación orientada a objetos, permite ejecutar un mismo programa en diversos sistemas operativos y ejecutar el código en sistemas remotos de manera segura.

```
1 package com.java.coolsound.controller;
2
3 import java.util.List;
4
5 import javax.servlet.http.HttpServletRequest;
6
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.web.bind.annotation.GetMapping;
9 import org.springframework.web.bind.annotation.PathVariable;
10 import org.springframework.web.bind.annotation.PostMapping;
11 import org.springframework.web.bind.annotation.RequestBody;
12 import org.springframework.web.bind.annotation.RequestMapping;
13 import org.springframework.web.bind.annotation.RestController;
14
15 import com.java.coolsound.model.Usuario;
16 import com.java.coolsound.service.UserService;
17
18 /**
19  * The Class UsuariosController.
20  *
21  * @author fernando
22  */
23 @RestController
24 @RequestMapping(value = "/usuarios")
25 public class UsuariosController {
26
27     /** The UserService */
28     @Autowired
29     UserService us;
30
31     /**
32      * Login.
33      *
34      * @param nombre the nombre
35      * @param clave the clave
36      * @return the usuarios
37      */
38     @GetMapping("/login/{nombre}/{contraseña}")
39     public Usuario login(@PathVariable String nombre, @PathVariable String contraseña) {
40         return us.login(nombre, contraseña);
41     }
42
43     /**
44      * Gets the all usuarios.
45      *
46      * @return the all usuarios
47      */
48     @GetMapping("/getusuarios")
49     public List<Usuario> getAllUsuarios() {
50         return us.getAllUsuarios();
51     }
52 }
```

Esta captura referencia el controlador de la clase Usuarios desarrollado en Java.

2. DISEÑO DEL SISTEMA

2.1 ARQUITECTURA DEL SISTEMA

Actualmente existen diferentes lenguajes de programación y desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios dándose soluciones mediante lenguajes estáticos, pero que, con el paso del tiempo, las tecnologías se desarrollaron y surgieron nuevos problemas a solucionar, lo que dio lugar a crear lenguajes de programación para la web dinámicos, que permitiesen interactuar con los usuarios y que hicieran uso de los sistemas de Bases de Datos. Estos se clasifican en lenguajes del lado cliente (FrontEnd) y lenguajes del lado servidor (BackEnd).



Figura 1.- Lenguajes de Desarrollo y Editores de Texto

Nuestro proyecto está dividido en dos partes, FrontEnd (parte del cliente, o sea la que verá el Usuario) y el BackEnd (parte creativa del desarrollador o programador).

Para el BackEnd hemos hecho uso del lenguaje Java versión 11 con SpringBoot 1.5.10 release. Desde la raíz, lo hemos estructurado en una serie de carpetas que contienen toda la implementación de cada fase del proyecto. Hay carpetas dedicadas en exclusiva al desarrollo de los servicios para implementar las distintas funcionalidades y un fichero donde se aloja la configuración para conectar con la Base de Datos.

Tenemos una serie de carpetas dedicadas a la creación de objetos que a su vez son las distintas tablas que nos encontramos en nuestra Base de Datos.

En la siguiente imagen podemos apreciar la composición gráfica de la estructura de carpetas de nuestro proyecto.

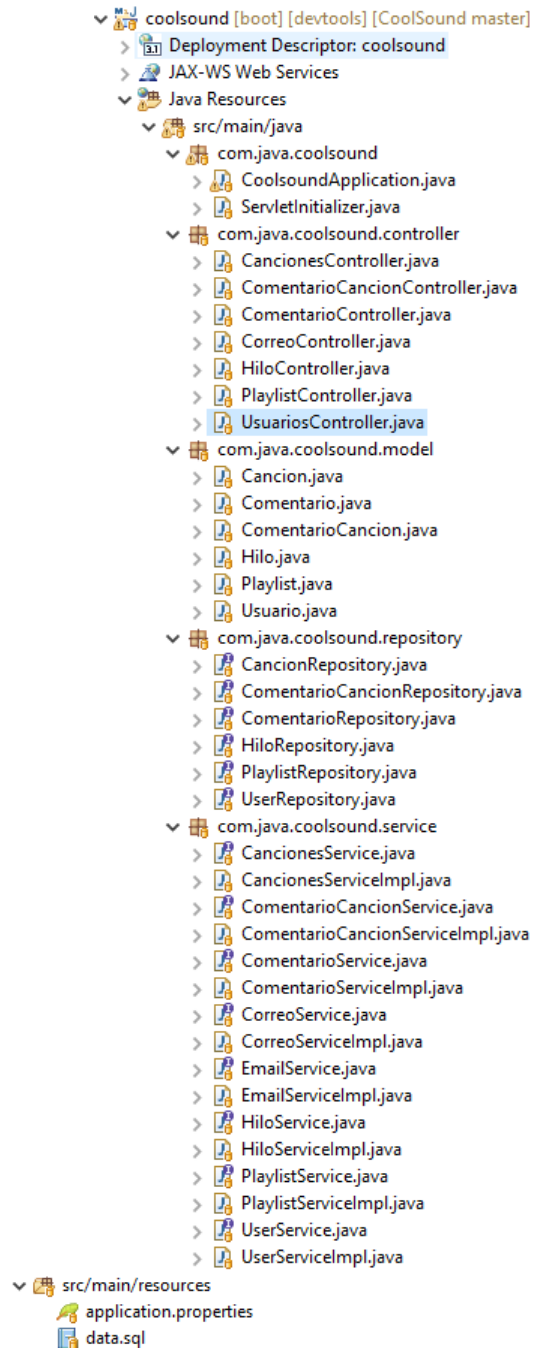


Figura 2.- Estructura de carpetas del BackEnd.

Entre las carpetas del Backend podemos encontrarnos con las siguientes:

- src/main/java: Aquí se encuentran los paquetes, Controller, Repository, Service y Modelo, los cuales contendrán las respectivas clases para hacer funcionar el Backend.
 - Model: contiene las clases correspondientes a las Entidades de nuestro proyecto (atributos, constructores, etc.)
 - Repository: contiene las clases que albergan todas las acciones de consulta en base de datos.
 - Service: contiene el código de los servicios creados para esta aplicación.
 - Controller: contiene los controladores que dan funcionalidad a la aplicación.
- src/main/resources: contiene dos ficheros;
 - application.properties: contiene la configuración para la conexión a base de datos.
 - data.sql: contiene consultas que se inicializan automáticamente al iniciarse la aplicación.
- pom.xml: Fichero donde se encuentran todas las dependencias utilizadas en esta aplicación.

Para una buena organización y buena eficiencia del código se ha usado el modelo de Arquitectura de 3 capas, en la siguiente imagen podríamos ver cómo funciona.

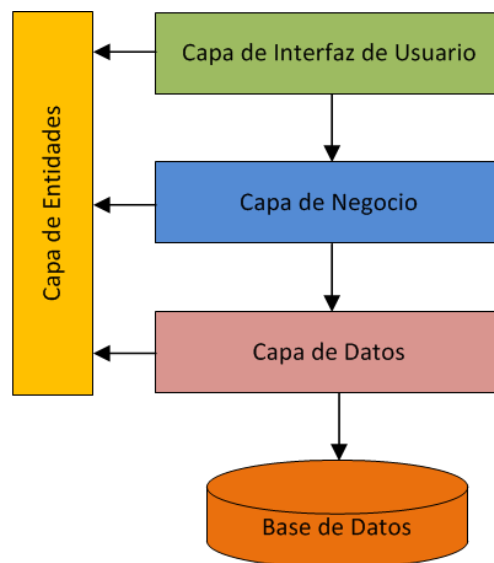


Figura 3. Arquitectura de 3 capas.

En este enlace podremos encontrar el código completo de la parte del Back del proyecto:
<https://github.com/FernandoRM97/CoolSound/tree/master/COOLSOUND-BACKEND/coolsound>.

Para el apartado del FrontEnd hemos usado el lenguaje Angular en su versión 8, así como Node v12.14.1., realizando una tarea similar en cuanto a la creación de carpetas desde la raíz. Estas carpetas y sus subcarpetas contienen las funciones necesarias para implementar el lado Cliente, donde se proporciona mediante códigos precisos la gestión de la página. Obsérvese dicha configuración en el siguiente cuadro.

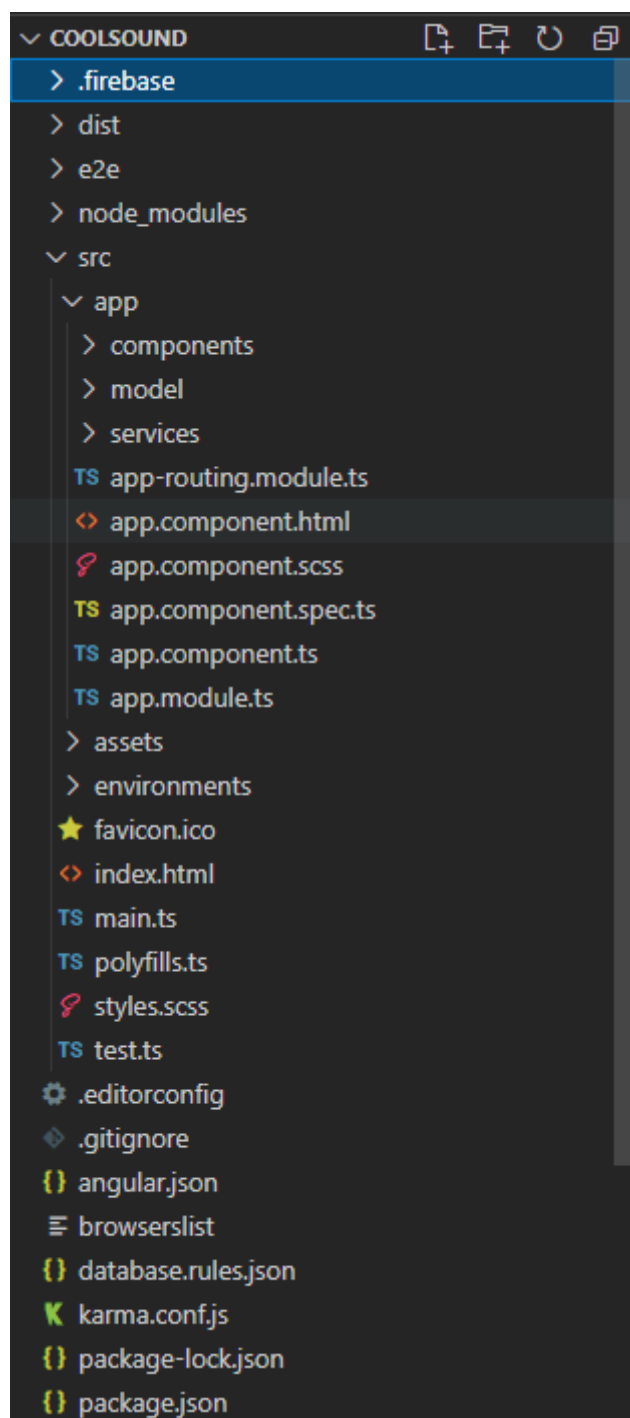


Figura 5.- Estructura de carpetas del FrontEnd

Entre las carpetas del Frontend podemos encontrarnos con las siguientes:

- src/app: contiene algunas de las carpetas donde se realiza la configuración de la aplicación, estas son:

- model: contiene los objetos correspondientes a las entidades del Backend.

- components: contiene los distintos componentes que forman la visualización y configuración del usuario en la aplicación.

- services: aquí se encuentran los métodos que se encargan de comunicarse con el Backend para traer o llevar información.

- src/assets: contiene una carpeta con algunas de las imágenes usadas en la aplicación como el logotipo de la web.

En este enlace podremos encontrar el código completo de la parte del Front del proyecto:

<https://github.com/FernandoRM97/CoolSound/tree/master/COOLSOUND-FRONTEND/coolsound>.

En la siguiente imagen aparece el diseño Entidad/Relación de la BD:

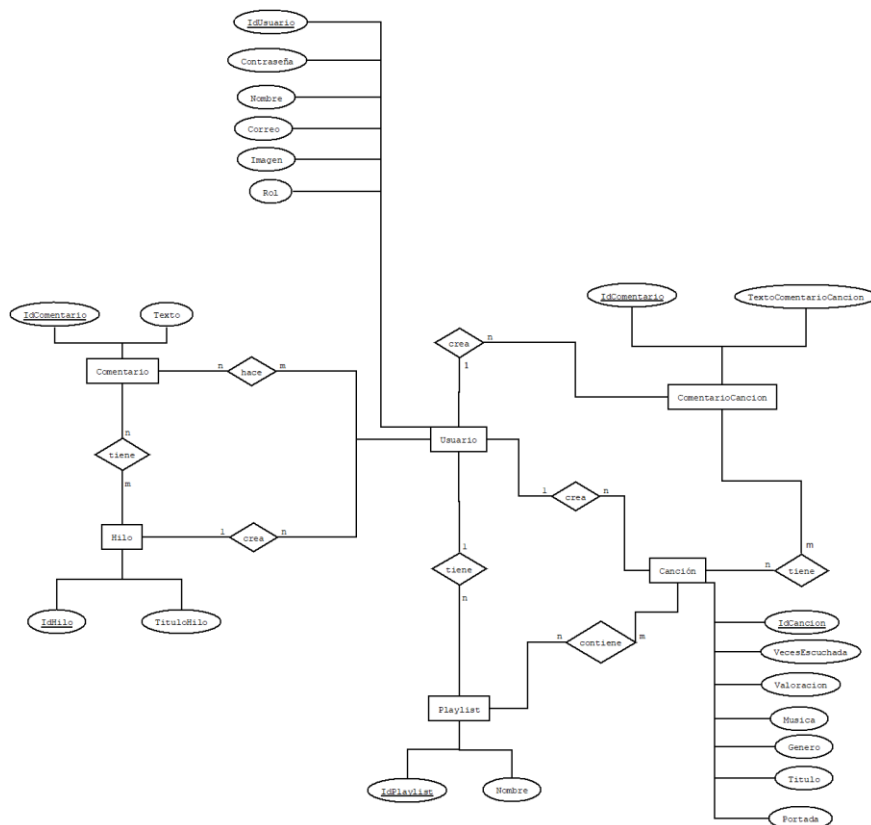


Figura 6.- Diseño E/R base de datos.

2.2 REQUISITOS DEL SISTEMA

Requisitos técnicos

| | |
|--------------|---|
| ARQUITECTURA | Nuestro sitio Web será compatible con la mayoría de los navegadores que operan en Internet: Internet Explorer, Google Chrome, Mozilla, Opera y Safari |
| ARQUITECTURA | Nuestro sitio Web será compatible con una parte importante de dispositivos periféricos de futuros Usuarios. (Web Responsive) |
| SEGURIDAD | La aplicación permitirá la gestión de Usuarios por el Administrador del Sistema. |

Requisitos operativos

| | |
|-----------|---|
| OPERATIVA | Nuestro sitio Web tendrá un diseño atractivo y limpio, incluyendo el logotipo comercial. |
| OPERATIVA | La estructura de los contenidos será clara e intuitiva, con botones de acceso nítidos. |
| OPERATIVA | La aplicación permitirá compartir la información contenida en las áreas comunes a todos los Usuarios autenticados |
| OPERATIVA | La gestión de los contenidos podrá ser realizada por usuarios sin conocimientos en programación, de forma fácil e intuitiva, dentro de las áreas específicas a los roles de los diferentes Usuarios |
| OPERATIVA | Permitirá subir y borrar contenido subido por el mismo usuario. |

2.3 DEFINICIÓN DE INTERFACES DE USUARIO

Los interfaces del usuario (UI) son el conjunto de los controles y canales sensoriales mediante los cuales un usuario puede comunicarse con una máquina. Por ejemplo, en un pc de sobremesa o portátil serían, la pantalla, el teclado, el ratón y los altavoces. Forman la parte de la interfaz de la utilidad donde todas ellas hacen que entre o salga información del equipo. Una buena interfaz de usuario se caracteriza por tener un alto grado de usabilidad, y por ser intuitiva.

Pues con este principio, aplicable a nuestro caso, hemos querido implementar una página web donde el diseño sea efectivo, y lograr que los usuarios del sitio puedan acceder con facilidad a los contenidos, interactuar con eficacia con todos los componentes y sentirse cómodos de forma permanente.

Una parte importante de los éxitos o fracasos de los sitios dependen en gran medida de este elemento fundamental del diseño. Además, el diseño de la interface debe perseguir hacer webs usables y a su vez amigables.

Hemos desarrollado una página limpia y sencilla, donde los controles de las acciones están diseñados con claridad, basándonos en una gama cromática de tonos blancos y negros, persiguiendo que su uso no produzca cansancio visual y que, de alguna forma, anime al usuario a seguir un poco más cada vez que los use.



Figura 7. – Interfaces de Usuario

Referencias utilizadas

- Guía para la realización y verificación de proyectos final de grado en el ámbito de la web y apuntes de las asignaturas del curso 2019-2020.

A continuación, se realizará la descripción general del sistema desarrollado.

Perspectiva del producto

La aplicación desarrollada ofrecerá información sobre videoclips, así como información más específica para cada tipo de usuario que esté registrado. Se podrá acceder a la aplicación desde cualquier sistema operativo que tenga conexión a Internet utilizando un navegador web.

Restricciones

Al tratarse de una aplicación web, se requiere como mínimo un ordenador / portátil / teléfono móvil / Tablet, etc. con un navegador convencional y una conexión a Internet básica.

Funciones del producto

A continuación, se muestran las funciones que conforman la aplicación, según el tipo de usuario que se encuentre conectado con sus funciones, características del/los usuarios, supuestos, restricciones, y áreas de trabajo.

2.4 CASOS DE USO

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

En esta web tendremos dos tipos de usuario o actores:

Usuario Tipo:

Estos usuarios pueden acceder a la aplicación a través de su nombre de usuario y contraseña, representa a un usuario normal en la aplicación.

Usuario Administrador:

Estos usuarios pueden acceder a la aplicación a través de sus credenciales, representan al administrador de la aplicación, tendrán control absoluto en la aplicación.

ACTORES



Usuario Tipo

REGISTRARSE
INICIAR SESIÓN
CONSULTAR VIDEOCLIPS
ESCUCHAR VIDEOCLIP
AÑADIR VIDEOCLIP A PLAYLIST
EVALUAR VIDEOCLIP
COMENTAR CANCIÓN
ENVIAR DUDA O SUGERENCIA
SUBIR VIDEOCLIP
CREAR PLAYLIST
ESCUCHAR PLAYLIST
CREAR HILO DE DISCUSIÓN
CONSULTAR FORO
CREAR COMENTARIO EN HILO
CONFIGURAR PERFIL
CERRAR SESIÓN



Usuario Administrador

REGISTRARSE
INICIAR SESIÓN
CONSULTAR VIDEOCLIPS
ESCUCHAR VIDEOCLIP
AÑADIR VIDEOCLIP A PLAYLIST
EVALUAR VIDEOCLIP
COMENTAR VIDEOCLIP
ENVIAR DUDA O SUGERENCIA
SUBIR VIDEOCLIP
CREAR VIDEOCLIP
ESCUCHAR VIDEOCLIP
CREAR HILO DE DISCUSIÓN
CONSULTAR FORO
CREAR COMENTARIO EN HILO
CONFIGURAR PERFIL
ACCEDER AL PANEL DE ADMINSTRACIÓN
BORRAR HILO DE DISCUSIÓN
BORRAR COMENTARIO EN HILO
BORRAR VIDEOCLIP
BORRAR USUARIO
CERRAR SESIÓN

3. DESARROLLO DEL SISTEMA

3.1 PLANIFICACIÓN DE LAS ACTIVIDADES DE DESARROLLO

La planificación es una parte esencial de cualquier proyecto de diseño o desarrollo web. Contar con un cronograma del proyecto nos permitirá perfilar, evaluar y comunicar los hitos y tareas esenciales del proyecto, así como planificar los entregables con los plazos y recursos asignados en cada caso.

En cualquier proyecto podemos hacer uso de dos tipos de cronogramas, que son el Work BreakDown Structure (WBS) o Estructura Detallada del Trabajo, una herramienta para descomponer analíticamente el proyecto en partes elementales que generalmente se representa en forma gráfica (estructura de árbol) descriptiva (estructura de índice), donde el objetivo fundamental es identificar en el último nivel de descomposición los llamados “works packages” mejor definidos como paquetes de trabajo. En los paquetes de trabajo se indican todas las instrucciones a realizar para efectuar una determinada tarea o actividad.

El otro tipo es el Diagrama de Gantt, una herramienta útil para planificar cualquier tipo de proyecto, pues nos proporciona una vista general de las tareas programadas, donde todas las partes implicadas sabrán qué tareas tienen que completarse y en qué fecha, convirtiendo el proceso en un método más eficiente y ameno para todos los equipos participantes.

- La fecha de inicio y finalización de un proyecto
- Qué tareas hay dentro del proyecto
- Quién está trabajando en cada tarea
- La fecha programada de inicio y finalización de las tareas
- Una estimación de cuánto llevará cada tarea
- Cómo se superponen las tareas y/o si hay una relación entre ellas

Claridad, Comunicación, Motivación, Gestión del Tiempo y Flexibilidad, son las principales ventajas de esta herramienta.

En nuestro caso particular, hemos optado por usar el diagrama de Gantt al ya tener una aproximación creada en clase, y de esta forma adaptarla a nuestro desarrollo, que en su modo más básico quedaba así:

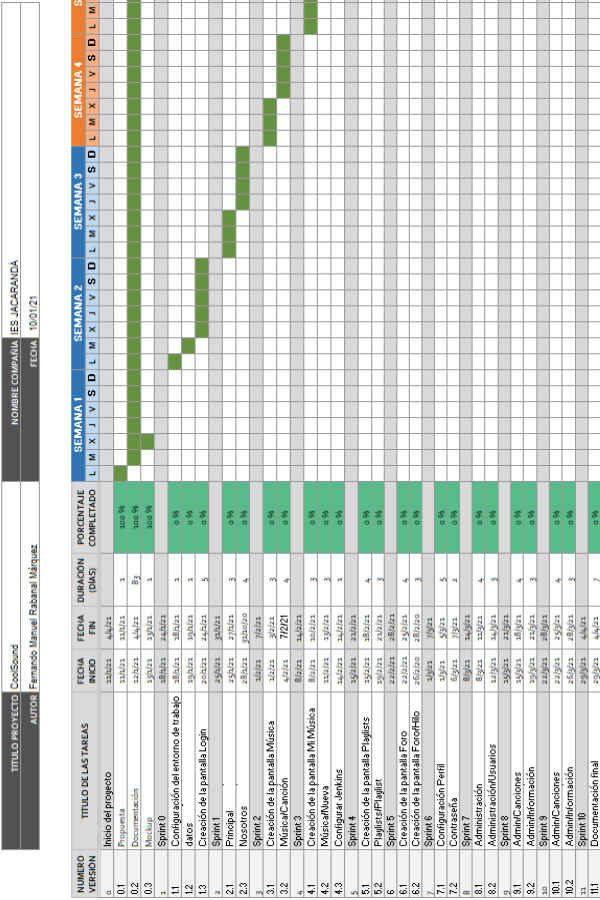


Figura 8 - Diagrama de Gantt

3.2 DESCRIPCIÓN DEL ENTORNO DE DESARROLLO

El entorno de desarrollo es un conjunto de procedimientos y herramientas que se utilizan para desarrollar un código fuente o programa. Este término se utiliza a veces como sinónimo de entorno de desarrollo integrado (IDE), que es la herramienta de desarrollo de software utilizado para escribir, generar, probar y depurar un programa. También proporcionan a los desarrolladores una interfaz de usuario común (UI) para desarrollar y depurar en diferentes modos.

A la hora de elegir el entorno de desarrollo o IDE (Integrated Development Environment) es fundamental tener definido que lenguaje de programación se va a utilizar tanto en el Backend como en el Frontend.

Como ya indicábamos en el apartado 2.1 Arquitectura del Sistema, definimos claramente que tipo de lenguajes usaríamos en cada apartado.

Para el BackEnd hemos usado Java por el conocimiento que tenemos sobre él, aparte de que es un lenguaje orientado a objetos, rápido seguro y fiable, estando indicado particularmente para aplicaciones cliente-servidor, y como entrono de desarrollo hemos utilizado Eclipse JEE.



Figura 9 – Icono Eclipse

Para el FrontEnd hemos utilizado Visual Studio Code, es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluyendo un soporte para la depuración, con control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. Se basa en Electron, un framework que se utiliza para implementar Chromium y Node.js como aplicaciones para escritorio, que se ejecuta en el motor de diseño Blink.

Puede extenderse a través de complementos, disponible a través de un repositorio central. Esto incluye adiciones al editor y soporte de idiomas. Una característica notable es la capacidad de crear extensiones que analizan código, como linters y herramientas para análisis estático, utilizando el Protocolo de Servidor de Idioma.



Figura 10 – Icono Visual Studio Code

3.3 DOCUMENTACION

Las páginas web pueden y deben diferenciarse unas de las otras, pero todas tienden a contener elementos comunes parecidos, a menos que estén destinadas a mostrar un vídeo, un juego a pantalla completa, que formen parte de un proyecto artístico concreto o, simplemente, que estén mal estructuradas. Estos elementos comunes son:

HEADER (Encabezado): Normalmente formado por una gran franja que cruza la parte superior de la página con un título de encabezamiento y/o un logo. Esta parte suele permanecer invariable mientras navegamos entre las páginas de un mismo sitio web.

NAVIGATION BAR (Barra/menú de navegación): Son los enlaces a las secciones principales del sitio web. Normalmente está formada por un menú con botones, enlaces o pestañas. Al igual que el encabezado, este contenido suele permanecer invariable en las diferentes páginas del sitio. Tener ambos componentes por separado es mejor por motivos de accesibilidad y porque los lectores de pantalla pueden leer mejor ambos elementos si están separados.

FOOTER (Pie de página): Es la parte inferior de la página, generalmente contiene la letra pequeña el copyright o la información de contacto. Es el sitio donde se coloca la información común (al igual que en el encabezado). El pie también se suele usar para propósitos SEO, (Search Engine Optimization) e incluye enlaces de acceso rápido al contenido más popular.

4. DESPLIEGUE E IMPLANTACIÓN

El despliegue de las aplicaciones web son aquellas las actividades que hacen que un sistema de software esté disponible para su uso.

El proceso de implantación general consiste en varias actividades interrelacionadas con posibles transiciones entre ellas. Estas actividades pueden ocurrir en el lado del desarrollador de software o en el lado del consumidor (cliente) o en ambos.

Debido a que cada sistema de software es único, los procesos o procedimientos precisos dentro de cada actividad difícilmente pueden definirse. Por lo tanto, la "implantación" debe interpretarse como un proceso general que debe personalizarse de acuerdo con los requisitos o características específicos a cada proyecto.

4.1 Implantación del sistema y pruebas

4.1.1 Instalación de servidores y aplicaciones.

Para el despliegue de la aplicación se ha decidido montar los siguientes contenedores Docker:

- Un contenedor mysql para almacenar la base de datos.
- Un contenedor wildfly que deberá montar un volumen para almacenar el backend.
- Un contenedor apache(nginx) que deberá montar un volumen para almacenar el frontend.

Todo esto se ha creado en un fichero docker-compose que al lanzarlo nos despliega todos los contenedores.

Este documento lo podemos encontrar en la siguiente ruta ().

MYSQL

Para instalar y desplegar Mysql a través de docker-compose es muy sencillo para ello montamos nuestro archivo con la siguiente configuración:

```
db:
  image: "mysql:latest"
  environment:
    - MYSQL_DATABASE=coolsound
    - MYSQL_USER=mysql
    - MYSQL_PASSWORD=mysql
    - MYSQL_ROOT_PASSWORD=supersecret
  #volumes:
    # - ./workdir/db/init:/docker-entrypoint-initdb.d/ # init database
    #- ./workdir/db/data:/var/lib/mysql/           # data storage
  ports:
    - "3306:3306"
phpmyadmin:
  image: phpmyadmin/phpmyadmin:5.0.1
  restart: always
  environment:
    PMA_HOST: db
  ports:
    - "8085:80"
```

Figura 11 – Docker-compose MySQL

Como podemos observar hemos indicado algunos parámetros para MySQL como son el nombre de la base de datos que se crea por defecto o las credenciales para entrar en MySQL.

Como complemento he añadido la configuración para el despliegue de PhpMyAdmin para tener una interfaz grafica donde manejar base de datos.

Wildfly

Para desplegar wildfly en Docker necesitamos añadir la siguiente configuración a nuestro documento docker-compose.yml:

```
app:
  image: "metz/wildfly"
  environment:
    - WILDFLY_USER=root
    - WILDFLY_PASS=root
    - DB_NAME=sample
    - DB_USER=mysql
    - DB_PASS=mysql
    - DB_URI=db:3306
  depends_on:
    - db
  ports:
    - "8080:8080" # application
    - "9990:9990" # admin console
```

Figura 12- Docker-compose Wildfly

Esto nos desplegará un contenedor wildfly pero antes de entrar para desplegar nuestra aplicación debemos crearnos un usuario administrador para acceder al panel de administración, para eso:

- Accedemos dentro del contener wildfly

```
PS C:\Users\raban> docker exec -it 18ca394123df bash
```

Figura 13- Comando para acceder a contener Docker

- Debemos acceder al fichero add-user.sh para crearnos un usuario dentro de la siguiente ruta:

```
[jboss@18ca394123df ~]$ cd wildfly/
[jboss@18ca394123df wildfly]$ cd bin/
[jboss@18ca394123df bin]$ ls
add-user.bat          common.bat          elytron-tool.sh     jdr.sh              vault.sh
add-user.properties  common.ps1          jboss-cli-logging.properties  launcher.jar        wildfly-elytron-tool.jar
add-user.ps1          common.sh           jboss-cli.bat      product.conf        wsconsume.bat
add-user.sh           domain.bat          jboss-cli.ps1      standalone.bat       wsconsume.ps1
appclient.bat         domain.conf         jboss-cli.sh       standalone.conf      wsconsume.sh
appclient.conf        domain.conf.bat     jboss-cli.xml      standalone.conf.bat  wsprovide.bat
appclient.conf.bat    domain.conf.ps1     jconsole.bat       standalone.conf.ps1  wsprovide.ps1
appclient.conf.ps1    domain.ps1          jconsole.ps1       standalone.ps1       wsprovide.sh
appclient.ps1         domain.sh           jconsole.sh        standalone.sh
appclient.sh          elytron-tool.bat   jdr.bat            vault.bat
client               elytron-tool.ps1   jdr.ps1            vault.ps1
```

Figura 14- Archivo para añadir documentos

Una vez creado nuestro usuario administrador podremos acceder al panel de administración de wildfly, donde podremos desplegar el .war de nuestra aplicación.

Para desplegar nuestra aplicación:

1- Accedemos al panel de administración y seleccionamos start en la sección de deployments.

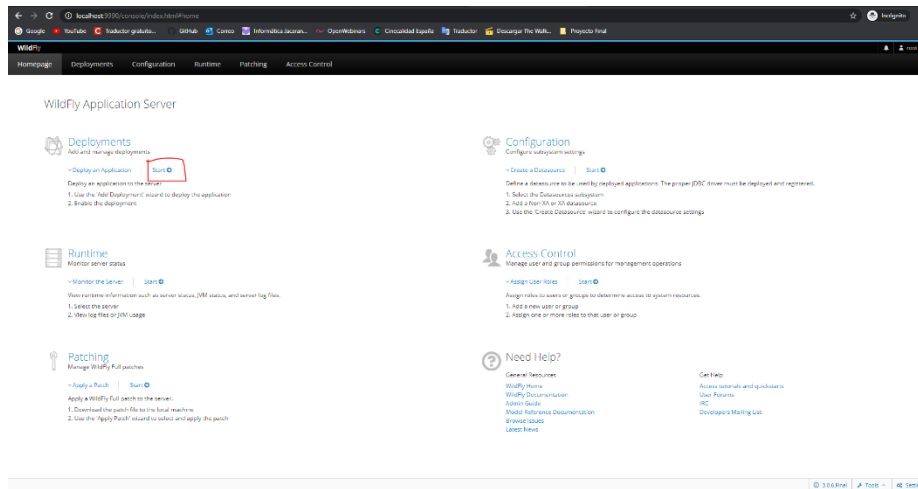


Figura 15- Panel de administración wildfly

2- En el desplegable de la pantalla deployments, seleccionamos la opción upload deployment.

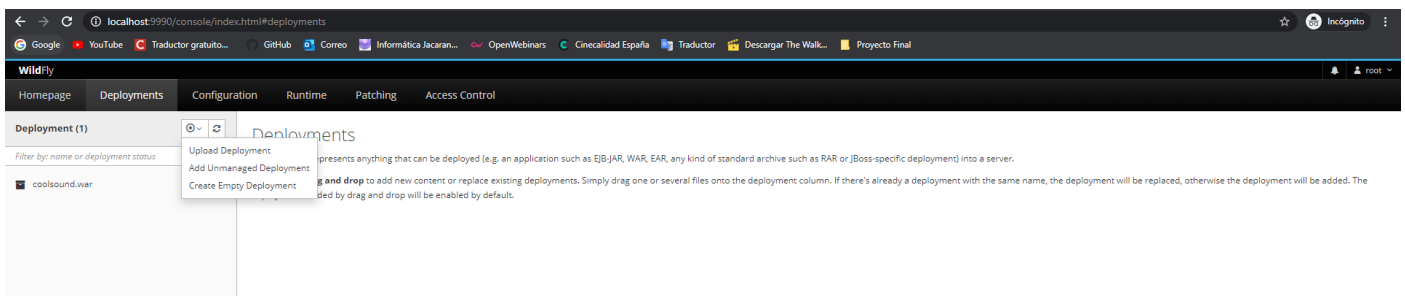


Figura 16- Desplegar proyecto Wildfly

3- En el desplegable que nos sale añadimos nuestro .war y le damos deploy

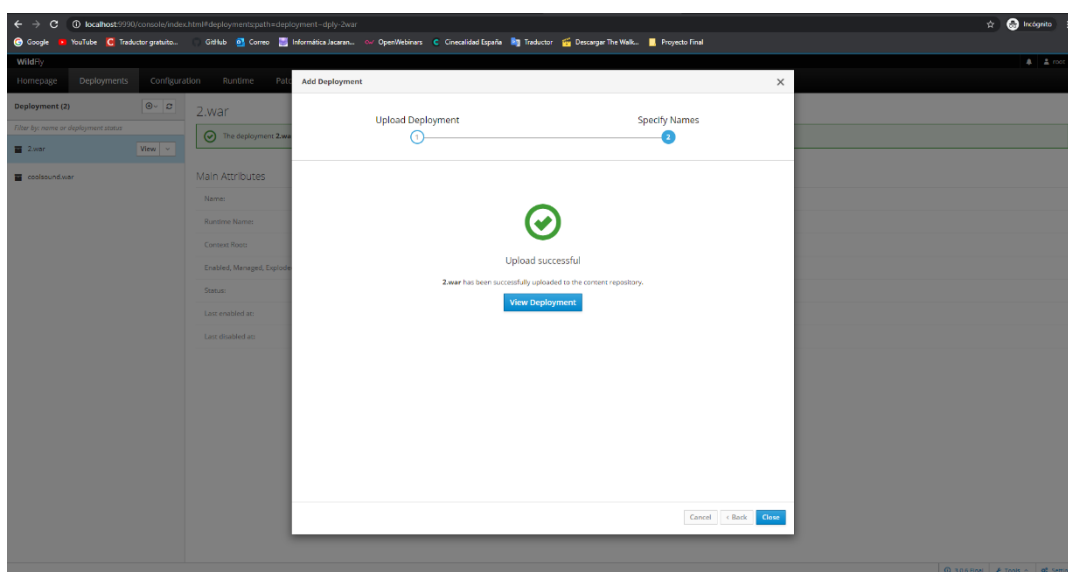


Figura 17- Proyecto desplegado Wildfly

Nginx

Para poder desplegar el front de nuestra aplicación, hemos añadido a nuestro Docker-compose.yml la siguiente información:

```
frontend:
  build:
    context: .
    dockerfile: front.Dockerfile
  restart: always
  ports:
    - "8082:80"
    - "8083:443"
```

Figura 18- Configuración Nginx

Esta configuración contiene otra relación con un archivo Dockerfile, este Dockerfile contiene la información para relacionar nuestro proyecto back compilado con nginx.

La configuración del Dockerfile es la siguiente:

```
C: > Users > raban > OneDrive > Escritorio > PROYECTO FINAL > REPOSITORIO > CoolSound > DOCK
1 FROM nginx:1.17.1-alpine
2 COPY nginx.conf /etc/nginx/nginx.conf
3 COPY angularWeb /usr/share/nginx/html
4
```

Figura 19- Dockerfile

Una vez configurado nuestro archivo Docker-compose.yml solo bastará con escribir el siguiente comando en la ubicación del Docker-compose para ejecutarlo y nos lance automáticamente todos los contenedores.

4.1.2 Mockups

En esa parte iré describiendo paso a paso cuales son las diferentes pantallas que componen la web.

MENÚ PRINCIPAL

Según el rol que tengamos dentro de la aplicación nos mostrara un Menú Principal u otro.

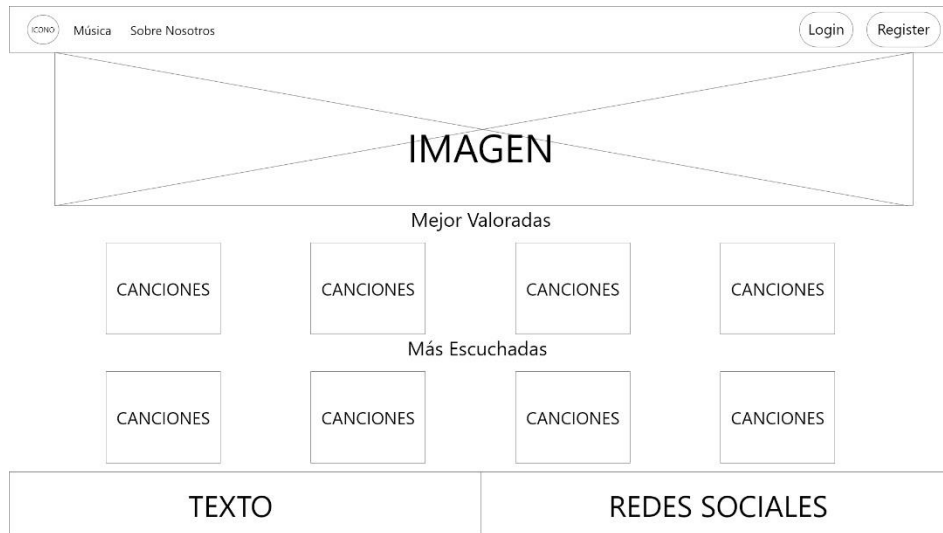


Figura 20- Mockup Menú principal

Si dentro de la aplicación no estamos logueados nos aparecerá esta pantalla donde podremos acceder solo a las pantallas de Música, Sobre Nosotros, Login y Register.

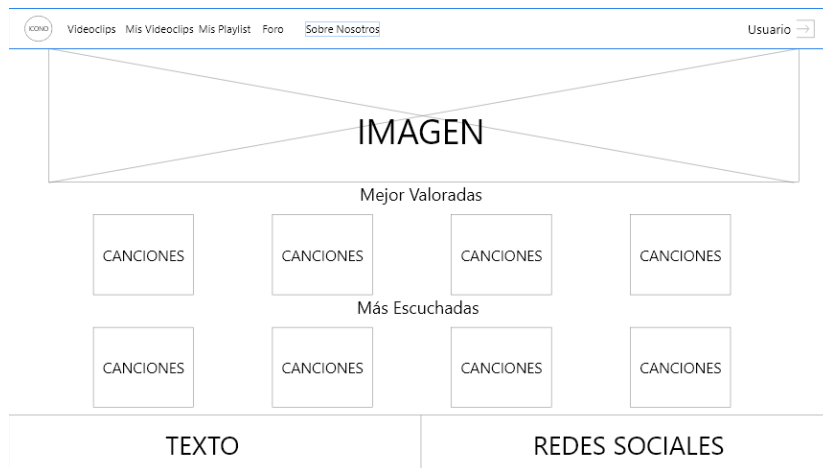


Figura 21- Mockup Menú Principal 2

Si nos logueamos nos aparecerá este Menú Principal, donde podremos acceder a las pantallas de Música, Mi Música, Mis Playlist, Foro y Sobre Nosotros. Como ya hemos hecho Login en la aplicación, los botones de Login y Register que aparecían a la derecha de la barra de navegación se sustituyen por el nombre de usuario seguido de un botón para desloguearse de la aplicación.

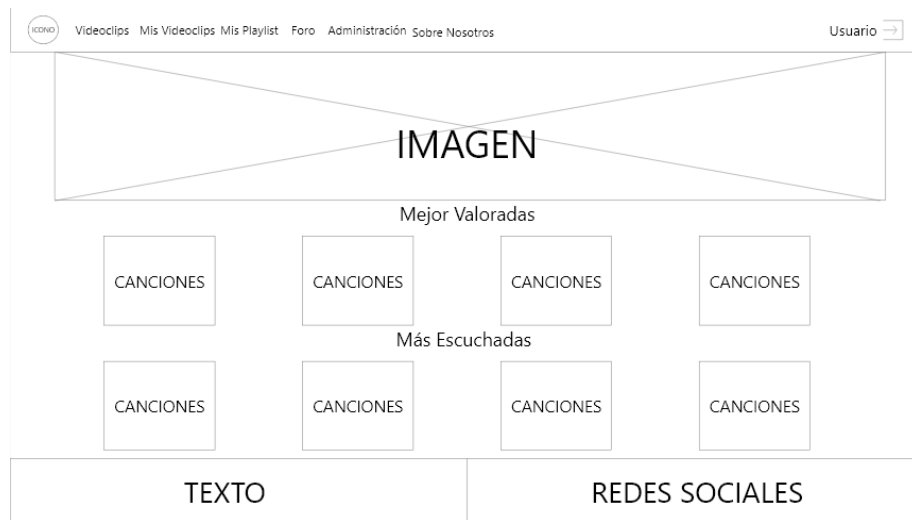


Figura 22- Mockup Menú Principal 3

Si entramos en la aplicación logueados como administrador, el menú principal cambia, es exactamente el mismo, con la única excepción que podemos acceder a la pantalla de Administración.

El contenido de las pantallas es exactamente el mismo, una imagen que ira con la temática de la web, esta imagen podrá ser cambiada por el administrador a través de los paneles de administración de la web. Luego en dos bloques aparecerán dos listas de 4 canciones mejor valoradas y 4 canciones más escuchadas. Cada canción será interactiva de forma que cuando pongamos el cursor encima de la canción podremos escucharla.

LOGIN Y REGISTER

Dentro de la pantalla de Login nos aparecerá un formulario por el cual podemos acceder a la aplicación, el usuario tendrá varios botones con diferentes funcionalidades.

El botón de Login nos da acceso a la aplicación con las credenciales introducidos, si las credenciales son incorrectas nos informara de ello y podremos rellenar el Login de nuevo.

Tendremos el botón registrarse que nos llevara a la pantalla de Registrarse donde podremos dar de alta a nuevos usuarios, y por último si ya estamos registrados, pero no nos acordamos de nuestra contraseña, tendremos un enlace a la pantalla de recuperar nuestra contraseña.



Figura 23- Mockup Login

La pantalla de Registrarse es bastante simple, esta pantalla contiene un formulario para poder crear un nuevo usuario para acceder a la aplicación, este formulario tendrá validaciones por si algún dato es introducido incorrectamente.

ICONO Música Sobre Nosotros

REGISTRATE

Usuario:
Contraseña:
Repite la contraseña:
Email:
REGISTRARSE

TEXTO REDES SOCIALES

Figura 24- Mockup Register

MUSICA

ICONO Videoclips Mis Videoclips Mis Playlist Foro Sobre Nosotros Usuario →

VIDEOCLIP VIDEOCLIP VIDEOCLIP VIDEOCLIP VIDEOCLIP

VIDEOCLIP VIDEOCLIP VIDEOCLIP VIDEOCLIP VIDEOCLIP

TEXTO REDES SOCIALES

Figura 25- Mockup Musica

En esta pantalla se muestran todos los videoclips que hay en la web paginadas para no saturar la misma pantalla. Estos videoclips serán interactivos, pasando el cursor por encima de ellas podemos clicar en ella y ver/escuchar dicho videoclip. Debajo de cada videoclip aparecerá un pequeño botón para acceder a ella, dicho botón nos llevará a la pantalla donde podremos todas las especificaciones del videoclip, escucharla, añadirla a nuestras playlists e incluso valorarla.

La barra de navegación cambia según el rol que tengamos en la web.

MUSICA/CANCION

The mockup shows a web interface for a music application. At the top is a navigation bar with a logo and links: Videoclips, Mis Videoclips, Mis Playlist, Foro, and Sobre Nosotros. On the right of the navigation bar is a 'Usuario' link with a right-pointing arrow. The main content area is divided into two columns. The left column contains a placeholder for an image labeled 'IMAGEN' and a video player placeholder labeled 'VIDEOCLIP' with a play button icon. The right column contains form fields for 'Título:' and 'Género:', followed by a dropdown menu for 'Añadir A Playlist' and an 'Añadir' button, a dropdown menu for 'Evaluar la Canción' and an 'Evaluar' button, and a text input for 'Modificar Portada' with a 'Modificar' button. Below these is a 'COMENTARIOS' section with three text input fields, each preceded by a radio button. At the bottom of the screen are two tabs: 'TEXTO' and 'REDES SOCIALES'.

Figura 26- Mockup Musica/Cancion

En esta pantalla como vemos en el diseño, aparecerá la portada del videoclip, el título y el género.

Aparecerá un reproductor interactivo con el cual el usuario podrá ver/escuchar el videoclip.

A través de unos desplegaes el usuario podrá añadir dicho videoclip a cualquiera de sus playlist. Con otro desplegable el usuario podrá evaluar la canción del 1 al 10.

En la parte inferior de la pantalla aparece una zona de comentarios en la cual cualquier usuario de la aplicación podrá poner un comentario acerca la canción.

SOBRE NOSOTROS

Ícono Música Mi Música Mis Playlist Foro Sobre Nosotros Usuario →

IMAGEN

TEXTO

Asunto:

Mensaje:

ENVIAR

TEXTO REDES SOCIALES

Figura 27- Mockup Sobre Nosotros

En esta pantalla a la cual el usuario podrá acceder a través de la barra de navegación, nos aparecerá una foto estática, un pequeño texto que describe la web y un formulario para contactar con el administrador de la web, dicho formulario estará controlado para la validación de los campos.

MIS VIDEOCLIPS

Ícono Videoclips Mis Videoclips Mis Playlist Foro Sobre Nosotros Usuario →

VIDEOCLIP VIDEOCLIP VIDEOCLIP VIDEOCLIP VIDEOCLIP

VIDEOCLIP VIDEOCLIP VIDEOCLIP VIDEOCLIP VIDEOCLIP

Añadir Nuevo Videoclip

TEXTO REDES SOCIALES

Figura 28- Mockup Mis Videoclips

En esta pantalla aparecerá todos los videoclips paginados que haya creado el usuario en la web. En la parte inferior hay un botón que nos llevara a la otra pantalla para poder añadir una nueva canción a nuestra web.

NUEVO VIDEOCLIP

Nuevo Videoclip

Titulo:

Género

URL de YouTube del Videoclip:

Añadir

TEXTO

REDES SOCIALES

Figura 29- Mockup Nuevo Videoclip

Esta pantalla contiene un formulario para poder añadir un nuevo videoclip a la web, el usuario tendría que introducir el título, en un desplegable deberá seleccionar el género del videoclip, y la url donde se aloja el videoclip.

El formulario tendrá validaciones para controlar que los campos del formulario se hayan introducido correctamente.

Al clicar en el botón añadir nos devolverá a la pantalla de Mi Música con la nueva canción añadida a nuestra lista de canciones.

PLAYLISTS

Playlist

Escuchar Borrar

Playlist

Escuchar Borrar

Playlist

Escuchar Borrar

Texto

Crear Nueva Playlist

TEXTO

REDES SOCIALES

Figura 30- Mockup Playlists

Esta pantalla contendrá la lista de las playlist que hayamos creado, si hacemos click en escuchar nos llevará a otra pantalla en la cual podremos ver nuestra playlist desplegada y escucharla. En la parte derecha de cada playlist tendremos un botón para borrar dicha playlist.

En la parte inferior de la pantalla tendremos un campo donde podemos crear nuevas playlists.

PLAYLISTS/DETALLE



Figura 31- Mockup Playlist/Detalle

Cuando accedamos a esta pantalla nos aparecerá un reproductor de videoclips interactivo, nos aparecerá la lista de canciones añadidas a la playlist, podremos avanzar de canción, retroceder y controlar el volumen.

FORO



Figura 32- Mockup Foro

La pantalla del foro cambiara según como estemos logueados en la aplicación, si estamos logueados como usuario podremos acceder a los hilos de discusión y crear nuevos hilos. Si estamos logueados como administradores podemos hacer todo lo anterior y borrarlos.

FORO/HILO



The mockup shows a navigation bar at the top with a circular icon labeled 'ICONO' and links for 'Música', 'Mi Música', 'Foro', 'Sobre Nosotros', and 'Administración'. On the right is a 'Usuario' link with a right-pointing arrow. Below the navigation bar is a container for thread replies. It contains two existing replies, each with a circular profile icon placeholder, a text input field labeled 'Texto', and a 'Borrar respuesta' button. At the bottom of this container is a 'Añadir una respuesta' button. Below the replies container is a footer bar with two buttons: 'TEXTO' and 'REDES SOCIALES'.

Figura 33- Mockup Foro/Hilo

Al acceder a esta pantalla nos aparecerán todos los comentarios que hayan puesto los usuarios en este hilo de discusión. También podrán crear nuevas respuestas.

Si estamos logueados como administrador podremos borrar comentarios que sean inapropiados.

CONFIGURACIÓN PERFIL



The mockup shows a navigation bar at the top with a circular icon labeled 'ICONO' and links for 'Música', 'Mi Música', 'Mis Playlist', 'Foro', and 'Sobre Nosotros'. On the right is a 'Usuario' link with a right-pointing arrow. Below the navigation bar is the profile configuration area. On the left is a circular profile picture placeholder with a diagonal 'X' and a 'Cambiar Imagen' button below it. On the right are input fields for 'Usuario:' and 'Email:', and a 'Cambiar Contraseña' button. Below this area is a footer bar with two buttons: 'TEXTO' and 'REDES SOCIALES'.

Figura 34- Mockup Configuración Perfil

Esta es una pantalla en la cual los usuarios podrán modificar su información, tal que su imagen y contraseña.

ADMINISTRACIÓN



Figura 35- Mockup Administración

A esta pantalla solo podrán acceder los administradores, en esta pantalla podrán acceder a otras pantallas donde pueden tener acceso a la administración de usuarios y videoclips.

5. MANTENIMIENTO

Facilidades de mantenimiento

El mantenimiento básico de la aplicación se llevará a cabo por el administrador. Sin embargo, cualquier cambio que se desee introducir y requiera de la modificación de la base de datos, así como implementar una nueva funcionalidad, deberá ser llevado a cabo necesariamente por el desarrollador de la web, y en sus diferentes modalidades de mantenimiento predictivo, preventivo, correctivo y evolutivo.

Portabilidad

La aplicación ha sido diseñada con tecnología libre, luego podrá ser soportada por cualquier plataforma y sistema operativo. Por lo mismo, se podrá acceder a ella desde cualquier periférico que esté conectado a internet.



6. CONCLUSIONES

6.1 Conclusiones científicas

Como todo proyecto técnico, el principal objetivo es satisfacer las necesidades del cliente, en este caso, SE pedía el desarrollo de un producto capaz de buscar, crear y reproducir videoclips. Para lograr dicho objetivo, se ha recurrido a una planificación validada por el tutor del proyecto. Los hitos marcados se han cumplido en el plazo previsto por lo que podemos afirmar que no ha existido una desviación de tiempo en la obtención del producto. Por otro lado, el desarrollo del producto ha implicado el aprendizaje y/o consolidación de conocimientos sobre las tecnologías vinculadas a Java y Angular. También se han puesto en práctica conocimientos teóricos como orientación a objetos. El producto final ha sido la consecuencia lógica del seguimiento de todas las tareas marcadas en la planificación inicial.

6.2 Conclusiones personales

Cabe destacar que, por tratarse de un prototipo, la funcionalidad de la aplicación web puede no ser completa. Muchas mejoras y ampliaciones, quedan fuera del alcance de este proyecto por su duración y extensión. Sin embargo, las funciones descritas en la especificación de requisitos están implementadas.

Desde el punto de vista personal, el proyecto ha constituido un gran reto para mí. El proyecto final es la culminación de un largo trabajo, conocimientos adquiridos durante dos años que se deben poner en práctica. Estos últimos meses han implicado un gran esfuerzo personal; días enteros de duro trabajo, largas noches a base de café, pocas horas de sueño... pero todo ello ha merecido la pena con creces. Extraigo una enriquecedora experiencia de esta asignatura.

7. LINEAS FUTURAS

Las líneas futuras de desarrollo son las siguientes:

- Incluir Migas de Pan.
- Configurar un Filtro de Búsqueda en la pantalla de Videoclips.
- Configurar la gestión de Videoclips como ficheros que se almacenan en el servidor.
- Configurar el despliegue automático de los contenedores a través de Jenkins.

8. BIBLIOGRAFIA

Stackoverflow

<http://stackoverflow.com/>

GitHub

<https://github.com/>

Openwebinars

<https://openwebinars.net/>

Udemy

<https://www.udemy.com/>

Wikipedia

<https://es.wikipedia.org/>