



MEMORIA FINAL DEL PROYECTO



CICLO FORMATIVO DE GRADO SUPERIOR

DESARROLLO DE APLICACIONES WEB

AUTORES

**JORGE DEL CAMPO PERIS
GUILLERMO MUELA LÓPEZ
SARA MERCEDES GÓMEZ GARRIGUES**

TUTOR Y COORDINADOR
ÁLVARO GONZÁLEZ SOTILLO



DEDICATORIAS Y AGRADECIMIENTOS

Gracias a todos y cada uno de los profesores que a lo largo de estos años nos han inculcado sus conocimientos y la pasión por el desarrollo web y que, siempre que lo hemos necesitado, han estado ahí para echarnos una mano.

Y agradecer especialmente a nuestro tutor y coordinador Álvaro, por transmitirnos su ilusión por la programación y haberla podido plasmar en Melomify.

Esperemos que disfrutéis con Melomify tanto como nosotros lo hemos hecho desarrollándola.



ÍNDICE

- 1.** Introducción
- 2.** Alcance del proyecto y análisis previo
- 3.** Estudio de viabilidad
 - 3.1.** Estado actual del sistema
 - 3.2.** Resumen de requisitos del cliente
 - 3.3.** Posibles soluciones
 - 3.4.** Solución elegida
 - 3.5.** Planificación temporal de las tareas del proyecto
 - 3.6.** Planificación de los recursos a utilizar
- 4.** Análisis
 - 4.1.** Requisitos funcionales
 - 4.2.** Requisitos no funcionales
- 5.** Diseño
 - 5.1.** Herramientas y tecnologías utilizadas
- 6.** Implementación
 - 6.1.** Implementación del modelo de datos
 - 6.2.** Carga de datos
 - 6.3.** Configuraciones realizadas en el sistema
 - 6.4.** Implementaciones de código realizadas
- 7.** Pruebas
 - 7.1.** Testing con Junit
- 8.** Explotación
 - 8.1.** Planificación
 - 8.2.** Preparación para el cambio
 - 8.3.** Manual de usuario
- 9.** Definición de procedimientos de control y evaluación
- 10.** Conclusiones
- 11.** Fuentes
 - 11.1.** Legislación
 - 11.2.** Bibliografía



1. INTRODUCCIÓN

El presente documento recoge el trabajo realizado para el módulo de Proyecto del CFGS en Desarrollo de Aplicaciones Web y que recibe el nombre de **Melomify**.

Este módulo profesional complementa la formación establecida para el resto de los módulos que integran el título en las funciones de análisis del contexto, diseño o desarrollo del proyecto y organización de la ejecución.

2. ALCANCE DEL PROYECTO Y ANÁLISIS PREVIO

Melomify se desarrolla con el objetivo de crear una red social para melómanos y amantes de la música que actúe de conexión directa entre los músicos y sus seguidores.

Los grupos y músicos pueden subir sus temas a Melomify junto con un pequeño comentario. Es lo que se conoce como Melom. A su vez, los melómanos van a generar una gran comunidad aportando sus comentarios y opiniones acerca de los Meloms.

Se pretende crear un marco único en el que los usuarios puedan descubrir y escuchar la música que aman y expresar sus sentimientos a sus artistas favoritos. Sin intermediarios.

También permite que los grupos que estén empezando en el mundo de la música se den a conocer al gran público. Los artistas más reconocidos pueden compartir extractos de sus próximos álbumes y canciones y saber qué gusta y qué no a sus fans y, de esta forma, publicar el disco que sus seguidores piden y esperan.

De esta forma, el desarrollo de este proyecto se ha llevado a cabo en varias fases: estudio de viabilidad, análisis, diseño, implementación y pruebas, y explotación o ejecución.

A continuación se detallan las actividades/tareas/procedimientos de cada una de estas fases.

3. ESTUDIO DE VIABILIDAD

En esta fase se considera si el proyecto se puede realizar teniendo en cuenta las circunstancias internas y externas de la empresa, las diferentes soluciones posibles y los recursos de los cuales se dispone.

Para ello se hace una valoración del estado actual del sistema y de los requisitos del cliente, se presentará un estudio de soluciones alternativas y la solución elegida por el cliente.

3.1. Estado actual del sistema

En la actualidad, la mayor parte de las aplicaciones web y plataformas online relacionadas con el mundo de la música tienen un carácter comercial mediante la venta de álbumes y canciones o por la concesión de servicios de streaming.



3.2. Resumen de requisitos del cliente

El cliente solicita el desarrollo de una aplicación no comercial enfocada en la creación de contenido y en compartirlo para fomentar la comunidad musical.

3.3. Posibles soluciones

Hay una herramienta similar en el mercado de las aplicaciones musicales. Se trata de SoundCloud. Twitter también permite algunas de las funcionalidades de Melomify.

3.4. Solución elegida

Se ha optado por una solución intermedia entre la comunidad musical de SoundCloud y la inmediatez e interfaz propias de Twitter. Así nace Melomify.

3.5. Planificación temporal de las tareas del proyecto

Para las historias de usuario previstas y mostradas más adelante, se han estimado necesarias tres personas para la realización del proyecto en una primera release.

3.6. Planificación de los recursos a utilizar

Para solventar los problemas que puede plantear el proyecto en un futuro de cara a actualizaciones y ampliaciones de contenido y funcionalidades, sería necesaria la contratación de más personal especializado en marketing y en legislación online.

4. ANÁLISIS

En esta fase se establecerán los requisitos del sistema que van a quedar recogidos de forma gráfica mediante un **diagrama de casos de uso** para los dos tipos de usuario de Melomify: melómanos y administradores.

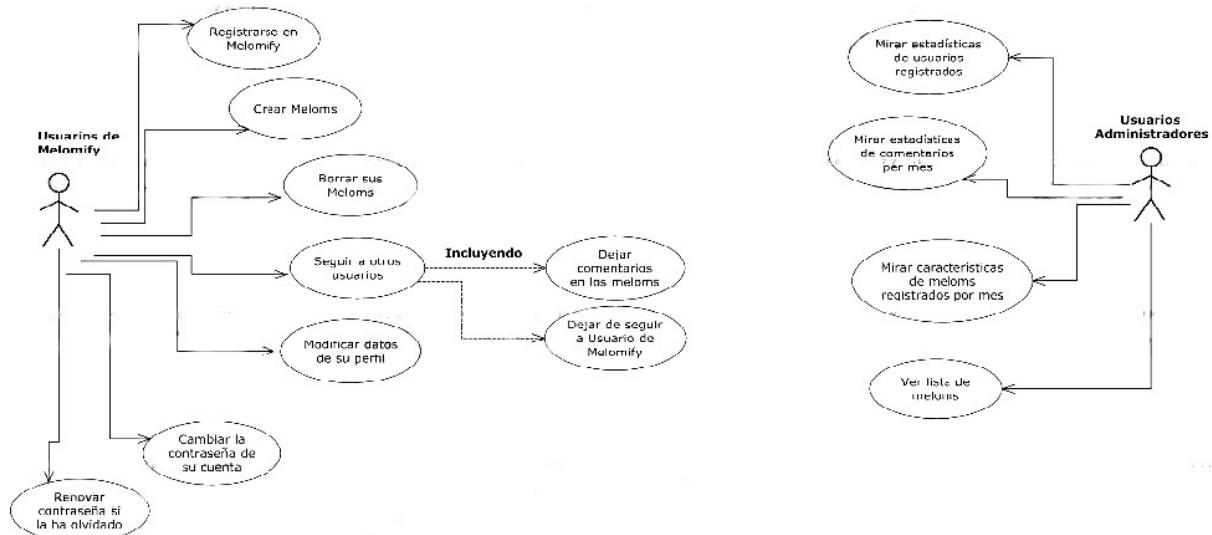


Ilustración 1 - Diagrama de casos de uso



A su vez, también se puede visualizar el análisis del proyecto gracias al **modelo de dominio** expuesto a continuación:

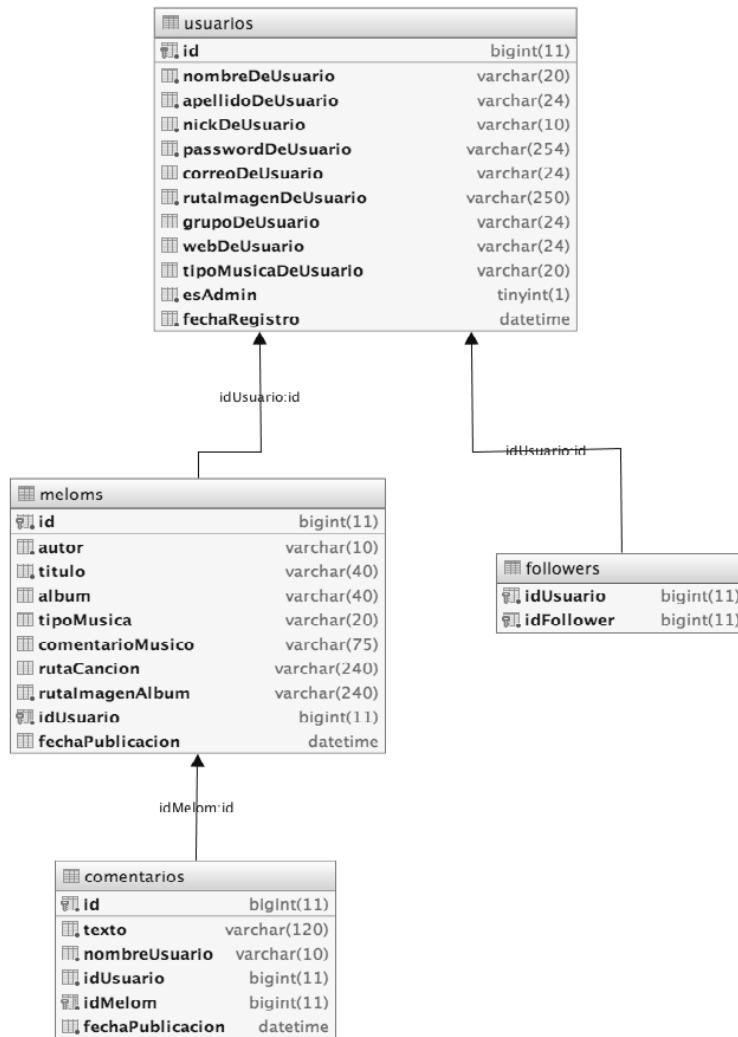


Ilustración 2 - Modelo de dominio

4.1. Requisitos funcionales

La aplicación tiene que ser capaz de gestionar los dos tipos de usuarios de Melomify: administradores y melómanos, según los casos de uso expuestos anteriormente para lograr la comunicación entre todos los miembros de la comunidad de Melomify.

4.2. Requisitos no funcionales

Se pretende que Melomify cuente con un diseño minimalista y accesible para todo tipo de usuarios y, así, conseguir que sea fácil de utilizar. También se busca una aplicación robusta y segura, al igual que con el hecho de que pueda ser ampliada con nuevas funcionalidades en el futuro si son requeridas por el cliente y la comunidad de Melomify.



5. DISEÑO

En esta fase se realiza una aproximación al diseño tecnológico de la aplicación.

5.1. Herramientas y tecnologías utilizadas

Para el desarrollo de Melomify se ha escogido el lenguaje de programación **Java** y Java Platform, Enterprise Edition o **Java EE** como plataforma de programación. Esta elección ha permitido crear una aplicación empresarial portable, escalable y robusta.

Otra de las ventajas que aporta Java EE es su coste nulo, pues su implementación se puede descargar de Oracle Corporation de manera gratuita, lo cual se ajusta a los requisitos iniciales de bajo coste económico para Melomify. Y, además, dispone de multitud de herramientas de código abierto que han facilitado el desarrollo de la aplicación, como se expondrán a continuación.

Para la gestión y construcción del proyecto se ha utilizado **Apache Maven** de Apache Software Foundation, pues puede descargar de forma dinámica plugins de un repositorio y facilitar así la accesibilidad y distribución de funcionalidades de código abierto. De los mismos desarrolladores se ha elegido como servidor web **Apache Tomcat** 8.0.30.

Como entorno de desarrollo integrado o IDE, se ha optado por **Eclipse Mars** 4.5.2 de Eclipse Foundation en plataformas Windows y por **IntelliJ IDEA** en OS X.

El sistema de gestión de bases de datos relacional elegido es **MySQL** 5.7.12 de Oracle Corporation, y para administrarla se han utilizado herramientas como **MySQL Workbench** 6.3.6 y **phpMyAdmin** 4.5.3.1.

Con respecto al front-end, se ha empleado una combinación de diferentes tecnologías: **Bootstrap** v3.3.6 de Twitter, **JavaServer Faces (JSF)** 2.2 de Sun Microsystems y una de sus librerías de componentes más conocidas como es **PrimeFaces** 5.3.

Bootstrap ha permitido recrear el estilo minimalista, funcional y responsive para plataformas móviles que demandaba Melomify gracias a los componentes de **Javascript** basados en la librería **jQuery** y a la hoja de estilos **CSS** que incorpora.

En el back-end se ha implementado un diseño **MVC** con el uso de anotaciones JSF para la Inyección de Dependencias y la administración de Beans.

Con el objetivo de asegurar la mayor eficiencia y confiabilidad en el mantenimiento del código fuente de la aplicación, se ha utilizado **Git** de Linus Torvalds como software de control de versiones con una rama Master y ramas Development para cada uno de los integrantes del proyecto. El cliente gratuito de Git **SourceTree** 1.8.3 de Atlassian ha permitido controlar las versiones del código fuente de manera muy accesible y visual.

A la hora de alojar el código fuente del proyecto y como plataforma de desarrollo colaborativo, junto con Git, se ha empleado una cuenta gratuita y pública de **GitHub**, desde la que se puede consultar cómo ha trabajado el grupo de desarrolladores de la aplicación.

Para facilitar el mapeo de atributos entre la base de datos relacional y el modelo de objetos de la aplicación en Java, se ha optado por la herramienta **Hibernate** 4.3.1, que permite establecer esas relaciones mediante archivos declarativos XML o anotaciones en los Beans.

Las pruebas unitarias se han realizado con un conjunto de bibliotecas conocidas como **JUnit**.



Permite ejecutar clases Java de forma controlada para evaluar su comportamiento.

En cuanto a la comunicación entre los integrantes del proyecto, se han utilizado multitud de tecnologías para cada etapa del desarrollo.

En un primer momento, se utilizó **Trello** para unificar en un único marco todas las ideas y propuestas iniciales para el proyecto. Al entrar en fase de producción, se utilizó **Taiga** para gestionar las historias de usuario, tareas y sprints. Para las reuniones diarias o daily meetings se emplearon las plataformas **TeamSpeak 3** y **Discord**, además de **WhatsApp** para el seguimiento 24/7 del progreso de cada uno de los integrantes del proyecto.

Todas estas tecnologías se han empleado bajo un marco de desarrollo ágil Scrum.

6. IMPLEMENTACIÓN

Partiendo del diseño, en esta fase se construye el sistema.

6.1. Implementación del modelo de datos

Se definen las siguientes tablas para una base de datos MySQL:

```
comentarios
  id BIGINT(11)
  texto VARCHAR(120)
  nombreUsuario VARCHAR(10)
  idUsuario BIGINT(11)
  idMelom BIGINT(11)
  fechaPublicacion DATETIME
  PRIMARY (id)
  idMelom (idMelom) → meloms (id)
  idMelom_idx (idMelom)

followers
  idUsuario BIGINT(11)
  idFollower BIGINT(11)
  PRIMARY (idUsuario, idFollower)
  followers_usuarios_id_fk (idUsuario) → usuarios (id)

hibernate_sequence
  next_val BIGINT(20)

meloms
  id BIGINT(11)
  autor VARCHAR(10)
  titulo VARCHAR(40)
  album VARCHAR(40)
  tipoMusica VARCHAR(20)
  comentarioMusico VARCHAR(75)
  rutaCancion VARCHAR(240)
  rutalmagenAlbum VARCHAR(240)
  idUsuario BIGINT(11)
  fechaPublicacion DATETIME
  PRIMARY (id)
  meloms_usuarios_id_fk (idUsuario) → usuarios (id)
  idUsuario_idx (idUsuario)

usuarios
  id BIGINT(11)
  nombreDeUsuario VARCHAR(20)
  apellidoDeUsuario VARCHAR(24)
  nickDeUsuario VARCHAR(10)
  passwordDeUsuario VARCHAR(254)
  correoDeUsuario VARCHAR(24)
  rutalmagenDeUsuario VARCHAR(250)
  grupoDeUsuario VARCHAR(24)
  webDeUsuario VARCHAR(24)
  tipoMusicaDeUsuario VARCHAR(20)
  esAdmin TINYINT(1)
  fechaRegistro DATETIME
  PRIMARY (id)
```

Ilustración 3 - Tablas de la base de datos MySQL



6.2. Carga de datos

Se inicia Tomcat y se mueve el .war a la carpeta de webapps para su posterior **despliegue**.

Una vez desplegada la aplicación, se deben ejecutar los archivos .sql incluidos en el .war en una base de datos MySQL.

6.3. Configuraciones realizadas en el sistema

Es necesario disponer de Java 8 y Tomcat 8 para la ejecución, Maven para la descarga de repositorios y un navegador Google Chrome v. 48 o superior.

Además, se debe configurar el archivo **hibernate.cfg.xml** según las necesidades propias de la base de datos.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <property name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
        <property name="hibernate.connection.url"></property>
        <property name="hibernate.connection.username"></property>
        <property name="hibernate.connection.password"></property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <property name="cache.provider_class">org.hibernate.cache.NoCacheProvider</property>
        <property name="show_sql">true</property>

        <mapping class="app.beans.UsuarioBean" />
        <mapping class="app.beans.ComentarioBean" />
        <mapping class="app.beans.MelomBean" />
        <mapping class="app.beans.FollowerBean" />
    </session-factory>
</hibernate-configuration>
```

Ilustración 4 - Configuración del archivo hibernate.cfg.xml

6.4. Implementaciones de código realizadas

El código fuente de Melomify se entrega junto a la presente documentación en formato CD recogido tanto en un archivo .war como en un .zip.

7. PRUEBAS

Son muchas las pruebas que pueden realizarse en un proyecto para eliminar los posibles errores y garantizar su correcto funcionamiento. Los casos de prueba establecen las condiciones/variables que permitirán determinar si los requisitos establecidos se cumplen o no.

A continuación se detallan algunos de los casos de prueba que se ejecutarán para comprobar la correcta construcción de este proyecto.

7.1. Testing con JUnit:

Para esta aplicación se ha elaborado un test con JUnit que permite comprobar el comportamiento de ciertas partes del código.



Las siguientes clases están cubiertas con **test unitarios**:

Beans (Usuario, Melom, Follower, Comentario)

Builders (Usuario, Melom, Follower, Comentario)

Las siguientes clases están cubiertas con **test de integración**:

DAO's (Usuario, Melom, Comentario)

Los test de integración usan una base de datos embebida en memoria (H2) gestionada por Hibernate que es una réplica exacta de la base de datos del cliente.

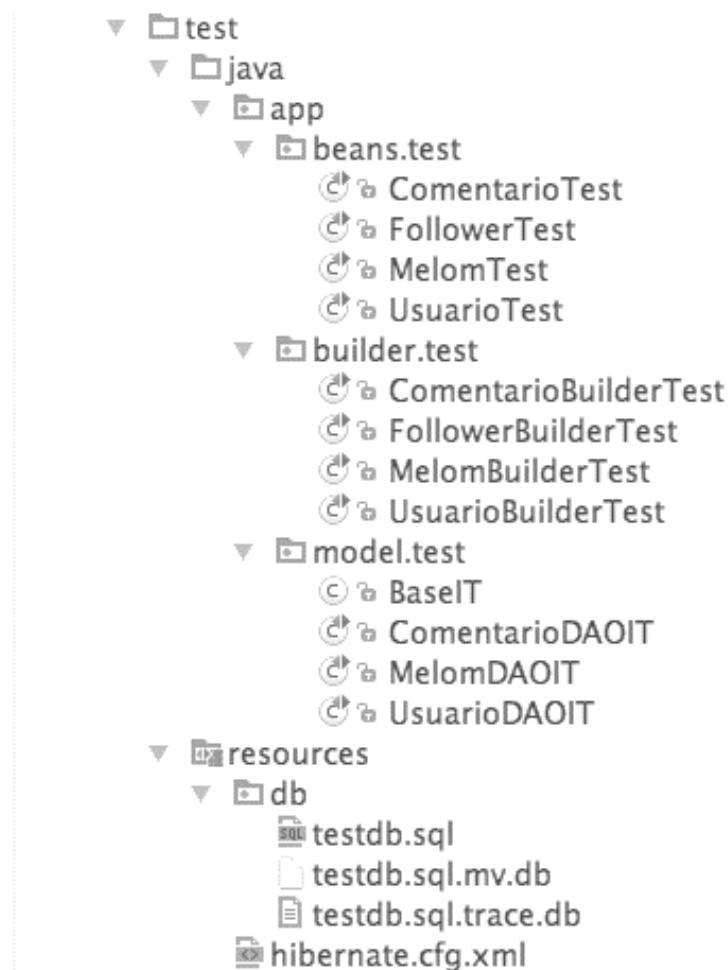


Ilustración 5 - Estructura de los tests de Melomify



Se emplea Maven para ejecutar los test mediante el comando **mvn test**, obteniendo el siguiente resultado:

```
MacBookProdegmuela:app guillermo$ mvn test
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for iesrey-daw:Melomify:war:RELEASE
[WARNING] 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-compiler-plugin is missing. @ line 169, column 12
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----
[INFO] Building Melomify RELEASE
[INFO] -----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Melomify ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 5 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Melomify ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Melomify ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 4 resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ Melomify ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Melomify ---
[INFO] Surefire report directory: /Users/guillermo/Documents/git/app/target/surefire-reports

-----
T E S T S
-----
Running app.beans.test.ComentarioTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.132 sec
Running app.beans.test.FollowerTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.008 sec
Running app.beans.test.MelomTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.013 sec
Running app.beans.test.UsuarioTest
Tests run: 18, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.062 sec
Running app.builder.test.ComentarioBuilderTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.003 sec
Running app.builder.test.FollowerBuilderTest
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.026 sec
Running app.builder.test.MelomBuilderTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec
Running app.builder.test.UsuarioBuilderTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.005 sec

Results :

Tests run: 30, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.690 s
[INFO] Finished at: 2016-06-13T04:18:54+02:00
[INFO] Final Memory: 11M/155M
```

Ilustración 6 - Realización de tests con JUnit



8. EXPLOTACIÓN

La implantación es la fase más crítica del proyecto ya que el sistema entra en producción, es decir opera en un entorno real, con usuarios reales.

8.1. Planificación

Para el despliegue de la aplicación se va a necesitar un hosting capaz de albergar la magnitud del proyecto, puesto que va a tener un gran número de usuarios subiendo contenido multimedia.

8.2. Preparación para el cambio

Se necesita disponer de la autorización del autor de la canción para poder subirla a la aplicación mediante su concesión de manera gratuita.

Es posible que algunos usuarios comentan infracciones con respecto a los derechos de autor, por utilizar canciones sin el consentimiento del artista.

8.3. Manual de usuario

Glosario de términos y de su significado:

TIMELINE: lugar donde se colocan, por orden cronológico, las canciones que han colgado los usuarios a los que sigues más tuyas.

MELOM: canciones que han subido los usuarios para que sean escuchadas.

MELÓMANO: todos los usuarios registrados en la aplicación, tanto si son músicos y creadores de contenido o seguidores que comentan las canciones que más les gustan.

COMENTARIO: mensajes escritos por los Melómanos en cada Melom.

ESCUCHAR(Seguir): funcionalidad de la aplicación para que aparezcan los nuevos Meloms en el timeline de aquellos a quien escuchas.

Nada más entrar en la aplicación Melomify, encontrar la siguiente pantalla y para poder empezar con el show de Rock & Roll tiene que hacer clic en nuestro **logotipo**.

Si estas registrado, ¡adelante! Disfruta del show mostrando las canciones que más te gusten y que merezcan ser escuchadas.



Ilustración 7 – Pantalla de bienvenida de Melomify



Para los indecisos, no hay problema, siempre hay solución. Se puede ver cómo va el funcionamiento de la aplicación dando en [Conócenos](#).



Ilustración 8 – Login de la pantalla de bienvenida

Si has visto el funcionamiento y deseas ser parte del espectáculo es muy sencillo. Da a [Regístrate](#) y sigue los pasos que indicamos.

Empieza a marcar el ritmo

Nombre de usuario*

Contraseña*

Nombre* Primera letra mayúscula.

Apellido* Primera letra mayúscula.

Correo electrónico*

Nombre del grupo

Género musical

Página web

+ Foto de perfil

Regístrate

Ilustración 9 – Pantalla de registro de Melomify



Una vez que hayas llenado este formulario, te llegará una notificación vía correo electrónico, para que puedas empezar a rockear.



Ilustración 10 – Email de bienvenida a Melomify

Este es el estilo de las notificaciones. Si ya te ha llegado, ¡ya tienes el pase para poder empezar el show!

Ahora que ya tienes cuenta, puedes ingresar en nuestra aplicación y disfrutar del espectáculo.

Ya has accedido a nuestro escenario, donde podrás colgar la música que más te guste y que no caiga en el olvido.

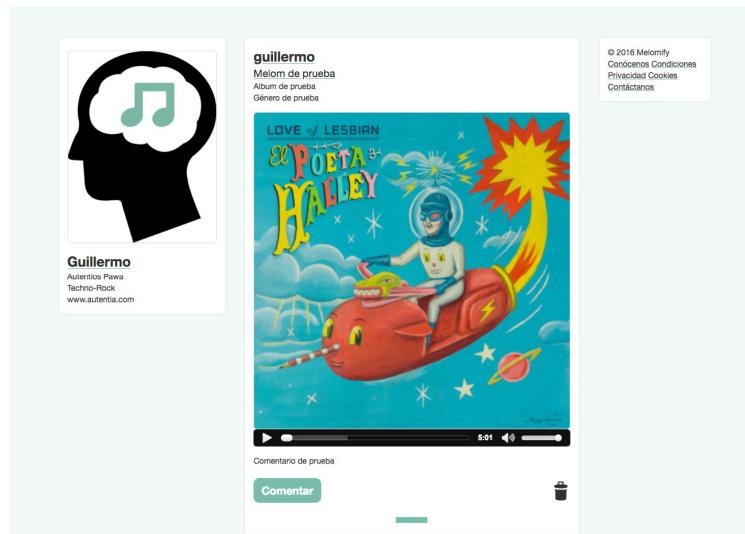


Ilustración 11 – Timeline de Melomify



Esta es la parte principal de la aplicación. Aquí es donde podrás colgar tus Meloms para que puedan ser escuchados.

Para poder colgar un Melom, mira la barra de navegación y busca un botón donde ponga **Melom!**.

Pon a sonar un Melom!

Título*

Álbum

Género

Comentario

[+ Imagen del álbum](#)

[+ Canción](#)

[Cerrar](#) [Melom!](#)

Obtendrás un formulario para llenar y subir la canción que quieras.

Puede ser que una canción no te haya gustado y quieras eliminarla, para ello es muy sencillo.

En tus Meloms, siempre está el siguiente ícono: un **cubo de basura**. Para poder eliminarlo, haz clic en el ícono y listo, tu Melom junto con los comentarios que tenga caerán en el olvido.

The screenshot shows the Melomify application interface. On the left, there's a profile card for "Guillermo" with the subtext "Autentico Pawa Techno-Rock www.autentia.com". In the center, there's a large button with the text "Empieza a marcar el ritmo...pon a sonar un Melom!" and a headphones icon. On the right, there's a sidebar with copyright information: "© 2016 Melomify" followed by links to "Conócenos", "Condiciones", "Privacidad", "Cookies", and "Contáctanos".



La gracia de esta aplicación es que puedes seguir a usuarios, y para hacerlo es muy fácil. Busca en la barra de la aplicación **Melóman@s**, en el que vendrá una lista en la que podrás seleccionar el usuario a quien quieras dándole al botón de **Escuchar** (y **Dejar de escuchar** si quieres dejar de seguirle). Ya tendrás, al instante, todas las canciones del usuario que sigues en tu Timeline, para que puedas disfrutar y comentar.



Ilustración 12 – Lista de Melóman@s

En este momento, tienes los Meloms del usuario al que acabas de seguir y quieres dejarle un comentario. Para hacerlo es muy simple. En su Timeline, verá los Meloms de otros usuarios, acompañados de un botón de **Comentar**. Pues haga clic en Comentar.

Obtendrá el siguiente formulario para llenar

The image shows a comment form. At the top, there is a user profile for 'guillermo' with the text 'Hola esto es un comentario'. Below this is a text input field with the placeholder 'Escribe un comentario...'. To the right of the input field is a green 'Comentar' button. At the bottom left of the form, there is a small number '45'.

Escribe tu comentario y da al botón **Comentar** y el comentario será registrado.



Si quieras cambiar algunas características de tu perfil, excepto el nick de usuario, puedes hacerlo mediante un clic en el botón **Modificar perfil**.

Nombre de usuario
Guillermo

Nombre
Guille

Apellido
Muela

Correo electrónico
gmuela@autentia.com

Nombre del grupo
Autentios Pawa

Género musical
Techno-Rock

Página web
www.autentia.com

Modificar perfil

Cuando acabes de modificar tus datos, simplemente da a **Guardar cambios** y los datos serán actualizados correctamente.

Modifica tus datos personales

Nombre de usuario	<input type="text" value="Guillermo"/>
Nombre	<input type="text" value="Guille"/>
Apellido	<input type="text" value="Muela"/>
Correo electrónico	<input type="text" value="gmuela@autentia.com"/>
Nombre del grupo	<input type="text" value="Autentios Pawa"/>
Género musical	<input type="text" value="Techno-Rock"/>
Página web	<input type="text" value="www.autentia.com"/>

Cambiar contraseña

Guardar cambios



Si deseas cambiar de contraseña, para tener más seguridad en tu cuenta, haz clic en el botón **Cambiar Contraseña**.

Modifica tu contraseña

Contraseña actual	<input type="text"/>
Nueva contraseña	<input type="text"/>
Confirmar contraseña	<input type="text"/>

Guardar cambios

Una vez hecho todo, haz clic en **Guardar cambios** y tu cuenta estará actualizada con tu nueva contraseña.

Para los usuarios que llevan mucho tiempo sin marcar el ritmo y quieran volver hacerlo, pero no recuerden su contraseña, el procedimiento para recuperarla es muy simple. En el login, hacer clic en **¿Olvidaste tu contraseña?**



Obtendrá el formulario que se muestra en la imagen. Para poder poner una nueva contraseña, escribe tu correo electrónico y dale a **enviar**.

Modifica tu contraseña

Contraseña actual	<input type="text"/>
Nueva contraseña	<input type="text"/>
Confirmar contraseña	<input type="text"/>

Guardar cambios

Cambio de contraseña █ Recibidos x 🖨️ 🖼

melomify@gmail.com 22:41 (hace 0 minutos) ☆ ↶ ↻ ⏪ ⏩

Has solicitado un cambio de contraseña

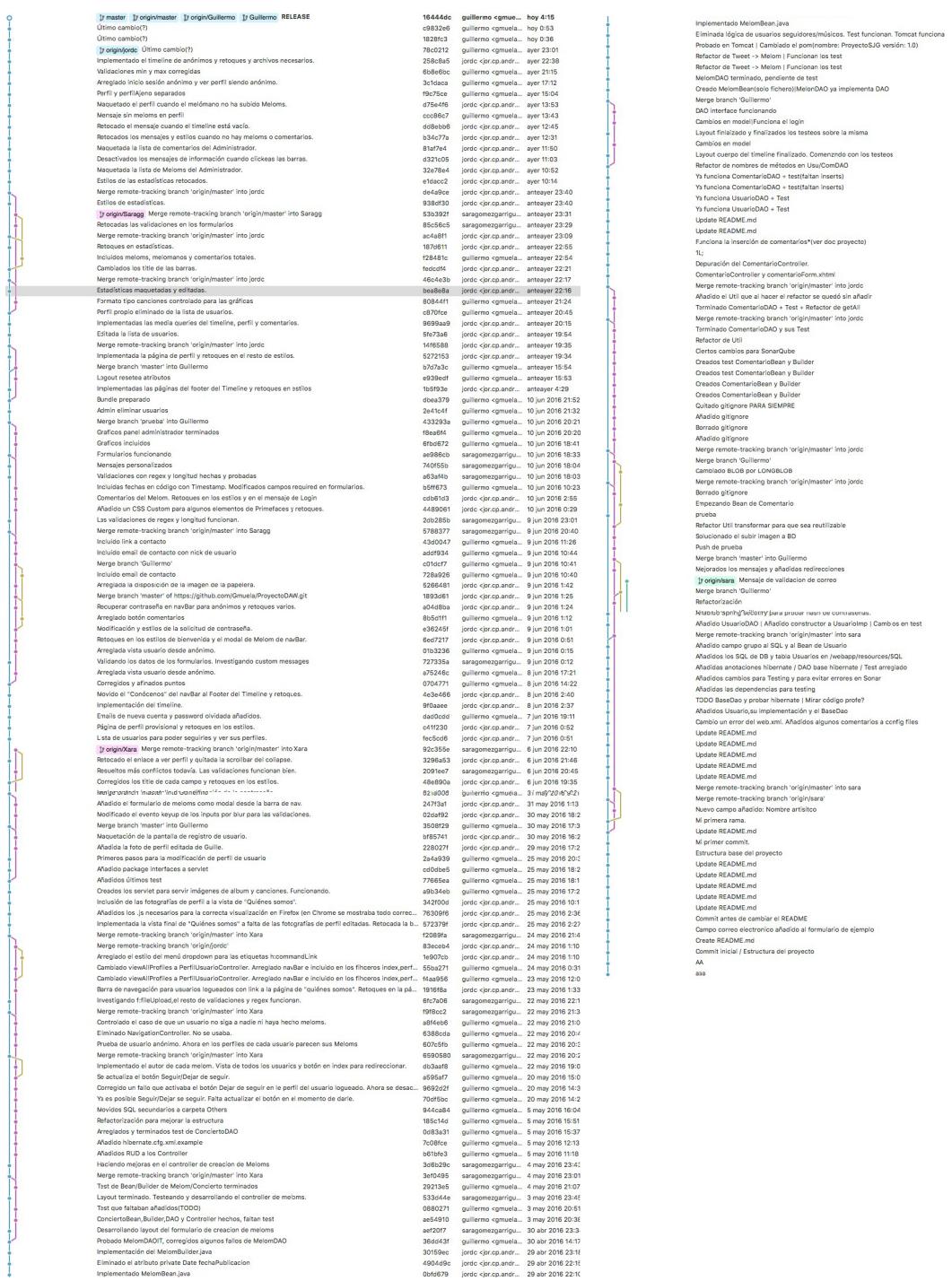
Tu nueva contraseña es: QHlipCeuh
No olvides cambiar la contraseña en el primer inicio de sesión y borrar este email. Tu cuenta podría correr riesgo.



9. DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN

Los cambios e incidencias producidos a lo largo del ciclo de vida del proyecto se han controlado y registrado gracias a la herramienta para control de versiones Git, facilitando el trabajo en equipo y la gestión de los conflictos con el código fuente.

A continuación se puede observar el árbol completo de progreso para cada una de las diferentes ramas de trabajo:





También se ha utilizado la plataforma online de gestión de proyectos Taiga, con la consecución de las historias de usuario correspondientes a cada sprint.

La siguiente tabla muestra los registros obtenidos en Taiga durante la realización del proyecto:

#Tarea	Asunto	Historia de usuario	Sprint	Dueño	Asignada a	Estado
57	Controlador timeline	46	Cuarto Sprint	gmuela	gmuela	Closed
61	Mostrar los datos de cada melom	46	Cuarto Sprint	gmuela	gmuela	Closed
101	Diseño y creación del Bean de usuarios	6	Primer sprint	gmuela	gmuela	Closed
102	Diseño y creación del back controlador login usuario	8	Primer sprint	gmuela	gmuela	Closed
104	Diseño y creación de la pantalla de bienvenida	9	Primer sprint	gmuela	jorc	Closed
105	Diseño y creación del formulario de registro de usuario	6	Primer sprint	gmuela	BlzBlaze	Closed
106	Diseño y creación del formulario de logueo de usuario	8	Primer sprint	gmuela	BlzBlaze	Closed
108	Diseño y creación de la tabla de meloms	107	Segundo Sprint	gmuela	BlzBlaze	Closed
109	Diseño y creación del Bean de Meloms	107	Segundo Sprint	gmuela	gmuela	Closed
110	Diseño y creación del formulario de creación meloms	107	Segundo Sprint	gmuela	jorc	Closed
113	Diseño y creación de la tabla de comentarios	111	Segundo Sprint	gmuela	BlzBlaze	Closed
114	Diseño y creación del Bean de comentarios	111	Segundo Sprint	gmuela	gmuela	Closed
115	Diseño y creación del formulario de creación de comentarios	111	Segundo Sprint	gmuela	jorc	Closed
116	Test bean usuario		Primer sprint	gmuela	gmuela	Closed
117	Test bean meloms		Segundo Sprint	gmuela	gmuela	Closed
118	Test bean comentarios		Segundo Sprint	gmuela	gmuela	Closed
122	Back controlador de usuarios	119	Tercer Sprint	gmuela	gmuela	Closed
123	Back DAO de usuarios	119	Tercer Sprint	gmuela	gmuela	Closed
124	Back controlador de meloms	120	Tercer Sprint	gmuela	gmuela	Closed
125	Back controlador de comentarios	121	Tercer Sprint	gmuela	BlzBlaze	Closed
126	Back DAO de comentarios	121	Tercer Sprint	gmuela	BlzBlaze	Closed
127	Back DAO de meloms	120	Tercer Sprint	gmuela	gmuela	Closed
128	Back controlador de seguidores(incluido en controlador usuario)	112	Tercer Sprint	gmuela	gmuela	Closed
129	Back DAO de seguidores(incluido como DAO usuario)	112	Tercer Sprint	gmuela	gmuela	Closed
130	Diseño y creación del Bean de seguidores	112	Tercer Sprint	gmuela	gmuela	Closed
132	Diseño y creación de la vista de "meet us"	131	Tercer Sprint	gmuela	jorc	Closed
134	Diseño y creación de la tabla seguidores	112	Tercer Sprint	gmuela	BlzBlaze	Closed
135	Test bean seguidores		Tercer Sprint	gmuela	gmuela	Closed
136	Test DAO usuarios		Tercer Sprint	gmuela	gmuela	Closed
137	Test DAO meloms		Tercer Sprint	gmuela	gmuela	Closed
138	Test DAO comentarios		Tercer Sprint	gmuela	gmuela	Closed
139	Test DAO seguidores		Tercer Sprint	gmuela	gmuela	Closed
140	Menú usuario registrado	59	Cuarto Sprint	gmuela	jorc	Closed
141	Menú usuario anónimo	59	Cuarto Sprint	gmuela	jorc	Closed
142	Diseño del logoípo	59	Cuarto Sprint	gmuela	jorc	Closed
143	Diseño de la web: colores, logos...		Cuarto Sprint	gmuela	jorc	Closed
144	Identidad corporativa		Cuarto Sprint	gmuela	jorc	Closed
145	Sesión de usuario	47	Cuarto Sprint	gmuela	gmuela	Closed
146	Passwords seguras	54	Cuarto Sprint	gmuela	gmuela	Closed
147	Diseño y creación del formulario de modificación	53	Cuarto Sprint	gmuela	jorc	Closed
148	Validación datos de login	47	Cuarto Sprint	gmuela	BlzBlaze	Closed
149	Diseño y creación de la página de perfil(propio y ajeno)	53	Cuarto Sprint	gmuela	jorc	Closed
150	Validación de datos formulario modificación	53	Cuarto Sprint	gmuela	BlzBlaze	Closed
151	Validación de datos de registro	54	Cuarto Sprint	gmuela	BlzBlaze	Closed
152	Controlador perfiles usuario(ajeno y propio)	53	Cuarto Sprint	gmuela	gmuela	Closed
153	Test builder de cada Bean		Cuarto Sprint	gmuela	gmuela	Closed
154	Diseño y creación de la vista del timeline	46	Cuarto Sprint	gmuela	jorc	Closed
155	Almacenamiento de ficheros en servidor con nombre aleatorio	54	Cuarto Sprint	gmuela	gmuela	Closed
156	Implementación de botón seguir/dejar de seguir	67	Cuarto Sprint	gmuela	gmuela	Closed
157	Actualización automática del botón seguir/dejar de seguir	67	Cuarto Sprint	gmuela	gmuela	Closed
158	Diseño y creación de la página lista de usuarios	31	Ultimo Sprint	gmuela	jorc	Closed
159	Back controlador lista usuarios(incluido en perfil usuario)	31	Ultimo Sprint	gmuela	gmuela	Closed
160	Diseño y creación de la página de error 404	95	Ultimo Sprint	gmuela	jorc	Closed
161	Configuración de la redirección a la página de error personalizada	95	Ultimo Sprint	gmuela	gmuela	Closed
164	Eliminación de meloms desde ambas vistas	88	Ultimo Sprint	gmuela	gmuela	Closed
165	Control del usuario anónimo	88	Ultimo Sprint	gmuela	gmuela	Closed
166	Eliminación de comentarios	97	Ultimo Sprint	gmuela	gmuela	Closed
167	Inclusión de todas las vistas para una navegación propia de usuario	80	Ultimo Sprint	gmuela	jorc	Closed
168	Control del usuario anónimo	31	Ultimo Sprint	gmuela	gmuela	Closed
175	Util controlador de emails autogenerados	90	Ultimo Sprint	gmuela	gmuela	Closed
176	Diseño y creación del formulario de petición de correo	90	Ultimo Sprint	gmuela	jorc	Closed
177	Validación formulario petición correo	90	Ultimo Sprint	gmuela	BlzBlaze	Closed
178	Inclusión del link en la pantalla de login	90	Ultimo Sprint	gmuela	jorc	Closed
180	Util controlador del envío del correo	179	Ultimo Sprint	gmuela	gmuela	Closed
183	Diseño y creación del front del Timeline	133	Ultimo Sprint	gmuela	jorc	Closed
184	Diseño y creación de la página donde se muestran los gráficos	91	Ultimo Sprint	gmuela	jorc	Closed
185	Back controlador del panel de administración	91	Ultimo Sprint	gmuela	gmuela	Closed



10. CONCLUSIONES

La aplicación web Melomify ha cumplido satisfactoriamente con los requisitos establecidos por el cliente. Dispone de las funcionalidades necesarias para crear una gran comunidad de amantes de la música.

Asimismo, la interfaz es minimalista y accesible para todo tipo de público. La navegación por todas las páginas es intuitiva y tiene una gran seguridad y robustez.

Gracias a la tecnología empleada durante su desarrollo, está preparada para albergar nuevas características y actualizaciones que demande el cliente o la comunidad en un futuro.

11 FUENTES

11.1 Legislación

DAW

Enseñanzas mínimas: Real Decreto 686/2010, de 20 de mayo (BOE 12/06/2010)

http://pdf/IFCSO3/titulo/RD20100686_TS_Desarrollo_Aplicaciones_Web.pdf

Curículo: Decreto 1/2011, de 13 de enero (BOCM 31/01/2011)

http://pdf/IFCSO3/curriculo/D20110001_TS_Desarrollo_Aplicaciones_Web.pdf

Definición de procedimientos de **control** y **evaluación**:

<http://www.xperta.es/es/descripcion.asp>

<http://www.xperta.es/es/aquienvadirigido.asp>

<http://churriwifi.wordpress.com/2010/04/10/gestion-de-incidencias/>

http://es.wikipedia.org/wiki/Control_de_versiones

11.2 Bibliografía

www.w3schools.com/

<http://librosweb.es/>

<http://stackoverflow.com/>

<https://www.adictosaltrabajo.com>

<http://getbootstrap.com/>

RAHMAN, Syed Fazle. **Jump Start Bootstrap**. SitePoint, 2014.

SHENOY, Aravind, SOSSOU, Ulrich. **Learning Bootstrap**. Packt Publishing, 2014.

NISKA, Christoffer. **Extending Bootstrap**. Packt Publishing, 2014.

SPURLOCK, Jake. **Bootstrap. Responsive Web Development**. O'Reilly Media, 2013.

DUCKETT, Jon. **JavaScript and Jquery: Interactive Front-End Web Development**. Wiley, 2013.

DUCKETT, Jon. **HTML and CSS: Design and Build Websites**. Wiley, 2011.

ÇIVICI, Çaðatay. **PrimeFaces User Guide** 5.3.

MKYONG. **JSF 2.0 Tutorial**. 2010-2013 <http://www.mkyong.com/tutorials/jsf-2-0-tutorials/>