



MySpot: Cambia como descubres nueva música

Jesús Martínez Maté – 2ºDAW



4 de junio de 2019

CIFP “Juan de colonia”

DEPARTAMENTO DE INFORMÁTICA

Índice de apartados de la memoria del proyecto

[1. Introducción 4](#_Toc8243869)

[1.1 Descripción 4](#_Toc8243870)

[1.2 Justificación 5](#_Toc8243871)

[2. Planificación 5](#_Toc8243872)

[2.1 Requisitos 5](#_Toc8243873)

[2.2 Recursos 6](#_Toc8243874)

[2.2.1 Recursos Hardware 6](#_Toc8243875)

[2.2.2 Recursos Software 6](#_Toc8243876)

[2.3 Planificación Temporal 6](#_Toc8243877)

[2.4 Planificación Económica 7](#_Toc8243878)

[2.4.1 Gastos de Hardware y Software 7](#_Toc8243879)

[2.4.2 Gastos de Desarrollo 7](#_Toc8243880)

[2.4.3 Total de Gastos Iniciales 7](#_Toc8243881)

[3. Tecnologías 8](#_Toc8243882)

[4. Tecnologías 9](#_Toc8243883)

[4.1 Diseño 9](#_Toc8243884)

[4.1.1 Diagrama de la aplicación 9](#_Toc8243885)

[4.1.2 Diagrama de la BBDD de la aplicación 12](#_Toc8243886)

[4.1.3 Diseño de interfaces 13](#_Toc8243887)

Índice de imágenes

[Imagen 1. 1 Pantalla inicial de la aplicación 4](#_Toc8157623)

[Imagen 2. 2 - Características del hardware usado…………………………………………………………………… 6](#_Toc8157630)

[Imagen 4 1 Diagrama de peticiones de contenido…………………………………………………………………… 9](#_Toc8242279) [Imagen 4 2 Diagrama de modelo de autenticación………………………………………………………………. 10](#_Toc8242280) [Imagen 4 3 Pantalla de formulario de petición de permisos al usuario…………………………………. 11](#_Toc8242281)

Índice de tablas

[Tabla 2 1 Planificación Temporal 6](#_Toc8242178)

[Tabla 2 2 Gastos Hardware y Software 7](#_Toc8242179)

[Tabla 2 3 Gastos de Desarrollo 7](#_Toc8242180)

[Tabla 2 4 Total de Gastos iniciales 7](#_Toc8242181)

[Tabla 4 1 - Tabla de Credenciales de la BBDD 12](#_Toc8242182)

[Tabla 4 2 Tabla de Usuarios de la BBDD 12](#_Toc8242183)

# Introducción

## Descripción

**MySpot** es una aplicación web cuyo objetivo es facilitar la administración de tus listas de reproducción dentro de Spotify y cambiar la manera en la que descubrimos nueva música dentro de la plataforma, proporcionando herramientas que permiten:

* Conocer información acerca de los artistas y canciones que el usuario ha escuchado recientemente o que más le han gustado en diferentes intervalos de tiempo pudiendo así incluso poder volver a descubrir música que hace mucho tiempo que no escucha.
* Crear listas de reproducción automáticamente a partir de las canciones que más ha escuchado el usuario para que pueda tenerlas todas reunidas en una misma *playlist*, a partir de las canciones que ha escuchado recientemente o seleccionando las canciones más populares de los artistas que más le gustan.
* Crear listas de reproducción automáticamente a partir de música totalmente nueva pero que entra dentro de los gustos del usuario gracias a la información que proporciona a la aplicación mediante sus “Top”
* Poder administrar sus listas de reproducción usando utilidades que no se proporcionan dentro de la aplicación de Spotify (por ejemplo, borrado múltiple de canciones)

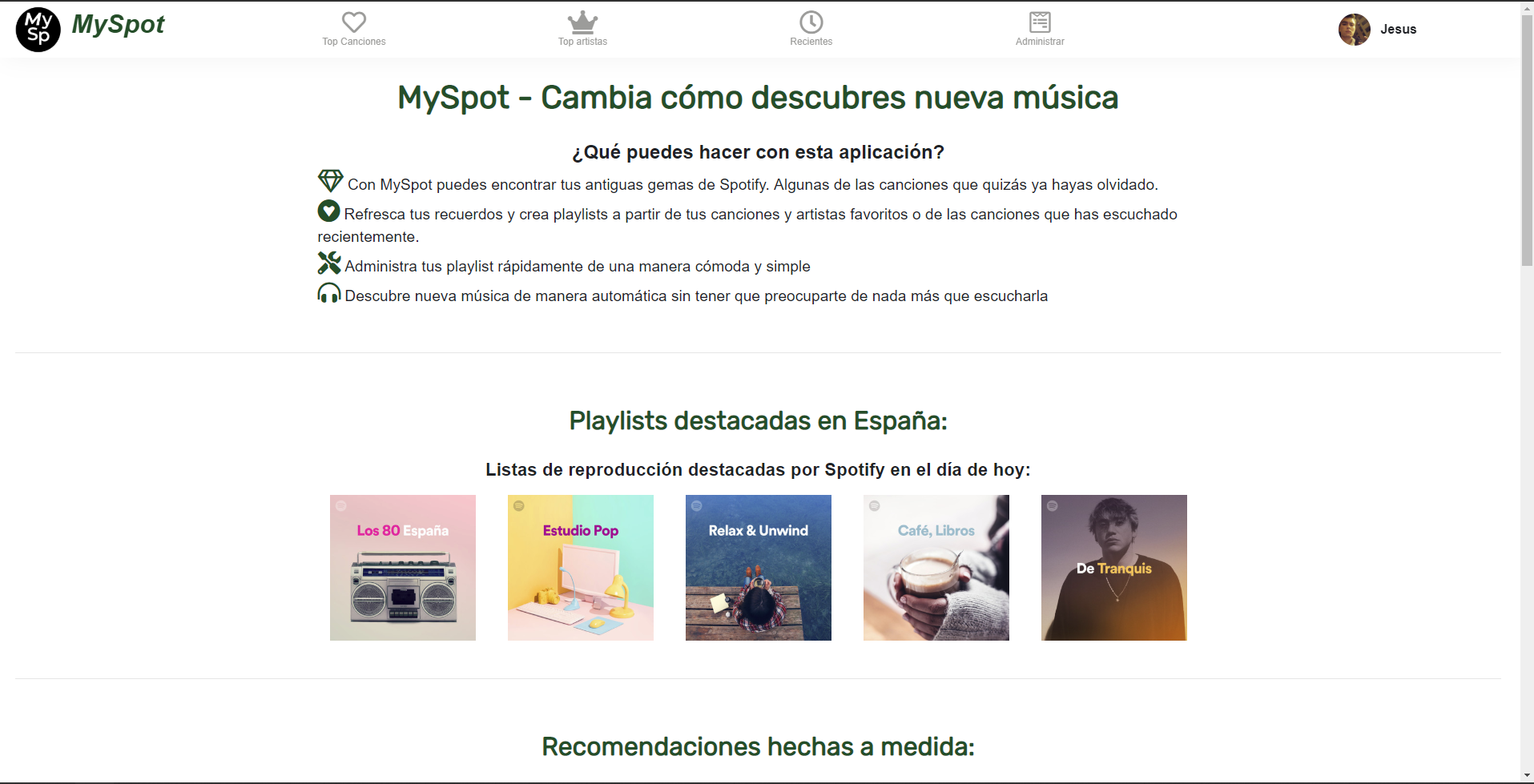


Imagen 1. 1 Pantalla inicial de la aplicación

## Justificación

Spotify es uno de los servicios de streaming de música online más utilizado en todo el mundo y que se encuentra constantemente en crecimiento incorporando nuevas funcionalidades y con un amplio programa para desarrolladores tanto dentro de la aplicación como para aplicaciones externas que utilicen su API.

Esta razón ha sido la principal para desarrollar MySpot, una aplicación que permite incorporar nuevas funcionalidades a la experiencia de uso diaria de los usuarios de Spotify facilitando de esta manera tareas como la administración de listas de reproducción personales o la creación automática de nuevas listas con música totalmente nueva (sustituyendo a la herramienta de DailyMix de Spotify, la cual no me satisface personalmente)

Todas las funcionalidades de la aplicación son utilidades que, personalmente, hecho de menos dentro de la plataforma al realizar uso de ella y que, al incorporarlas en MySpot, van a poder ser utilizadas por todos los usuarios que lo deseen.

# Planificación

## Requisitos

**Los requisitos funcionales de MySpot son los siguientes:**

* **Proporcionar información a los usuarios acerca de la música que más han escuchado** en diferentes intervalos de tiempo (último mes, últimos 6 meses, desde que se comenzó a usar el servicio de Spotify) **en forma de canciones y de artistas**
* **Proporcionar información sobre las últimas 50 canciones escuchadas** y dando la posibilidad de crear una lista con todas ellas
* Dar la posibilidad de **crear nuevas listas de reproducción** con:
  + Las canciones que forman el Top de canciones del usuario (en función del intervalo seleccionado)
  + Las canciones más populares de los artistas que forman el Top de artistas que más ha escuchado el usuario (en función del intervalo seleccionado)
  + Canciones similares a aquellas que forman el top de canciones del usuario consiguiendo una lista con música totalmente nueva que no haya escuchado pero que entre dentro de sus gustos personales
  + Canciones más populares de artistas relacionados con aquellos que forman el top de artistas que más le gustan al usuario, creando así una nueva lista en la que se disponga de nueva música que entre dentro los géneros que el usuario escucha y que le permita así descubrir nuevos artistas

**Los requisitos no funcionales:**

La aplicación se ha probado y ha sido optimizada para los diferentes tipos de dispositivos existentes (móviles, Tablets, ordenadores) al igual que para los navegadores más utilizados actualmente (Chrome, Firefox, Safari).

## Recursos

### Recursos Hardware

Los recursos hardware que se han utilizado para el desarrollo de este proyecto han sido los siguientes:

* Mi equipo personal (el desarrollo lo he realizado desde casa)

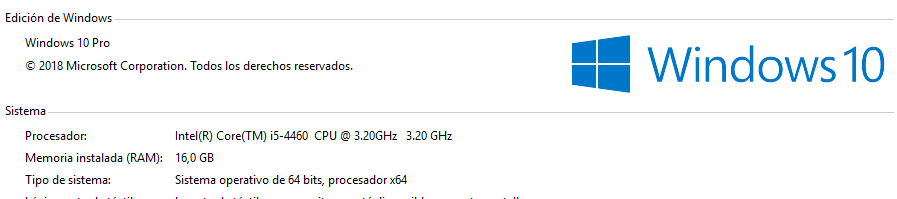


Imagen 2. 1 Características del hardware utilizado

### Recursos Software

Los recursos software que se han utilizado han sido los siguientes:

* Servidor de bases de datos PhpMyAdmin
* Servidor Web Nginx (Remoto), Servidor Web XAMPP (pruebas locales)
* Intérprete de lenguaje PHP
* Entorno de desarrollo Atom (facilidad de integración con control de versiones GitHub)
* Herramienta para el control de versiones de GitHub integrada en el entorno de desarrollo
* Gestor de dependencias para PHP **Composer** para poder hacer uso de la versión de la API de Spotify para el lenguaje PHP

## Planificación Temporal

En la tabla 2.1 se puede comprobar en detalle la planificación temporal inicial del proyecto

Tabla 2 1 Planificación Temporal

|  |  |
| --- | --- |
| **Descripción de la tarea** | **Horas invertidas** |
| Análisis de funcionalidades y requisitos y planteamiento de diseño inicial | 9 |
| Diseño y codificación de estilos de la aplicación web | 100 |
| Codificación de funcionalidades de la aplicación web | 150 |
| Rediseño del estilo y del funcionamiento de la aplicación | 20 |
| Pruebas de funcionamiento y correcciones de errores | 50 |
| Implementación de la aplicación | 2 |
| **TOTAL DE HORAS INVERTIDAS** | **331** |

## Planificación Económica

### Gastos de Hardware y Software

En la tabla 2.2 se puede ver al detalle el posible presupuesto necesario a invertir para mantener el proyecto en producción.

Tabla 2 2 Gastos Hardware y Software

|  |  |
| --- | --- |
| **Descripción del elemento** | **Presupuesto** |
| Suscripción del servicio de Hosting  (Permite obtener cambio de dominio, más tablas en BD, espacio en disco de servidor ilimitado entre otras funcionalidades necesarias para el crecimiento de la aplicación en un futuro cercano) | 48€/año |

### Gastos de Desarrollo

En la tabla 2.3 se pueden ver al detalle el posible presupuesto necesario a invertir en desarrollo para el futuro crecimiento y mejora de la aplicación web.

Tabla 2 Gastos de Desarrollo

|  |  |
| --- | --- |
| **Descripción de tarea** | **Presupuesto** |
| Mejoras en el funcionamiento actual de la aplicación y corrección de errores | 45€ |
| Configuración de servidor con la nueva suscripción y despliegue | 30€ |
| Diseño de nuevas secciones y maquetación de estas | 150€ |
| Codificación de las nuevas funcionalidades | 250€ |
| Pruebas de funcionamiento y depuración de errores | 200€ |
| Mantenimiento mensual de la aplicación | 35€ |
| **TOTAL PRESUPUESTO** | **710€** |

### Total de Gastos Iniciales

En la tabla 2.4 se puede observar el total del coste inicial del proyecto.

Tabla 2 4 Total de Gastos iniciales

|  |  |
| --- | --- |
| **Descripción de tarea** | **Presupuesto** |
| Gastos de hardware y Software | 48€/año |
| Gastos de desarrollo | 710€ |
| **TOTAL PRESUPUESTO** | **758€** |

# Tecnologías

* **HTML:**  Lenguaje de marcado utilizado para desarrollar la estructura de una página web. Hoy en día en considerado un estándar a la hora de crear aplicaciones y páginas web. Es una de las tres tecnologías principales usadas en la creación de páginas junto con CSS y JavaScript.

* **CSS:** Hojas de estilo en cascada, es un lenguaje de diseño gráfico utilizado para establecer la presentación de documentos escritos en un lenguaje de marcado como HTML. Habitualmente es usado para establecer el diseño visual de páginas web.
* **JAVASCRIPT:** Lenguaje de programación orientado al lado del cliente usado para crear páginas web dinámicas y mejorar la interfaz del usuario. Todos los navegadores interpretan el código JavaScript de las páginas web.
* **PHP:** Lenguaje de programación que funciona en el lado del servidor y que es usado principalmente para la generación dinámica de contenido en páginas web. Puede incorporarse directamente en un documento HTML. El código lo interpreta un servidor web usando un módulo procesador del lenguaje, el cual genera el HTML resultante que se muestra al usuario.
* **BOOTSTRAP:** Marco de trabajo o Framework web orientado al desarrollo del lado del cliente, principalmente para el desarrollo de la interfaz del usuario. Se forma de una serie de plantillas de diseño que contienen elementos como botones o menús entre otros elementos, así como extensiones adicionales de JavaScript.
* **AJAX:** Técnica de desarrollo web para crear aplicaciones dinámicas. Permite que sus aplicaciones, mientras se ejecutan en el cliente (navegador del usuario) mantengan una comunicación asíncrona con el servidor en segundo plano para poder realizar cambios en la página sin necesidad de recargarlas.
* **JQUERY:** Biblioteca de JavaScript que permite facilitar la manera en la que interactuamos con los elementos que formal el árbol de los documentos web (DOM), manejar eventos, animaciones y agregar interacción AJAX a nuestras páginas web.
* **SLICK:** Plug-in/Librería de JavaScript que permite implementar el uso de carruseles de elementos en nuestras páginas web utilizando un diseño responsivo y adaptable.
* **FONTAWESOME:** Conjunto de herramientas de fuentes e iconos basadas en CSS y LESS utilizada habitualmente en el marco de trabajo Bootstrap.
* **GOOGLE FONTS:** Librería de tipografías con licencia de uso gratuita que incorpora su propia página web en la que se pueden explorar dichas tipografías y hacer uso de su API para poder incorporar mediante CSS en nuestras páginas web las fuentes deseadas.

# Tecnologías

## Diseño

### Diagrama de la aplicación

#### Diagrama de solicitudes de contenido

En la imagen 4.3 se puede observar en detalle el esquema que siguen las peticiones del buscador cada vez que se solicita un contenido **dentro de** la página web (*independientemente de si se solicita el contenido de una sección, información mostrada en modal de creación de listas, …)* y cuando se realizan acciones dentro de ella (*creación de listas, administración de canciones, …*)

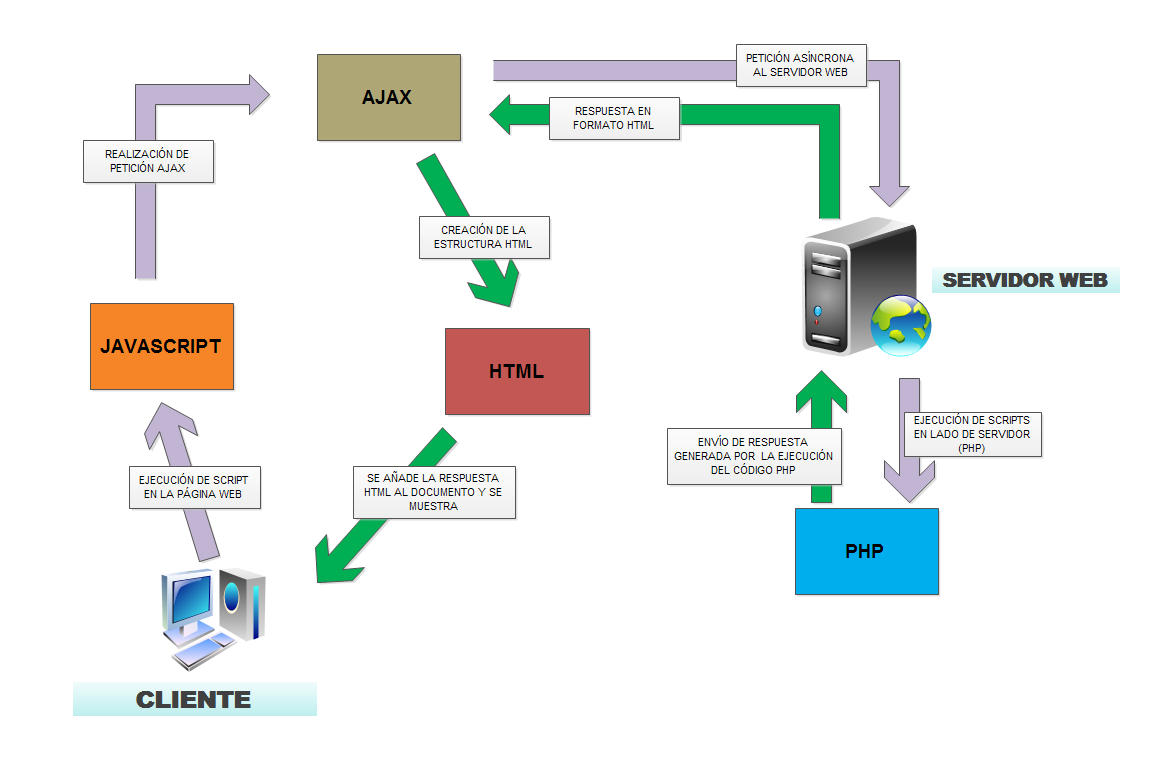


Imagen 4 1 Diagrama de peticiones de contenido

#### Diagrama del proceso de autenticación en la aplicación

En la imagen 4.2 se puede comprobar al detalle el modelo de autenticación que sigue la aplicación en el momento que un usuario intenta iniciar sesión.

Este modelo de autenticación, llamado ***Authorization Code Flow,*** es una de las múltiples variantes de modelos de autenticación que se ofrecen desde la API de Spotify para implementar en las aplicaciones que hagan uso de ella.

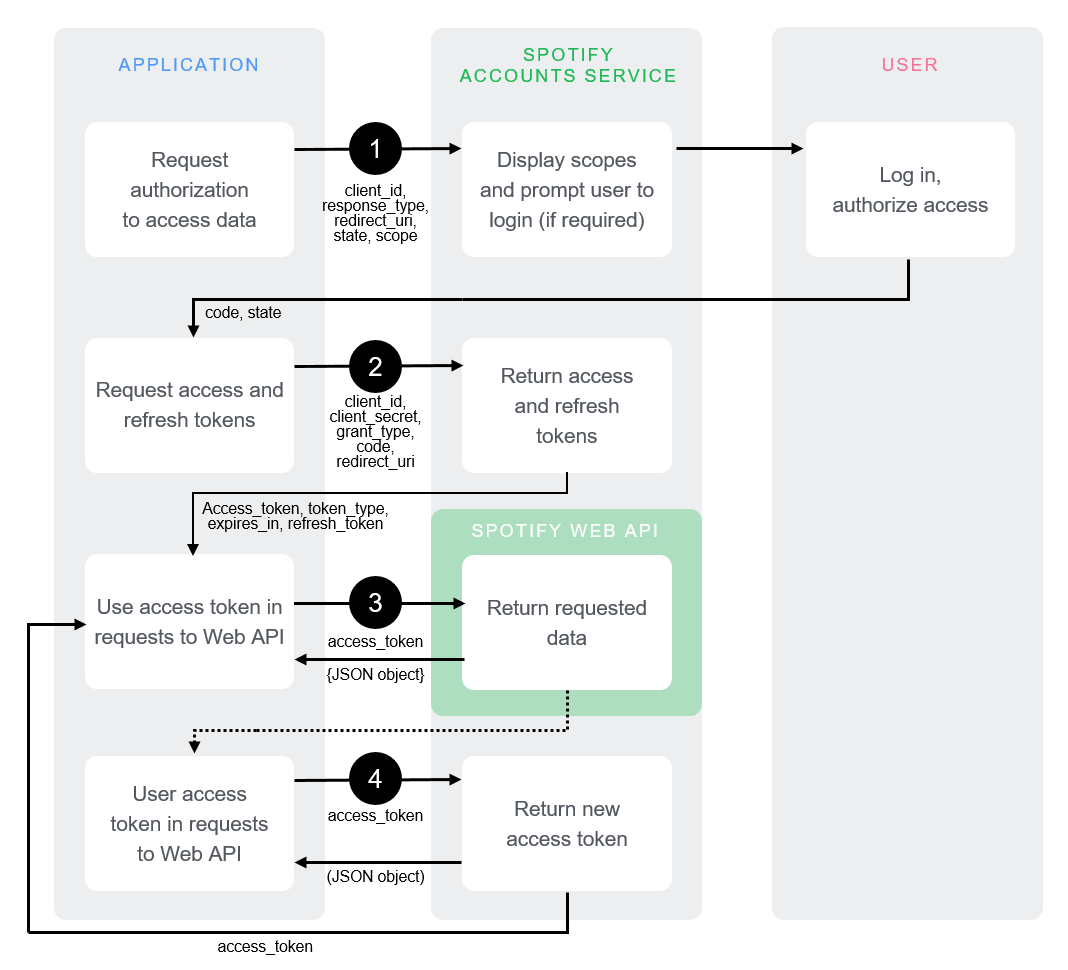


Imagen 4 2 Diagrama de modelo de autenticación

El proceso se divide en 4 pasos como se ve en la imagen anterior, procedo a explicarlos:

**PASO 1**

1. La aplicación pide autorización a Spotify para poder acceder a los datos del usuario que inicia sesión en ella (almacenados en su BD), enviando para ello:
   1. Sus datos asociados según el programa de aplicaciones de Spotify (client\_ID, redirect\_uri)
   2. Los scopes o puntos de alcance de la aplicación dentro de la API, es decir, las acciones que va a poder realizar la aplicación con los datos del usuario haciendo uso de la API (*como por ejemplo recuperar datos sobre sus canciones preferidas*)
2. El servicio de autenticación de Spotify informará al usuario de lo que la aplicación va a poder tener permiso de realizar en su cuenta si él lo permite (gracias a este modelo de autenticación el usuario solamente necesitará dar su permiso una sola vez) como se ve en la imagen 4.3:



Imagen 4 3 Pantalla de formulario de petición de permisos al usuario

1. El usuario garantiza el acceso y puede acceder a la aplicación

**PASO 2**

* Una vez que el usuario garantiza el acceso, la aplicación pasa a solicitar a Spotify los códigos o “tokens” necesarios para poder trabajar con la API. En este modelo de autenticación Spotify ofrece a las aplicaciones **dos tokens**, uno de acceso a los datos y otro de refresco (el cual permite extender la validez del token de acceso cuando este expira, normalmente al cabo de una hora)

**PASO 3**

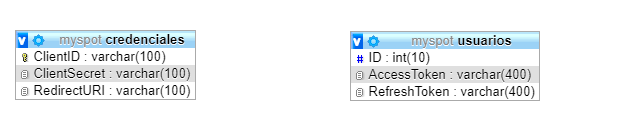
* Cuando la aplicación ya dispone de los tokens de acceso y refresco, ya puede realizar peticiones a la API de Spotify haciendo uso del token del acceso mientras este tenga validez

**PASO 4**

* Este paso del modelo solamente se realiza cuando es necesario refrescar el token de acceso porque ha llegado a su tiempo máximo de validez. El proceso para renovarlo es:
  1. Haciendo uso del token de refresco, se hace una nueva petición a la API de Spotify en la que se solicita un nuevo token de acceso
  2. La API valida el token de refresco y devuelve el nuevo token de acceso el cual tiene que sustituir al anterior para poder continuar trabajando.

### Diagrama de la BBDD de la aplicación

En la imagen 4.4 se puede ver la base de datos de la propia aplicación:



Como se puede apreciar, la base de datos de la aplicación no tiene un gran número de tablas y relaciones. Esto es debido a que, como he explicado en el 4.1.1.2, la aplicación basa la autenticación y la solicitud de datos en la BBDD de Spotify. Aclarado una vez más el funcionamiento de la solicitud de datos a la BBDD de Spotify, paso a detallar el contenido de las tablas propias de la aplicación:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tabla 4 1 - Tabla de Credenciales de la BBDD   |  |  |  | | --- | --- | --- | | **CREDENCIALES** | | | | **Campo** | **Tipo** | **Descripción** | | ClientID | VARCHAR (100) | Almacena el ID de cliente que identifica de manera pública a la aplicación | | ClientSecret | VARCHAR (100) | Almacena un ID **secreto** que solamente puede conocer el desarrollador de la aplicación con el que se identifica realmente a la aplicación | | RedirectURI | VARCHAR (100) | Almacena una URL “blanca” a la que la API redirige después de lograr o fracasar en la autenticación (normalmente apunta a un archivo en el que se realizan las operaciones pertinentes) | |

Tabla 4 2 Tabla de Usuarios de la BBDD

|  |  |  |
| --- | --- | --- |
| **USUARIOS** | | |
| **Campo** | **Tipo** | **Descripción** |
| ID | VARCHAR (100) | Identificador único del usuario proporcionado por la API de Spotify |
| AccessToken | VARCHAR (100) | Token de acceso a la API para realizar peticiones en cada sesión del usuario |
| RefreshToken | VARCHAR (100) | Token de refresco solamente usado para renovar el token de acceso cuando sea necesario |

### Diseño de interfaces