|  |  |
| --- | --- |
| Una imagen de una carretera de curvas con árboles  PP0511  Julián Sánchez | Descripción breve  Memoria de un reproductor de música  Fecha  17/12/2019 |
|  |  |

Contenido

[Descripción del proyecto - 2 -](#_Toc28017174)

[Requisitos del proyecto - 3 -](#_Toc28017175)

[Análisis - 4 -](#_Toc28017176)

[Funcionalidades - 4 -](#_Toc28017177)

[Requisitos operativos - 4 -](#_Toc28017178)

[Elementos en pantalla - 4 -](#_Toc28017179)

[API - 5 -](#_Toc28017180)

[Planificación - 6 -](#_Toc28017181)

[Diseño - 7 -](#_Toc28017182)

[Manual de estilos - 7 -](#_Toc28017183)

[Prototipos & Mockups - 8 -](#_Toc28017184)

# Descripción del proyecto

Better than Spotify (**BTS**) es un reproductor de música en línea en la cual el usuario puede agregar su propia música y organizarla por playlists de forma sencilla y elegante.

BTS se basa en la comunidad ya que todo el contenido es subido por los usuarios, con esto en mente nuestro objetivo es que sea lo más personalizable posible, dando así dos modos en los que la página será visible; Un tema claro y un tema oscuro entre el que se podrá elegir de manera dinámica.

Todo el contenido subido a la plataforma será gestionable por el autor de dicho contenido, por lo que siempre se tendrá control sobre tus recursos.

También se proporcionará un sistema para poder compartir tu música favorita con un sistema de tops en los cuales se verán reflejadas las canciones y las playlists más escuchadas para cuando no tengas ninguna canción en específico en mente.

Contará con un navegador sencillo en el cual podrás buscar tus artistas, canciones o álbumes favoritos, los cuales podrás poner a reproducir de manera instantánea.

# Requisitos del proyecto

* Hardware
  + Servidor Debian

Aquí es donde estará hosteada la aplicación y el cual se encargará de entregar todos los datos necesarios.

* Software
  + Se requerirá de un Apache o similar para poder entregar la aplicación web al usuario final.
  + Se requerirá un servicio MYSQL o similar para poder guardar los datos.
  + Se requerirá un intérprete PHP o similar el cual se encargará de interpretar la lógica del servicio y que comunicará MYSQL con el usuario final.
* Se requerirá una separación del backend y el frontend para futuras ampliaciones.
* La conexión deberá estar cifrada por SSL mediante el protocolo HTTPS (Let’s Encrypt o similares).

# Análisis

BTS nace con el ánimo de poder compartir esas canciones que tanto te gustan de manera sencilla y cómoda a tus amigos. De momento la plataforma sólo estará disponible en versión Web, pero en su desarrollo se tendrá en mente una futura migración a dispositivos móviles.

Para poder alcanzar ese sentimiento de comunidad BTS tendrá que cumplir ciertos requisitos, tanto funcionales como operativos.

## Funcionalidades

* + - * Proporcionar una herramienta que permita al usuario compartir música.
      * Proporcionar un sistema de ‘Tops’ en los cuales se listarán las ~10 canciones más escuchadas de forma mensual.
      * Un método de búsqueda en el cual el usuario podrá ingresar tanto álbumes como nombres de artistas o canciones.
      * Una manera sencilla y directa en la cual el usuario puede saber que canción está escuchando actualmente y cuál será la siguiente.

## Requisitos operativos

* + - * Controles de la música que se está escuchando como ‘Pausa’, ‘Siguiente’, ‘Mezclar’, etc.
      * Un sistema de cuentas en las cuales poder iniciar sesión y registrarse.
      * Proporcionar una herramienta que gestione la música subida por el usuario.
      * Mostrar un listado de las últimas canciones y playlists subidas por la comunidad ordenadas de maneras cronológica.

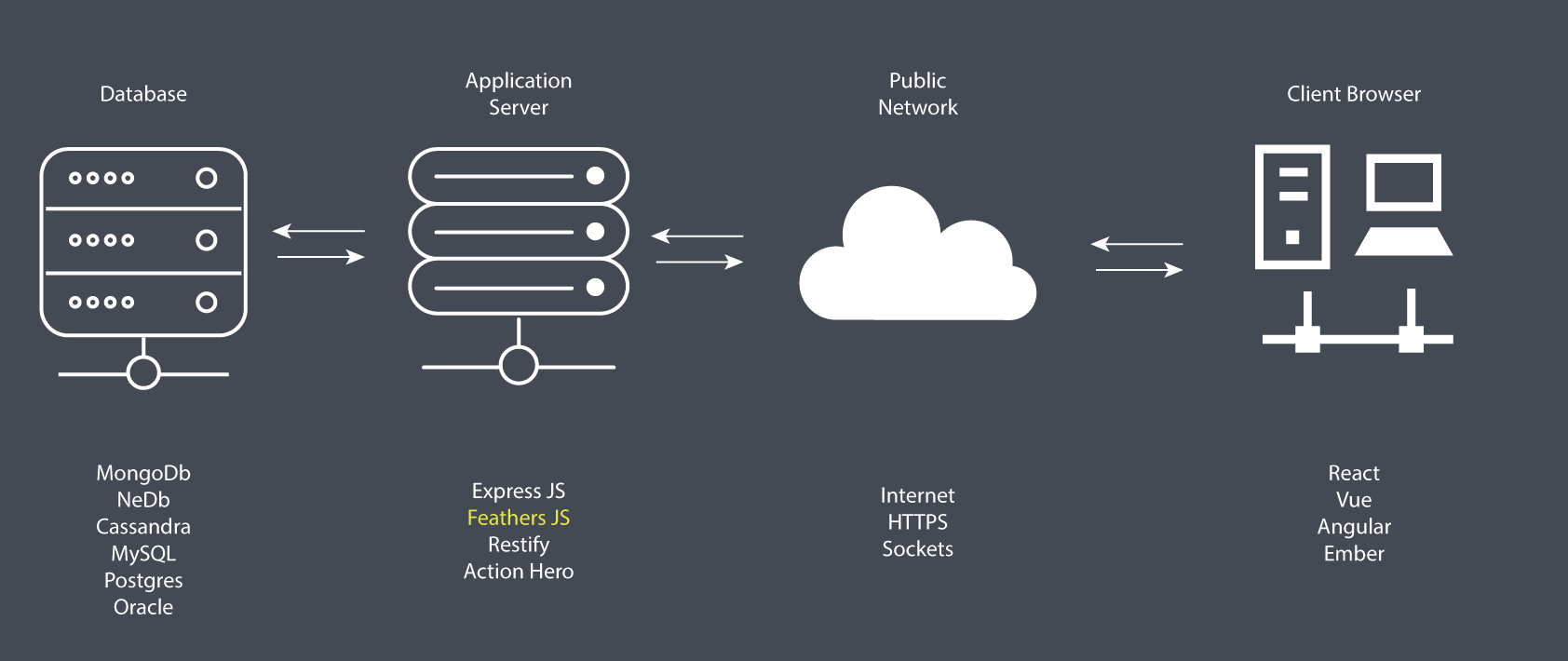
## Elementos en pantalla

* En todo momento tendrán que ser visibles tanto el nombre de la canción que se está reproduciendo como sus controles y carátula.
* El usuario si se está logueado en su defecto el inicio de sesión.
* El menú será siempre visible si se está en modo escritorio o Tablet.

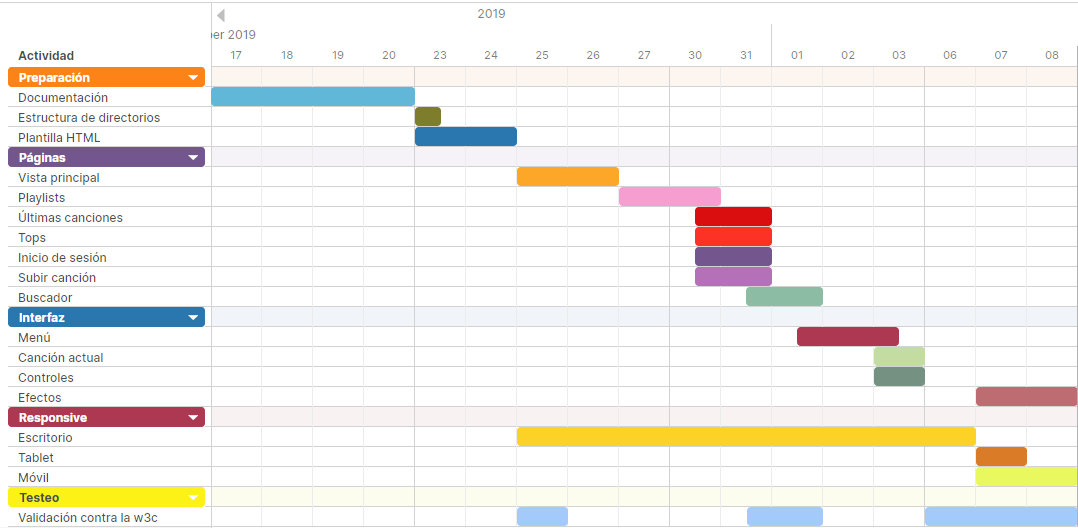
## API

El cliente web se comunicará con una API la cual será totalmente transparente y pública dando así la capacidad de poder ser accedida por cualquier usuario y ofreciendo la posibilidad de que otras aplicaciones sean creadas en base a BTS.

Un buen ejemplo de lo que se quiere conseguir con esta API sería seguir el siguiente esquema (No se tienen porque aplicar los frameworks descritos en el)



# Planificación



# Diseño

## Manual de estilos

La fuente a usar será “**Roboto Regular**” y tendrá un interlineado de 1.5em

La web deberá respetar los siguientes esquemas de colores. Aunque su similitud por separado no se pueda apreciar es importante respetarlos ya que en gran medida será completamente visible la diferencia de colores.

**Color secundario**

#111

RGB (17, 17, 17)

**Color principal**

#0f0f0f

RGB (15, 15, 15)

**Color terciario**

#0c0c0c

RGB (12, 12, 12)

Este se usará en los hipervínculos si los hay o en cualquier elemento en el cual se tenga que llamar la atención al usuario.

**Color de acentuación**

#008080

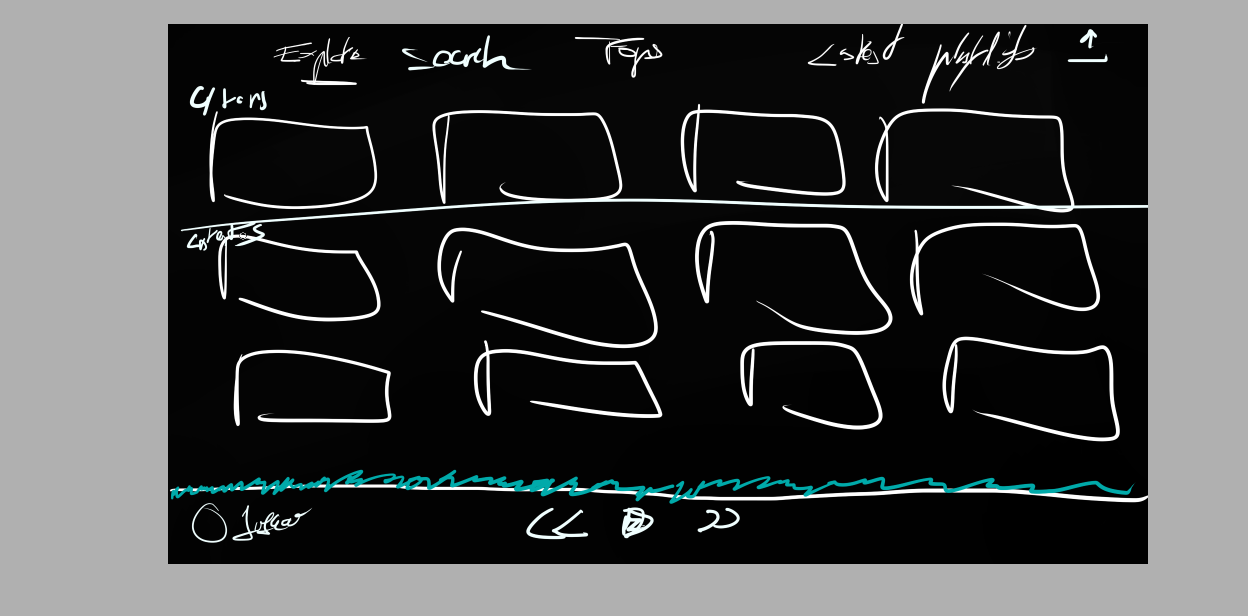
RGB (0, 128, 128)

Las letras serán tendrán un color blanco (#FFFFFF) y a aquellas en las que su valor sea de menor importancia (subtítulos, etc.) se les aplicará una opacidad de 0.8.

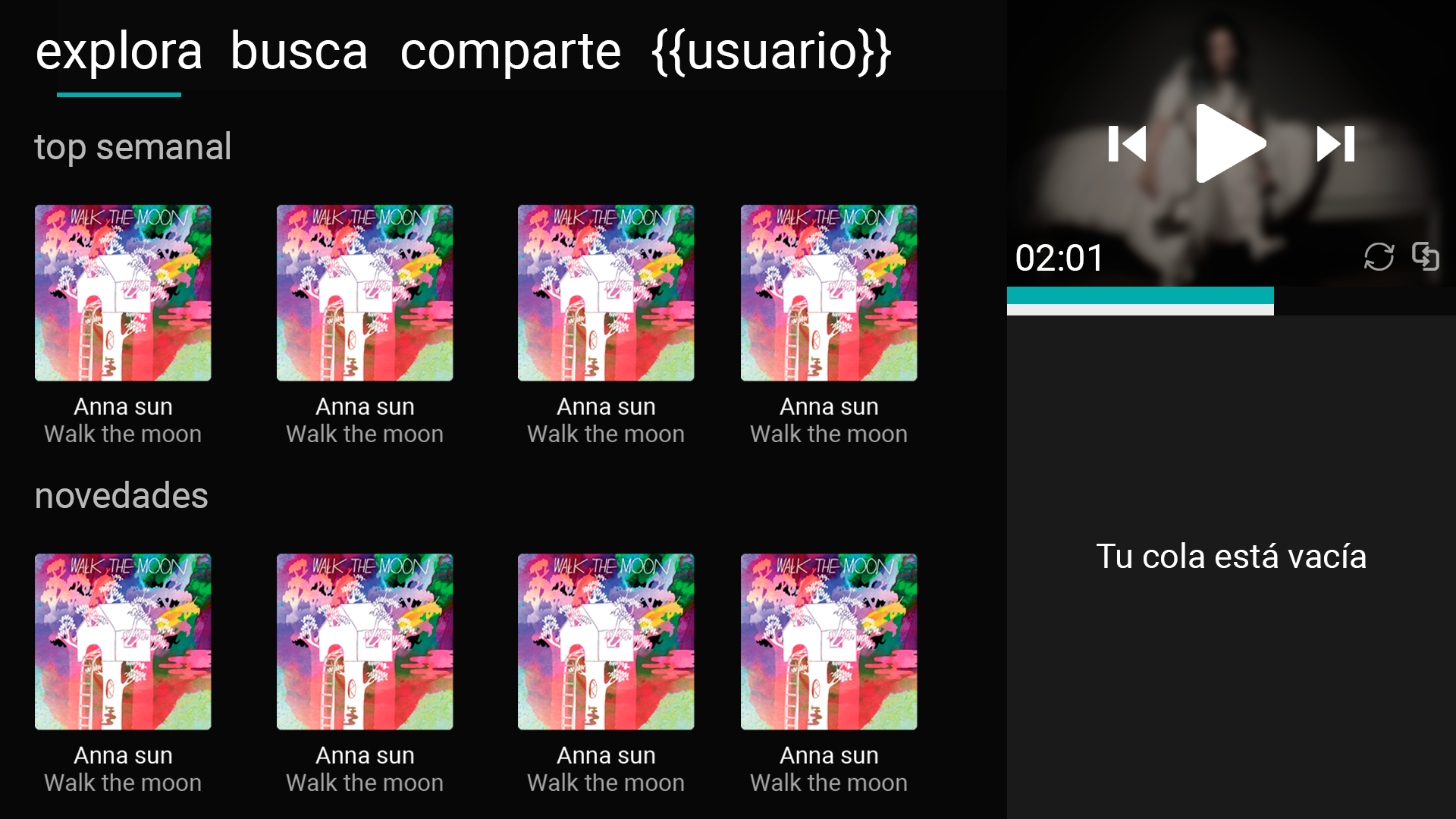
Como ya se ha dicho anteriormente es importante que el usuario tenga siempre visible tanto la canción actual tanto como sus controles independientemente de la página en la que se encuentre.

## Prototipos & Mockups

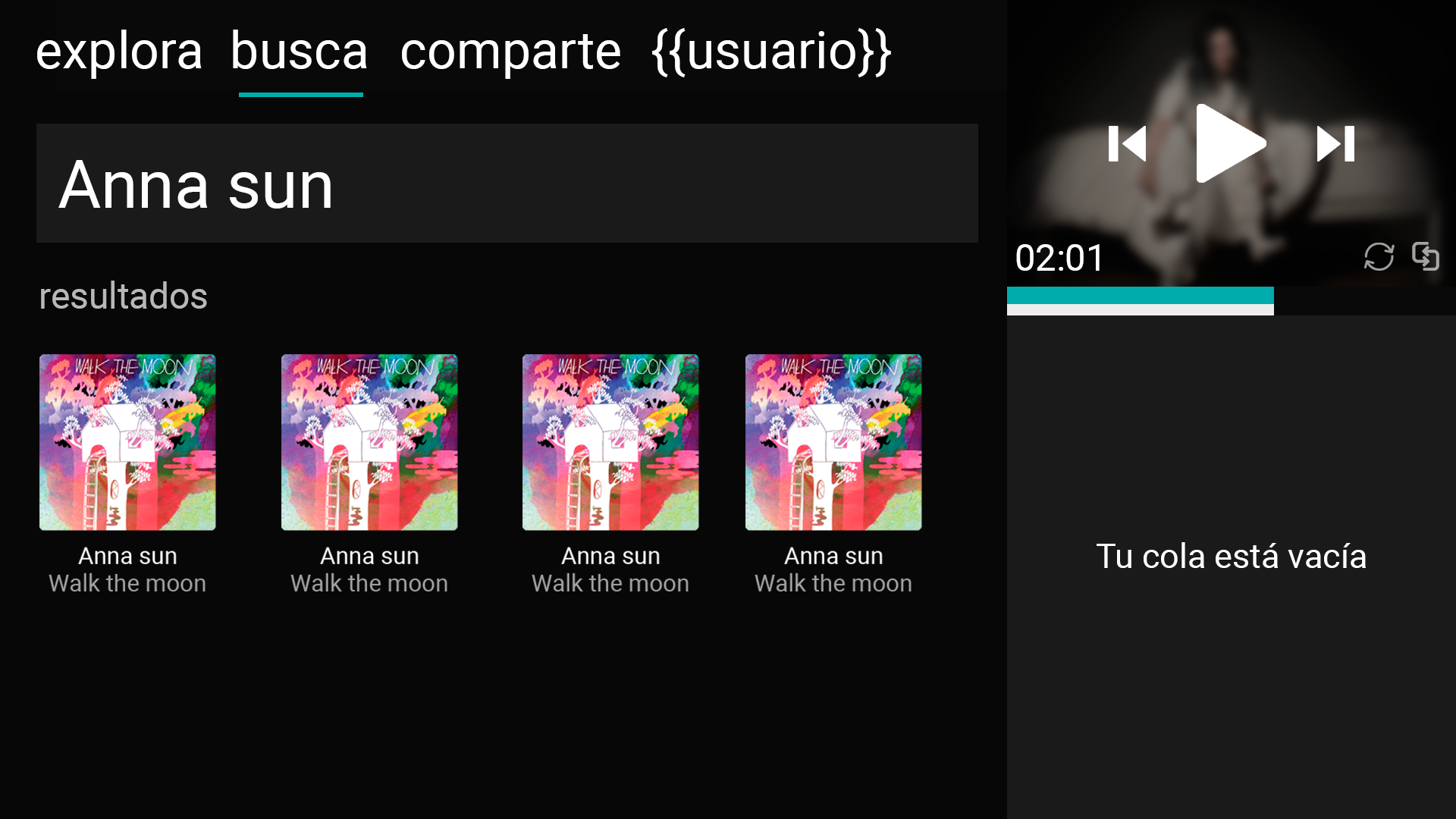
* Boceto inicial



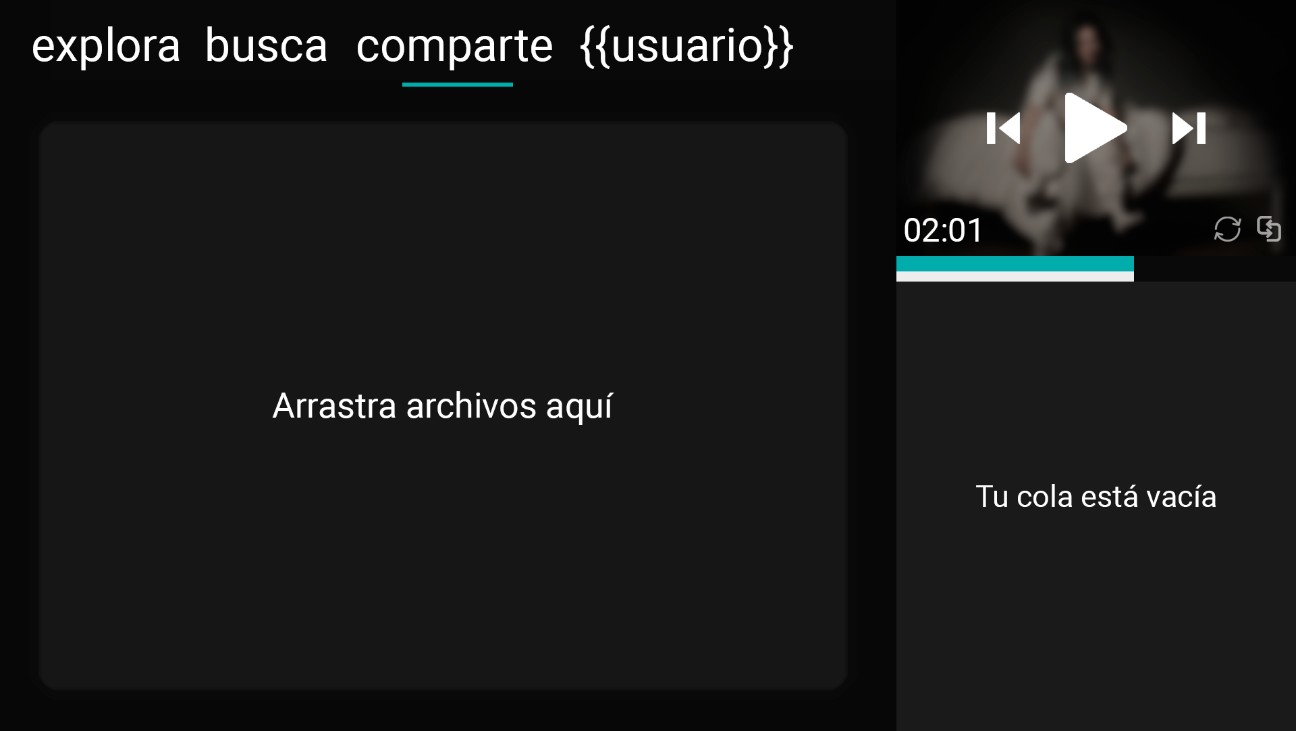
* Prototipo del index



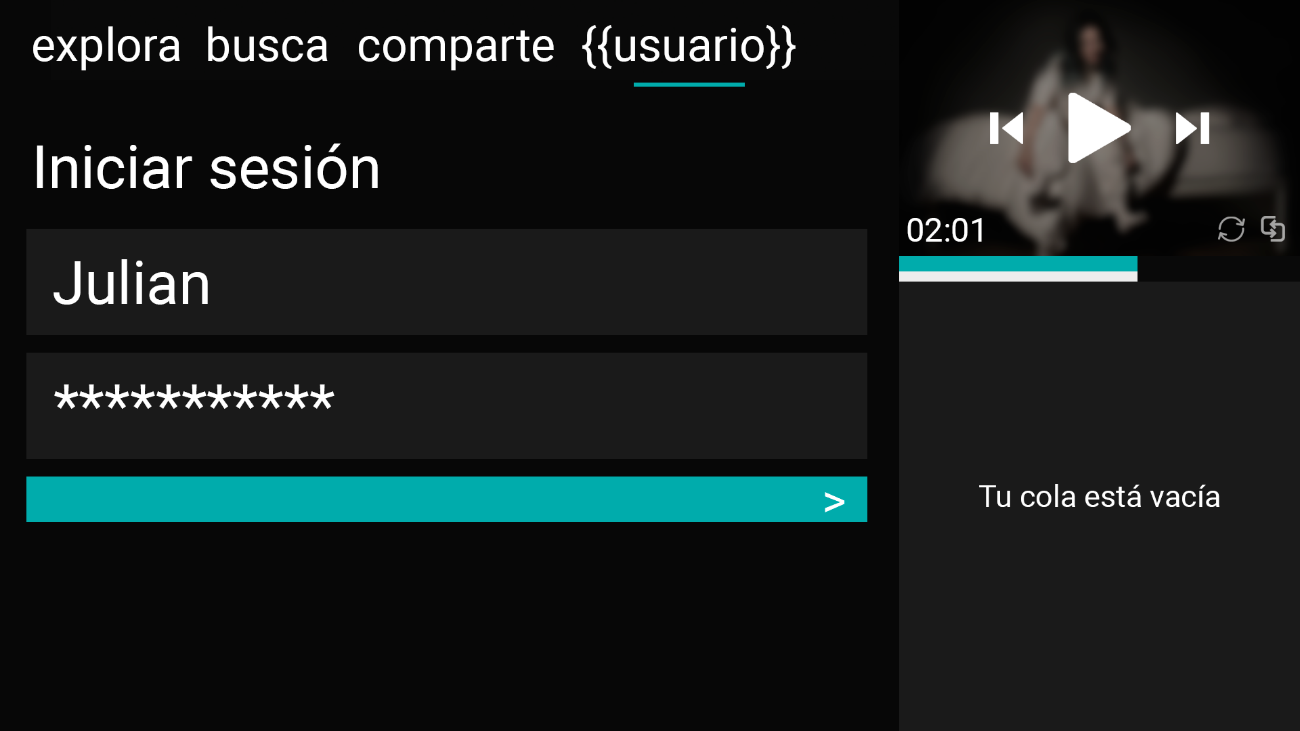
* Prototipo del buscador



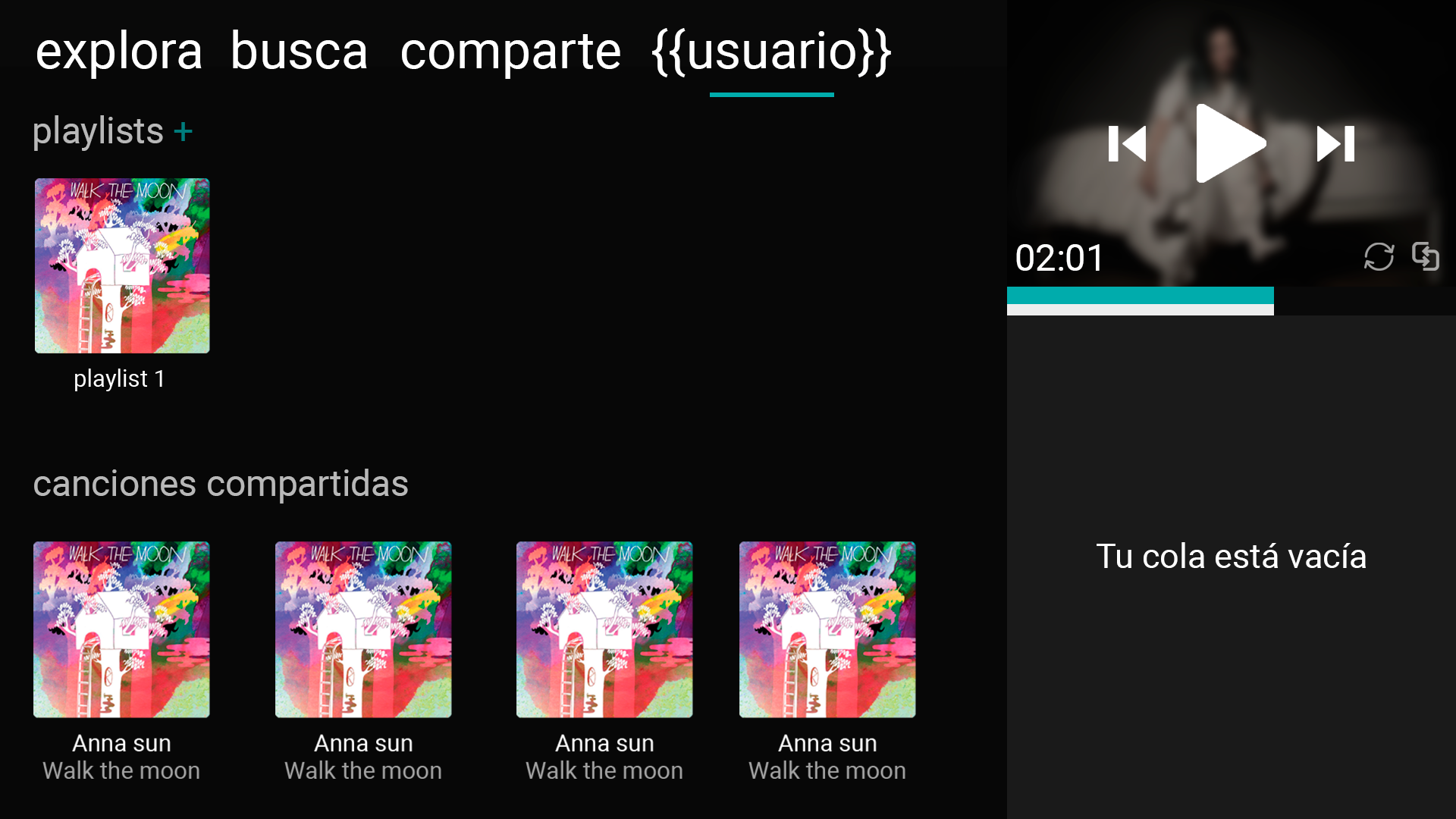
* Prototipo del sistema de subida de canciones



* Usuario sin loguearse



* Usuario con sesión



* Página de una playlist

