

**UNIVERSIDAD
PRIVADA DEL VALLE**



**“MANUAL TÉCNICO DE UN REPRODUCTOR DE MÚSICA WEB ESTILO
SPOTIFY”**

PROGRAMACIÓN WEB I

DEPARTAMENTO: SISTEMAS Y TECNOLOGÍA INFORMÁTICA

CARRERA: INGENIERÍA DE SISTEMAS INFORMÁTICOS

Estudiante: Laime Arcani, Cristhian Gabriel

DOCENTE: Miranda Ordonez, Henry

LA PAZ-BOLIVIA

2025

Introducción

Este manual técnico presenta el funcionamiento interno y la estructura general de mi proyecto, un clon básico de Spotify desarrollado como parte de la materia de Programación Web. El objetivo principal de esta página es permitir al usuario buscar canciones, reproducir música y visualizar información del artista, utilizando una API pública para obtener los datos en tiempo real.

En este documento explico de manera detallada cómo está organizado el proyecto, qué tecnologías utilicé y cómo se comunican entre sí los diferentes archivos que lo componen. También describo el proceso de instalación, la estructura de carpetas, el rol de cada archivo HTML, CSS y JavaScript, así como el flujo general del funcionamiento del sistema.

La intención es que cualquier persona que necesite revisar, mantener o mejorar este clon de Spotify pueda entender rápidamente cómo está construido y cómo funciona internamente. Por eso, el manual sirve como una guía técnica completa que muestra tanto las bases del proyecto como las decisiones principales tomadas durante su desarrollo.

Objetivos del Manual

- Explicar el funcionamiento interno del sistema
- Describir el código y la arquitectura

Tecnologías usadas

Aquí listas todo lo que usaste:

Frontend:

- HTML
- CSS
- JavaScript
- API (Deezer API, Free Lyrics)

- Fontawesome

Herramientas:

- Visual Studio Code
- GitHub

5. Arquitectura del sistema

Carpetas

1. css/ → Contiene todos los archivos de estilos (.css)
2. figma-img/ → Imágenes exportadas desde Figma (diseños, mockups)
3. img/ → Imágenes generales del sitio (logos, fotos, iconos)

Archivos HTML

1. ayuda.html → Página de ayuda/soporte
2. iniciar-sesion.html → Página para iniciar sesión (nota: tiene un typo - "sección" debería ser "sesión")
3. premium.html → Página de contenido premium
4. Proyecto PW1.html → Posiblemente tu página principal o índice
5. registrate.html → Página de registro de usuarios

Archivos JavaScript

1. login.js → Lógica para el inicio de sesión
2. main.js → JavaScript principal/global del sitio
3. usuario-avatar.js → Manejo del avatar/perfil de usuario

README.md

Documentación básica del proyecto

5.1 Estructura de carpetas

PROYECTO PW1/

```
|
|
|— estilos/      # css
|
|
|— imagenes/     # figma
|
|
|— main.js
|— login.js
|— usuario-avatar.js
|— Proyecto PW1.html
|— iniciar-sesion.html
|— registrate.html
|— premium.html
|— ayuda.html
|
|— README.md
```

5.2 Explicación de archivos

1. Explicación de Proyecto PW1.html (Página principal)

Este archivo es la estructura principal de Spotify. Aquí se arma todo el contenido que el usuario ve: la barra de navegación, las filas de canciones, la biblioteca, el sidebar derecho con información del artista y el reproductor. En otras palabras, es el “esqueleto” de la página.

A continuación, explico cada parte:

Encabezado (<head>)

- Declaro el tipo de documento y el idioma.

- Defino la codificación, la escala para móviles y el título de la página.
- Importo el archivo CSS principal (spotify_styles.css) que se encarga del diseño visual.

Esta parte solo prepara lo necesario para que la página funcione y se vea bien.

Barra de navegación superior

Dentro del <header> se encuentra la barra superior con:

- El logo de Spotify.
- El título.
- El buscador de canciones.
- Botones de navegación como “Premium”, “Ayuda”, “Registrate” e “Iniciar sesión”.
- Un botón tipo menú hamburguesa para dispositivos móviles.

Esta sección permite al usuario desplazarse por las partes principales de mi página.

Barra de navegación inferior (versión móvil)

Incluye botones como:

- Inicio
- Buscar
- Tu biblioteca
- Premium

Esta barra aparece en pantallas pequeñas, como la app de Spotify en celulares.

Estructura principal (div.principal)

Contiene tres partes:

Sidebar izquierdo – Tu biblioteca

Es un panel de navegación donde se mostrarán canciones guardadas o favoritas.

Tiene:

- Un título.
- Un listado dinámico que se llena desde JavaScript.

Contenido central (<main>)

Aquí van las tres filas principales del contenido:

Fila 1: Canciones en tendencia

Muestra canciones populares con su imagen, nombre y artista. Tiene botones de flechas para deslizar.

Fila 2: Artistas populares

Similar a la primera, pero con artistas.

Fila 3: Sugerencias para hoy

Recomendaciones de canciones o álbumes.

Estas filas primero tienen contenido vacío o de ejemplo, y luego main.js las llena con datos reales usando la API.

También aquí está el **footer**, donde coloco:

- Logo de Spotify.
- Enlaces a redes sociales reales.
- Enlaces a planes.
- Accesos a otras secciones del proyecto.

Sidebar derecho – Información del artista

Este panel aparece cuando el usuario selecciona una canción o artista.

Aquí se muestra dinámicamente:

- La foto del artista.
- Su nombre.
- Seguidores.

- Álbumes.
- Un texto descriptivo.
- La letra de la canción (si se usa).

Se abre y se cierra desde JavaScript.

Reproductor de música

El `<div id="reproductor-musica">` es donde yo inserto el reproductor desde JavaScript, para escuchar las canciones o los previews.

Todo el sistema de audio se maneja desde `main.js`.

Scripts

Al final del archivo importo:

- `main.js`: controla todo lo dinámico (API, clicks, reproductor, sidebar).
- `usuario-avatar.js`: controla el avatar del usuario.
- `FontAwesome`: para íconos en botones, flechas y menús.

2. Página de Planes Premium (premium.html)

Esta página muestra los diferentes planes de Spotify. Su estructura es la siguiente:

Encabezado

- Incluye el logo de Spotify dentro de un `<div class="logo">`.
- Un título principal `<h1>` con el nombre de la plataforma.

Cuerpo (main)

- Uso un contenedor `<div class="planes">` para organizar los tres planes.
- Cada plan está dentro de una etiqueta `<section>` con su respectiva clase:
 - **individual**
 - **free**
 - **estudiantes**

Cada sección de plan contiene:

- El logo de Spotify y el título del plan.
- El tipo de plan con un color representativo.
- El precio.
- Una lista de características usando ``.
- Un botón para obtener ese plan.

Pie de página (footer)

- Incluye nuevamente el logo de Spotify y el nombre.
- Una sección “Empresa” con enlaces a Facebook, Instagram y TikTok.
- Una sección “Planes de Spotify” con enlaces internos.
- Una sección “Enlaces útiles” con accesos a Inicio, Premium y Ayuda.

También enlazo un archivo JavaScript llamado **login.js**.

3. Página de Login (login.html)

Esta página permite al usuario iniciar sesión.

Encabezado

Similar a la página anterior: logo + título.

Cuerpo

- Contiene un formulario dentro de un `<div class="login">`.
- El formulario incluye:
 - Campo para correo electrónico.
 - Campo para contraseña.
 - Botón para continuar.

No tiene validación real porque el formulario apunta a `action="#"`.

Pie de página

Es igual al de la página de planes: logo, empresa, planes y enlaces útiles.

4. Página de Registro (signup.html)

La estructura es casi igual a la del login, pero adaptada para registrarse.

Encabezado

Logo y título “Spotify”.

Cuerpo

- Un contenedor `<div class="sign">` con un título invitando al registro.
- Un formulario que pide:
 - Email
 - Contraseña
 - Botón “Registrar”

Pie de página

El mismo que en las otras páginas para mantener coherencia visual.

5. Explicación de Ayuda (ayuda.html)

Esta página está diseñada para que el usuario pueda contactarse y enviar un mensaje acerca de un problema. Mantiene la misma estructura general que el resto del proyecto para conservar coherencia visual y funcional.

Encabezado (header)

El encabezado contiene:

- Un contenedor `<div class="logo">` donde se muestra el logo de Spotify.
- Un título `<h1>` con el nombre de la plataforma.

Esto permite identificar de forma clara la página y mantener la uniformidad con el resto del sitio.

Cuerpo principal (main)

Dentro de `<main>` se encuentra la sección principal de la página:

- Un contenedor `<div class="contactanos">` que agrupa todo el contenido relacionado con la ayuda.
- Un título doble dentro de `<div class="titulo-ayuda">`:
 - “¿Cómo te podemos ayudar?”
 - “Contáctanos”
- Un formulario creado con `<form action="#">` que incluye:
 - Un campo para ingresar el correo electrónico (`<input type="email">`).
 - Un campo para describir el problema (`<input type="text">`).
 - Un botón para enviar los datos (`<button type="submit">Enviar</button>`).

Este formulario no envía datos reales, ya que la acción está definida con `"#"`.

Pie de página (footer)

El footer sigue el mismo diseño que las otras páginas del proyecto e incluye cuatro secciones:

- **Logo-footer:** muestra nuevamente el logo y el nombre de Spotify.
- **Empresa:** contiene enlaces externos a Facebook, Instagram y TikTok.
- **Planes de Spotify:** muestra accesos a los distintos tipos de planes (Individual, Estudiantil y Free).
- **Enlaces útiles:** ofrece navegación interna hacia Inicio, Premium y Ayuda.

Archivos enlazados

Al final de la página se incluye:

- Un archivo JavaScript `login.js`, que es compartido entre varias páginas del sitio.

Explicación de styles_general.css

1. Estilos globales

Se eliminan los márgenes y rellenos por defecto de todos los elementos, se activa el modelo de caja “border-box” para facilitar el control del tamaño, y se define Arial como fuente principal.

El fondo de toda la página es negro.

Los enlaces no tienen subrayado y mantienen el color del texto del lugar donde están.

Los botones cambian el cursor al pasar el mouse, mostrando que son clicables.

2. Barra superior de navegación (navbar)

La barra superior se estructura con flexbox, distribuyendo los elementos a los extremos y alineándolos en el centro.

El lado izquierdo (logo y enlaces) está alineado horizontalmente, es de color blanco y tiene espacios entre cada elemento.

El lado derecho también está alineado horizontalmente.

El botón “Iniciar sesión” tiene un fondo blanco, texto negro, bordes redondeados y un relleno que lo hace más visible, similar al estilo de Spotify.

3. Estilos del buscador

El cuadro de búsqueda tiene un fondo gris muy oscuro, texto blanco, forma redondeada, sin borde y un ancho fijo.

Sigue la estética moderna del buscador de Spotify.

4. Sección principal (sidebar + contenido + footer)

La zona principal de la página utiliza flex para colocar el menú lateral y el contenido principal uno al lado del otro con un espacio entre ellos.

5. Sidebar izquierdo

El panel lateral tiene fondo oscuro, texto blanco, bordes redondeados y un ancho fijo que no se reduce en pantallas más pequeñas.

Dentro de él, los textos y enlaces están organizados en columna, con suficiente separación.

6. Sección principal de contenido (filas de música)

Cada fila de canciones funciona como un carrusel horizontal.

Las tarjetas dentro de la fila se colocan una junto a otra, sin que salten hacia abajo, y se muestra una barra de desplazamiento horizontal que está oculta visualmente.

El desplazamiento es suave para dar sensación profesional.

También existen flechas laterales (izquierda y derecha) que aparecen únicamente al pasar el mouse encima de una fila.

Estas flechas permiten moverse por el carrusel y están posicionadas al centro vertical del contenedor.

7. Tarjetas de canciones (items)

Cada tarjeta tiene una animación suave al pasar el cursor:

- se eleva ligeramente,
- aparece una sombra,
- el fondo se vuelve un poco más claro.

Además, cuando se pasa el mouse, aparece un ícono de “play” encima de la portada de la canción, animado para que aparezca suavemente.

Este es uno de los efectos más característicos del estilo de Spotify.

8. Pie de página (footer)

El footer tiene fondo oscuro, relleno amplio y se divide en columnas usando flex para mostrar secciones como Empresa, Enlaces o Planes.

Los títulos del footer están en blanco y los enlaces en un tono gris.

Cuando se coloca el mouse sobre cada enlace, se subraya o cambia ligeramente, mostrando interactividad.

9. Versión para pantallas pequeñas (responsive)

Cuando la pantalla es menor a 550 píxeles:

- El sidebar izquierdo desaparece para dejar más espacio al contenido principal.
- Aparece una barra inferior fija estilo “navegación móvil”, con iconos tipo Spotify móvil.
- La barra superior se convierte en un menú desplegable tipo hamburguesa.

- El contenido principal se centra y las filas se alinean al centro.
- El footer se organiza en una columna para adaptarse mejor a teléfonos.

Esto convierte tu página en una versión completamente usable en dispositivos móviles.

10. Reproductor de música inferior

Existe un reproductor de música que aparece en la parte inferior cuando se selecciona una canción.

Este reproductor está oculto inicialmente y aparece sobre la pantalla con un fondo negro, controles de audio y texto alineado horizontalmente.

11. Sidebar derecho (letra e información del artista)

Es un panel lateral que se muestra cuando se hace clic en una canción.

Tiene:

- fondo oscuro,
- bordes redondeados,
- espacio interno,
- un área para la letra,
- información del artista,
- una imagen superior del artista.

Normalmente está oculto y se activa cuando el usuario selecciona una canción.

En pantallas pequeñas, ocupa todo el ancho.

12. Efectos y animaciones

Tu archivo CSS incluye varias animaciones y transiciones suaves, por ejemplo:

- los botones cambian suavemente al pasar el mouse,
- las tarjetas de canciones se elevan y muestran sombras,
- aparece el icono de “play” con un efecto gradual.

Esto mejora la experiencia del usuario y hace que tu página se parezca más a Spotify.

Explicación de styles_general.css

1. Estilos globales

Se eliminan los márgenes y rellenos por defecto de todos los elementos y se activa el modelo de caja “border-box” para controlar mejor el tamaño.

Se define la tipografía Arial/Helvetica y se establece un fondo negro en toda la página, con texto en color blanco.

Los enlaces no tienen subrayado y heredan el color del elemento padre.

Los botones muestran el cursor en forma de mano al pasar encima.

2. Logo y encabezado

La imagen del logo se ajusta a un tamaño de 50×50 píxeles.

El encabezado se organiza con flexbox, alineando horizontalmente el logo y el título.

El título tiene un tamaño de 20 píxeles.

3. Sección principal

El contenido dentro de <main> se centra horizontalmente.

4. Estilos del formulario de login

El contenedor de login tiene fondo oscuro, altura de 600 px y ancho de 300 px, alineando su contenido al centro mediante flexbox.

El formulario se organiza en columna con separación entre sus elementos.

Los inputs tienen fondo oscuro, texto blanco y borde blanco.

El botón del login es verde, con texto negro, bordes redondeados y un estilo llamativo similar a Spotify.

El título del login es grande, muy grueso y está centrado verticalmente.

5. Estilos del footer

El pie de página usa un fondo gris muy oscuro y texto blanco.

Se divide en secciones mediante flexbox, con un espacio amplio entre columnas.

El logo dentro del footer mantiene el tamaño de 50×50 px.

Las listas no tienen viñetas y los elementos tienen separación vertical.

Los títulos del footer son más grandes que los textos y tienen un pequeño margen inferior.

6. Estilos de la página Sign Up

La página de registro usa los mismos estilos base que login.

El título es más ancho (650 px) para permitir textos grandes centrados.

La caja .sign tiene las mismas dimensiones y estructura que .login.

Los inputs y el botón mantienen la misma estética.

El propósito es mantener coherencia visual en todas las páginas.

7. Estilos de la página Ayuda

El título de la sección de ayuda es grande, centrado y con un margen amplio superior e inferior.

El contenedor .contactanos tiene fondo gris oscuro, bordes redondeados, un tamaño de 850×600 px y su contenido se centra con flexbox.

El formulario dentro de esta sección ocupa el 45% del ancho y organiza sus inputs en una columna con separación.

Los inputs tienen el mismo estilo que en login, pero el botón es blanco con texto negro, creando un contraste distintivo.

8. Responsividad (Media Queries para móvil)

Cuando la pantalla es menor a 600 px:

- El footer cambia a diseño vertical, centrado y con mayor separación entre secciones.
- El tamaño del texto en el footer aumenta para mejorar la legibilidad.
- Los títulos de Sign Up y Ayuda reducen su tamaño y su ancho para ajustarse a la pantalla.
- El contenedor de ayuda se hace más pequeño, cambia su altura a automática y agrega relleno interno.
- El formulario de ayuda pasa a ocupar el 70% del ancho para una mejor usabilidad en pantallas pequeñas.

Esto garantiza una buena experiencia tanto en computadora como en celular.

9. Animación del botón de envío

El botón de envío utiliza:

- una animación de aparición al cargarse la página,
- una transición suave,
- un efecto de elevación (hover) con sombra,
- y un pequeño rebote cuando se presiona (active).

Estas animaciones hacen el formulario más interactivo.

10. Animaciones del formulario

El contenedor del login aparece con una animación de desvanecimiento.

Los inputs de tipo email y password aumentan ligeramente su tamaño al estar en foco, gracias a una transición suave.

Estas animaciones mejoran la experiencia de usuario sin sobrecargar la página.

Explicación de `styles_premium.css`

El archivo CSS define los estilos visuales de toda la página web, organizados en secciones como estilos globales, cabecera, tarjetas de planes, footer y ajustes para pantallas pequeñas.

1. Estilos globales:

Se eliminan los márgenes y rellenos predeterminados, se activa el box-sizing para manejar mejor los tamaños y se define la fuente general. El cuerpo de la página usa fondo negro (`#0E0E0E`) y texto blanco. Los enlaces no tienen subrayado y heredan el color del texto. Los botones muestran el cursor de mano al pasar por encima.

2. Logo y encabezado:

El logo tiene un tamaño fijo de 50x50 píxeles. El encabezado utiliza flexbox para alinear elementos horizontalmente y su título principal tiene un tamaño de fuente de 20px.

3. Sección principal (planes):

El contenedor de planes usa flexbox, centra las tarjetas, permite que se ajusten a múltiples filas y mantiene un espacio entre ellas. Los textos como títulos y subtítulos tienen márgenes específicos para acomodar la presentación.

Cada plan es una tarjeta con fondo gris oscuro (#191919), bordes redondeados, un tamaño de 300x400px, márgenes superiores e inferiores amplios y un diseño interno en columna.

Las imágenes dentro de los planes tienen tamaño de 35x35px. Las listas dentro de las tarjetas tienen tamaño de fuente pequeño y negrita.

El botón de cada plan ocupa todo el ancho, tiene borde redondeado y texto en negrita.

4. Footer:

El pie de página tiene fondo gris oscuro, color blanco y un amplio relleno. Usa flexbox para organizar el contenido en varias columnas con un espacio del 18% entre ellas.

El logo del footer también se muestra con tamaño 50x50px. Las listas no muestran viñetas y los elementos tienen tamaño de fuente pequeño (10px). Los títulos dentro del footer tienen un tamaño de 15px.

5. Responsividad para celular:

Cuando la pantalla tiene un ancho máximo de 600px, el footer cambia a una disposición vertical, centra todos los elementos, reduce el relleno y ajusta el tamaño de fuentes. Esto hace que sea más legible y ordenado en móviles.

6. Animaciones y efectos:

Las tarjetas de los planes tienen una animación de aparición suave que incrementa la opacidad al cargarse la página.

Al pasar el mouse sobre una tarjeta, esta se agranda ligeramente y aparece una sombra para dar efecto de elevación.

Los botones dentro de las tarjetas también tienen una transición suave que los hace aumentar un poco de tamaño y disminuir un poco la opacidad al pasar el mouse.

Explicación de main.js

1. Menú hamburguesa

La primera parte controla el menú que se abre y cierra en móviles.

- Usa `querySelector` para obtener el nav, el botón de abrir y el de cerrar.
- Cuando haces click en "abrir", se agrega la clase "activo" al menú, se oculta el botón de abrir y aparece el de cerrar.
- Cuando haces click en "cerrar", se quita "activo", se oculta el de cerrar y vuelve el de abrir.

2. Cuando la página carga

El `window.onload` hace tres cosas:

- Carga las canciones/artistas iniciales.
- Configura el buscador.
- Muestra las canciones que guardaste en favoritos.

3. `cargarInicio()`

Carga contenido por defecto:

- Primera fila: canciones de Eminem.
- Segunda fila: artistas populares.
- Tercera fila: canciones de Shakira.

Cada una llama a `buscarYMostrar`.

4. `buscarYMostrar(nombre, contenedor)`

- Hace una búsqueda en la API de Deezer con el nombre enviado.
- Si el contenedor es el de artistas (fila 2), trae más resultados.
- Cuando recibe la respuesta:
 - En la fila 2 muestra artistas.
 - En las otras filas muestra canciones.

5. `mostrarCanciones(lista, contenedor)`

- Limpia el contenedor.
- Crea máximo 7 tarjetas de canciones.
- Cada tarjeta contiene:
 - Imagen del álbum.
 - Nombre de la canción.
 - Nombre del artista.
- Al hacer clic:
 - Reproduce la canción.
 - Muestra letra e info en el sidebar.

6. reproducirCancion(cancion)

- Crea un reproductor si no existe.
- Muestra:
 - Foto del álbum.
 - Título.
 - Artista.
 - Audio con preview.
 - Botón para agregar a favoritos.
- Lo hace visible en la pantalla.

7. configurarBuscador()

- Detecta cuando presionas Enter.
- Si está vacío, vuelve al inicio.
- Si hay texto, busca todo.
- También funciona si haces clic en el botón de buscar.

8. buscarTodo()

Actualiza los títulos de las filas para mostrar resultados.

Hace una búsqueda más grande (50 resultados) y reparte así:

- Fila 1: primeras 7 canciones.
- Fila 2: artistas relacionados (extraídos de todas las canciones).
- Fila 3: canciones de la posición 14 a la 21.

9. ponerTitulosNormales()

Devuelve los títulos de las filas a su estado inicial:

- Canciones en tendencia.
- Artistas populares.
- Álbumes y sencillos.

10. mostrarInfoEnSidebar(cancion)

- Muestra el panel derecho.
- Obtiene información del artista.
- Obtiene la letra de la canción.

11. obtenerInfoArtista(idArtista)

- Llama a la API de Deezer.
- Coloca:
 - Foto del artista.
 - Número de seguidores.
 - Número de álbumes.
 - Texto resumen.

12. cerrarSidebar()

- Oculta el panel lateral derecho.

13. Scroll horizontal de las filas

Cuando el documento carga:

- Cada fila obtiene botones de izquierda y derecha.
- Al hacer clic se mueve el scroll horizontalmente.
- También oculta o muestra los botones según corresponda.

14. agregarAFavoritos(cancion)

- Toma la lista de favoritos de localStorage.
- Si la canción ya existe, muestra alerta.
- Si no, la agrega y actualiza la biblioteca.

15. mostrarFavoritos()

- Si la librería está vacía, muestra un mensaje.
- Si hay canciones, las coloca con:
 - Imagen
 - Título
 - Artista
 - Botón para eliminar
- Si haces clic en la canción (no en el basurero), la reproduce buscando en Deezer.

16. eliminarFavorito(index)

- Quita la canción de favoritos.
- Actualiza la biblioteca.

17. buscarYReproducirFavorita(busqueda)

- Busca una canción en Deezer con título + artista.

- Reproduce la primera que encuentre.

18. mostrarArtistas(listaCanciones, contenedor)

- Extrae artistas únicos (máximo 7).
- Crea tarjetas de artistas con:
 - Foto
 - Nombre
 - Texto "Artista"
- Al hacer clic:
 - Muestra la info del artista en el sidebar.

19. mostrarInfoArtistaEnSidebar(id)

- Solo muestra en el sidebar la info del artista.
- Limpia la parte de la letra.

20. obtenerLetraCancion(artista, titulo)

- Limpia caracteres especiales.
- Usa la API de Lyrics.ovh.
- Si la encuentra, la muestra en el sidebar.
- Si falla, muestra un mensaje.

21. mostrarLetraEnSidebar(letra)

- Inserta la letra convertida a saltos de línea HTML dentro del div correspondiente.

22. Botones para cambiar de página (inicio y premium)

Cuando se hace clic, redirigen a:

- Página principal.
- Página premium.

23. Buscador en móviles (pantallas pequeñas)

En pantallas menores a 550px:

- El botón “buscar” en versión móvil hace que el buscador se abra o se oculte.

Explicación de login.js

1. Registro: Guarda el email y la contraseña en localStorage y redirige al login.

Login: Comprueba el email y contraseña guardados; si son correctos, marca la sesión como iniciada y guarda iniciales del usuario, luego redirige a la página principal.

2. Iniciales de usuario: Extrae las dos primeras letras del nombre antes de la “@” o las primeras letras si hay un punto.

3. Formulario de ayuda: Guarda los problemas enviados en localStorage y muestra un mensaje de confirmación.

4. Botones de planes: Antes de seleccionar un plan, verifica si el usuario está logueado; si no, redirige al login. Si está logueado, guarda el plan seleccionado en localStorage y muestra un mensaje de confirmación.

Explicación de usuario-avatar.js

Este script se encarga de transformar el botón de “Iniciar sesión” en un avatar de usuario y mostrar un menú con opciones cuando el usuario está logueado. Breve explicación:

- **Verifica sesión:** Comprueba si is_logged_in está en localStorage.
- **Transforma el botón:** Si el usuario está logueado, reemplaza el botón de iniciar sesión por un círculo con las iniciales del usuario y un estilo de avatar.
- **Interacción con el avatar:** Al hacer clic, muestra un menú desplegable con: avatar grande, email del usuario, plan seleccionado, botón para cerrar sesión y botón para cerrar el menú.
- **Cerrar sesión:** Elimina los datos de usuario de localStorage y recarga la página.

- **Animaciones:** Aplica animación al aparecer el menú y efectos de escala al pasar el cursor sobre el avatar.
- **Iniciales:** Calcula las iniciales a partir del email si no están guardadas.

Instalación de proyecto

La sección de Instalación del proyecto explica cómo descargar y ejecutar tu proyecto desde GitHub. Para tu URL, se puede explicar así de manera clara y breve:

- Clonar el repositorio:
Abre la terminal o consola y ejecuta:
`git clone https://github.com/cristhianlaimearcani-jpg/PROYECTO-SPOTIFY-PW1.git`
Esto descargará todo el proyecto en tu computadora.
- Entrar en la carpeta del proyecto:
`cd PROYECTO-SPOTIFY-PW1`
Esto te sitúa dentro del proyecto para poder trabajar con los archivos.
- Abrir el proyecto en un editor de código:
Por ejemplo, en VS Code:
`code .`
- Ejecutar la página:
Como tu proyecto es una web estática, solo abre el archivo Proyecto PW1.html en un navegador (doble clic o desde el editor).

Manual de Mantenimiento

El mantenimiento del proyecto debe enfocarse en asegurar su correcto funcionamiento y la actualización de sus componentes. Se recomienda:

- **Actualización de dependencias:** Verificar periódicamente las claves de la API de Deezer y cualquier librería utilizada para garantizar su compatibilidad.
- **Archivos del proyecto:** Mantener organizados los archivos HTML, CSS y JS; no renombrar ni eliminar archivos esenciales.

- **RespalDOS:** Guardar copias de seguridad del proyecto y de los datos almacenados en localStorage si se requiere conservar información de usuarios.
- **Depuración:** Utilizar la consola del navegador para identificar errores y asegurar que todos los elementos interactivos funcionen correctamente.
- **Documentación:** Mantener comentarios en el código y actualizar este manual cuando se agreguen nuevas funcionalidades o cambios relevantes.

Requisitos del sistema

Para un correcto funcionamiento del proyecto se requieren los siguientes elementos:

- **Hardware mínimo:** Computadora o laptop con capacidad para ejecutar un navegador web moderno.
- **Software:**
 - Navegadores compatibles: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari.
 - Editor de código recomendado: Visual Studio Code o similar.
- **Conexión a Internet:** Necesaria para acceder a la API de Deezer y reproducir canciones.
- **Extras opcionales:** Extensión Live Server para desarrollo local y pruebas de las funciones que usan fetch o AJAX.

Conclusión

El proyecto “PROYECTO PW1” permite a los usuarios explorar y reproducir canciones mediante la API de Deezer, gestionar favoritos y seleccionar planes de suscripción de manera interactiva. Su diseño facilita la instalación, el mantenimiento y la actualización, garantizando compatibilidad con navegadores modernos y ofreciendo una experiencia de usuario intuitiva. La correcta documentación y mantenimiento asegurarán que el proyecto siga funcionando eficientemente a lo largo del tiempo.