|  |  |
| --- | --- |
| UNIDAD 1:  Usabilidad y Accesibilidad | |
| **Desarrollo de Interfaces** | |
|  | |
| Autor: Sergio Ramos Torres  Titulación: Desarrollo de Aplicaciones Multiplataforma  Módulo profesional: Desarrollo de Interfaces | |
| Fecha: dd/mm/yyyy | Versión: 0 ~0 |

Contenido

[1. Análisis Heurístico de GitKraken 3](#_Toc150625689)

[3](#_Toc150625690)

[1.1Visibilidad del estado de sistema 4](#_Toc150625691)

[1.2 Consistencia entre sistema y mundo real: orden lógico, lenguaje familiar 6](#_Toc150625692)

[1.3 Control del usuario: abandonar en cualquier momento, deshacer o repetir una acción 6](#_Toc150625693)

[1.4 Consistencia y estándares: lenguaje coherente 7](#_Toc150625694)

[1.5 Prevención de errores 7](#_Toc150625695)

[1.6 Es mejor reconocer que recordar: objetos, acciones y opciones a la vista 7](#_Toc150625696)

[1.7 Flexibilidad y eficiencia de uso: preparado para satisfacer a usuarios novatos como a los usuarios experimentados 8](#_Toc150625697)

[1.8 Diseño práctico y sencillo 8](#_Toc150625698)

[1.9 El usuario debe disponer de ayuda para reconocer, diagnosticar y deshacer errores 9](#_Toc150625699)

[1.10 Ayuda y documentación 9](#_Toc150625700)

[2.Análisis heurístico de Steam 10](#_Toc150625701)

[2.1Visibilidad del estado del sistema. Usuario siempre informado 10](#_Toc150625702)

[2.2Consistencia entre sistema y mundo real: orden lógico, lenguaje familiar 11](#_Toc150625703)

[2.3Control del usuario: abandonar en cualquier momento, deshacer o repetir una acción 12](#_Toc150625704)

[2.4 Consistencia y estándares: lenguaje coherente 12](#_Toc150625705)

[2.5 Prevención de errores 12](#_Toc150625706)

[2.6 Es mejor reconocer que recordar: objetos, acciones y opciones a la vista 13](#_Toc150625707)

[2.7 Flexibilidad y eficiencia de uso: preparado para satisfacer a usuarios novatos como a los usuarios experimentados 13](#_Toc150625708)

[2.8 Diseño práctico y sencillo 13](#_Toc150625709)

[2.9 El usuario debe disponer de ayuda para reconocer, diagnosticar y deshacer errores 13](#_Toc150625710)

[2.10 Ayuda y documentación 13](#_Toc150625711)

# Análisis Heurístico de GitKraken

# 

Imagen de referencia

Este programa nace de la necesidad de una interfaz más comprensible y útil del uso de Git. Por lo tanto, se creó con la idea de amplificar la usabilidad y accesibilidad de este sistema de repositorios, que, aunque útil, es complicado para una multitud de usuarios que no están familiarizados ni quieren hacerlo con los terminales.

Partiendo de este punto voy a ir analizando paso a paso las directrices de Jakob Nielsen. En este caso, la aplicación está dirigida a un público mayoritariamente adulto dado que suple unas necesidades a un software que principalmente se utiliza en el panorama laboral. Aunque muchas personas lo utilizan en su día a día para repositorios privados, su uso es extendido en el mundo de la programación. Centraré así pues el análisis del programa a la población activa.

## 1.1Visibilidad del estado de sistema

En este apartado voy a destacar dos partes del programa que encapsulan este punto.

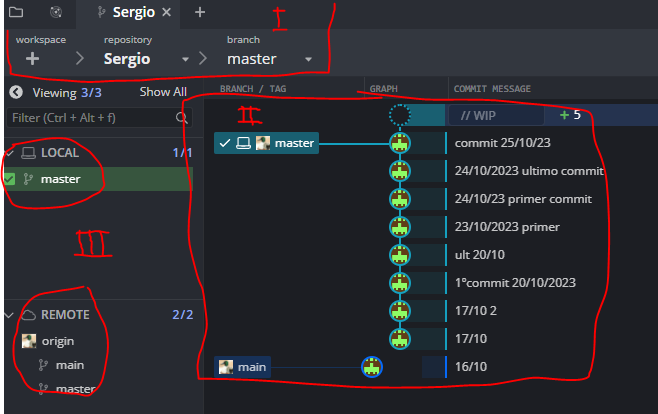


Imagen de referencia 2

En esta primera parte que comprende la sección izquierda de la aplicación podemos comprobar rápidamente tres cosas:

1. El usuario que está conectado al programa y la rama en la que se está trabajando principalmente.
2. El árbol de ramas y los diferentes commits que se han ido haciendo, con sus correspondientes mensajes. En este caso vemos dos ramas y que master es la que se ha ido actualizando.
3. En esta zona podemos vemos que en el local tenemos una sola rama descargada y en el remoto (origin) las dos ramas que hemos visualizado antes.

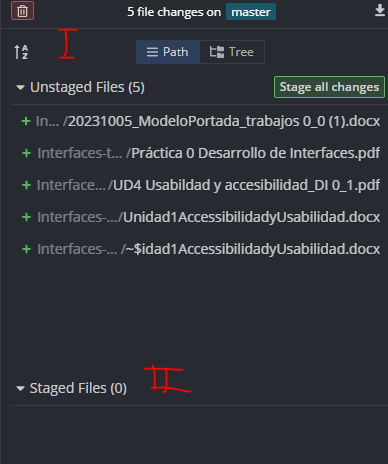


Imagen de referencia 3

En esta segunda parte situada a la derecha de la aplicación observamos que:

1. Aquí tenemos la zona de no preparados pudiendo stagear los cambios.
2. Y aquí la zona donde ya están los preparados con un botón para hacer los commits.

Atendiendo a las razones del primer principio, un usuario que viene de usar git, lo va a encontrar todo a primera vista sin problema.

## 1.2 Consistencia entre sistema y mundo real: orden lógico, lenguaje familiar

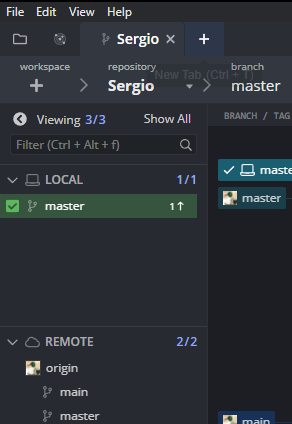


Imagen de referencia 4

El orden lógico sigue las convenciones de la mayoría de los programas, es decir tiene el sistema de directorios arriba a la izquierda, la barra de exploración a la izquierda también.

El lenguaje será familiar sólo si conoces git, dado que es una aplicación dirigida a personas que busquen simplificar su trabajo con este software. Por lo tanto, este punto es negativo en ese aspecto.

## 1.3 Control del usuario: abandonar en cualquier momento, deshacer o repetir una acción

Mediante la barra de tareas el usuario es capaz de controlar las acciones de la aplicación sin mucho esfuerzo.



Imagen de referencia 5

El problema es que no se puede abandonar en cualquier momento debido a la misma naturaleza del software git, en el que hay que seguir unos pasos (aunque rápidos en la aplicación) que si salimos directamente no guardará el trabajo hecho.

## 1.4 Consistencia y estándares: lenguaje coherente

El lenguaje es coherente con el proceso de git, sino se conoce el uso del software el usuario probablemente no entienda lo que muchas herramientas significan, además el programa está en inglés.

## 1.5 Prevención de errores

El programa cuenta con varias herramientas para tratar errores.

|  |
| --- |
|  |
|  |
| Imagen de referencia 6 |

Pero no parece contar con un sistema de prevención.

## 1.6 Es mejor reconocer que recordar: objetos, acciones y opciones a la vista

En este caso el programa es muy similar a cualquier otro entorno de diseño o codificación, las herramientas se encuentran en los sitios usuales etc. Pero una vez más aparte de la barra de tareas, la mayoría del resto de términos probablemente confundan a un usuario que no tenga experiencia con el software.

## 1.7 Flexibilidad y eficiencia de uso: preparado para satisfacer a usuarios novatos como a los usuarios experimentados

Como ya he ido explicando en apartados anteriores es necesario un conocimiento previo del software git para poder manejar esta aplicación. Un usuario que empieza desde cero no sabrá que el uso de este software va ligado al uso de la descarga de git ni de la localización de los repositorios.

Se pone al alcance de la mano todas las herramientas necesarias para trabajar sin tener que abrir la terminal, pero los usuarios novatos deberían comprender el funcionamiento de git de antemano.

## 1.8 Diseño práctico y sencillo

La aplicación consta de una serie de iconos y herramientas que, (conociendo los términos) no da lugar a ningún problema. Junto a el árbol de ramas se encuentra el botón para hacer stage de los cambios en la rama.



Imagen de referencia 7

Justo debajo está la ventana para poder escribir el mensaje del commit y hacerlo.

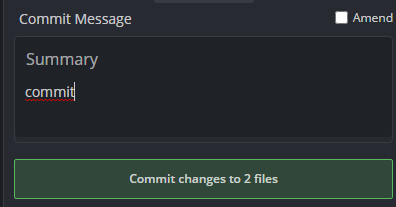


Imagen de referencia 8

Y arriba se encuentra la barra de tareas con la que ir haciendo el resto de acciones:

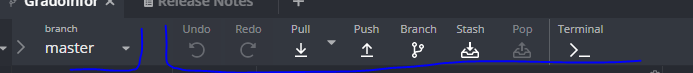


Imagen de referencia 9

## 1.9 El usuario debe disponer de ayuda para reconocer, diagnosticar y deshacer errores

Esta aplicación cuenta con un log para poder seguir los errores se han hecho capturas en el punto 1.5 pero las repito aquí:

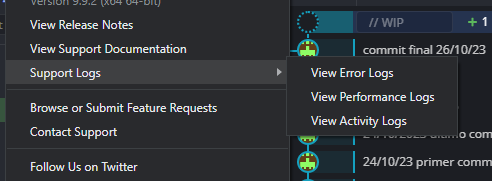


Imagen de referencia 10

## 1.10 Ayuda y documentación

Este programa cuenta con documentación a la que se puede acceder desde el mismo, en la pestaña de help. Además, se puede contactar para recibir ayuda más personalizada.

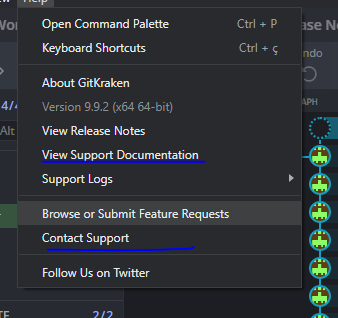


Imagen de referencia 11

# 2.Análisis heurístico de Steam



Imagen de referencia 12

Steam es una Plataforma de distribución digital de videojuegos creada en 2003 por Valve. Desde su nacimiento ha experimentado multitud de cambios y actualizaciones que la han llevado a ser el referente en cuanto a plataformas en el campo de los videojuegos en pc.

No solo es un Marketplace para las desarrolladoras de videojuegos para vender sus juegos, sino que ha creado una robusta comunidad detrás incidiendo en el aspecto social del mundo de juegos tanto online como singleplayer. El público al que va destinado es completamente heterogéneo, buscan alcanzar usuarios de todo tipo y edades. Una forma muy sencilla de conseguirlo es mediante los diferentes juegos que venden ya que los juegos mismos suelen filtrar la población activa de éstos según su género.

## 2.1Visibilidad del estado del sistema. Usuario siempre informado

La primera vez que se entra a la plataforma puede ser un poco abrumadora ya que lo primero que nos ofrece son descuentos y novedades.



Imagen de referencia 13

Pero está todo visible y al alcance del usuario mediante el menú que, sin importar por donde estemos scrolleando sigue fijo en los bordes superior:



Imagen de referencia 14

E inferior:



Imagen de referencia 15

## 2.2Consistencia entre sistema y mundo real: orden lógico, lenguaje familiar

El orden que sigue no es muy lógico en mi opinión ya que prioriza siempre las ventanas de tienda y ventas, cuando una plataforma de este estilo es interesante por la biblioteca de juegos que posee el usuario.

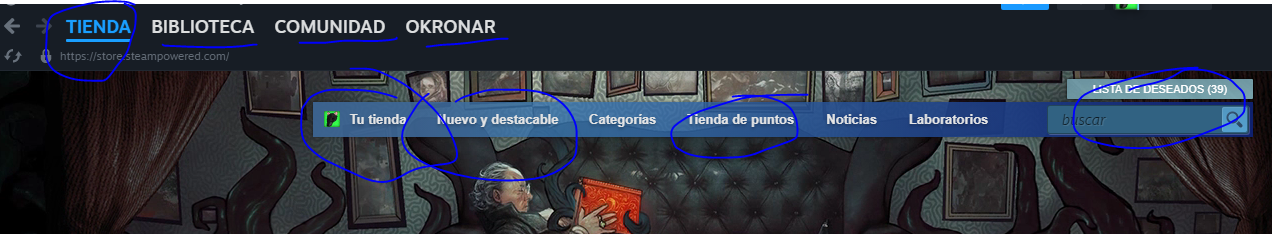


Imagen de referencia 16

Como se puede apreciar en la imagen lo primero que nos aparece al abrir la aplicación es la campaña de descuentos si hay alguna y el orden de la barra de tareas empieza por ofrecernos la tienda. Además, en esta primera bienvenida ya se nos enfoca a comprar mediante la barra de debajo.

El lenguaje es accesible para todos los públicos, dado que la aplicación no se lucra de la descarga de esta, sino de la venta de productos a través de ella.

## 2.3Control del usuario: abandonar en cualquier momento, deshacer o repetir una acción

Se puede deshacer tanto la última búsqueda hecha:

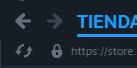


Imagen de referencia 17

Como volver a la pestaña donde se estaba previamente, además también se pueden deshacer los carritos de compra e incluso anular las compras si no ha pasado mucho tiempo.

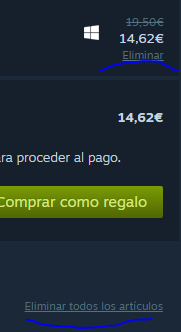


Imagen de referencia 18

## 2.4 Consistencia y estándares: lenguaje coherente

La aplicación se encuentra en multitud de idiomas y utiliza un lenguaje cómodo de entender.

## 2.5 Prevención de errores

En este caso no he podido encontrar un sistema de prevención de errores, es cierto que cada martes se realiza una actualización de la plataforma, pero aparte de esto, no se encuentra demasiada información al respecto.

## 2.6 Es mejor reconocer que recordar: objetos, acciones y opciones a la vista

Como he dicho antes Steam peca de saturación al usuario. Si el usuario es recurrente y utiliza la plataforma a diario no tendrá ningún problema a la hora de navegar entre las pestañas, pero si es un usuario novato joven o bastante mayor, es bastante probable que se vea sobresaturado con la cantidad de información presentada. Una persona que lo utiliza por primera vez y busca jugar a los juegos de su biblioteca no entenderá porque de primeras se ve transportado a la galería de la tienda ni la cantidad de títulos que se le ofrecen.

## 2.7 Flexibilidad y eficiencia de uso: preparado para satisfacer a usuarios novatos como a los usuarios experimentados

Creo que steam podría beneficiarse ampliamente de un tutorial para usuarios noveles dado que la cantidad de pestañas que hay para navegar incluso dentro de un mismo juego es alarmante.

## 2.8 Diseño práctico y sencillo

Me reitero en mis palabras anteriores, dado que Steam se lucra de la venta de los juegos en su plataforma, oraganiza cambios de interfaz cada muy poco tiempo que a usuarios noveles puede despistar muy fácilmente.

Una vez se acostumbra uno a donde están sus cosas(biblioteca) y como navegar por donde quiere se hace mucho más sencillo, pero hasta entonces puede dar lugar a confusión.

## 2.9 El usuario debe disponer de ayuda para reconocer, diagnosticar y deshacer errores

Aunque no parezca haber un sistema de prevención, si que lo hay de informe de errores si sucede alguno, con un correspondiente mensaje de que hacer una vez el error ha aparecido.

## 2.10 Ayuda y documentación

Steam posee información acerca de los temas legales al comprar productos etc, pero no tiene una documentación sobre como manejarse en la aplicación. Lo que si que posee es un servicio de soporte tanto automatizado como personalizado llegado a necesitarse el caso.



Imagen de referencia 19

# 3.Analisis de accesibilidad Spotify Web

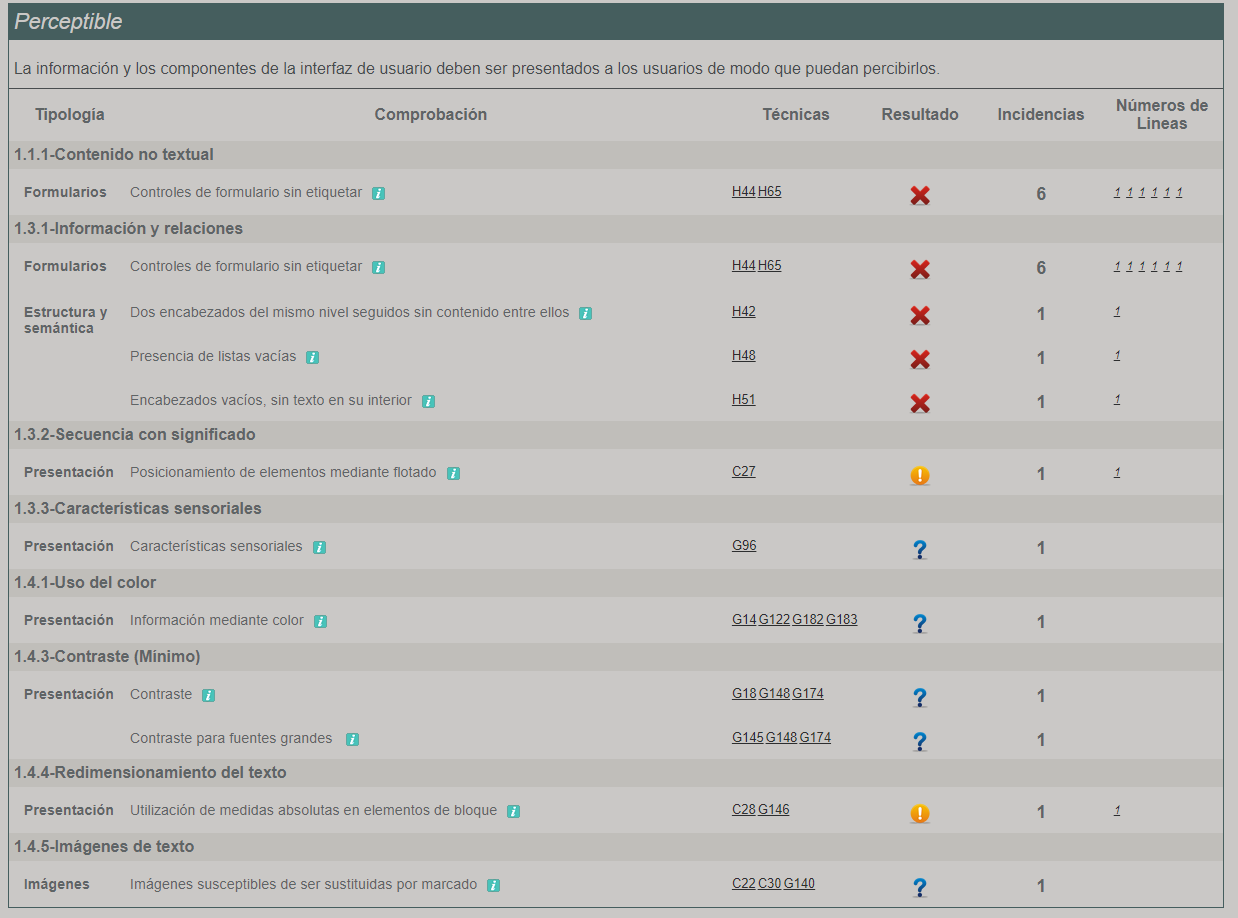


Imagen de referencia



Imagen de referencia

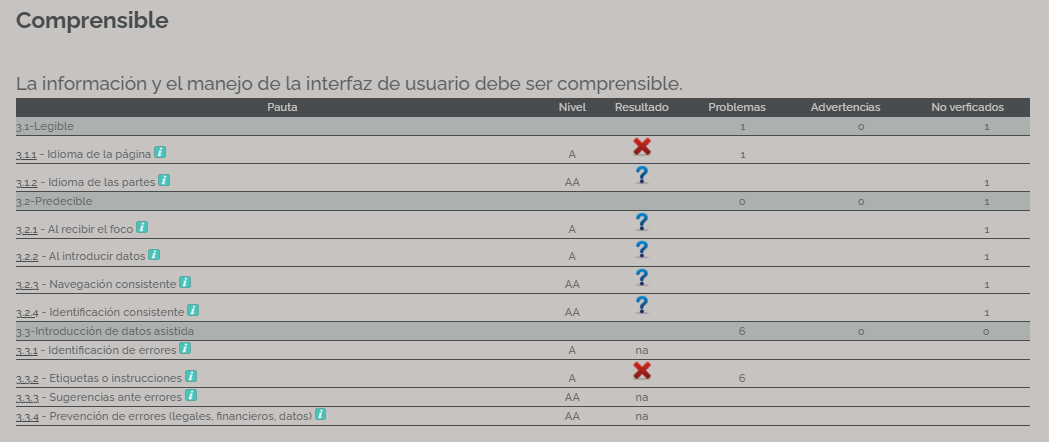


Imagen de referencia

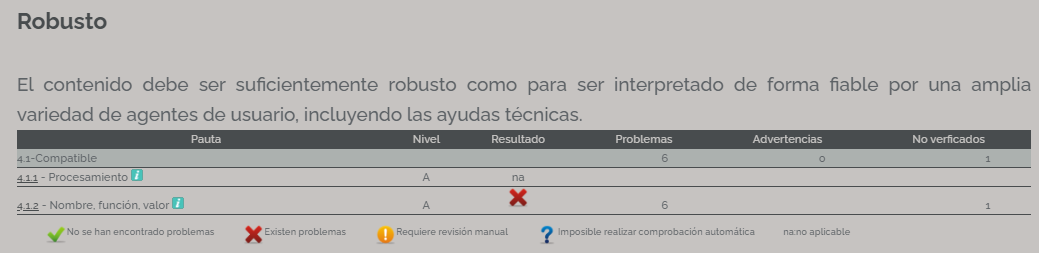


Imagen de referencia

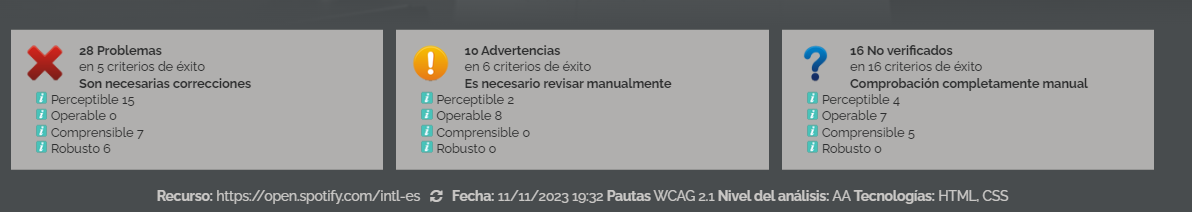
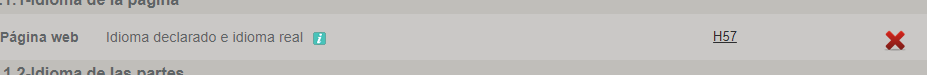


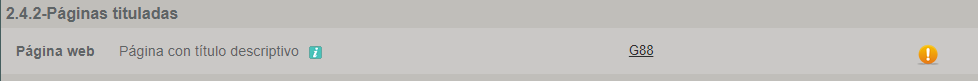
Imagen de referencia 24

Como se puede comprobar en la imagen 24, se han encontrado 28 problemas de los cuales 15 son de perceptibilidad, 7 de tipo compresible y 6 de robustez.

El problema de la siguiente imagen salta porque aunque se ponga “.en” se actualiza automáticamente al idioma de la región en la que estemos. Con lo cual habrá personas que buscando inglés se encuentren español, por ejemplo.



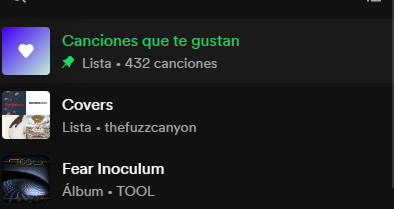
Tawdis marca también posibles problemas con el título de la página web:



Supongo que esto se debe a que el título es bastante largo y no se ve en la pestaña.



Uno de los problemas de perceptibilidad que detecta Tawdis tiene que ver con el uso de colores en la página web a la hora de transmitir información. Según la siguiente imagen:



Podemos comprobar que la lista seleccionada es la de canciones que te gustan, sin embargo, de no ser por el color verde no existe ningún otro elemento identificativo ya que el símbolo del pin simplemente significa que está anclada a la ventana de inicio. Esto puede ocasionar a usuarios que padezcan daltonismo una perdida grave de información. Con un simple subrayado o un tachado se puede mejorar la visibilidad de esta característica.

|  |
| --- |
|  |
|  |
| Sergio Ramos Torres  Desarrollo de Aplicaciones Multiplataforma  Desarrollo de Interfaces  2023-2024 |