

# UNIDAD 1:

## Usabilidad y Accesibilidad

### Desarrollo de Interfaces

---



Autor: Sergio Ramos Torres

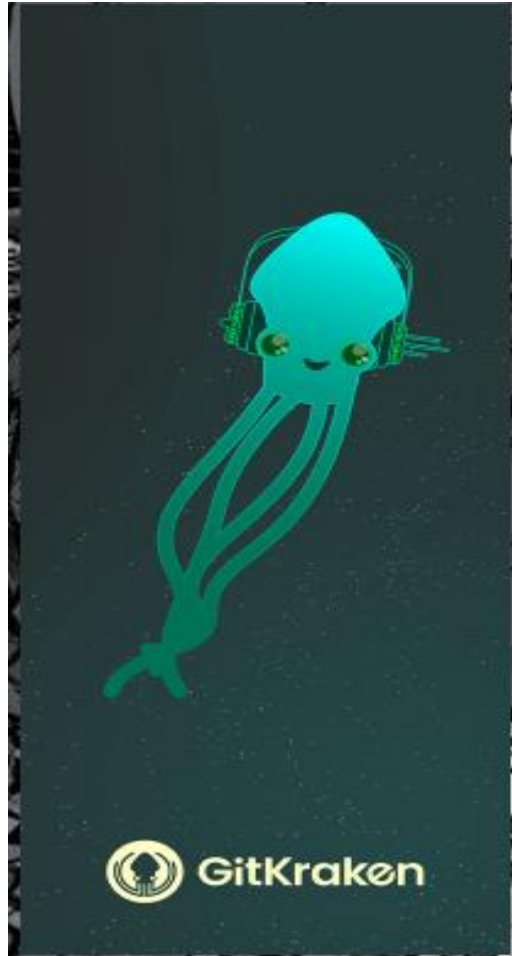
Titulación: Desarrollo de Aplicaciones Multiplataforma  
Módulo profesional: Desarrollo de Interfaces

Fecha: 17/11/2023	Versión: 1
-------------------	------------

# Contenido

1. Análisis Heurístico de GitKraken.....	3
1.1 Visibilidad del estado de sistema .....	4
1.2 Consistencia entre sistema y mundo real: orden lógico, lenguaje familiar.....	6
1.3 Control del usuario: abandonar en cualquier momento, deshacer o repetir una acción ...	7
1.4 Consistencia y estándares: lenguaje coherente .....	7
1.5 Prevención de errores .....	7
1.6 Es mejor reconocer que recordar: objetos, acciones y opciones a la vista .....	8
1.7 Flexibilidad y eficiencia de uso: preparado para satisfacer a usuarios novatos como a los usuarios experimentados .....	8
1.8 Diseño práctico y sencillo .....	9
1.9 El usuario debe disponer de ayuda para reconocer, diagnosticar y deshacer errores.....	10
1.10 Ayuda y documentación .....	10
2. Análisis heurístico de Steam.....	12
2.1 Visibilidad del estado del sistema. Usuario siempre informado .....	12
2.2 Consistencia entre sistema y mundo real: orden lógico, lenguaje familiar.....	14
2.3 Control del usuario: abandonar en cualquier momento, deshacer o repetir una acción..	14
2.4 Consistencia y estándares: lenguaje coherente .....	15
2.5 Prevención de errores .....	15
2.6 Es mejor reconocer que recordar: objetos, acciones y opciones a la vista .....	16
2.7 Flexibilidad y eficiencia de uso: preparado para satisfacer a usuarios novatos como a los usuarios experimentados .....	16
2.8 Diseño práctico y sencillo .....	16
2.9 El usuario debe disponer de ayuda para reconocer, diagnosticar y deshacer errores.....	16
2.10 Ayuda y documentación .....	16
3. Análisis de accesibilidad Spotify Web .....	18
4. Análisis de accesibilidad Yale school of art .....	23
5. Conclusión .....	28
6. Bibliografía .....	28

# 1. Análisis Heurístico de GitKraken

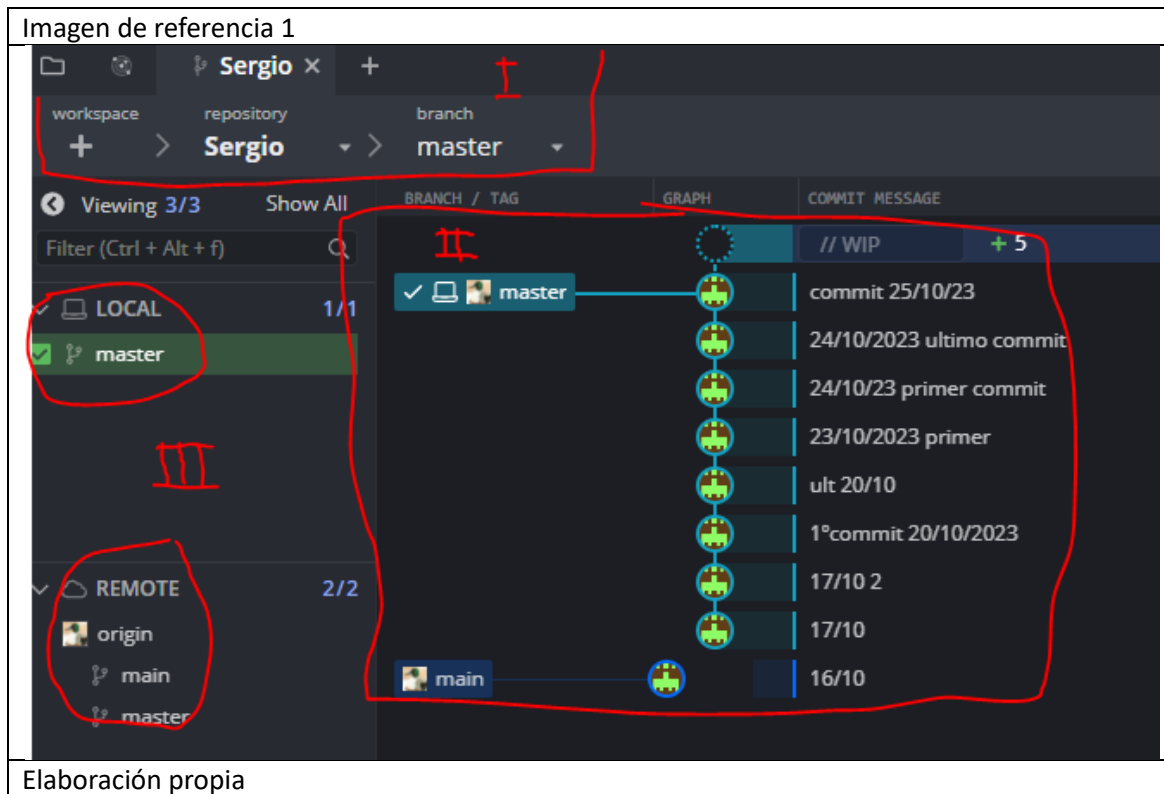


Este programa nace de la necesidad de una interfaz más comprensible y útil del uso de Git. Por lo tanto, se creó con la idea de amplificar la usabilidad y accesibilidad de este sistema de repositorios, que, aunque útil, es complicado para una multitud de usuarios que no están familiarizados ni quieren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

## 1.1 Visibilidad del estado de sistema

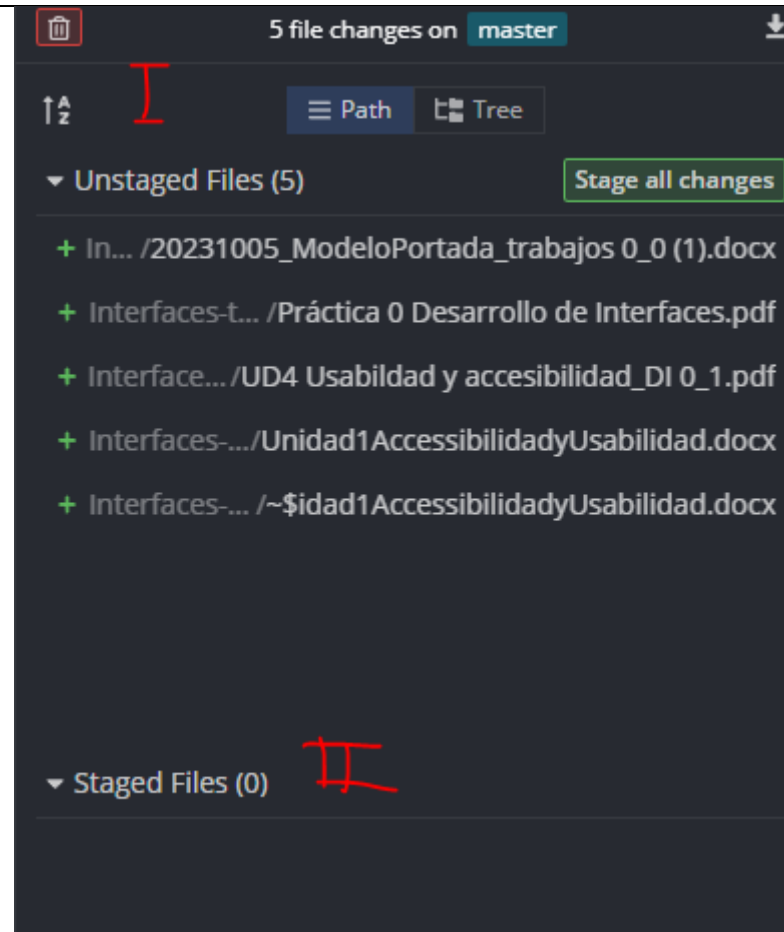
En este apartado voy a destacar dos partes del programa que encapsulan este punto.



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

- I. El usuario que está conectado al programa y la rama en la que se está trabajando principalmente.
- II. 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.
- III. En esta zona podemos ver que en el local tenemos una sola rama descargada y en el remoto (origin) las dos ramas que hemos visualizado antes.

Imagen de referencia 2



Elaboración propia

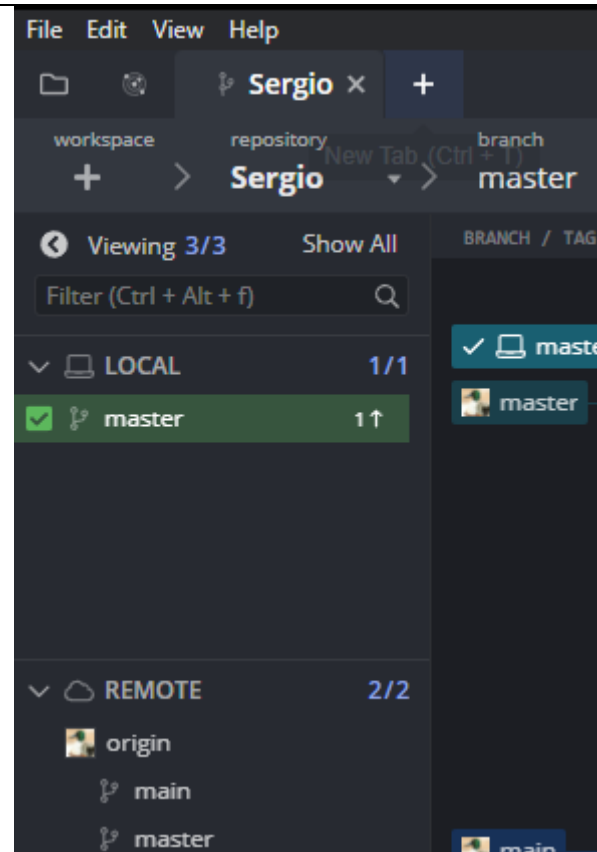
En esta segunda parte situada a la derecha de la aplicación observamos que:

- I. Aquí tenemos la zona de no preparados pudiendo stagear los cambios.
- II. 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 3



Elaboración propia

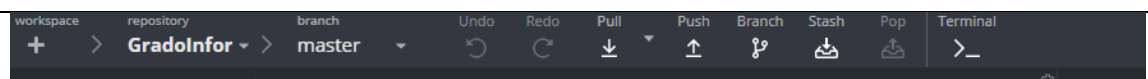
El orden lógico sigue las convenciones de la mayoría de los programas, es decir tiene el sistema de directorios arriba a la izquierda, la barra de exploración a la izquierda también.

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 4*



Elaboración propia

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.

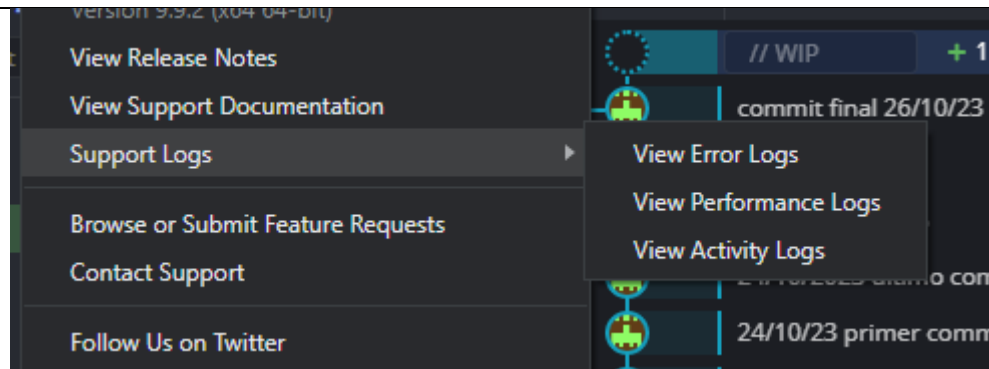
## 1.4 Consistencia y estándares: lenguaje coherente

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, analizar y visualizar errores.

Imagen de referencia 5



Elaboración Propia

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.

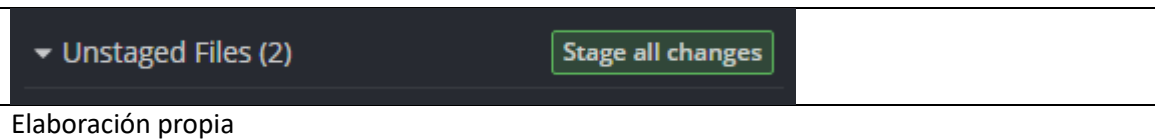
Se pone al alcance de la mano todas las herramientas necesarias para trabajar sin tener que abrir la terminal, pero los usuarios novatos deberían comprender el funcionamiento de git de antemano.



## 1.8 Diseño práctico y sencillo

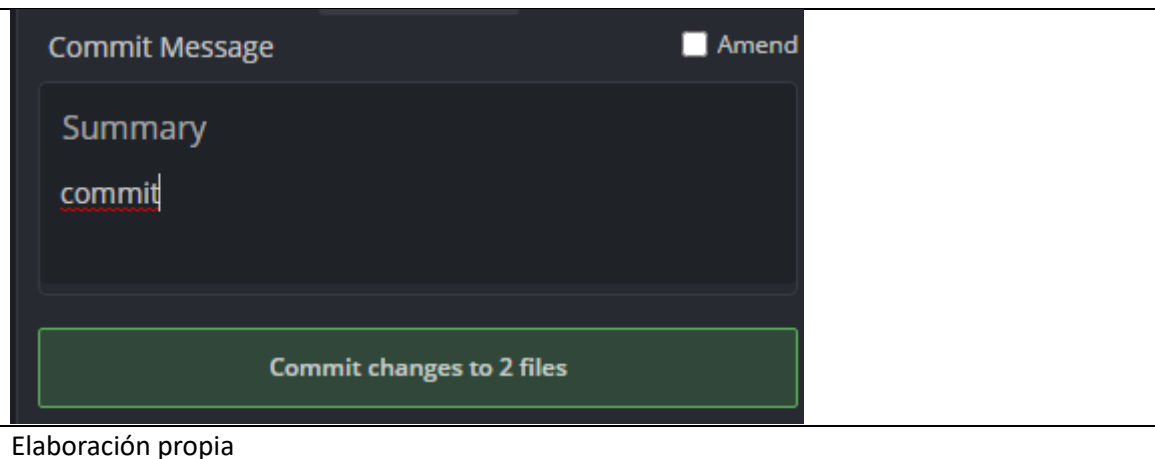
La aplicación consta de una serie de iconos y herramientas que, (conociendo los términos) no da lugar a ningún problema. Junto a el árbol de ramas se encuentra el botón para hacer stage de los cambios en la rama.

Imagen de referencia 6



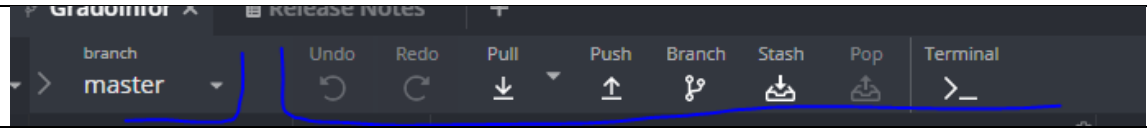
Justo debajo está la ventana para poder escribir el mensaje del commit y hacerlo.

Imagen de referencia 7



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

Imagen de referencia 8

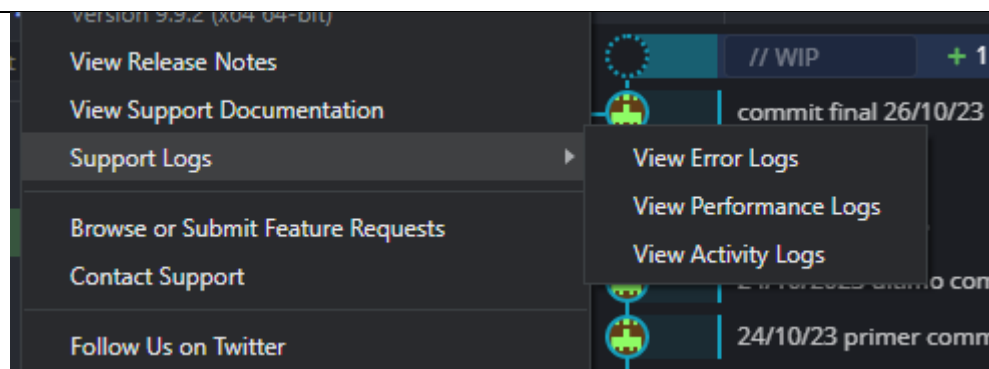


Elaboración propia

## 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 9

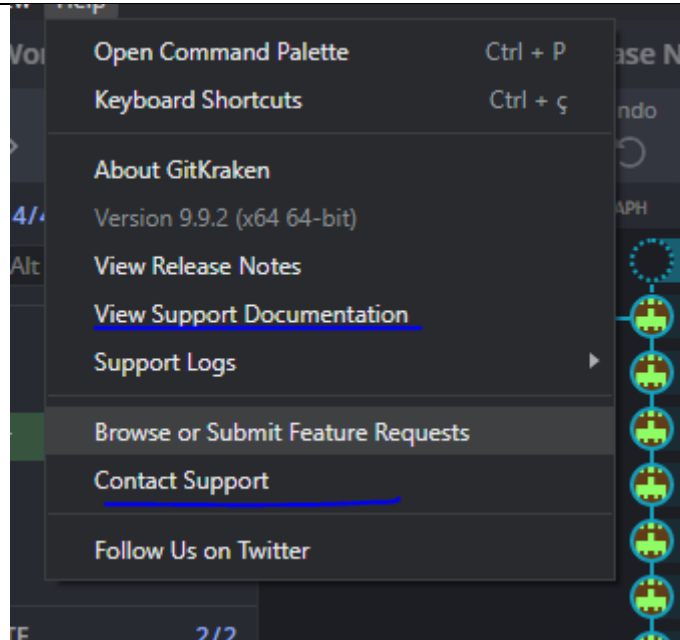


Elaboración propia

## 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 10



Elaboración propia

## 2.Análisis heurístico de Steam

Imagen de referencia 11



Elaboración propia

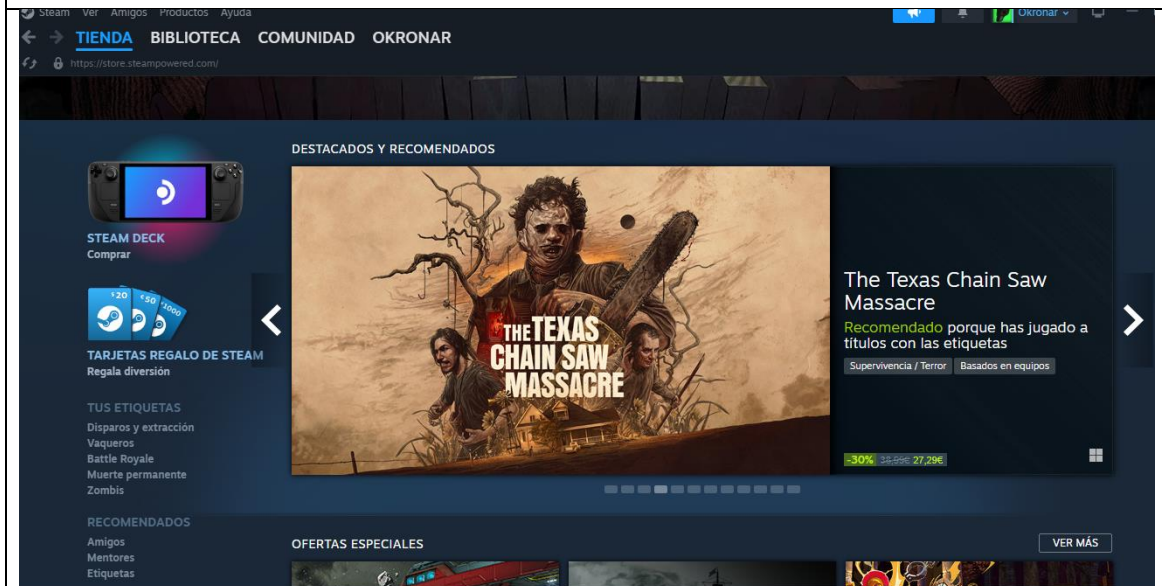
Steam es una Plataforma de distribución digital de videojuegos creada en 2003 por Valve. Desde su nacimiento ha experimentado multitud de cambios y actualizaciones que la han llevado a ser el referente en cuanto a plataformas en el campo de los videojuegos en pc.

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.

### 2.1 Visibilidad 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 12



Elaboración propia

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:

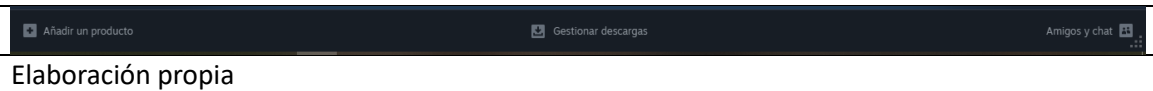
Imagen de referencia 13



Elaboración propia

E inferior:

Imagen de referencia 14

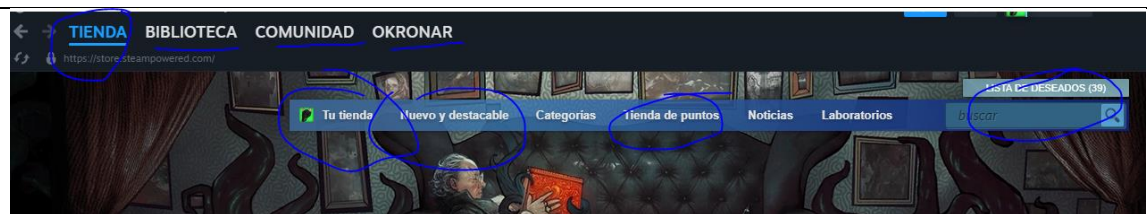


Elaboración propia

## 2.2 Consistencia 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 15



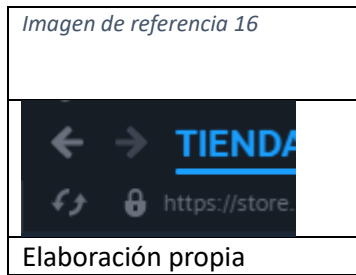
Elaboración propia

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.

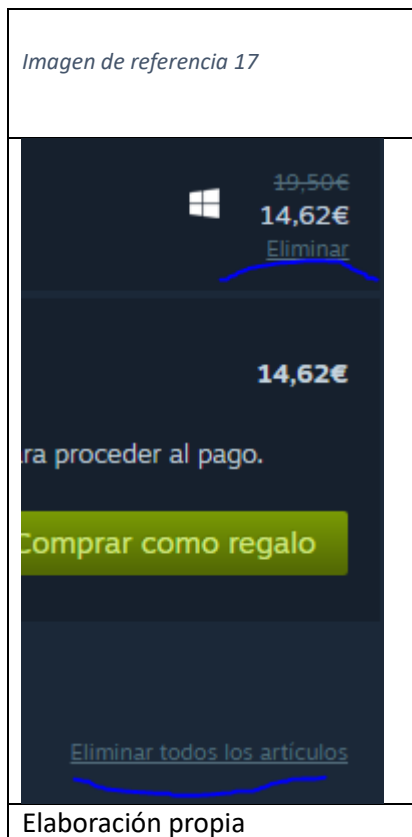
El lenguaje es accesible para todos los públicos, dado que la aplicación no se lucra de la descarga de esta, sino de la venta de productos a través de ella.

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

Se puede deshacer tanto la última búsqueda hecha:



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.



## 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, organiza 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

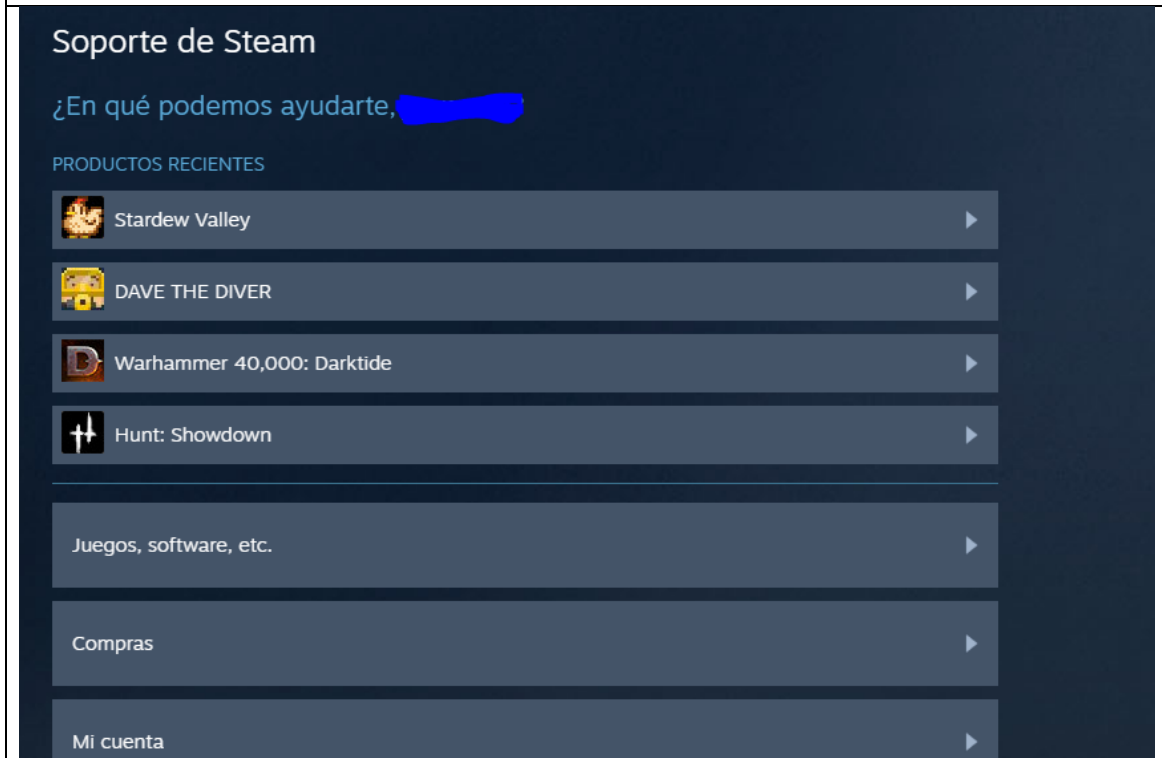
Aunque no parezca haber un sistema de prevención, si que lo hay de informe de errores si sucede alguno, con un correspondiente mensaje de que hacer una vez el error ha aparecido.

## 2.10 Ayuda y documentación

Steam posee información acerca de los temas legales al comprar productos etc, pero no tiene una documentación sobre cómo 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 18



Elaboración propia

### 3. Análisis de accesibilidad Spotify Web

Imagen de referencia 19






Perceptible						
La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que puedan percibirlos.						
Tipología	Comprobación	Técnicas	Resultado	Incidencias	Números de Líneas	
1.1.1-Contenido no textual						
Formularios	Controles de formulario sin etiquetar	H44 H65	✗	6	1 1 1 1 1 1	
1.3.1-Información y relaciones						
Formularios	Controles de formulario sin etiquetar	H44 H65	✗	6	1 1 1 1 1 1	
Estructura y semántica	Dos encabezados del mismo nivel seguidos sin contenido entre ellos	H42	✗	1	1	
	Presencia de listas vacías	H48	✗	1	1	
	Encabezados vacíos, sin texto en su interior	H51	✗	1	1	
1.3.2-Secuencia con significado						
Presentación	Posicionamiento de elementos mediante flotado	C27	!	1	1	
1.3.3-Características sensoriales						
Presentación	Características sensoriales	G96	?	1		
1.4.1-Uso del color						
Presentación	Información mediante color	G14 G122 G182 G183	?	1		
1.4.3-Contraste (Mínimo)						
Presentación	Contraste	G18 G148 G174	?	1		
	Contraste para fuentes grandes	G145 G148 G174	?	1		
1.4.4-Redimensionamiento del texto						
Presentación	Utilización de medidas absolutas en elementos de bloque	C28 G146	!	1	1	
1.4.5-Imágenes de texto						
Imágenes	Imágenes susceptibles de ser sustituidas por marcado	C22 C30 G140	?	1		

Elaboración propia

## Imagen de referencia 20

## Operable

Los componentes de la interfaz de usuario y la navegación deben ser operables.

Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
2.1-Accesible mediante el teclado			0	0	1
2.1.1 - Teclado 	A	?			1
2.1.2 - Sin bloqueos de teclado 	A	?			1
2.2-Tiempo suficiente			0	0	1
2.2.1 - Tiempo ajustable 	A	?			1
2.2.2 - Pausar, detener, ocultar 	A	?			1
2.3-Provocar ataques			0	0	1
2.3.1 - Umbral de tres destellos o menos 	A	?			1
2.4-Navegable			0	8	5
2.4.1 - Evitar bloques 	A	!		1	2
2.4.2 - Páginas tituladas 	A	!		1	
2.4.3 - Orden del foco 	A	!		1	1
2.4.4 - Propósito de los enlaces (en contexto) 	A	✓			
2.4.5 - Múltiples vías 	AA	?			1
2.4.6 - Encabezados y etiquetas 	AA	!		5	
2.4.7 - Foco visible 	AA	?			1

Elaboración propia

## Imagen de referencia 21

## Comprensible

La información y el manejo de la interfaz de usuario debe ser comprensible.

Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
3.1-Legible			1	0	1
3.1.1 - Idioma de la página 	A	✗	1		
3.1.2 - Idioma de las partes 	AA	?			1
3.2-Predecible			0	0	1
3.2.1 - Al recibir el foco 	A	?			1
3.2.2 - Al introducir datos 	A	?			1
3.2.3 - Navegación consistente 	AA	?			1
3.2.4 - Identificación consistente 	AA	?			1
3.3-Introducción de datos asistida			6	0	0
3.3.1 - Identificación de errores 	A	na			
3.3.2 - Etiquetas o instrucciones 	A	✗	6		
3.3.3 - Sugerencias ante errores 	AA	na			
3.3.4 - Prevención de errores (legales, financieros, datos) 	AA	na			

Elaboración propia

Imagen de referencia 22

Robusto					
El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de agentes de usuario, incluyendo las ayudas técnicas.					
Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
4.1-Compatible			6	0	1
4.1.1 - Procesamiento	A	na			
4.1.2 - Nombre, función, valor	A		6		1
No se han encontrado problemas            Existen problemas            Requiere revisión manual            Imposible realizar comprobación automática           na: no aplicable					

Elaboración propia

Imagen de referencia 23

<b>28 Problemas</b> en 5 criterios de éxito <b>Son necesarias correcciones</b> Perceptible 15 Operable 0 Comprensible 7 Robusto 6	<b>10 Advertencias</b> en 6 criterios de éxito <b>Es necesario revisar manualmente</b> Perceptible 2 Operable 8 Comprensible 0 Robusto 0	<b>16 No verificados</b> en 16 criterios de éxito <b>Comprobación completamente manual</b> Perceptible 4 Operable 7 Comprensible 5 Robusto 0
Recurso: <a href="https://open.spotify.com/intl-es">https://open.spotify.com/intl-es</a> Fecha: 11/11/2023 19:32 Pautas WCAG 2.1 Nivel del análisis: AA Tecnologías: HTML, CSS		

Elaboración propia

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

Imagen de referencia 25


1.1 Idioma de la página		
Página web	Idioma declarado e idioma real	H57
1.2 Idioma de las partes		

Elaboración propia

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

Imagen de referencia 26

## 2.4.2-Páginas tituladas

Página web    Página con título descriptivo 



G88



Elaboración propia

Supongo que esto se debe a que el título es bastante largo y no se ve en la pestaña, la solución que yo propongo es simple, reducir el máximo el título poniendo algo como Music:Spotify yo al revés. De esta manera se ve claramente qué es y la temática de la web.

Imagen de referencia 27

 Reproductor web de Spotify: música 

Elaboración propia

Otro 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:

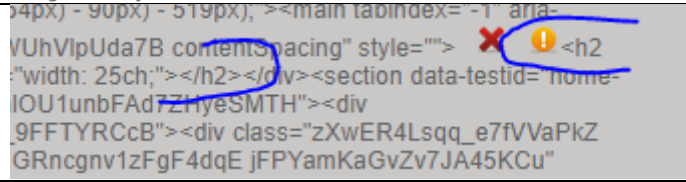
Imagen de referencia 28



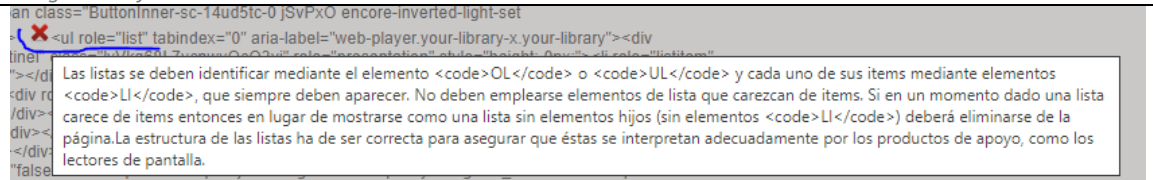
Elaboración propia

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 pérdida grave de información. Con un tachado o puesto en cursiva se puede mejorar la visibilidad de esta característica.

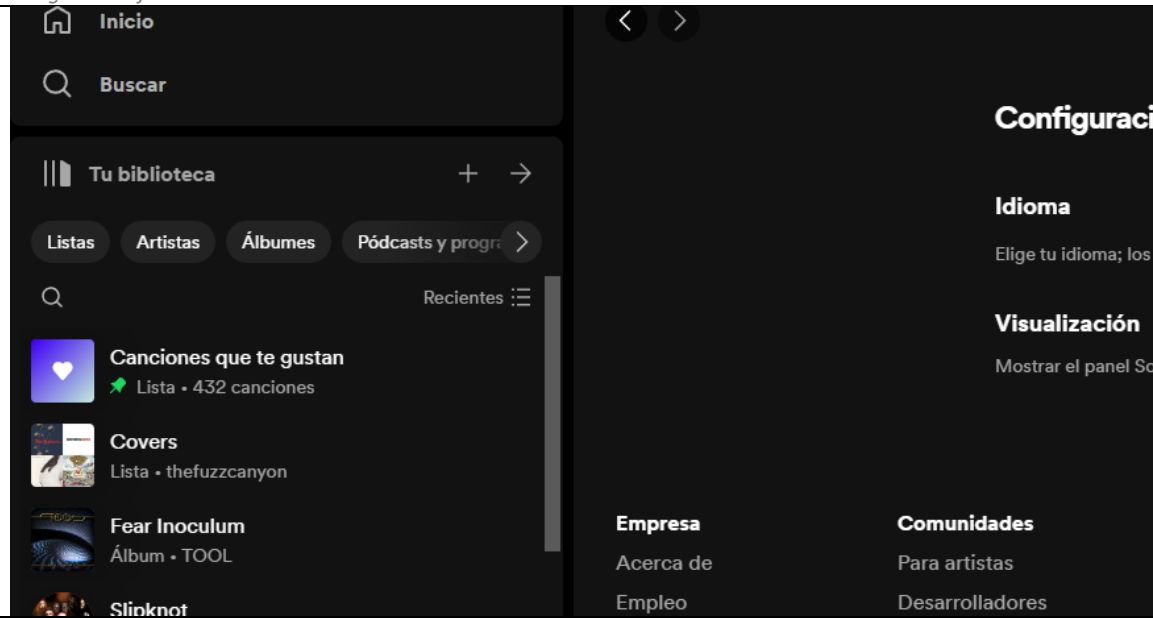
En la parte de perceptible (imagen 20) también encontramos que salta el error de tener dos o más encabezados seguidos sin contenido. En el html se puede comprobar:

<p>Imagen de referencia 29</p>  <pre> 44px) - 90px) - 519px), &gt;&gt;main tabindex= -1 aria- VUhVlpUda7B contents.pacing" style=""&gt; &lt;h2&gt; "width: 25ch;"&gt;&lt;/h2&gt;&lt;/div&gt;&lt;section data-testid= "home- IOU1unbFAd7ZHyeSMTH"&gt;&lt;div 9FFTYRCcB"&gt;&lt;div class="zXwER4Lsqq_e7fVVaPkZ GRncgnv1zFgF4dqE jFPYamKaGvZv7JA45KCcu" </pre>
Elaboración propia

Además, presenta problemas con la nomenclatura usual de las listas en html:

<p>Imagen de referencia 30</p>  <pre> an class= "ButtonInner-sc-14ud5tc-0 jsVPxO encore-inverted-light-set " role="list" tabindex="0" aria-label="web-player.your-library-x.your-library"&gt;&lt;div "width: 25ch;"&gt;&lt;/div&gt;&lt;div class="zXwER4Lsqq_e7fVVaPkZ GRncgnv1zFgF4dqE jFPYamKaGvZv7JA45KCcu" </pre> <p>Las listas se deben identificar mediante el elemento <code>&lt;ol&gt;</code> o <code>&lt;ul&gt;</code> y cada uno de sus items mediante elementos <code>&lt;li&gt;</code>, que siempre deben aparecer. No deben emplearse elementos de lista que carezcan de items. Si en un momento dado una lista carece de items entonces en lugar de mostrarse como una lista sin elementos hijos (sin elementos <code>&lt;li&gt;</code>) deberá eliminarse de la página. La estructura de las listas ha de ser correcta para asegurar que éstas se interpretan adecuadamente por los productos de apoyo, como los lectores de pantalla.</p>
Elaboración propia

Por último, aunque no aparezca como fallo sino como “sin revisar”, me parece que la visibilidad de la página podría mejorar si se le permitiera al usuario un control más extenso de la web. A lo que me refiero principalmente es al contraste de los colores ya que como se puede ver en la siguiente imagen, la paleta del menú es gris y gris mas oscuro. Pensaba que era por una extensión de dark themes pero no, es el tema nativo de la web.

<p>Imagen de referencia 31</p> 
Elaboración propia

Si se le permitiera al usuario cambiar los colores de fondo estoy seguro de que muchos usuarios agradecerían el poder distinguir mejor los elementos y opciones.

## 4. Análisis de accesibilidad Yale school of art

En páginas web de uso cotidiano la web de Spotify es común que cueste encontrar errores destacables dada la cantidad de tráfico y fondos que se generan con dicha web. Ahora quiero poner un ejemplo de lo contrario, una web terrorífica que nos puede resaltar problemas claros que enmendar.

Imagen de referencia 31

Perceptible						
La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que puedan percibirlos.						
	Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
1.1-Textos alternativos				0	2	0
1.1.1 - Contenido no textual <a href="#">?</a>		A	!		2	
1.2-Medios basados en el tiempo				0	0	0
1.2.1 - Solo audio y solo video (grabaciones) <a href="#">?</a>		A	na			
1.2.2 - Subtítulos (pregrabados) <a href="#">?</a>		A	na			
1.2.3 - Audiodescripción o Medio Alternativo (Pregrabado) <a href="#">?</a>		A	na			
1.2.4 - Subtítulos (en directo) <a href="#">?</a>		AA	na			
1.2.5 - Descripción auditiva (Pregrabada) <a href="#">?</a>		AA	na			
1.3-Adaptable				11	1	1
1.3.1 - Información y relaciones <a href="#">?</a>		A	✗	11	1	
1.3.2 - Secuencia con significado <a href="#">?</a>		A	na			
1.3.3 - Características sensoriales <a href="#">?</a>		A	?			1
1.4-Distinguible				0	4	3
1.4.1 - Uso del color <a href="#">?</a>		A	?			1
1.4.2 - Control del audio <a href="#">?</a>		A	na			
1.4.3 - Contraste (Mínimo) <a href="#">?</a>		A	?			1
1.4.4 - Redimensionamiento del texto <a href="#">?</a>		AA	!		4	
1.4.5 - Imágenes de texto <a href="#">?</a>		AA	?			1

Elaboración propia

Imagen de referencia 32

Operable						
Los componentes de la interfaz de usuario y la navegación deben ser operables.						
	Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
2.1-Accessible mediante el teclado				0	0	1
2.1.1 - Teclado <a href="#">?</a>		A	?			1
2.1.2 - Sin bloqueos de teclado <a href="#">?</a>		A	?			1
2.2-Tiempo suficiente				0	0	1
2.2.1 - Tiempo ajustable <a href="#">?</a>		A	?			1
2.2.2 - Pausar, detener, ocultar <a href="#">?</a>		A	?			1
2.3-Provocar ataques				0	0	1
2.3.1 - Umbral de tres destellos o menos <a href="#">?</a>		A	?			1
2.4-Navegable				0	13	5
2.4.1 - Evitar bloques <a href="#">?</a>		A	!		2	2
2.4.2 - Páginas tituladas <a href="#">?</a>		A	!		1	
2.4.3 - Orden del foco <a href="#">?</a>		A	?			1
2.4.4 - Propósito de los enlaces (en contexto) <a href="#">?</a>		A	✓			
2.4.5 - Múltiples vías <a href="#">?</a>		AA	?			1
2.4.6 - Encabezados y etiquetas <a href="#">?</a>		AA	!		10	
2.4.7 - Foco visible <a href="#">?</a>		AA	?			1

Elaboración propia

Imagen de referencia 33

**Comprensible**

La información y el manejo de la interfaz de usuario debe ser comprensible.

Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
3.1- Legible			0	0	3
3.1.1 - Idioma de la página	A	✓			
3.1.2 - Idioma de las partes	AA	?			1
3.2- Predecible			0	0	1
3.2.1 - Al recibir el foco	A	?			1
3.2.2 - Al introducir datos	A	?			1
3.2.3 - Navegación consistente	AA	?			1
3.2.4 - Identificación consistente	AA	?			1
3.3- Introducción de datos asistida			0	0	0
3.3.1 - Identificación de errores	A	na			
3.3.2 - Etiquetas o instrucciones	A	na			
3.3.3 - Sugerencias ante errores	AA	na			
3.3.4 - Prevención de errores (legales, financieros, datos)	AA	na			

Elaboración propia

Imagen de referencia 34

**Robusto**

El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de agentes de usuario, incluyendo las ayudas técnicas.

Pauta	Nivel	Resultado	Problemas	Advertencias	No verificados
4.1- Compatible			0	0	1
4.1.1 - Procesamiento	A	na			
4.1.2 - Nombre, función, valor	A	?			1

Elaboración propia

Imagen de referencia 35

**11 Problemas**  
en 1 criterios de éxito  
Señ necesarias correcciones

- Perceptible 11
- Operable 0
- Comprendible 0
- Robusto 0

**20 Advertencias**  
en 6 criterios de éxito  
Es necesario revisar manualmente

- Perceptible 7
- Operable 13
- Comprendible 0
- Robusto 0

**18 No verificados**  
en 18 criterios de éxito  
Comprobación completamente manual


- Perceptible 4
- Operable 8
- Comprendible 5
- Robusto 1

Elaboración propia

Antes de nada, decir que la página web es funcional, pero es un gran ejemplo de la necesidad de adaptación de antiguas paginas web para ser accesibles a todo tipo de personas.

Un problema que ocurre a menudo en esta web es que no se implementa correctamente el css en html, encontrándonos a veces con estilos in-line. Mi primera solución para evitar estos fallos es simplemente diseñar la página web como una persona normal y hacer uso de los CSS.

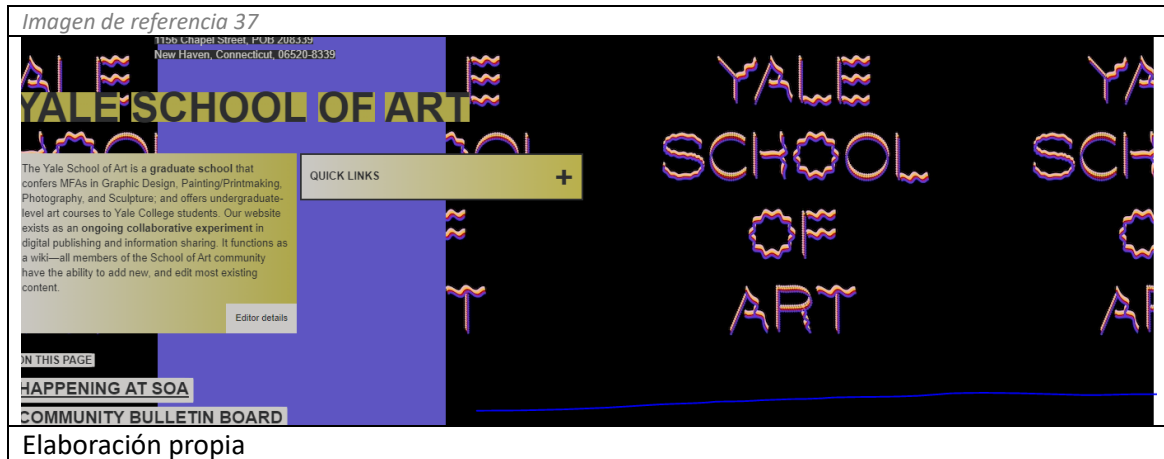
Imagen de referencia 36

65.  <font size="2">The Yale School of Art is a <strong>graduate school</strong> that confers M  
ale College students. Our website exists as an <strong>ongoing collaborative experiment</strong> i  
bility to add new, and edit most existing content.</font>  
66. </div>

Elaboración propia

Uno de los errores más importantes y que se puede ver a simple vista es la elección de fondo en mi opinión. Como se puede ver en la imagen 36 el fondo es completamente oscuro con líneas brillantes siendo muy desorientador.





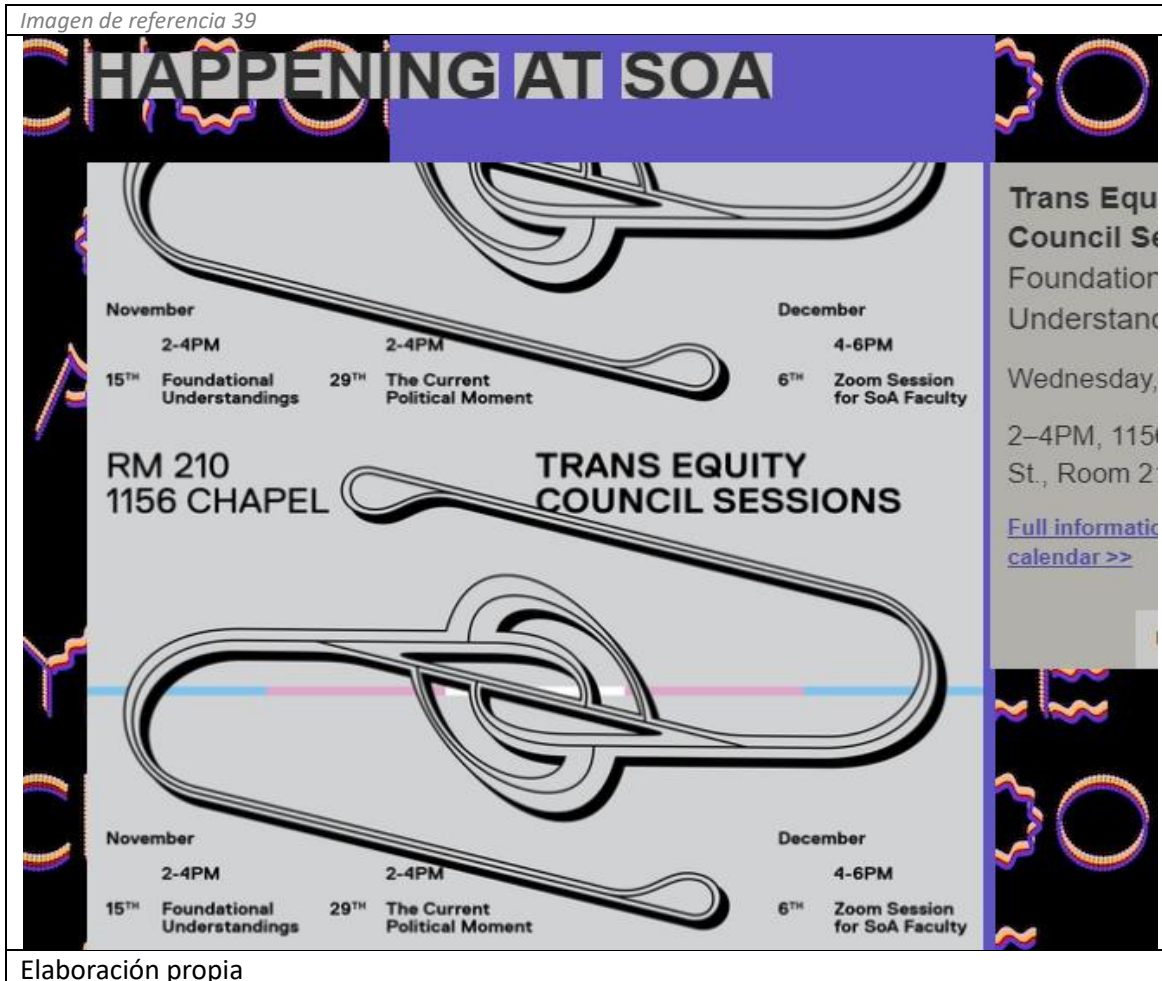
Además, siendo una web oficial de la universidad de Yale, no se entiende como puede ser que toda la funcionalidad de la página quede relegada a la parte izquierda, siendo una sola columna en la que aparecen las publicaciones.

Al comprobar el site en el móvil vemos que efectivamente, está diseñada para móvil principalmente. Así que mi segunda mejora de página sería hacerla responsive, que estamos hablando de una universidad privada en Estados Unidos, fondos hay.



No se encuentra ningún título extra para las imágenes en caso de que no carguen (imagen 39) lo cual salta las alarmas de tawdis.

Imagen de referencia 39



Elaboración propia

Dejando de lado la ironía del gusto artístico de una página de una universidad de artes, en la parte comprensible de tawdis no se consigue revisar la navegación consistente, probablemente porque no hay ninguna cohesión para empezar (Imagen40). No parece haber una estructura concreta y todos los apartados se vomitan sobre el usuario que entre el fondo negro (no está en modo dark theme) y la cantidad de diferencia de formatos en los cards está destinado a la confusión.

Imagen de referencia 40

functions as a wiki, all members of the School of Art community—graduate students, faculty, staff, and alumni—have the ability to add new content and pages, and to edit most of the site's existing content.

Content is the property of its various authors. When you contribute to this site, you agree to abide by Yale University academic and network use policy, and to act as a responsible member of our community.

Visitor: Log in

Edit this page

Page history

Page last changed by: Lindsey Mancini

Background changed by: Lindsey Mancini

ON THIS PAGE

**HAPPENING AT SOA**

**COMMUNITY BULLETIN BOARD**

**CALENDARS & NEWSLETTERS**

## Register for a 2023 Virtual Open House

(click on an event to link out to pre-registration page)

**November 15, 2023 - Wednesday**

10:00 AM Graphic Design Open House  
4:00 PM Sculpture Open House

**November 16, 2023 - Thursday**

10:00 AM Photography Open House  
4:00 PM Painting & Printmaking Open House

Editor details

**SCHOOL OF ART**

**HAPPENING AT SOA**

Elaboración propia

Para acabar de sacar errores en esta página, que podríamos estar un rato, terminaré con un error que nos redirige al principio, al título. Este error del apartado operable resalta que el título no es suficientemente descriptivo, en este caso no hay problema con el título en cuanto a palabras, sino que no se ha colocado siquiera una imagen para él.

Imagen de referencia 41

Home - Yale School of Art

Elaboración propia

## 5.Conclusión

Cabría pensar que a día de hoy no habría ni que pensar en términos de accesibilidad ni usabilidad, pero lamentablemente no es así, y creo que hay que seguir concienciando a los diseñadores, tanto de aplicaciones de escritorio como los diseñadores de páginas web, para intentar alcanzar a cuanto más público mejor.

De las aplicaciones de escritorio que he analizado cabe recalcar que una está muy dirigida a un público (gitkraken) en particular, pero eso no la salva de críticas y la otra (Steam), la mayoría de los problemas son debidos a la naturaleza del mercado en sí. Las páginas web una tiene una mejoría sencilla de implementar ya que todo funciona bastante bien para el usuario común(spotify) y la de Yale, necesitaría de un completo overhaul.

## 6.Bibliografía

Todas las capturas son de elaboración propia y los materiales han sido proporcionados por el profesor.



**Sergio Ramos Torres**  
Desarrollo de Aplicaciones Multiplataforma  
Desarrollo de Interfaces

2023-2024