|  |  |
| --- | --- |
| 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](#_Toc149330735)

[3](#_Toc149330736)

[1.1 Visibilidad del estado de sistema 4](#_Toc149330737)

[1.2 Consistencia entre sistema y mundo real: orden lógico, lenguaje familiar 6](#_Toc149330738)

[1.3 Control del usuario: abandonar en cualquier momento, deshacer o repetir una acción 6](#_Toc149330739)

[1.4 Consistencia y estándares: lenguaje coherente 7](#_Toc149330740)

[1.5 Prevención de errores 7](#_Toc149330741)

[1.6 Es mejor reconocer que recordar: objetos, acciones y opciones a la vista 7](#_Toc149330742)

# 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.

## Visibilidad del estado de sistema

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

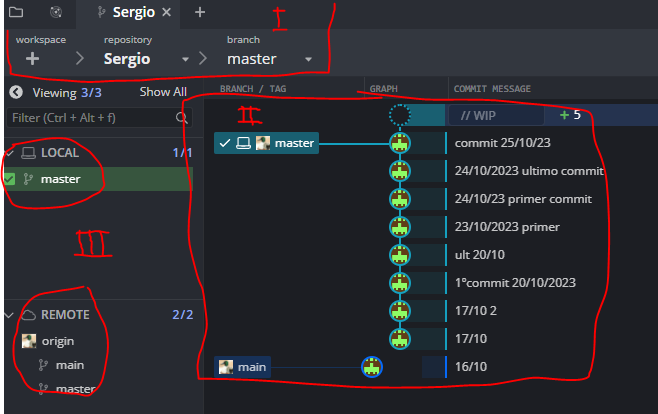


Imagen de referencia

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.

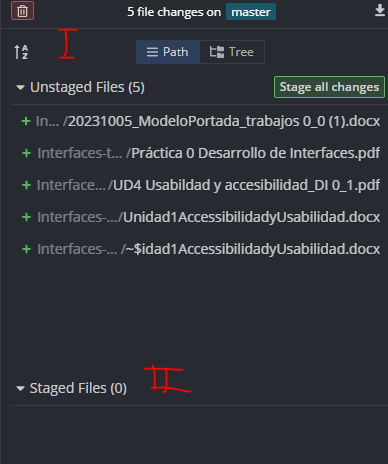


Imagen de referencia

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

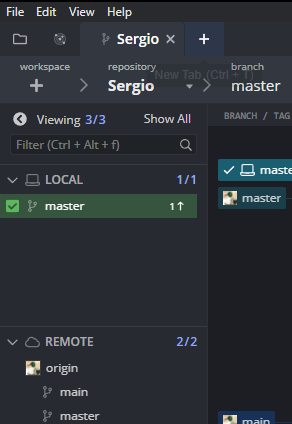


Imagen de referencia

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

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.

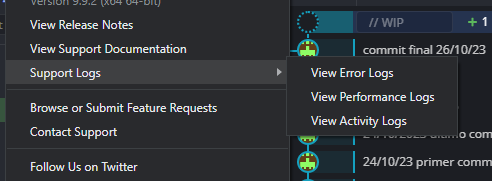


Imagen de referencia

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

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

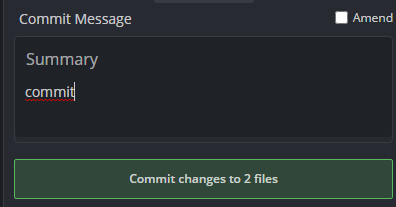


Imagen de referencia

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

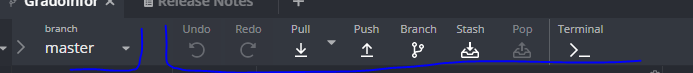


Imagen de referencia

## 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:

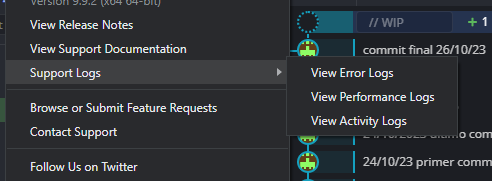


Imagen de referencia

## 1.10

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.

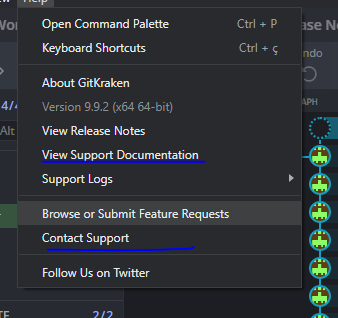


Imagen de referencia

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