

1. *A que se le domina control de versiones*

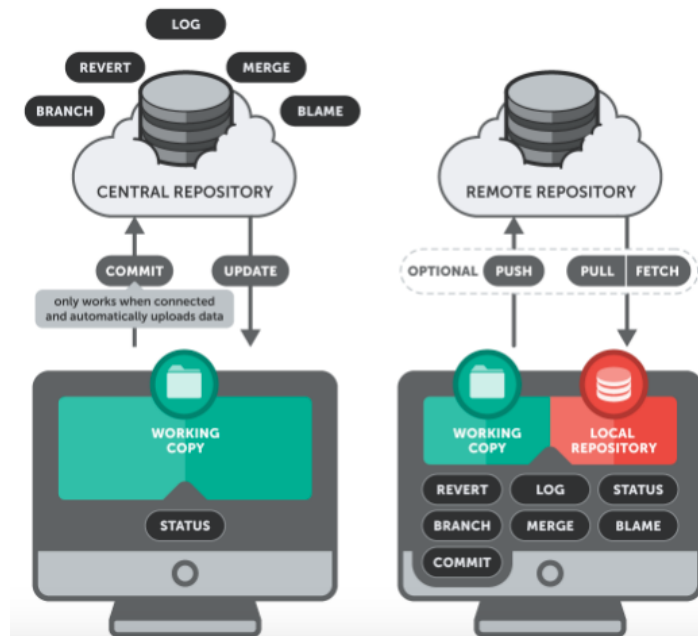
El control de versiones es el sistema que permite gestionar y realizar un seguimiento de los cambios en archivos y proyectos a lo largo del tiempo. Facilita la colaboración, evita la pérdida de datos y permite revertir a versiones anteriores si es necesario.

2. *¿Cuáles son los tipos de sistemas de control de versiones y que los diferencias?*

**Control de versiones local:** Los cambios se registran en el mismo dispositivo de un usuario, limitando la colaboración y dificultando el acceso remoto.

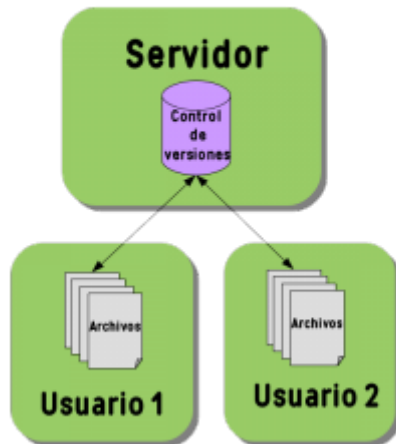
**SUBVERSION**

**GIT**



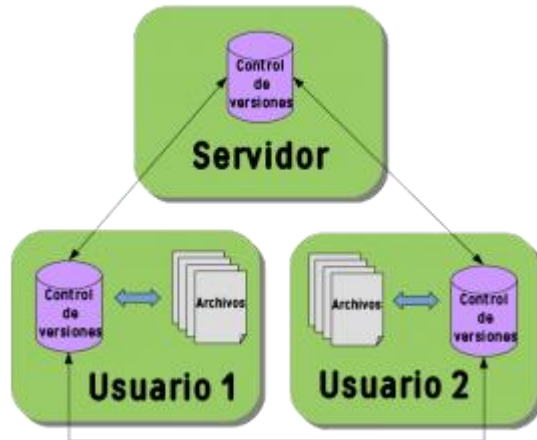
**Control de versiones centralizado (CVCS):** Tiene un servidor central que almacena todas las versiones, permitiendo el acceso a múltiples usuarios, pero depende del servidor para funcionar.

**Modelo centralizado**



**Control de versiones distribuido (DVCS):** Cada usuario tiene una copia completa del proyecto y su historial, lo que permite trabajar sin conexión y facilita la colaboración. Ejemplo: Git.

### Modelo distribuido



### 3. ¿Qué es Git?

Git es un sistema de control de versiones distribuido que permite a múltiples desarrolladores trabajar en un proyecto de manera simultánea. Es ampliamente utilizado en el desarrollo de software por su eficiencia y capacidad de trabajar en proyectos de cualquier tamaño.



**Herramienta de control de versiones distribuida**

**Herramienta de código abierto que los desarrolladores instalan localmente para gestionar el código fuente**



**Plataforma basada en la nube**

**Servicio en línea al que los desarrolladores que utilizan Git pueden conectarse y cargar o descargar recursos**

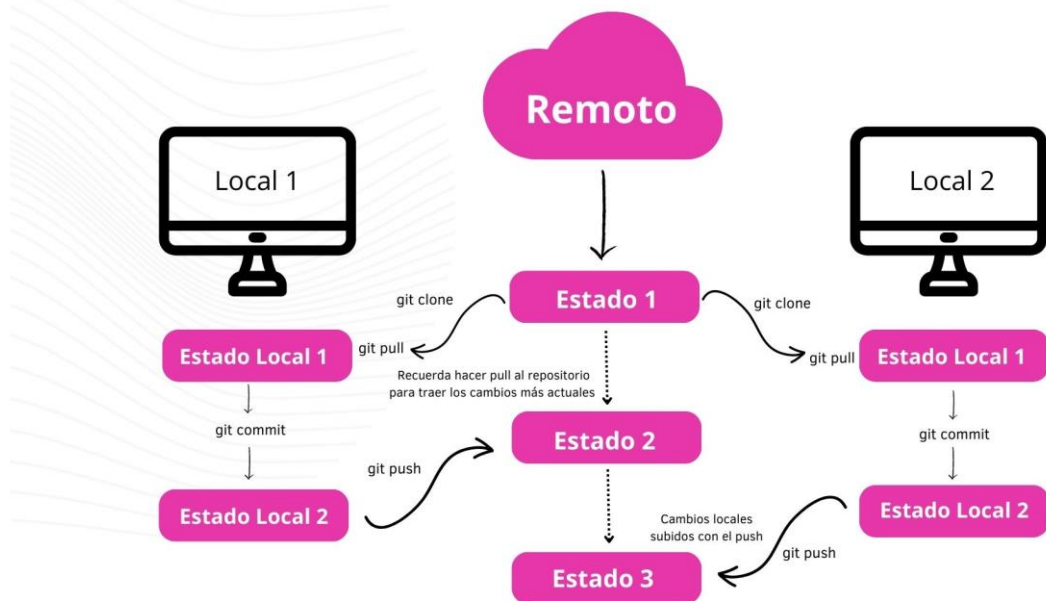
4. ¿Cuáles son las principales características y ventajas de Git?

**Distribuido:** Cada usuario tiene una copia completa del historial.

**Eficiencia:** Diseñado para gestionar grandes proyectos con rapidez.

**Colaboración:** Facilita el trabajo en equipo y la integración de cambios.

**Integridad:** Protege el historial de cambios contra alteraciones accidentales o maliciosas.



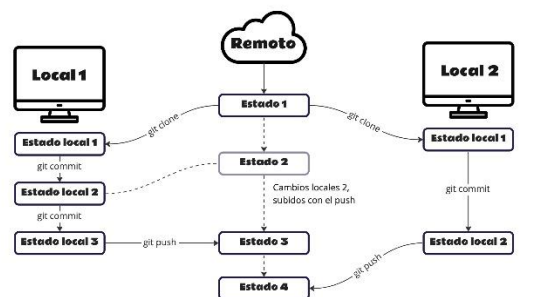
5. ¿Cuál es el funcionamiento del control de cambios del Software?

En Git, los cambios se registran en "commits", que crean puntos de restauración del proyecto. Estos cambios pueden fusionarse, revertirse o compartirse entre múltiples desarrolladores

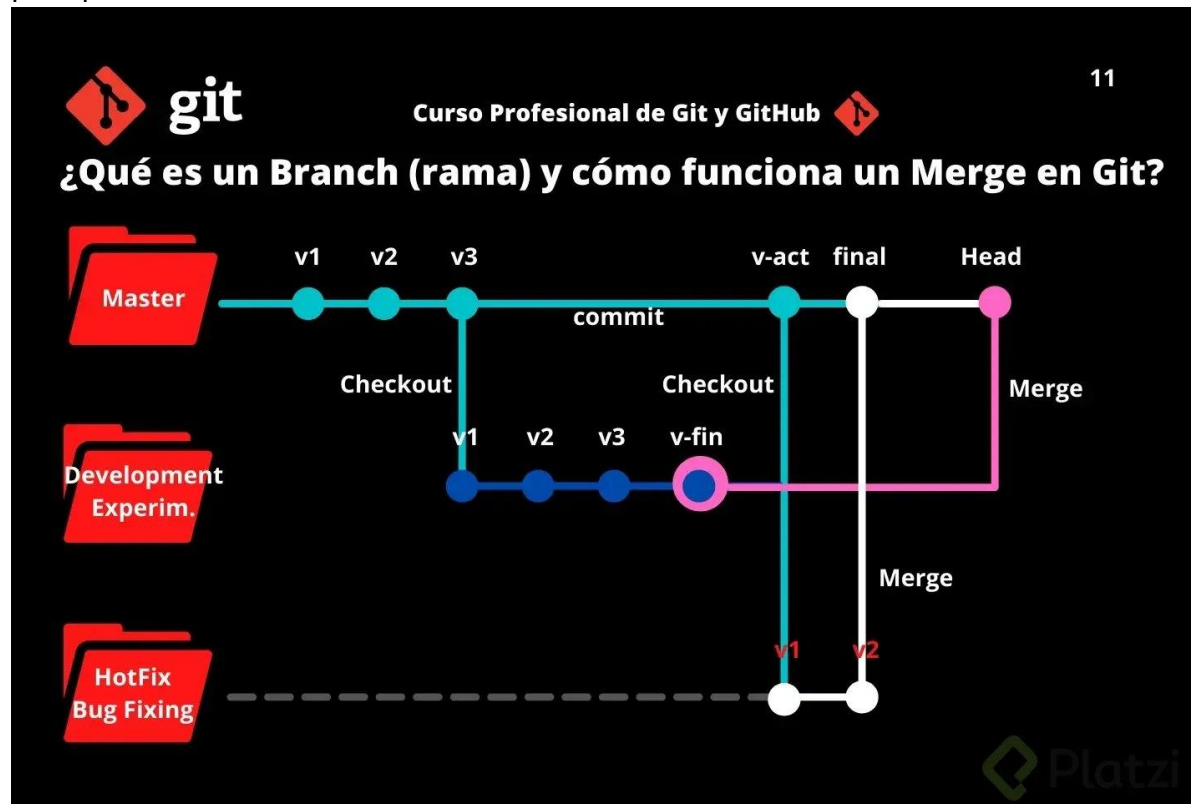
6. Defina los siguientes conceptos:

- ¿Qué es un repositorio?

Lugar donde se almacenan los archivos y el historial de cambios de un proyecto.



- ¿Qué es una rama o Branch?  
Una línea de desarrollo separada, que permite experimentar sin afectar la rama principal.



- ¿Qué es commit?

Una instantánea de los cambios en el proyecto en un momento específico.

## 3 FORMAS DE HACER COMMIT DE TU CÓDIGO

**1** `git commit`

Abre tu editor de código de texto para **que** **escribas el mensaje más detallado.**

**2** `git commit -m "mensaje"`

Te permite **escribir el mensaje en el mismo comando** entre comillas.

**3** `git commit -am "mensaje"`

**Agrega todos los archivos modificados y te deja escribir el mensaje en un sólo paso.**

Aprende a controlar las versiones de tus proyectos en:  
[ed.team/cursos/git](https://ed.team/cursos/git)

- ¿Qué es un readme?

Archivo de texto que proporciona una descripción del proyecto, su propósito, y cómo utilizarlo o contribuir a él

Overview Repositories Projects Packages

octocato / README.md

Hi there 🐱

- 🔗 I'm currently working on something cool!
- 📖 I'm currently learning with help from docs.github.com
- 🗨️ Ask me about GitHub

Pinned

atom  
Forked from atom/atom  
The hackable text editor  
JavaScript

vscode  
Forked from microsoft/vscode  
Visual Studio Code  
TypeScript



7. Realice un listado con todos los comandos de Git y describa cual es su función

git init: Crea un repositorio nuevo.

git clone: Copia un repositorio existente.

git add: Prepara archivos para un commit.

git commit: Guarda los cambios en el historial.

git push: Sube los cambios a un repositorio remoto.

git pull: Actualiza el repositorio local con cambios de un repositorio remoto.

git status: Muestra el estado de los archivos en el repositorio.

git branch: Gestiona ramas en el proyecto.



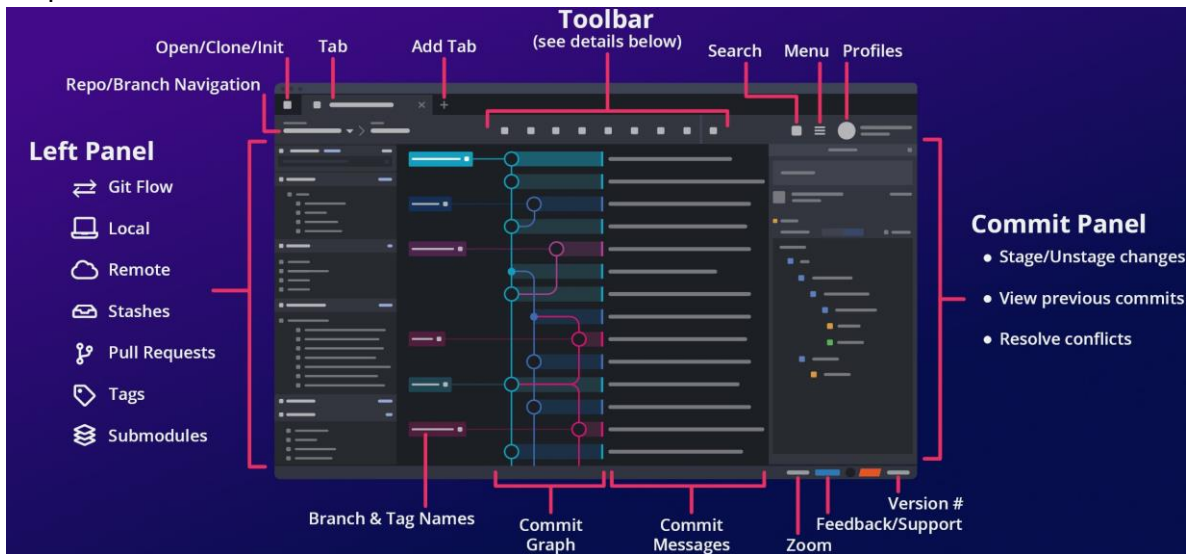
8. Realice un listado con los diferentes softwares que proporcionan un interfaz grafica (GUI) para el control de versionamiento de código con Git, por ejemplo, GitHub Desktop, GitKraken, Source Tree, adjunte una captura de pantalla de cada uno y

*Explique cuales son los requerimientos de hardware y sus principales características*

**GitHub Desktop:** Interfaz sencilla para trabajar con repositorios de GitHub.



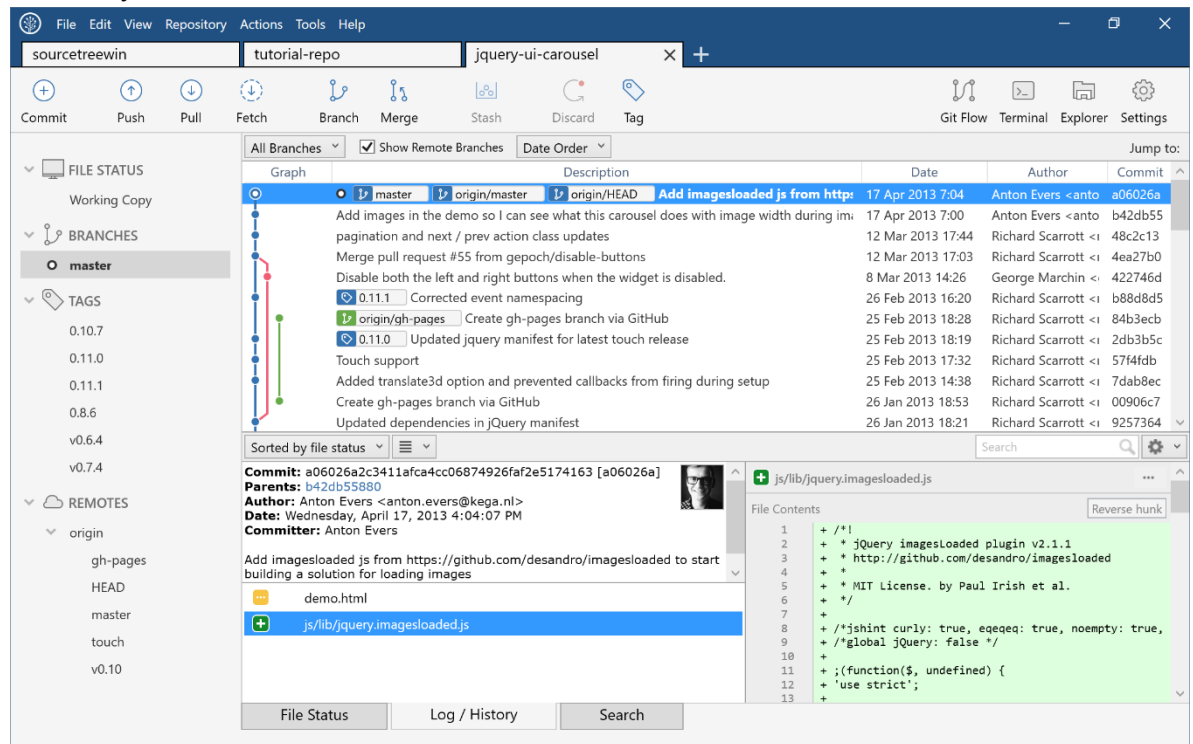
**GitKraken:** Visualizador y administrador de Git con herramientas colaborativas y de productividad.



**SourceTree:** Administrador de repositorios Git compatible con GitHub y Bitbucket.

**Tower:** Software para manejar repositorios Git con características avanzadas de

revisión y edición.



9. ¿En que consiste la sintaxis Markdown y cuáles son sus aplicaciones?

Markdown es un lenguaje de marcado que facilita la creación de texto formateado (negritas, listas, enlaces, etc.) de forma sencilla y legible. Se usa en archivos README, blogs, documentación y comentarios en GitHub.





Elaborar un mapa mental que reúna y sintetice los siguientes conceptos: 1. ¿Qué es Git?, comandos de Git y su función

Rta //:

<https://mm.tt/app/map/3499712215?t=H48HYx2DEA>



### 3.3 – Actividades de apropiación del conocimiento (Conceptualización y teorización)

1. A través de capturas de pantalla, explique cuál es el proceso para crear una cuenta en Github.

#### 1. Ir a la página de GitHub

- Abre tu navegador y ve a <https://github.com>.
- En la página principal, verás un botón que dice **Sign up** o **Registrarse**. Haz clic en él.

#### 2. Ingresar tus datos

- En la página de registro, se te pedirá que ingreses algunos datos:
  - **Correo electrónico**: Escribe tu dirección de correo electrónico.
  - **Crear una contraseña**: Elige una contraseña segura.
  - **Nombre de usuario**: Elige un nombre de usuario único. Este nombre será tu identificador en GitHub.
  - **Preferencias**: GitHub puede preguntar si deseas recibir notificaciones o correos electrónicos sobre productos y actualizaciones.
- Luego de llenar estos datos, haz clic en **Continue** o **Continuar**.

#### 3. Verificación de seguridad

- Completa el desafío de seguridad para demostrar que no eres un robot. Esto puede ser un "captcha" o un pequeño acertijo.

#### 4. Elegir tu plan

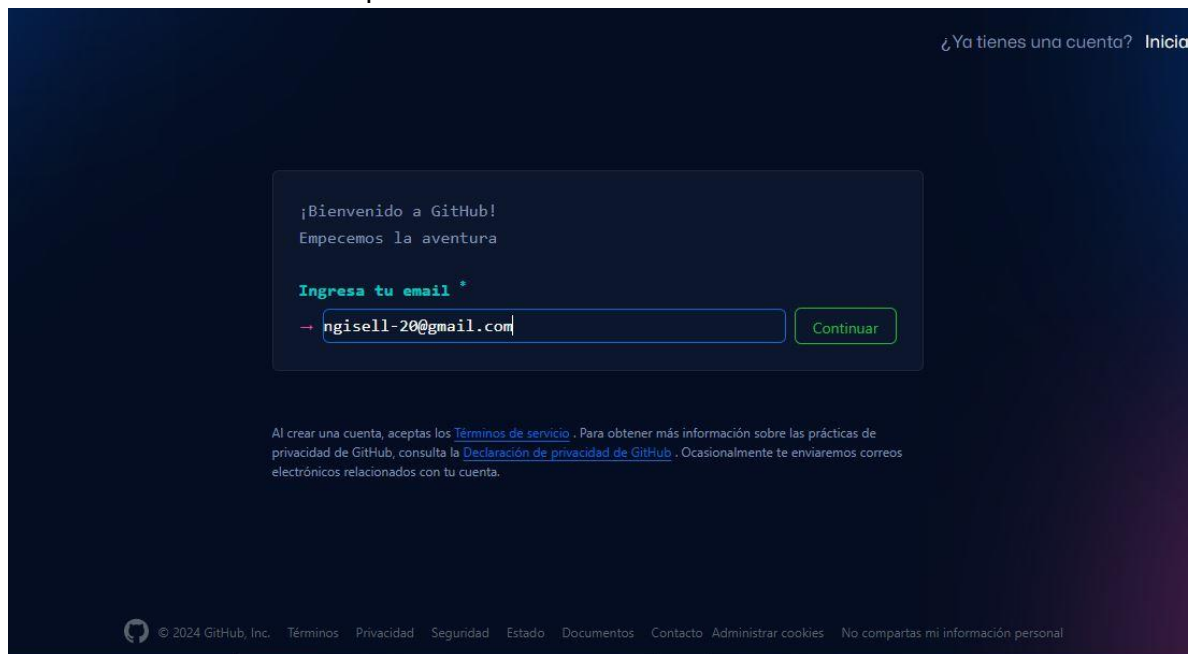
- GitHub ofrece diferentes planes. Para la mayoría de los usuarios, el plan **Free** es suficiente, ya que permite crear repositorios ilimitados, tanto públicos como privados.
- Selecciona el plan deseado y haz clic en **Continue**.

#### 5. Verificación de correo electrónico

- Después de completar el registro, GitHub enviará un correo electrónico de verificación a la dirección que ingresaste.
- Abre tu bandeja de entrada, busca el correo de GitHub y haz clic en el enlace de verificación para activar tu cuenta.

#### 6. Completar el perfil

- Una vez que confirmes tu correo, GitHub te llevará a la página de inicio de tu cuenta. Aquí podrás personalizar tu perfil, como añadir una foto, bio, y enlaces a redes sociales o sitios web personales si deseas.




¿Ya tienes una cuenta? [Inicia sesión](#)

¡Bienvenido a GitHub!  
Empecemos la aventura

Ingresar tu email \*

ngisell-20@gmail.com [Continuar](#)

Al crear una cuenta, aceptas los [Términos de servicio](#). Para obtener más información sobre las prácticas de privacidad de GitHub, consulta la [Declaración de privacidad de GitHub](#). Ocasionalmente te enviaremos correos electrónicos relacionados con tu cuenta.

 © 2024 GitHub, Inc. [Términos](#) [Privacidad](#) [Seguridad](#) [Estado](#) [Documentos](#) [Contacto](#) [Administrar cookies](#) [No compartas mi información personal](#)

¿Ya tienes una cuenta? [Iniciar sesión](#) →

¡Bienvenido a GitHub!  
Empecemos la aventura

**Ingresar tu email \***

✓ ngisell-20@gmail.com

**Crear una contraseña \***

✓ .....

**Introduzca un nombre de usuario \***

→ Gisell-hue

Continuar

Gisell-hue está disponible.

¡Bienvenido a GitHub!  
Empecemos la aventura

**Verificar su cuenta**

### Desafío de audio

¿Cuál opción representa el sonido que hacen los gatos?

Escribe un número como respuesta y luego presiona "Enter"  
o el botón de "Listo" que aparece a continuación.

▶ Reproducir

3

Listo

Visual

Reiniciar



# Bienvenido a GitHub

Nos alegra que estés aquí.

Esto nos ayudará a guiarte hacia las herramientas que mejor se adaptan a tus proyectos.

¿Cómo te describirías a ti mismo?

☒ Alumno

☐ Maestro

☐ Desarrollador en activo

☐ Otro

¿Cuántos miembros del equipo trabajarán contigo?

☒ Sólo yo

☐ 2-5

☐ 5-10

☐ 10-20

☐ 20-50

☐ 50+

Continuar



# Cuéntanos un poco sobre ti.

Le ayudaremos a empezar con el pie derecho.

¿Cuáles son las dos cosas más importantes que quieres hacer con GitHub?

Seleccione hasta 2 opciones

☒ Iniciar un nuevo proyecto

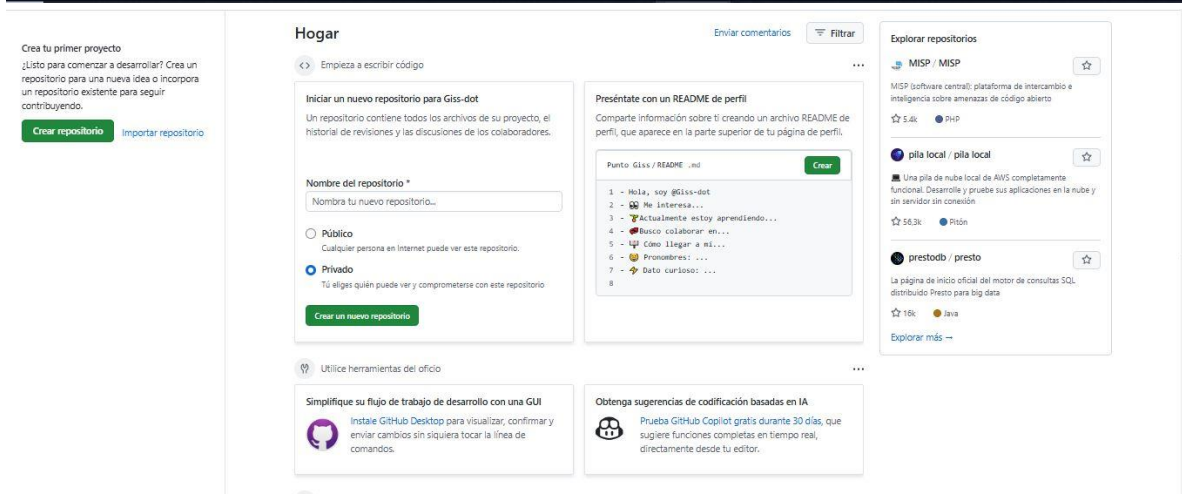
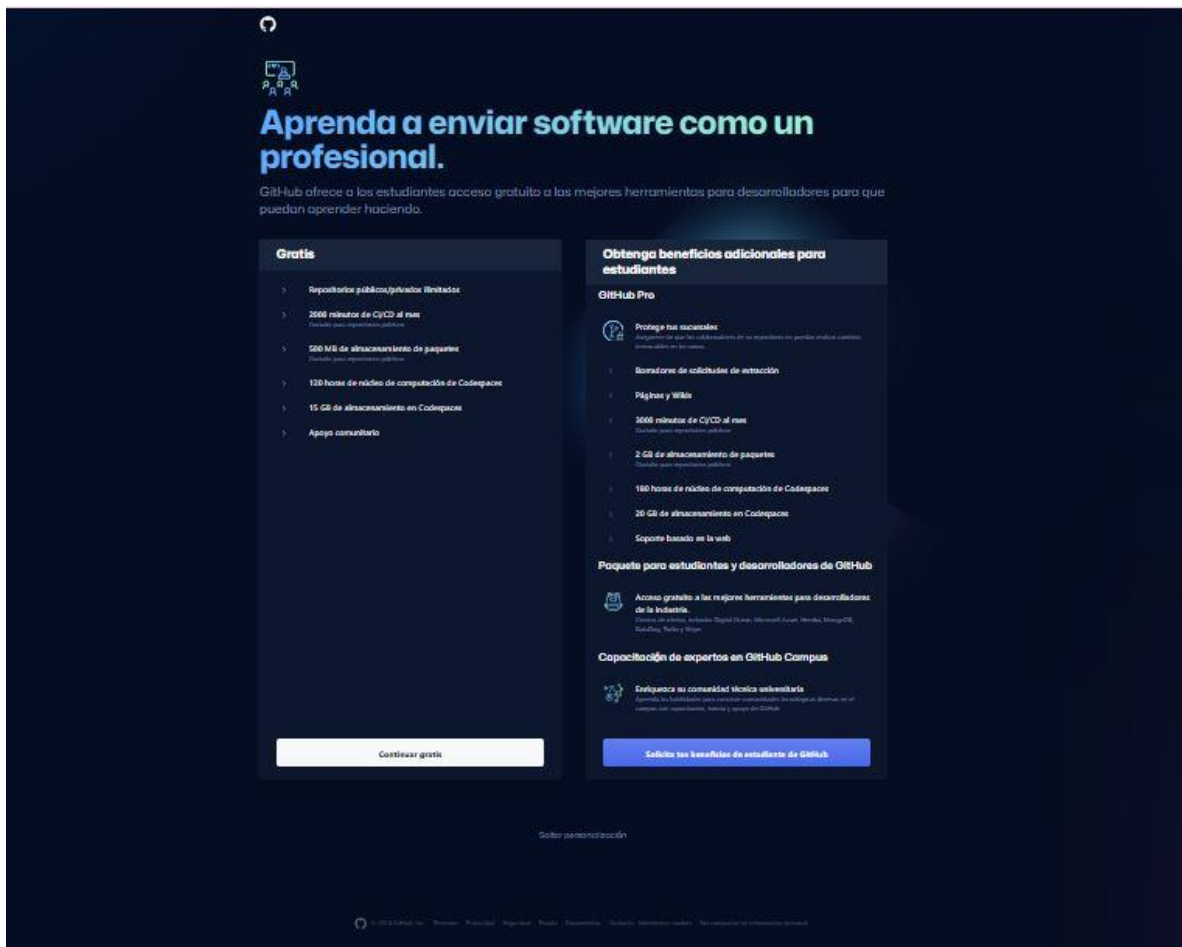
☐ Iniciar o expandir mi negocio

☐ Utilice GitHub Copilot

☐ Conéctese con otros desarrolladores

☐ Contribuir a proyectos de código abierto

Continuar



2. A través de capturas de pantalla, explique cuál es el proceso para crear un repositorio en Github.

Panel

Encuentra un repositorio...

Gisel134 / Inventario-licorera

Hogar

Empieza a escribir código

Nombre del repositorio \*

Nombra tu nuevo repositorio...

Público

Cualquier persona en Internet puede ver este repositorio.

Privado

Tú eliges quién puede ver y comprometerse con este repositorio

Crear un nuevo repositorio

Preséntate con un README de perfil

Comparte información sobre ti creando un archivo README de perfil, que aparece en la parte superior de tu página de perfil.

Gisel134 / README .md

Crear

1 - Hola, soy @Gisel134

2 - Me interesa...

3 - Actualmente estoy aprendiendo...

4 - Busco colaborar en...

5 - Cómo llegar a mí...

6 - Pronombres: ...

7 - Dato curioso: ...

8

Utilice herramientas del oficio

Enviar comentarios

Filtrar

Explorar repositorios

Vuelo iNav / inav

INAV: Software de control de vuelo habilitado para navegación

3.2k

do

tjenkinson / marquesina dinámica

Una pequeña biblioteca para crear marquesinas.

121

JavaScript

Azure / azure-sdk-for-js

Este repositorio es para el desarrollo activo del SDK de Azure para JavaScript (NodeJS y navegador). Para los consumidores del SDK, recomendamos visitar nuestra documentación pública para desarrolladores en <https://docs.microsoft.com/...>

2.1k

Mecanografiado

Explorar más

Nuevo repositorio

Encuentra un repositorio...

Crear un nuevo repositorio

Un repositorio contiene todos los archivos del proyecto, incluido el historial de revisiones. ¿Ya tienes un repositorio de proyectos en otro lugar? [Importa un repositorio.](#)

Los campos obligatorios están marcados con un asterisco (\*).

Dueño \*

Gisel134

Nombre del repositorio \*

Proyecto-inventario

Proyecto-inventario está disponible.

Los buenos nombres de repositorios son breves y fáciles de recordar. ¿Necesitas inspiración? ¿Qué tal?

Guacamole amigable ?

Descripción (opcional)

Proyecto para creación de un programa para el manejo de inventario, ventas y registro de una tienda licorera.

Público

Cualquier persona en Internet puede ver este repositorio. Tú eliges quién puede contribuir.

Privado

Tú eliges quién puede ver y comprometerse con este repositorio.

Inicialice este repositorio con:

Guacamole amigable ?

Descripción (opcional)

Proyecto para creación de un programa para el manejo de inventario, ventas y registro de una tienda licorera.

Público

Cualquier persona en Internet puede ver este repositorio. Tú eliges quién puede contribuir.

Privado

Tú eliges quién puede ver y comprometerse con este repositorio.

Inicialice este repositorio con:

Agregar un archivo README

Aquí puedes escribir una descripción detallada de tu proyecto. [Obtén más información sobre los archivos README.](#)

Agregar .gitignore

Plantilla .gitignore : Ninguna

Seleccione los archivos que no desea rastrear de una lista de plantillas. [Obtenga más información sobre cómo ignorar archivos.](#)

Elija una licencia

Licencia : Ninguna

Una licencia indica a los demás lo que pueden y no pueden hacer con su código. [Obtenga más información sobre las licencias.](#)

Esto establecerá `main` como rama predeterminada. Cambie el nombre predeterminado en su [configuración](#).

Estás creando un repositorio público en tu cuenta personal.



Gisell34 / Proyecto-inventario

Escribe para buscar

<> Código

Asuntos

Solicitudes de extracción

Comportamiento

Proyectos

Wiki

Seguridad

Perspectivas

Ajustes

Proyecto-inventario Público

Alfiler

Dejar de ver 1

Tenedor 0

Estrella 0

principal 1 sucursal Etiquetas

Ir al archivo

Agregar archivo

Código

Gisell34 Compromiso inicial 4de71ef · now 1 Compromiso

.gitignore Compromiso inicial now

README.md Compromiso inicial now

README

Proyecto-inventario

Proyecto para creación de un programa para el manejo de inventario, ventas y registro de una tienda licorera.

Acerca de

Proyecto para creación de un programa para el manejo de inventario, ventas y registro de una tienda licorera.

Léame

Actividad

0 estrellas

1 mirando

0 tenedores

Lanzamientos

No hay comunicados publicados

[Crear una nueva versión](#)

Gisell34 / Proyecto-inventario

Escribe para buscar

<> Código

Asuntos

Solicitudes de extracción

Comportamiento

Proyectos

Wiki

Seguridad

Perspectivas

Ajustes

Proyecto-inventario /

Arrastre archivos aquí para agregarlos a su repositorio

elige tus archivos

Confirmar cambios

Añadir archivos mediante carga

Añade una descripción ampliada opcional...

Abrir

Documentos Proyecto

Buscar en Proyecto

Organizar Nueva carpeta

OneDrive - Person

Este equipo

Descargas

Documentos

Escritorio

Imágenes

Música

Objetos 3D

Videos

Disco local (C:)

Gisell34 (D:)

Nombre

Fecha de modificación

Tipo

Metodos de recoleccion

5/11/2024 3:36 p. m.

Documento de

Proyecto (1)

5/11/2024 3:36 p. m.

Documento de

Nombre de archivo:

Todos los archivos

Abrir

Cancelar

Proyecto-inventario /

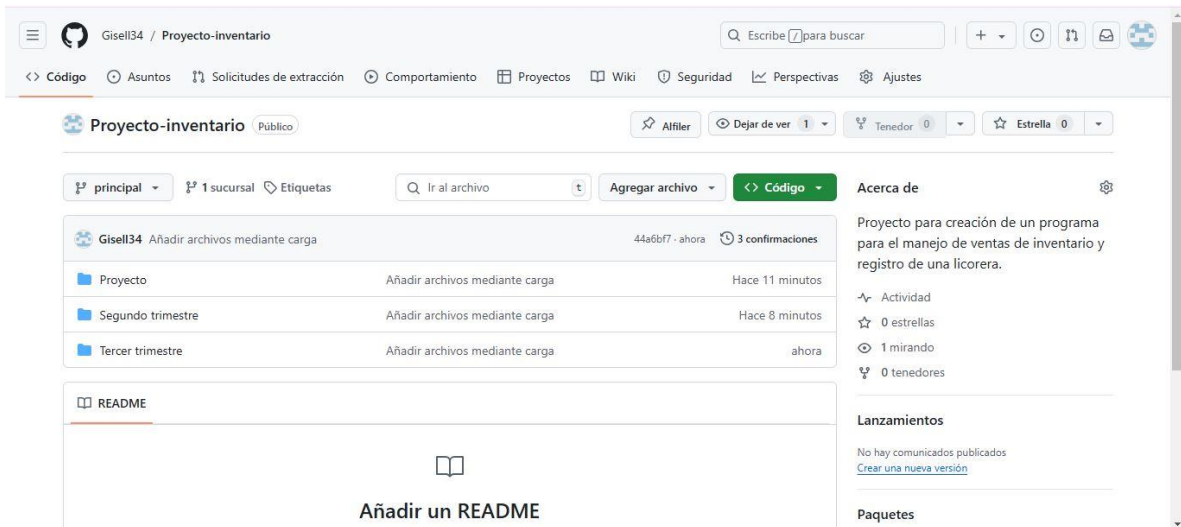
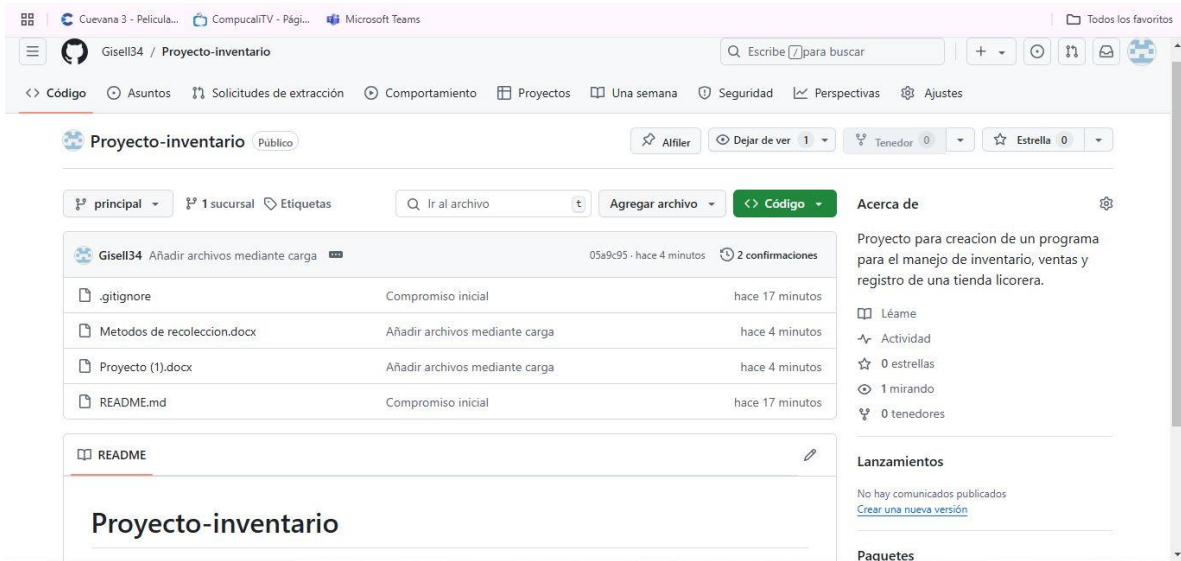
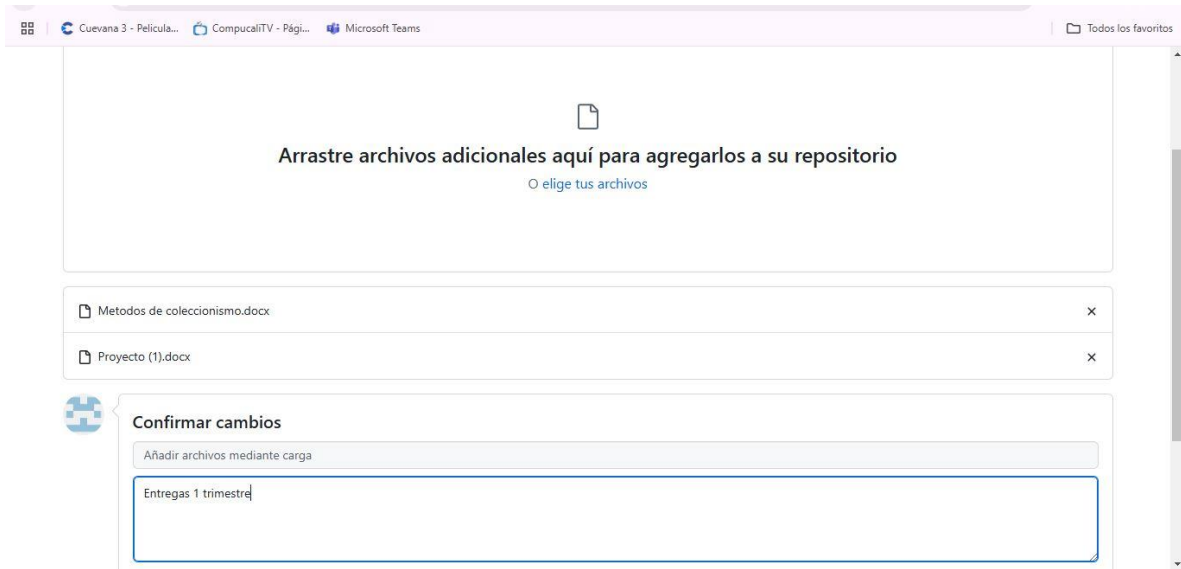
Arrastre archivos aquí para agregarlos a su repositorio

elige tus archivos

Confirmar cambios

Añadir archivos mediante carga

Añade una descripción ampliada opcional...



### 1. Iniciar sesión en tu cuenta de GitHub

- Abre tu navegador y ve a <https://github.com>.
- Inicia sesión con tu correo electrónico y contraseña.

### 2. Ir a la página de creación de repositorios

- Una vez que hayas iniciado sesión, en la esquina superior derecha de la página, haz clic en el icono + (al lado de tu foto de perfil).
- Selecciona **New repository** o **Nuevo repositorio** en el menú desplegable.

### 3. Configurar el repositorio

- En la página de creación, verás campos para configurar tu nuevo repositorio:
  - **Owner:** Selecciona tu nombre de usuario o una organización de la que seas miembro, si aplica.
  - **Repository name:** Ingresa el nombre de tu repositorio. Asegúrate de que sea único en tu cuenta.
  - **Description (optional):** Agrega una breve descripción del propósito del repositorio (opcional).
- **Elegir la visibilidad:**
  - **Public:** Cualquiera puede ver este repositorio. Ideal para proyectos de código abierto.
  - **Private:** Solo tú y los colaboradores que invites podrán ver este repositorio.

### 4. Opciones adicionales

- **Initialize this repository with a README:** Marca esta opción si deseas que GitHub genere un archivo README.md, donde podrás añadir una introducción al proyecto.
- **Add .gitignore:** Puedes seleccionar una plantilla de .gitignore específica para tu proyecto, útil para excluir archivos que no quieres incluir en el repositorio (por ejemplo, archivos temporales o binarios).
- **Choose a license:** GitHub te permite añadir una licencia. Si es un proyecto público, es recomendable agregar una licencia para definir cómo otros pueden usar tu código.

### 5. Crear el repositorio

- Una vez que hayas completado todos los campos y configurado las opciones, haz clic en **Create repository** o **Crear repositorio**.

## 6. Explorar tu nuevo repositorio

- GitHub te redirigirá a la página principal de tu nuevo repositorio. Aquí podrás ver el archivo README (si lo creaste), opciones para subir archivos o hacer tu primer commit, y un enlace para clonar el repositorio.

3. Consultar acerca de la sintaxis Markdown y cómo se aplicaría esta sintaxis en la creación de un readme en un repositorio GitHub.

Markdown es ampliamente usado en GitHub para dar formato a archivos README y documentar proyectos de manera clara y estructurada. Este lenguaje permite añadir estilos básicos y organizar la información en secciones, lo cual ayuda a otros usuarios a entender el propósito y el uso del proyecto.

### Principales elementos de la sintaxis Markdown y su aplicación en un README de GitHub

#### 1. Encabezados

- Los encabezados se crean usando #, donde un solo # es el título principal, ## el subtítulo, y así sucesivamente hasta ##### para subtítulos menores.

markdown

Copiar código

# Título del Proyecto

## Descripción

### Instalación

#### 2. Texto en Negrita y Cursiva

- Para **negrita** usa **\*\*texto\*\*** o **\_\_texto\_\_**.
- Para *cursiva* usa *\*texto\** o *\_texto\_*.
- Para ~~tachado~~ usa ~~~~texto~~~~.

markdown

Copiar código

Este es un **\*\*proyecto importante\*\*** con *\*características avanzadas\**.

#### 3. Listas Ordenadas y Desordenadas

- Listas desordenadas: utiliza -, \*, o +.
- Listas ordenadas: utiliza números seguidos de un punto.

markdown

Copiar código

- Funcionalidad 1

- Funcionalidad 2

1. Paso uno

2. Paso dos

#### 4. Enlaces e Imágenes

- Enlaces: [Texto del enlace](URL)
- Imágenes: ![Texto alternativo](URL de la imagen)

markdown

Copiar código

□ Para más información, visita [GitHub](https://github.com).

![Logo de GitHub](https://github.githubassets.com/images/modules/logos\_page/GitHub-Mark.png)

#### 1. □ Código y Bloques de Código

- Código en línea: Usa ` alrededor del texto.
- Bloques de código: Usa tres backticks ``` para iniciar y cerrar el bloque.

markdown

Copiar código

Aquí está el comando `git clone` para clonar el repositorio.

```
```bash
```

```
git clone https://github.com/usuario/proyecto-ejemplo.git
```

Copiar código

#### 2. Tablas

- Usa | para crear columnas y --- para definir la línea de encabezado.

markdown

Copiar código

Función   Descripción
-----------------------

-----   -----
---------------

Función 1   Realiza X acción
------------------------------

| Función 2 | Realiza Y acción |

### 3. Citas

- Usa > para resaltar citas o mensajes importantes.

markdown

Copiar código

> Nota: Asegúrate de tener instaladas todas las dependencias.

### Ejemplo Completo de README.md en GitHub

Aquí tienes un ejemplo de un archivo README.md estructurado con Markdown para un proyecto en GitHub:

markdown

Copiar código

# Nombre del Proyecto

## Descripción

Este proyecto realiza la tarea X de manera eficiente y es fácil de instalar y configurar.

## Instalación

1. Clona el repositorio en tu máquina local:

```
``bash
```

```
git clone https://github.com/usuario/proyecto-ejemplo.git
```

2. Navega al directorio del proyecto e instala las dependencias:

```
bash
```

Copiar código

```
cd proyecto-ejemplo
```

```
pip install -r requirements.txt
```

### Uso

- Ejecuta el programa principal.
- Usa el siguiente comando para iniciar:

```
bash
```

Copiar código



python main.py

### **Contribuciones**

Las contribuciones son bienvenidas. Por favor, sigue el flujo de trabajo en [CONTRIBUTING.md](#).

### **Licencia**

Este proyecto está bajo la licencia MIT. Consulta LICENSE para más detalles.

css

Copiar código