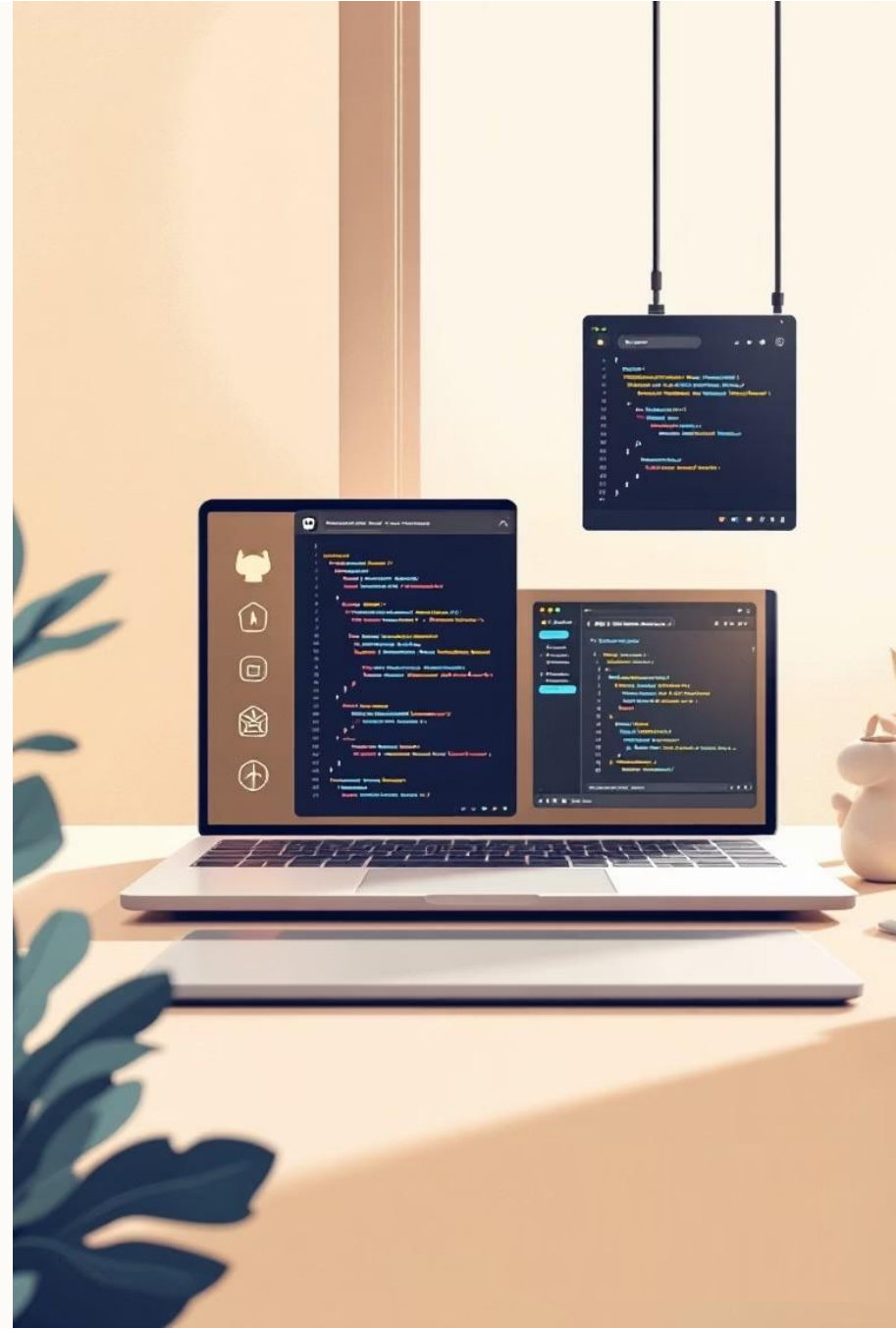


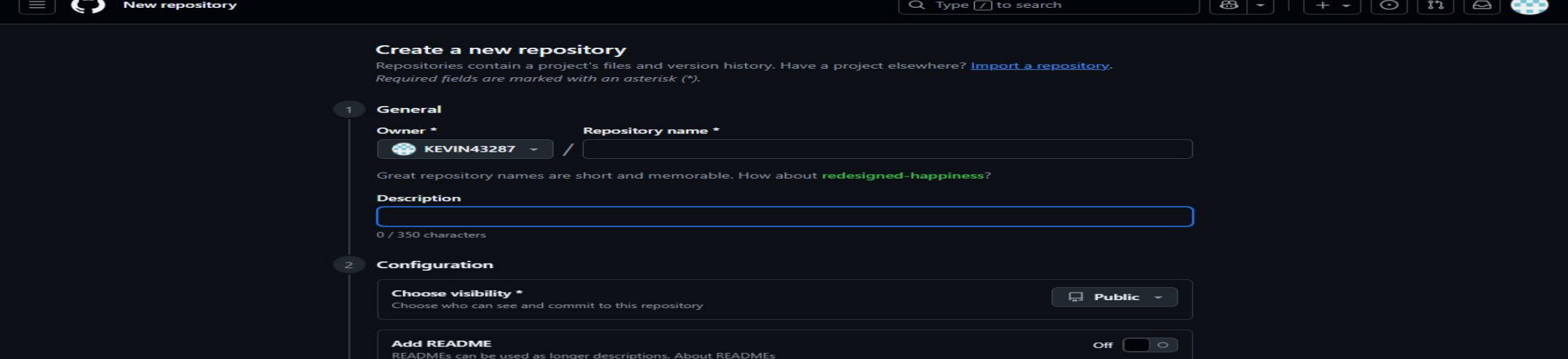
Manual

Cómo subir tu página web de VS Code a GitHub

Esta guía paso a paso te mostrará cómo publicar tu proyecto web directamente desde Visual Studio Code a GitHub, haciéndolo accesible para el mundo.

Kevin Alfredo Macha Bruno





Paso 1: Crear un Nuevo Repositorio en GitHub

El primer paso es preparar un lugar en GitHub para tu proyecto. Un repositorio es como una carpeta de proyecto en la nube.

Accede a GitHub

Inicia sesión en tu cuenta de GitHub. Si no tienes una, regístrate, ¡es gratis!

Nuevo Repositorio

Haz clic en el botón **"New"** o en el signo **"+"** en la esquina superior derecha y selecciona **"New repository"**.

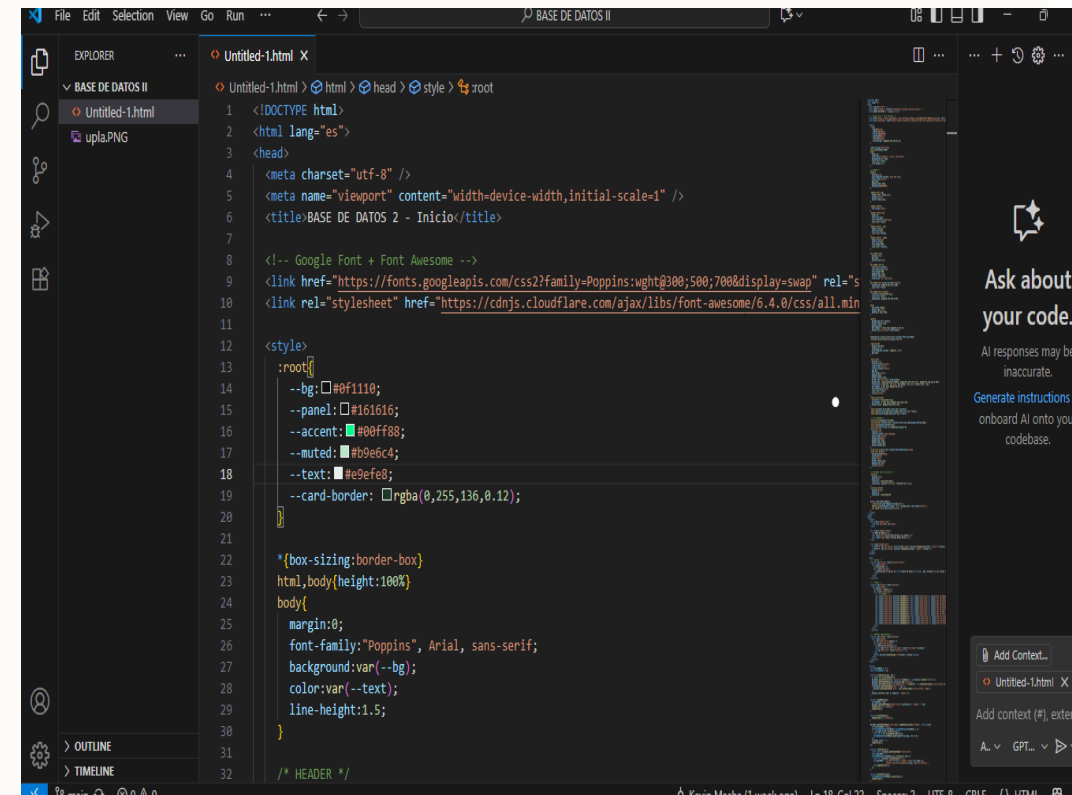
Nombra tu Repositorio

Asigna un nombre descriptivo a tu repositorio, como **mi-pagina-web**. Asegúrate de que sea público si quieres que tu sitio sea visible.

Paso 2: Prepara tu Proyecto en Visual Studio Code

Antes de subir tu código, asegúrate de que todo esté listo y funcionando correctamente en tu entorno local.

- Abre tu proyecto web en VS Code. Esto incluye archivos HTML, CSS, JavaScript y cualquier otro recurso.
- Verifica que tu página se vea y funcione como esperas en tu navegador local.
- Organiza tus archivos en una estructura de carpetas lógica para facilitar la gestión.
- ¡Prepara tu código para ser compartido con el mundo!



Paso 3: Inicializa Git en tu Proyecto Local

Git es el sistema de control de versiones que utilizaremos. Primero, necesitas inicializarlo en la carpeta de tu proyecto.



Abre la Terminal

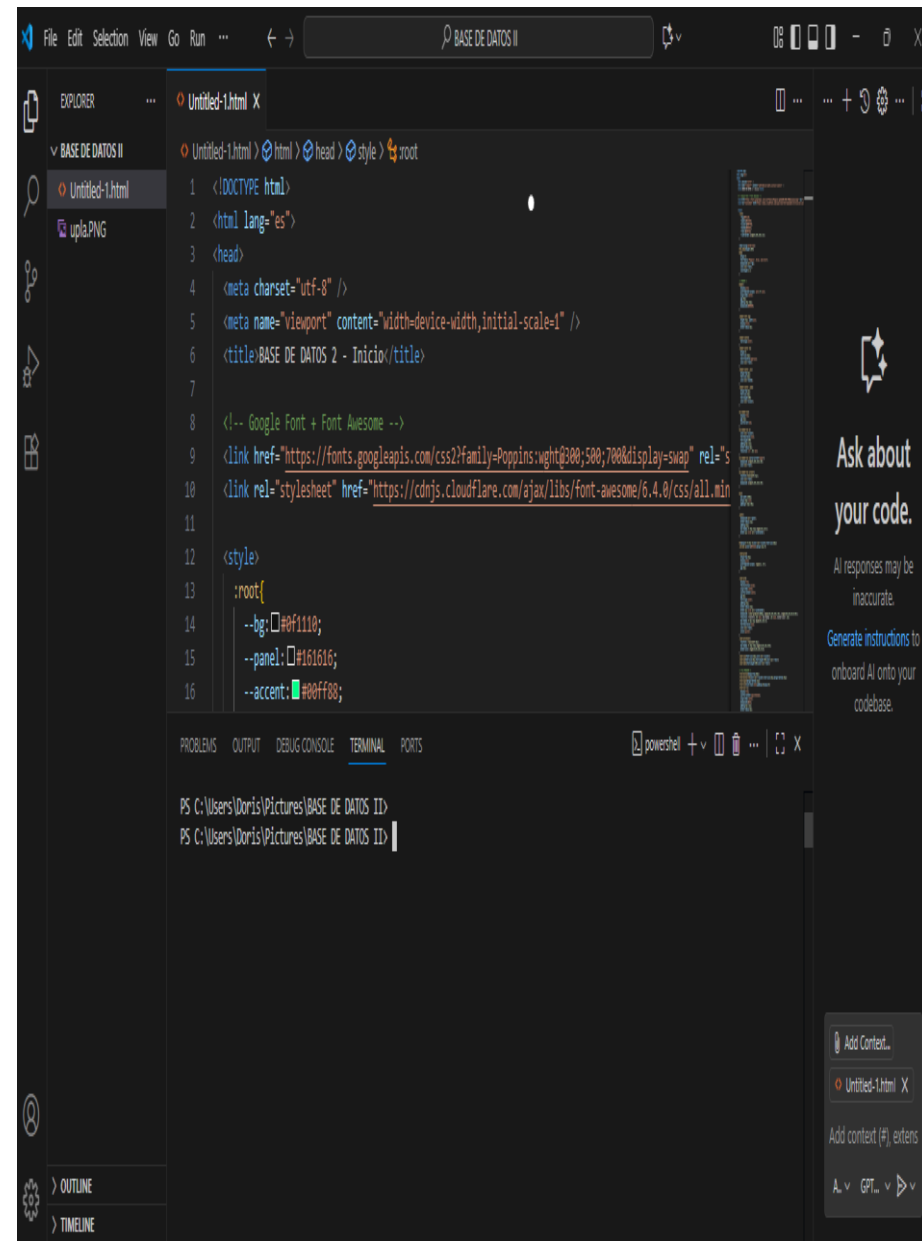
En VS Code, ve a **Terminal > New Terminal**. Esto abrirá una terminal integrada en tu editor.



Inicializa Git

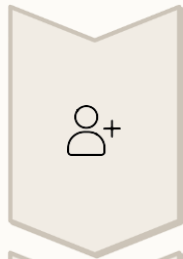
Escribe `git init` y presiona Enter. Esto creará una carpeta oculta `.git` en tu proyecto, indicando que ahora es un repositorio Git local.

Este comando es crucial, ya que transforma una carpeta común en un repositorio Git, permitiéndote rastrear y gestionar cambios.



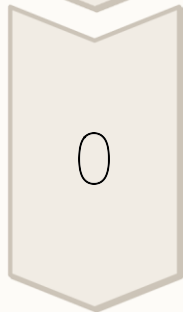
Paso 4: Agrega tus Archivos y Confirma los Cambios

Una vez que Git está inicializado, debes decirle qué archivos quieres que rastree y luego guardar una instantánea de esos archivos.



Añade los Archivos

Usa `git add .` para incluir todos los archivos de tu proyecto en el "staging area", listos para ser confirmados.



Confirma tus Cambios

Escribe `git commit -m "Primer commit de mi página web"`. Esto crea un "commit" o punto de guardado con un mensaje que describe los cambios.

Cada `commit` es un registro de los cambios, como una versión guardada de tu proyecto.

```
Command Prompt
nothing to commit, working tree clean

C:\xampp\htdocs\pisori\pis>clear
clear' is not recognized as an internal or external command,
operable program or batch file.

C:\xampp\htdocs\pisori\pis>git commit -m "Your Message" . && git push origin master
[master 1a7fbad0] Your Message
1 file changed, 1 insertion(+)
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 490 bytes | 81.00 KiB/s, done.
Total 6 (delta 5), reused 0 (delta 0)
To 10.4.0.204:permana/pis.git
540bdb46..1a7fbad0 master -> master
```

Paso 5: Conecta tu Proyecto Local con GitHub

Ahora que tu proyecto está bajo control de versiones localmente, es hora de vincularlo con el repositorio que creaste en GitHub.

¡Importante!

Reemplaza `TU_USUARIO` con tu nombre de usuario de GitHub y `mi-pagina-web.git` con el nombre exacto de tu repositorio.

En la terminal, escribe el siguiente comando:

```
git remote add origin https://github.com/TU_USUARIO/mi-pagina-web.git
```

Este comando establece una conexión, indicando a Git local dónde está el repositorio remoto al que enviarás tu código.

Paso 6: Sube tus Archivos a GitHub

¡Estás a un paso de tener tu página web en línea! Este comando enviará todos tus commits al repositorio remoto en GitHub.

Comando Final:

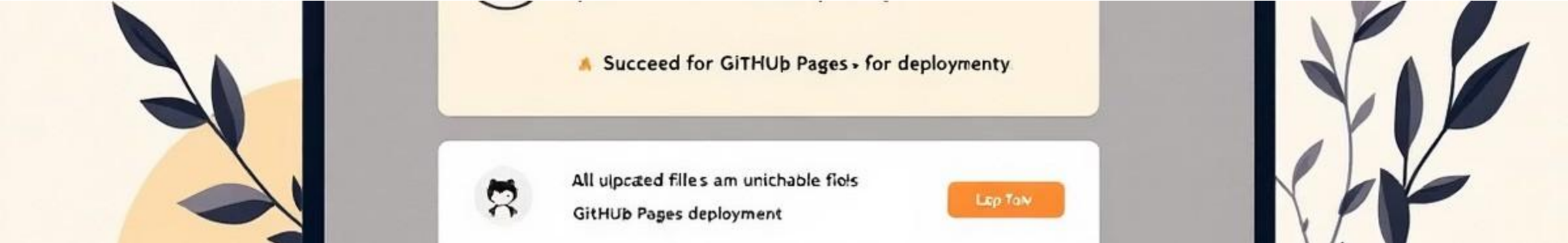
En la terminal, escribe:

```
git push -u origin main
```

Esto subirá la rama `main` (o `master`, dependiendo de tu configuración) de tu repositorio local al repositorio remoto `origin` en GitHub. La primera vez, `-u` establece la conexión de seguimiento.

```
Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git remote add crio "https://github.com/akashadr/Crio.git"

Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git push -u crio master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 209 bytes | 104.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/akashadr/Crio/pull/new/master
remote:
To https://github.com/akashadr/Crio.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'crio'.
```



Paso 7: Verifica tu Página en GitHub

¡Felicidades! Tu proyecto ya debería estar en GitHub. Ahora es el momento de confirmarlo y, opcionalmente, configurar GitHub Pages.



Navega a tu Repositorio

Abre tu navegador y ve a tu cuenta de GitHub. Haz clic en el repositorio que acabas de subir.



Confirma los Archivos

Deberías ver todos tus archivos de proyecto cargados y listos. Esto confirma que la subida fue exitosa.



Configura GitHub Pages

Para publicar tu sitio, ve a **Settings > Pages** en tu repositorio. Selecciona la rama `main` (o `master`) como fuente y guarda. Tu sitio estará disponible en una URL como `https://TU_USUARIO.github.io/mi-pagina-web/`.