Para cargar una página web a **GitHub** y publicarla como un sitio web con **GitHub Pages**, sigue estos pasos:

**1. Crear un repositorio en GitHub**

1. Ve a [GitHub](https://github.com/) e inicia sesión en tu cuenta.
2. Haz clic en el botón **New Repository** (Nuevo repositorio) en la esquina superior derecha.
3. Ponle un nombre a tu repositorio, por ejemplo: mi-sitio-web.
4. Asegúrate de que el repositorio sea **público** (esto es necesario para que funcione con GitHub Pages).
5. Si lo deseas, puedes agregar un archivo README.md (opcional).
6. Haz clic en **Create repository** (Crear repositorio).

**2. Subir los archivos del proyecto a GitHub**

Si tienes **Git** instalado en tu computadora, puedes hacer lo siguiente:

1. Abre una terminal (CMD, Git Bash, etc.) en la carpeta de tu proyecto.
2. Inicializa un repositorio local con el siguiente comando:

bash

Copiar código

git init

1. Añade todos los archivos a tu repositorio local:

bash

Copiar código

git add .

1. Haz el primer commit con un mensaje:

bash

Copiar código

git commit -m "Subiendo mi sitio web"

1. Conecta tu repositorio local al remoto que creaste en GitHub:

bash

Copiar código

git remote add origin https://github.com/tu-usuario/mi-sitio-web.git

(Reemplaza tu-usuario por tu nombre de usuario de GitHub y mi-sitio-web por el nombre de tu repositorio).

1. Sube tu proyecto a GitHub:

bash

Copiar código

git push -u origin master

**3. Habilitar GitHub Pages**

1. Ve al repositorio que acabas de subir en GitHub.
2. Haz clic en la pestaña **Settings** (Configuraciones).
3. Desplázate hacia abajo hasta la sección **Pages**.
4. En el menú desplegable **Branch**, selecciona master o main (dependiendo de cómo nombraste tu rama principal) y la carpeta raíz (/root).
5. Haz clic en **Save** (Guardar).
6. GitHub generará una URL para tu sitio, algo como: https://tu-usuario.github.io/mi-sitio-web/.

**4. Ver tu sitio web**

Después de unos minutos, tu sitio web estará disponible en la URL que GitHub te proporcionó. Solo tienes que acceder a esa dirección y verás tu página web en línea.

**Si no tienes Git instalado:**

Puedes subir los archivos manualmente:

1. Ve al repositorio en GitHub.
2. Haz clic en el botón **Add file** -> **Upload files**.
3. Sube los archivos HTML, CSS, JS, etc.
4. Luego habilita **GitHub Pages** como expliqué en el paso 3.

¡Y eso es todo! Ahora tienes tu página web alojada en **GitHub Pages**.

**Para subir el archivo CNAME a tu repositorio en GitHub, sigue estos pasos:**

**1. Crear el archivo CNAME localmente**

* Crea un archivo llamado CNAME en la raíz de tu proyecto. Asegúrate de que no tenga ninguna extensión (ni .txt ni ninguna otra).
* Dentro de este archivo, escribe tu dominio, por ejemplo:

Copiar código

dagahelpdesk.com

**2. Abrir la terminal (o línea de comandos)**

Navega hasta la carpeta de tu proyecto en tu computadora utilizando la terminal (o Git Bash, si estás en Windows).

**3. Agregar el archivo CNAME a tu repositorio**

Asegúrate de que el archivo CNAME esté en la raíz de tu proyecto (es decir, en el mismo nivel que tus archivos index.html u otros archivos del sitio).

Luego, ejecuta los siguientes comandos:

1. **Agregar el archivo a tu staging area** (preparar el archivo para el commit):

git add CNAME

1. **Hacer commit del archivo CNAME**:

git commit -m "Agregando archivo CNAME para dominio personalizado"

1. **Subir los cambios a GitHub**:

git push origin main

Nota: Si tu rama principal es diferente (como master), usa ese nombre en lugar de main.

**4. Verifica en GitHub**

* Dirígete a tu repositorio en GitHub y verifica que el archivo CNAME se haya subido correctamente en la raíz del proyecto.
* Si ya configuraste los registros DNS correctamente, tu dominio personalizado debería empezar a funcionar poco después de que GitHub procese el archivo CNAME.

**Para subir cambios a tu repositorio en GitHub, sigue estos pasos:**

Para actualizar los cambios que has realizado, pro ejemplo en la página index.html en tu repositorio en GitHub desde Visual Studio Code, sigue estos pasos:

**1. Asegúrate de que todos los archivos estén guardados:**

* Asegúrate de que los cambios en index.html u otros archivos estén guardados (Ctrl + S).

**2. Añadir los cambios al commit:**

* Ve al panel de **Control de Código Fuente** en Visual Studio Code (el icono de la ramita).
* Verás el archivo index.html bajo "Cambios". Haz clic en el icono de **+** (stage) al lado del archivo para añadirlo al commit. Si quieres añadir todos los archivos modificados, haz clic en "Stage All Changes".

**3. Escribe un mensaje de commit:**

* En la parte superior del panel de Control de Código Fuente, encontrarás un campo de texto donde puedes escribir un mensaje para describir los cambios, como por ejemplo:

css

Copiar código

Actualización de la página index.html

* Después de escribir el mensaje, haz clic en el icono de **check** (✓) para hacer el commit de los cambios.

**4. Realizar el push de los cambios:**

* Después de hacer el commit, ahora puedes enviar (push) los cambios al repositorio en GitHub. Para esto:
  + Haz clic en los tres puntos (...) en la parte superior del panel de Control de Código Fuente.
  + Selecciona **Push**. Esto enviará los cambios al repositorio remoto.

**5. Verificar en GitHub:**

* Ve a tu repositorio en GitHub (<https://github.com/eingal/mi-sitio-web>).
* Verifica que los cambios en index.html se hayan reflejado en el repositorio.

Si todo está bien, tu index.html y cualquier otro archivo que hayas modificado deberían estar actualizados en tu repositorio de GitHub.