Para que GitHub muestre tu proyecto público como un sitio web HTML, debes usar **GitHub Pages**, que te permite alojar sitios web estáticos directamente desde un repositorio público en GitHub. Ya hemos creado un proyecto completo para Estudio Maniac 370 con páginas HTML, menús dinámicos y estilos. Ahora, te guiaré paso a paso para subir este proyecto a GitHub, habilitar GitHub Pages y hacer que el sitio sea visible públicamente como un sitio HTML. Esto permitirá que cualquier persona acceda a tu sitio (por ejemplo, en https://tu-usuario.github.io) y lo vea como una página web renderizada, no solo como código.

**Paso 1: Verificar tu Proyecto Local**

Asegúrate de que tu proyecto esté completo y funcional localmente. Basado en lo que hemos trabajado, tu proyecto debería tener una estructura como esta:

proyecto-estudio-maniac-370/

├── index.html

├── indexestrategias.html

├── estrategias-redes-sociales.html

├── estado-contables.html

├── plan-negocios.html

├── resumen-ejecutivo.html

├── analisis-mercado.html

├── estrategia-marketing.html

├── plan-operativo.html

├── proyecciones-financieras.html

├── estrategia-crecimiento.html

├── header-menu.html

├── right-sidebar-menu.html

├── left-sidebar-menu.html

├── styles.css

├── scripts.js

├── load-menus.js

└── server.js

**Probar Localmente**

Antes de subir el proyecto a GitHub, asegúrate de que funcione correctamente en tu máquina:

1. Inicia el servidor local:

bash

node server.js

1. Abre http://localhost:3000 en tu navegador y verifica que todas las páginas (index.html, plan-negocios.html, etc.) se carguen correctamente, incluyendo los menús dinámicos.

Si todo funciona bien, estás listo para subir el proyecto a GitHub.

**Paso 2: Crear o Usar una Cuenta en GitHub**

Si no tienes una cuenta en GitHub, crea una:

1. Ve a [github.com](https://github.com/) y regístrate con tu correo electrónico, un nombre de usuario (por ejemplo, tu-usuario) y una contraseña.
2. Verifica tu correo electrónico para activar tu cuenta.

Si ya tienes una cuenta, inicia sesión.

**Paso 3: Crear un Repositorio Público en GitHub**

GitHub Pages requiere un repositorio público para alojar tu sitio web. Puedes usar un repositorio con un nombre especial (tu-usuario.github.io) para que el sitio se publique en https://tu-usuario.github.io, o puedes usar cualquier repositorio público y publicarlo en https://tu-usuario.github.io/nombre-repositorio. Te explicaré ambas opciones.

**Opción 1: Repositorio de Sitio de Usuario (**tu-usuario.github.io**)**

Esta opción publica tu sitio directamente en https://tu-usuario.github.io.

1. **Crear el Repositorio**:
   * Ve a [github.com](https://github.com/) e inicia sesión.
   * Haz clic en el botón **+** en la esquina superior derecha y selecciona **New repository**.
   * Nombra el repositorio exactamente como tu-usuario.github.io (reemplaza tu-usuario con tu nombre de usuario de GitHub, por ejemplo, juanperez.github.io).
   * Asegúrate de que el repositorio sea **público**.
   * Selecciona **Initialize this repository with a README**.
   * Haz clic en **Create repository**.

**Opción 2: Repositorio de Proyecto (Cualquier Nombre)**

Esta opción publica tu sitio en https://tu-usuario.github.io/nombre-repositorio.

1. **Crear el Repositorio**:
   * Haz clic en **+** y selecciona **New repository**.
   * Dale un nombre cualquiera, por ejemplo, estudio-maniac-370.
   * Asegúrate de que sea **público**.
   * Selecciona **Initialize this repository with a README**.
   * Haz clic en **Create repository**.

Voy a proceder con **Opción 1** (tu-usuario.github.io), ya que es la más común para un modelo de sitio web. Si prefieres la Opción 2, el proceso es muy similar, y te indicaré las diferencias al final.

**Paso 4: Subir tu Proyecto a GitHub**

Vamos a usar Git para subir tu proyecto al repositorio. Si no estás familiarizado con Git, te guiaré paso a paso.

**4.1. Instalar Git (si no lo tienes)**

* Descarga e instala Git desde [git-scm.com](https://git-scm.com/):
  + **Windows**: Descarga el instalador y sigue las instrucciones.
  + **Mac**: Git suele estar preinstalado; verifica con git --version. Si no está, instálalo con Homebrew (brew install git).
  + **Linux**: Instala Git con tu gestor de paquetes (por ejemplo, sudo apt install git en Ubuntu).

**4.2. Configurar Git**

Configura tu identidad en Git:

bash

git config --global user.name "Tu Nombre"

git config --global user.email "tu-correo@example.com"

**4.3. Clonar el Repositorio**

1. Ve a tu repositorio en GitHub (https://github.com/tu-usuario/tu-usuario.github.io).
2. Haz clic en el botón **Code** y copia la URL HTTPS (por ejemplo, https://github.com/tu-usuario/tu-usuario.github.io.git).
3. Abre una terminal (o el Símbolo del Sistema en Windows) y navega a una carpeta donde quieras trabajar:

bash

cd /ruta/a/tu/carpeta

1. Clona el repositorio:

bash

git clone https://github.com/tu-usuario/tu-usuario.github.io.git

Esto creará una carpeta llamada tu-usuario.github.io.

**4.4. Copiar los Archivos del Proyecto**

1. Copia todos los archivos de tu proyecto (index.html, styles.css, etc.) a la carpeta tu-usuario.github.io. Puedes hacerlo manualmente o con un comando:
   * **Linux/Mac**:

bash

cp -r /ruta/a/proyecto-estudio-maniac-370/\* tu-usuario.github.io/

* + **Windows**: Usa el Explorador de Archivos para copiar los archivos.

1. Asegúrate de que index.html esté en la raíz de la carpeta tu-usuario.github.io, ya que GitHub Pages usará este archivo como página principal.

**4.5. Confirmar y Subir los Cambios**

1. Navega a la carpeta del repositorio:

bash

cd tu-usuario.github.io

1. Agrega todos los archivos al control de versiones:

bash

git add .

1. Confirma los cambios:

bash

git commit -m "Subiendo el sitio web de Estudio Maniac 370"

1. Sube los cambios a GitHub:

bash

git push origin main

* + Nota: Si tu rama principal se llama master en lugar de main, usa git push origin master.

**Paso 5: Habilitar GitHub Pages**

Ahora configuraremos GitHub Pages para que publique tu sitio web.

1. Ve a tu repositorio en GitHub (https://github.com/tu-usuario/tu-usuario.github.io).
2. Haz clic en la pestaña **Settings** (Configuración).
3. Desplázate hacia abajo hasta la sección **Pages** (en la categoría "Code and automation").
4. En la sección **Build and deployment**:
   * En **Source**, selecciona **Deploy from a branch**.
   * En **Branch**, selecciona main (o master, dependiendo de tu rama principal) y asegúrate de que la carpeta sea / (root).
   * Haz clic en **Save**.
5. GitHub Pages comenzará a construir tu sitio. Esto puede tomar entre 1 y 10 minutos.

**Verificar que el Sitio Está Activo**

* Después de unos minutos, regresa a la sección **Pages** en las configuraciones de tu repositorio.
* Deberías ver un mensaje que dice: "Your site is live at [https://tu-usuario.github.io](https://tu-usuario.github.io/)".
* Abre esa URL en tu navegador (por ejemplo, https://juanperez.github.io) para ver tu sitio web renderizado como HTML.

**Paso 6: Probar el Sitio Público**

1. **Acceder al Sitio**:
   * Abre https://tu-usuario.github.io en tu navegador.
   * Deberías ver la página principal (index.html) renderizada correctamente, con los estilos aplicados y los menús dinámicos funcionando.
2. **Navegar por las Páginas**:
   * Haz clic en los enlaces del menú del encabezado (por ejemplo, "Plan de Negocios") para ir a https://tu-usuario.github.io/plan-negocios.html.
   * Verifica que las barras laterales (izquierda y derecha) se carguen correctamente.
   * Asegúrate de que las páginas de las etapas del plan de negocios (resumen-ejecutivo.html, analisis-mercado.html, etc.) también funcionen.
3. **Solucionar Problemas (si los hay)**:
   * Si los menús no se cargan, abre la consola del navegador (presiona F12 y ve a la pestaña "Console") para ver si hay errores.
   * Un error común es un problema con las rutas de los archivos. En GitHub Pages, las rutas son relativas a la raíz del repositorio. Asegúrate de que en load-menus.js las URLs de fetch sean correctas:

javascript

fetch('header-menu.html')

fetch('right-sidebar-menu.html')

fetch('left-sidebar-menu.html')

Estas rutas asumen que los archivos están en la raíz del repositorio, lo cual debería ser correcto si seguiste los pasos anteriores.

**Paso 7: Hacer que el Proyecto Sea Público y Compartible**

Tu repositorio ya es público (lo configuramos como tal al crearlo), lo que significa que:

* Cualquier persona puede ver el código fuente en https://github.com/tu-usuario/tu-usuario.github.io.
* Cualquier persona puede visitar el sitio renderizado en https://tu-usuario.github.io.

**Compartir tu Sitio**

* Comparte la URL https://tu-usuario.github.io con otros para que vean tu proyecto.
* Si quieres que otros puedan contribuir al proyecto, puedes invitarlos como colaboradores:
  1. Ve a tu repositorio en GitHub.
  2. Haz clic en **Settings** > **Collaborators**.
  3. Agrega a otros usuarios de GitHub como colaboradores.

**Paso 8: Diferencias si Usas la Opción 2 (Repositorio de Proyecto)**

Si usaste un nombre diferente para el repositorio (por ejemplo, estudio-maniac-370), el proceso es casi idéntico, pero con estas diferencias:

* La URL de tu sitio será https://tu-usuario.github.io/estudio-maniac-370 en lugar de https://tu-usuario.github.io.
* Cuando configures GitHub Pages, asegúrate de seleccionar la rama main (o master) y la carpeta / (root).

El resto del proceso (subir archivos, habilitar GitHub Pages, probar el sitio) es el mismo.

**Paso 9: Actualizar el Instalador (Opcional)**

Si deseas que tu instalador (install.sh o install.bat) automatice la creación y subida del repositorio a GitHub, necesitarías usar la API de GitHub o herramientas como gh (GitHub CLI). Sin embargo, esto requiere autenticación y puede ser complicado para un script simple. En su lugar, te recomiendo seguir los pasos manuales anteriores.

Si aún quieres automatizarlo, puedes instalar GitHub CLI (gh):

1. Descarga e instala GitHub CLI desde [cli.github.com](https://cli.github.com/).
2. Autentícate con GitHub:

bash

gh auth login

1. Modifica el instalador para crear el repositorio y subir los archivos usando gh. Por ejemplo, en install.sh podrías agregar:

bash

gh repo create tu-usuario.github.io --public --source=. --remote=origin --push

Esto crea el repositorio y sube los archivos automáticamente.

**Notas y Mejoras**

1. **Errores Comunes**:
   * **"404 Not Found" al visitar la URL**: Asegúrate de que index.html esté en la raíz del repositorio y que hayas habilitado GitHub Pages correctamente.
   * **Menús no se cargan**: Verifica que los archivos header-menu.html, right-sidebar-menu.html y left-sidebar-menu.html estén en la raíz del repositorio y que las rutas en load-menus.js sean correctas.
   * **Estilos o scripts no se cargan**: Asegúrate de que las rutas en tus archivos HTML (por ejemplo, <link rel="stylesheet" href="styles.css">) sean correctas.
2. **SEO y Accesibilidad**:
   * Para que tu sitio sea más visible en motores de búsqueda, asegúrate de que cada página tenga un título único (<title>) y una meta descripción (<meta name="description">).
   * Agrega etiquetas alt a cualquier imagen que añadas en el futuro para mejorar la accesibilidad.
3. **Dominio Personalizado (Opcional)**:
   * Si tienes un dominio (por ejemplo, www.estudiomaniac370.com), puedes configurarlo en GitHub Pages:
     1. Ve a **Settings** > **Pages** en tu repositorio.
     2. En **Custom domain**, ingresa tu dominio.
     3. Configura los registros DNS en tu registrador para apuntar a GitHub Pages (consulta la documentación de GitHub Pages para los detalles).
4. **Hacer un Fork del Repositorio (Opcional)**:
   * Si quieres que otros puedan usar tu proyecto como plantilla, puedes habilitar la opción **Template repository**:
     1. Ve a **Settings** en tu repositorio.
     2. Marca la casilla **Template repository**.
     3. Esto permitirá que otros hagan un "fork" de tu repositorio y lo usen como base para sus propios proyectos.

**Resultado Final**

* Tu proyecto estará disponible públicamente en https://tu-usuario.github.io (o https://tu-usuario.github.io/estudio-maniac-370 si usaste un nombre diferente para el repositorio).
* Cualquier persona podrá visitar esa URL y ver tu sitio web renderizado como HTML, con todas las páginas, menús dinámicos y estilos funcionando.
* El código fuente estará visible en https://github.com/tu-usuario/tu-usuario.github.io para que otros puedan inspeccionarlo o contribuir.

Si tienes algún problema (por ejemplo, el sitio no se publica o hay errores al cargar las páginas), déjame saber y te ayudaré a solucionarlo. ¿Te gustaría agregar algo más, como un dominio personalizado o una página adicional? ¡Avísame!