






## Crear sitio web con 3 páginas sobre tu animal preferido.

Deberás crear un sitio web sobre animales controlado por Git  y más tarde crear un repositorio en Github  donde subirás todo el proyecto. También publicarás el contenido en tu dominio mediante Filezilla .


### Paso 1. Crear carpeta en tu ordenador

En tu ordenador crea una **carpeta** llamada “animal-preferido” dentro de la carpeta de “sitio web”.

### Paso 2. Área de trabajo en Visual Studio Code

Configura en **Visual Studio Code** un **área de trabajo** a la que añadirás la carpeta “animal-preferido”.

### Paso 3. Iniciar repositorio

Inicia en la carpeta “animal-preferido” un repositorio de **Git** .

Para este proyecto puedes trabajar en una única rama que se llame **main** (deberás renombar la rama **master** que genera Git por defecto al nuevo nombre **main** con el comando `git branch -m` ).

### Paso 4. Crear archivos

Haz **3 archivos HTML** con **diseño libre**, una estructura básica y un **CSS** sencillo, sin tener en cuenta el SEO.

Mientras generas la construcción de las páginas **genera varios commit** que **guarden puntos clave del desarrollo del trabajo**, por ejemplo:

- ✓ Cuando ya tengas el HTML de cada una de las páginas bien creado.
- ✓ Cuando tengas la info completa, cuando el CSS ya parezca correcto.
- ✓ Cuando generes una modificación de la estructura o CSS que te parezca una mejora importante.

ESTAS SON LAS 3 PÁGINAS QUE DEBES HACER:

#### ➊ 1. **index.html** será la página inicial que tendrá:

- Cabecera con imagen de fondo de tu animal preferido.
- Menú de navegación para enlazar con las otras dos páginas.
- Imagen dentro del contenido con otra imagen de tu animal preferido.
- Texto breve con **información general** de tu animal preferido.
- Footer con copyright.

➡ **2. arte-nombre-animal.html** con información de la presencia de ese animal en el mundo del arte que tendrá:

- Cabecera con imagen de fondo de tu animal preferido.
- Menú de navegación para enlazar con las otras dos páginas.
- alguna imagen o imágenes con apariciones de ese animal en el cine, televisión, literatura, pintura...por ejemplo si eliges el elefante podrías buscar alguna imagen de la película de “Dumbo”
- Texto breve con **información de esa aparición del animal en el mundo del arte**, por ejemplo si hablas de la película “Dumbo” puedes detallar cuando se estrenó, su argumento, protagonistas, etc.

➡ **3. contacto.html** con un formulario que tendrá:

- Cabecera con imagen de fondo de tu animal preferido.
- Menú de navegación para enlazar con las otras dos páginas.
- Breve texto solicitando al visitante que mande la información que le gustaría publicar en nuestra página sobre nuestro animal favorito (anécdotas personales, información de interes...)
- Formulario con campos para preguntar **nombre**, **compartir la información** y **aceptar** política de privacidad (la política de privacidad sería un enlace a esas condiciones de envío, pero puedes simplemente colocar un enlace con el href="#" ):

Nombre

Comparte tu información...

☒ Acepto la Política de Privacidad

**ENVIAR INFORMACIÓN**

## Paso 5. Crear repositorio vacío en Github

Crea un repositorio vacío en tu cuenta de **Github**  para subir el sitio web de tu animal preferido.


## Paso 6. Conectar repositorios

Conecta tu repositorio local de **Git**  con el repositorio que has creado en **Github**  y sube los archivos.

## Paso 7. Generar README.md

Tras subir los archivos de local al remoto, desde **Github**, genera un archivo **README.md** para contar brevemente el objetivo de este trabajo que es compartir información de tu animal favorito.

## Paso 8. Subir con Filezilla a tu dominio


En tu dominio personal conectándote mediante **Filezilla** , genera una carpeta que se llame como tu animal favorito, por ejemplo si has elegido el elefante la carpeta se puede llamar “elefantes”.

Mediante **Filezilla**  sube los archivos a la carpeta de tu dominio que has generado para este trabajo.

Finalmente se debería poder visualizar desde Internet el contenido publicado. Por ejemplo si tu dominio es <http://davidmartin.com> el nuevo contenido se debería ver en <http://davidmartin.com/elefantes>

## Paso 9. Nueva rama con mejoras SEO

Ahora vamos a mejorar los contenidos optimizando el SEO de esta página, para lo cual deberás **trabajar en local** con los archivos html creados.

Antes de comenzar a modificar los archivos, debes crear en tu repositorio **Git**  **una rama nueva** a la que llamaremos **mejoraseo**.


**Trabajarás la mejora del SEO en esta rama nueva.**

Genera algún **commit** en tu repositorio **Git**  con puntos clave de la mejora SEO.

Sigue estos pasos para optimizar tu SEO:

1. **Decide las palabras clave** por las que quieres se encontrado en cada una de las páginas HTML que has creado. Por ejemplo si tu animal es el elefante:
  - ✓ La página de inicio podría intentar se encontrada por “elefante” ,”información sobre elefantes” , ”el mundo de los elefantes”
  - ✓ La página del arte podría intentar se encontrada por “elefante en el cine” , “elefante en el arte”
  - ✓ La página de contacto podría intentar se encontrada por “compartir tu afición por los elefantes”.
2. **Revisa** tu código HTML para reforzar la aparición de las palabras clave elegidas dentro la estructura. **Inserta** etiquetas nuevas si son necesarias para mejorar.
3. **Reescribe** si es necesario algún texto para mejorar las palabras clave elegidas de cada página.
4. Analiza con algún software, por ejemplo **SEOquake**, el estado de tu implementación SEO.
5. Genera un **sitemap.xml** que publicarás y además enviaras a Google en tu Search Console.
6. Genera un archivo **robots.txt** que de momento solo tendrás:

```
User-agent: *
Allow: /
Sitemap: url de este nuevo sitemap
```

\*  el archivo robots.txt debería estar situado en la raíz del dominio, pero para este trabajo de ejemplo podemos dejarlo en la subcarpeta creada.

## Paso 10. Subir rama a Github

Sube la nueva rama **mejoraseo** a **Github** .



Cuando intentes hacer `git push` para subir la rama local al remoto te informará de un error porque esa rama no existía en remoto, pero en la información que aparece tienes la solución.

## Paso 11. Hacer algún cambio más


Haz algún cambio más en **local** dentro de la rama **mejoraseo** y sube de nuevo esos cambios a **Github** .

## Paso 12. Merge


Una vez que la optimización SEO esté finalizada, en **Git**  vamos a pasar los cambios generados en la rama **mejoraseo** a la rama **main** mediante un **merge**.

## Paso 13. Subir cambios a Github y dominio

Sube todos los cambios a tu **repositorio** de **Github** .

Sube también mediante **Filezilla**  todos los cambios en tu **dominio**, para que se vea en tu **URL** publicada.

## Paso 14. Eliminar ramas

Si consideramos que la rama **mejoraseo** ya ha cumplido su misión podemos **eliminarla** de **Git**  ejecutando este comando: `git branch -d mejoraseo`

Si también queremos **eliminar** la rama **mejorarseo** que aun está en el repositorio remoto de **Github**  puedo ejecutar este comando: `git push origin --delete mejoraseo`