

# — Cómo comentar en CSS

```
.main-nav {  
  display: block; /* Mostramos el menú */  
}
```

```
.menu-icon {  
  display: none; /* Ocultamos el ícono de menú */  
}
```

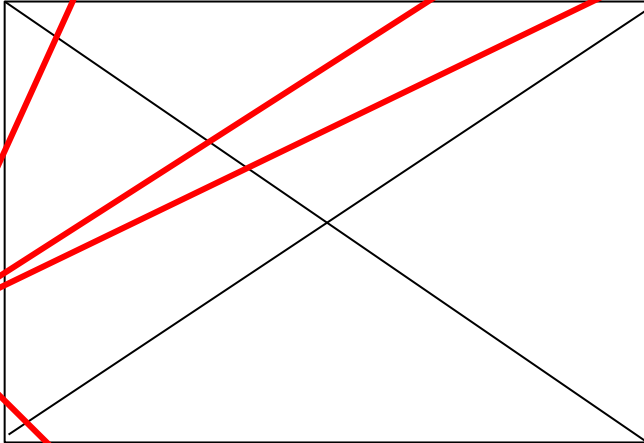
# Consistencia de Interfaz

Se refiere a la uniformidad y coherencia en la presentación y comportamiento de los elementos dentro de un sitio o aplicación.

Logo | nombre

Barra de navegación con links a las páginas

Título de la Página/Sección



---

---

---

---

---

---

---

**BOTON**

Caja de contenido, de nota, evento,  
ect.

Caja de contenido, de nota, evento,  
ect.

Pie de página con íconos de redes sociales y datos de contacto

Siempre  
presente en  
TODAS LAS  
PÁGINAS  
en el mismo  
orden

# Estructura de mi carpeta de trabajo

## Carpeta de mi sitio

index  
.html

galeria  
.html

personajes  
.html

backstage  
.html

merchandize  
.html

estilos  
.css

## Carpeta imágenes

imagen1  
.jpg

imagen2  
.jpg

imagen3  
.jpg

# Responsive Web Design

Técnica de desarrollo web que busca que los sitios web se vean y funcionen de manera óptima en todos los dispositivos.



# MediaQueries

**“Consulta al Medio”**  
permite aplicar estilos CSS  
específicos según el tamaño de la  
pantalla o el dispositivo.

Se consulta al navegador para  
saber el tipo de pantalla y su  
tamaño para determinar el CSS  
a utilizar:

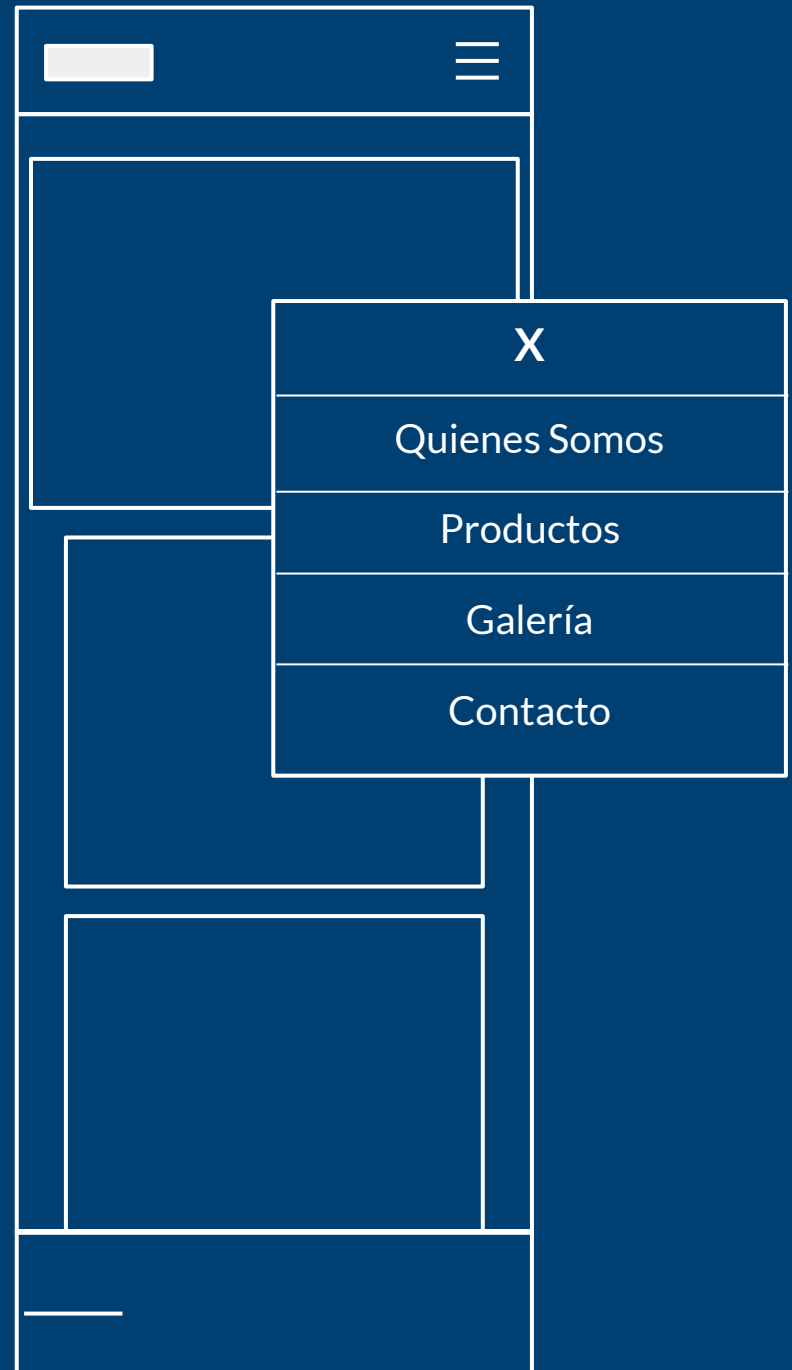
- **Desktop 1300px**
- **Tablets 768px**  
`@media (max-width: 768px) {`  
`}`
- **Smartphone 480px**  
`@media (max-width: 480px) {`  
`}`

---

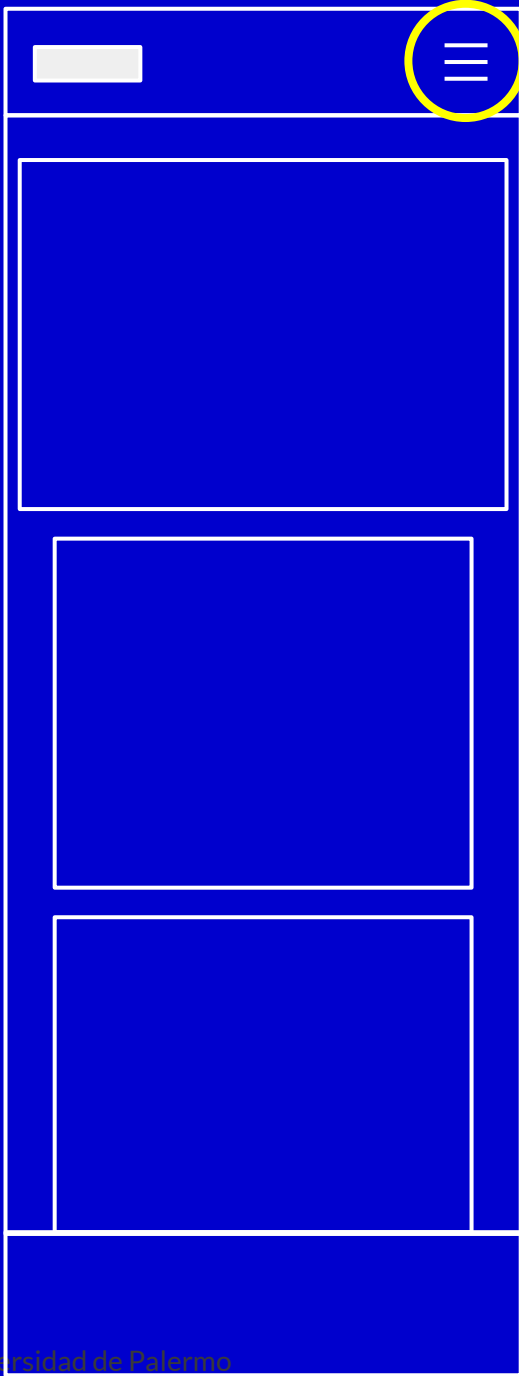
```
<meta name="viewport"  
content="width=device-  
width, initial-scale=1.0">
```

# Barra de Navegación Responsive

En la vista para celulares se oculta el navegador, y aparece el ícono, que es el que abre y cierra la barra de navegación







# JavaScript

Es el lenguaje de programación para la web.

**HTML** define el **contenido**

**CSS** especifica el **diseño**

**JavaScript** programa el **comportamiento**

Puede actualizar y cambiar atributos tanto de HTML como de CSS.

**JavaScript** se ejecuta en el navegador del usuario.

El código de JavaScript se inserta en el HTML entre las etiquetas **<script>** y **</script>**.

Uno de los métodos de HTML más usados de JavaScript es **getElementById()**.

Una **función** es un bloque de código diseñado para realizar una tarea en particular, y se ejecuta cuando “**algo**” la “**llama**”.

# Menú Responsive

```
<header class="site-header">  
  <div class="logo">Mi Logo</div>  
  <!-- Ícono de menú: visible sólo en móviles -->  
  <div class="menu-icon">≡</div>  
  <!-- Menú principal -->  
  <nav class="main-nav">  
    <a href="#">Inicio</a>  
    <a href="#">Servicios</a>  
    <a href="#">Contacto</a>  
  </nav>  
</header>
```

# Visualización por defecto (Escritorio y tabletas):

Se muestra el menú y se oculta el ícono.

```
.main-nav {  
  display: block; /* Mostramos el menú */  
}
```

```
.menu-icon {  
  display: none; /* Ocultamos el ícono de menú */  
}
```

# Media Query para Celulares

```
@media (max-width: 480px) {  
  .main-nav {  
    display: none; /* Ocultamos el menú en móviles */  
  }  
  
  .menu-icon {  
    display: block; /* Mostramos el ícono de menú */  
    cursor: pointer;  
    font-size: 1.8em;  
  }  
}
```

# Toggling del Menú con JavaScript

```
<script>
```

```
// Selecciona el ícono y el menú por el nombre de la clase  
const menuIcon = document.querySelector(".menu-icon");  
const mainNav = document.querySelector(".main-nav");
```

```
// Agregamos el evento de clic al ícono  
menuIcon.addEventListener("click", function() {  
    // Alterna el display del menú:  
    // Si está oculto, se muestra, y viceversa.  
    if (mainNav.style.display === "block") {  
        mainNav.style.display = "none";  
    } else {  
        mainNav.style.display = "block";  
    }  
});
```

```
</script>
```

# Toggling que agregue una **clase**

```
<script>
  document.addEventListener('DOMContentLoaded', function() {
    const menulcon = document.querySelector('.menu-icon');
    const mainNav = document.querySelector('.main-nav');

    menulcon.addEventListener('click', function() {
      mainNav.classList.toggle('open');
    });
  });
</script>
```

# Posicionamiento

La **propiedad position** en **CSS** te permite decidir dónde exactamente quieres colocar cada elemento en la pantalla, y cómo se comporta cuando se mueve la pantalla.

**Static** - (por defecto) se coloca en el lugar donde lo escribiste en el código, siguiendo el orden natural.

**Relative**: Se mueves un poco de su posición original, pero sin afectar a los demás elementos. También sirve como ancla para elementos interiores con **Absolute**.

**Absolute**: Se despegas y se coloca donde uno quiera, sin importar dónde estaba antes.

**Fixed**: Se pega a la ventana del navegador. Aunque muevas la pantalla, siempre se queda en el mismo lugar.

**Sticky**: Una combinación de **relative** y **fixed**. Se comporta como **relative** hasta que llega a un punto específico de la página, y a partir de ahí se comporta como **fixed**.



# Propiedades necesarias de Position

Además de **position**, hay otras propiedades que trabajan juntas para posicionar los elementos:

**top:** Mueve el elemento hacia arriba.

**right:** Mueve el elemento hacia la derecha.

**bottom:** Mueve el elemento hacia abajo.

**left:** Mueve el elemento hacia la izquierda.

La **propiedad position** te da un gran control sobre cómo se colocan los elementos en la pantalla.

## Antoni Gaudí

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec augue lacus, aliquam in, luctus in, gravida vel, mi. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; [Nunc aliquet](#) ultrices ante Mauris nunc dolor, egestas vel, auctor sed, dignissim vitae, enim. Phasellus tincidunt. Phasellus vel wisi id neque cursus rhoncus. Morbi cursus tristique est.



### Título noticia 2

Nulla pulvinar dui sed augue. Praesent laoreet arcu in justo. Donec velit. Nam malesuada bibendum metus.

[ver más](#)

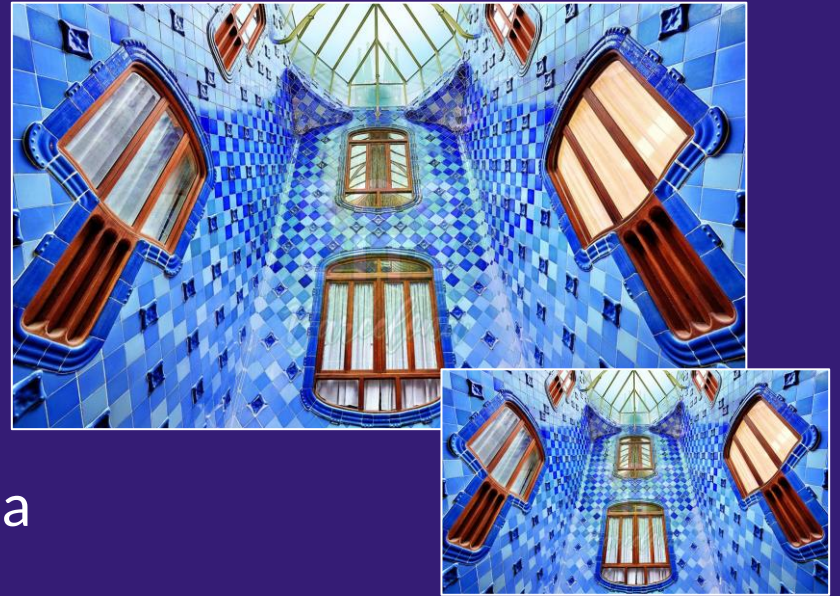


# Imágenes Responsive

Creando imágenes responsivas con la etiqueta <picture>

<https://imagecompressor.com/es/>

<https://pixlr.com/es/express/>



# <picture>

Contenedor diseñado específicamente para manejar múltiples versiones de una misma imagen, donde las imágenes se adaptan automáticamente al tamaño de la pantalla del dispositivo.

**<source>** Especifica diferentes versiones de la imagen.

**media:** Atributo que define el medio para el que está destinada la imagen, usando consultas de media para determinar el ancho de la pantalla.

**srcset:** Atributo que indica la URL de la imagen.

**<img>:** Es la etiqueta de la imagen por defecto que se mostrará si ninguna de las opciones de **<source>** coincide.

---

# Estructura básica <picture>

HTML

<picture>

<source media="(min-width: 768px)" srcset="imagen-grande.jpg">

<source media="(min-width: 480px)" srcset="imagen-mediana.jpg">



</picture>

# Agregar Clase al Desplazar

```
$(window).scroll(function(){  
  
    if ($(this).scrollTop() > 50) {  
        $('nav').addClass('nuevaclase');  
  
    } else {  
        $('nav').removeClass('nuevaclase');  
    }  
  
});
```

# SLIDER

html:

```
<div class="cajaslider">  
    
    
    
  
  <a class="prev" onclick="prevSlide()">&#10094;</a>  
  <a class="next" onclick="nextSlide()">&#10095;</a>  
</div>
```

JavaScript:

```
let currentSlide = 0;
```

```
function showSlide(index) {  
  const slides = document.querySelectorAll('.slide');
```

# JavaScript

Puede actualizar y cambiar tanto HTML como CSS.

Primero se establece **currentSlide** en 1. (Primera imagen)

Luego llama a `showSlide()` para mostrar la primera imagen.

La función `showSlide()` oculta (`display = "none"`) todos los elementos con el nombre de clase "**slide**", y muestra (`display = "block"`) el elemento con el **currentSlide** dado.

Si **currentSlide** es mayor que el número de elementos (`x.length`), **currentSlide** se establece en cero.

Si **currentSlide** es menor que 1, se establece en número de elementos (`x.length`).



Slider de imagenes

**El tamaño de las imágenes  
debe ser el mismo**

Caja de contenido, de  
nota, evento, ect.

Caja de contenido, de  
nota, evento, ect.

Caja de contenido, de  
nota, evento, ect.

---

# Navegación en una misma pantalla

Se definen anclas con ID, y se vinculan con un # adelante

`id="parte1"`

`<a href="#parte1">ver sección</a>`

`html { scroll-behavior: smooth; }`

---

# Slider - Transición animada

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $("a").on('click', function(event) {
        if (this.hash !== "") {
            event.preventDefault();
            var hash = this.hash;
            $('html, body').animate({
                scrollTop: $(hash).offset().top
            }, 800, function(){
                window.location.hash = hash;
            });
        } // End if
    });
});
</script>
```

---

# Publicar Sitio con GitHub



Plataforma de desarrollo colaborativo que permite almacenar, compartir y trabajar en proyectos de código, gestionando sus versiones. Además, incluye herramientas para publicar sitios web, gestionar proyectos y colaborar en código de forma remota.

<https://github.com/>

# Crear Repositorio con GitHub

En la esquina superior derecha, hacer clic en el botón “+” y seleccionar **“New repository”**.

- Darle un **nombre** al repositorio (por ejemplo, “mi-sitio-web”)
- Selecciona la opción **“Public”** para que sea accesible públicamente.
- Marca la casilla “Add a README file”.
- Haz clic en **“Create repository”**.

# Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (\*).

Owner \*



Repository name \*

misitio

✔ misitio is available.

Great repository names are short and memorable. Need inspiration? How about **congenial-tribble** ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

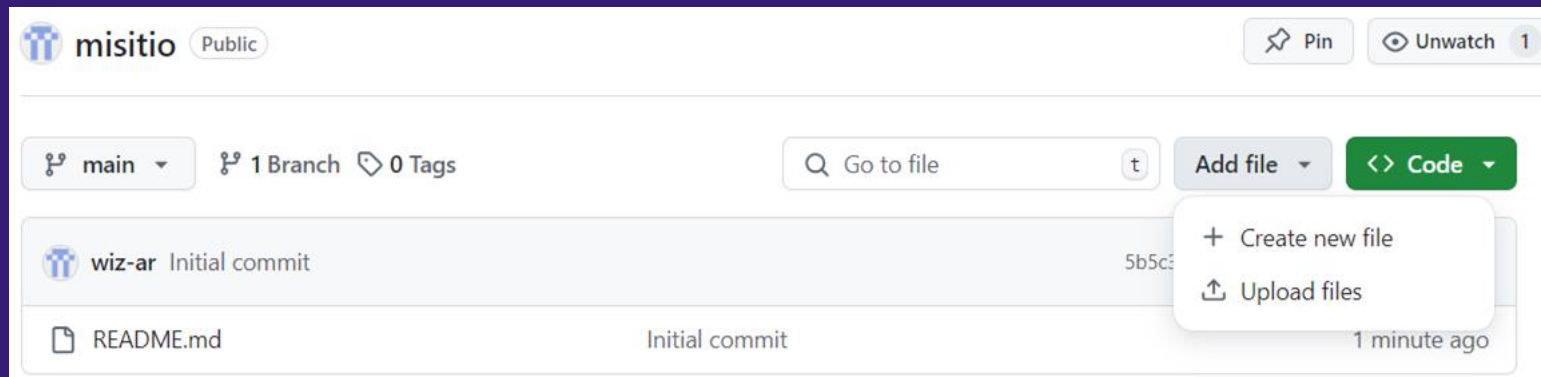


You are creating a public repository in your personal account.

Create repository

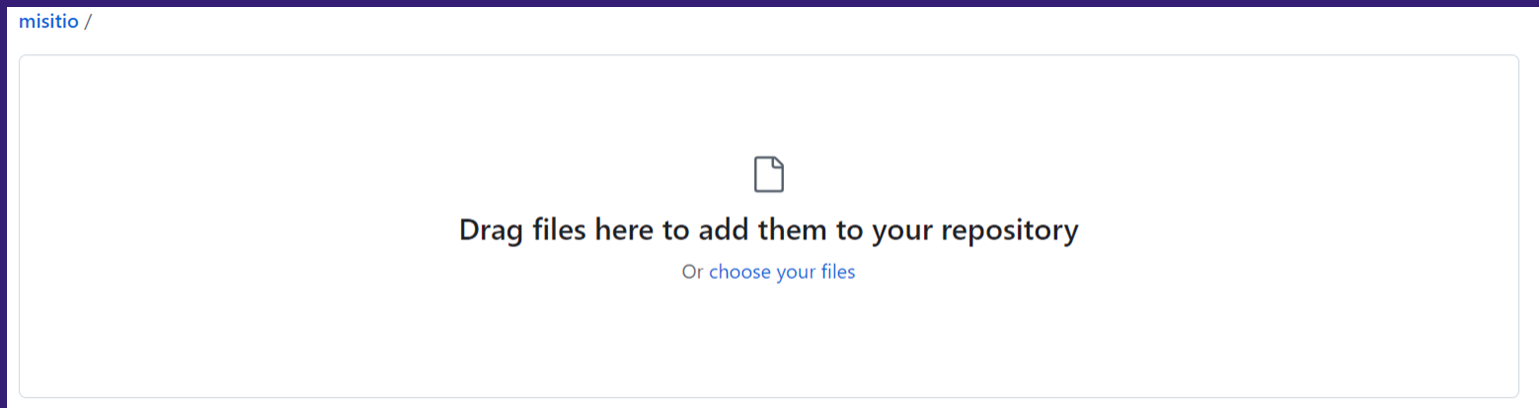
# Subir los archivos del Sitio

1. En la página del nuevo repositorio, hacer clic en **Add file** y seleccionar **Upload files**.



# Subir los archivos del Sitio

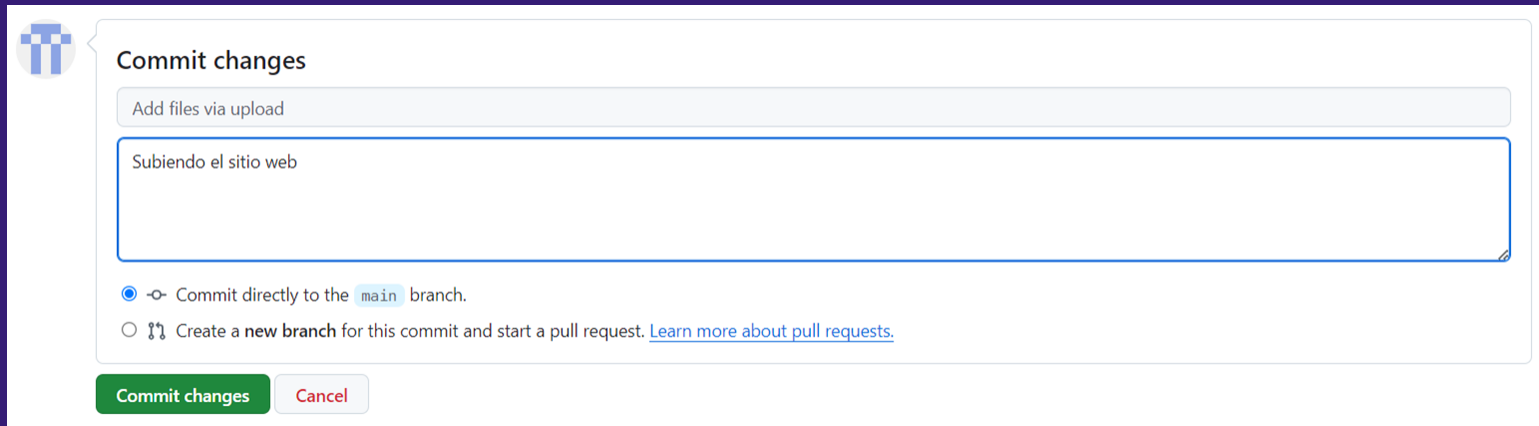
2. Arrastrar y soltar los archivos de la carpeta del sitio web en el área marcada, o hacer clic en **“choose your files”** para seleccionarlos desde la computadora.








# Subir los archivos del Sitio

3. Una vez que los archivos estén cargados, hacer clic en “**Commit changes**”. Esto confirma que los archivos o cambios que se suben están completos y listos para ser guardados en el repositorio de manera oficial. Es un paso esencial para mantener el control sobre las versiones del proyecto y asegurar de que los cambios queden bien registrados.



The screenshot shows the 'Commit changes' dialog box in GitHub. It features a header with the GitHub logo and the title 'Commit changes'. Below the header is a section titled 'Add files via upload' which contains a text area with the message 'Subiendo el sitio web'. At the bottom of the dialog, there are two radio button options: the first is selected and reads 'Commit directly to the `main` branch.', and the second is unselected and reads 'Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)'. At the very bottom, there are two buttons: a green 'Commit changes' button and a grey 'Cancel' button.

 main ▾

 1 Branch  0 Tags


 Go to file

 t


Add file ▾

 Code ▾



wiz-ar Add files via upload 

05c2016 · 1 minute ago

 2 Commits



css

Add files via upload

1 minute ago



js

Add files via upload

1 minute ago



README.md

Initial commit

14 minutes ago



anuncio.html

Add files via upload

1 minute ago



contacto.html

Add files via upload

1 minute ago



email.html

Add files via upload

1 minute ago



estilos.css

Add files via upload

1 minute ago



galeria.html

Add files via upload

1 minute ago



index.html

Add files via upload

1 minute ago



presentacion.html

Add files via upload

1 minute ago



productos.html

Add files via upload

1 minute ago



que\_hacemos.html

Add files via upload

1 minute ago



quienes\_somos.html

Add files via upload

1 minute ago



respuesta.html

Add files via upload

1 minute ago



script.js

Add files via upload

1 minute ago



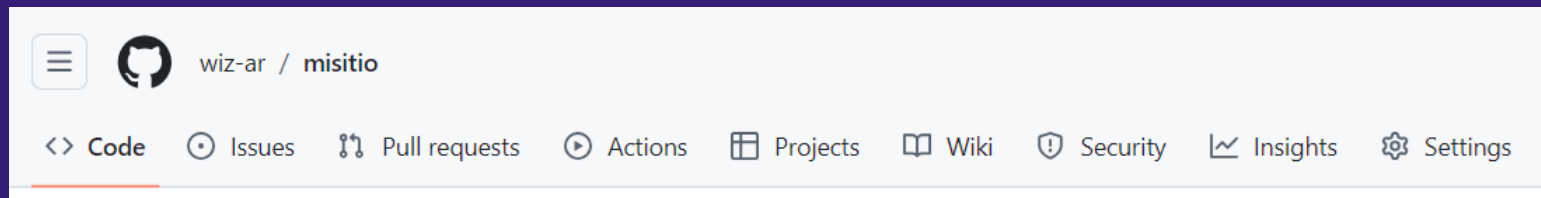
word.html

Add files via upload

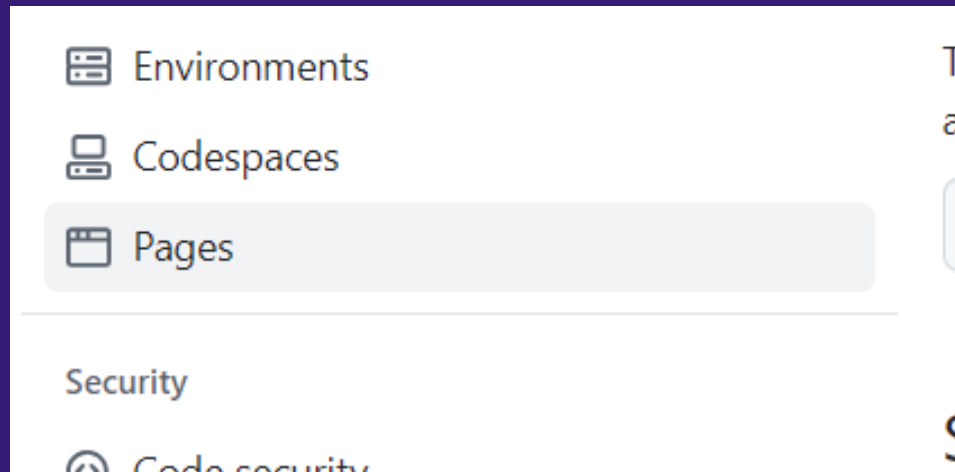
1 minute ago

# Publicar el sitio con GitHub Pages

Después de subir los archivos, hacer clic en la pestaña **“Settings”**.

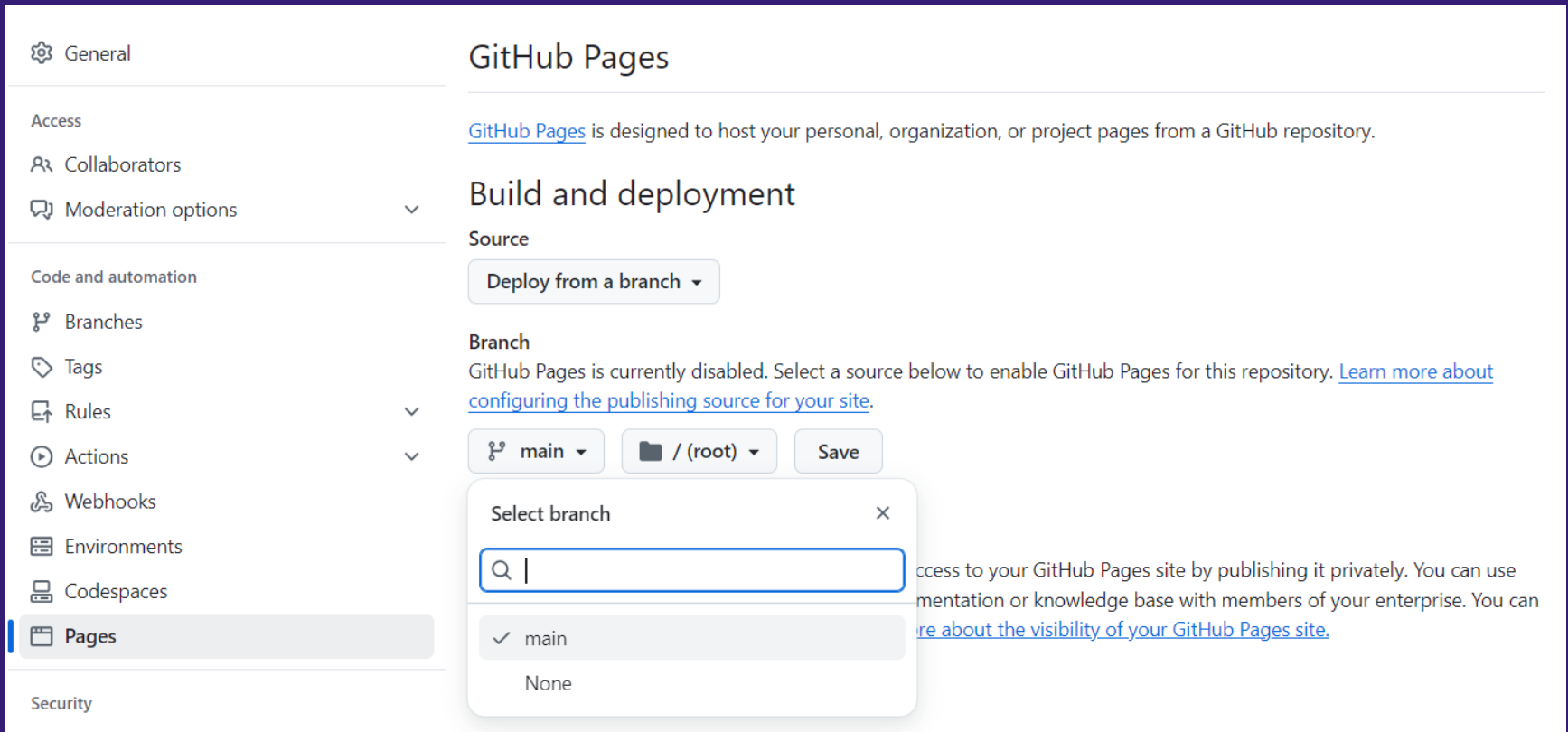


Desplazar hacia abajo hasta encontrar la sección **GitHub Pages**



# Publicar el sitio con GitHub Pages

Abajo de Source, en “**Branch**”, seleccionar “**main**” y luego “**/root**”.  
Y a continuación clic en “**Save**”.

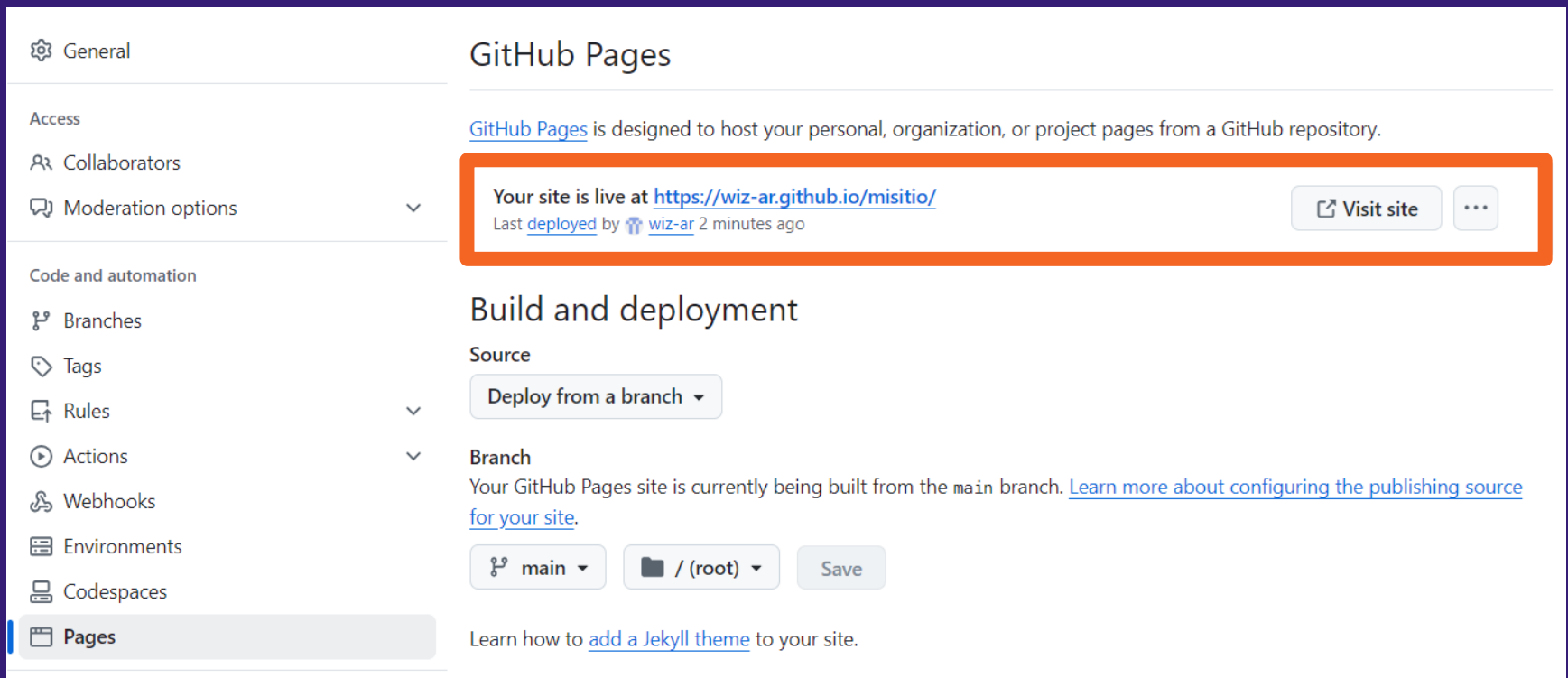


The screenshot shows the GitHub repository settings page for GitHub Pages. On the left, a sidebar contains navigation links: General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, Pages (highlighted), and Security. The main content area is titled 'GitHub Pages' and includes a description: 'GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.' Below this, the 'Build and deployment' section is active, showing 'Source' as 'Deploy from a branch'. Under 'Branch', it states 'GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository.' A dropdown menu for 'main' is open, showing a search bar and a list with 'main' (checked) and 'None'. A 'Save' button is visible next to the branch dropdown. The right side of the page contains text about publishing the site privately, with a link to 'Learn more about configuring the publishing source for your site.'

# Publicar el sitio con GitHub Pages

GitHub genera una **URL para el sitio**.

Esta URL aparece en la sección de **GitHub Pages**, algo como **<https://tu-usuario.github.io/misitio/>**.



The screenshot shows the GitHub repository settings page for GitHub Pages. On the left is a sidebar with navigation links: General, Access, Collaborators, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, and Pages (which is highlighted). The main content area is titled 'GitHub Pages' and contains the following information:

- A description: 'GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.'
- A status box (highlighted with an orange border) stating: 'Your site is live at <https://wiz-ar.github.io/misitio/>'. Below this, it says 'Last deployed by wiz-ar 2 minutes ago'. To the right of the URL are buttons for 'Visit site' and a three-dot menu.
- A section titled 'Build and deployment' with a 'Source' dropdown set to 'Deploy from a branch'.
- A 'Branch' section indicating the site is built from the 'main' branch, with a dropdown set to 'main' and a path dropdown set to '/ (root)'. A 'Save' button is next to these.
- A link at the bottom: 'Learn how to [add a Jekyll theme](#) to your site.'

# Actualizar el sitio

Todo cambio que se haga en los archivos, hay que volver subirlos al repositorio.

Hacer clic en **Add file > Upload files**, subir las versiones actualizadas de los archivos y hacer commit de los cambios.

Así **GitHub** actualiza automáticamente el sitio con las nuevas versiones.

# Problema de Actualización online

## Caché del Navegador:

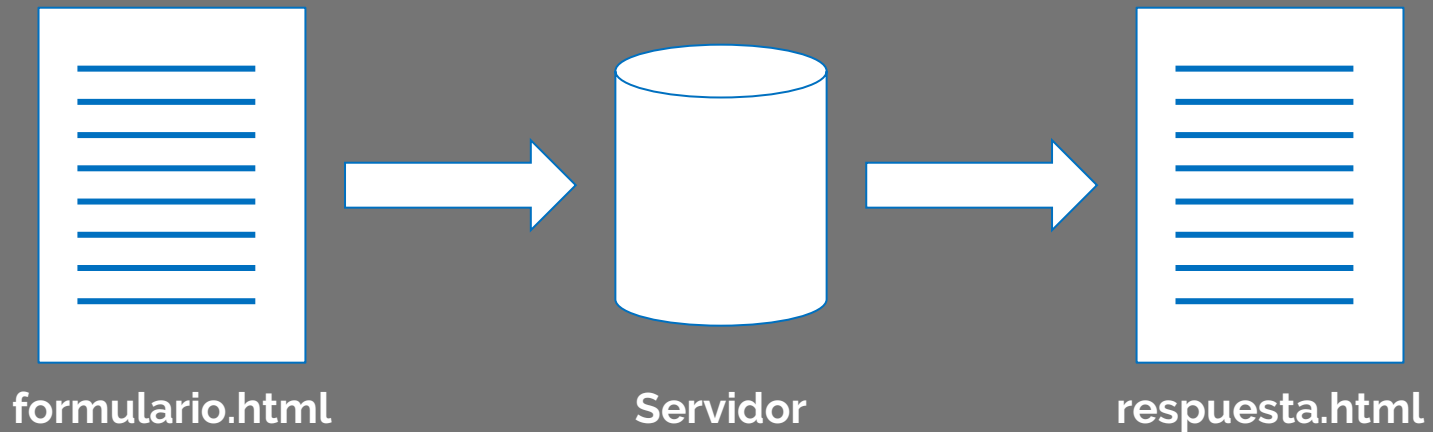
Los navegadores almacenan versiones de las páginas en caché para que carguen más rápido. Hacer una recarga forzada (Ctrl + F5 en Windows o Cmd + Shift + R en Mac). También se puede abrir el sitio en una ventana de navegación privada o incógnito para asegurarte de no estar utilizando la versión en caché.

## Tiempo de espera de GitHub Pages:

GitHub Pages puede tardar unos minutos en actualizar los cambios después de hacer commit, dependiendo del tráfico en los servidores de GitHub. Esperar unos minutos y vuelve a intentar acceder al sitio.

# Formularios

## <form>





---

# Formularios

Un formulario HTML recopila la información ingresada por un usuario. Esta información se envía a un servidor para ser procesada.

El elemento `<form>` es el contenedor para los diferentes tipos de elementos de ingreso de datos:

- campos de texto `<input type="text">`
- casillas de verificación `<input type="checkbox">`
- botones de opción `<input type="radio">`
- menú de selección `<select><option>`
- botones de envío `<input type="submit">`
- botones para click `<input type="button">`
- etc.

# iFrame

Es un marco que se usa para insertar videos de YouTube, mapas de Google, u otros documentos dentro del HTML.

La etiqueta **<iframe>** especifica un marco/caja en línea.

Un marco inline se utiliza para insertar un documento dentro del HTML donde lo escribo.

```
<iframe src="archivo.html"
width="100%" height="300"
name="marco" frameborder="0"
scrolling="no"></iframe>
```