



UNIVERSIDAD TECNOLÓGICA DE HONDURAS

Desarrollo Frontend

Ejemplo Práctico Paso a Paso

Impartido por:

Ing. Carlos Roberto Reina

Máster en Ingeniería de Software y Sistemas Informáticos

HTML5

CSS3 / Flexbox

Bootstrap 5

JavaScript



📁 Estructura del Proyecto

```
mi-proyecto/  
├── index.html      ← Estructura HTML  
├── styles/  
│   ├── style.css  ← Estilos principal  
│   └── base.css   ← Variables globales  
├── js/  
└── app.js         ← Lógica JavaScript
```

💡 Separar HTML, CSS y JS es una **buena práctica fundamental**: mantiene el código organizado y fácil de mantener.

Responsabilidad de cada archivo

H **index.html** — Define la estructura y el contenido de la página

C **style.css** — Controla la apariencia visual: colores, fuentes, layout

J **app.js** — Agrega interactividad: responde a clics y eventos del usuario



index.html — Cabecera del documento

```
<!DOCTYPE html>
<html lang="es">
<head>
  <!-- Codificación UTF-8 -->
  <meta charset="UTF-8">
  <!-- Compatible con IE -->
  <meta http-equiv="X-UA-Compatible"
    content="IE=edge">
```

¿Qué hace cada meta?

- **DOCTYPE html** — Indica al navegador que es HTML5
- **lang="es"** — Idioma del documento (importante para SEO y accesibilidad)
- **charset="UTF-8"** — Soporte de tildes, ñ y caracteres especiales
- **viewport** ★ — *Clave para el diseño responsivo*. Sin esto, el móvil escala el sitio como escritorio



index.html — Cargando Bootstrap e Íconos vía CDN

```
<!-- Bootstrap CSS (CDN) -->
<link
  href="https://cdn.jsdelivr.net/npm/
bootstrap@5.3.0-alpha1/dist/css/
bootstrap.min.css"
  rel="stylesheet"
>
```

¿Por qué CDN?

- ☁ No necesitas descargar ni instalar nada
- ⚡ Los archivos están en servidores rápidos globales
- 🔄 Siempre disponible y actualizado
- 📁 Ahorra espacio en tu proyecto

⚠ Orden importa: Bootstrap primero, luego tu CSS propio. Así puedes *sobreescribir* estilos de Bootstrap con los tuyos.

◆ Bootstrap Icons permite usar `<i class="bi bi-list">` para mostrar íconos SVG sin imágenes.





index.html — Estructura del Header

```
<!-- HEADER -->
<div class="cmp-header">
  <div class="container" < Bootstrap
    <div class="row" < Bootstrap
      <div class="col cmp-header__container">

<!-- 1. LOGO -->
```

Los 3 elementos del Header



Logo

Imagen de marca. Con CSS se controla el tamaño (height: 40px).



Menú de navegación

Lista con enlaces. Tiene **dos clases**: base + --active que JS controla.

☰ Ícono hamburguesa

Solo visible en móvil. Al hacer clic, JS muestra/oculta el menú.





index.html — Menú de navegación y Scripts

Menú de navegación:

```
<ul>
  <li><a href="#">Inicio</a></li>
  <li><a href="#">Desarrollo web</a></li>
  <li><a href="#">Diseño UX/UI</a></li>
  <li><a href="#">Blog</a></li>
  <li><button>Contacto</button></li>
</ul>
```

Scripts al final del body (buena práctica):

```
<!-- Bootstrap JS -->
<script src="https://cdn.jsdelivr.net/npm/
bootstrap@5.3.0-alpha1/dist/js/
bootstrap.bundle.min.js"></script>
```

¿Por qué scripts al final?

- 1 El navegador carga el HTML de **arriba hacia abajo**
- 2 Si el `<script>` está en el `<head>`, bloquea la carga visual
- 3 Al final del `<body>`, el HTML ya existe cuando JS se ejecuta
- 4 Página carga más rápido y JS puede acceder al DOM

El `<button>` en el menú (en lugar de `<a>`) es semánticamente correcto para acciones, no para navegación.





base.css — Variables CSS Globales

El archivo `base.css` (importado en `style.css`) define las variables de color para todo el proyecto:

```
/* base.css */  
  
:root {  
  
  --light-gray-color: #F5F5F5;  
  --yellow-color:     #F5C518;  
  --dark-blue-color: #1A2D4F;  
  --white-color:     #FFFFFF;  
}
```

```
/* Uso en style.css */  
  
.cmp-header {  
  background-color:  
  var(--light-gray-color);  
  
  border-bottom: 8px solid  
  var(--yellow-color);  
}
```

¿Por qué usar variables CSS?

- 🎨 Cambia el color en **un solo lugar** y se actualiza en todo el sitio
- 📄 El código es más **legible**: `--yellow-color` vs `#F5C518`
- 🔄 Permite implementar **temas** (modo oscuro/claro) fácilmente
- ❤️ Todo el equipo usa los **mismos colores**

💡 `:root` es el selector del elemento raíz del documento (equivale a `html`). Las variables definidas aquí son **globales**.



style.css — Header con Flexbox

```
/* Fuente global */
* { font-family: 'Poppins', sans-serif; }

/* HEADER — contenedor externo */
.cmp-header {
  min-height: 98px;
  background-color: var(--light-gray-color);
```

Flexbox aplicado al Header

LOGO Inicio · Web · Blog · **Contacto** ☰

justify-content: space-between

- `display: flex` — Activa Flexbox en el contenedor
- `justify-content: space-between` — Empuja logo a la izquierda, ícono a la derecha
- `align-items: center` — Alinea verticalmente al centro



style.css — Menú de Navegación

```
/* MENÚ – Lista horizontal */  
.cmp-header__container-menu ul {  
  margin: 0;
```

```
  display: flex;
```

Conceptos Clave

- **list-style-type:none** — Elimina los puntos de la lista
- **gap:30px** — Espacio entre items del menú (propiedad Flexbox)
- **text-decoration:none** — Quita el subrayado de los links
- **border-radius:40px** — Hace el botón con bordes totalmente redondeados (píldora)
- **display:none** — El ícono hamburguesa **no se ve** en pantallas grandes

💡 **display:none** oculta el elemento completamente (no ocupa espacio). Es la base del diseño responsivo.



style.css — Media Query & Responsive

```
/* — RESPONSIVE: pantallas ≤ 768px — */
```

```
@media (max-width: 768px) {
```

```
/* Oculta menú en móvil */
```

```
.cmp-header__container-menu {
```

```
  display: none;
```

```
}
```

Desktop vs Móvil



Desktop >768px
Menú visible en fila
Ícono ☰ oculto



Móvil ≤768px
Menú oculto
Ícono ☰ visible

- `calc(100vh - 98px)` — Altura dinámica: toda la ventana menos el header
- `position: absolute` — El menú flota sobre el contenido
- `flex-direction: column` — Cambia los items de fila a columna
- `!important` — Fuerza el estilo sobre la regla `display: none`



⚡ app.js — Menú Hamburguesa con JavaScript

```
// 1. Seleccionar el ícono ☰  
const burgerIcon =  
  document.querySelector(  
    ".cmp-header__container-icon"  
  );
```

¿Cómo funciona paso a paso?

- 1 **querySelector** busca en el DOM el elemento con esa clase CSS
- 2 **addEventListener("click")** escucha cuando el usuario hace clic en el ícono ☰
- 3 **classList.toggle** — Si la clase existe la *quita*; si no existe la *agrega*
- 4 La clase `--active` activa el CSS que muestra el menú a pantalla





Nueva Sección — Banner Hero

Agregamos un **banner principal** con imagen de fondo y texto encima, justo debajo del header:

```
<!-- BANNER HERO -->
<section class="cmp-banner">
  <div class="cmp-banner__overlay">
    <div class="container">
      <h1 class="cmp-banner__title">
        Hola, soy <span>Carlos Roberto</span>
      </h1>
      <p class="cmp-banner_subtitle">
```

CSS del Banner

```
.cmp-banner {
  background-image
    url('images/banner.jpg');
  background-size: cover;
  background-position: center;
  min-height: 500px;
}

.cmp-banner__overlay {
```



NEW

Nueva Sección — Cards de Proyectos

```
<!-- SECCIÓN PROYECTOS -->
<section id="proyectos"
  class="cmp-projects">
  <div class="container">
    <h2>Mis Proyectos</h2>
    <div class="row">

      <!-- CARD 1 -->
```

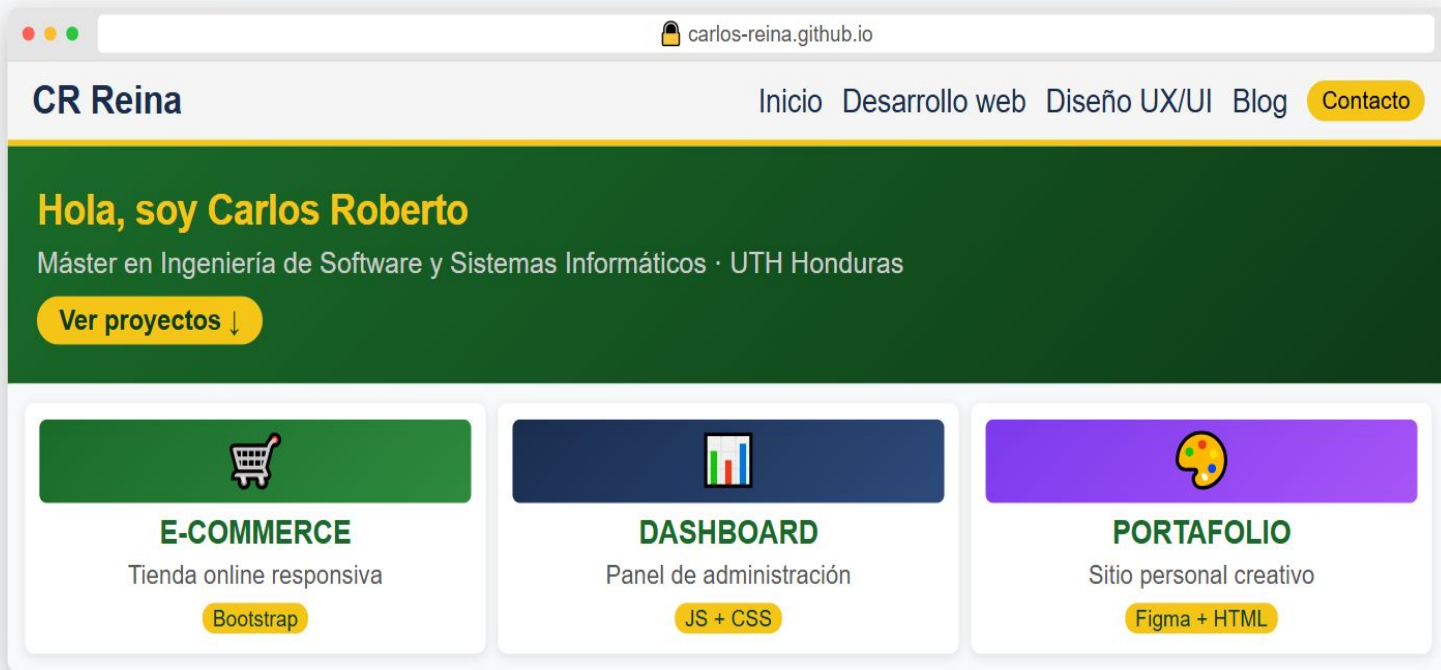
CSS de las Cards

```
.cmp-card {
  border-radius: 12px;
  overflow: hidden;
  box-shadow: 0 4px 16px
    rgba(0,0,0,0.12);
  transition: transform 0.3s;
  margin-bottom: 24px;
}
```

✨ col-md-4 de Bootstrap hace que las 3 cards queden en una fila en desktop y en columna en móvil automáticamente.



Vista Previa — Sitio Web Completo



The screenshot shows a web browser window with the address bar displaying "carlos-reina.github.io". The website header includes the name "CR Reina" and a navigation menu with links for "Inicio", "Desarrollo web", "Diseño UX/UI", "Blog", and "Contacto". The main content area features a dark green background with the text "Hola, soy Carlos Roberto" and "Máster en Ingeniería de Software y Sistemas Informáticos · UTH Honduras". A yellow button labeled "Ver proyectos ↓" is positioned below the text. The footer consists of three white cards with colored headers: "E-COMMERCE" (green header, shopping cart icon, "Tienda online responsiva", "Bootstrap"), "DASHBOARD" (dark blue header, bar chart icon, "Panel de administración", "JS + CSS"), and "PORTAFOLIO" (purple header, palette icon, "Sitio personal creativo", "Figma + HTML").

carlos-reina.github.io

CR Reina Inicio Desarrollo web Diseño UX/UI Blog [Contacto](#)

Hola, soy Carlos Roberto

Máster en Ingeniería de Software y Sistemas Informáticos · UTH Honduras

[Ver proyectos ↓](#)

E-COMMERCE

Tienda online responsiva

[Bootstrap](#)

DASHBOARD

Panel de administración

[JS + CSS](#)

PORTAFOLIO

Sitio personal creativo

[Figma + HTML](#)



Metodología BEM — Nombrado de Clases CSS

Bloque__Elemento--Modificador

Block__Element--Modifier

```
/* BLOQUE — componente independiente */  
.cmp-header { ... }  
  
/* ELEMENTO — parte del bloque */  
.cmp-header__container { ... }  
  
.cmp-header__container-logo { ... }
```

¿Por qué BEM?

- 🔍 Las clases son **auto-descriptivas** — sabes dónde vive cada elemento
- 🛡️ Evita **conflictos de CSS** entre componentes
- 👥 Facilita el trabajo en **equipo**
- ♻️ Los componentes son **reutilizables**
- 🛠️ Fácil de **mantener y escalar**

📌 En el proyecto: cmp = "component". Es un prefijo propio del desarrollador para identificar sus componentes personalizados.





Flujo Completo de Desarrollo

- 1 Diseño en Figma** — Wireframe y mockup del header, banner y cards. Definir colores y tipografías.
- 2 Estructura HTML** (index.html) — Escribir la estructura semántica: header, section banner, section proyectos.
- 3 Variables CSS** (base.css) — Definir colores en `:root` antes de escribir estilos.
- 4 Estilos CSS** (style.css) — Maquetar desktop primero con Flexbox, luego media queries para móvil.
- 5 JavaScript** (app.js) — Agregar interactividad: menú hamburguesa con `classList.toggle`.
- 6 Publicar en GitHub Pages** — Subir al repositorio y activar GitHub Pages. ¡Sitio en línea!





Herramientas Usadas en Este Proyecto



VS Code

Editor + Live
Server



Bootstrap 5

Grid +
Componentes



Bootstrap Icons

Ícono
hamburguesa



Google Fonts

Poppins



CDN jsDelivr

Bootstrap + Icons



DevTools

Inspector +
Flexbox



GitHub Pages

Publicación
gratuita



W3Schools

Referencia
CSS/JS





Resumen del Código del Proyecto

Archivo	Concepto	Técnica usada
<code>index.html</code>	Estructura del Header	HTML semántico + Bootstrap Grid
<code>index.html</code>	Carga de librerías	CDN (Bootstrap, Icons, Fonts)
<code>base.css</code>	Colores globales	Variables CSS <code>:root</code>
<code>style.css</code>	Layout del Header	Flexbox (justify-content, align-items)
<code>style.css</code>	Diseño responsivo	Media Query <code>@media</code> (<code>max-width:768px</code>)
<code>style.css</code>	Banner + Cards nuevas	background-image, overlay, hover
<code>app.js</code>	Menú hamburguesa	querySelector + addEventListener + classList.toggle



Ejercicio Práctico

Tarea para el estudiante:

- 1 Clonar o descargar el proyecto base del profesor
- 2 Abrirlo en VS Code y ejecutar **Live Server**
- 3 Cambiar el color `--yellow-color` en `base.css` por uno propio
- 4 Agregar tu nombre al menú y cambiar el logo
- 5 Agregar la sección Banner con tu propio texto
- 6 Crear 3 cards con tus proyectos reales o ficticios
- 7 Subir a GitHub y publicar con **GitHub Pages**

Criterios de evaluación:

✓ HTML semántico

Uso correcto de etiquetas y estructura del documento

✓ CSS organizado

Variables, BEM y media queries implementados

✓ Diseño responsivo

Menú hamburguesa funciona en móvil correctamente

✓ Publicado en línea

Sitio accesible vía GitHub Pages





¡Lo lograste!

En este ejemplo aprendiste a construir un

Header · Banner · Cards

con HTML, CSS, Flexbox, Bootstrap y JavaScript

HTML5

CSS Variables

Flexbox

Media Queries

Bootstrap 5

BEM

JavaScript DOM

GitHub Pages

Capacitación impartida por

Ing. Carlos Roberto Reina

Máster en Ingeniería de Software y Sistemas Informáticos · UTH 2026

