



## **Coursework: Web Site Development**

**Internet Technology and Security**

**José Palacios Beortegui**

**Bachelor in Applied Computing – Level 4**

**23/04/2026**

# INDICE

|                                                            |    |
|------------------------------------------------------------|----|
| 1. Introducción .....                                      | 3  |
| 2. Planificación y estructura .....                        | 3  |
| 3. Tecnologías utilizadas .....                            | 3  |
| 4. Paso a paso del desarrollo .....                        | 4  |
| Paso1- Variables CSS y estilos base .....                  | 4  |
| Paso2 – Navegación responsive.....                         | 5  |
| Paso3 – Página de inicio (index.html).....                 | 5  |
| Paso4 – Pagina de formación (education.html) .....         | 6  |
| Paso5 – Pagina de proyectos (projects.html) .....          | 6  |
| Paso6 – Pagina de contacto (contact.html) .....            | 6  |
| Paso7 – Sistema multiidioma (ES/EN/DE) .....               | 7  |
| Paso8 – Animaciones de scroll.....                         | 7  |
| Paso9 – Diseño responsive.....                             | 7  |
| Paso10 – Publicación.....                                  | 8  |
| 5. Correcciones realizadas durante el desarrollo.....      | 8  |
| Correccion1 – Script en posición incorrecta .....          | 8  |
| Correccion2 – Imagen no cargaba en producción.....         | 8  |
| Correccion3 – Formulario no enviaba mensajes .....         | 8  |
| Correccion4 – Errores y warnings en el validador W3C ..... | 9  |
| Correccion5 – Selector de idioma no visible en móvil ..... | 9  |
| 6. Cumpliendo de requisitos.....                           | 9  |
| 7. Extras Implementados.....                               | 10 |

## 1. Introducción

Este proyecto consiste en el desarrollo de un sitio web personal tipo CV Interactivo/Portfolio, desarrollando como parte del modulo de Internet Technology and Security. El objetivo es presentar mi perfil profesional de forma atractiva, accesible y funcional.

LINKS:

URL del sitio web: <https://josepalacios.site>

Repositorio GitHub: <https://github.com/jjpp01x/portfolio>

## 2. Planificación y estructura

Antes de escribir ninguna línea de código, se definió la arquitectura del proyecto. El sitio web compuesto por cuatro paginas HTML independientes, una hoja de estilos CSS compartida y dos archivos JavaScript, organizados de la siguiente manera:

|    |                         |                                                      |
|----|-------------------------|------------------------------------------------------|
| 1  | Pagina_Web_CV/          |                                                      |
| 2  | ├─ index.html           | → Página principal: presentación y sobre mí          |
| 3  | ├─ education.html       | → Formación académica y experiencia laboral          |
| 4  | ├─ projects.html        | → Proyectos académicos y personales                  |
| 5  | ├─ contact.html         | → Información de contacto y formulario               |
| 6  | ├─ favicon.svg          | → Icono personalizado de la pestaña del navegador    |
| 7  | ├─ css/                 |                                                      |
| 8  | │ └─ styles.css         | → Hoja de estilos única compartida por las 4 páginas |
| 9  | ├─ js/                  |                                                      |
| 10 | │ └─ translations.js    | → Textos traducidos en español, inglés y alemán      |
| 11 | │ └─ i18n.js            | → Lógica del selector de idioma                      |
| 12 | ├─ img/                 |                                                      |
| 13 | │ └─ jpalacios.JPG      | → Fotografía de perfil                               |
| 14 | └─ docs/                |                                                      |
| 15 | └─ CV Jose Palacios.pdf | → Currículum descargable en PDF                      |

Se tomó la decisión de usar una única hoja de estilos para todas las páginas, evitando duplicar código y garantizando coherencia visual en toda la web. Del mismo modo, la barra de navegación y el pie de pagina se repiten en los cuatro archivos HTML para mantener una experiencia de usuario uniforme.

El diseño se planifico siguiendo un enfoque mobile-first, asegurando desde el principio que la web se vea correctamente en móvil, tablet y escritorio.

## 3. Tecnologías utilizadas

Para el desarrollo de este proyecto se emplearon tecnologías estándar del desarrollo web expuestas por el profesor en el material didáctico.

HTML5 es el lenguaje de marcado utilizado para estructurar el contenido de las cuatro páginas. Se han empleado etiquetas semánticas propias de HTML5 como <nav>, <main>, <section>, <article> y <footer>, que mejoran la accesibilidad y ayudan a los motores de búsqueda a interpretar correctamente el contenido.

CSS3 es la tecnología encargada de todo el diseño visual. Se han utilizado funcionalidades modernas como variables CSS (--color-acento). Flexbox y Grid para los layouts, @media queries para el diseño responsive, animaciones con @keyframes y transiciones con transition.

JavaScript se ha utilizado sin ninguna librería externa para implementar cuatro funcionalidades: el menú hamburguesa en móvil, el sistema de cambio de idioma, las animaciones de entrada al hacer scroll y la validación y envío del formulario de contacto.

Formspree es un servicio externo gratuito que permite recibir los mensajes del formulario de contacto directamente en el email del propietario, sin necesidad de un servidor propio.

Para la publicación del sitio se utilizó GitHub Pages como plataforma de hosting gratuita, conectada al dominio personalizado josepalacios.site registrado en Porkbun.

| Tecnología   | Versión / Plan | Uso                       |
|--------------|----------------|---------------------------|
| HTML5        | Estándar W3C   | Estructura de las páginas |
| CSS3         | Estándar W3C   | Diseño y responsive       |
| JavaScript   | ES6+           | Interactividad            |
| Formspree    | Plan gratuito  | Envío del formulario      |
| GitHub Pages | Plan gratuito  | Hosting                   |
| Porkbun      | Dominio .site  | josepalacios.site         |

## 4. Paso a paso del desarrollo

### Paso1- Variables CSS y estilos base

Lo primero fue establecer la base visual del proyecto en styles.css. Se definieron variables CSS en el selector: root que permiten controlar los colores de toda la web desde un único punto. Si en el futuro se quisiera cambiar el color de acento, bastaría con modificar una sola línea:

```

/* Variables de color – cambia aquí y cambia en toda la web */
:root {
  --color-primario: #0f172a; /* Azul muy oscuro (navbar, footer) */
  --color-acento: #3b82f6; /* Azul brillante (botones, links) */
  --color-fondo: #f8fafc; /* Blanco grisáceo (fondo general) */
  --color-texto: #1e293b; /* Gris oscuro (texto principal) */
  --color-texto-claro: #64748b; /* Gris medio (texto secundario) */
  --color-blanco: #ffffff;
  --fuente-principal: 'Segoe UI', Arial, sans-serif;
  --sombra: 0 2px 10px rgba(0, 0, 0, 0.1);
  --radio-borde: 8px;
}

/* Reset – elimina estilos por defecto del navegador */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

```

También se aplicó un reset CSS con `* {margin: 0; padding: 0; box-sizing: border-box;}` para eliminar los estilos por defecto del navegador y garantizar un comportamiento uniforme en Chrome, Firefox y Safari.

## Paso2 – Navegación responsive

Se creó una barra de navegación fija con `position: sticky` que permanece visible al hacer scroll. Incluye el nombre como enlace al inicio, los cuatro enlaces de navegación, el sector de idioma y un botón de hamburguesa exclusivo para móvil.

El atributo `aria-label` en el botón hamburguesa y `aria-expanded` que se actualiza con Java script son elementos de accesibilidad que permiten a los lectores de pantalla comunicar el estado del menú a personas con discapacidad visual.

```

<!-- Botón menú hamburguesa (solo visible en móvil) -->
<button class="nav-toggle" aria-label="Abrir menú" aria-expanded="false">
  <span></span>
  <span></span>
  <span></span>
</button>

```

En CSS, el botón está oculto en el escritorio con `display: none` y solo se muestra en pantallas menores de 768px mediante una `@media` query. Al pulsarlo, JavaScript añade la clase `abierto` al menú desplegado los enlaces en la columna.

## Paso3 – Página de inicio (index.html)

La página principal se divide en tres secciones. La sección hero es la primera que ve el visitante: fondo degradado oscuro, foto de perfil circular, nombre, título profesional descripción y tres botones de acción. La foto se recorta en círculo con `border-radius:50%` y `object-fit: cover` para que no se deforme independientemente de las dimensiones originales de la imagen.

La sección Sobre mí usa CSS Grid con dos columnas: texto descriptivo a la izquierda y tarjetas de datos rápidos a la derecha. La sección Habilidades presenta cuatro categorías con etiquetas de colores diferenciados – azul para tecnología, verde para idiomas y amarillo para certificaciones – usando border-radius: 20px para el aspecto de píldora

## Paso4 – Pagina de formación (education.html)

Se diseñó un timeline vertical para mostrar la trayectoria en orden cronológico. La línea vertical se genera con el pseudoelemento ::before del contenedor, evitando añadir elementos HTML innecesarios:

```
/* Línea vertical del timeline */
.timeline::before {
  content: '';
  position: absolute;
  left: 0;
  top: 0;
  bottom: 0;
  width: 2px;
  background-color: #e2e8f0;
}
```

Cada entrada del timeline es un <article> semántico con fecha, título del puesto, empresa y lista de responsabilidades.

## Paso5 – Pagina de proyectos (projects.html)

Se creó un grid de cuatro tarjetas en dos columnas. Cada tarjeta tiene un banner superior de color único con gradiente, título, descripción, etiquetas de tecnologías y una etiqueta de estado. Se añadió un efecto hover que eleva visualmente la tarjeta:

```
.proyecto-card {
  background: var(--color-blanco);
  border-radius: var(--radio-borde);
  box-shadow: var(--sombra);
  overflow: hidden;
  display: flex;
  flex-direction: column;
  transition: transform 0.2s, box-shadow 0.2s;
}

.proyecto-card:hover {
  transform: translateY(-4px);
  box-shadow: 0 8px 24px rgba(0, 0, 0, 0.12);
}
```

## Paso6 – Pagina de contacto (contact.html)

La página de contacto combina información de contacto y un formulario funcional. El formulario se conecta al servicio de Formspree mediante el atributo action del <form>. Antes de enviar,

JavaScript valida que ningún campo este vacío y que el email tenga el formato correcto. Si la validación pasa, se realiza una petición fetch que envía los datos sin recargar la página:

```
// Si todo es válido, enviar a Formspree
if (valido) {
  const data = new FormData(formulario);
  fetch(formulario.action, {
    method: 'POST',
    body: data,
    headers: { 'Accept': 'application/json' }
  }).then(response => {
    if (response.ok) {
      formulario.reset();
      document.getElementById('exito-msg').hidden = false;
    }
  });
}
```

## Paso7 – Sistema multiidioma (ES/EN/DE)

Se implementó un sistema de internalización propio sin librerías externas. Cada elemento traducible del HTML lleva un atributo data-i18n con su clave identificadora:

```
<h1 data-i18n="edu.titulo">Formación y Experiencia</h1>
```

El archivo translation.js contiene un objeto con los textos en los tres idiomas. El archivo i18n.js recorre todos los elementos con data-i18n y sustituye su contenido por la traducción correspondiente. El idioma seleccionado se guarda en localStorage para que persista al navegar entre páginas:

```
// Aplica las traducciones a todos los elementos con data-i18n
function aplicarIdioma(idioma) {
  idiomaActual = idioma;
  localStorage.setItem('idioma', idioma); // Lo recuerda al cambiar de página
}
```

## Paso8 – Animaciones de scroll

Se usó la API nativa IntersectionObserver para detectar cuando un elemento entra en el viewport y activar su animación. Los elementos comienzan invisibles y desplazados hacia abajo (opacity: 0; transform: translateY(30px)) y al entrar en pantalla reciben la clase visible que activa la transición. El observador deja de vigilar el elemento tras animarlo para optimizar el rendimiento.

## Paso9 – Diseño responsive

Se utilizaron @media queries para adaptar el diseño de distintos tamaños de pantalla. En pantallas menores de 786px los grids de dos columnas pasan a una sola columna, el hero se reorganiza en vertical con la foto encima del texto, y el menú de navegación se sustituye por el

menú hamburguesa. La foto de perfil se reduce de 220px a 150px de diámetro para no ocupar demasiado espacio en pantallas pequeñas.

## Paso10 – Publicación

El proyecto se subió a un repositorio publico en GitHub y se activo GitHub Pages desde la configuración del repositorio. Para conectar el dominio josepalacios. site se añadieron cuatro registros DNS tipo A apuntando a las IPs de GitHub y un registro CNAME para el subdominio www. Finalmente se activo HTTPS desde la configuración de GitHub Pages para garantizar una conexión segura.

## 5. Correcciones realizadas durante el desarrollo

Durante el proceso de desarrollo surgieron varios problemas que fue necesario identificar, analizar y resolver. A continuación, se documentan las incidencias mas relevantes.

### Corrección1 – Script en posición incorrecta

El bloque `<script>` de la pagina principal estaba colocado entre las etiquetas `</head>` y `<body>`, una posición invalida en HTML5. Esto provoca que el JavaScript se ejecutara antes de que el navegador hubiera cargado los elementos del DOM, por lo que `document.querySelector('.nav.toggle')` devolvía null y el menú hamburguesa no respondía al pulsarlo en móvil.

La solución fue mover todos los scripts al final del `<body>`, justo antes de `</body>`, que es la posición correcta para garantizar que el HTML este completamente cargado antes de que JavaScript intente acceder a los elementos.

### Corrección2 – Imagen no cargaba en producción

La fotografía de perfil se mostraba correctamente en local, pero no al acceder a la web publicada en GitHub Pages. El motivo fue una diferencia entre sistemas operativos: Windows no distingue entre mayúsculas y minúsculas en los nombres de archivo, por lo que `jpalacios.jpg` y `jpalacios.JPG` son equivalentes en local. Sin embargo, el servidor de GitHub Pages funciona sobre Linux, que se distingue entre mayúsculas y minúsculas. El archivo real era `jpalacios.JPG` pero el atributo `src` del HTML usaba `jpalacios.jpg`, lo que provocaba en error 404 en el servidor. Se corrigió actualizando el atributo `src` para que coincidiera exactamente con el nombre real del archivo.

### Corrección3 – Formulario no enviaba mensajes

El formulario de contacto estaba correctamente conectado a Formspree mediante el atributo `action`, pero los mensajes no llegaban. El problema era que el evento `submit` tenía e `preventDefault()` que bloqueaba el comportamiento nativo del formulario, pero el código

posterior solo mostraba un mensaje de éxito en pantalla sin enviar realmente los datos. Se reescribió la lógica sustituyendo el envío directo por una petición fetch que valida los campos primero y, si todo es correcto, envía los datos a Formspree de forma asíncrona sin recargar la página.

## Corrección4 – Errores y warnings en el validador W3C

Al validar el código con el validador oficial del W3C([validator.w3.org](https://validator.w3.org)) se detectaron los siguientes problemas:

- Role=" list" en <ul>: El atributo era redundante porque el elemento <ul> ya tiene ese rol ARIA por defecto. Se eliminó de los cuatro archivos HTML.
- aria-required=" true" en campos con required: Igualmente redundante cuando el atributo nativo required ya comunica la obligatoriedad del campo. Se eliminó de todos los inputs del formulario.
- Jerarquía de encabezados incorrecta en projects.html: los títulos de las tarjetas de proyecto usaban <h3> siendo el siguiente nivel después del <h1> de la cabecera, saltándose el nivel <h2>. Esto es un error de accesibilidad ya que los lectores de pantalla navegan por encabezados en orden jerárquico. Se corrigieron los títulos de las tarjetas cambiándose a <h2>.

## Corrección5 – Selector de idioma no visible en móvil

Los botones ES/EN/DE estaban ocultos en pantallas pequeñas mediante display: none en la @media query de 480px, ya que no cambian en la barra de navegación junto al menú hamburguesa. Para solucionarlo se añadió una copia de los botones dentro del propio menú hamburguesa como un elemento <li> adicional, visible únicamente en móvil. De esta forma el usuario puede cambiar el idioma tanto en escritorio como en dispositivos móviles.

## 6. Cumpliendo de requisitos

El sitio web cumple con todos los requisitos establecidos en el enunciado. Se han desarrollado las cuatro páginas obligatorias: index.html con la presentación y sección "Sobre mí", education.html con la trayectoria académica y laboral en formato timeline, projects.html con cuatro proyectos en tarjetas interactivas, y contact.html con información de contacto y formulario funcional.

Todo el código está desarrollado en HTML5 con etiquetas semánticas (<nav>, <main>, <section>, <article>, <footer>) y CSS con variables, Grid, Flexbox, animaciones y transiciones. Se ha aplicado accesibilidad mediante el atributo Lang="es", alt en imágenes, aria-label en botones y jerarquía correcta de encabezados. El diseño es responsive gracias a @media queries y el atributo meta viewport. El código ha sido validado en el W3C y los errores detectados fueron corregidos tal como se documenta en el punto 5.

## 7.Extras Implementados

Además de los requisitos mínimos, se implementaron las siguientes funcionalidades adicionales para mejorar la experiencia de usuario y demostrar un mayor dominio técnico:

- Selector de idioma trilingüe (ES/EN/DE): Sistema de internacionalización propio sin librerías externas, con persistencia del idioma entre paginas mediante localStorage.
- Animaciones de entrada al hacer scroll: Implementadas con la API nativa IntersectionObserver, los elementos aparecen suavemente al entrar en el viewport.
- Menú Hamburguesa: Menú de navegación adaptado a móvil controlado con JavaScript, con atributo aria-expanded para accesibilidad.
- Botón volver arriba: Aparece automáticamente al bajar 400px y sube al inicio con scroll suave.
- Favicon SVG personalizado: Icono con las iniciales JP en la pestaña del navegador.
- Formulario de contacto funcional: Integrado con Formspree para recibir mensajes reales en el email del propietario.
- CV descargable en PDF: Disponible directamente desde la página de inicio.
- Dominio propio con HTTPS: Sitio publicado en <https://josepalacios.site> con certificado SSL activo mediante GitHub Pages.
- Año del footer actualizado con new Date().getFullYear().

