



**HTML**



**CSS**



# HTML5, CSS3 y ES6 **La Plantilla**

Centro público integrado de  
formación profesional  
**Alan Turing**

**JS**





## Índice

### ¿Qué veremos en esta sección?

BLOQUE 1 – Estructura básica de un documento HTML5

BLOQUE 2 – Cabecera (`<header>`) con texto principal

BLOQUE 3 – Menú de navegación simple (`<nav>` + enlaces)

BLOQUE 4 – Menú horizontal con submenú desplegable

BLOQUE 5 – Contenido principal (`<main>`) y pie de página (`<footer>`)



## BLOQUE 1 – Estructura básica de un documento HTML5

# BLOQUE 1 – Estructura básica de un documento HTML5



## BLOQUE 1 – Estructura básica de un documento HTML5

Objetivo: hoy dejamos lista la base de cualquier web y entendemos para qué sirve cada parte del esqueleto.

1) Creamos la estructura del proyecto en VS Code

En VS Code → File → Open Folder... abrid una carpeta llamada proyecto-plantilla.

Dentro, en el Explorer (barra lateral):

- Clic derecho en la carpeta → New Folder → css
- Clic derecho en css → New File → styles.css
- Clic derecho en la raíz → New File → index.html

Estructura que debéis ver:

proyecto-plantilla/

```
|— index.html
|— css/
    |— styles.css
```



## BLOQUE 1 – Estructura básica de un documento HTML5

2) Pegad este contenido en index.html

Este archivo queda listo para que en los siguientes bloques añadamos header, nav, main y footer.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="La Plantilla en HTML5" />
  <title>JavaScript ES6</title>
  <link rel="icon" type="image/x-icon" href="../img/favicon.ico" />
  <link rel="stylesheet" href="../css/styles.css" />
</head>
<body>
  <!-- Aquí irán header, nav, main y footer en los siguientes bloques -->
</body>
</html>
```





## BLOQUE 1 – Estructura básica de un documento HTML5

### 3) Fijaos en lo importante

- `<!DOCTYPE html>`: activa modo estándar (HTML5).
- `<html lang="es">`: idioma del documento (accesibilidad y SEO).
- `<meta charset="UTF-8">`: tildes y ñ sin problemas.
- `<meta name="viewport" ...>`: la página se adapta al móvil.
- `<meta name="description" ...>`: descripción corta para buscadores.
- `<title>...</title>`: lo que veréis en la pestaña del navegador.
- `<link rel="stylesheet" href="./css/styles.css">`: conecta el CSS (cuidado con la ruta).
- `<body> ... </body>`: aquí irá todo lo visible (lo añadiremos en los siguientes bloques).

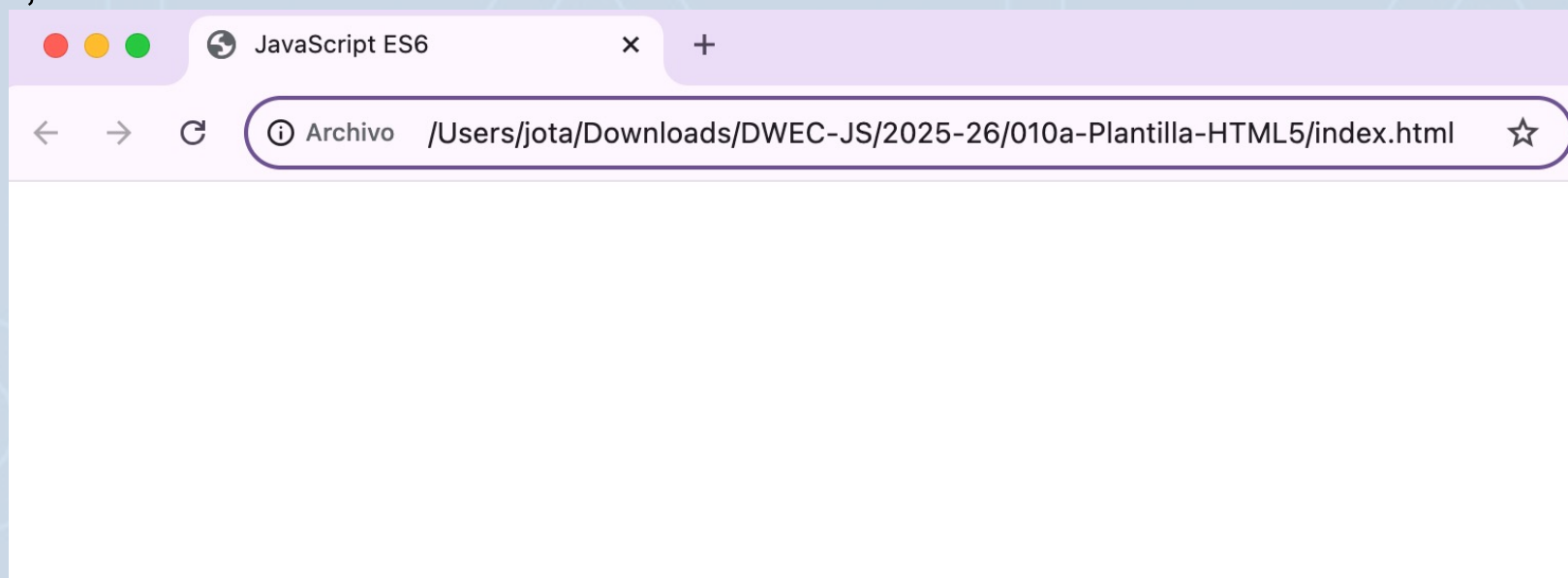


## BLOQUE 1 – Estructura básica de un documento HTML5

### 4) Comprobación rápida

Abrid index.html en el navegador. Debe verse en blanco y la pestaña debe decir “JavaScript ES6”.

Si algo se ve raro con los acentos, revisad UTF-8. Si más adelante no se aplican estilos, revisad la ruta al CSS.





## BLOQUE 2 – Cabecera (<header>) con texto principal

# BLOQUE 2 – Cabecera (<header>) con texto principal





## BLOQUE 2 – Cabecera (<header>) con texto principal

Objetivo: añadimos la franja superior con los títulos del sitio y un hueco para la foto de perfil.

1) Lo que vamos a tocar/crear

- Modificar: index.html
- Modificar: css/styles.css
- Crear: carpeta img/ y dentro un perfil.jpg (cualquier foto de prueba)

Estructura tras este bloque:

proyecto-plantilla/

```
|— index.html
|— css/
|   |— styles.css
|— img/
|   |— perfil.jpg
```



## BLOQUE 2 – Cabecera (<header>) con texto principal

### 2) Añadimos el HTML del header

En index.html, dentro de <body> y arriba del todo, pegad este bloque:

```
<header>
  <div class="header-content">
    <h3>Ejemplos - FUNDAMENTOS DE JS (ES6)</h3>
    <h4>Desarrollo WEB en entorno cliente</h4>
    <h5>Plantilla</h5>
  </div>
  <div class="header-profile">
    
  </div>
</header>
```

Si aún no tenéis la imagen, no quites el <img>; déjalo con un buen alt.  
Mantiene el layout y mejora la accesibilidad, aunque todavía no tengas el archivo real.



## BLOQUE 2 – Cabecera (<header>) con texto principal

### 3) Estilizamos el header con Flexbox

En css/styles.css, al final, añadid:

#### styles.css

```
/* Cabecera principal */
header {
  display: flex;                          /* Alinea en fila */
  justify-content: space-between; /* Texto-izda, foto-dcha */
  align-items: center;
  padding: 5px;
  color: rgba(255, 250, 250, 0.9);
  text-shadow: 2px 2px 4px rgba(0,0,0,0.75);
  background: #2a5aa3; /* Usamos color sólido por ahora */
}

/* Contenedor de los textos */
.header-content {
  flex-grow: 1; /* Ocupa el espacio disponible */
}
```

```
/* Imagen de perfil */
.header-profile img {
  width: 60px;
  height: 60px;
  border-radius: 50%;
  object-fit: cover;
  border: 2px solid rgba(255, 255, 255, 0.7);
}
```

Más adelante, cuando veamos recursos gráficos, cambiaremos el color por una imagen de fondo sin repetirnos.



## BLOQUE 2 – Cabecera (<header>) con texto principal

4) Qué quiero que entendáis aquí

- <header> es semántico: indica cabecera del documento.
- Flexbox nos resuelve la alineación sin hacks.
- alt en la imagen = accesibilidad y buen hábito desde el día 1.
- La ruta relativa ./img/perfil.jpg parte desde index.html.



## BLOQUE 2 – Cabecera (<header>) con texto principal

### 5) Comprobación rápida

Recargad la página: debéis ver la franja superior con los títulos y el avatar redondo a la derecha.

Si el header se apila en columna, falta display:flex o hay un error de llaves en el CSS.

Si no aparece la imagen, comprobad que img/perfil.jpg existe y la ruta es correcta.

**Ejemplos – FUNDAMENTOS DE JS (ES6)**

**Desarrollo WEB en entorno cliente**

**Plantilla**







**BLOQUE 3 – Menú de navegación simple (<nav> + enlaces)**

# **BLOQUE 3 – Menú de navegación simple (<nav> + enlaces)**



## BLOQUE 3 – Menú de navegación simple (<nav> + enlaces)

Objetivo: añadir una barra de navegación horizontal con enlaces útiles. Sin submenús todavía.

1) Qué vamos a tocar/crear

Modificar: index.html

Modificar: css/styles.css

Crear (opcional, para que el enlace funcione): carpeta ejercicios/ con un index.html sencillo dentro

Estructura tras este bloque:

proyecto-plantilla/

- |— index.html
- |— css/
  - |— styles.css
- |— ejercicios/
  - |— index.html (opcional pero recomendable)



## BLOQUE 3 – Menú de navegación simple (<nav> + enlaces)

### 2) Código nuevo a añadir

#### 2.1 HTML (añadir en index.html)

Dónde: inmediatamente debajo de <header>.

```
<nav>
  <ul class="navbar">
    <li><a href="https://www.w3schools.com/js/default.asp">w3schools</a></li>
    <li><a href="https://es.javascript.info/js">javascript.info</a></li>
    <li><a href="https://developer.mozilla.org/es/docs/Web/JavaScript">MDN</a></li>
    <li><a href="./ejercicios/">Ejercicios</a></li>
  </ul>
</nav>
```

El enlace “Ejercicios” apunta a /ejercicios/. Si no creas esa carpeta aún, el enlace dará 404 (no pasa nada mientras tanto).



## BLOQUE 3 – Menú de navegación simple (<nav> + enlaces)

### 2.2 CSS (añadir en css/styles.css)

Dónde: al final del archivo (o en una sección “Navegación”).

#### styles.css

```
/* Navegación simple (sin submenús) */
nav {
  background: #6a3d16; /* color base */
}

.navbar {
  list-style: none;
  margin: 0;
  padding: 0;
  display: flex; /* Menú horizontal */
  flex-wrap: wrap;
  align-items: center;
}
```

```
.navbar li { margin: 0; }

nav a {
  display: block;
  padding: 14px 16px;
  color: #f7f7f7;
  text-decoration: none;
}

nav a:hover {
  background: rgba(255,255,255,0.12);
}
```

Nada de submenús todavía: solo dejamos la barra horizontal y el estado hover.



## BLOQUE 3 – Menú de navegación simple (<nav> + enlaces)

### 3) (Opcional) crear la portada de “Ejercicios”

Para que el menú no lleve a una página vacía:

- Crea la carpeta ejercicios/.
- Copia tu index.html actual a ejercicios/index.html y cámbiale solo el contenido del `<body>` para que muestre un título como “Listado de ejercicios”.  
(No repito el código: duplicas y ajustas los textos; la ruta al CSS desde ahí será `../css/styles.css`.)

### 4) Qué quiero que entendáis aquí

- `<nav>` es semántico: indica navegación principal del sitio (accesibilidad y SEO).
- Los menús en web se construyen con listas (`<ul>` → `<li>` → `<a>`).
- Para que sea horizontal, el truco limpio es `display:flex` en la lista contenedora.
- Usamos rutas relativas: `./ejercicios/` parte desde `index.html`.





## BLOQUE 3 – Menú de navegación simple (<nav> + enlaces)

### 5) Comprobaciones rápidas

1. Al recargar, debe aparecer una barra justo debajo del header con 4 enlaces.
2. El hover debe cambiar el fondo del enlace.
3. “Ejercicios” debería abrir /ejercicios/ si creaste esa página.
4. Si los enlaces se ven en columna, falta display:flex en .navbar o hay un error de llaves } en el CSS.



## BLOQUE 3 – Menú de navegación simple (<nav> + enlaces)

### 6) Resultado esperado

Una barra de navegación horizontal funcional, lista para evolucionar en el próximo paso.





**BLOQUE 4 – Menú horizontal con submenú desplegable**

# **BLOQUE 4 Menú horizontal con submenú desplegable**



## BLOQUE 4 – Menú horizontal con submenú desplegable

Objetivo: convertir el enlace “Ejercicios” del menú en un dropdown que muestre enlaces a ejercicios al pasar el ratón (desktop).

1) Qué vamos a tocar/crear

- Modificar: index.html
- Modificar: css/styles.css
- Crear ejercicios/ejer01/index.html y ejercicios/ejer02/index.html si aún no existen (para que los enlaces funcionen).

Estructura recomendada:

```
proyecto-plantilla/  
├─ index.html  
├─ css/  
│   └─ styles.css  
└─ ejercicios/  
    ├─ ejer01/  
    │   └─ index.html  
    └─ ejer02/  
        └─ index.html
```



## BLOQUE 4 – Menú horizontal con submenú desplegable

### 2) Cambios en HTML (en index.html)

Dónde: dentro del <nav>, sustituye el último <li> (“Ejercicios” simple) por este bloque:

```
<li class="dropdown">
  <a href="#" aria-haspopup="true">Ejercicios</a>
  <ul class="dropdown-content" aria-label="submenu">
    <li><a href="./ejercicios/ejer01/index.html">Ejercicio 01</a></li>
    <li><a href="./ejercicios/ejer02/index.html">Ejercicio 02</a></li>
  </ul>
</li>
```

Importante: no borres el resto del menú. Solo reemplaza la entrada “Ejercicios” que teníamos en el Bloque 3.





## BLOQUE 4 – Menú horizontal con submenú desplegable

### 3) Añadidos en CSS (en css/styles.css)

Dónde: al final de tu hoja o en la sección donde agrupes la navegación.

#### styles.css

```
/* El contenedor de cada item del menú debe poder posicionar su submenú */
.navbar li { position: relative; }

/* Submenú oculto por defecto */
.dropdown-content {
  display: none;
  position: absolute;
  top: 100%;           /* justo debajo del item padre */
  left: 0;
  min-width: 200px;
  background: #ffffff;
  box-shadow: 0 8px 16px rgba(0,0,0,0.2);
  z-index: 10;
  margin: 0;
  padding: 0;
  list-style: none;
}
```

```
/* Mostrar submenú al pasar el ratón o al enfocar con teclado */
.dropdown:hover .dropdown-content,
.dropdown:focus-within .dropdown-content {
  display: block;
}

/* Estilo de los enlaces del submenú */
.dropdown-content li a {
  display: block;
  padding: 12px 16px;
  color: #111;
  text-decoration: none;
}

.dropdown-content li a:hover {
  background: #eee;
}
```

Usamos `:hover` (ratón) y `:focus-within` (teclado) para que sea usable también con tabulador. No repetimos estilos del Bloque 3; esto amplía tu menú.



## BLOQUE 4 – Menú horizontal con submenú desplegable

4) Qué quiero que entendáis aquí

El submenú es una lista anidada (<ul> dentro de <li>).

Para que aparezca “debajo”, el padre necesita position: relative y el hijo position: absolute con top: 100%.  
:hover abre con ratón; :focus-within lo hace accesible con teclado (cuando el enlace Ejercicios recibe foco).

z-index evita que el submenú se oculte detrás de otros elementos.

5) Comprobaciones rápidas

Al pasar el ratón por “Ejercicios”, debe abrirse un panel blanco con dos enlaces.

Con el tabulador, al enfocar “Ejercicios”, también debe mostrarse.

Si el submenú aparece en un lugar raro, revisa que .navbar li { position: relative; } está aplicado.

Si queda tapado por otros elementos, aumenta z-index del submenú (por ejemplo, z-index: 100;).

6) Nota sobre móvil (lo veremos más adelante)

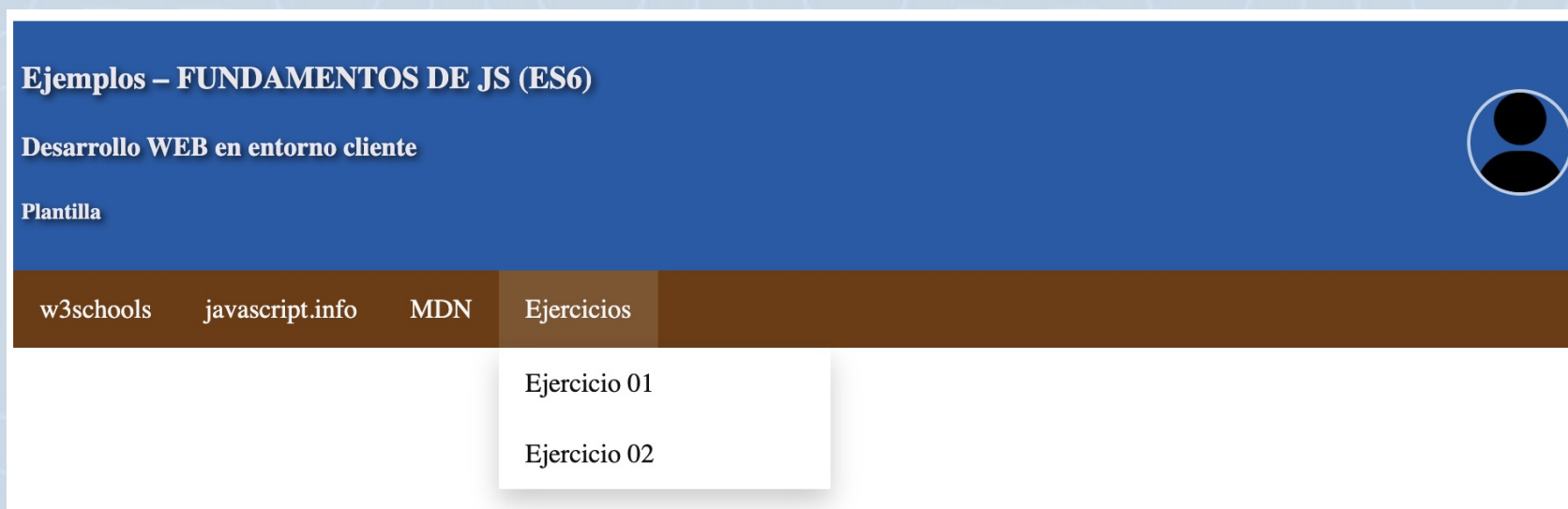
En pantallas táctiles no existe “hover”. De momento, dejamos el enlace padre con href="#" para que no navegue; más adelante añadiremos un pequeño JS para abrir/cerrar el submenú con toque y mejorar la accesibilidad (sin perder la versión CSS pura).



## BLOQUE 4 – Menú horizontal con submenú desplegable

### 7) Resultado esperado

Tu navegación sigue siendo horizontal y ahora “Ejercicios” muestra un desplegable limpio y usable con ratón y teclado.





**BLOQUE 5 – Contenido principal (<main>) y pie de página (<footer>)**

# **BLOQUE 5 – Contenido principal (<main>) y pie de página (<footer>)**



## BLOQUE 5 – Contenido principal (<main>) y pie de página (<footer>)

Objetivo: añadir una sección de bienvenida clara y un pie de página consistente. ✓

1) Qué vamos a tocar/crear

- Modificar: index.html
- Modificar: css/styles.css
- Estructura (no cambia de carpetas en este bloque).





## BLOQUE 5 – Contenido principal (<main>) y pie de página (<footer>)

### 2) Código nuevo a añadir

#### 2.1 HTML (en index.html)

- Debajo de <nav>, añade:

```
<main>
  <section class="intro">
    <h2>Bienvenido</h2>
    <p>Selecciona un ejercicio desde el menú superior para empezar.</p>
  </section>
</main>
```

- Al final del <body> (después de <main>), añade:

```
<footer>
  © Luis José Sánchez & José García ☺
</footer>
```

**Nota pedagógica:** uso <h2> en la sección porque el título global del sitio ya está implícito en la cabecera; así mantenemos la jerarquía de encabezados.



## BLOQUE 5 – Contenido principal (<main>) y pie de página (<footer>)

### 2.2 CSS (en css/styles.css)

Añade al final del archivo:

```
/* Contenido principal */
main {
  display: flex;
  justify-content: center;
  flex-wrap: wrap;
  gap: 16px;
  padding: 20px;
}

/* Pie de página */
footer {
  background: #2a5aa3;          /* mismo color base que el header para coherencia visual */
  color: rgba(255, 250, 250, 0.9);
  text-shadow: 2px 2px 4px rgba(0,0,0,0.75);
  text-align: center;
  padding: 20px;
}
```



## BLOQUE 5 – Contenido principal (<main>) y pie de página (<footer>)

### 3) Qué quiero que entendáis aquí

- <main>: debe haber uno por página y contiene el contenido principal. No se anida dentro de otras secciones.
- Distribución con Flexbox: centramos la sección de bienvenida y dejamos espacio (gap) por si añadimos tarjetas/ejercicios más adelante.
- <footer>: cierra la página con información breve (autoría, licencia, enlaces legales, etc.). Lo hacemos visualmente coherente con el header reutilizando color.
- Jerarquía de títulos: no repetimos un <h1> sin necesidad; usamos <h2> para secciones internas y mantenemos un orden lógico.

### 4) Comprobaciones rápidas

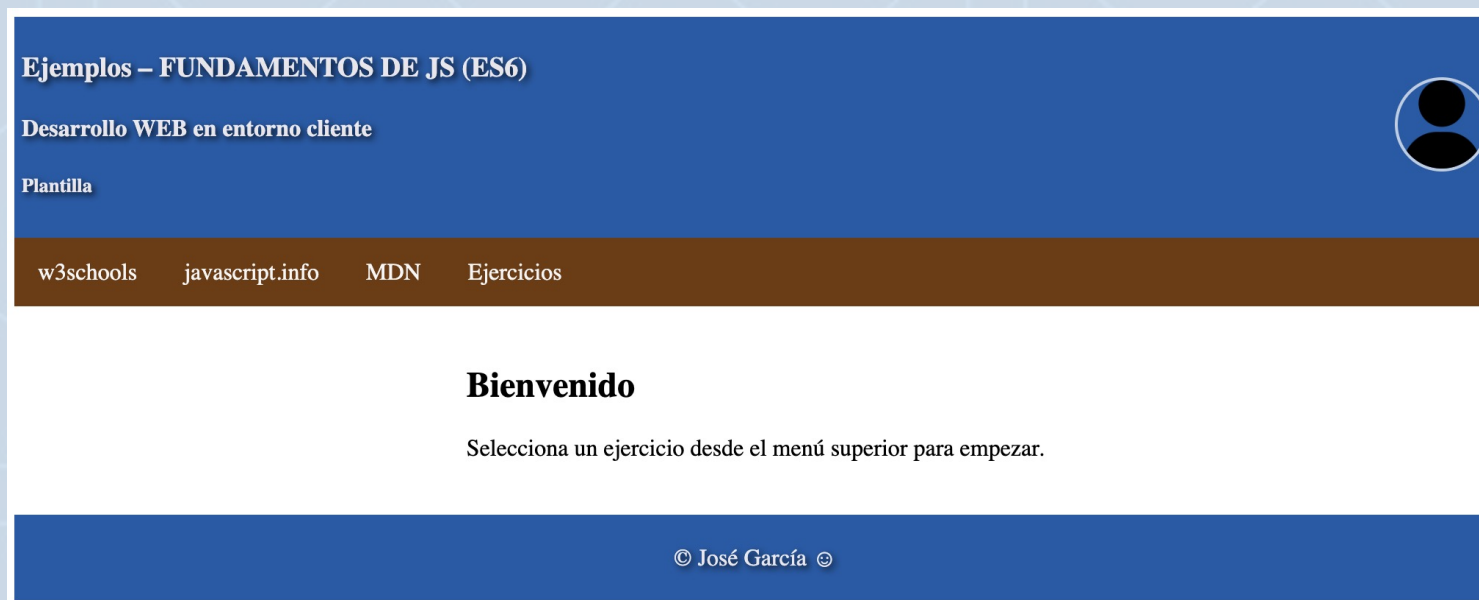
1. Debes ver un título “Bienvenido” centrado con un párrafo debajo.
2. El footer aparece abajo, con el mismo color base que el header y texto legible.
3. Si el contenido no se centra, revisa que main tenga display:flex y justify-content:center.
4. Si el footer no ocupa todo el ancho, revisa que no haya reglas de anchura extrañas en el CSS (por defecto, footer es bloque a 100%).



## BLOQUE 5 – Contenido principal (<main>) y pie de página (<footer>)

### 5) Resultado esperado

Una página con estructura semántica completa: header → nav → main → footer, lista para empezar a alojar nuestros ejercicios y, mejorar la navegación con submenús y estilos globales.





Ejercicios

HTML5 - CSS3 - JS6 – La Plantilla



# Ejercicios





## Ejercicios. Enunciado

### Ejercicio

Diseña tu propia plantilla para la entrega de ejercicios.

