

## Enunciado del reto

Tu tarea es experimentar con el **posicionamiento en CSS**. En 30 minutos deberás crear una página que contenga varias cajas con diferentes propiedades de `position` (static, relative, absolute, fixed, float). Además, comprobarás cómo funcionan las coordenadas (top, left, right, bottom) y cómo afectan al flujo del documento.

## Objetivo(s) de la actividad

- **Aplicar** distintos tipos de posicionamiento en CSS.
- **Usar** coordenadas CSS (top, left, right, bottom) para mover elementos.
- **Comparar** cómo cada posicionamiento afecta el flujo de los elementos.
- **Analizar** los resultados en el navegador usando el inspector.

## Materiales y preparación previa (checklist)

- Editor de texto.
- Navegador web con DevTools.
- Archivos: `index.html` y `styles.css`.

## Requisitos técnicos y datos

- HTML5 semántico.
- CSS externo enlazado.
- Cajas con estilos diferenciados.

## Instrucciones paso a paso con micro-checks

**Tiempo total: 30 minutos**

### 1. Estructura HTML inicial (5')

- a. Crea `index.html`:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
```

```

scale=1.0">
  <title>Ejemplo Posicionamiento CSS</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Ejemplos de Posicionamiento</h1>
  <div class="caja caja--static">Static</div>
  <div class="caja caja--relative">Relative</div>
  <div class="caja caja--absolute">Absolute</div>
  <div class="caja caja--fixed">Fixed</div>
  <div class="caja caja--float">Float</div>
</body>
</html>

```

*Micro-check:* 5 cajas visibles en pantalla.

## **2. Definir estilos base (10')**

a. Crea `styles.css`:

```

body {
  font-family: Arial, sans-serif;
  margin: 20px;
}

.caja {
  width: 120px;
  height: 80px;
  margin: 10px;
  padding: 10px;
  color: white;
  text-align: center;
  line-height: 80px;
}

.caja--static {
  background: navy;
  position: static; /* por defecto */
}

.caja--relative {
  background: teal;
  position: relative;
}

```

```

    top: 20px;
    left: 30px;
}

.caja--absolute {
    background: crimson;
    position: absolute;
    top: 50px;
    right: 20px;
}

.caja--fixed {
    background: orange;
    position: fixed;
    bottom: 10px;
    right: 10px;
}

.caja--float {
    background: green;
    float: right;
}

```

*Micro-check:* cada caja tiene color y comportamiento distinto.

### **3. Exploración en el navegador (10')**

- a. Abre DevTools (Ctrl+Shift+I).
- b. Selecciona cada caja y observa cómo cambian las coordenadas y su referencia (body, padre posicionado o viewport).
- c. Haz scroll y verifica cómo la caja con `fixed` permanece en su lugar.
- d. Observa cómo el texto fluye alrededor de la caja con `float`.

### **4. Comparación final (5')**

- a. Documenta las diferencias clave:
  - i. `static`: sigue el flujo normal.
  - ii. `relative`: se mueve sin afectar el flujo.
  - iii. `absolute`: sale del flujo y depende de un padre posicionado.
  - iv. `fixed`: siempre visible en la pantalla.
  - v. `float`: alinea y permite flujo alrededor.

*Micro-check:* escribe un breve resumen en un archivo de texto.

## Criterios de éxito y errores comunes

### Éxito:

- Cada caja muestra un tipo de posicionamiento diferente.
- Coordenadas (`top`, `left`, etc.) aplicadas y observadas en navegador.
- Se analiza cómo afecta el flujo del documento.
- Uso de inspector para validar.

### Errores comunes:

- No asignar un padre con `relative` para probar `absolute`.
- Confundir `relative` con `absolute`.
- Olvidar `float` y su efecto en elementos vecinos.

**Tiempo total:** 30'

### Plan de contingencia:

- Si `absolute` no funciona como esperado, envolverlo en un `div` padre con `position: relative`.
- Si `fixed` no aparece, revisar que el `bottom` y `right` no estén fuera de viewport.

### Extensión opcional (+10'):

- Añadir un ejemplo de `sticky`.
- Crear un layout básico usando `float` vs. `flexbox`.

# SOLUCIÓN PROPUESTA

## index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Ejemplo Posicionamiento CSS</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Ejemplos de Posicionamiento</h1>
  <div class="caja caja--static">Static</div>
  <div class="caja caja--relative">Relative</div>
  <div class="caja caja--absolute">Absolute</div>
  <div class="caja caja--fixed">Fixed</div>
  <div class="caja caja--float">Float</div>
</body>
</html>
```

## styles.css

```
body {
  font-family: Arial, sans-serif;
  margin: 20px;
}

.caja {
  width: 120px;
  height: 80px;
  margin: 10px;
  padding: 10px;
  color: white;
  text-align: center;
  line-height: 80px;
}
```

```
.caja--static {  
    background: navy;  
    position: static; /* por defecto */  
}
```

```
.caja--relative {  
    background: teal;  
    position: relative;  
    top: 20px;  
    left: 30px;  
}
```

```
.caja--absolute {  
    background: crimson;  
    position: absolute;  
    top: 50px;  
    right: 20px;  
}
```

```
.caja--fixed {  
    background: orange;  
    position: fixed;  
    bottom: 10px;  
    right: 10px;  
}
```

```
.caja--float {  
    background: green;  
    float: right;  
}
```