

**PROGRAMA DE ESTUDIOS**

# **DESARROLLO DE SISTEMAS DE INFORMACIÓN**

**FRONTEND ORIENTADO  
A SERVICIOS**

**Tema**

**DISEÑO AVANZADO CON FLEXBOX  
PARTE II**

## DISEÑO AVANZADO CON FLEXBOX PARTE II

En esta segunda parte, nos enfocaremos en cómo utilizar Flexbox para diseñar estructuras complejas y en cómo aprovechar las propiedades de alineación y distribución para organizar los elementos de manera precisa.

### Uso Avanzado de Flexbox en Estructuras de Diseño Complejas

Para abordar estructuras complejas, Flexbox nos permite ajustar la alineación y la distribución de elementos con propiedades adicionales que ayudan a crear layouts más detallados y específicos. A medida que avanzamos en el uso de Flexbox, comenzamos a ver cómo estos conceptos se combinan para crear estructuras que antes requerían técnicas complicadas.

#### 1. Flex Containers y Flex Items:

- Los **contenedores Flex** y los **elementos flexibles** (flex items) son la base de cualquier diseño con Flexbox. Los contenedores se definen con `display: flex`, y los elementos dentro de estos se comportan de acuerdo con las propiedades de Flexbox.
- En diseños complejos, es común anidar contenedores Flex y combinar sus propiedades para obtener layouts dinámicos.

#### 2. Uso de Propiedades Avanzadas

- `flex-grow`: Define cuánto espacio adicional puede tomar un elemento en el contenedor. Un valor de 1 permite que el elemento se expanda para llenar el espacio restante.

- **flex-shrink:** Controla cuánto puede reducirse un elemento cuando el espacio es limitado. Un valor de 0 evita que el elemento se reduzca.
- **flex-basis:** Especifica el tamaño inicial del elemento antes de distribuir el espacio restante.

Ejemplo de uso:

CSS

```
.flex-container {  
  display: flex;  
  flex-wrap: wrap;  
}  
.flex-item {  
  flex-grow: 1;  
  flex-basis: 100px;  
}
```

### Alineación y Distribución de Elementos en Ejes

Flexbox permite alinear y distribuir los elementos en dos ejes: el **eje principal** (determinado por `flex-direction`) y el **eje transversal**. Los elementos pueden alinearse tanto en el eje principal como en el transversal usando varias propiedades que permiten diseños flexibles y adaptables.

#### 1. Propiedades de Alineación en el Eje Principal (`justify-content`)

- Controla cómo se distribuyen los elementos a lo largo del eje principal.

- **Valores:**

- flex-start: Alinea los elementos al inicio del eje principal.
- flex-end: Alinea los elementos al final del eje principal.
- center: Alinea los elementos en el centro del eje principal.
- space-between y space-around: Distribuyen el espacio entre los elementos.

## 2. Propiedades de Alineación en el Eje Transversal (align-items y align-self)

- align-items: Controla la alineación de todos los elementos en el eje transversal.
- align-self: Permite ajustar la alineación de un solo elemento, ignorando align-items.

- **Valores:**

- flex-start y flex-end: Alinean los elementos en la parte superior o inferior del contenedor, respectivamente.
- center: Alinea los elementos en el centro del eje transversal.
- stretch: Los elementos se estiran para llenar el espacio disponible (valor predeterminado).

## 3. Propiedad align-content para Alineación Múltiple

- align-content afecta la alineación de líneas completas de elementos flexibles cuando hay espacio vertical adicional.

- **Valores:**

- stretch: Las líneas se estiran para ocupar el espacio disponible.
- space-between, space-around, center y flex-start: Distribuyen o alinean las líneas.

**Ejemplo:**

```
CSS

.flex-container {
  display: flex;
  flex-wrap: wrap;
  align-content: space-around;
}

.flex-item {
  width: 30%;
  margin: 10px;
}
```

**Ejercicios Prácticos**

**Ejercicio 1: Crear un Layout de Página Completo**

**Objetivo:** Crear una estructura de página con un encabezado, un contenido principal dividido en dos columnas (una más grande que la otra) y un pie de página.

**Instrucciones:**

Utilizar display: flex para el contenedor principal de la página.

Definir el encabezado y el pie de página ocupando el ancho completo.

En el contenido principal, usar flex-grow para dar más espacio a una columna que a la otra.

**Ejercicio 2: Galería de Imágenes con Flexbox**

**Objetivo:** Crear una galería en la que las imágenes se distribuyan uniformemente en filas, ajustándose al ancho de pantalla con espacios entre ellas.

**Instrucciones:**

Crear un contenedor `.gallery` con imágenes.

Configurar `display: flex;` y `flex-wrap: wrap` en `.gallery`.

Usar `justify-content` y `align-items` para alinear y distribuir las imágenes.





INSTITUTO  
**KHIPU**