

## **PROYECTO: Tienda**

### **Objetivo**

La finalidad de éste proyecto consiste en crear la interfaz gráfica de un ecommerce utilizando utilizando HTML5, CSS3, Grid, FlexBox, animaciones, transiciones, ...

### **Descripción de la práctica**

El proyecto consta de una página para las distintas fases del proyecto: inicio, listado artículos, detalle artículo, ...

#### **1. Aspecto visual**

El aspecto visual del sitio debe ser homogéneo y estar personalizado (logos, colores, fuentes web, iconos,...).

Además debe ser responsive, al menos, para dos tipos de dispositivos: móviles y tablets/grandes/

#### **3. Componentes principales**

##### **Inicio:**

###### **Secciones;**

- **Header:**
  - **Elementos identificación:** logo, login/inicio, añadir carrito
  - **Menú navegación**
- **Sección hero:**
- **Eslogan publicitario:** de la propia página.
- **Imágenes:** Cada imagen hará referencia a una categoría de producto, como “zapatos”, “chaquetas”, “accesorios”, ...
- **Síguenos:** Formada por dos components: imagen o imágenes junto a pequeño texto publicitario e iconos redes sociales
- **Recibe nuestras ofertas:**
- **Footer:** cuatro apartados: Ayuda, Nosotros, Legal, Métodos de pago. Además deben aparecer los iconos de redes sociales

##### **Categorías:**

- **Header**
  - **Elementos identificación:** logo, login/inicio, añadir carrito
  - **Menú navegación**
- **Sección productos:** galería de productos. Cada producto debe mostrar: imagen, descripción y precio
- **Footer**

## 4. Herramientas CSS3 principales

- GRID para el layout general.
- GRID para el layout responsive de la sección servicios/artículos/opinions ...
- FlexBox como herramienta secundaria para el diseño o construcción de cada componente.

## 5. Notas

Puntos de ruptura habituales

- Uso de variables para colores, tamaños y fuentes.
- Inicializar valores
- Uso de, al menos, dos fuentes
- (esto no es obligatorio) Si tienes la idea de usar más de un documento CSS, la distribución será como sigue:
  - Un documento=un componente CSS
  - Además, dispondrás de un documento CSS para inicializar valores, importar fuentes, declarar variables, declarar GRID general, ...
  - Tu documento HTML solo llamará a un documento CSS principal, este documento estará formado por la importación de todos los anteriores en el orden adecuado.

### Ejemplo GRID 12 columnas:

- Primera fila igual al alto de la vista, resto se ajustan al contenido
- Primera y última columna iguales, su ancho es igual a una fracción del espacio restante, resto de columnas se ajustan al contenido con un máximo de 11rem.

```
.container {  
    display: grid;  
    grid-template-columns:  
        [container-start] 1fr [center-start] repeat(  
            10,  
            [col-start] minmax(min-content, 11rem) [col-end]  
        )  
        [center-end] 1fr [container-end];  
    grid-template-rows: 100vh repeat(8, min-content);  
}
```

### Ejemplo GRID 10 columnas:

- Primera y última columna al menos un ancho de 6rem como máximo una fracción del espacio disponible. Resto de columnas se ajustan al contenido mínimo o bien, 6rem o una fracción del espacio disponible
- Filas: todas las filas se ajustan al contenido mínimo

```
.container {  
    display: grid;  
    grid-template-columns: minmax(6rem, 1fr) repeat(8, minmax(min-content,  
16rem)) minmax(6rem, 1fr);  
    grid-template-rows: repeat(4, min-content);  
}
```

### Ejemplo GRID responsivo:

```
.container {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(40rem, 1fr));  
    grid-gap: 2rem;  
}
```

## Diseño

- Diseño responsivo: Técnica mobile first o desktop first
- Menú visible en todas las vistas y páginas
- Fuentes web: al menos hay que usar dos (títulos y contenido)
- Iconografía
- Colores: gama de colores adecuada al tema elegido y que cumpla requisitos de accesibilidad.
- Uso de CSS3:
  - Layout principal GRID. Podemos usar FlexBox en contenedores secundarios
  - Animaciones, transiciones, bordes, sombras.... Que hagan más atractivo el sitio. (sin abusar)
  - Uso de variables: colores, tamaños, fuentes, ....
  - Unidades de medida responsiva: rem, em, vh, vw, %
  - Propiedad calc: adaptar tamaño texto, bordes, relleno, ...
  - Ficheros independientes en el diseño cuando sea necesario.
- Uso de mediaqueries cuando sea necesario. Puntos de ruptura: 576, 768, 992, 1200, 1400

| Breakpoint        | Infijo de clase | Dimensiones |
|-------------------|-----------------|-------------|
| X-Small           | None            | <576px      |
| Small             | sm              | ≥576px      |
| Medium            | md              | ≥768px      |
| Large             | lg              | ≥992px      |
| Extra large       | xl              | ≥1200px     |
| Extra extra large | xxl             | ≥1400px     |

Ejemplo GRID 12 columnas:

```
.container {  
  display: grid;  
  grid-template-columns:  
    [container-start] 1fr [center-start] repeat(  
      10,  
      [col-start] minmax(min-content, 11rem) [col-end]  
    )  
    [center-end] 1fr [container-end];  
  grid-template-rows: 100vh repeat(8, min-content);  
}
```

## Para entregar:

Enlace a GitHub

- Proyecto terminado
- Pequeño documento donde aparecerá:
- Tipografía usada
- Mapa de colores
- Iconografía usada