

# Taller Práctico:

## Diseño Responsivo con Flexbox y Grid

Julian F. Latorre - Bootcamp Desarrollo Web Full Stack



## Introducción

En este taller, aplicaremos los conceptos de diseño responsivo utilizando Flexbox y Grid de CSS. Crearemos una página web simple pero atractiva que se adapte a diferentes tamaños de pantalla.

### Objetivo del Taller

Crear una página web responsiva para una tienda de plantas en línea llamada “Vi-veroVerde” utilizando HTML y CSS, con énfasis en Flexbox y Grid.

## 1. Configuración Inicial

Comienza creando dos archivos: `index.html` y `styles.css`. Utiliza el siguiente código como punto de partida:

## index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ViveroVerde - Tu Tienda de Plantas</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>ViveroVerde</h1>
    <nav>
      <ul>
        <li><a href="#inicio">Inicio</a></li>
        <li><a href="#productos">Productos</a></li>
        <li><a href="#sobre-nosotros">Sobre Nosotros</a></li>
        <li><a href="#contacto">Contacto</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <!-- Contenido principal aqui -->
  </main>

  <footer>
    <p>&copy; 2024 ViveroVerde. Todos los derechos reservados.</p>
  </footer>
</body>
</html>
```

## styles.css

```
* {
  box-sizing: border-box;
  margin: 0;
  padding: 0;
}

body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  color: #333;
}

/* Estilos adicionales aqui */
```

## 2. Tarea 1: Header Responsivo con Flexbox

### Objetivo

Crear un header responsivo que muestre el logo y la navegación en una fila en pantallas grandes, y apile estos elementos en pantallas pequeñas.

### Pasos:

1. Agrega estilos al header utilizando Flexbox.
2. Usa media queries para cambiar la disposición en pantallas pequeñas.

### Código base:

```
header {
  background-color: #4CAF50;
  padding: 1rem;
  /* Añade estilos Flexbox aquí */
}

nav ul {
  list-style-type: none;
  /* Añade estilos Flexbox aquí */
}

nav a {
  color: white;
  text-decoration: none;
  padding: 0.5rem 1rem;
}

@media (max-width: 768px) {
  /* Añade estilos para pantallas pequeñas aquí */
}
```

### Pista

Utiliza `display: flex;` y `flex-direction: column;` para apilar elementos en pantallas pequeñas.

### 3. Tarea 2: Grid de Productos

#### Objetivo

Crear una sección de productos utilizando CSS Grid que se adapte a diferentes tamaños de pantalla.

#### Pasos:

1. Agrega una sección de productos en el HTML.
2. Utiliza CSS Grid para crear una cuadrícula responsiva.
3. Ajusta el número de columnas según el tamaño de la pantalla.

#### Código base HTML:

```
<section id="productos">
  <h2>Nuestras Plantas</h2>
  <div class="productos-grid">
    <div class="producto">
      
      <h3>Planta 1</h3>
      <p>Descripción de la planta 1</p>
    </div>
    <!-- Añade más productos aquí -->
  </div>
</section>
```

#### Código base CSS:

```
.productos-grid {
  /* Añade estilos Grid aquí */
}

.producto {
  border: 1px solid #ddd;
  padding: 1rem;
  text-align: center;
}

@media (min-width: 768px) {
  .productos-grid {
    /* Ajusta el grid para pantallas medianas */
  }
}

@media (min-width: 1024px) {
  .productos-grid {
    /* Ajusta el grid para pantallas grandes */
  }
}
```

Pista

Utiliza `grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));` para crear un grid responsivo que se ajuste automáticamente.

## 4. Tarea 3: Sección "Sobre Nosotros" con Flexbox

### Objetivo

Crear una sección "Sobre Nosotros" que utilice Flexbox para mostrar texto e imagen lado a lado en pantallas grandes y apilados en pantallas pequeñas.

### Pasos:

1. Agrega una sección "Sobre Nosotros" en el HTML.
2. Utiliza Flexbox para disponer el contenido.
3. Usa media queries para ajustar el diseño en diferentes tamaños de pantalla.

### Código base HTML:

```
<section id="sobre-nosotros">
  <h2>Sobre Nosotros</h2>
  <div class="sobre-nosotros-contenido">
    <div class="texto">
      <p>Vivero Verde es tu destino para encontrar las mejores
      plantas y consejos de jardiner a...</p>
    </div>
    <div class="imagen">
      
    </div>
  </div>
</section>
```

### Código base CSS:

```
.sobre-nosotros-contenido {
  /* Añade estilos Flexbox aquí */
}

.texto, .imagen {
  /* Estilos para el texto y la imagen */
}

@media (max-width: 768px) {
  .sobre-nosotros-contenido {
    /* Ajusta el diseño para pantallas pequeñas */
  }
}
```

### Pista

Usa `flex-direction: row;` para pantallas grandes y `flex-direction: column;` para pantallas pequeñas.

## 5. Tarea 4: Formulario de Contacto Responsivo

### Objetivo

Crear un formulario de contacto responsivo utilizando Flexbox y CSS Grid.

### Pasos:

1. Agrega un formulario de contacto en el HTML.
2. Utiliza Flexbox para la disposición general del formulario.
3. Usa CSS Grid para organizar los campos del formulario.

### Código base HTML:

```
<section id="contacto">
  <h2>Cont ctanos</h2>
  <form class="formulario-contacto">
    <div class="form-grupo">
      <label for="nombre">Nombre:</label>
      <input type="text" id="nombre" name="nombre" required>
    </div>
    <div class="form-grupo">
      <label for="email">Email:</label>
      <input type="email" id="email" name="email" required>
    </div>
    <div class="form-grupo">
      <label for="mensaje">Mensaje:</label>
      <textarea id="mensaje" name="mensaje" required></textarea>
    </div>
    <button type="submit">Enviar</button>
  </form>
</section>
```

### Código base CSS:

```
.formulario-contacto {
  /* Añade estilos Flexbox aquí */
}

.form-grupo {
  /* Añade estilos Grid aquí */
}

@media (max-width: 768px) {
  .form-grupo {
    /* Ajusta el diseño para pantallas pequeñas */
  }
}
```

Pista

Utiliza `display: grid;` y `grid-template-columns: 1fr 2fr;` para alinear las etiquetas y los campos de entrada.



## 6. Tarea Final: Integración y Ajustes

### Objetivo

Integrar todas las secciones y realizar ajustes finales para asegurar un diseño coherente y totalmente responsivo.

### Pasos:

1. Revisa la disposición general de la página.
2. Ajusta los márgenes, padding y espaciado entre secciones.
3. Asegúrate de que la transición entre diferentes tamaños de pantalla sea fluida.
4. Prueba la página en diferentes dispositivos o utilizando las herramientas de desarrollador del navegador.

### Consideraciones finales:

- Utiliza unidades relativas (em, rem, %) para tamaños y espaciados.
- Asegúrate de que todos los elementos sean accesibles en todos los tamaños de pantalla.
- Considera añadir transiciones suaves para mejorar la experiencia del usuario.

### Consejo Final

Recuerda que el diseño responsivo es un proceso iterativo. No temas hacer ajustes y mejoras continuas para lograr la mejor experiencia de usuario posible.

## Conclusión

Recuerda que la práctica constante es clave para dominar estas habilidades. Continúa experimentando con diferentes layouts y técnicas para mejorar tus habilidades en diseño web responsivo.

 ¡Buen trabajo!