

Guía práctica de CSS para Portafolio Personal

Esta guía explica paso a paso los estilos del portafolio y, para cada bloque de código, qué hace **cada propiedad** y **qué se observa visualmente** al aplicarla. Está pensada para que la uses como referencia mientras construyes y explicas el diseño.

1. Estilos globales

Objetivo: establecer una base consistente y predecible para todo el documento.

```
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
  font-family: Arial, sans-serif;  
}
```

Propiedades y efecto visual:

- `*` (selector universal): selecciona todos los elementos del documento.
- `margin: 0`: elimina los márgenes predeterminados del navegador; visualmente hace que los elementos no tengan separación externa por defecto.
- `padding: 0`: elimina el relleno interno por defecto; visualmente evita espacios inesperados dentro de cajas.
- `box-sizing: border-box`: hace que el ancho y alto de un elemento incluyan `padding` y `border`. Visualmente evita que el tamaño de la caja crezca cuando agregas padding (facilita el cálculo de layouts).
- `font-family: Arial, sans-serif`: define la tipografía base; cambia la apariencia del texto inmediatamente a la fuente especificada.

Práctica: aplicar este bloque primero y verás que desaparecen saltos y espacios inesperados en el layout.

2. Body

Objetivo: definir el fondo y parámetros de legibilidad generales.

```
body {  
  background-color: #f5f5f5;  
  color: #333;  
  line-height: 1.6;  
  padding: 20px;  
}
```

Propiedades y efecto visual:

- `background-color`: pinta el fondo de la página; visual: un fondo claro que hace resaltar las secciones blancas.
- `color`: color del texto por defecto; visual: define el contraste y legibilidad.
- `line-height`: controla el espaciado vertical entre líneas; visual: mejora la lectura al separar líneas.
- `padding`: agrega espacio interno alrededor del contenido principal; visual: evita que el contenido quede "pegado" a los bordes de la ventana.

3. Header

Objetivo: centrar y dar estilo al encabezado principal.

```
header {  
  text-align: center;  
  margin-bottom: 20px;  
}  
header h1 { color: #2c3e50; }  
header p { font-size: 1.1em; }
```

Propiedades y efecto visual:

- `text-align: center`: centra el texto horizontalmente dentro del contenedor; visual: título y subtítulo centrados.
- `margin-bottom`: crea separación debajo del header; visual: separa el header del resto del contenido.
- `color` en `h1`: cambia el color del título; visual: hace que destaque.
- `font-size` en `p`: aumenta ligeramente el tamaño del subtítulo; visual: mejora jerarquía y legibilidad.

4. Separadores (hr)

Objetivo: dividir visualmente las secciones.

```
hr {  
  margin: 30px 0;  
  border: 0;  
  border-top: 2px solid #ccc;  
}
```

Propiedades y efecto visual:

- `margin: 30px 0`: separa la línea de contenido arriba y abajo.

- `border: 0` y `border-top`: eliminan la línea por defecto y definen una raya controlada; visual: línea horizontal limpia y consistente.

5. Secciones (`section`)

Objetivo: convertir cada `section` en una tarjeta visual.

```
section {  
  background-color: #fff;  
  padding: 20px;  
  border-radius: 8px;  
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);  
  margin-bottom: 20px;  
}  
section h2 { color: #2980b9; margin-bottom: 15px; }
```

Propiedades y efecto visual:

- `background-color: #fff`: fondo blanco que contrasta con el body; visual: cada sección parece una tarjeta flotante.
- `padding`: espacio interno que evita que el contenido toque los bordes de la tarjeta.
- `border-radius`: redondea esquinas; visual: apariencia más moderna y suave.
- `box-shadow`: sombra alrededor de la tarjeta; visual: sensación de profundidad (la tarjeta "flota" sobre el fondo).
- `margin-bottom`: separación entre tarjetas.
- `color` y `margin-bottom` en `h2`: color de título y espacio por debajo para diferenciar el encabezado del contenido.

6. Imágenes y tablas

Objetivo: estilizar avatar y mostrar datos ordenados.

```
img {  
  display: block;  
  margin-top: 10px;  
  border-radius: 50%;  
  width: 150px;  
  height: 150px;  
}  
  
table {  
  width: 100%;  
  border-collapse: collapse;  
}  
  
th, td {
```

```

border: 1px solid #ccc;
padding: 10px;
text-align: left;
}

th {
background-color: #2980b9;
color: #fff;
}

```

Propiedades y efecto visual:

- `display: block` en `img`: quita el comportamiento inline y evita espacios debajo de la imagen.
- `border-radius: 50%`: convierte la imagen en circular; visual: avatar redondeado.
- `width` y `height`: controlan tamaño fijo; visual: consistencia.
- `border-collapse: collapse` en `table`: une los bordes de celdas; visual: tabla más compacta y profesional.
- `th/td border` y `padding`: delimitan celdas y dan respiración al contenido.
- `th background-color` y `color`: encabezado destacado para jerarquía visual.

7. Listas

Objetivo: controlar viñetas y sangría.

```
ul { list-style-type: disc; padding-left: 20px; }
```

Propiedades y efecto visual:

- `list-style-type`: define el tipo de viñeta (disc, circle, none, etc.); visual: estilo de listas.
- `padding-left`: crea sangría para alinear viñetas y texto.

8. Flexbox (ej. sección "Sobre mí")

Objetivo: alinear imagen y texto horizontalmente o verticalmente.

```

#about-me {
display: flex;
justify-content: center;
align-items: center;
flex-direction: column;
text-align: center;
gap: 10px;
}

```

Propiedades y efecto visual:

- `display: flex`: activa Flexbox; visual: los hijos se colocan en una fila por defecto (o columna si se cambia la dirección).
- `justify-content: center`: centra los elementos a lo largo del eje principal (horizontal si `flex-direction: row`); visual: elementos centrados horizontalmente.
- `align-items: center`: centra en el eje cruzado (vertical si `flex-direction: row`); visual: elementos alineados verticalmente.
- `flex-direction: column`: apila elementos verticalmente; visual: imagen encima, texto debajo.
- `gap`: espacio entre elementos; visual: separación consistente.

Notas visuales: Para centrar texto dentro de un elemento hijo (p. ej. `p`), también puedes usar `text-align: center`.

9. Grid (Formulario de contacto)

Objetivo: diseñar un formulario con columnas para nombre/correo y un textarea ocupando toda la fila.

```
.contact-form {  
  display: grid;  
  grid-template-columns: 1fr 1fr;  
  gap: 15px;  
}  
  
.form-group { display: flex; flex-direction: column; }  
.form-group.nombre { grid-column: 1; }  
.form-group.correo { grid-column: 2; }  
.form-group.mensaje { grid-column: 1 / -1; }  
.form-group.button-group { grid-column: 1 / -1; }
```

Propiedades y efecto visual:

- `display: grid`: activa Grid; visual: permite dividir el área en filas y columnas.
- `grid-template-columns: 1fr 1fr`: crea dos columnas de igual tamaño; visual: layout de dos columnas.
- `gap`: espacio entre filas y columnas para respiración visual.
- `.form-group { display: flex; flex-direction: column; }`: cada par label+input se comporta como columna, garantizando que el label quede encima del input.
- `grid-column: 1 / -1`: hace que el elemento ocupe desde la columna 1 hasta la última; visual: textarea y botón ocupan toda la fila debajo de las dos columnas.

Efecto final: en pantallas grandes, nombre y correo quedan lado a lado; el textarea y el botón están en filas completas debajo.

10. Inputs, Textarea y Botón

Objetivo: mejorar la usabilidad y apariencia de los campos de formulario.

```
input, textarea {
  padding: 10px;
  border: 1px solid #ccc;
  border-radius: 5px;
  width: 100%;
}

button {
  padding: 10px;
  background-color: #2980b9;
  color: white;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}

button:hover { background-color: #3498db; }
```

Propiedades y efecto visual:

- `padding` en inputs: aumenta área clicable y espacio interno; visual: campos más cómodos.
- `border` y `border-radius`: borde definido y esquinas suaves; visual: inputs con apariencia moderna.
- `width: 100%`: hace que el input ocupe todo el ancho de su celda en Grid.
- `button background-color` y `color`: botón con contraste que destaca; visual: botón visible y primario.
- `cursor: pointer`: cambia el cursor a mano indicando interactividad.
- `:hover`: efecto visual al pasar el mouse (feedback inmediato).

11. Responsive (media queries)

Objetivo: adaptar el layout a pantallas pequeñas.

```
@media (max-width: 600px) {
  .contact-form { grid-template-columns: 1fr; }
}
```

Propiedades y efecto visual:

- `@media (max-width: 600px)`: reglas que se aplican cuando la pantalla es igual o menor a 600px.
- `grid-template-columns: 1fr` dentro de la media: convierte el layout de dos columnas en una columna apilada; visual: campos apilados verticalmente, mejor lectura en móviles.

12. Footer

```
footer {  
  text-align: center;  
  padding: 10px;  
  color: #777;  
  font-size: 0.9em;  
}
```

Propiedades y efecto visual:

- `text-align: center`: centra el texto del pie de página.
- `padding`: espacio interno para que no quede pegado.
- `color` y `font-size`: tipografía más discreta para no competir con el contenido principal.

Sección: explicación rápida por propiedad (resumen)

A continuación un resumen por propiedad común que usarás frecuentemente y su efecto visual directo:

- `margin`: espacio exterior de la caja. Visual: separa un elemento de otros.
- `padding`: espacio interior. Visual: crea respiración dentro de la caja.
- `border`: línea alrededor de la caja. Visual: delimita visualmente.
- `border-radius`: curva las esquinas. Visual: suaviza cajas y botones.
- `box-shadow`: sombra externa. Visual: profundidad, separación del fondo.
- `background-color`: color de fondo. Visual: contraste y jerarquía.
- `color`: color del texto. Visual: legibilidad y contraste.
- `display` (`block`, `inline`, `flex`, `grid`): determina el modelo de caja. Visual: cómo se ordenan y fluyen los elementos.
- `justify-content` (Flex/Grid): alinea elementos a lo largo del eje principal. Visual: centrado o distribución horizontal.
- `align-items` (Flex): alinea en eje cruzado. Visual: alineación vertical de elementos en fila.
- `grid-template-columns`: define columnas del grid. Visual: estructura de columnas.
- `gap`: espacio entre filas/columnas. Visual: orden y respiración.
- `width` / `height`: tamaño del elemento. Visual: control del tamaño y proporción.
- `box-sizing`: cambia cómo se calcula ancho/alto. Visual: evita que el padding aumente el tamaño total.
- `text-align`: alinea texto interno. Visual: texto centrado/izquierda/derecha.
- `cursor`: cambia el cursor del ratón. Visual: indica interactividad.

Consejos

1. **Aplica estilos en este orden:** globales → body → header → sections → componentes (Flex/Grid) → inputs → responsive.
2. **Muestra cambios visuales inmediatamente:** modifica una propiedad, guarda y recarga; pide a los estudiantes que observen.

3. **Usa herramientas del navegador:** inspeccionar elemento para ver caja, padding, margin, y experimentar en vivo.
 4. **Haz ejercicios pequeños:** por ejemplo, pedir que cambien `border-radius` o `box-shadow` para ver efectos.
-