



TIPOGRAFÍAS

Otra de las cosas esenciales en una aplicación es el tipo de letra. A continuación, una lista de tips:

- Número **mínimo** de fuentes (No usar muchos distintos tipos de letras)
- Fuente **estándar** por el soporte. Utilizar principalmente estas
- **Limitar** cantidad de texto
- Legibles en **diferentes tamaños** (Tamaños de pantallas, dispositivos)
- Altos nivel de **entrelineado** (mantener líneas separadas, saltos de línea, etc.)
- **Contraste** (suficiente contraste del texto con el fondo). Por ejemplo, blanco y amarillo
-> se confunde
- Evitar usar **animaciones intermitentes**

Combinación de fuentes/ tipografías | Psicología de tipografía

- **Serif:** Es una fuente tradicional, sofisticada, confiable, práctica y formal. Se puede usar en logos, textos de párrafos, títulos o impresos. Nook Antigua. Courier. Garamond.
- **Sans-Serif:** Es una fuente moderna, limpia, humanista, geométrica y universal. Se puede usar en logos, textos de párrafos, títulos y títulos pequeños.
- **Script (Cursiva):** Es una fuente elegante, clásica, formal, sofisticada y estilizada. Se puede usar en logos, títulos e invitaciones y para textos cortos. Termina en puntita
- **Moderna:** Tendencia, inteligente, estilo, futurista, tecnológica. Apropia para aportar modernidad.
- **Decorativa:** Divertida, casual, única, exclusiva. Apropia para aportar personalidad.

Layout y grillas.

Grilla: Es un sistema de columnas para los diferentes tamaños de dispositivo. Es un sistema de columnas creado por nosotros de acuerdo a las necesidades para ajustar nuestros elementos y componentes. Por ejemplo, podemos usar un sistema de 4 columnas para dispositivos móviles, y que al extenderse a dispositivos de escritorio, sean de 12 columnas.

Los **mixin**, es una clase que encontramos en **css** que nos ayuda a manejar los **breakpoints** (tipos

de pantalla). Entonces, nos sirven para manejar mejor los distintos breakpoints/tipos de pantalla

Para las grillas, los configuramos seleccionando la clase :root y configurando --columns.

Display: grid: configuramos la disposición de pantalla en formato de grilla

Grid-column-gap: distancia entre las grillas

Grid-template-column: nos permite tener la estructura.

Grid-column: 1/5 -> 1 es la columna donde empieza, y el 5 es la columna donde termina.

```
1 // Breakpoint variables.
2 $xs: 360px;
3 $s: 440px;
4 $m: 768px;
5 $l: 1280px;
6 $xl: 1440px;
7
8 // Mixin para manejar distintos breakpoints.
9 @mixin for-size($size) {
10   @if $size == xs {
11     @media (max-width: $s) { @content; }
12   } @else if $size == m {
13     @media (min-width: $m) { @content; }
14   } @else if $size == l {
15     @media (min-width: $l) { @content; }
16   } @else if $size == xl {
17     @media (min-width: $xl) { @content; }
18   }
19 }
```

```
// Configuración de las columnas.
:root {
  --columns: 4;
  --column-gap: 6.67%;

  @include for-size(m) {
    --columns: 12;
    --column-gap: 2.27%;
  }

  @include for-size(l) {
    --columns: 12;
    --column-gap: 2.19%;
  }
}
```

```
// Configuración de la grilla.
.grid {
  margin: 0 16px;

  @include for-size(m) {
    margin: 0 32px;
  }

  @include for-size(l) {
    margin: 0 80px;
  }

  @include for-size(xl) {
    margin: 0 auto;
    max-width: 1280px;
  }
}
```

```
// Configuración de la grilla y la subgrilla.  
.grid,  
.subgrid {  
  display: grid;  
  grid-column-gap: var(--column-gap);  
  grid-template-columns: repeat(var(--columns), minmax(0, 1fr));  
  position: relative;  
}
```

STYLE GUIDE

Es una página de muestra donde tenemos nuestros componentes y tienen su propia hoja de CSS