

Enunciado del reto

Tu equipo decidió usar **Sass** para mejorar la modularización de estilos en un proyecto web. En 30 minutos deberás crear un sistema de estilos que incluya:

- Variables
- Anidación (&)
- Parciales (_variables.scss, _buttons.scss, _header.scss)
- Mixins
- Uso de @extend y @use

Deberás organizar el proyecto en carpetas, compilar con Sass y verificar que los estilos se apliquen correctamente en el navegador.

Objetivo(s) de la actividad

- **Definir** variables en Sass para colores, tipografías y tamaños.
- **Aplicar** anidación y el selector padre &.
- **Organizar** el código en parciales reutilizables.
- **Implementar** mixins y @extend para optimizar código.
- **Compilar** un CSS final limpio y escalable.

Materiales y preparación previa (checklist)

- Tener instalado Node.js y Sass (`npm install -g sass`).
- Editor de código.
- Carpeta de proyecto con estructura base.

Requisitos técnicos y datos

- Archivos .scss dentro de /scss.
- Archivos parciales enlazados en main.scss.
- CSS compilado en /css/main.css.

Instrucciones paso a paso con micro-checks

Tiempo total: 30 minutos

1. Estructura de proyecto (5')

- a. Crea carpetas y archivos:

```
proyecto-sass/  
├── index.html  
├── scss/  
│   ├── _variables.scss  
│   ├── _buttons.scss  
│   ├── _header.scss  
│   └── main.scss  
└── css/  
    └── main.css
```

Micro-check: estructura lista en el editor.

2. Definir variables y estilos base (7')

- a. En `_variables.scss`:

```
$color-primario: #007bff;  
$color-secundario: #6c757d;  
$color-peligro: #dc3545;  
$fuente-base: Arial, sans-serif;  
$espaciado: 10px;
```

- b. En `main.scss`:

```
@use "variables";  
@use "buttons";  
@use "header";  
  
body {  
    font-family: variables.$fuente-base;  
    margin: 20px;  
}
```

Micro-check: compilar y verificar fuente aplicada.

3. Mixin y anidación (10')

- a. En `_buttons.scss`:

```

@use "variables";

@mixin boton($bg, $color) {
  padding: variables.$espaciado;
  border: none;
  border-radius: 5px;
  background-color: $bg;
  color: $color;
  cursor: pointer;

  &:hover {
    opacity: 0.8;
  }
}

.btn {
  @include boton(variables.$color-secundario, white);
}

.btn-primary {
  @extend .btn;
  background-color: variables.$color-primario;
}

.btn-danger {
  @extend .btn;
  background-color: variables.$color-peligro;
}

```

Micro-check: botones muestran hover y heredan estilos.

4. Estilos con parciales y extend (8')

a. En `_header.scss`:

```

@use "variables";

.header {
  background-color: variables.$color-primario;
  padding: 15px;
  text-align: center;

  &__title {

```

```
    color: white;
    font-size: 1.5rem;
  }
}
```

Micro-check: encabezado aparece con fondo azul y texto blanco.

5. HTML de prueba (compilación y validación) (5')

a. Crea index.html:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Proyecto Sass</title>
  <link rel="stylesheet" href="css/main.css">
</head>
<body>
  <header class="header">
    <h1 class="header__title">Proyecto con Sass</h1>
  </header>
  <section>
    <button class="btn">Secundario</button>
    <button class="btn-primary">Primario</button>
    <button class="btn-danger">Peligro</button>
  </section>
</body>
</html>
```

b. Compila con:

```
sass scss/main.scss css/main.css
```

Micro-check: botones y header estilizados en navegador.

Criterios de éxito y errores comunes

Éxito:

- Proyecto organizado en parciales.
- Variables, mixins, anidación y extend aplicados.
- HTML muestra encabezado y botones correctamente estilizados.
- CSS final compilado limpio.

Errores comunes:

- No compilar antes de abrir en navegador.
- Usar `@import` en lugar de `@use`.
- Confundir `&` en anidación.

Tiempo total: 30'

Plan de contingencia:

- Si `@use` no funciona, probar con `@import`.
- Si Sass no compila, usar `sass --watch scss/main.scss css/main.css`.

Extensión opcional (+10'):

- Crear un mixin para tarjetas (card) con colores personalizables.
- Implementar modo oscuro con variables alternas.

SOLUCIÓN PROPUESTA

index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Proyecto Sass</title>
  <link rel="stylesheet" href="css/main.css">
</head>
<body>
  <header class="header">
    <h1 class="header__title">Proyecto con Sass</h1>
  </header>
```

```
<section>
  <button class="btn">Secundario</button>
  <button class="btn-primary">Primario</button>
  <button class="btn-danger">Peligro</button>
</section>
</body>
</html>
```

scss/_variables.scss

```
$color-primario: #007bff;
$color-secundario: #6c757d;
$color-peligro: #dc3545;
$fuente-base: Arial, sans-serif;
$espaciado: 10px;
```

scss/_buttons.scss

```
@use "variables";

@mixin boton($bg, $color) {
  padding: variables.$espaciado;
  border: none;
  border-radius: 5px;
  background-color: $bg;
  color: $color;
  cursor: pointer;

  &:hover {
    opacity: 0.8;
  }
}

.btn {
  @include boton(variables.$color-secundario, white);
}

.btn-primary {
  @extend .btn;
  background-color: variables.$color-primario;
}
```

```
.btn-danger {  
  @extend .btn;  
  background-color: variables.$color-peligro;  
}
```

scss/_header.scss

```
@use "variables";  
  
.header {  
  background-color: variables.$color-primario;  
  padding: 15px;  
  text-align: center;  
  
  &__title {  
    color: white;  
    font-size: 1.5rem;  
  }  
}
```

scss/main.scss

```
@use "variables";  
@use "buttons";  
@use "header";  
  
body {  
  font-family: variables.$fuente-base;  
  margin: 20px;  
}
```