

Enunciado del reto

Tu equipo está comenzando un nuevo proyecto y ha decidido **usar Sass como preprocesador** para organizar y modularizar estilos. En 30 minutos deberás crear un pequeño sistema de estilos aplicando:

- Variables
- Anidación
- Mixins
- Extends
- Parciales

Objetivo(s) de la actividad

- **Comprender** el rol de Sass como preprocesador CSS.
- **Usar** variables, anidación, mixins y extends para evitar redundancias.
- **Organizar** el proyecto en parciales para modularizar estilos.
- **Validar** un sistema escalable y mantenible en un proyecto real.

Materiales y preparación previa (checklist)

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

Requisitos técnicos y datos

- Archivos `.scss` organizados en `/scss` y compilados a `/css`.
- Uso de `@use` o `@import` para incluir parciales.
- Página base `index.html`.

Instrucciones paso a paso con micro-checks

Tiempo total: 30 minutos

1. **Preparar estructura del proyecto (5')**
 - a. Crea carpetas:

```
proyecto-sass/  
├── index.html  
├── scss/  
│   ├── _variables.scss  
│   ├── _mixins.scss  
│   ├── _base.scss  
│   ├── _components.scss  
│   └── main.scss  
└── css/  
    └── main.css
```

Micro-check: estructura creada en tu proyecto.

2. Definir variables y mixins (8')

a. En `_variables.scss`:

```
$color-primario: #007bff;  
$color-secundario: #6c757d;  
$fuente-base: Arial, sans-serif;
```

b. En `_mixins.scss`:

```
@mixin boton($bg, $color) {  
  padding: 10px 15px;  
  border: none;  
  border-radius: 5px;  
  background-color: $bg;  
  color: $color;  
  cursor: pointer;  
}
```

Micro-check: variables y mixins listos para reutilización.

3. Estilos base y extend (10')

a. En `_base.scss`:

```
body {  
  font-family: $fuente-base;  
  margin: 20px;  
}
```

```
.card {  
  border: 1px solid #ddd;  
  border-radius: 8px;  
  padding: 16px;  
  max-width: 400px;  
  margin: auto;  
}
```

b. En `_components.scss`:

```
@use "variables" as *;  
@use "mixins" as *;  
  
.btn {  
  @include boton($color-secundario, white);  
}  
  
.btn-primary {  
  @extend .btn;  
  background-color: $color-primario;  
}
```

Micro-check: botones reutilizables definidos con `@mixin` y `@extend`.

4. Compilación y prueba (7')

a. En `main.scss`:

```
@use "variables";  
@use "mixins";  
@use "base";  
@use "components";
```

b. Compila con:

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

c. 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>
  <section class="card">
    <h2>Producto Sass</h2>
    <p>Este producto está estilizado con Sass.</p>
    <button class="btn">Secundario</button>
    <button class="btn-primary">Primario</button>
  </section>
</body>
</html>
```

Micro-check: al abrir en navegador, se ven los estilos compilados.

Criterios de éxito y errores comunes

Éxito:

- Proyecto estructurado con parciales y `main.scss`.
- Uso correcto de variables, mixins y extends.
- Botones y tarjeta estilizados y reutilizables.
- CSS final compilado sin errores.

Errores comunes:

- Usar directamente CSS en lugar de Sass.
- Olvidar compilar antes de abrir en navegador.
- No modularizar archivos.

Tiempo total: 30'

Plan de contingencia:

- Si falla compilación, usar `sass --watch scss/main.scss css/main.css`.
- Si no se aplican estilos, revisar rutas de `<link>` y de `@use`.

Extensión opcional (+10'):

- Crear un mixin de `card($bg)` que aplique colores diferentes según contexto.
- Implementar `dark mode` usando variables condicionales.

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>
  <section class="card">
    <h2>Producto Sass</h2>
    <p>Este producto está estilizado con Sass.</p>
    <button class="btn">Secundario</button>
    <button class="btn-primary">Primario</button>
  </section>
</body>
</html>
```

scss/_variables.scss

```
$color-primario: #007bff;
$color-secundario: #6c757d;
$fuente-base: Arial, sans-serif;
```

scss/_mixins.scss

```
@mixin boton($bg, $color) {
  padding: 10px 15px;
  border: none;
  border-radius: 5px;
  background-color: $bg;
  color: $color;
  cursor: pointer;
```

```
}
```

scss/_base.scss

```
body {  
  font-family: $fuente-base;  
  margin: 20px;  
}
```

```
.card {  
  border: 1px solid #ddd;  
  border-radius: 8px;  
  padding: 16px;  
  max-width: 400px;  
  margin: auto;  
}
```

scss/_components.scss

```
@use "variables" as *;  
@use "mixins" as *;
```

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

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

scss/main.scss

```
@use "variables";  
@use "mixins";  
@use "base";  
@use "components";
```