

# Examen Práctico RA2. Crea interfaces web homogéneos definiendo y aplicando estilos.

## PARTE 1

### **1. ¿Qué son los preprocesadores? ¿Qué es SASS y para que se puede utilizar?**

- Es una herramienta que permite desarrollar estilos más mantenibles, con mejor organización y mejor sintaxis. Estas herramientas permiten usar condicionales, variables y estructuras anidadas, así las líneas están mejor organizadas y evita la reescritura de código.
- Sass es un preprocesador de CSS que extiende las capacidades de CSS estándar con características de lenguajes de programación, como variables, anidamiento y funciones, para escribir estilos de manera más eficiente y organizada.

### **2. Indica todas las propiedades que ofrece SASS con un ejemplo de cada una de ellas.**

- Variables. Permiten guardar valores y no tener que reescribir constantemente.
- Anidamiento. Permite agrupar reglas dentro de una misma, parecido a la jerarquía que utiliza HTML.
- Funciones. Permite reutilizar bloques de código con una función pasándole diferentes parámetros.
- Operaciones. Permite realizar operaciones como suma/resta/multiplicación/división/etc.

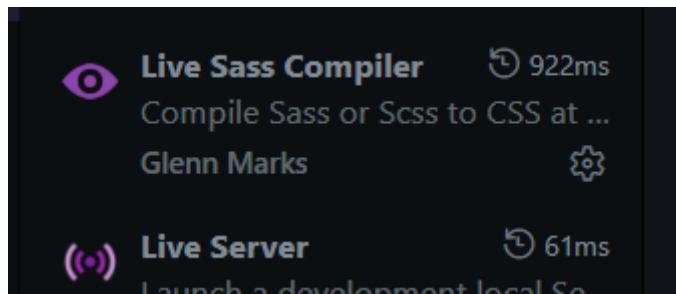
### **3. Explica el proceso de instalación/utilización de SASS.**

Hay varios pasos que son muy sencillos. El primero es instalar la extensión de Live saas compiler, a partir de ahí crear un archivo con la extensión .scss. Puedes escribir el código directamente en este archivo y ahí ya se podrá usar todas las ventajas que tiene el SASS.

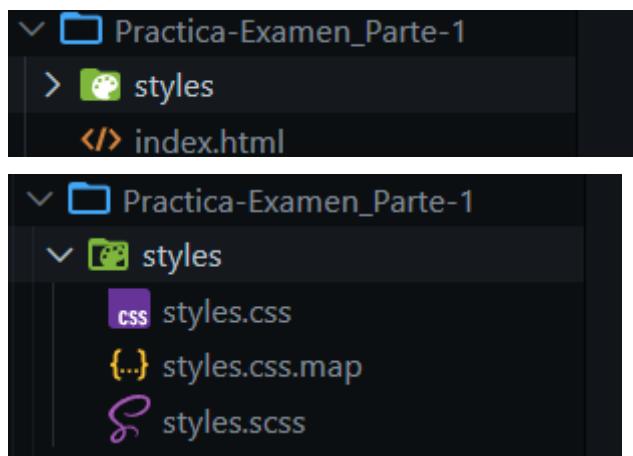
Para que funcione deberás de activar el sass compiler con un botón que se creará en la parte inferior del VS code. una vez activo, se creará un archivo .css automáticamente, este archivo lo enlazas al html y ya estás list@ para hacer uso de los servicios de SASS.

### **4. Realiza capturas y explica cómo lo has añadido en tu VisualStudio.**

Lo primero de todo es instalar la extensión de “Live sass compiler”.

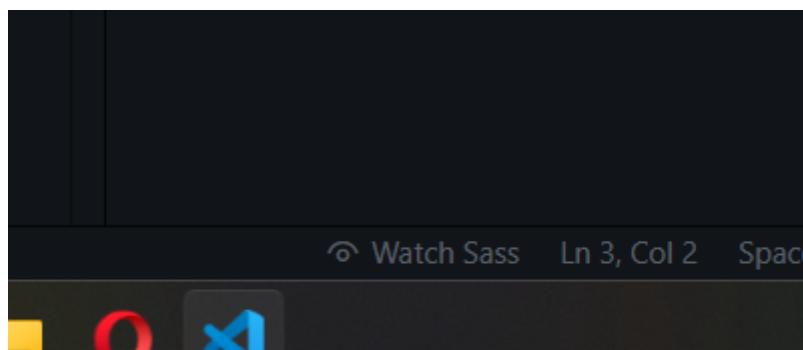


Una vez instalado, creamos el archivo de los estilos, este archivo se creará con una extensión .scss. Automáticamente cuando se active, se generarán automáticamente 2 archivos nuevos, así que es mejor ponerlo en una carpeta aparte del html, para tenerlo mejor organizado.



El archivo creado ha sido styles.scss, los otros dos se han creado automáticamente al activar la extensión instalada anteriormente.

Abajo a la derecha del VS code deberá aparecer un botón como este:



Una vez le hacemos click deberia salir un mensaje como el siguiente:

The screenshot shows the VS Code interface with the 'OUTPUT' tab selected in the top bar. The main area displays the following log output:

```
PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE Filter
-----
Watching...
-----
Not Watching...
-----
Generated:
d:\DAW II\Diseño Interfaces\Tema 2\Practica-Examen_Parte-1\styles\styles.css.map
d:\DAW II\Diseño Interfaces\Tema 2\Practica-Examen_Parte-1\styles\styles.css
-----
Watching...
-----

```

At the bottom, there is a toolbar with icons for Windows, Search, File, Open, Save, and Close, along with a 'Watching...' status indicator.

Aquí nos avisa de que se ha generado 2 archivos, un map del css y un archivo css generado a través del compilador (La extensión instalada). por último solo tendremos que enlazar el archivo css generado al html usando la etiqueta link.

The screenshot shows the VS Code editor with an HTML file open. The code includes a link to the generated CSS file:

```
<title>Document</title>
<link rel="stylesheet" href=".//styles/styles.css" type="text/css">
</head>
<body>
```

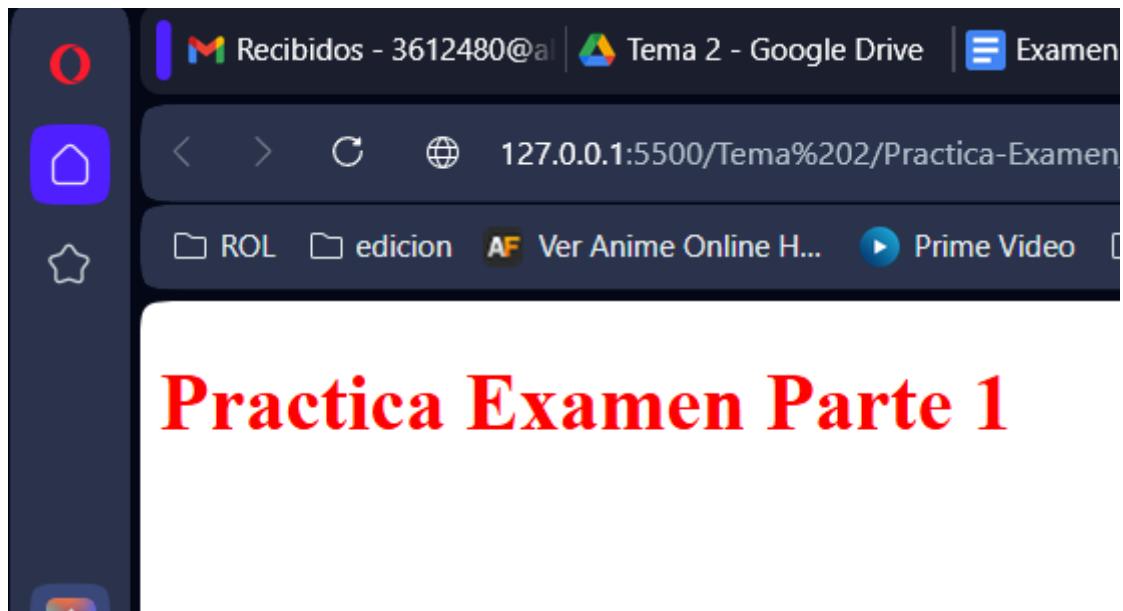
y ya debería de funcionar

The screenshot shows the VS Code editor with two files open: 'index.html' and 'styles.scss'. The 'index.html' file contains the following code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<link rel="stylesheet" href=".//styles/styles.css" type="text/css">
</head>
<body>
<h1>Practica Examen Parte 1</h1>
</body>
</html>
```

The 'styles.scss' file contains the following CSS rule:

```
h1 {
  color: red;
}
```



5. Crea una breve página y emplea en ella SASS para añadir los estilos en lugar de CSS. (Algo sencillo como el video que se muestra en el tema QUE NO SEA IGUAL QUE EL VIDEO) explicando lo que vas utilizando.

Está el código comentado.

6. Dado el siguiente código HTML, que no puede ser modificado, y la siguiente referencia visual del resultado obtenido, crea el archivo styles.scss partiendo de las especificaciones que encontrarás más abajo.

```
<h1>Mi tienda online</h1>

<p>Explora nuestros productos y ofertas exclusivas.</p>

<button class="primary-button">Ver Productos</button>

<button class="secondary-button">Registrarse</button>

<h2>Destacados</h2>

<div class="product">

  <h3>Producto 1</h3>

  <p>Descripción del Producto 1.</p>

  <p class="important-text">$19.99</p>

  <button class="buy-button">Comprar</button>

</div>
```

```

</div>

<div class="product">
    
    <h3>Producto 2</h3>
    <p>Descripción del Producto 2.</p>
    <p class="important-text">$24.99</p>
    <button class="buy-button">Comprar</button>
</div>

<div class="product centered">
    
    <h3>Producto 3</h3>
    <p>Descripción del Producto 3.</p>
    <p class="important-text">$29.99</p>
    <button class="buy-button">Comprar</button>
</div>

```

## **Especificaciones:**

1. Crea un archivo SCSS llamado `styles.scss`.
2. Define las siguientes **variables** en tu archivo SCSS:
  - `\$primary-color`: #3498db
  - `\$secondary-color`: #e74c3c
  - '\$font-family': Arial
  - '\$base-font-size': 16px
3. Aplica los siguientes estilos:
  - **<body>** El fondo del body es de color #f2f2f2, la fuente de tipo \$font-family, y el texto alineado en el centro.
  - **<h1>** El título <h1> debe tener un tamaño de fuente de 2rem y usar el color definido en \$primary-color.

- **<p>** Los párrafos `<p>` deben tener un tamaño de fuente de `1.2rem` y utilizar un color gris oscuro (#555).
- **<button>** Los botones deben tener un padding de `10px 20px`, sin bordes, un radio de borde de `5px`, un margen de `10px` y una transición suave de `0.3s` para el cambio de color de fondo en el efecto hover.
- **.product** Los productos tienen un fondo blanco, un borde sólido de `1px`, un radio de borde de `5px`, un margen de `10px` y un sombreado sutil en hover.
- **<img>** Las imágenes de los productos deben tener una anchura máxima del `100%` y una altura automática.
- **<h3>** Los títulos de los productos `<h3>` deben tener un tamaño de fuente de `1.5rem`.
- **.important-text** Los precios de los productos con la clase `.important-text` deben tener la fuente en negrita y un color definido en `$secondary-color`.

**7. Define un mapa llamado `\$button-styles` que almacene estilos para botones. Cada entrada en el mapa debe tener los siguientes nombres y valores:**

- `primary-button` → `$primary-color`
- `secondary-button` → `#2ecc71`
- `buy-button` → `#f39c12`

**8. Crea un mixin llamado `center-element` que permita centrar elementos tanto horizontal como verticalmente. Puedes utilizar Flexbox.**