



CONCRETE JUNGLE

Urban streetwear

Alan Rabinerson

Índice

Principales cambios en el diseño	3
Cambios a otras páginas	3
Cambios comunes	3
Cambios en el diseño móvil	3
Explicación SASS	4
Variables	4
Mixins	4
Ejemplo de nesting	5

Principales cambios en el diseño

No hay ningún cambio significativo respecto al diseño original, ya que este no existe. Al planear originalmente que el carrito tan solo fuera en formato sticky, no se incluyó un diseño específico para esta página en la guía de estilo. En cambio, sí se han realizado cambios a posteriori para otras páginas, que se detallarán a continuación.

Cambios a otras páginas

Cambios comunes

- En la homepage, se ha modificado el diseño de las tarjetas de producto para añadir un select para elegir la talla, animaciones y transiciones.
- Se han modificado algunos estilos de ambas páginas para igualar más los headers.
- Se ha modificado el estilo del carrito de homepage para hacerlo más parecido al de producto.
- Para tanto producto como homepage, ahora el carrito muestra un scroll vertical para navegar por el carrito en caso de que los elementos hagan overflow en el eje vertical.
- Se ha eliminado el cambio de color de los botones de comprar ahora y añadir al carrito de la homepage al hacer hover sobre ellos. Ahora se cambia su tamaño al hacer hover.
- En los carritos sticky se ha añadido un botón “ver detalles” en la parte superior que redirige a la página del detalle del carrito y un botón para eliminar el producto del carrito.
- En los carritos también se muestra la talla del producto para diferenciar entre productos si se compran varias tallas del mismo.
- Al agregar un producto al carrito, se muestra un mensaje de éxito/error en la parte inferior derecha.

Cambios en el diseño móvil

- En los carritos sticky ahora también se muestran los botones de añadir y restar cantidad a un ítem. Originalmente estaban ocultos al pensar que serían demasiado pequeños como para ser útiles; tras implementar la funcionalidad, se ha decidido que es mejor que se pueda acceder a ella en todas las plataformas.

Explicación SASS

Variables

Se han creado variables para cada uno de los colores y fuentes usados en el diseño. No hace falta explicar la función de estas, ya que sus respectivos nombres ya son suficientemente descriptivos.

```
@font-face {
  font-family: Roboto;
  src: url("/student024/Shop/assets/fonts/Roboto/Roboto-VariableFont_wght.ttf");
}
@font-face {
  font-family: Montserrat;
  src: url("/student024/Shop/assets/fonts/Montserrat/Montserrat-VariableFont_wght.ttf");
}
$color-fondo: #003049;
$color-fondo-footer: rgba(0,0,0,0.3);
$color-texto: #fdf0d5;
$color-accent: #780000;
$color-secundario: #669BBC;
$color-primario: #C1121F;
$fuente-principal: 'Roboto', sans-serif;
$fuente-secundaria: 'Montserrat', sans-serif;
```

Mixins

```
// mixin para botones con personalización de tamaño, colores y redondez
@mixin boton($size, $bg-color, $text-color, $hover-bg-color, $roundness) {
  width: $size;
  height: auto;
  background-color: $bg-color;
  color: $text-color;
  border: none;
  border-radius: $roundness;
  display: flex;
  align-items: center;
  justify-content: center;
  cursor: pointer;
  font-size: 1rem;
  font-weight: 600;

  &:hover {
    background-color: $hover-bg-color;
    transform: scale(1.2);
  }
}
```

Solo se ha creado un mixin para botones, al cual se le han de pasar parámetros para el tamaño, color de fondo, color de texto, fondo en hover y el radio del borde.

Se ha usado para los botones de login, logout, añadir y restar cantidades, el botón de checkout y el botón de eliminar producto. A continuación se puede ver la captura de la implementación para los botones de añadir y restar cantidades.

```
.cantidad-container {
  display: flex;
  align-items: center;
  gap: 0.5rem;
  background: rgba($color-fondo, 0.5);
  padding: 0.4rem 0.75rem;
  border-radius: 10px;
  border: 2px solid rgba($color-secundario, 0.3);
  width: fit-content;

  @media (min-width: 1024px) {
    padding: 0.5rem 1rem;
    gap: 0.75rem;
  }

  @media (min-width: 768px) {
    margin-top: 1rem;
  }
}

button {
  @include boton(auto, $color-secundario, $color-texto, lighten($color-secundario, 10%), 8px);
  min-width: 32px;
  transition: all 0.2s ease;
  box-shadow: 0 2px 8px rgba($color-secundario, 0.3);
}
```

Ejemplo de nesting

El ejemplo de nesting es la definición de los estilos para el contenedor de la cantidad y sus botones, donde mediante nesting se le pasa el estilo a los botones de añadir y restar cantidad y se aplican los media queries.

- Estilos del padre: Se aplica display: flex, alineando los elementos en el centro y con una separación de 8px (0.5rem).
- Un margen interno de ~6.5 px y 12 px (0.4 rem y 0.75 rem) para el eje vertical y horizontal, respectivamente.
- Un borde de 2 px sólido y de un azul claro (variable color-secundario con 0.3 de opacidad) y un radio del borde de 10 px.
- Un ancho ajustado al contenido.

Mediante nesting se aplican los media queries y el estilo para los botones:

- Se utiliza el mixin mediante el cual le damos un ancho automático, un color de fondo azul claro guardado en la variable color-secundario, un color para el texto beige guardado en la variable color-texto, en estado de hover, se cambiará el color de fondo a uno un 10% más claro y el radio del borde será de 8 px.
- Un sombreado de color azul claro creado añadiendo un 0.3 de opacidad al color secundario.
- Indicamos que se apliquen todas las transiciones durante 0.2 s mediante ease para que las animaciones entren y salgan de manera suave.

Mediante nesting, a los estilos de los botones se aplican estilos a sus pseudoclases :active y :hover:

- Para hover se ha añadido que se cambie la escala a 1.15, haciéndolo un 15% más grande, y se agranda el sombreado.
- Para active, se transforma la escala a .95, haciendo el botón un 5% más pequeño al hacer clic sobre él.

```
.cantidad-container {
  display: flex;
  align-items: center;
  gap: 0.5rem;
  background: rgba($color-fondo, 0.5);
  padding: 0.4rem 0.75rem;
  border-radius: 10px;
  border: 2px solid rgba($color-secundario, 0.3);
  width: fit-content;

  @media (min-width: 1024px) {
    padding: 0.5rem 1rem;
    gap: 0.75rem;
  }

  @media (min-width: 768px) {
    margin-top: 1rem;
  }

  button {
    @include boton(auto, $color-secundario, $color-texto, lighten($color-secundario, 10%), 8px);
    min-width: 32px;
    transition: all 0.2s ease;
    box-shadow: 0 2px 8px rgba($color-secundario, 0.3);

    &:hover {
      transform: scale(1.15);
      box-shadow: 0 4px 12px rgba($color-secundario, 0.5);
    }

    &:active {
      transform: scale(0.95);
    }
  }
}
```