

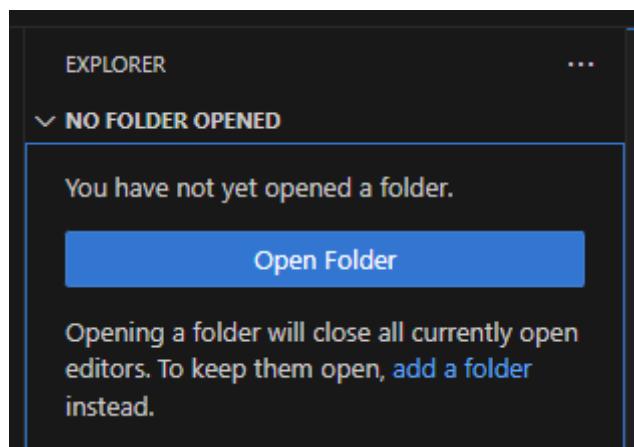
TALLER DE DISEÑO MULTIMEDIA TALLER 1

DESARROLLO FRONTEND USANDO HTML Y CSS - PARTE 1

OBJETIVO: Aplicar los conocimientos de HTML y CSS creando una página web sencilla para un negocio ficticio de ventas, enfocándose únicamente en el frontend y basando el diseño sobre uno previamente entregado.

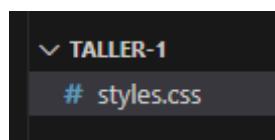
1. Parámetros base

Crea una carpeta donde se almacenará todo el proyecto y ábrela en Visual Studio Code.



Dentro de la carpeta del proyecto, crea un archivo llamado `styles.css`. Este archivo contendrá todos los estilos globales de la página web.

Nota: Cada vista de la página tendrá su propio archivo `.html`. Aún no los crearemos; primero definiremos el diseño base.



El diseñador ha proporcionado la siguiente paleta de colores que utilizaremos en todo el sitio.



De acuerdo con las especificaciones del diseñador, los estilos tipográficos para el proyecto son los siguientes:

- **Fuente base:** Quicksand, sans-serif
- **Tamaños de fuente:**
 - **SM:** 14px
 - **MD:** 16px
 - **LG:** 18px

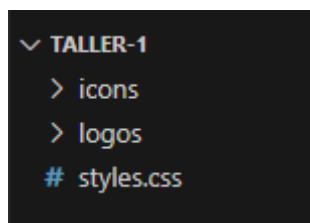
Con la paleta de colores y tipografía definida, ahora podemos comenzar a desarrollar los estilos globales en [styles.css](#). Estos estilos se aplicarán a todas las vistas para mantener coherencia visual.



```
1  :root {  
2      --white: #FFFFFF;  
3      --black: #000000;  
4      --very-light-pink: #C7C7C7;  
5      --text-input-field: #F7F7F7;  
6      --rojo: #E5252E;  
7      --sm: 14px;  
8      --md: 16px;  
9      --lg: 18px;  
10 }  
11 body {  
12     margin: 0;  
13     font-family: 'Quicksand', sans-serif;  
14 }
```

Lo que estamos viendo con el `:root` es un selector pseudo-clase que apunta al elemento raíz del documento y, aquí dentro, definimos todas nuestras variables. De esta forma, cuando queramos hacer referencia a un color o tamaño, en vez de escribirlo directamente, podemos llamar a la variable que lo representa, haciendo más sencillo el proceso de desarrollo.

Finalmente, vamos a crear dos carpetas adicionales en la carpeta en las cuales vamos a guardar nuestros *icons* y *logos*. Estos los encontrarás adjunto a este taller.



Todos los documentos `.html` deben tener la misma estructura en su `<head>` con excepción del `<title>`. Esto con motores gráficos o librerías como React, o Angular podemos hacerlo a través de plantillas, pero en nuestro caso nos aseguraremos de agregar manualmente los siguientes elementos al `<head>` de cada archivo.

HTML

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Google Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Quicksand:wght@300;500;700&display=swap" rel="stylesheet">

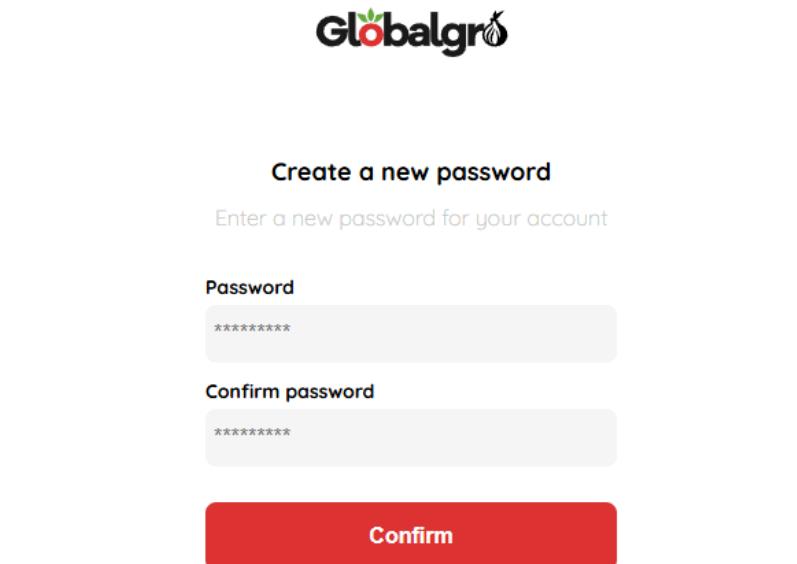
<!-- Hoja de estilos -->
<link rel="stylesheet" href="styles.css">

<!-- Favicon -->
<link rel="icon" type="image/svg" href="./logos/favicon.svg">

<title>Document</title>
```

2. Interfaz de recuperar contraseña

El diseñador nos entrega esta vista para la interfaz de creación de contraseña / recuperar contraseña. Es nuestro trabajo acercarnos lo más posible al diseño. Adicionalmente, también se indica que el logo de la empresa debe desaparecer en tamaños mayores a 640px de ancho.



Crea un archivo `password.html` con la estructura básica de HTML junto con el `<head>` mencionado anteriormente.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7
8   <!-- Google Fonts -->
9   <link rel="preconnect" href="https://fonts.googleapis.com">
10  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
11  <link href="https://fonts.googleapis.com/css2?family=Quicksand:wght@300,700&display=swap" rel="stylesheet">
12
13  <!-- Hoja de estilos -->
14  <link rel="stylesheet" href="styles.css">
15
16  <!-- Favicon -->
17  <link rel="icon" type="image/svg" href="./logos/favicon.svg">
18
19  <title>Document</title>
20 </head>
21 <body>
22
23 </body>
24 </html>
```

Si analizamos el diseño proporcionado, nos damos cuenta que tenemos un contenedor para nuestro login, que a su vez contiene tanto un formulario, como un logo, un encabezado y un subtítulo. El formulario cuenta con dos campos de texto y un botón de confirmación. Esto podemos maquetarlo de la siguiente manera:

```
1 <div class="login">
2   <div class="form-container">
3     
4
5     <h1 class="title">Create a new password</h1>
6     <p class="subtitle">Enter a new password for your account</p>
7
8     <form action="/" class="form">
9       <label for="password" class="label">Password</label>
10      <input type="password" id="password" placeholder="*****" class="input input-password">
11
12      <label for="new-password" class="label">Confirm password</label>
13      <input type="password" id="new-password" placeholder="*****" class="input input-password">
14
15      <input type="submit" value="Confirm" class="primary-button login-button">
16    </form>
17  </div>
18 </div>
```

Lo que resta ahora es asignar los estilos a cada uno de los elementos. Primero, para ambos contenedores, vamos a aplicar un sistema de casillas usando `display: grid`, de tal forma que podamos distribuir la información en formato de formulario.

```
1 .login {
2   width: 100%;
3   height: 100vh;
4   display: grid;
5   place-items: center;
6 }
7 .form-container {
8   display: grid;
9   grid-template-rows: auto 1fr auto;
10  width: 300px;
11 }
```

Para el logo, el título y el subtítulo, aplicamos reglas específicas relacionadas principalmente con los tamaños de tal forma que se vean coherentes con el diseño. Recordemos que tenemos acceso a las variables globales para los colores y los tamaños de la tipografía, así que vamos a utilizarlas:

```

1 .logo {
2   width: 150px;
3   margin-bottom: 48px;
4   justify-self: center;
5   display: none;
6 }
7 .title {
8   font-size: var(--lg);
9   margin-bottom: 12px;
10  text-align: center;
11 }
12 .subtitle {
13   color: var(--very-light-pink);
14   font-size: var(--md);
15   font-weight: 300;
16   margin-top: 0;
17   margin-bottom: 32px;
18   text-align: center;
19 }
```

```

1 .form {
2   display: flex;
3   flex-direction: column;
4 }
5 .label {
6   font-size: var(--sm);
7   font-weight: bold;
8   margin-bottom: 4px;
9 }
10 .input {
11   background-color: var(--text-input-field);
12   border: none;
13   border-radius: 8px;
14   height: 30px;
15   font-size: var(--md);
16   padding: 6px;
17   margin-bottom: 12px;
18 }
19 .primary-button {
20   background-color: var(--rojo);
21   border-radius: 8px;
22   border: none;
23   color: var(--white);
24   width: 100%;
25   cursor: pointer;
26   font-size: var(--md);
27   font-weight: bold;
28   height: 50px;
29 }
30 .login-button {
31   margin-top: 14px;
32   margin-bottom: 30px;
33 }
```

Nota: Es importante fijarse que el logo tiene `display: none`, lo que significa que no se va a visualizar normalmente. Más adelante debemos agregar una forma de que solo sea visible en tamaños de pantalla inferiores a 640px, según los requerimientos de diseño.

Por último, terminamos los estilos relacionados con el formulario. Aquí usamos `display: flex` para tener mayor libertad para ubicar los elementos y evitar el colapso de márgenes.

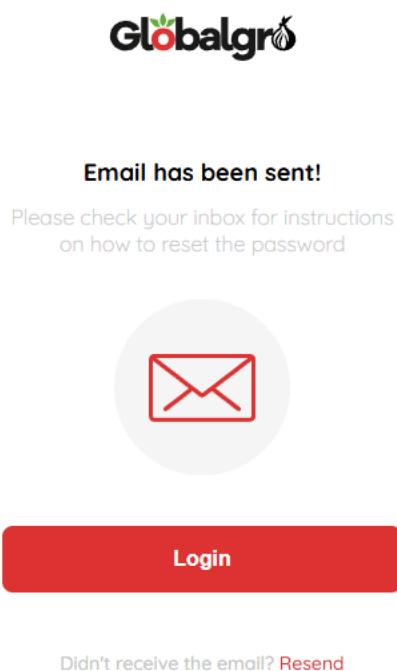
Lo único que resta de esta vista es añadir el responsive para tamaños de pantalla superiores a 640px. Esto lo logramos a través de un media query, en el cual aplicamos un estilo basándonos en una condición. En este caso, vamos a aplicar `display: block` hasta que lleguemos a 640px.

```

1 @media (max-width: 640px) {
2   .logo {
3     display: block;
4   }
5 }
```

3. Interfaz de envío de correo

Para esta interfaz, el diseñador nos entrega la siguiente vista. Usando los conocimientos y procedimientos aplicados en el paso anterior, podemos transformar esta interfaz en realidad.



Crea un archivo `email.html` con una estructura similar a `password.html`. Asegúrate que tenga el `<head>` correspondiente.

Tip: Reutiliza gran parte del código y los estilos. Aquí vemos que no tenemos un formulario como en `password.html`, pero sí tenemos el botón, el título, el subtítulo y el logo en la parte superior (el cual también tiene que aparecer solamente en tamaños inferiores a 640px). Lo único que restaría aquí es colocar la nueva imagen que se encuentra en los recursos y el mensaje de la parte inferior.

Puedes realizar esta vista con tan solo 5 reglas de estilos más.

4. Interfaz de login

Para esta interfaz, al igual que la anterior, el diseñador nos entrega la siguiente vista. Usando los conocimientos y procedimientos aplicados, también podemos llevarlo a la realidad.

The image shows a login interface for 'Globalgrado'. At the top is the 'Globalgrado' logo. Below it is a form with two input fields: 'Email address' containing 'email@uniboyaca.edu.co' and 'Password' containing '*****'. A large red button labeled 'Log in' is centered below the fields. To the right of the 'Log in' button is a link 'Forgot my password' in red. Below the 'Forgot my password' link is a 'Sign up' button with a red border and red text.

Crea un archivo `login.html` con una estructura similar a `password.html` y a `email.html`. Asegúrate que tenga el `<head>` correspondiente.

Tip: De igual forma, reutiliza gran parte del código y los estilos. Lo nuevo aquí es el link de “Forgot my password”, el cual es una etiqueta `<a>` a la cual puedes asignarle una clase o id para los estilos o directamente acceder a ella como hija del formulario. Para el botón de registro, utiliza un borde e invierte los colores de fondo y texto.

Puedes realizar esta vista con tan solo 2 reglas de estilos más.

5. Interfaz de registro

El diseñador nos entrega la siguiente vista. Para este caso, debemos hacer algunos ajustes, específicamente en cada uno de los campos para la visualización.

My account

Name
Jose

Email
email@uniboyaca.edu.co

Password

My account

Name
Jose

Email
email@uniboyaca.edu.co

Password

Create

Create

Crea un archivo `signup.html` con una estructura similar a `login.html`. Asegúrate que tenga el `<head>` correspondiente. El diseñador nos indica que esta vista debe tener dos visualizaciones: Una para tamaños de pantalla menores a 640px, en dónde el formulario se ajuste al tamaño vertical del dispositivo y uno para tamaños mayores a 640px.

Tip: Para resolver el requerimiento, debemos trabajar con otro `@media` query tal como se muestra en la interfaz de recuperar contraseña.

6. Reto: Interfaz de edición de datos

Esta última parte consistirá de un pequeño reto. Debes desarrollar un archivo llamado `edit.html` con una estructura similar a la interfaz de registro `signup.html`.

Esta vista contará con campos para los datos del usuario, pero sin posibilidad de editarse. Puedes usar párrafos o etiquetas `` estilizadas para simplemente mostrar esta información.

Adicionalmente, la vista contará con un botón sin relleno y con bordes de color --rojo, tal como el botón de “Sign up” de la vista `login.html`.