**Ubicación en mi ordenador:**

**C:\Users\jpi\Documents\Proyectos\torres**

**Estructura de Archivos**

Todos los archivos fuente en los que tendréis que trabajar están en la carpeta SRC dentro de la carpeta TORRES:

Utilizo SASS para la pre-compilación de CSS, por tanto los archivos fuentes de estilos tienen la extensión .SCSS.

Dentro de la carpeta SCSS:

**Layout** – Archivos referentes a módulos de estructura (layout) de la aplicación (cabecera, contenido, pie, grid, etc). Dentro del markup os daréis cuenta de qué clases tienen que ver con el layout porque el nombre empieza por “l-nombre” (l-content, l-header, l-footer, etc).

**Modules** – Todos los componentes que conforman la página (botones, labels, títulos, etc)  
 **Theme** – Todos los archivos que corresponden al tema (colores, especificaciones básicas de fuentes, márgenes, tamaños, etc). Dentro del markup te darás cuenta de qué clases tienen que ver con el tema de cada familia porque el nombre empieza por el nombre de la familia y termina en “Theme”.  
  
Aquí también podréis encontrar los temas de cada familia (o marca), cada familia tiene una hoja de SASS especifica (nombredelamarcaTheme), cada uno de estos archivos contiene estilos específicos para cada familia que define sobre todo gama de colores y fuentes. Como veréis en el markup cada familia tiene una clase (esta clase representa el tema de la familia) que se incluye en en la etiqueta <body>.

<body class="stTheme">

La idea es que tan sólo poniendo esta clase cada familia herede los estilos de todos los componentes (colores, fuentes, tipos de letra).  
  
**Si se tiene que crear una nueva familia:**1. Crear el archivo HTML (lo puedes copiar de otra familia ya que todas tienen la misma estructura, lo único que varía son los componentes como redes sociales, logos, fotos, etc)

2. Incluir en la configuración de Webpack la llamada a esta nueva página. En el archivo webpack.dev.js dentro de la carpeta raíz (línea 91) añadir un nuevo plugin tal que:

new HtmlWebpackPlugin({

template: "./src/distWines.html",

inject: false,

hash: false,

filename: "distWines.html"

}),

En el parámetro de template y filename reemplazar por el nombre de la nueva página. Para que el entorno reconozca los cambios tendrás que parar el servidor local (CTRL + C en la consola) y volverlo a arrancar ( **npm run dev**).  
  
3. La app en desarrollo tiene un índice que nos sirve de guía para ver todas las páginas que se han creado (index.html). Al hacer un **npm run dev** en la consola del editor es lo primero que se cargará en el navegador. Cuando se crea una página nueva hay que añadirla también al índice.

4. Luego hay que crear el theme de la familia para añadirle estilos. Dentro del markup añadir la clase del tema de la familia a la etiqueta <body>:

<body class="familiaTheme">

5. Crear archivo “\_nombrefamiliaTheme.scss” dentro de la carpeta SCSS/THEME (Se puede copiar de otra familia ya que son iguales). El nombre de la clase tiene que ser igual al nombre del archivo .scss.

6. Incluir el archivo .scss en “styles.scss” dentro de la carpeta SCSS. Hay que importar el modulo SCSS para que el precompilador lo pueda interpretar:

@import "theme/\_temadelafamilia.scss";

**LEVANTAR ENTORNO DE DESARROLLO FRONT**

Si necesitaras trabajar en el proyecto desde mi ordenador:

1. Abrir el Visual Studio Code

2. Verificar que estas en la ruta del proyecto en la consola (la verás debajo en el mismo editor, si no la ves **CTRL+Ñ**)

3. Dentro de la ruta del proyecto ejecutar: **NPM RUN DEV**. Al final del proceso verás que te dice: “COMPILES SUCCESFULLY”, ya puedes empezar a trabajar.

4. Abrir navegador y poner **LOCALHOST:8080** (Esto cargará la aplicación en tu local)

5. Mientras vas haciendo las modificaciones podrás ver cómo quedan al momento de guardar el archivo en caliente (no necesitas refrescar el navegador)

6. Una vez hayas terminado necesitas compilar todos los archivos para crear tus archivos de producción (para nuestro caso una carpeta **DIST**)

**Generar archivos de Producción en local para integrarlos a SharePoint:**

1. Dentro de la ruta del proyecto ejecutar: **NPM RUN BUILD** Al final del proceso verás que te dice : “COMPILES SUCCESFULLY”, y verás que se ha creado en la carpeta del proyecto una nueva carpeta DIST.

2. Esta carpeta contiene TODA la maqueta preparada para integrarla a SharePoint.

**Repositorio GIT Torres  
URL:** <https://github.com/lowtrux/Torres>  
  
Luego de hacer algún cambio, tendréis que actualizar el repositorio GIT:  
  
**En consola de Studio Code: (CTRL + Ñ)**

1. **git add .** Añade todos los archivos para que se incluyan en el repositorio GIT
2. **git commit -m** “mensaje descriptivo del commit” (no olvidéis poner un mensaje que describa qué es lo que se está subiendo).
3. **git push** Para subir los archivos al repositorio GIT