



MERCADO SHOPS

Guía para configurar tu código HTML

Para que tu HTML pueda verse en tu tienda debes subir un archivo comprimido de extensión .zip de hasta 1MB. El archivo debe contener en su interior los siguientes archivos:

- **header.hbs**
- **footer.hbs**
- **layout_styles.css**
- **widgets.json**

Estos archivos son necesarios para que el archivo .zip sea considerado válido, además de estos pueden agregarse otros archivos, como por ejemplo, templates parciales de handlebars.

Pasos para preparar tu entorno de desarrollo local

Para configurar tu entorno de desarrollo es necesario que tengas instalado los paquetes node (versión 10) y npm, los cuales los puedes adquirir desde <https://nodejs.org/download/release/v10.24.1/>

Una vez instalados, y desde la consola de tu PC, debes pararte en la carpeta base del entorno (donde está ubicado este mismo archivo) y ejecutar el siguiente comando:

- **npm install**

Este proceso puede demorar unos minutos, pero una vez finalizado, verán que se abrirá automáticamente en su navegador una pestaña con el proyecto cargado y tendrán su entorno listo para comenzar a trabajar.

Adicionalmente, todos los cambios que hagan en los archivos del proyecto se verán reflejados de manera instantánea con lo que están viendo en su navegador.

Una vez finalizado el trabajo de diseño, solo resta volver a la consola y ejecutar el comando:

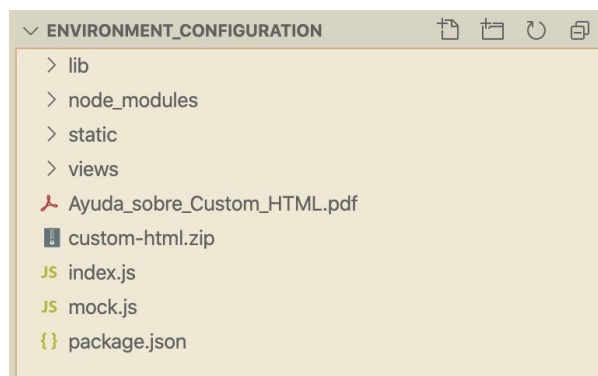
- **npm run create-zip**

Esto ejecutará un proceso que encargará de recorrer el entorno y generar un archivo **custom-html.zip** listo para ser subido en la configuración de custom HTML de su tienda.



Estructura del proyecto

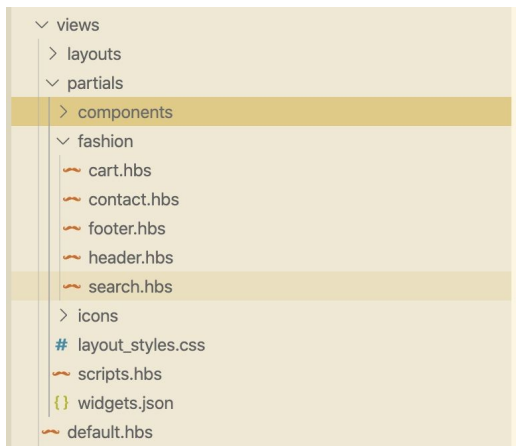
Si siguieron los pasos mencionados hasta ahora, van a notar que tienen una estructura de carpetas como la de la siguiente imagen:



veamos en detalle a qué corresponde cada una:

La carpeta **views/layouts** se corresponde con la **configuración propia del proyecto y no debe ser tomada en cuenta**, los cambios que se hagan en ella no se contemplarán en el resultado final.

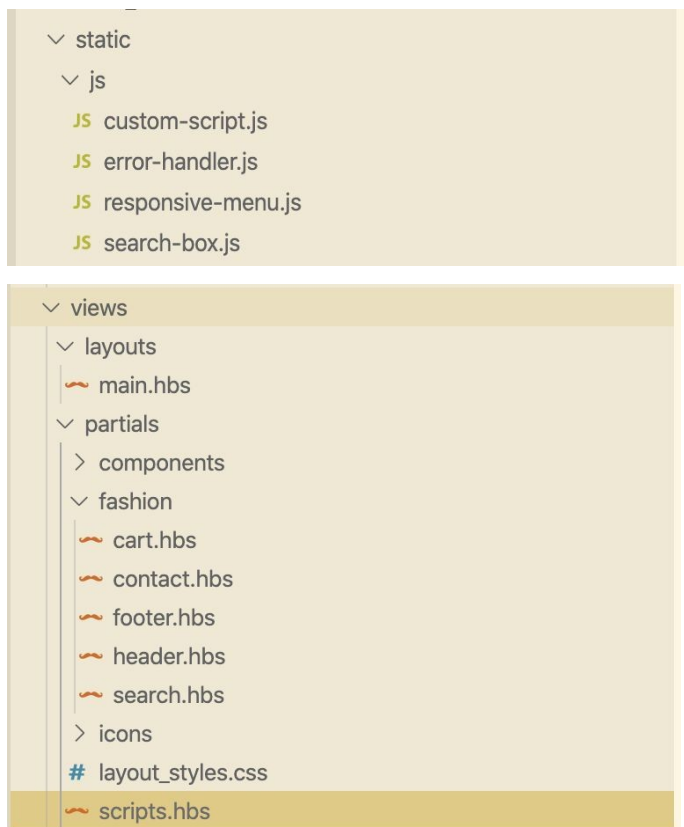
Sobre la carpeta **views/partials** es en donde **centraremos gran parte del trabajo de desarrollo**. Allí pueden encontrar los templates del header y del footer junto a otros componentes necesarios para su visualización.

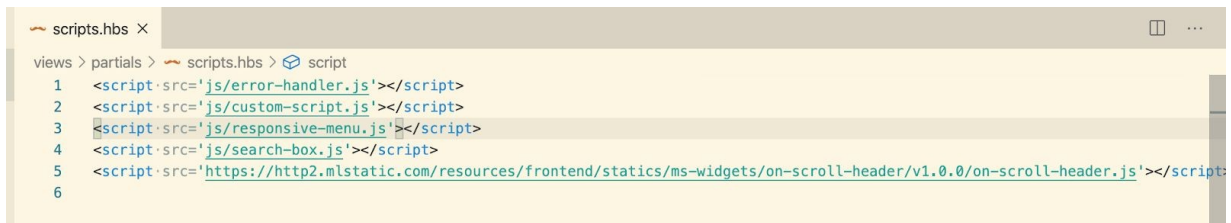


Los archivos de esta carpeta serán los que finalmente se reflejarán en el diseño de su tienda. Sobre ellos se pueden hacer todas las modificaciones necesarias, e incluso se pueden agregar nuevos archivos .hbs sobre los que trabajar. **La única condición es que los archivos de header.hbs y footer.hbs no sean borrados**, ya que estos 2 son los principales y necesarios para el archivo .zip que se debe subir.

Cómo agregar código Javascript en tu proyecto

Todo el **código javascript** se debe incluir en la carpeta **static/js** y luego **referenciado** en el archivo **views/partials/scripts.hbs**, como se puede ver en la siguiente imagen:





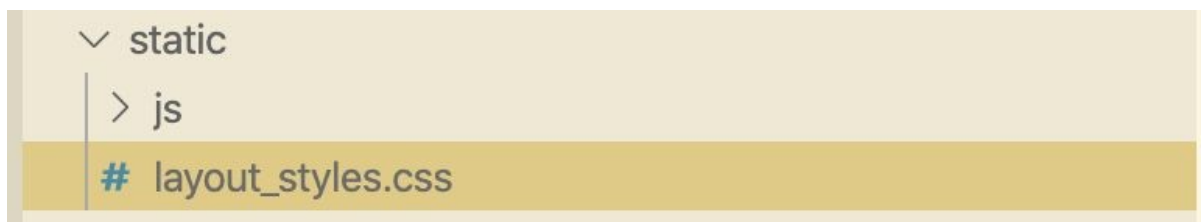
```
views > partials > scripts.hbs > script
1 <script src='js/error-handler.js'></script>
2 <script src='js/custom-script.js'></script>
3 <script src='js/responsive-menu.js'></script>
4 <script src='js/search-box.js'></script>
5 <script src='https://http2.mlstatic.com/resources/frontend/statics/ms-widgets/on-scroll-header/v1.0.0/on-scroll-header.js'></script>
6
```

Adicionalmente, se pueden referenciar archivos externos como se puede apreciar en la línea 5 de la última imagen. De esta manera, todo el código JS que incluyan se tomará en cuenta a la hora de crear el .zip para la configuración de la tienda.

Considerar que el código JS incluido por default en el template, está preparado para funcionar correctamente con los archivos por default de header y footer, por lo que si se hacen modificaciones en estos archivos, seguramente sea necesario reemplazar el código JS original, ya sea modificándolo o borrándolo.

Hojas de estilos

La hoja de estilos del diseño custom se encuentra en la carpeta **static**:

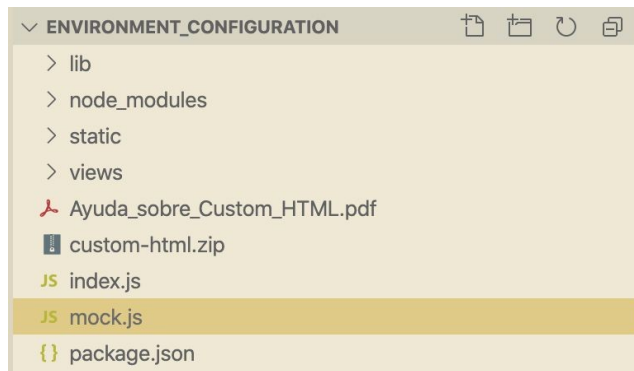


Esta incluye todos los estilos referidos al template, puede modificarse a necesidad pero debe permanecer en la carpeta static y mantener el nombre de **layout_styles.css** para la correcta generación del archivo .zip final.

Variables del template

En los archivos de handlebars, podrán acceder a variables que tendrán disponibles y que harán referencia a las diferentes configuraciones que se pueden hacer desde el administrador de diseño. El listado de estas variables junto a su utilización será detallado más adelante en este documento.

Para simular el comportamiento de estas variables, se incluye un archivo mock.js en donde están listadas y separadas por header y footer.



Pueden modificar libremente este archivo para ver cómo impactan esos cambios en su diseño, sin embargo, los cambios en ese archivo no se verán reflejados en el archivo .zipfinal, dado que esas configuraciones deben hacerse desde el editor de diseño.

Lenguaje de templatizado Handlebars - documentación

-<https://handlebarsjs.com/>

Variables Header

Estas son las variables que podés utilizar en el archivo de código del header:

flash: componente flash

- **enabled:** permite habilitar o no un mensaje de tipo flash.
- **message:** mensaje que se muestra en el componente de flash.
- **action**
 - **href:** url adonde se redirigirá al presionar sobre el mensaje.
 - **message:** mensaje con link clickable dentro del flash.

menu: es un array con las categorías que se muestran en el header y en el footer.

- **url:** url adonde dirige ese link del menu
- **name:** nombre de la categoría.

En handlebars, para recorrer todas las entradas de un array usamos el comando

#each.Ej. de uso:

```
{{#each menu}}  
  
  <li class="nav-fashion-list__item">  
  
    <a href="{{this.url}}"  
    class="header--custom-text">{{this.name}}</a>  
  </li>  
{{/each}}
```

headerTransparent: valor del color del header parametrizado para que se vea con la transparencia correcta.

Ej de uso: `style="background: rgba({{headerTransparent}})"`

sale: categoría destacada

- **url:** link configurado para la categoría destacada.
- **text:** texto configurado.

logo: url que apunta a la imagen configurada para el logo de la tienda.

shop.name o **name:** nombre configurado de la tienda.

contact.href: link de redirección a la página de

contacto.**cart.href:** link de redirección al carrito.

bannerStyles: esta variable solo estará disponible si se configura un imagen en lugar de unslider. Contiene la url de la imagen configurada junto con estilos de visualización.

```
{{#if bannerStyles}}  
  <div class="nav-header__banner">  
    <div class="nav-header__banner-image"  
style="{{bannerStyles}}" itemscope></div>  
  </div>  
{{/if}}
```

Variables footer

Estas son las variables que podés utilizar en el archivo del footer:

logo: url que apunta a la imagen configurada para el logo de la

tienda.**shop.name** o **name:** nombre configurado de la tienda.

logoCaption: mensaje personalizado dentro del footer, este mensaje en el footer de modase muestra debajo del logo por default, pero puede colocarse en cualquier lugar del footer.

socialNetworks: array con las redes sociales configuradas.

- url: link que apunta a la red social configurada
- nombre: texto configurado para la red social en cuestión.

showContact: configuración para mostrar o no la información de contacto, devuelve “true” o “false”.

showPhone: configuración para mostrar o no el teléfono, devuelve “true” o “false”.

phone.raw: valor del número de teléfono configurado sin formatear, generalmente utilizado junto con el “tel:” dentro de un href para que el navegador entienda que debe abrir la aplicación para llamar por teléfono.

Ej. de uso:

```
<a class="header--custom-text" itemprop="telephone" href="tel:
{{phone.raw}}" rel="nofollow">Tel:
{{phone.formatted}}</a>
```

phone.formatted: valor configurado del teléfono, formateado de manera amigable para mostrarlo en pantalla.

showEmail: configuración para mostrar o no la dirección de email, devuelve “true” o “false”.

email: dirección de correo electrónico configurada.

address: datos de dirección

- **address:** calle y número configurado.
- **city:** ciudad configurada.
- **state:** estado configurado.

Para tiendas de Brasil

showBusinessData: configuración para mostrar o no datos fiscales.

fiscalData: datos fiscales configurados

- **legal_name**
- **legal_doc**

Para tiendas de Argentina

showQrData: configuración para mostrar códigos QR AFIP o no, devuelve “true” o “false”.

fiscalData: datos fiscales configurados

- **qr_link:** link hacia los datos de la tienda en la afip.
- **qr_img:** url para mostrar la imagen del código qr.