

Ejercicio guiado - Maquetando sitio de empresa

Houstel

Ejercicio guiado - Maquetando sitio de empresa Houstel	1
Contexto	2
Entregables	2
Guías de estilo	2
Representaciones visuales	3
Imágenes	3
Organizando parciales del proyecto	4
Layout	5
Header	5
Main	5
Form	6
¿En dónde integró estos layout dentro del patrón 7-1?	6
Components	7
Inputs	7
Botones	8
¿Cómo integrarlos en el proyecto?	9
Base	9
¿Cómo integrarlos en el proyecto?	10
Abstracts	11
¿Cómo integrarlos en el proyecto?	11
¿Qué hacer con las carpetas restantes?	12
Importar parciales al manifiesto	12



¡Comencemos!

Contexto

Imaginemos que una empresa llamada Houstel desea maquetar su nueva página de inicio. Esta página de inicio se encuentra totalmente lista para ser creada, ya que diseñadores UX/UI identificaron las necesidades del usuario y tradujeron esas necesidades en una representación visual.

Entregables

Guías de estilo

El primer entregable es la guía de estilos usada en el mockup. Esta incluye la información relacionada a la paleta de colores, tipografías, tamaños de fuentes, así como también los componentes que contiene la representación visual.

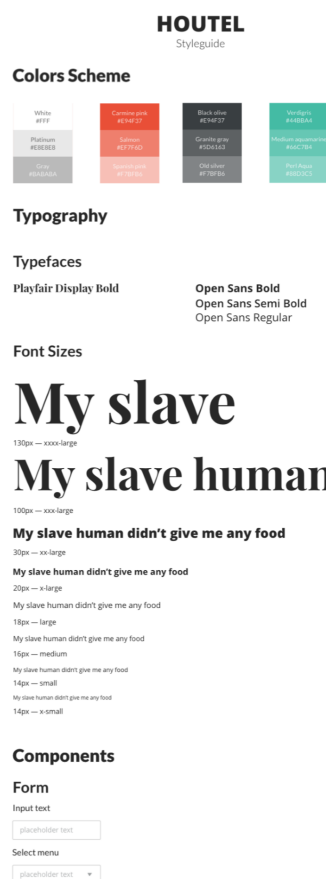


Imagen 1. Houtel - Styleguide.
Fuente: Desafío Latam.

Representaciones visuales

El otro entregable cedido por Hostel es la representación visual del home. Este está compuesto por un mockup con la interfaz del sitio.

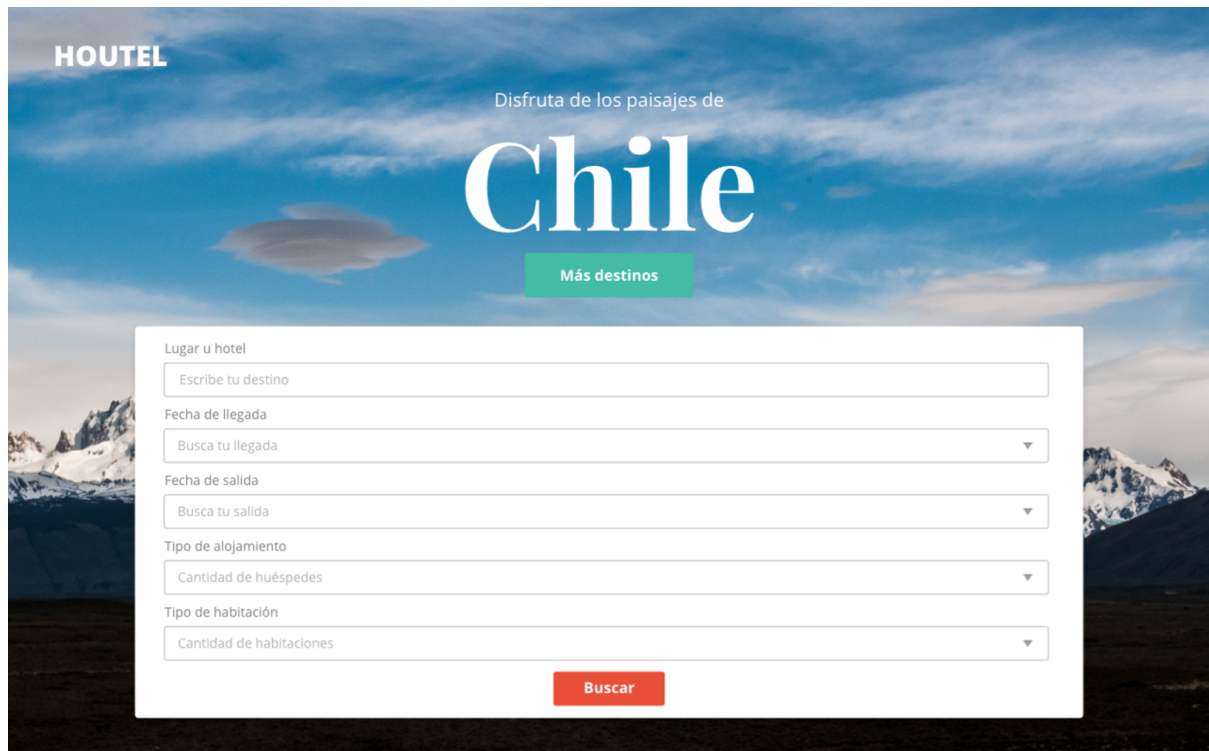


Imagen 2. Representación visual del home.
Fuente: Desafío Latam.

Imágenes

Las imágenes corresponden al fondo principal y al logo del sitio. Con los elementos entregados por el UI tendremos el material suficiente para poder comenzar con nuestro proyecto.

Podrás acceder al material de apoyo para realizar este ejercicio con el nombre **Material Apoyo Ejercicio guiado - Hostel.zip**

¿Listo? ¡Comencemos! El primer paso es crear la base de nuestro proyecto. Para eso utilizaremos la terminal.

Creemos el directorio principal del proyecto. Lo llamaremos `houstel_site`, para esto crearemos una carpeta llamada `houstel-site` y la abriremos desde nuestro editor favorito y desde el editor crearemos la siguiente estructura.

```
houstel-site/  
  assets/  
    css/  
      main.css  
    sass/  
      abstracts/  
      base/  
      componentes/  
      layout/  
      pages/  
      themes/  
      vendors/  
      main.scss  
    images/  
  index.html
```

Organizando parciales del proyecto

Para organizar los archivos parciales que utilizaremos en nuestro proyecto es necesario identificar en que directorio irá cada uno de los elementos que componen nuestra interfaz. Primero revisemos que elementos componen esta interfaz.

Layout

Lo primero que podremos hacer es definir las partes que componen el diseño del sitio a maquetar.

Este sitio lo podemos separar de la siguiente manera.

Header



Imagen 3. Separación del sitio.
Fuente: Desafío Latam.

La primera sección que podemos identificar es el header o cabecera del sitio.

Main

La siguiente parte del diseño que podemos identificar es el main o contenido principal del sitio.



Imagen 4. Main (contenido principal del sitio).
Fuente: Desafío Latam.

Form

El último layout identificable es el form o formulario.


Un formulario web con un borde rojo. Contiene los siguientes campos: un campo de texto para "Lugar u hotel" con el placeholder "Escribe tu destino"; un campo de selección para "Fecha de llegada" con el placeholder "Busca tu llegada"; un campo de selección para "Fecha de salida" con el placeholder "Busca tu salida"; un campo de selección para "Tipo de alojamiento" con el placeholder "Cantidad de huéspedes"; y un campo de selección para "Tipo de habitación" con el placeholder "Cantidad de habitaciones". Debajo de estos campos hay un botón rojo con el texto "Buscar".

Imagen 5. Formulario.
Fuente: Desafío Latam.

Estos elementos que vimos anteriormente son la base de los componentes integrados al interior de cada uno de los layouts.

¿En dónde integró estos layout dentro del patrón 7-1?

Integrar estos layout dentro del directorio de Sass es fácil. Primero debemos ingresar a la carpeta especializada para estos estilos llamada "Layout" y luego crear tres archivos con nombres que representen su tipo. En base a lo visto anteriormente sus nombres deberán ser: `_header.scss`, `_main.scss` y `_forms.scss`.

Con el archivo `_main.scss` es mejor usar un nombre más específico debido al alcance de nombre del manifiesto, por lo tanto un buen nombre para este archivo es `_main-section.scss`.

Ahora sí, ingresemos a la carpeta layout y creamos los archivos `_header.scss`, `_main-section.scss` y `_forms.scss`.

```
houstel-site/  
  assets/  
    css/  
      main.css  
    sass/  
      layout/  
        _header.scss  
        _main-section.scss  
        _forms.scss  
      main.scss  
    images/  
  index.html
```

Components

Para agregar los componentes usaremos la guía de estilos facilitada por el diseñador UI. En él encontraremos:

Inputs

El primer componente que encontraremos serán los inputs. Esto como sabemos forma parte del formulario de búsqueda.

Input text

Select menu

 ▼

Imagen 6. Inputs del formulario.
Fuente: Desafío Latam.

Botones

El otro componente que incluye esta interfaz son los botones.

Estos botones contienen información muy importante como el tamaño de los botones, además de los colores que tienen estos dependiendo de la acción que realice el usuario.

Button size



Large button



Medium button



Small button

Button color scheme



Default



Positive

Imagen 7. Botones de la interfaz.
Fuente: Desafío Latam.

¿Cómo integrarlos en el proyecto?

Para integrarlos al proyecto deberemos realizar la misma acción hecha anteriormente, o sea, volver a la raíz de la carpeta Sass, ingresar a components y finalmente crear los dos componentes vistos en la guía de estilos.

```
houstel-site/  
  assets/  
    css/  
      main.css  
    sass/  
      layout/  
        _header.scss  
        _main-section.scss  
        _forms.scss  
      components/  
        _inputs.scss  
        _buttons.scss  
      main.scss  
    images/  
  index.html
```

Base

Ahora que ya tenemos definido el diseño y los componentes que podremos utilizar en nuestra maqueta el siguiente paso es agregar los archivos que serán la base de nuestro nuestros estilos.

Si revisamos Sass Guidelines, podremos encontrar que dentro de este directorio se pueden incluir tres parciales.

El primero es llamado `_base.scss`, en este archivo se pueden definir algunas reglas de estilo para elementos comúnmente usados por HTML.

El segundo es `_reset.scss`. En el que podremos agregarles reglas para resetear y normalizar nuestros estilos en todo tipo de navegadores. Este tema lo abordaremos con más detalle más adelante.

El último parcial que contendrá este directorio es `_typography.scss`. Aquí podremos agregar todas las reglas relacionadas a las tipografías del proyecto.

De manera optativa, y si el proyecto lo requiere, podemos agregar un parcial que integre en su interior las animaciones de nuestra maqueta.

¿Cómo integrarlos en el proyecto?

Agreguemos estos parciales volviendo al directorio sass, luego ingresando a base y finalmente agregando los archivos al directorio.

```
houstel-site/  
  assets/  
    css/  
      main.css  
    sass/  
      layout/  
        _header.scss  
        _main-section.scss  
        _forms.scss  
      components/  
        _inputs.scss  
        _buttons.scss  
      base/  
        _base.scss  
        _reset.scss  
        _typography.scss  
      main.scss  
    images/  
  index.html
```

Abstracts

El último paso será agregar los parciales dentro del directorio abstracts del patrón 7-1. Como sabemos en este directorio se encontrarán todas las herramientas de Sass que reutilizaremos a lo largo del proyecto.

En nuestro caso utilizaremos sólo dos herramientas para realizar este proyecto. La primera serán las variables y la segunda serán los mixin los cuáles nos permitirán crear algunas reglas CSS que podremos reutilizar dentro de nuestras hojas de estilo agrandando algunos argumentos en su interior.

Estas funciones de Sass las revisaremos más adelante, así que por ahora crearemos estos parciales dentro de la carpeta abstracts.

¿Cómo integrarlos en el proyecto?

Para hacerlo volveremos al directorio sass, posteriormente entraremos a la carpeta abstracts y finalmente crearemos los parciales `_variables.scss` y `_mixins.scss`.

```
houstel-site/  
  assets/  
    css/  
      main.css  
    sass/  
      layout/  
        _header.scss  
        _main-section.scss  
        _forms.scss  
      components/  
        _inputs.scss  
        _buttons.scss  
      base/  
        _base.scss  
        _reset.scss  
        _typography.scss  
      abstracts/  
        _variables.scss  
        _mixins.scss  
      main.scss  
    images/  
  index.html
```

¿Qué hacer con las carpetas restantes?

Las carpetas restantes como “pages” y “themes” las dejaremos como están, ya que por el tamaño del proyecto no son necesarias. En cuanto al directorio “vendors” más adelante la utilizaremos para agregar algunas dependencias que nos serán útiles a medida que avancemos con el proyecto.

Importar parciales al manifiesto

El último paso a realizar es el más importante de toda la serie, ya que importamos todos los parciales que incluimos dentro de los directorios del patrón 7-1 hacia el manifiesto.

Es importante recordar que el orden de los parciales en este archivo es importante para el buen funcionamiento de todos nuestros estilos a lo largo del proyecto.

El orden que deberían tener los `@imports` es el siguiente:

```
// main.scss # Manifest File
// abstracts
@import 'abstracts/variables';
@import 'abstracts/mixins';
// Vendors
// Base
@import 'base/base';
@import 'base/reset';
@import 'base/typography';
// Layout
@import 'layout/header';
@import 'layout/main-section';
@import 'layout/forms';
// Components
@import 'components/buttons';
@import 'components/inputs';
// Pages
// Theme
```

Usemos como plantilla este orden para hacerlo en nuestro proyecto.

Comencemos agregando los parciales de abstracts.

```
// abstracts
@import 'abstracts/variables';
@import 'abstracts/mixins';
```

Vendor por ahora no contendrá nada.

```
// Vendors
```

Sigamos con base.

```
// Base
@import 'base/base';
@import 'base/reset';
@import 'base/typography';
```

Sólo falta layout.

```
// Layout
@import 'layout/header';
@import 'layout/main-section';
@import 'layout/forms';
```

Y finalmente components.

```
// Components
@import 'components/buttons';
@import 'components/inputs';
```

Los directorios pages y themes no los usaremos en este proyecto, pero es una buena idea mantener comentado el lugar que deberían ir estos dentro del manifiesto.

Probemos si no existen errores en nuestra implementación corriendo nuevamente en la terminal el comando `sass --watch`, con el nuevo input y output.

Volvamos al directorio raíz del proyecto y luego corramos el comando.

```
cd ../../..
sass --watch assets/sass/main.scss:assets/css/main.css
Sass is watching for changes. Press Ctrl-C to stop.
```

Bien, con esto hemos terminado de crear y configurar nuestro proyecto con Sass.