

Flujo de trabajo _

Parte I

Proceso de crear un sitio web



**Flujo para crear
interfaces de
usuario que
posteriormente
traduciremos a
código**

Estrategia

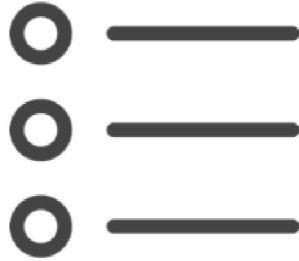


ESTRATEGIA

¿Qué queremos obtener del sitio?

¿Qué quieren nuestros usuarios?

Alcance

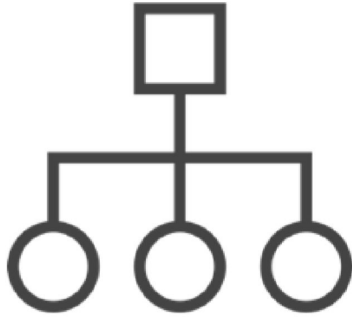


ALCANCE

¿Qué funcionalidades debe incluir el sitio?

¿Qué contenido requiere el sitio?

Estructura



ESTRUCTURA

¿Cómo organizar el contenido?

¿Cómo navegará el usuario?

Esqueleto



ESQUELETO

¿Qué componentes permitirán
que las personas usen el sitio?

Diseño Visual



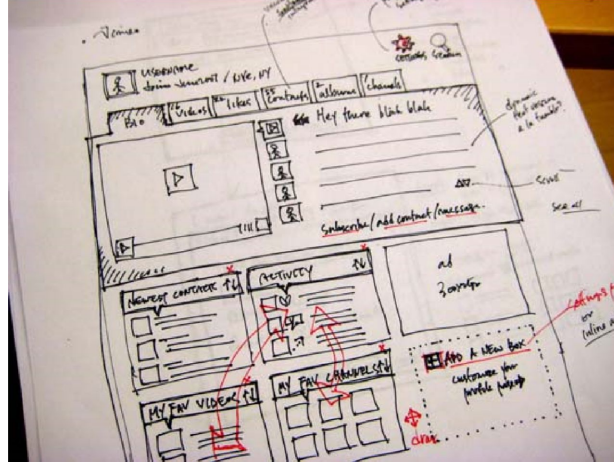
DISEÑO VISUAL

¿Cómo se verá la página terminada?

Representaciones visuales



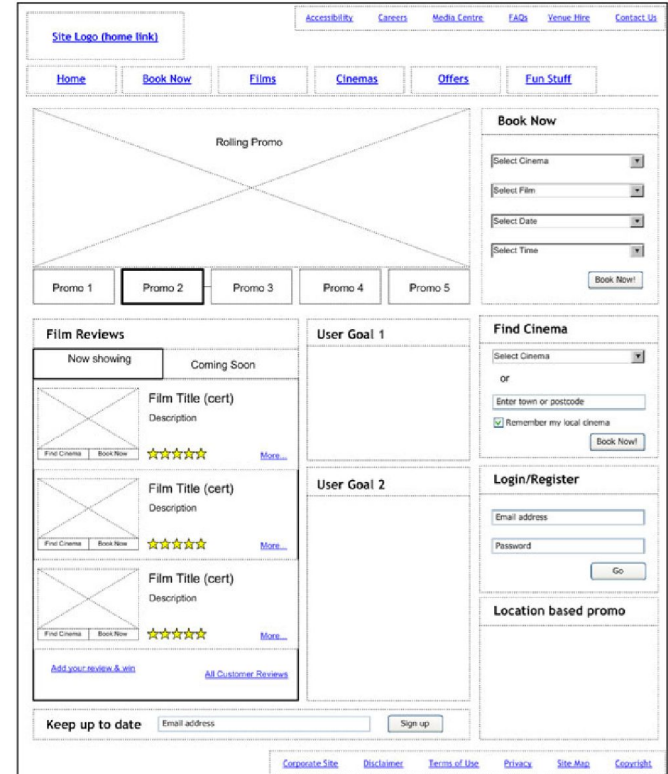
Sketch



- Es la representación en papel de una página web o aplicación.
- Es la forma más rápida de visualizar una página web.

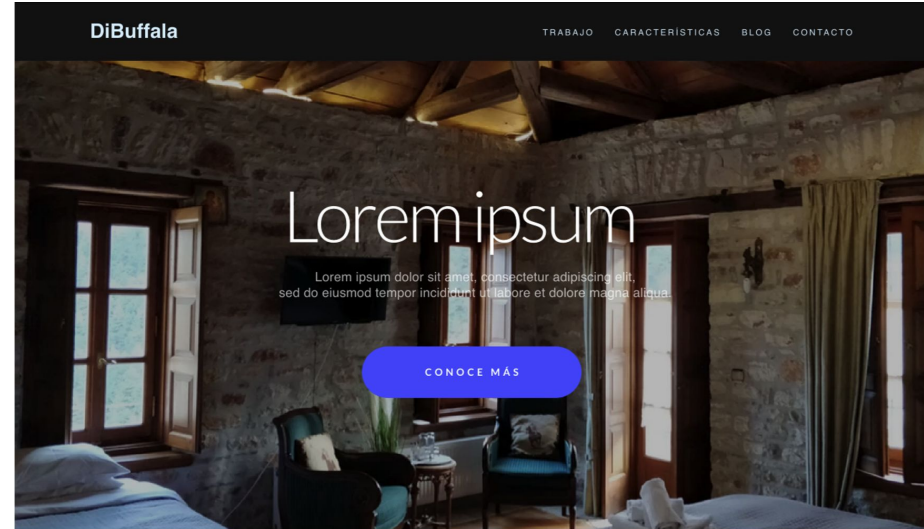
Wireframe

- Es el equivalente a un esqueleto de una web o aplicación.
- Es utilizado para describir la estructura, contenidos y características de una web o aplicación.
- No contiene diseño.
- Es generalmente creado por programas destinados para este tipo de representación (por ejemplo: Balsamiq, Axure R, entre otros)



Mockup

- Representación con diseño de interfaz.
- Reúne todas las características de un wireframe, con las características propias de un diseño visual (colores, fuentes, imágenes, logos, iconos, entre otros).
- Se crean utilizando diversos softwares como por ejemplo photoshop, illustrator, sketch, adobe XD, FIGMA, entre otros.



Prototipo

- Ofrece una representación de alta fidelidad de una página web o aplicación.
- Es un mockup enriquecido de la experiencia del usuario como interacciones y animaciones.

Diseñadores de experiencia de usuarios (UX)

- Investigar la conducta del usuario.
- Analizar los resultados de la investigación del usuario.
- Informar a los miembros del equipo sus hallazgos.
- Monitorear el desarrollo del proyecto.
- Asegurar que los resultados sean implementados en el proyecto.

Diseñadores de Interfaces de usuario (UI)

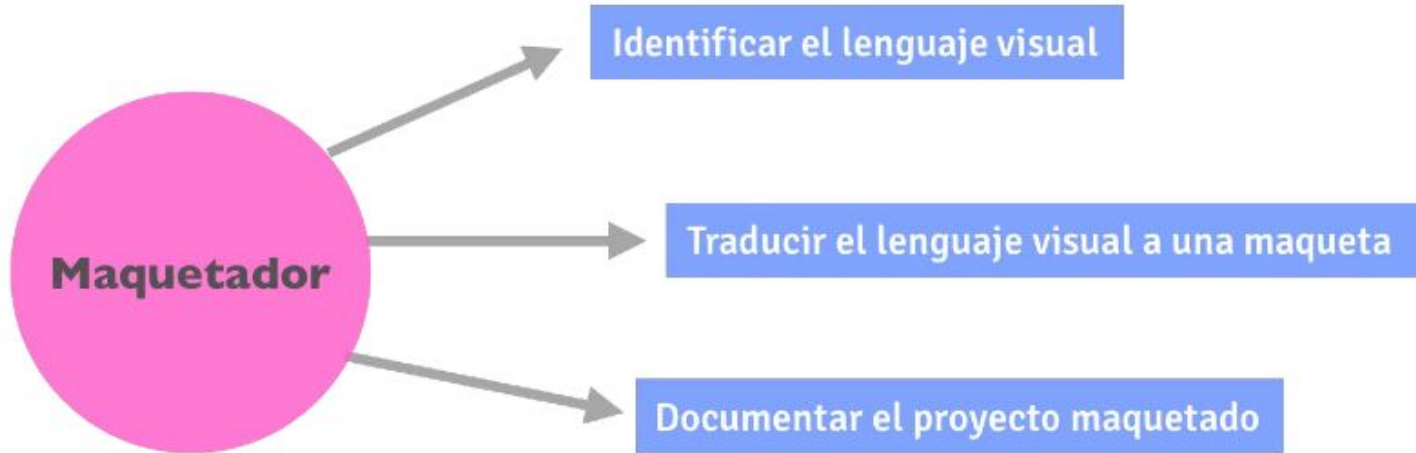
- Diseñar las interfaces con las cuales el usuario va a interactuar.
 - Es la suma de:
 - Arquitectura de la información
 - Elementos visuales de la página
 - Patrones de interacción

Maquetación



Acción de traducir representaciones visuales de manera fiel en código HTML, CSS y JavaScript. Deberá contener todas las características visuales y funcionales reunidas por los diseñadores UX/UI.

Rol del maquettador



Guía de estilos

Logos

Colors

Typography

Iconography

Illustration

Buttons/Links

Input

Navigation

Progress

Exercises

Colors

We want to make education fun, so we use a wide range of vibrant and juicy colors.

<div>KIWI 7AC70C</div> <div>8EE000</div> <div>BFF199</div>	<div>BLUEBERRY 1CB0F6</div> <div>14D4F4</div> <div>BCE9FF</div>	<div>RASPBERRY D33131</div> <div>E53B3B</div> <div>FF9797</div>	<div>BANANA FFB020</div> <div>FFC300</div> <div>FFF277</div>
<div>PUMPKIN FA811B</div> <div>FF9400</div> <div>FFCF7A</div>	<div>JELLY 8549BA</div> <div>A560EB</div> <div>CA94FF</div>	<div>COCOA 864E1B</div> <div>A86425</div> <div>DEAC5F</div>	<div>LICORICE 4C4C4C</div> <div>777777</div> <div>AFAFAF</div>
<div>CLOUD CFCFCF</div> <div>DDDDDD</div>	<div>CREME FFFFFF</div>		

Traducir el lenguaje visual a una maqueta

- Separar los elementos de una interfaz en pequeñas piezas de manera que sea más fácil construir estas piezas en código.
- Crear un sistema o seguir una metodología de trabajo que nos ayude a organizar nuestros proyectos.
- Usar herramientas que nos ayuden a trabajar de manera consistente en la construcción de nuestras maquetas.
- Seguir buenas prácticas al escribir código HTML y CSS a modo de hacer fácil la lectura y mantención de nuestras maquetas por terceras personas.



Documentar el proyecto maquetado

- Descripción de los elementos incluidos dentro del proyecto.
- Presentar una guía con recomendaciones relacionadas a la estructura y estilos aplicados.

Sass

- No se debe aprender ninguna sintaxis nueva y puedes seguir usando los archivos que ya utilizas.
- Nos permite retomar con facilidad proyectos de gran magnitud reduciendo el tiempo de creación y mantención de css, es limpio, fácil y con menos css en una construcción de programación.
- Organizado y escalable para proyectos personales como colaborativos, conteniendo menos código.
- Compatible con todas las versiones de CSS.
- Para mantenerse flexible y compatible con versiones futuras de CSS, Sass tiene soporte general que cubre casi todas las reglas de forma predeterminada.
- Incluye varias funciones que permite manipular con facilidad los colores, texto y otros valores.

Syntactically Awesome Style Sheets

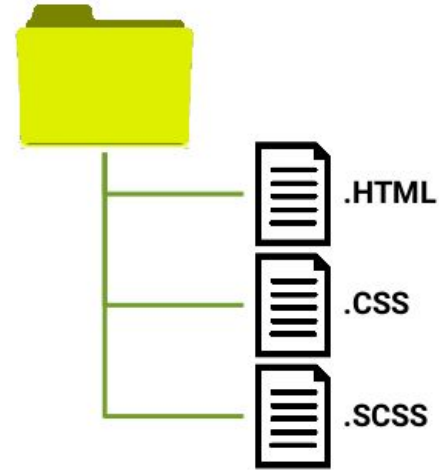
¿Cómo instalar Sass?



[https://marketplace.visualstudio.com/items?
itemName=glenn2223.live-sass](https://marketplace.visualstudio.com/items?itemName=glenn2223.live-sass)

Estructura básica de un proyecto con Sass

1. Crear el directorio raíz del proyecto.
2. Dentro de la carpeta creada anteriormente:
 - a. Crear archivo HTML
 - b. Crear archivo CSS
 - c. Agregar un archivo compatible con Sass



Sintaxis

Existen dos tipos de sintaxis para escribir su código:



.SASS



.SCSS

Variables

Pueden ser usadas a lo largo de todo el proyecto.

```
//SCSS
$font-stack: helvetica, sans-serif;
$primary-color: #333;
body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

```
//CSS
body {
  font: 100% Helvetica, sans-serif;
  color: #333
}
```

Anidación

Permite anidar selectores, evitando repetir los prefijos coincidentes en los selectores CSS.

```
//SCSS
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }
  li { display: inline-block; }
  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

```
//CSS
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav li {
  display: inline-block;
}
nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

Selector &

Ahorra escribir un selector en reiteradas ocasiones. Es un selector que llama al mismo selector.

```
.accordion {  
  max-width: 600px;  
  margin: 4rem auto;  
  width: 90%;  
  font-family: "Raleway", sans-serif;  
  background: #f4f4f4;  
  &__copy {  
    display: none;  
    padding: 1rem 1.5rem 2rem 1.5rem;  
    color: gray;  
    line-height: 1.6;  
    font-size: 14px;  
    font-weight: 500;  
    &--open {  
      display: block;  
    }  
  }  
}
```

Extender

```
/* This CSS will print because %message-shared is extended. */
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}
// This CSS will not print because %equal-heights is never extended.
%equal-heights {
  display: flex;
  flex-wrap: wrap;
}
.message {
  @extend %message-shared;
}
.success {
  @extend %message-shared;
  border-color: green;
}
.error {
  @extend %message-shared;
  border-color: red;
}
.warning {
  @extend %message-shared;
  border-color: yellow;
}
```

Mixin

```
//SCSS
@mixin transform($property) {
  -webkit-transform: $property;
  -ms-transform: $property;
}
.box { @include transform(rotate(30deg)); }

//CSS
.box {
  -webkit-transform: rotate(30deg);
  -ms-transform: rotate(30deg);
  transform: rotate(30deg);
}
```

¿Por qué estructurar proyectos con Sass?

- Orden
- Mantenable
- Separación lógica visual
- Sass posee características especiales para organizar proyectos

Características de Sass

Archivos parciales

- Archivos que contienen pequeñas partes de código CSS.
- Es cualquier archivo con un guión bajo.
- Se deben importar a otro archivo para generarlo en CSS.

Manifiestos

- Archivo que contiene un grupo de archivos importados que juntos forman una unidad.
- Administra todos los archivos parciales importados y los ensambla en una sola hoja de estilo.
- Los parciales son importados al manifiesto utilizando la propiedad `@import`.

Características de Sass

Import

- Extensión de la regla `@import` la cual puede importar archivos SCSS y Sass.
- Todos los archivos que son importados con `@import` son combinados en el único archivo CSS.
- Esta directiva es crucial cuando utilizamos archivos de manifiesto.

Comentarios

- Soporta dos tipos de comentarios de múltiples líneas (`/**/`) y de línea simple (`//`).
- Los comentarios de línea simple solo se verán en Sass.
- Utilizar comentarios es una buena implementación para documentar, definir, ordenar y declarar el código en un archivo Sass.

Crear un proyecto en Sass

Patrón 7-1

```
sass/
|
|- abstracts/
|   |- _variables.scss    # Sass Variables
|   |- _functions.scss    # Sass Functions
|   ...                  # etc..
|
|- base/
|   |- _reset.scss        # Reset/normalize
|   |- _typography.scss   # Typography rules
|   ...                  # Etc.
|
|- components/
|   |- _buttons.scss       # Buttons
|   |- _carousel.scss     # Carousel
|   ...                  # Etc.
```

```
|
|- layout/
|   |- _navigation.scss   # Navigation
|   |- _grid.scss        # Grid system
|   ...                  # Etc.
|
|- pages/
|   |- _home.scss         # Home specific styles
|   |- _contact.scss     # Contact specific styles
|   ...                  # Etc.
|
|- themes/
|   |- _theme.scss        # Default theme
|   |- _admin.scss       # Admin theme
|   ...                  # Etc.
|
|- vendors/
|   |- _bootstrap.scss    # Bootstrap
|   |- _jquery-ui.scss   # jQuery UI
|   ...                  # Etc.
|
|- main.scss              # Main Sass file
```

Orden de carga patrón 7-1

```
abstracts/  
vendors/  
base/  
layout/  
components/  
pages/  
themes/
```

```
// main.scss # Manifest File  
@import 'abstracts/variables';  
@import 'abstracts/functions';  
@import 'abstracts/mixins';  
@import 'abstracts/placeholders';  
@import 'vendors/bootstrap';  
@import 'vendors/jquery-ui';  
@import 'base/reset';  
@import 'base/typography';  
@import 'layout/navigation';  
@import 'layout/grid';  
@import 'layout/header';  
@import 'layout/footer';  
@import 'layout/sidebar';  
@import 'layout/forms';  
@import 'components/buttons';  
@import 'components/carousel';  
@import 'components/cover';  
@import 'components/dropdown';  
@import 'pages/home';  
@import 'pages/contact';  
@import 'themes/theme';  
@import 'themes/admin';
```

**Creando la arquitectura
del proyecto**

Layout

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

Componentes

```
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

```
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
```


Abstract

```
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
```

Importar parciales al manifiesto

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

BEM

Metodologías de Arquitectura CSS

- OOCSS
- SMACSS
- Atomic Design
- SUITCSS
- **BEM**

¿Qué es BEM?

- Nomenclatura utilizada para describir clases, evitando la repetición de estilos previniendo conflictos asociados a la cascada CSS.



Bloque
Elemento
Modificador

Bloques

```
<header class="header">
  <nav class="navigation">
    <ul>
      <li>Inicio</li>
      <li>Acerca de</li>
      <li>Blog</li>
    </ul>
  </nav>
</header>
```

Elementos

```
<header class="header">

  <nav class="navigation">
    <ul class="navigation__list">
      <li class="navigation__item">Inicio</li>
      <li class="navigation__item">Acerca de</li>
      <li class="navigation__item">Blog</li>
    </ul>
  </nav>
</header>
```

Modificador

Para bloques:

```
<footer class="footer footer--dark-theme">  
  <small class="footer__copyright">2018. Todos los derechos reservados</small>  
</footer>
```

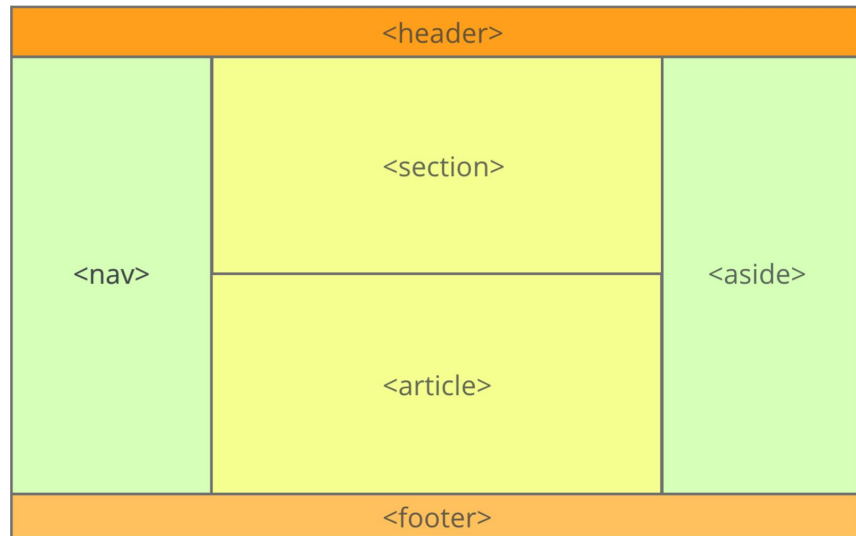
Para elementos:

```
<form class="form">  
  <div class="form__container">  
    <input type="text" class="form__input form__input--focused">  
  </div>  
</form>
```


Etiquetas semánticas

Etiquetas semánticas

Son elementos HTML que describen el significado semántico de un elemento en específico.



Header

```
<body>  
  <header class="header">  
  </header>  
</body>
```

Main

```
<main class="main">
```

```
</main>
```

Form

```
<form action="#" class="form">  
  
</form>
```

Variables Sass en un proyecto

Color

White #FFF
Platinum #E8E8E8
Gray #BABABA

Carmin pink #E94F37
Salmon #EF7F6D
Spanish pink #F7BFB6

Black olive #393E41
Granite gray #5D6163
Old silver #818486

Verdigris #448BA4
Medium aquamarine #66C7B4
Perl Aqua #88D3C5

```
// Color Scheme

// First Color Scheme
$white: #FFF;
$platinum: #E8E8E8;
$gray: #BABABA;
```

Fuentes

Typefaces

Playfair Display Bold

Open Sans Bold

Open Sans Semi Bold

Open Sans Regular

```
//Font Family  
$playfair: 'Playfair Display', serif;  
$open-sans: 'Open Sans', sans-serif;
```


Peso de tipografía

```
// Typography  
  
// Font Weight  
$regular: 400;  
$semi-bold: 600;  
$bold: 700;
```

Tamaño de fuente

My slave

130px — xxxx-large

My slave human

100px — xxx-large

My slave human didn't give me any food

30px — xx-large

My slave human didn't give me any food

20px — x-large

My slave human didn't give me any food

18px — large

My slave human didn't give me any food

16px — medium

My slave human didn't give me any food

14px — small

My slave human didn't give me any food

12px — x-small

```
// Font Sizes
```

```
$xxxx-large: 130px;
```

```
$xxx-large: 100px;
```

```
$xx-large: 30px;
```

```
$x-large: 20px;
```

```
$large: 18px;
```

```
$medium: 16px;
```

```
$small: 14px;
```

```
$x-small: 12px;
```

