**Workshop Guía**

# ¿Qué es una Landing Page?

En español pagina de aterrizaje, es una herramienta diseñada para generar acción por parte del usuario digital, el que por un tiempo muy limitado se convierte en usuario de tu web y que como marca queremos que pase de visitante a contacto.

Sirve para generar contactos o ventas, o ambas.

# Crear una landing page

Es sencillo pero complicado a la vez. Requiere mucha practica, prueba y error y de mucho testing.

Una landing page consta de:

1. Un titular corto, directo y sencillo.
2. Un diseño limpio y alineado y muy exacto.
3. Un menú principal y Footer minimizados.
4. Textos concisos y breves.

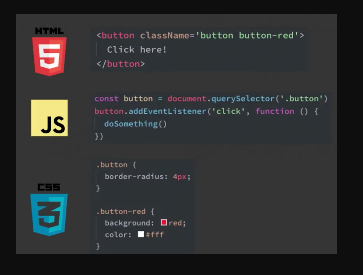
Normalmente el desarrollo de una Landing Page consta de 3 herramientas

1. HTML para el maquetado
2. CSS para los estilos
3. JS para eventos, controladores, funciones, y todo el adicional mostrado

Aunque existen otras herramientas usa en el proceso de desarrollo

1. Línea de comandos / terminal, para cambiar de proyectos
2. Manejanejador de versiones
3. Preprocesadores de CSS
4. Framework
5. Librerias

# Para este proyecto



Esta web estará centralizada en HTML al cual le adicionaremos CSS y JS, obtener el proyecto final usaremos algunas otras herramientas.

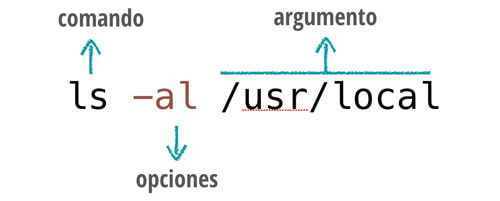
* Terminal o línea de comandos del sistema operativo para hacer movimientos en carpetas y archivos.
* Git y GitHub para el control de versiones
* SASS para poner estilos (para organizar mejor nuestros estilos)
* Babel para leer JS moderno (para poder usar las mejoras mas actuales de JS)
* Webpack para interpretar Babel y SASS, y obtener el entorno de desarrollo
* Node y NPM para descargar Webpack y sus paquetes de apoyo.

# Command Line

* ¿Qué es una línea de comandos?

Es una aplicación basada en texto para ver, manejar y manipular archivos en tu ordenador

* ¿Para qué sirve una línea de comandos?
  + Crear una secuencia de comandos para apagar siempre ciertos servicios o programas cuando el primer usuario inicia sesión.
  + Tener acceso directo a las funciones del sistema operativo.
* Anatomía de una línea de comandos.
* Un nombre con el que se invoca el comando.
* Opciones que modifican el comportamiento del comando (Opcional).
* Argumentos sobre los que actúa el comando (Opcional).



* Mostrar contenido del directorio actual:

Dir / pwd

* Cambiar de directorio

cd NombreDirectorio

* Crear carpeta

md nombre

* Renombrar directorio

ren nombreviejo nombrenuevo

* Eliminar directorio vacio

rmdir nombrearchivo

* Eliminar directorio con archivos

rmdir nombre /s

* Crear archivos

copy con nombre.extencion

ctrl + z

enter

* Eliminar archivos

del nombre.extencion

# Git

* ¿Qué es Git?

Es un software de control de versiones.

* ¿Qué es un control de versiones?

Se define como control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

**Comandos básicos.**

**Crear repositorio**

Ejemplo para crear un repositorio desde la creación de una carpeta.

*mkdir hola*

mkdir es un comando para crear nuevos directorios

*cd hola*

cd nos permite entrar a la carpeta previamente creada

*touch hello.html*

touch crea un archivo

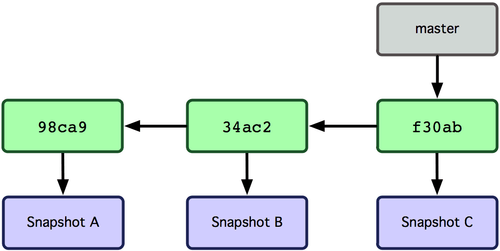
*git init* permite inicializar un repositorio

*git add* al mencionar este elemento vamos a agregar el archivo que se desea agregar o si se escribe *git add .* podemos agregar todos los elementos de ese directo al repositorio

*git commit -m “Prueba”* Permite agregar comentarios

* **¿Qué es una rama?**

Una rama Git es simplemente un apuntador móvil apuntando a una de esas confirmaciones. La rama por defecto de Git es la rama master. Con la primera confirmación de cambios que realicemos, se creará esta rama principal master apuntando a dicha confirmación.



**Crear una rama en github**

*git branch* (nombre de la rama)

Entrar a una nueva rama

*git checkout* (nombre de la rama)

*git push* es usado para subir el contenido de un repositorio local a un repositorio remoto

*git pull* es usado para actualizar el contenido de un repositorio local desde un repositorio remoto

git merge (nombre branch) hace una mezcla entre dos ramas , la dirección del merge sería entre el Branch que indiquemos en el comando y el Branch donde estemos ubicados

git status indica el estado en el que se encuentra un repositorio

git clone (dirección URL) clona un proyecto un proyecto de git

# Github

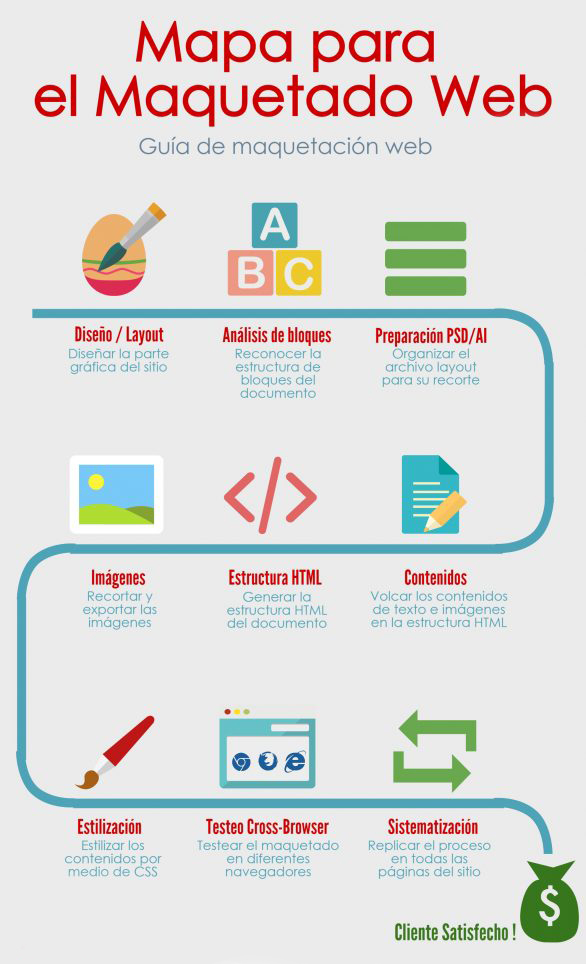
¿Qué es GitHub?

GitHub es un sistema de gestión de proyectos y control de versiones de código, así como una plataforma de red social diseñada para desarrolladores.

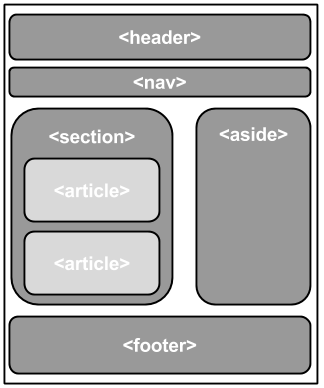
# Html

¿Qué es HTML?

es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet y no debe confundirse con un lenguaje de programacion. Sus siglas corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto.



**Estructura básica de HTML5**



# Sass

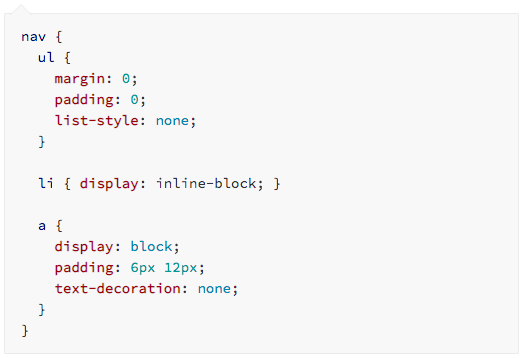
¿Qué es SASS?

Un preprocesador de CSS es una herramienta que nos permite escribir pseudo-código CSS que luego será convertido a CSS real.

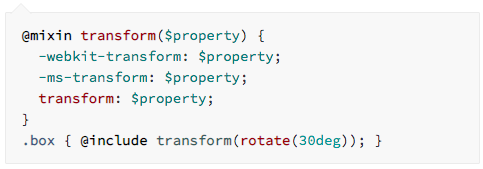
Variables



Nesting



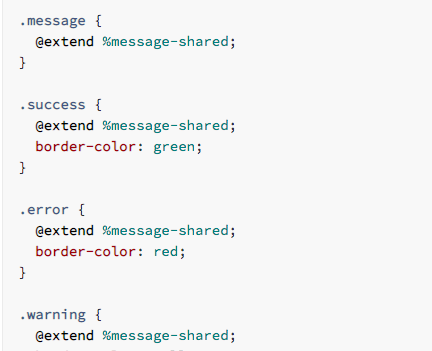
Mixins



Operatiors



Extend/inheritance

# JavaScript

Crear pequeños programas que luego son insertados en una página web.

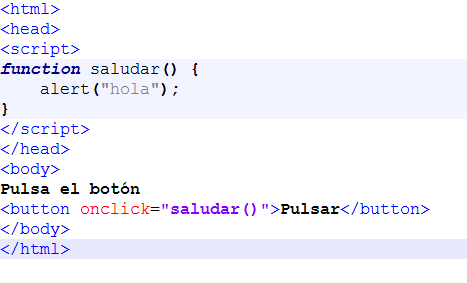
Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios.

**¿Para qué se utiliza JavaScript?**

Javascript sirve para que en una página web se consiga un mayor índice de interactividad con los usuarios y, por tanto, la experiencia de estos sea mucho mejor y más enriquecida.

Facilitar la comunicación y, además, otorgar todas las mejoras posibles sin apenas incidir en el peso de la página.

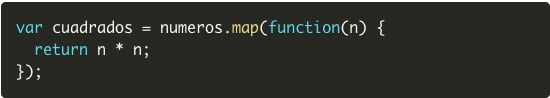
**Ejemplo código de JavaScript**



Con Babel se pueden agregan algunas funcionalidades interesantes que aun no son soportadas por el navegador.

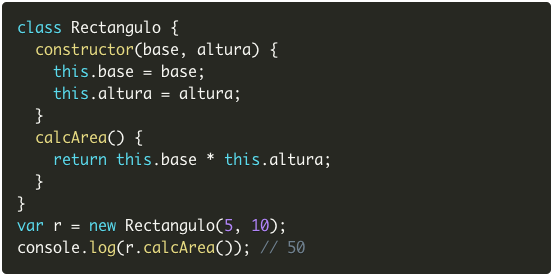
Arrow Functions

Aquí tomamos un array de números y con map transformamos el mismo array guardando estos cambios en la variable

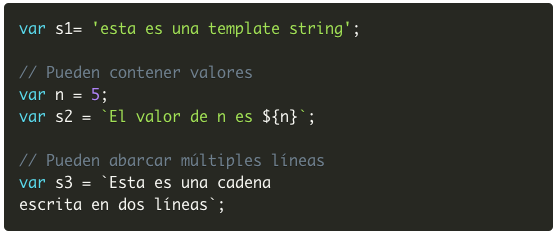




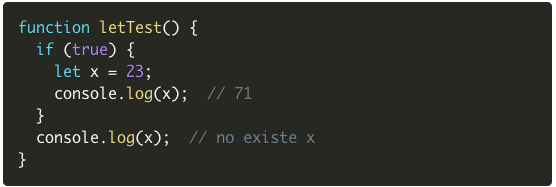
Clases



Template Strings

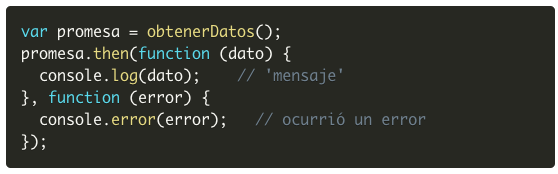


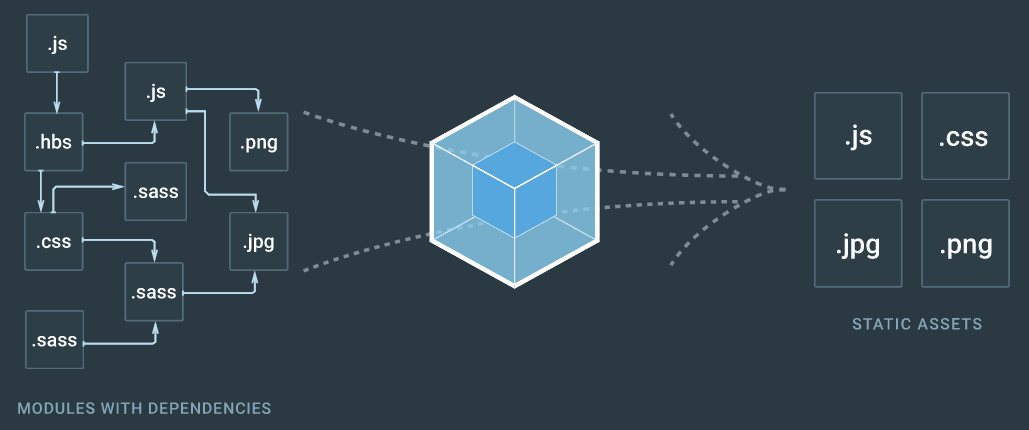
Let y Const





Promesas





Webpack es un empaquetador de módulos (bundler) con extenciones

* Gestión de dependencias
* Ejecución de tareas
* Conversión de formatos (Para SASS)
* Servidor de desarrollo (Para ver los cambios en automático)
* Carga y uso de módulos de todo tipo (para interpretar con Babel)
* Y mas…

Es un buen apoyo cuando desarrollas con la filosofía modular, es decir separando el código en módulos que luego se utilizaran como dependencias de otros módulos ya sea en JS, CSS o HTML, con apoyo de herramientas como Frameworks, preprocesadores, etc.

Para utilizar Webpack primero debemos instalarlo desde NodeJS, porque en si es un paquete de node,

Es un entorno de ejecución de JS, construido en el motor de JS llamado Chrome’s V8. Es decir que permite ejecutar un Javascript en la PC. Es como un compilador. Aun asi tiene muchas herramientas extras guardadas en NPM de donde se les puede llamar. Sabiendo esto podemos empezar a descargar estas herramientas y comenzar a utilizarlas.