

# ENVIROMAP

MAPA DE ENTORNO

DESARROLLO WEB - 2022

## DISEÑO

Orientado a la Programación

Publicidad

Branding

Ilustración

Fotografía

Ps

Ai

Id

## UX/UI

user experience / user interface

1 Bocetado

2 Wireframe

3 Prototipado

4 MockUps

Zeplin

Xd Adobe Xd

Sketch

Figma

**User Interface:**  
Interfaces de Usuario

Diseño de interfaces que permitan transmitir las experiencias buscadas.

En el **desarrollo web** crean las vistas y casos de uso de desktop y responsive de cada sitio o aplicación.

**User Experience:**  
Experiencia de Usuario

**Identifica y estudia** patrones de uso de las interfaces con el fin de aportar experiencia de uso confortables para los usuarios y obtener resultados esperados por su parte.

# FRONTEND

La cara visible de nuestro trabajo

## Estructura

HTML

Lenguaje de Marcado de etiquetas o “*Hiper Text Markup Language*”.

Es la **sintaxis** (idioma) con el que a través de etiquetas como `<p>Hola Mundo</p>` le marcamos al navegador lo que queremos mostrar, por ejemplo como en este caso, un párrafo con el texto **Hola Mundo**.

es potenciado por

## Template Engines

Dinamismo en nuestro HTML

Ejs Handlebars

Nunjucks Pug

Existen muchos más y para distintos lenguajes de programación.

Son “*motores de plantilla*” que nos permiten hacer **magia** en el mismo archivo donde **marcamos nuestro HTML**, como por ejemplo: **Iterar** (repetir) elementos iguales con distinto texto o **condicionar** si un elemento se muestra o no.

Usar Template Engines facilita la tarea de crear nuestros sitios o aplicaciones web. Al finalizar la etapa de **Desarrollo** ejecutamos un comando que nos entregará archivos .html estáticos listos para subir a nuestro hosting.

Para utilizar estas herramientas es preciso montar servidores locales que puedan interpretar nuestros archivos para mostrar en vivo lo que vamos creando.

## Estilos y Animaciones

CSS

Hojas de Estilo en Cascada o “*Cascading Style Sheet*”.

Es la **sintaxis** (idioma) con la que le indicamos al navegador como queremos que muestre nuestro HTML.

es potenciado por

## Frameworks

Trabajo rápido y eficiente.

Bootstrap

tailwindcss

Componentes pre-diseñados.

Animaciones y efectos.

Estilos consistentes entre si.

Utilidades para layouts.

Se basan en “paquetes” de clases CSS con estilos y comportamientos preestablecidos y listos para utilizar en nuestro HTML. Reduce drásticamente nuestro uso de CSS nativo.

## Preprocesadores

Magia extra, mejores resultados.

Sass

PostCSS

A diferencia de los Frameworks (que nos dan CSS “pre-fabricado”) los preprocesadores son mejoradores de CSS que nos brindan herramientas para trabajar más rápido y de forma más eficiente, con la posibilidad de utilizar funciones, condicionales y otros recursos en nuestro desarrollo.

Al igual que los Template Engines en HTML, estos necesitan ser interpretados o traducidos a CSS tradicional y consolidados en un único archivo estático.

## Dinamismo

JAVASCRIPT

Lenguaje de **programación** por excelencia en la web. Es el lenguaje que **entienden y utilizan** los navegadores más populares.

En el Frontend lo utilizamos para escalar nuestros sitios estáticos realizados con **HTML** y **CSS** a través de efectos, comportamientos e interacción con el usuario.

es potenciado por

Typescript

Un superset de Javascript que le otorga al lenguaje (entre otras cosas) la posibilidad de trabajar con tipado estático y validar la información trabajada.

## Frameworks

Todo en uno y a trabajar.

Angular

React

Vue

Svelte

Nos dan una estructura de carpetas, estándares de trabajo y paquetes de código javascript con soluciones pre-fabricadas a problemas comunes. Se orientan al desarrollo de Aplicaciones Web (Webapps) y de Aplicaciones de una sola Página (SPA's).

## Static Site Generators

Al igual que varios lenguajes de programación (entre ellos Javascript, PHP, Ruby, GO), algunos de los Frameworks mencionados poseen sus propios frameworks que extienden el uso de estas herramientas para crear sitios web estáticos o dinámicos de más de una sola página, por ejemplo:

JavaScript

Eleventy

Ruby

Jekyll

Vue

NuxtJS

GO

Hugo

React

Gatsby

Next.js

Los Generadores de Sitios Estáticos (SSG) usan Template Engines por defecto.

# BACKEND

Lógica detrás de escena

Cuando deseamos que nuestros sitios web contengan **dinamismo**, donde el usuario **interactúa** con nuestro sitio lo que exige que exista un “*intercambio de información*”, como un **registro** o **login** de usuario, una **tienda en línea** o una **aplicación web**, precisamos de lenguajes de programación que implementen la lógica detrás de ese funcionamiento.

## LENGUAJES

Con el avance de herramientas que permiten la interacción multilenguaje y multiplataforma, no existe casi limitación a la hora de elegir un lenguaje para desarrollarlos en backend destinados a la web.

## BASE DE DATOS

Son las herramientas que nos permiten guardar los datos de nuestros sitios web dinámicos.

mongoDB

Postgre

MySQL

# UTILIDADES

**Node Package Manager:** gestor de paquetes de NodeJS que nos permite descargar paquetes de código con soluciones a problemas comunes a través de la consola.

**Webpack:** empaquetador de código. Es el encargado de traducir nuestros template engines y preprocesadores CSS para entregarnos archivos estáticos.

**git** Gestor de versiones de nuestros proyectos. Identifica nuestros cambios y nos permite revertirlos.

**GitHub** Herramienta remota para guardar y administrar nuestro código que hemos versionado previamente con GIT.