

**PREGRADO**



UNIDAD 2 | FRONTEND WEB APPLICATIONS

# **WEB FRONTED APPLICATIONS**

Al finalizar la unidad, el estudiante desarrolla y comunica una solución de aplicación frontend integrada con RESTful web services con una Arquitectura Orientada a Servicios en un ambiente de desarrollo ágil e inclusivo

---

# AGENDA

JAVASCRIPT

WEB FRAMEWORKS



# Introducción

El **desarrollo web moderno** es hoy muy dinámico, con mucha experiencia enriquecida y con mucha demanda de programadores con buena habilidad.

**JavaScript** es un lenguaje de programación muy fácil de aprender, nos permite crear aplicaciones web interactivas, con cambios en los contenidos en tiempo real, crear cookies, validación de formularios, efectos personalizados, reactivo a las acciones del usuario, y muchas características más.

**JavaScript** prácticamente se encuentra en todas las páginas web por las que navega, y hace que estas cobren vida para usted.

## ¿Quienes lo usan?

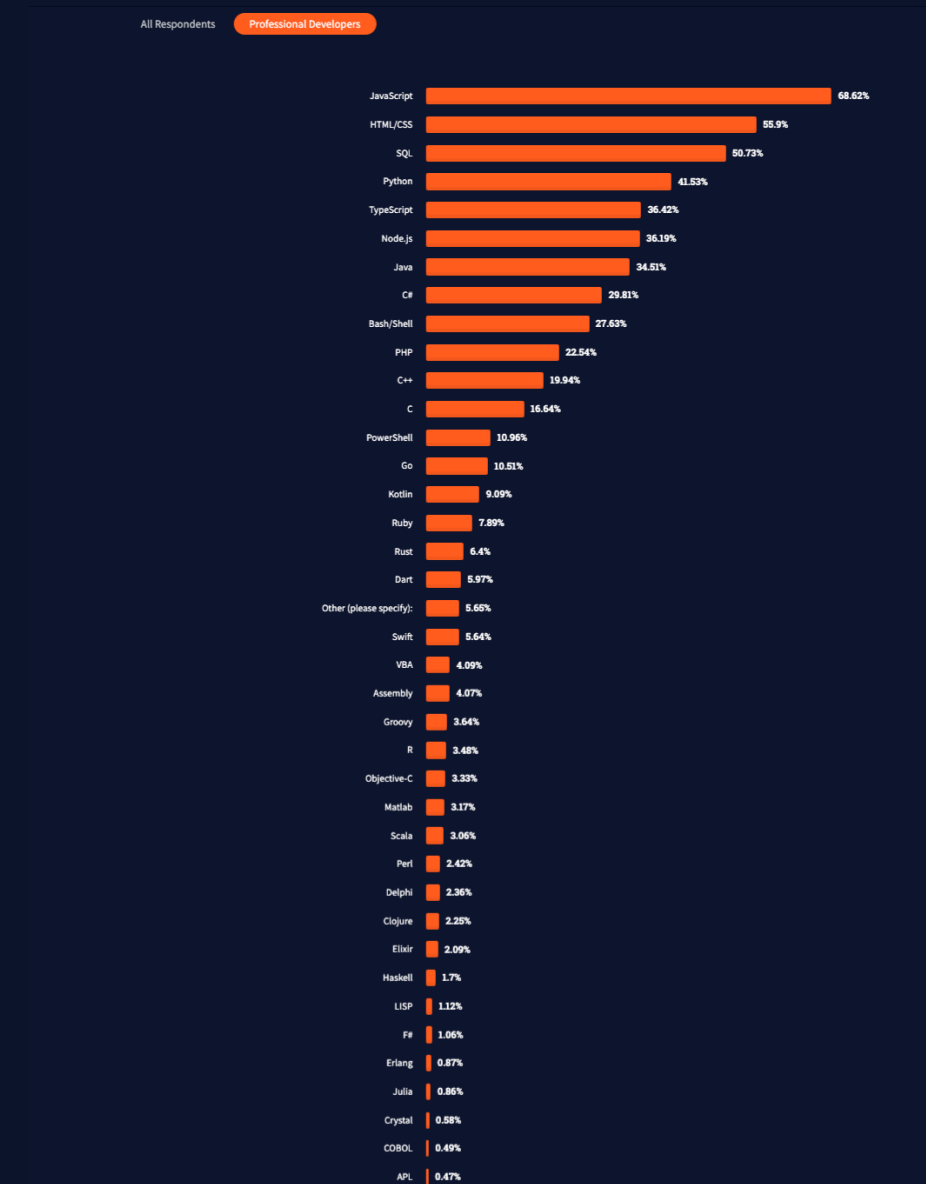
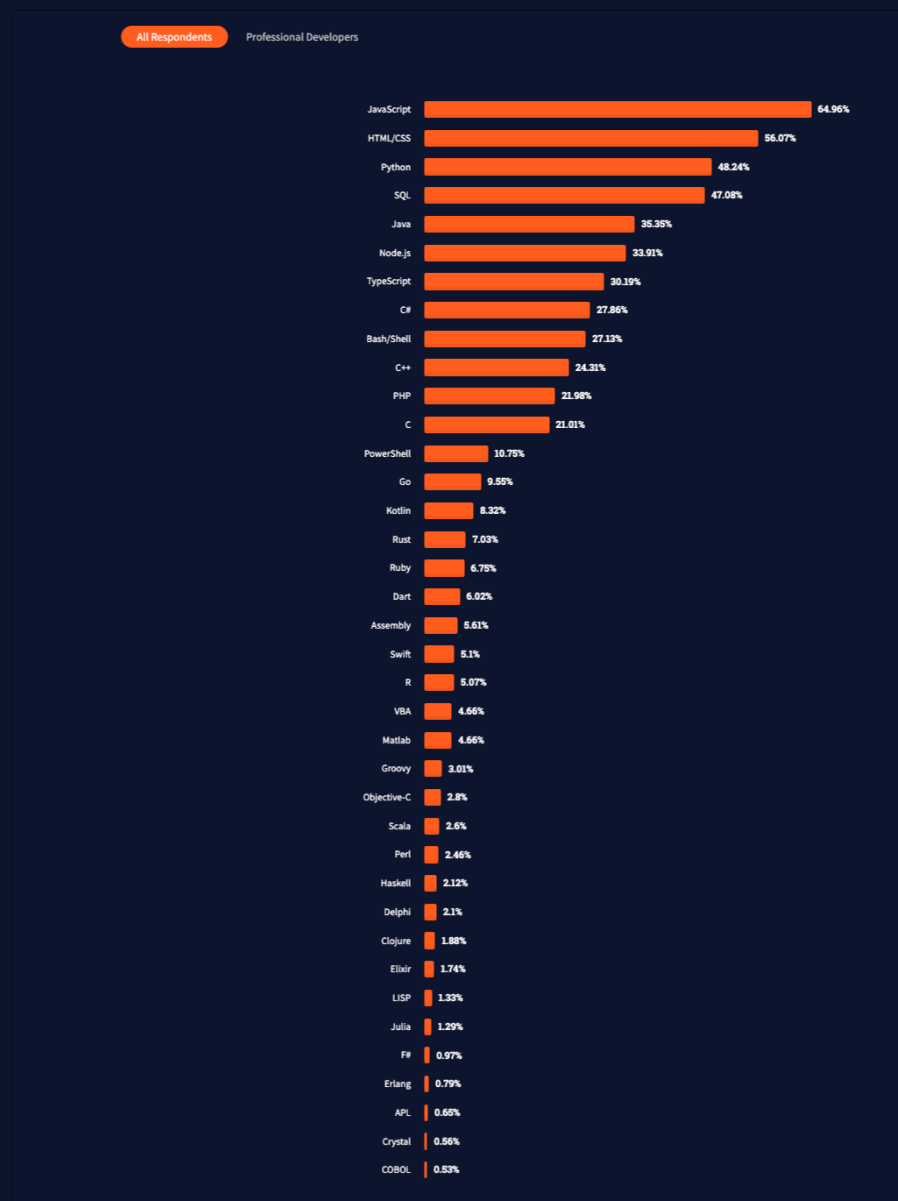
Basados en las tecnologías analizadas por StackShare.io, JS cuenta con una amplia lista de empresas que lo utilizan para brindar servicios a millones de usuarios todos los días.

Dentro de los más relevantes, podemos destacar: *Airbnb, Instagram, Reddit, Slack, StackShare, eBay, Coursera y Asana.*

Fuente: <https://www.stackshare.io/javascript>

# Introducción – Demanda

La demanda actual de JS o Node.js (En el lado del servidor) representa una de las tecnologías con más relevancia para el ecosistema del desarrollo web según datos del StackOverflow *Overflow developers survey report 2021*



Fuente: <https://insights.stackoverflow.com/survey/2021>

# ECMAScript

*JavaScript* es un lenguaje de programación que se ha vuelto tan popular en los últimos años que ha necesitado mejoras del lenguaje para adecuarse a los nuevos tiempos.

**ECMAScript** es la especificación del lenguaje JavaScript, en el caso de ECMAScript 6 o **ES6** (lanzado en 2015) es una de las evoluciones mas importantes del lenguaje con mejoras que lo hacen legible, intuitivo y sólido.

# Compatibilidad de ES6

Actualmente ES6 es soportado por la mayoría de los navegadores: Chrome 51+, Firefox 54+, Safari 10+, Edge 15+, por lo que se puede usar de forma nativa sin problemas.

# ECMAScript 2015 (ES6) OTHER

Usage % of all users ?  
Global 94.03% + 3.54% = 97.58%

Support for the ECMAScript 2015 specification. Features include Promises, Modules, Classes, Template Literals, Arrow Functions, Let and Const, Default Parameters, Generators, Destructuring Assignment, Rest & Spread, Map/Set & WeakMap/WeakSet and many more.

Current aligned Usage relative Date relative Apply filters Show all ?

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser
	<sup>2</sup> 12-14	2-5	4-20	3.1-7	10-12.1	3.2-6.1									
	<sup>2 3</sup> 15-18	6-53	21-50	7.1-9.1	15-37	7-9.3		2.1-4.3					4		
6-10	<sup>2</sup> 79-80	<sup>2</sup> 54-75	<sup>2</sup> 51-80	10-13	<sup>2</sup> 38-67	10-13.3		4.4-4.4.4	12-12.1				<sup>2</sup> 5-10.1		
<sup>1 2</sup> 11	<sup>2</sup> 81	<sup>2</sup> 76	<sup>2</sup> 81	13.1	<sup>2</sup> 68	13.4	all	<sup>2</sup> 81	<sup>2</sup> 46	<sup>2</sup> 81	<sup>2</sup> 68	<sup>2</sup> 12.12	<sup>2</sup> 11.1	<sup>2</sup> 1.2	7.1
		<sup>2</sup> 77-78	<sup>2</sup> 83-85	TP											

Fuente: <https://caniuse.com/#feat=es6>

Notes Sub-features (10) Known issues (0) Resources (4) Feedback



## ES6- Tipo de Variables

Se puede declarar tres tipos de variables con las siguientes palabras clave:

- **let** para declarar variables que existen únicamente dentro de un ámbito.
- **var** para declarar variables globales o que existen en todos los ámbitos.
- **const** para declarar constantes es decir su valor no se puede cambiar.

Tener en cuenta que un **ámbito** (*scope*): es un bloque de código dentro de una estructura de control (if, while, case), estructura de bucle (for, foreach, while), funciones o clases.

# ES6- Tipo de Variables

```
1  const MAX_CHARS = 8;
2  let x = 1;
3
4  if (x === 1) {
5      let x = 2;
6      var y = 5;
7      console.log('x: ', x);
8      console.log('y: ', y);
9  }
10
11 console.log('x: ', x);
12 console.log('y: ', y);
```

# ES6- Arrow functions

En ES6 es posible declarar funciones utilizando `=>` o llamada *arrow function*.

```
1 // declare function
2 let circleArea = (radio) => {
3   return 2 * Math.PI * radio;
4 };
5
6 // use function
7 console.log('circleArea: ', circleArea(5));
```

## ES6- Arrow functions

Los arrow functions tienen algunas ventajas: si la función acepta **un sólo parámetro**, **no hace falta** los paréntesis iniciales, si la función **devuelve un único valor** se puede prescindir de las llaves y de la palabra reservada **return**.

```
1  // expanded version
2  let squareArea = (width) => {
3      return width * width;
4  };
5
6  // simplified version
7  let squareArea = width => width * width;
```

## ES6- Parametros opcionales

Es posible declarar funciones con parámetros **opcionales**, esto permite crear funciones que se comportan de forma diferente de acuerdo a la cantidad de parámetros que reciben.

```
1  // declare function
2  let sayHello = (name = 'World') => {
3    console.log('Hello ' + name);
4  };
5
6  sayHello();           // Hello World
7  sayHello('Kodetop'); // Hello Kodetop
```

# ES6- Clases

ES6 introduce la palabra clave **class** para la declaración de clases.

El método **constructor** permite inicializar la clase.

Esta es una mejora de sintaxis, antes las clases se declaraban con la palabra clave **function**, lo cual era poco claro.

```
1  // declare class
2  class Popup {
3    constructor() {
4      console.log('init popup');
5    }
6
7    open() {
8      console.log('open popup');
9    }
10
11   close() {
12     console.log('close popup');
13   }
14 }
15
16 // instance object
17 let popup = new Popup();
18
19 // use methods
20 popup.open();
21 popup.close();
```

# ES6- Módulo

Los **módulos** permiten agrupar el código en diferentes archivos de forma que se pueda separar la lógica o las funcionalidades para facilitar el mantenimiento.

Un módulo se ubica en un archivo **.js**, por defecto las variables y métodos declarados son privados y sólo se exponen los que son declarados con `export`.

```
1  // file: areas.js
2  let squareArea = (width) => {
3    return width * width;
4  };
5
6  let circleArea = (radio) => {
7    return 2 * Math.PI * radio;
8  };
9
10 // export methods
11 export { squareArea, circleArea };
```

## ES6- Módulo

Una vez declarado un módulo se puede importar con la palabra clave import.

```
1 // file main.js
2 import { squareArea } from './areas.js';
3
4 // use module
5 console.log( squareArea(5) );
```

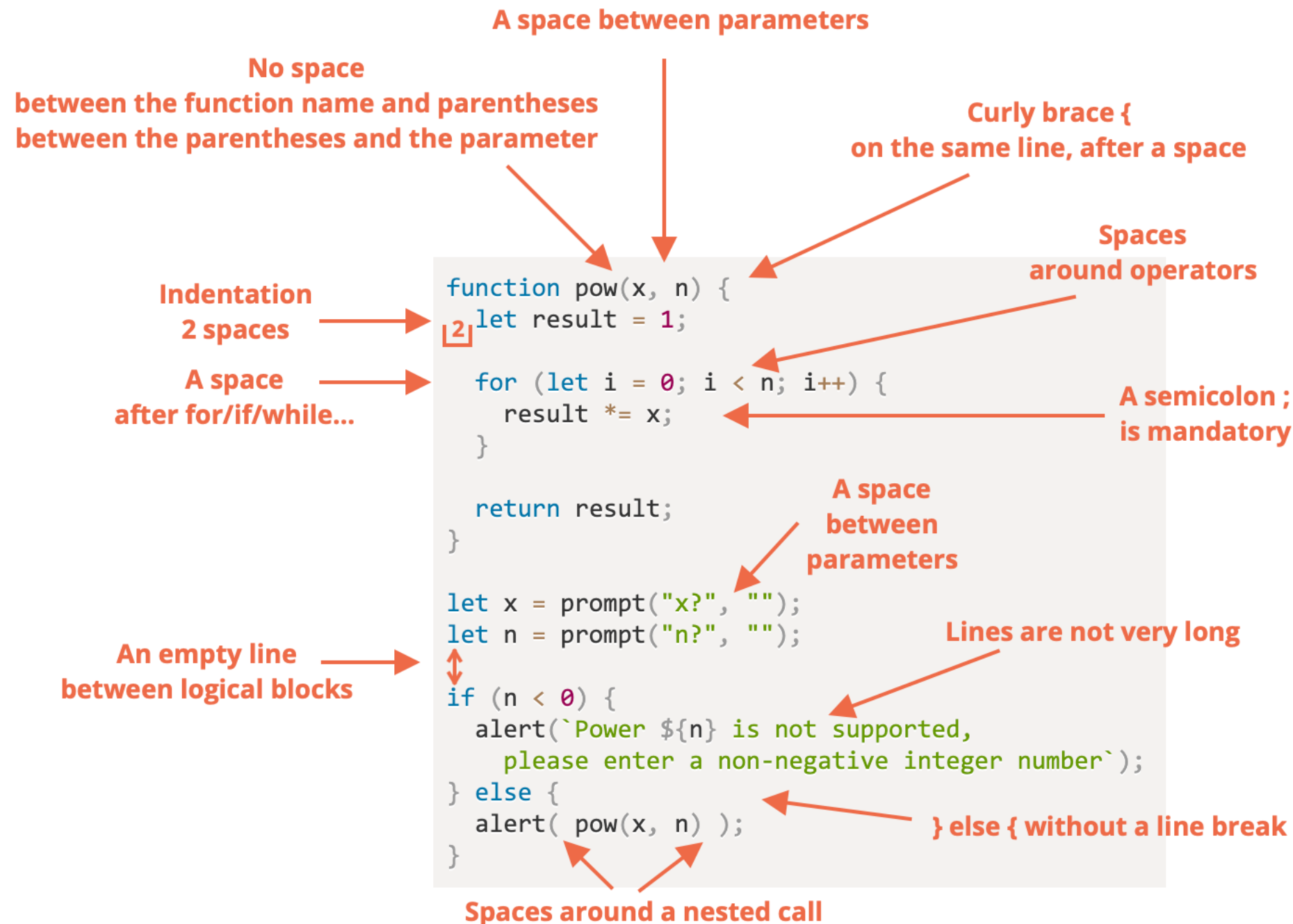
No es necesario incluir todos los archivos JavaScript en el HTML, sólo se incluye el archivo que utiliza los módulos pero utilizando el atributo type="module".

```
1 <script src="main.js" type="module"></script>
```



# Coding Style

Nuestro código debe ser correcto y entendible por las personas.



---

# AGENDA

JAVASCRIPT

WEB FRAMEWORKS



# Web Framework

Un framework es una herramienta de desarrollo web que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de bibliotecas y/o funcionalidades ya creadas.

Los framework hacen que el desarrollador no esté continuamente “reinventado la rueda” y se centre en el problema que quiere resolver y no en la implementación de funcionalidades que normalmente son de uso común y que ya están resueltas por otros.

# Angular Framework

**Angular** is a front-end framework that specializes in building rich Single-Page applications. It's able to build complete client-side applications.

Angular 1.x used JavaScript, but later releases adopted **TypeScript**, which is a superset of JavaScript.

Angular's main cons are its size compared to other frameworks, and the fact it's not SEO friendly by nature, though it can be SEO optimized.

Google developed angular, and Google, Microsoft, and Paypal use it.

# React Frontend Library

**React** is not a framework, it's a frontend library, but many developers consider it a framework and it's usually compared in that context.

React was the first to adopt the component-based architecture that Angular and Vue, and many other frameworks started to adopt later on.

React's virtual dom makes the dom-manipulation much faster, and it's quite easy to pick up, mainly thanks to its JSX syntax.

React could be used server-side or client-side.

It was developed and maintained by Facebook, and Facebook and Instagram use it.

Fuente : <https://hackr.io/blog/top-10-web-development-frameworks-in-2020>

# Vue.js Framework

**Vue.js** started as an individual project and quickly grew into becoming one of the most trending JS frameworks out there.

Vue it's a **progressive framework**, which means that if you have an existing project, you can adopt Vue for one portion of the project, and everything would work just fine.

It brings along the **component architecture** to play, and the Vue ecosystem can help you build complete frontend applications.

Some people are wary of using Vue since big companies like Facebook or Google do not support it, but that's quickly changing as big names are starting to invest in Vue.

Fuente : <https://hackr.io/blog/top-10-web-development-frameworks-in-2020>

# Ember Framework

**Ember** was named the best Javascript framework back in 2015.

Today, the Ember community is enormous, and it's ever-expanding, with new features, and releases added regularly.

Ember possesses the **two-way data binding** that Angular boasts, and it comes with a lot of features and components that you can use out of the box.

Google, Microsoft, Heroku, and Netflix use this framework often.

Ember revolves around the developer's productivity and attempts to maximize it by either eliminating the need for time-wasting activities or adopting some JS best practices in its core design.

Fuente : <https://hackr.io/blog/top-10-web-development-frameworks-in-2020>

# Backbone Framework

**Backbone** is an extremely light front-end framework that's fit for building rich Single-Page applications.

It follows an MV\* pattern and partly implements the MVC design.

Backbone only has one core dependency, which is the Underscore library, and it has a vibrant ecosystem, that when added to Mustache and Marionette, allows you to build complete client-side applications.



---

# RESUMEN

## Recordemos

JavaScript es uno de los lenguajes más utilizados dentro del ámbito de desarrollo de software.

Existen varias Bibliotecas y Frameworks sobre JavaScript, resaltando nombres como Angular, React, Ember y Vue.



---

# REFERENCIAS

Para profundizar

<https://www.ecma-international.org/ecma-262/6.0/>

<https://github.com/lukehoban/es6features>

<https://hackr.io/blog/top-10-web-development-frameworks-in-2020>

<https://vuejs.org/guide/introduction.html>

<https://auth0.com/blog/how-to-make-secure-http-requests-with-vue-and-express/>



# PREGRADO

## Ingeniería de Software

Escuela de Ingeniería de Sistemas y Computación | Facultad de Ingeniería



### UPC

Universidad Peruana  
de Ciencias Aplicadas

Prolongación Primavera 2390,  
Monterrico, Santiago de Surco  
Lima 33 - Perú  
T 511 313 3333  
<https://www.upc.edu.pe>

***exígete, innova***