

# NUEVAS TENDENCIAS EN DESARROLLO WEB

AUTOR: Zelaya Reyes, Carlos Javier

[carlos.zelaya@itcha.edu.sv](mailto:carlos.zelaya@itcha.edu.sv)

## RESUMEN

Por desarrollo web entendemos aquellas disciplinas, técnicas y herramientas tecnológicas que permiten la creación de sitios o aplicaciones web que funcionan en un ambiente web y que se acceden a ellas a través de un navegador.

Las tendencias en desarrollo web, frameworks y lenguajes de programación son muy cambiantes y existen muchas alternativas para cada propósito, por tal razón, los desarrolladores web deben actualizarse constantemente en herramientas robustas y modernas para desarrollar productos aceptables y funcionales en la web.

En este artículo se hace una breve mención de las principales tecnologías, herramientas y lenguajes de programación, que actualmente se utilizan y serán el futuro del desarrollo web. Estas mismas herramientas para el desarrollo web, ha permitido la evolución de los programadores, lo que era novedad para años anteriores probablemente en estos momentos no lo sea.

Lo que está bien definido en los últimos años es la separación del desarrollo de aplicaciones en dos grandes áreas, por un lado, desarrolladores que se dedican a crear las interfaces que interactúan con el usuario conocido técnicamente como Front-End y desarrolladores Back-End, que construyen API's o servicios que conectan con la base de datos y exponen esos servicios para que sean consumidos por el Front-End.

## PALABRAS CLAVES

Desarrollador Front-end, Desarrollador Back-end, Desarrollador Full stack, Framework, Responsive Web Design, Rest API

## INTRODUCCION

Las tecnologías que los programadores deben dominar en la actualidad y a futuro, se enfocan en características como: división entre el Back-End y Front-End, las webs spa, cuyo propósito es crear contenido en una sola página con peticiones asíncronas mediante Ajax con frameworks de Javascript novedosos como: Angular, React y Vue o mediante el mismo lenguaje puro en su versión Vanilla JS. Se introduce un nuevo estándar conocido como API Rest para crear servicios a nivel de Back-End respetando la arquitectura http, ejemplo de ello tenemos: autenticación por Facebook, la autenticación en los servicios de Google.

## Teoría y Definiciones

**Definiciones:** veamos algunas definiciones para comprender más el contexto del artículo

**API:** (Application Programming Interface), interfaz de programación de aplicaciones, es un conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizada por otro software.

**API Rest** (Representational State Transfer): se originó en el año 2000, descrito en la tesis de Roy Fielding, padre de la especificación HTTP. Un servicio REST no es una arquitectura software, sino un conjunto de restricciones con las que podemos crear un estilo de arquitectura software. Un API REST permite crear servicios. Esto se debe a que es un estándar lógico y eficiente para la creación de servicios web. Las operaciones más importantes que nos permiten manipular los recursos son: GET para consultar y leer, POST para crear, PUT para editar y DELETE para eliminar.

**Front-End:** parte del desarrollo que se encarga de crear las interfaces con las que el usuario interactúa, son todas aquellas tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios. Para esta tarea se requiere tecnología como HTML5 para manejo del contenido, CSS para crear el

diseño, maquetación y la parte visual de la aplicación y el lenguaje de programación Javascript para usar algunos frameworks o librerías que expanden capacidades de interacción de las interfaces y consumir los servicios proporcionados por el backend.

**Backend:** es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos. El Backend también accede al servidor, que es una aplicación especializada que entiende la forma como el navegador realiza las peticiones.

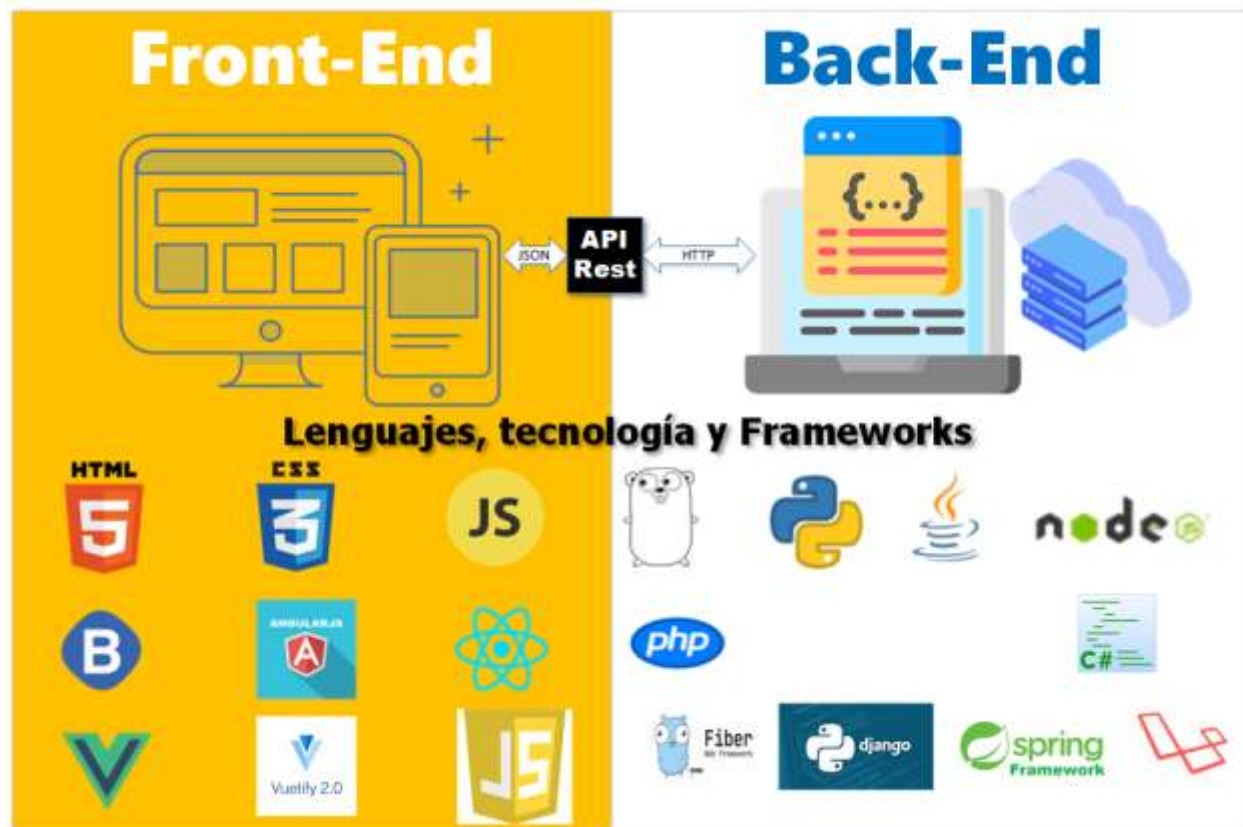
Algunos de los lenguajes de programación de Backend son: Python, Go, Java, PHP, Ruby y C#

**Ajax:** acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

**Framework:** o marco de trabajo, es el esquema o estructura que se establece y se aprovecha para desarrollar y organizar un producto de software. Esto facilita la programación de cualquier aplicación actual, existen para muchas tecnologías para el front-end y el back-end

## Teoría

El desarrollo web consiste en la creación de aplicaciones web implementadas en internet o intranet, estas aplicaciones contienen procesos, servicios o funcionalidades del quehacer empresarial o institucional. Para lograr esto, se requiere el uso de una serie de tecnologías a nivel de vista de usuario y del lado del servidor, que deben garantizar la interactividad y el rendimiento de las aplicaciones. Como he mencionado anteriormente, el desarrollo web en la actualidad se divide en Front-End y Back-End. En el siguiente esquema se muestra las principales tecnologías, frameworks y lenguajes de programación que están y siguen siendo utilizadas para esta actividad.



El esquema anterior representa por un lado la división del desarrollo web y por otro los lenguajes de programación, tecnología y frameworks que se utilizan a nivel de front-end y back-end. Por otra parte, la comunicación a través de los servicios que se expone o brinda el back-end por medio de la implementación de API Rest, estos a su vez son consumidos por el front-end por peticiones asíncronas (Ajax), utilizando el lenguaje Javascript, actualmente los desarrolladores front-end utilizan frameworks de Javascript para crear interactiva la interfaz del usuario y consumir los servicios de una API Rest, entre los que más se utilizan están: Angular, React JS, VueJS y en su forma nativa por medio de Vanilla JS. La información se traslada por JSON, debido a que es un formato ligero de intercambio de datos, que resulta sencillo de leer y escribir para los programadores y simple de interpretar y generar para las máquinas.

## CUADRO RESUMEN DE PRINCIPALES TECNOLOGIAS Y LENGUAJES DE PROGRAMACION PARA DESARROLLO WEB

Tecnología / Lenguaje	Propósito	Frameworks
HTML5	Lenguaje de marcado para el manejo del contenido de una aplicación web.	Bootstrap, Foundation
CSS	Crear la presentación de las aplicaciones, así como la maquetación de la misma.	Bootstrap, Foundation
JavaScript	Crear la parte interactiva del front-end y también en el back-end con NodeJS	Angular, React,Vue, Vanilla JS
Python	Desarrollar back-end, así como utilizado para otros propósitos como machine Learning, Big Data, Data Science e Inteligencia artificial.	Django Pyramid Bottle
Go	Desarrollar back-end, pensado en aprovechar sistemas con múltiples procesadores	Fiber
Java	Desarrollar el núcleo de una aplicación a nivel de back-end	Spring Boot
PHP	Desarrollar back-end	Laravel, CodeIgniter, Symfony
C#	Desarrollar back-end	Entity Framework
Ruby	Desarrollar back-end	Ruby on Rails, Hobbit,Ramaze, Ruby Grape

### Discusión

Entre las tendencias de desarrollo web más observables esta: la migración de aplicaciones a la nube para mantener alta disponibilidad, rendimiento, mayor seguridad, ventaja competitiva, entre otras. La separación del desarrollo web en front-end y back-end bien marcada, cada día se requieren profesionales que manejen tecnologías en ambas tareas conocidos como desarrolladores Full Stack. A nivel de front-end se requiere dominio de JavaScript puro o frameworks reactivos orientados a la vista como Angular de Google, React de Facebook o Vue creado por Evan You ex trabajador de Google y ahora le dan soporte una comunidad de desarrolladores, que además es

progresivo facilitando la renderización de los elementos web y el desarrollo evolutivo. Sin embargo, para el desarrollo empresarial la tendencia es utilizar Angular en el front-end por su metodología, arquitectura y por el uso nativo de TypeScript, esto ayuda en proyectos grandes. Si por el contrario se requiere mayor flexibilidad se recomienda usar React. Es recomendable dominar Javascript puro, por ejemplo, Vanilla JS es una iniciativa, en forma de framework que intenta enseñar las grandes ventajas de no usar frameworks y potenciar nuestras aplicaciones sin necesidad de añadir grandes archivos extras y cada vez tener código más limpio en las aplicaciones, esto se ve reflejado en la nueva versión 5 (beta) de Bootstrap, en donde saca a JQuery como framework de JavaScript e incorpora Vanilla JS.

Por otro lado, en el desarrollo back-end Python, está siendo uno de los lenguajes de programación más utilizados en proyectos nuevos, por su simplicidad, flexibilidad, ciencia de datos, ruta de aprendizaje más corta y mejor rendimiento comparado con uno de los lenguajes de programación más potentes y robustos en la última década, como es el caso de Java, sin embargo, esto no significa la muerte de Java, debido a que existen miles de proyectos de software empresariales implementados en esta plataforma de múltiples propósitos.

Es importante destacar a Go o Golang, lenguaje muy joven de código abierto creado por Google, que según el sitio web de StackOverflow en el 2019, fue uno de los lenguajes mejor pagados y posee características de sintaxis amigable parecida a Python, velocidad similar a la de C, mantiene el rendimiento con grandes volúmenes de información, su curva de aprendizaje es suave en comparación a la de C o Java, por eso es importante pensar en desarrollar back-end con Go.

Finalmente, en diseño web hay ciertas características para las interfaces de usuario cuya tendencia es a mantenerse, por ejemplo: diseño 3D, esta cobra perspectiva y se traduce en nuevas representación y experiencias al usuario; el uso de tipografía e iconografía sencillas y llamativas que de alguna manera rompen las reglas tradicionales; animaciones necesarias y profesionales como material design de Google cada vez se ven en numerosos sitios web.

Minimalismo, “menos es más”, las paginas sencillas y limpias no solo son más atractivas, sino que captan la atención del usuario cuando estamos ante un diseño más lineal que uno más cargado de elementos. Los chatbots están para quedarse, si creamos un ecommerce, un servicio de atención al cliente es a través de chatbot puede ser decisivo para un proceso de compra por parte de un potencial cliente. Estas son solo algunas características de diseño web que en la actualidad y a futuro se deben implementarse para que un sitio o aplicación se considere moderno y atractivo al usuario.

#### Citas o referencias bibliográficas.

Jose Maria Baquero. *Hoy vs. Ayer: unas reflexiones sobre el desarrollo web*, Tomado de <https://www.developersti.es/reportajes-developer/evolucion-desarrollo-web/>

Maldeadora de Platzi. *Que es Frontend y Backend*. <https://platzi.com/blog/que-es-frontend-y-backend/>

Bego Romero. *11 Tendencias de diseño web para 2020*, Tomado de <https://www.inboundcycle.com/blog-de-inbound-marketing/tendencias-diseno-web-para-2020>