

## #YoProgramo

◀ Volver al menu principal

Material de lectura

### Introducción a Desarrollo Web y Aplicaciones

#### ¿Desarrollo Web full stack?

El *desarrollador web full stack* puede crear aplicaciones web **dinámicas**. Esto se logra en base a los [estándares web](#) y utilizando tecnologías web que pueden variar según la pila de desarrollo. Tiene por objetivo la creación de aplicaciones web dinámicas. Como vimos anteriormente, las aplicaciones web dinámicas modifican su contenido en función del usuario que acceden permitiendo mostrar uno u otro contenido dependiendo del usuario y de su interacción para con la aplicación web (ej. Facebook, Instagram, etc.) mientras que las aplicaciones web estáticas muestra siempre el mismo contenido independientemente del usuario visitante (ej. un blog).

Pero ¿qué son los estándares web y qué buscadores existen? Los estándares web son tecnologías que se utilizan para crear aplicaciones web. Los mismos son creados por organismos de estándares - instituciones que invitan a grupos de personas de diferentes compañías de tecnología a unirse y acordar cómo deben funcionar las tecnologías de la mejor manera posible para cumplir con todos sus casos de uso. El [W3C](#) es el organismo de estándares web más conocido, pero hay otros como [WHATWG](#) (responsables de la modernización del lenguaje HTML), [ECMA](#) (publica el estándar para ECMAScript, en el que se basa JavaScript), [Khronos](#) (publica tecnologías para gráficos 3D, como WebGL) y otras [MDN Web Docs](#).

#### ¿Pero qué son los Stacks?

Lo que denominamos **stack tecnológico**, o también denominado **stack de soluciones** o ecosistema de **datos**, es un conjunto de todas las herramientas tecnológicas utilizadas para construir y ejecutar una sola aplicación.

Recordemos lo que ya hemos visto en la arquitectura Cliente /Servidor a su vez al código que navega el cliente por el navegador le llamamos Front End y la parte de código que accede a los datos se le llama Back End , si no recuerdas has clic [aquí](#),

Cada una de estas partes de la arquitectura puede crearse con distintos lenguajes de programación, el que quieras, por citar algunos ejemplo:

**Front End:** Para el desarrollo del Front End se necesita.

Estructura y estilos: HTML, CSS o framework como Bootstrap

Lenguaje programación: JavaScript, java, PHP o bien pueden ser framework como Angular

**Back End:** Para el desarrollo de Back End se necesita un web server, una base de datos y un lenguaje de programación.

Base de datos: puede ser MySQL, PostgreSQL, etc

Lenguaje: Java, PHP, etc

Web server: Apache, NGINX, etc

Como son muchas las opciones de lenguaje de programación, bases de datos y frameworks. Se ha creado el concepto de Stacks o pila que agrupa las tecnologías (que funcionan bien en conjunto o mas utilizadas) para crear una aplicación desde el Front End y Back End usando ese Stack y ahorrarnos tener que investigar a

Ayuda

Texto a voz

detalle que tecnología podemos utilizar, es como un combo de tecnologías. Esto también ayuda a las empresas a identificar el conocimiento que tiene un programador al nombrar el Stack tecnológico que utiliza.

Sitios como la red social Facebook, han sido desarrolladas por una combinación de frameworks de codificación y lenguajes, entre los que se incluyen JavaScript, HTML, CSS, PHP y ReactJS. Así podemos decir que este es el "stack tecnológico" de Facebook.

Podríamos preguntarnos cuáles son los stacks que usan Netflix, Whatsapp, Instagram...

Uno de los stacks o pila de tecnologías más utilizado por los desarrolladores es el que se conoce por **LAMP** : Linux, Apache, MySQL y PHP. Cualquier web hecha con Wordpress, Drupal o Prestashop, por ejemplo, están hechas sobre estos cuatro pilares. Pero se pueden hacer las variaciones que se crean convenientes, puesto que muchas de estas tecnologías son intercambiables por otras similares. Por ejemplo, NginX en lugar de Apache, PostgreSQL en lugar de MySQL o Ruby on Rails en lugar de PHP.

Otro stack muy utilizado es el llamado **MEAN** , que se compone de MongoDB, Express, Angular y NodeJS. te dejamos un imagen ilustrativa del Stack MEAN.



Backend para la Arquitectura de Aplicaciones Frontend. (s. f.). IBM Developer. Recuperado 11 de octubre de 2021, de <https://developer.ibm.com/es/patterns/create-backend-for-frontend-application-architecture/>

Ayuda

Texto a voz

Fazt. (s. f.). YouTube. Recuperado 11 de octubre de 2021, de <https://www.youtube.com/channel/UCX9NJ471o7Wie1DQe94RVlg>

### Algunos ejemplos de Stacks

- **LAMP:** Linux - Apache - MySQL - PHP. Es la más antigua y el formato de aplicaciones predominante se denomina "**Multi Page Application o MPA**" o aplicación de múltiples páginas.
- **MEAN:** MongoDB - Express - AngularJS - Node.js. Es la más moderna y el formato de aplicaciones predominante se denomina "**Aplicación** de una sola página o **SPA**" o aplicación de una sola página.
- **.NET:** .NET + WebApi + IIS - SQL Server. Utilizan Microsoft . [Núcleo neto](#) como plataforma de desarrollo.
- **Patrón BFF:** Se puede usar una arquitectura Backend for Frontend (BFF) para crear backends para aplicaciones web o móviles orientados al cliente. Los BFF pueden ayudar a respaldar una aplicación con múltiples clientes al mismo tiempo que mueven el sistema a un estado menos acoplado que un sistema monolítico. El [acoplamiento](#) es "es el grado en que los módulos de un programa **depende** unos de otros". Una aplicación monolítica es una unidad cohesiva de código. Este patrón de código ayuda a los equipos de desarrollo a iterar las funciones más rápido y tener control sobre los backends para aplicaciones móviles sin afectar la experiencia de la aplicación móvil o web correspondiente.
- **Django:** Python - Django - MySQL
- **Ruby on Rails:** Ruby - SQLite - Rails
- **LEMP:** Linux - Nginx - MySQL - PHP
- **MERN:** MongoDB - Express - React - Node.js.
- **Otros Stacks:** Existen otros stacks y probablemente surjan nuevos en el futuro cercano. Una solución completa involucra no solo el stack, que en ocasiones conlleva la inversión en las correspondientes licencias, sino también las consideraciones referidas al costo de hardware, entre ellos el del servidor físico. Así por ejemplo, en torno al lenguaje de programación Java y de la mano del servidor web JBoss, se pueden implementar en servidores con sistemas operativos RedHat soluciones Web. El hardware puede ser por ejemplo de Dell, HP, entre otros. Otra posible configuración es el frontend desarrollado en torno a React, el backend en Python y Django y usando como base de datos SQLite.

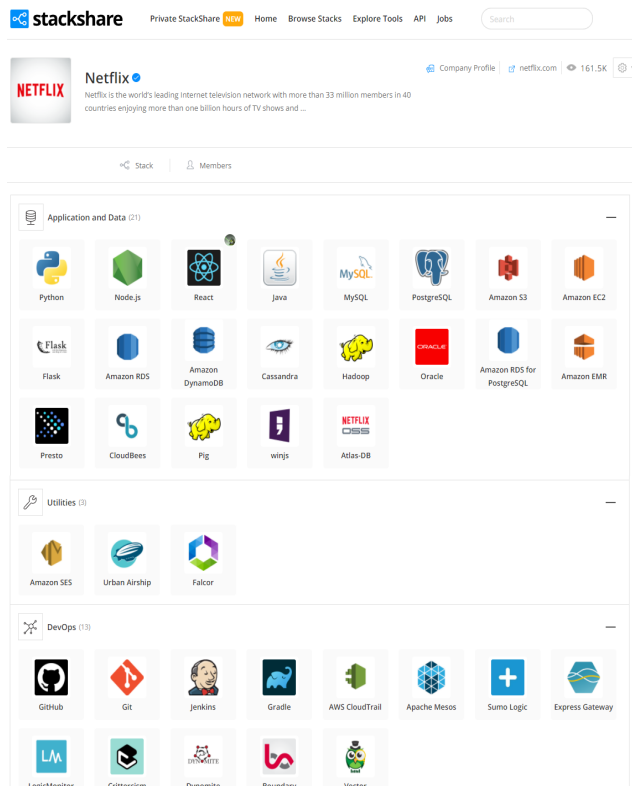
### ¿Todas las empresas usan si o si estos Stacks?

No siempre, tengamos en cuenta que los stacks son un grupo de herramientas tecnológica sugeridas para crear una aplicación web, por ende, las empresas pueden elegir las herramientas que necesiten sin tener en cuenta los Stack antes nombrados, Pero recuerda cuando una empresa solicita un programador LAMP y te presentan ellos asumirán que sabrás manejar ese grupo de tecnologías. Existen otras empresas que buscan perfiles con conocimiento solo en Front End o en Back End, estas empresas focalizan al programador en un solo área independientemente del stack.

En las siguientes imágenes te compartimos algunas de las tecnologías que utilizan unas empresas:

Ayuda

Texto a voz



Software and technology stacks used by top companies. (s. f.). StackShare. Recuperado 11 de octubre de 2021, de <https://stackshare.io/>

Como has notado las empresas utilizan distintas tecnologías

### ¿Con qué tecnología trabajaremos en el curso?

Luego de investigar y consultar la demanda requerida de las empresas vs las tecnologías mas utilizadas en Argentina, utilizaremos las siguientes tecnologías:

#### Para Front End:

Estructura y estilos: HTML, CSS y framework como Bootstrap

Lenguaje programación: TypeScript y el framework como Angular ambos basados en Javascript

#### Para Back End:

Base de datos: Usaremos MySQL

Lenguaje: Java con Framework Spring Boot

Web server: Apache Tomcat

Ahora que sabemos con las tecnologías que vamos a trabajar podemos decir que será ya tenemos nuestro stack tecnológico.

### ¿Qué significa ser un desarrollador full stack?

El desarrollador full stack es un perfil que tiene conocimiento de lenguajes de programación de Front End, Back End, APIs y Bases de datos. Esta amplitud de conocimiento es muy buscada hoy en día por las empresas, esto significa que al momento de trabajar en un proyecto podrías estar asignado a cualquiera de esas áreas.

Recuerda que tener el conocimiento de un programador full stack te amplía las posibilidades de trabajo, por ello, vale la pena el esfuerzo de aprender y practicar. En el procesos podrás descubrir que tienes facilidad o te gusta mas alguna parte del Full Stack (Front End, Back End o Bases de datos) y posiblemente te conviertas en experto en alguno. Por eso a seguir estudiando y practicando.

Ayuda

Texto a voz



Actividad previa

◀ Full Stack - Infografía

Siguiente actividad

Procesos de petición Infografía ▶

1 Resumen de retención de datos



Ayuda

Texto a voz