

Rutas para el aprendizaje de tecnologías del desarrollo Web en la actualidad

Herbert Brice Arias Silva¹

¹Departamento de Ingeniería de Software, Universidad Nacional Mayor de San Marcos

²Introducción a las Ciencias e Ingeniería

Resumen—En la actualidad la tecnología tiene un avance vertiginoso y esto genera mucho desconcierto al enfrentarse con la decisión de empezar el estudio de una de sus ramas, y este avance tiene un cambio mucho mayor en el ámbito de la informática ya que comunidades enteras de software así como empresas muy grandes del rubro están trabajando en el desarrollo de nuevas y mejoradas tecnologías que están reemplazando muy rápido a otras tecnologías consideradas nuevas y muy usadas años atrás. En este artículo se realiza un breve recopilatorio de esas tecnologías su origen y uso en la Programación Web, veremos en líneas generales un panorama de cómo un estudiante de primeros ciclos puede abarcar desde el inicio, una carrera profesional enfocada a la Programación Web y todo los conocimientos extras que implica adquirir a lo largo de su estudio universitario.

I. INTRODUCCIÓN

Hoy en día casi para todo se requiere algún tipo de programación. Así pues, ¿qué es? Programar es básicamente explicarle a tu ordenador que quieres que haga por ti. Pero citemos lo que piensan sobre esto los grandes programadores que han revolucionado algún sector con la programación:

- Gabe Newell (creador de Valve): “Cuando estás programando le estás enseñando a la cosa posiblemente más estúpida del universo, un ordenador, a hacer algo”.
- Mark Zuckerberg (creador de Facebook): “Programar es una de las pocas cosas en el mundo que puedes hacer cuando estás sentado y simplemente crear algo completamente nuevo desde cero”
- Drew Houston (creador de Dropbox): “Realmente no es muy diferente de tocar un instrumento o practicar un deporte. Empieza siendo algo muy intimidante, pero terminas por cogerle el truco.”
- Chris Bosh (programador de la NBA): “Programar es algo que puede aprenderse. Y sé que puede ser intimidante, muchas cosas son intimidantes. Pero ya sabes ¿qué no lo es?”

La programación es algo absolutamente necesario en nuestra época, ya que está en el centro de los mejores productos de la tierra, el software ya se apoderó del mundo. Saber código hace a cualquier profesional mejor, ya que si un profesional aprende a programar esto le dará una capacidad impresionante de cambiar su profesión. Las personas más exitosas, la gente que tiene los proyectos más exitosos, grandes y de crecimiento en Internet son aquellos que tienen la intersección de dos conocimientos y uno de esos conocimientos necesarios en muchos de esos casos es la programación.



Figura 1. Lenguajes, sistemas y tecnologías Web

En el colegio hemos aprendido cosas muy complejas, y más para ingresar a la universidad ya que se requieren tener cierto nivel de conocimientos complicados como por ejemplo en química, balancear una ecuación por Redox o en física el uso de ecuaciones para interpretar el movimiento parabólico de los cuerpos con masa y aceleración... Entender los fundamentos de la programación es mucho más sencillo que todo eso aunque aprender física o química incluso nos puede acercar en cierto momento a querer aprender programación ya que aprender sobre los semiconductores en química o los circuitos y teoría de transistores en física nos acerca a la programación porque tienen mucho que ver. Pero ¿por qué muchos sino la mayoría no aprendemos programación durante el colegio o más crítico aún durante la universidad?. Esto tiene que ver con el Álgebra y el cálculo, y es que con éstos saltamos a las matemáticas que casi siempre son útiles para la Ingeniería Civil o para la ingeniería Bioquímica pero no necesariamente para la Ingeniería de Sistemas, Ingeniería de Software o las ciencias de la computación, por ejemplo nos enseñan límites, nos enseñan integrales, nos enseñan a calcular el área bajo la curva, y es algo muy importante pero a su vez es algo muy denso, es como si pasáramos de aprender a conducir un automóvil automático a aprender a conducir un automóvil de la fórmula uno, y programar deja de ser prioridad en la vida universitaria de muchos futuros ingenieros.

La mayor parte de la programación está relacionada con

la Web y es que la programación web es muy interesante y amplia, el hecho de que el lenguaje de programación más popular y usado sea JavaScript, que es el lenguaje que nos permite interactuar con los navegadores Web, no hace más que confirmar lo anteriormente mencionado, es por ésto que es importante conocer un poco más la esencia de la Web en la actualidad.

El objetivo principal de este breve artículo no es solo el de informar qué tecnologías web son a las que podemos apuntar aprender sino el de instar a formular proyectos desde ahora ya que el camino de la programación Web implica adquirir conocimientos, pero más importante aplicarlos y en el camino descubrir qué nuevos conocimientos podemos y debemos adquirir.

Inicialmente el artículo busca motivar a mis compañeros a involucrarse con el desarrollo Web y que conozcan las herramientas y tecnologías así como recursos de aprendizaje a la que todos podemos acceder ya que el Internet es libre, pero muchas veces no conocemos los recursos y no logramos encontrarlos y aprovecharlos.

II. PANORAMA DE LA PROGRAMACIÓN WEB EN EL 2018

Hay que tener en cuenta que en la actualidad el internet funciona de diferente manera a que lo hacía en los 90's y es que actualmente la Web a cambiado mucho. Según Sergio Luján Mora, en pocos años, Internet a invadido casi todos los aspectos de nuestras vidas. Podemos comunicarnos a través de Internet de distintas formas (correo electrónico, radio y televisión online, telefonía IP). Podemos comprar diversos productos en Internet (libros, entradas de cine, ropa). Podemos conocer gente a través de Internet (chat, foros de discusión, Facebook). Para que ello funcione, hace falta profesionales especializados en "programación en Internet". En su libro él explica cómo programar la parte cliente dentro de la arquitectura cliente/servidor. Para ello enseña HTML y JavaScript principalmente. Sergio Luján menciona que en las aplicaciones web suelen distinguirse tres niveles: nivel superior que interacciona con el usuario (el cliente web, normalmente un navegador), el nivel inferior que proporciona los datos (la base de datos) y el nivel intermedio que procesa los datos (el servidor web).[1].

La forma en cómo interactuamos con Internet a través de la Web en el 2018 sigue una dirección que se trazó desde inicios del 2000 ya que la web tuvo un punto de quiebre y los cambios se visualizan en la actualidad ya que ahora se trata de la Web Social o Web 2.0, donde destacan plataformas como Wikipedia, Youtube, Facebook, Google, Flickr, Tumblr, Twitter, Instagram, Uber, etc. Y se tratan de plataformas construidas para la gran masa de usuarios y son éstos los que crean los contenidos que otros consumen. Según Natalia Vazquez, esta web está basado en dos principios: **inteligencia colectiva** que es la suma del saber de cada uno de los individuos que al compartirse puede dar lugar a una obra colectiva y la **arquitectura de la participación** que implica una nueva forma de construir los sitios web para permitir la participación de la gran masa de usuarios. La web

se convierte entonces en una plataforma de servicios donde los usuarios asumen una filosofía que supone compartir los recursos propios a la vez que nos beneficiamos de los ajenos, saldando así una especie de deuda con la comunidad.[2].

Para construir una página y luego una aplicación Web es necesario el conocimiento de muchas tecnologías y lenguajes de programación. Para empezar podemos citar un lenguaje de programación que es muy fácil de aprender, emplear y mantener. Según Marzal Andrés, Python es un un lenguaje de muy alto nivel que permite expresar algoritmos de forma casi directa, incluso se puede considerar un "pseudocódigo ejecutable" y gracias la experiencia en la enseñanza de los autores del libro "Introducción a la programación con Python 3" han concluido que es un lenguaje particularmente adecuado para el primer punto de contacto con la programación[3].

Actualmente existen muchos lenguajes de programación, cada lenguaje tiene sus ventajas y desventajas así como su comunidad de programadores que las respalda, pero los lenguajes de programación manejan la misma lógica, lo que las diferencia es su forma de manejar la traducción del código, los paradigmas, y principalmente su sintaxis. Así los lenguajes de programación nos permiten comunicar instrucciones a las computadoras y lograr que éstas hagan trabajo por nosotros, es necesario entonces entender la lógica de programación y hay varias formas de entenderla sin involucrarse con un lenguaje de programación en específico, la manera más aceptada y es la que enseña el autor Luis Joyanes en su libro es el Pseudocódigo[4].



Figura 2. Sistema de Control de Versiones Git

El Software Libre es el software que respeta la libertad del usuario y la solidaridad social de su comunidad. El software que no es libre se llama Software Privativo porque priva de la libertad a sus usuarios y los mantiene divididos e impotentes; divididos porque cada uno está prohibido de compartir con los demás e impotentes porque no tienen el código fuente, por lo tanto no pueden cambiar el programa ni siquiera pueden

averiguar lo que realmente hace el software. El Software Libre brinda al usuario cuatro libertades esenciales: Libertad 0 que es la libertad de ejecutar el programa como quieras, Libertad 1 que es la libertad de estudiar el código fuente del programa y poder cambiarlo para que el programa haga lo que quieras, Libertad 2 que es la libertad de ayudar a tu prójimo de poder hacer copias y distribuirlos a los demás cuando quieras, Libertad 3 que es la libertad de contribuir a tu comunidad es la libertad de hacer y distribuir copias de tus versiones cambiadas a los demás cuando quieras. El desarrollar un programa libre es una contribución a la sociedad, mas o menos según los detalles técnicos. El Open Source es una variación práctica de la ideología del Software Libre que no toma en cuenta las consideraciones éticas de la Libertad pero que en la práctica tienen mucho en común[5].



Figura 3. Proyecto GNU - Richard Stallman

as tecnologías web que tienen mayor éxito son las que tienen detrás una comunidad de desarrolladores grande y activa. Hay comunidades de que usan lenguajes de programación como JavaScript, Python, Ruby, PHP, etc. Estos lenguajes de programación son el núcleo del desarrollo de frameworks como AngularJS de JavaScript, Django de Python, Ruby on Rails de Ruby, o Laravel de PHP. Según Javier Gutierrez, el framework es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta.[6]

La colaboración de la comunidad de programadores para el desarrollo de proyectos como los frameworks se dan a través de plataformas web por ejemplo tenemos: GitHub (la más popular), Gitlab, Bitbucket, SourceForge, etc. En estas plataformas Web gratuitas y en otras podemos encontrar el código de estos proyectos, estudiarlos modificarlos y aportar a la comunidad. A continuación tenemos algunos enlaces a los proyectos de frameworks Web: para AngularJS tenemos <https://github.com/angular>, Django <https://github.com/django>, Ruby on Rails <https://github.com/rails>, Laravel <https://github.com/laravel>. GitHub también funciona como una red social de programadores en la que se pueden votar los proyectos y los usuarios. Cada usuario puede establecer vínculos con

otros usuarios y puede seguir lo que hace. Muchas empresas buscan programadores en esta plataforma hasta el punto de ser indispensable tener cuenta de GitHub para conseguir trabajo como programador[7].

Los proyectos almacenados en GitHub son controlados mediante Git. Git es un Sistema de Control de Versiones. ¿Qué es un controlador de versiones y por qué debería importarnos? Un Controlador de versiones (VCS) es un sistema que graba los cambios realizados en un archivo o conjunto de archivos en el mismo tiempo que se realizan, así es posible rescatar distintas versiones específicas después, ésta es una herramienta fundamental para realizar proyectos de Software. Antes de Git existían los llamados sistemas de control de versiones locales que lo que hacían era copiar los archivos dentro de otros directorios de una forma muy común y simple, pero eran propensos a los errores; luego se usaban los sistemas de control de versiones centralizados que fueron implementados para permitir el desarrollo colaborativo y fue por mucho años el estándar para el Control de Versiones ya que ofrecía muchas ventajas frente a los VCS locales pero presentaban serias desventajas, la más obvia es que el proyecto y toda su historia se centralizan en un solo punto, así si este servidor centralizado falla, durante el tiempo que falle nadie podrá colaborar o guardar los cambios y versiones en las que se estén trabajando y si el disco duro de la base central de datos se corrompe junto con los respaldos, se pierde absolutamente toda la historia entera del proyecto excepto la copia que mantienen localmente los desarrolladores que trabajan en ella, pero la imagen del proyecto se pierde es por esto que luego nacen los Sistemas de Control de Versiones Distribuidos (DVCSs) como Git, Mercurial, Bazaar o Darcs. Linus Torvalds es un Ingeniero de Software que es el desarrollador de Linux que es el núcleo que corre en los sistema GNU/Linux y también es el desarrollador de Git que nace de la necesidad que tenían los desarrolladores del Kernel de Linux de trabajar colaborativamente. Así Git nace en el seno del Open Source y Linux[8].

A inicios del año 1983, Richard Stallman inició el proyecto GNU que significa “GNU no es Linux” y que estaba soportado por la Free Software Foundation con el objetivo de crear un sistema operativo que sea capaz de ser usado de forma libre respetando los lineamientos de la FSF con sus cuatro libertades. Richard y su equipo consiguieron todas las herramientas necesarias para hacer que un computador funciones con software libre, pero hacía falta el núcleo que gestione los recursos y tareas del computador, para eso por el año 1987 el profesor Andrew S. Tanenbaum desarrolló un clon del Sistema Operativo Unix y lo distribuyó junto con su código fuente y tenía por finalidad la enseñanza, Linus Torvalds aprendió de Minix y logró desarrollar el kernel Linux que terminó formando parte del proyecto GNU. Linux solo, no era capaz de hacer nada, se integró con todas las herramientas creadas por el proyecto GNU y logran instalar el primer Sistema Operativo en un computador usando solamente software Libre. Actualmente Google corre miles de miles de servidores Linux que soporta su tecnología de

búsqueda, incluso en su sistema operativo Android corre el kernel Linux. Facebook construye y desarrolla su sitio web usando todo lo relacionado con LAMP stack (Linux, Apache, Web server, MySQL database y PHP). Las organizaciones financieras que tienen trillones de dólares corriendo sobre la velocidad y seguridad de sus sistemas operativos dependen fuertemente de Linux[9].

III. RUTAS DE APRENDIZAJE

Si se analiza, en la actualidad hay dos mundos que coexisten en el desarrollo web: Fronted y Backend. Entonces al ingresar a la programación Web se puede acceder a dos rutas de aprendizaje: Fronted o Backend.

Si aprendes Fronted serás un Fronted Developer. ¿Qué significa ser Fronted Developer? Significa ser un desarrollador capaz de convertir en real un diseño de UX/UI (User Experience / Use Interface) y programar la reacción ante la interacción que pueda tener el usuario dentro de la web y además conocer bien cómo recibir y enviar datos del servidor (backend). Fronted es más rápido de aprender que Backend, pero no necesariamente lo más fácil.

Si aprendes Backend serás un Backend Developer. Un desarrollador Backend es un programador que trabaja del lado del servidor. Permitiendo que todo lo que vemos cuando interactuamos con una aplicación o sitio web, funcione. Dicho de una forma más informal, es el que trabaja detrás del escenario, moviendo los hilos para que todo salga bien. Ahora hay una ruta que se abre si eliges el camino del Backend y es la del DevOps que viene de Development Operations. ¿Qué es DevOps? Es un camino para implementar Software con esfuerzo y responsabilidad compartida, así un DevOps Developer conoce el cómo implementar un software, su ciclo de vida y mantenimiento, actualmente es un cargo de moda que se requiere mucho en la industria y profesionales con éste perfil tienen muchos beneficios financieros.

III-A. Fronted

Existen tres tecnologías pilares en la parte Fronted de la programación Web, tenemos: HTML, CSS, JavaScript. HTML es un lenguaje de marcado para la elaboración de páginas Web. Es un estándar que está a cargo del *World Wide Consortium* (W3C). HTML permite organizar la información, para eso usa etiquetas, atributos y valores. En HTML5 se cuenta con etiquetas que dan significado a una porción de información útil para buscadores como Google, así Google va directo a la información contenida dentro de una etiqueta y la interpreta de forma inteligente, esto le sirve por ejemplo para clasificar las páginas web por su contenido, encontrar imágenes y videos, y destacar la información importante que ofrece la página al momento de hacer una búsqueda, esto se denomina Web semántica.

Una etiqueta se inicia con el signo “<” y se finaliza con “>”. Dentro de una etiqueta va su nombre y también pueden ir

atributos que a su vez pueden tener valores.

Podemos ver en la imagen 4 un ejemplo de cómo se vería un esquema html básico. HTML se puede aprender gratis visitando la siguiente documentación [10].

```

1 <!DOCTYPE html>~
2 <html>~
3   <head>~
4     <meta charset="utf-8" />~
5     <title>Introducción a la Ingeniería</title>~
6   </head>~
7   <body>~
8     <div class="cuerpo">~
9       <header class="encabezado">~
10        <nav>~
11          <a href="#">Inicio</a>~
12          <a href="#">Productos</a>~
13          <a href="#">Acerca de</a>~
14          <a href="#">Contacto</a>~
15        </nav>~
16      </header>~
17      <section class="main">~
18        <article class="contenido">~
19        </article>~
20        <aside class="sidebar">~
21        </aside>~
22      </section>~
23    </div>~
24  </body>~
25 </html>~

```

Figura 4. Etiquetado en HTML

Al inicio todo el mundo Web estaba hecho solo con HTML; sin embargo dar estilo a una página Web para hacerla ver más atractiva y organizada era muy complicado de hacer solo usando HTML, es por eso que surge CSS. Actualmente ya no se da estilo a una página web usando HTML ya que éste se dedica solo al estructurado de la información. CSS (Cascading Style Sheets) es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado como HTML. CSS funciona con el uso de selectores (de etiquetas HTML, id o clases), propiedades que adoptan valores como podemos ver en la imagen 5.

Es necesario practicar bastante el lenguaje CSS realizando muchos proyectos hasta que se llegue a un punto en el que se puede realizar un diseño y maquetarlo “en términos de CSS”; también es recomendable relacionarse con diseñadores, hablar de diseño, entender el diseño hasta que se adquiera un gusto de diseñador, éste último no es necesario pero si marca diferencia al momento de crear interfaces de usuario amigables, atractivos que otorguen una experiencia positiva al usuario.

CSS por otro lado es muy importante en la actualidad porque permite un desarrollo con el llamado “*Responsive Design*”, que hace que nuestro diseño se adapte a diferentes dispositivos con diferente tamaño de pantalla. Así con el uso de módulos como: *Media query*, *Flexbox*, *Grid*, etc. Podemos definir una disposición en una pantalla de 48 pulgadas y otra en un dispositivo Smartphone, el diseño se adapta.

HTML y CSS no son lenguajes de programación como lo es JavaScript. JavaScript es un lenguaje de programación que nos permite interactuar con el navegador. Nos brinda la posibilidad de otorgarle a nuestra página web interactividad. Los paneles desplegables, las ventanas emergentes, los slides o imágenes que se mueven, etc.

Si has elegido el camino del Fronted, el primer lenguaje recomendable de aprender es JavaScript, porque es un lenguaje fácil y muy versátil, hoy en día está en casi todos lados:

Sistemas Operativos, Desarrollo móvil, servidores de Internet, base de datos, plataformas de juegos, administración de sistemas, tanto Linux como Windows y si nos centramos en la web incluso se puede usar JavaScript como el único lenguaje de programación, desde el cliente hasta el servidor pasando por la base de datos. El famoso stack MEAN (MongoDB + Express + AngularJS + NodeJS) utiliza JavaScript como único lenguaje. Puedes visitar la siguiente documentación para estudiar JavaScript, que por cierto no tiene nada que ver con Java excepto el nombre [11] [12]. Una vez ya aprendiste lo básico de HTML, CSS y JavaScript puedes iniciar en el uso de Frameworks.

Inicialmente aparecieron Frameworks de CSS entre las cuales aún es muy usada una que fue creada por un empleado de Twitter: Bootstrap4. Este framework tiene plantillas que permiten crear páginas usando Flexbox de CSS e incluye JavaScript manipulando solamente HTML, esto facilita mucho el trabajo y muchos prototipos nacen con este Framework. Luego se empezaron a popularizar otros frameworks pero esta vez de JavaScript y actualmente tenemos como opciones las siguientes: AngularJS, React JS y Vue JS. AngularJS es un proyecto que cuenta con soporte de Google, tiene una curva de aprendizaje elevada pero una vez se logra dominarla te permite crear rápidamente plataformas muy adaptables y eficientes. React JS, fue creado por Facebook y también es muy popular entre los programadores, de igual manera es muy práctico y cumple su cometido, la plataforma Facebook está creada usando este Framework. Vue JS, es un proyecto que tiene soporte enteramente de la comunidad y está teniendo mucho eco en los últimos años.

```

79 aside {~
80   width: 30%;~
81   background: #161f26;~
82   padding: 20px;~
83   color: #fff;~
84   float: left;~
85 }~
86 ~
87 aside .widget {~
88   margin-bottom: 20px;~
89 }~
90 ~
91 aside .widget .titulo {~
92   color: #ff8000;~
93   border-bottom: 1px solid #ff8000;~
94   margin-bottom: 10px;~
95 }~
96 ~

```

Figura 5. Selectores y propiedades en CSS

III-B. Backend

El backend comprende a los servidores y a los lenguajes de programación que trabajan sobre éstos servidores. Los servidores nos permiten controlar servicios y realizar procesamiento incluyen base de datos y mensajes entre servidores y servicios. Lo básico incluye los lenguajes de programación que necesitarás conocer para trabajar sobre servidores web, puedes escoger entre los siguientes lenguajes: Ruby, NodeJS, Python, PHP7. **Ruby** es un lenguaje de programación bastante usado y si nos enfocamos en este lenguaje entonces podremos una vez aprendido a manejar el lenguaje usar su framework más

popular llamado Rails Ruby on Rails”que nos permite realizar los proyectos de una forma más estructurada, también es importante manejar el testing del lenguaje llamado Rspec que nos permite realizar pruebas de unidad y probar funcionalidad, finalmente su package manager llamado RubyGems que nos permiten administrar los paquetes y dependencias que nos permite añadir a nuestro proyecto paquetes externos. **NodeJS**, no es un lenguaje de programación sino que es un software que nos permite implementar código de JavaScript en nuestro servidor, es conveniente su uso ya que con unas cuantas líneas de código en JavaScript podemos montar un pequeño servidor Web y a administrarlo de la forma que lo harías con otro lenguaje más complejo”, su framework más recomendado es Express ya que a través de sus módulos que nos brindan muchas funcionalidades podemos crear servicios, Apis, servidores web muy fáciles, en cuanto al testing tenemos a Mocha que igual que con Ruby nos permite probar nuestro código y su funcionalidad, en cuanto a sistema de paquetería tenemos algo llamado npm que nos permite administrar paquetería de manera muy fácil a través de línea de comandos. **Python**, tenemos al framework Django nos permite implementar de forma muy sencilla la administración de servicios y base de datos ya que python es un lenguaje muy fácil de aprender e implementar, su testing se realiza con Pytest y su package manager se llama pip igual su uso es en línea de comandos [13] [3] [14] [3]. **PHP7**, es la versión actual tiene como framework recomendado a Laravel para poder implementar proyecto de calidad con PHP, su testing es realizado con PHPUnit y su package manager es llamado Composer para la administración de tus paquetes y dependencias externas. Algunos otros lenguajes que podemos optar aprender está C# que está vinculado con .Net, y también Java.

Luego de aprender un lenguaje de programación, su framework y saber manejar su paquetería y testing, entonces podemos aprender a manejar algo llamado **Negociador de mensajes** y es recomendable uno que tiene nombre RabbitMQ que nos permite comunicar servicios. La comunicación entre servicios y backend y fronted se realiza con archivos llamados JSON. El almacenamiento podemos dividirlo en Cache y Base de datos. **Cache**, es una memoria que permite acceder a sus información de forma rápida en éste apartado podemos usar REDIS que es un motor de base de datos pero que nos permite administrar la cache. **Base de datos**, nos permite almacenar datos en forma estructurado o no, tenemos las bases de datos relacionales (sql) o las no relacionales (no sql); en base de datos relacionales tenemos: Mysql, MariaDB, Oracle; en base de datos no relacionales tenemos MongoDB que básicamente contiene una estructura que puede contener información a diferentes niveles de anidación dependiendo si un objeto contenido tiene propiedades o una colección de objetos.

IV. RESULTADOS

Los resultados obtenidos por la encuesta **Global States of JS del 2018** [15], así como los datos de tendencia de usos de **Google Trends** tanto regionales como globales [16] y también del número de tags del portal StackOverflow son los siguientes:

IV-A. Nueva versión de JavaScript

JavaScript ha estado cambiando desde la revolución de HTML5 hasta la actualidad, y han habido muchos cambios pero la más importante es una actualización de JavaScript llamada ECMAScript 6, el mismo es un cambio radical de la forma en la que nosotros implementamos el lenguaje y las capacidades que tiene el lenguaje tanto en el navegador como en el servidor. Hay otras organizaciones que han tratado de hacer cambios con JavaScript como por ejemplo con Typescript apoyado por Microsoft, Flow, Reason, Elm, ClosureScript, etc. Según la encuesta Global States of JS el 86 % de programadores de JavaScript han usado ECMAScript 6 y planean seguir usándolo y 45 % han usado TypeScript y planean seguir usándolo y como ECMAScript 6 resuelve muchos de los problemas que tiene TypeScript probablemente no vamos a necesitar TypeScript en el futuro.

IV-B. Fronted Frameworks

Segun resultados de la encuesta Global se puede verificar que solo el 3 % del total de encuestados han usado Polymer y lo volverían a usar mientras que un 51.5 % han escuchado de éste framework pero no están interesados en aprender a usarlo. React lidera la lista con un 64.8 % de personas que lo han usado y volverían a usarlo, seguido por Vue.js con un sorprendente 46.6 % de desarrolladores que han escuchado sobre éste y quisieran aprender a usarlo lo que hace pensar que seguirá compitiendo, y un sorprendente tercer lugar para Angular que años atrás lideraba este apartado y registra un 33.8 % de desarrolladores que lo han usado y no lo volverían a usar de nuevo. Se puede verificar la imagen 6.



Figura 6. Encuesta Framework - Fronted

Si se visualiza la tendencia de búsquedas usando Google Trends, que puede observarse en la imagen 7 en la que podemos visualizar cómo a partir de Junio de año 2018 React ha empezado a superar en número de búsquedas a Angular y cómo también Vue.js no ha dejado de crecer. También se puede verificar el número de tags del portal Stackoverflow que nos puede ayudar a ver cuánta actividad se realiza entorno a una tecnología y cuántas dudas se escriben al respecto, se puede observar cómo lidera JQuery (1719753 tags) seguido por Angular (255968 tags) que duplica en número

de tags a React (111162 tags) y esto puede demostrar dos cosas: lo difícil que es Angular respecto a React o también la importancia histórica de búsquedas de Angular ya que éste fue más popular, incluso JQuery fue más popular que Angular hace tan solo un año.



Figura 7. Google Trends. React vs Angular vs Vue.js

IV-C. Manejo y transmisión de datos, Backend Frameworks

Analizando el apartado del manejo del estado y transmisión de los datos, compiten tres Frameworks: Redux, GraphQL y Apollo. Redux domina el uso, pero GraphQL queda en el segundo lugar y es el que genera mayor interés por aprender sobre él. GraphQL es una manera distinta de manejar la capa de datos en JavaScript y representa el futuro, ya que se integra con aplicaciones web nativas, con aplicaciones híbridas y con los sitios web.



Figura 8. Frameworks de la capa de datos

Ahora en el backend compiten: Express, Next.js, Koa, etc. Express es el más popular mientras que Next.js promete volverse más popular. En cuanto a editores de código para programar en JavaScript destaca Visual Studio Code sobre los demás, seguido por Vim, Atom y Webstorm.

Luego de ver estos frameworks, se verifica que GraphQL es la tecnología que genera mayor interés y esto se podrá ver reflejado con una subida en su uso para el año 2019, React es la tecnología más usada y la más recomendada por la comunidad y que no entró en la encuesta fue Vuex que está relacionada con Vue.js. Facebook es el equipo que ha generado mayor aporte con tecnologías más usadas con React y GraphQL, ¿por qué no seguiría este camino Reason?.

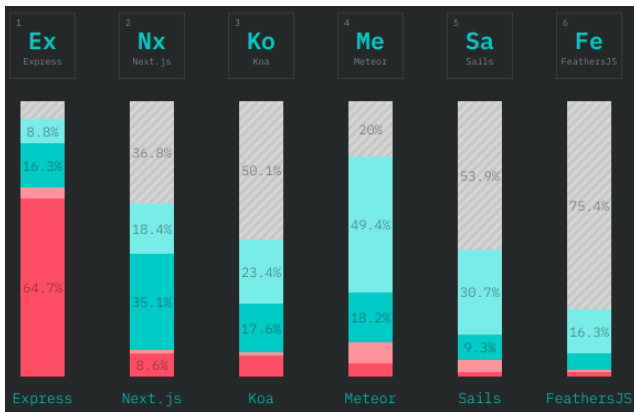


Figura 9. Frameworks de backend

V. CONCLUSIONES

Es recomendable seguir la tendencia de la tecnología Web: React como el Framework más usado nos ofrece una gran oportunidad, así como la proyección de GraphQL o Vue.js. Aprender todas estas tecnologías nos va a servir para poder llevar a cabo algún proyecto, hay que comprender que la tecnología cambia a un ritmo acelerado, mucho de lo que aprendemos deja de ser usado y cambiado por algo diferente, es necesario estar siempre actualizado con las tendencias, ya que nuestros proyectos tienen que estar a la vanguardia sino estarán condenados al fracaso.

Es necesario también recordar que la programación al ser apasionante no es un fin sí mismo sino mas bien una herramienta que nos permite crear soluciones, por lo tanto es importante conocer la problemática de nuestra sociedad, relacionarnos con otros profesionales: de la salud, psicología, administración, sociología, química, eléctrica, etc. Esto puede darnos la experiencia necesaria para crear proyectos que realmente sean útiles y de impacto para la sociedad, dicho también que si eres no eres un profesional de Ingeniería de Software o Sistemas y aprendes a programar podrás crear cosas muy interesantes relacionadas con tu carrera.

REFERENCIAS

- [1] S. Luján-Mora, *Programación en Internet: clientes web*. 2001. dirección: <http://hdl.handle.net/10045/16994>.
- [2] N. Arroyo-Vázquez, «Educación y biblioteca, n. 161», *Educación y biblioteca*, págs. 69-74, 2007. dirección: <https://gredos.usal.es/jspui/handle/10366/108883>.
- [3] G. P. Marzal Andrés Gracia Isabel, *Introducción a la programación con Python 3*. 2014. dirección: <http://repositori.uji.es/xmlui/handle/10234/102653?locale-attribute=es>.
- [4] L. J. Aguilar, *Fundamentos de Programación*. 2008.
- [5] R. M. Stallman, *Software libre para una sociedad libre*. 2008. dirección: https://www.gnu.org/philosophy/fsfs/free_software.es.pdf.
- [6] J. J. Gutiérrez., «¿Qué es un framework web?», pág. 4, 2014. dirección: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf.

- [7] L. J. S. González, *Git y GitHub. Guía de Supervivencia*. 2016. dirección: <http://leanpub.com/gitygithub>.
- [8] B. S. Scott Chacon, *Pro Git Second Edition*. 2014. dirección: <https://git-scm.com/book/en/v2>.
- [9] C. Negus, *Linux Bible Ninth Edition, The comprehensive, tutorial resource*. 2005.
- [10] R. Data. (2018). HTML5 Tutorial, dirección: <https://www.w3schools.com/html/>.
- [11] —, (2018). JavaScript Tutorial, dirección: <https://www.w3schools.com/js/default.asp>.
- [12] B. L. Jim Cooper. (2018). JavaScript.com by Plural-sight, dirección: <https://www.javascript.com/try>.
- [13] G. van Rossum. (2018). The Python Tutorial, dirección: <https://docs.python.org/3/tutorial/index.html>.
- [14] A. F. Montoro, *Python 3 al descubierto*. 2012.
- [15] M. R. Sacha Greif Raphael Benitte. (2018). The State of JavaScript 2018, dirección: <https://2018.stateofjs.com/>.
- [16] G. Google. (2018). React, Angular, Vue.js - Google Trends, dirección: <https://trends.google.es/trends/explore?q=%2Fm%2F01211vxxv,%2Fm%2F0j45p7w,%2Fg%2F11c0vmgx5d>.