**¿FRONTEND O BACKEND?**

**ESA ES LA CUESTIÓN**

1. **¿Qué es la programación?**

Dar un grupo de instrucciones a un computador o máquina para que las ejecute. Entre más complejas sean, más complejo es el resultado.

1. **¿Qué es el desarrollo web?**

Desarrollo del contenido de una página web, teniendo en cuenta la comunicación cliente servidor y la seguridad de la red. El cliente se refiere al acceso a los datos del usuario y proveer información. El lado del servidor necesita almacenar y procesar la información.

1. **¿Qué debe aprender un programador?**

* Desarrollar la lógica computacional y tener conocimientos de las estructuras de datos que puede emplear en cualquier lenguaje.
* Saber dónde buscar la documentación pertinente.
* Controles de versionamiento.

1. **¿Qué trabajos realiza un programador?**

Se encarga de escribir, depurar y revisar todo el código fuente de un software informático para que lleve a cabo determinadas tareas. También puede actualizar o mejorar un programa existente.

1. **¿Qué es una roadmap en desarrollo?**

Hoja de ruta de un proyecto con la vista de alto nivel de las entregas y objetivos generales del proyecto. Es lo primero que se crea al empezar un proyecto para tener la visión general del proyecto, sin instrucciones de cómo será el paso a paso.

1. **¿Qué debe aprender un desarrollador web?**

Aprender HTML, CSS, JavaScript, algún lenguaje backend, frameworks, bases de datos relacionales y no relacionales. Inglés también es fundamental para leer documentación, foros y mejorar las oportunidades de un trabajo remoto o presencial con empresas internacionales.

1. **¿Qué trabajos realiza un desarrollador web?**

Crea sitios web utilizando varios lenguajes de programación. Sus responsabilidades incluyen el diseño de páginas web y el desarrollo de la funcionalidad del sitio para satisfacer las necesidades de los usuarios. A veces, una vez terminado el proyecto, un programador web puede encargarse de mantener el rendimiento del sitio web, realizar actualizaciones y comprobar si hay problemas técnicos.

1. **¿Qué es FRONTEND?**

Se refiere al diseño web, la funcionalidad y otros elementos del lado del cliente. El objetivo de front-end es hacer que un sitio web sea visualmente atractivo y totalmente interactivo. Se muestran y se reciben datos por medio de la interfaz gráfica.

1. **¿Qué es BACKEND?**

Se refiere a la infraestructura del sitio web y de los elementos del lado del servidor. Allí se asegura que un sitio web tenga un rendimiento óptimo en todas las funcionalidades. Se procesan, almacenan y envían los datos entre la base de datos y el front.

1. **¿Qué es el nivel Junior en desarrollo?**

Experiencia menor a 2 años. Etapa donde se adquieren conocimientos y buenas prácticas en cuanto al manejo de herramientas y la forma apropiada de desarrollar proyectos según las necesidades de los clientes. Se encarga de tareas más sencillas y de bajo impacto.

1. **¿Qué es el nivel Senior en desarrollo?**

Se consigue este nivel a partir de los 10 años de experiencia y esto conlleva a que lidere grupos de programadores junior. El código generado y las soluciones en general son de mayor calidad, ya que son funcionales, eficientes, se pueden entender, implementar y mejorar de la forma más rápida posible. Está en la capacidad de desarrollar un software de cero de forma independiente.

1. **¿Qué es el lenguaje HTML?**

(Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. "Hipertexto" hace referencia a los enlaces que conectan páginas web entre sí, ya sea dentro de un único sitio web o entre sitios web. Un elemento se distingue de otro en un documento mediante etiquetas, que consisten en el nombre del elemento rodeado por "< >".

1. **¿Qué es el lenguaje CSS?**

Las Hojas de estilo en cascada (del inglés Cascading Stylesheets CSS) se utiliza para definir la estructura y la semántica del contenido, se usa para darle estilo y posicionarlo visualmente. Se puede cambiar la fuente, el color, el tamaño y el espaciado del contenido, para formar múltiples columnas, añadir animaciones y otros elementos decorativos.

1. **¿Qué es SASS?**

Sass es un lenguaje de hoja de estilos en cascada inicialmente diseñado por Hampton Catlin y desarrollado por Natalie Weizenbaum. Significa Syntactically Awesome Stylesheet.

1. **¿Qué es BEM?**

Es una metodología que permite crear componentes reutilizables en el desarrollo de estilos frontend.

1. **¿Qué es el lenguaje JavaScript?**

Es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes

1. **¿Qué es el lenguaje TypeScript?**

Extensión de JavaScript con características adicionales que hacen que podamos escribir código con menos errores, más sencillo, coherente y fácil de probar, en definitiva, más limpio y sólido. Fue creado por Microsoft en 2012. Se usa en proyectos extensos.

1. **¿Qué es el lenguaje Python?**

Lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software es gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

1. **¿Qué es el lenguaje PHP?**

HyperText Preprocessor es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML. Se destaca porque es de código abierto.

1. **¿Qué es el lenguaje MySQL?**

Lenguaje de código abierto que permite gestionar bases de datos relacionales. Es la más usada en las aplicaciones web.

1. **¿Qué es una Base de Datos?**

Es la recopilación organizada de información o datos estructurados, que se almacenan en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a base de datos.

1. **¿Qué es NoSQL?**

Bases de datos no relacionales están diseñadas específicamente para esquemas flexibles de aplicaciones modernas. Las bases de datos NoSQL son ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala.

1. **¿Qué es el lenguaje JAVA?**

Es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems. Su paradigma está orientado a objetos.

1. **¿Qué es un framework?**

Es una estructura base utilizada como punto de partida para elaborar un proyecto. Utilizar frameworks puede simplificar una tarea o proceso, de ahí que se trate de una de las herramientas habituales porque ayuda a ser más ágiles y productivos. Generalmente, los frameworks son usados por programadores porque permiten acelerar el trabajo y favorecer que este sea colaborativo, reducir errores y obtener un resultado de más calidad.

1. **¿Qué es una librería?**

Conjunto de archivos de código para facilitar la programación ya que proporciona funcionalidades comunes que ya han sido resueltas para poder emplearlas en el código sin necesidad de repetirlas, disminuyendo también los errores y el tiempo de desarrollo.

1. **¿Qué es BootStrap?**

Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario. Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso.

1. **¿Qué es Tailwind?**

Es un framework CSS que permite un desarrollo ágil, basado en clases de utilidad que se pueden aplicar con facilidad en el código HTML y unos flujos de desarrollo que permiten optimizar mucho el peso del código CSS. Es menos robusto, pero más rápido que bootstrap.

1. **¿Qué es Laravel?**

Es un framework de PHP para ayudarnos en desarrollo sobre aplicaciones escritas en este lenguaje de programación. Nos ayuda en muchas cosas al desarrollar una aplicación, por medio de su sistema de paquetes y de ser un framework del tipo MVC (Modelo-Vista-Controlador) lo cual facilita la diferenciación de las partes de un proyecto.

1. **¿Qué es NodeJS?**

Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables. Se emplea en backend.

1. **¿Qué es VUE.JS?**

Es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Creado por Evan You, de código abierto para crear interfaces de usuario (UI) y aplicaciones de una sola página. Este marco utiliza alto desacoplamiento, lo que permite a los desarrolladores crear progresivamente interfaces de usuario (UI).

1. **¿Qué es REACT?**

Es una librería open source de JavaScript para desarrollar interfaces de usuario. Fue lanzada en el año 2013 y desarrollada por Facebook, quienes también la mantienen actualmente junto a una comunidad de desarrolladores independientes.

1. **¿Qué es ANGULAR?**

Angular es un Framework de JavaScript de código abierto escrito en TypeScript. Su objetivo principal es desarrollar aplicaciones de una sola página. Google se encarga del mantenimiento y constantes actualizaciones de mejoras para este framework. Se divide en componentes que son fragmentos de código que define una clase que contiene la lógica y los datos de la aplicación. Un componente por lo general define una parte de la interfaz de usuario.

1. **¿Qué es una API?**

Interfaz de programación de aplicaciones. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas. Define la comunicación entre un módulo de software con otro.

1. **¿Cuánto tiempo tarda una persona en aprender un lenguaje de programación?**

Se estima entre 3 a 6 meses, pero depende según variables como el tiempo de desempeño, los conocimientos previos, la modalidad de aprendizaje, si tiene hábitos autodidactas, si es bootcamp, entre otros.

1. **¿Qué es un STACK en desarrollo?**

Un stack en desarrollo es un paquete de software que agrupa el sistema operativo de un sitio, sus servidores web, los marcos de programación, frameworks y entre otras herramientas comprendidas por el backend o el frontend, de manera que la descarga e implementación simultánea de estos componentes se dé fácilmente y se simplifique el trabajo del programador.

1. **¿Qué es un desarrollador FULLSTACK?**

Dado que los stacks pueden dividirse en front-end y en back-end, un programador full-stack es aquel que tiene la capacidad de trabajar en los dos ámbitos del desarrollo web.

1. **¿Qué es un paradigma de programación?**

Los paradigmas de programación son los diferentes estilos de programación documentados y adecuados a cada lenguaje que llevarán a dar solución al problema o requerimiento computacional. En algunos lenguajes de programación solamente existe un paradigma, mientras en otros existen múltiples, es decir, existen diferentes métodos de programar con diferentes estilos, donde varían la estructuración y organización de las tareas a llevar a cabo, las órdenes que se pueden implementar y su sintaxis.

1. **¿Cuáles son los lenguajes de programación más demandados en el mercado colombiano?**

Los 5 lenguajes de programación más usados en 2021 en orden de relevancia fueron JavaScript, Python, TypeScript, Java y PHP.

1. **¿Cuáles son los lenguajes de programación más demandados en el mercado Internacional?**

Según el ranking elaborado por DevJobsScaner (2022), donde se evalúan los lenguajes de programación solicitados en las ofertas de empleo relacionadas con el desarrollo web y el desarrollo de software, se concluyó que los lenguajes de programación más demandados en orden descendente de importancia son JavaScript, TypeScript, Python, C#, PHP, C y C++, Ruby y Go.

Mientras tanto, el portal de LinkedIn (2022) expuso que los lenguajes de programación más solicitados por los reclutadores son Java, JavaScript, Python, SQL y Go.

1. **¿Cuánto puede ganar un desarrollador FrontEnd?**

En el mercado colombiano el promedio salarial anual es de $ 48.000.000 (aproximadamente $ 10.000 USD), ubicándose en un rango de $ 36.000.000 ($7.550 USD) para los cargos de nivel inicial y de hasta $ 84.000.000 ($ 17.600 USD) al año para profesionales con mayor experiencia.

Por otro lado, según el portal indeed, en otros países como en Estados Unidos la base salarial media es de $90.982 USD anualmente, sin embargo, en empresas como Meta, Cisco Systems o eBay se pueden alcanzar salarios superiores a $ 190.000 USD.

1. **¿Cuánto puede ganar un desarrollador BackEnd?**

En Colombia un desarrollador BackEnd gana anualmente en promedio $ 54.000.000 COP ($11.313 USD), manteniéndose en un rango entre $36.000.000 COP ($7.540 USD) a $79.200.000 COP ($16.592 USD) según el nivel de experiencia del profesional.

Mientras tanto en Estados Unidos la base salarial media es de $93.463 USD anualmente, llegando a alcanzar salarios superiores a los 200.000 USD en empresas como Twitter.

1. **¿Cuáles son los requisitos para ser desarrollador FrontEnd?**

* Contar con formación académica en ingeniería informática, programación y afines.
* Dominar HTML5, CSS, JavaScript y frameworks ngularJS, ReactJS o VueJS.
* Tener conocimientos y experiencia en Rest APIs y Node JS.
* Experiencia metodologías agile (Scrum)
* Debugging y testing E2E.
* WordPress y otros CMS como Graph CMS.

1. **¿Cuáles son los requisitos para ser desarrollador Backend?**

* Tener formación académica en ingeniería de sistemas, ciencias de la computación o afines.
* Contar con habilidades en el análisis, desarrollo y mantenimiento de bases de datos.
* Comprender criterios de diseño para construir una arquitectura operativa.
* Diseñar plantillas, temas e interfaces.
* Dominio de manejo de bases de datos (Oracle, MySQL) y tecnologías SQL y ORM.
* Dominar lenguajes de scripting (Java, JavaScript, Python, C++, C#).
* Tener experiencia en desarrollo de aplicaciones web con frameworks relevantes (Django, Struts, Spring, entre otras)
* Estar apto para atender incidencias de integración.

1. **¿Qué es el control de versiones?**

El sistema de control de versiones se encarga de dar seguimiento a todos los cambios y ediciones que se ejecuten en el código, tomando una instantánea de los archivos. Esta instantánea puede ser recuperada posteriormente, con el fin de evitar que el programador se vea en la necesidad de tener varias copias de seguridad.

1. **¿Qué es GIT?**

GIT es el sistema de control de versiones distribuido que se ha convertido en el estandarizado mundial para esta labor, pues se cuenta con un repositorio de control de versiones completo que copia el proyecto local. Es diferente del control de versiones centralizado, pues no se debe sincronizar el código con un servidor antes de crear nuevas versiones, sino que se cuenta con una copia del repositorio que esta sincronizada con la copia del servidor. Por ello, en GIT es posible trabajar fácilmente de forma remota y sin conexión.

1. **¿Qué es GITHUB?**

GitHub es una plataforma que ofrece servicios de alojamiento para crear repositorios usando el sistema de control de versiones GIT y que el código sea guardado de forma segura en la nube. De esta manera, se puede centralizar el contenido del repositorio y otros desarrolladores pueden colaborar en la organización del proyecto en tiempo real.

1. **¿Qué es una rama de desarrollo?**

Las ramas de desarrollo son usadas por los sistemas de control de versiones. Son copias de los archivos de un programa pero que, aunque se mantienen aisladas del directorio principal, siguen estando dentro del proyecto. En otras palabras, son copias del código de un proyecto que siguen conectadas a este y dependen la una de la otra.

1. **¿Cuáles son las ramas del desarrollo?**

Las ramas del desarrollo en el control de versiones se pueden clasificar en la rama principal de desarrollo o también llamada rama máster y en las ramas nuevas que son creadas como copias de los archivos del programa, las cuales pueden ser fusionadas con la rama principal.

1. **¿Qué es NPM?**

NPM o Node Package Manager, es un gestor de paquetes de NodeJS, un entorno que permite ejecutar JavaScript dentro y fuera de un navegador web para poder utilizarlo en sistemas de escritorio y servidores web. Es decir, NPM permite la instalación de paquetes JavaScript para ejecutar los códigos que requiera el programador.

1. **¿Qué es Webpack?**

Es definido como un empaquetador de módulos de JavaScript, pues es una herramienta que partiendo de diferentes módulos de códigos que están separados y son necesarios para que un proyecto funcione, permite la compilación de los mismos para generar un archivo único.

1. **¿Qué es Prettiers?**

Es un formateador automático de código, pues otorga una configuración determinada sobre la guía de estilo de código para múltiples lenguajes y frameworks. Este análisis y transcripción del código se da con el fin de tener un estilo de código consistente y legible, y por tanto, un espacio de trabajo más limpio.

1. **¿Qué es EsLint?**

Se trata de un instrumento o herramienta de JavaScript para ayudar al profesional a identificar y solucionar problemas con su sintaxis, pues tal como Prettiers, se encarga de analizar el código y detectar posibles problemas de escritura por medio de patrones y en caso de estar a su alcance, se encargará de corregirlos automáticamente. Sin embargo, a diferencia de Prettiers, EsLint no formatea ni reescribe el código cada vez que se almacena un archivo.

1. **¿Qué es MVC?**

Se define como el Modelo Vista Controlador, un estilo de arquitectura de software que se divide en tres componentes diferenciados:

1. Modelo: Hace referencia a la representación de los datos, la lógica del sistema y los mecanismos de persistencia.
2. Vista: Comprende la información que se envía al cliente y los mecanismos de interacción con este, es decir, la interfaz del usuario.
3. Controlador: El tercer componente juega el papel de intermediario entre el Modelo y la Vista, regulando la información que se intercambia entre ellos y la transformación de los datos a las necesidades de cada uno de los anteriores componentes.
4. **¿Qué es la metodología SCRUM?**

Esta metodología de trabajo es aplicada para optimizar el proyecto final entregado a los clientes. Este framework permite la colaboración con otros equipos, donde su fundamento principal está en el aprendizaje, el mejoramiento continuo y en la adaptabilidad de los equipos a variables y factores altamente fluctuantes, para lo cual, incluye herramientas, reuniones y funciones que ayudan a que los equipos colaboren entre sí y puedan estructurar su trabajo eficientemente

El proceso de SCRUM se divide en tres fases principales:

1. La identificación de los roles y cada una de sus responsabilidades (Dueño del producto, Scrum Máster y Miembros del equipo de desarrollo)
2. Se define el sprint o tiempo prefijado para trabajar en el proyecto a entregar.
3. Se determinan las herramientas que utilizará cada miembro.
4. **¿Qué es la metodología AGILE?**

Mientras Scrum se considera como un marco de trabajo, la metodología AGILE se concibe como una mentalidad o filosofía para optimizar el trabajo y el valor agregado en proyectos de desarrollo de software. En esta filosofía se fomenta la creación de productos que respondan a las necesidades específicas de los clientes, por lo cual, se deben tomar en consideración los factores variables en torno a los requerimientos de los clientes y el mercado, con el fin de adaptar los productos ofrecidos y continuar agregando valor.

1. **¿Qué es TESTING?**

Es una metodología que consiste en la aplicación de un conjunto de pruebas consistentes y rigurosas para verificar cada etapa de desarrollo del código, con el fin de detectar tempranamente inconsistencias en el funcionamiento del programa.

1. **¿Qué son los patrones de diseño?**

Se definen como modelos y esquemas determinados que describen y guían la resolución de problemas comunes en el proceso del desarrollo del software, el diseño de interacción e interfaces, sin importar el lenguaje de programación que se esté utilizando. Dentro de los objetivos de estos patrones se encuentra la estandarización del lenguaje entre programadores, la minimización del tiempo empleado en solucionar problemas y la creación de un código reusable.

1. **¿Qué es el diseño UML?**

Hace referencia al lenguaje de modelado unificado, el cual permite a representación visual de los objetos, estados y procesos dentro de un sistema. Así mismo, este lenguaje puede ser usado como modelo para definir una arquitectura de información estructurada en diferentes proyectos y ayuda a que los programadores describan los sistemas en una presentación fácilmente comprensible para personas ajenas al campo del desarrollo web y la programación.

Sus diagramas son utilizados para representar objetos individuales, relaciones de jerarquía, acciones, comportamiento e interacción entre distintos elementos.

1. **¿Cuáles son los roles en el desarrollo de software?**

Para conformar un equipo de desarrollo de software eficiente se requieren 7 roles fundamentales

1. Dueño del producto: Responsable del producto, de su creación de valor y del cumplimiento de requisitos.
2. Project Manager: Se encarga del desarrollo, organización y entrega del producto.
3. Diseñadores UX y UI: Deben garantizar un producto final enfocado en el usuario y en su experiencia final en términos de usabilidad, funcionalidad y rendimiento.
4. Analista Comercial: Dan apoyo a las necesidades comerciales del cliente.
5. Desarrolladores de Software: Encargados de programar el proyecto final cumpliendo tareas según su nivel de experiencia. Se les conoce como frontend, backend y full stack
6. Team lead y tech lead: Están a cargo del rendimiento general, la organización del equipo y de la supervisión de los requerimientos técnicos, tanto del proceso del desarrollo de software como de elementos complementarios.
7. Scrum máster: Se asegura de que el equipo siga metodologías ágiles.
8. **¿Cuál es el camino que deseas seguir? ¿Desarrollador FrontEnd o Desarrollador BackEnd?**

El camino que se encuentra en mayor congruencia con mis preferencias personales es el de especializarme como **Desarrollador Frontend**, pues es de mi interés profundizar mis habilidades en el desarrollo de interfaces gráficas intuitivas que faciliten la comunicación con el usuario, mientras también fortalezco mi capacidad creativa en el diseño web.

**Bibliografía**

* Qué es un project roadmap y cómo crear uno <https://asana.com/es/resources/project-roadmap>
* ¿Que aprender para ser desarrollador web? <https://victorroblesweb.es/2017/06/23/que-aprender-para-ser-desarrollador-web/>
* ¿Qué hace un Desarrollador Web y Cómo Convertirse en Uno? <https://www.hostinger.co/tutoriales/como-ser-programador-web>
* 5 diferencias entre un programador junior vs senior <https://keepcoding.io/blog/diferencias-entre-un-programador-junior-y-un-programador-senior/>
* HTML: Lenguaje de etiquetas de hipertexto <https://developer.mozilla.org/es/docs/Web/HTML>
* Sass Introduction <https://www.w3schools.com/sass/sass_intro.php>
* BEM — Introduction <https://getbem.com/introduction/>
* Stacks de Desarrollo: Qué son y cómo elegirlos <https://www.tecnologias-informacion.com/stacksdesarrollo.html>
* Cuáles son los tipos de Stacks en programación <https://www.hostingplus.pe/blog/cuales-son-los-tipos-de-stacks-en-programacion/>
* Paradigmas de la programación (Que es y tipos) <https://okhosting.com/blog/paradigma-de-programacion/>
* 5 lenguajes de programación más usados durante el 2021 <https://www.imaginacolombia.com/articulos/5-lenguajes-de-programacion-mas-usados-durante-el-2021>
* Los lenguajes de programación más demandados en 2022 <https://hipertextual.com/2022/07/lenguajes-programacion-mas-demandados-2022>
* Programming Languages Most in Demand Right Now | LinkedIn <https://www.linkedin.com/pulse/programming-languages-most-demand-right-now-michael-spencer-/>
* Salario para Desarrollador Frontend en Colombia. <https://co.talent.com/salary?job=Desarrollador+Frontend>
* Back end developer salary in United States https://www.indeed.com/career/back-end-developer/salaries
* Front end developer salary in United States. <https://www.indeed.com/career/front-end-developer/salaries>
* Salario para Desarrollador Backend en Colombia. <https://co.talent.com/salary?job=Desarrollador+backend>
* ¿Qué hace un Desarrollador Backend? <https://www.freelancermap.com/blog/es/que-hace-desarrollador-backend/>
* ¿Qué hace un Desarrollador Front-End? <https://www.freelancermap.com/blog/es/que-hace-desarrollador-front-end/>
* ¿Qué es el control de versiones? Microsoft Learn <https://learn.microsoft.com/es-es/devops/develop/git/what-is-version-control>
* ¿Qué es Git? Microsoft Learn <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>
* Qué es GitHub y cómo empezar a usarlo de forma correcta. Platzi. <https://platzi.com/blog/que-es-github-como-funciona/>
* Git - ¿Qué es una rama? <https://git-scm.com/book/es/v2/Ramificaciones-en-Git-%C2%BFQu%C3%A9-es-una-rama%3F>
* ¿Qué es NPM? <https://lenguajejs.com/npm/introduccion/que-es/>
* What is Webpack? | What is Webpack? <https://www.educba.com/what-is-webpack/>
* What is Webpack? - Learn ReactJS <https://learnreactjs.org/what-is-webpack/>
* ¿Qué es Prettiers? <https://www.freecodecamp.org/espanol/news/que-es-prettier/>
* Find and fix problems in your JavaScript code - EsLint - Pluggable JavaScript Linter <https://eslint.org/>
* Modelo vista controlador (MVC) <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
* ¿Qué es una metodología ágil? <https://platzi.com/clases/1750-scrum/24278-que-es-una-metodologia-agil/>
* ¿Cómo funciona la metodología Scrum? Qué es y sus 5 fases <https://platzi.com/blog/metodologia-scrum-fases/>
* ¿Qué es el TESTING de software? <https://craft-code.com/que-es-el-testing-de-software/>
* Design patterns: ¿Qué son los patrones de diseño y para qué sirven? <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-son-los-design-patterns/>
* UML: lenguaje unificado de modelado orientado a objetos <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/uml-lenguaje-unificado-de-modelado-orientado-a-objetos/>
* Roles Fundamentales en un Equipo de Desarrollo de Software <https://projectcor.com/es/blog/roles-fundamentales-en-un-equipo-de-desarrollo-de-software/>