

¡Hola!

Si estás en esta instancia de la carrera seguramente te estés preguntando cómo es el mundo laboral para el cual te estás formando.

¡Queremos acompañarlos en ese proceso!

Nuestro primer objetivo es que conozcas posibles puestos de trabajo en los que te podrías desempeñar a futuro y también, profundizar cómo se posiciona cada uno de ellos en el mercado laboral.

Deseamos que esto te ayude a decidir cuál de esas posiciones, tareas son las que más te motivan.

### **Posibles caminos en tu recorrido profesional**

El objetivo de este módulo es que descubras las posibilidades que tu carrera ofrece. Vamos a conocer posibles puestos de trabajo en los que te podrías desempeñar a futuro y también, vamos a profundizar cómo se posiciona cada uno de ellos en el mercado laboral.

#### **¿Te interesa el Front End?:**

¿Sabías que los Frameworks de Javascript son la herramienta más demandada por la industria IT? React - Angular - Vue. Todos los frameworks van a crear un tipo de app web SPA y reactiva ¿Qué es? Una app que no recarga la pantalla en ningún momento, es totalmente reactivo e instantáneo.

Si te interesa crecer dentro del desarrollo Front End o especializarte en los frameworks relacionados al área, te contamos en qué tipo de proyectos podes participar, qué tipo de plataformas podes desarrollar y qué oportunidades de aprendizaje aporta a tu carrera profesional.

La decisión que tomes va a variar de acuerdo a tus gustos, y a la arquitectura que más se adapte a tus habilidades y curva de aprendizaje.

Sabemos que ya conoces React y por eso queremos acercarte otros posibles caminos para especializarte en el mundo Front End:

## 1. Angular (orientado a la creación de componentes):

Es una herramienta para construir tu sitio/app web con JS, permite que tengas todas las herramientas unificadas, sin necesidad de instalar paquetes o softwares (módulos http, router, etc.), marca casi todos los límites y lineamientos. Su mantenimiento puede extenderse a largo plazo ya que es de Google, y eso trae una gran ventaja (Angular 2 en adelante, no confundir con AngularJS). En resumen, es un framework completo a diferencia de React o Vue, es decir, que no es necesario elegir qué agregarle para continuar creciendo, ni preguntarnos qué bibliotecas de enrutamiento implementar, ya que lo trae todo incorporado.

Angular es más demandado en Europa, y algunos estados de EEUU, mientras React se utiliza mucho más en LATAM. Es decir que dependiendo del país donde proyectes vivir, quizá te convenga más una herramienta que otra ¿De qué depende que empresa necesite a un Desarrollador React o Angular? Simplemente varía de acuerdo a la tecnología que cada compañía elija implementar, es una cuestión organizacional. Brinda a una empresa la posibilidad de tener un sitio web escalable con un nuevo diseño. Para una empresa de software, utilizar este framework tiene varios beneficios y usos, entre los que se encuentran: crear foros, portales de negocios, plataformas de crowdfunding, mercados, listados de propiedades, etc.

Entre las empresas que han hecho uso de este framework para sus páginas y aplicaciones se encuentran: GitHub (foro), Google, Forbes, Indiegogo, BMW (calculadora de precios, buscador de ofertas), portal de desarrolladores de Deutsche Bank, Crunchbase, Nike, Xbox, Udacity , YoutubeTV, Firebase, AT&T y Adobe.

¿Cuándo es recomendable utilizarla?: Aplicaciones de una sola página Las aplicaciones de una sola página no son más que

aquellas webs donde todo el contenido es mostrado en una página con navegación fluida, sin que haya que saltar a una segunda página. Estas son por lo general ideales para aplicaciones o sitios webs con poco contenido, además de ser más fáciles de utilizar para el usuario. Uno de los mejores frameworks para crear estas páginas es Angular. Para poder implementarla, es fundamental que quien desarrolle dicha plataforma tenga incorporado Typescript (lenguaje superconjunto de Microsoft), que permite añadir funcionalidades a JS, lo que permite crear aplicaciones grandes y de forma más simple.

¡Dato interesante! ? Si ya tenés una interacción cercana con entornos enterprise (frameworks como Spring, .NET, C#), y optás por aprender Angular, será una gran decisión debido a que la combinación de estos frameworks, facilita muchísimo el aprendizaje y desarrollo de proyectos, por ende, podrás incorporarlo de forma muy rápida y fácil.

## 2. Vue:

Hace referencia a la vista, a la interfaz, al igual que React cuando creamos interfaces en un navegador.

Es un framework progresivo, lo que significa que cuando creamos una app, podemos empezar con proyectos pequeños como si estuvieses usando una biblioteca, pero a medida que este crece podemos agregarle complementos, códigos, módulos y paquetes que podrían transformarlo en un framework completo como Angular. Toma las mejores características de React y Angular para construir aplicaciones web.

Es una herramienta flexible y productiva, ya que al ser tan fácil de implementar, podemos agregarlo a proyectos que ya están funcionando y que fueron escritas con otras bibliotecas como JQuery u otras. De esta manera permite comenzar a crear de manera ágil y rápida. Es muy sencillo comenzar a utilizarlo, ya que no es necesario incorporar conceptos como Typescript o JSX, sino que simplemente se puede utilizar con HTML, CSS y Javascript, los estándares que conocen todos los desarrolladores front end. Su rendimiento es prácticamente igual al de React y Angular, con la diferencia de que es muy fácil de aprender ya que no necesitas conocer otras herramientas más complejas, tiempo que te ahorras de aprendizaje para comenzar a trabajar.

Es muy popular en el desarrollo de aplicaciones open source, ya que no depende de Google ni de Facebook. Por ese motivo es muy elegido entre profesionales que prefieren que su tecnología favorita no dependa de otras empresas, sino que se determina por un

sólo desarrollador y el apoyo de la comunidad, quienes tienen la libertad de decidir hacia donde va dirigido, y de qué manera mejorarlo. Al permitir esta amplitud de contexto de trabajo, este framework es recomendable para programadores junior o también, para profesionales que hagan proyectos chicos, propios o freelance.

A continuación detallamos un cuadro comparativo para que conozcas cómo están posicionados cada uno de estos frameworks, de acuerdo a las variables que ves a continuación, junto con una “calificación”. Esta última, está organizada de acuerdo a la posición en la que se encuentra cada uno (1: Muy bueno/Recomendable - 2: Bueno - 3: intermedio-malo).

	React	Angular	Vue
Curva de aprendizaje	3	2	1
Flexibilidad	2	3	1
Limpieza (sistemas de vista, plantillas, modelo de servicios, etc.)	3	1	2
Limpieza (sistemas de vista, plantillas, modelo de servicios, etc.)	3	1	2
Robustez (permite hacer app a nivel empresarial o más complejas)	2	1	3
Velocidad	1	3	2
Componentes	1	1	1
Reactividad (cambio en un objeto y que pueda identificar rápidamente a la vista)	1	1	1
Lenguaje de plantillas/Templates	3	1	2
Comunicación e/ componentes	1	1	1
Formularios y validación	1	2	3
Servicios y HTTP	2	1	3
Routing	3	2	1
Facilidad de uso	3	2	1
Empleo y demanda laboral (en LATAM)	1	2	3

### [Demanda de frameworks JS 2021](#)

Dejamos una imagen y enlace al pie de página para que conozcas la posición en la cual están posicionados de acuerdo a la demanda laboral:

## Rankings

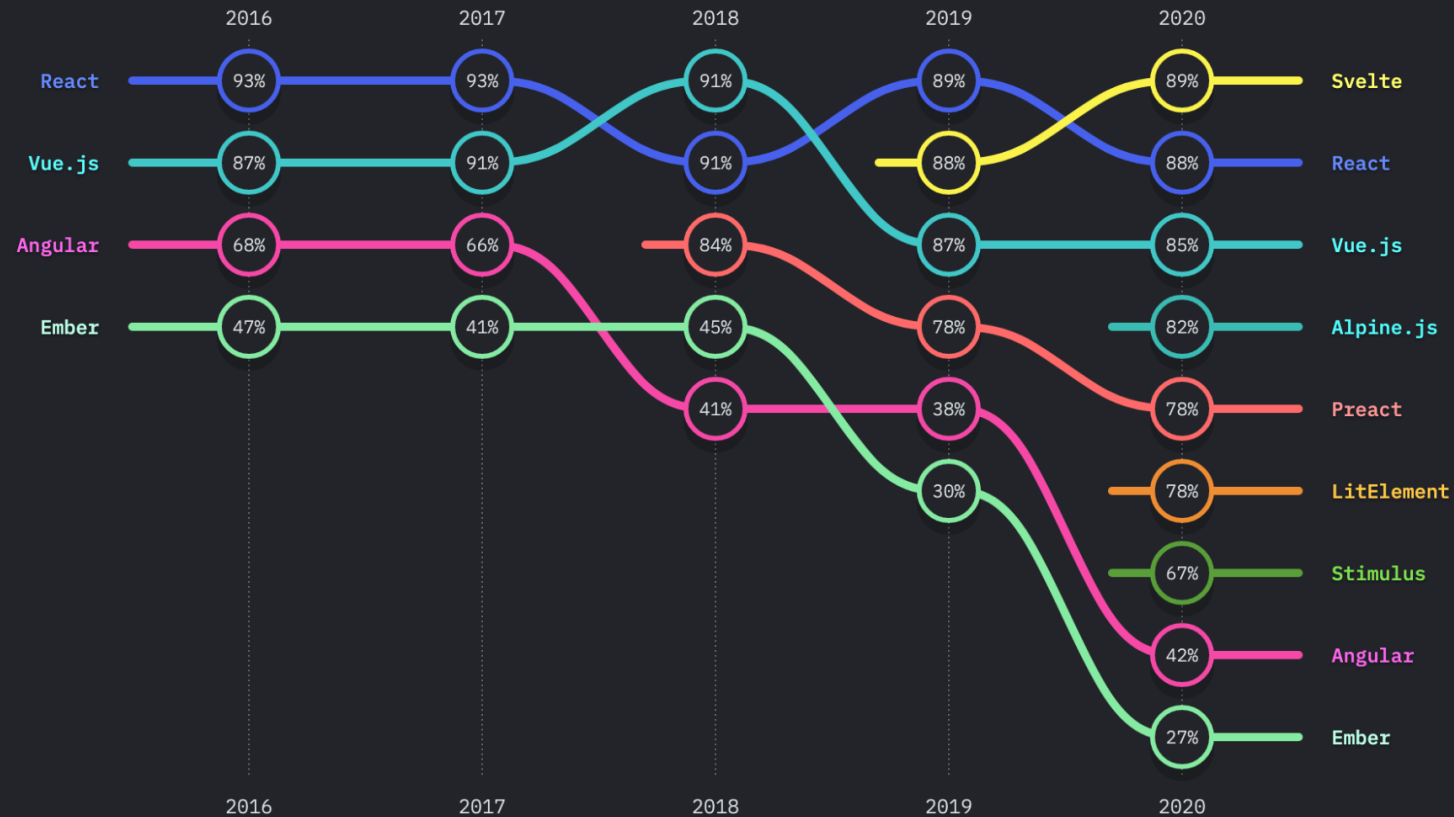
Satisfaction

Interest

Usage

Awareness

Satisfaction, interest, usage, and awareness ratio rankings.



Technologies with less than 10% awareness not included. Each ratio is defined as follows:

- Satisfaction:  $\text{would use again} / (\text{would use again} + \text{would not use again})$
- Interest:  $\text{want to learn} / (\text{want to learn} + \text{not interested})$
- Usage:  $(\text{would use again} + \text{would not use again}) / \text{total}$
- Awareness:  $(\text{total} - \text{never heard}) / \text{total}$

## ¿Te interesa el Back End?:

### 1. NodeJS:

Es un código abierto de JavaScript para la creación de aplicaciones de red. Permite a los desarrolladores programadores ejecutar el código en el lado del servidor y generar aplicaciones web de forma altamente optimizada. Esta herramienta suele ser un excelente complemento para hacer desarrollo Back End. Si manejas JS y sus demás frameworks, NodeJS permitirá programar con este lenguaje fuera del navegador.

Esta es una novedad que los últimos años fue ganando protagonismo en el mundo de la tecnología y principalmente en programación. Lo que permite Node, es quitarle el motor al navegador y de esta manera posibilita que el trabajo de los desarrolladores pueda hacerse desde cualquier sitio.

Gracias a esto permite generar APIs, crear servidores, usar paquetes express y separar las responsabilidades. Su curva de aprendizaje es mucho más sencilla que React y Angular, y además, abre la posibilidad de ampliar tu perfil a Programador Web Full Stack, ya que si participas de un proyecto y usas, por ejemplo, React y Node, esa app o sitio web, la desarrollaste tanto desde el lado front como back end.

Principales ventajas de Node.js:

- Velocidad (se ejecuta en el motor JS de Google).
- Paradigma de E / S sin bloqueo.
- Transmisión de datos (solicitudes y respuestas HTTP como un solo evento).
- Mismo lenguaje (javascript), tanto en el cliente como en el servidor.
- Consultas de bases de datos unificadas (formato JSON).
- Codificación fácil y rápida.

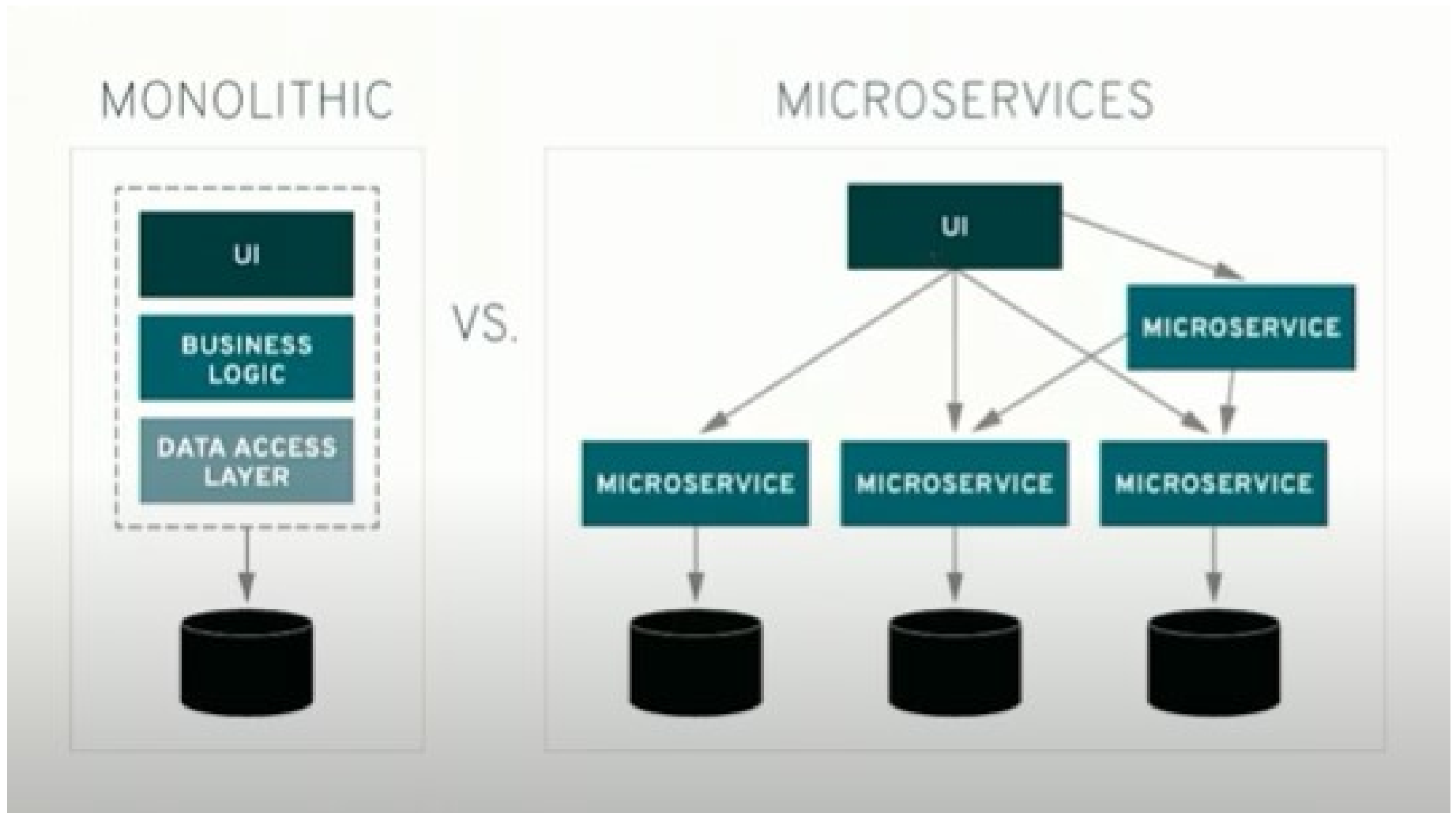
- Buena gestión de paquetes gracias a NPM (y Yarn)
- Capacidad del servidor proxy.
- Ciclos de desarrollo rápidos.
- Lógica de negocio en el servidor.

Empresas que eligen NodeJs y por qué:

- Netflix: Esta gran compañía especializada en streaming eligió Node.js por tres razones: la facilidad con que los desarrolladores de JS lo aprenden, el hecho de que eliminó la necesidad de escribir código dos veces y el alto rendimiento. También con Node.js disminuyeron los tiempos de compilación y habilitaron la personalización del usuario. Como resultado, el equipo de Netflix ha mejorado la velocidad de carga de la aplicación en un 70% .
- Paypal: Uno de los factores más importantes para la decisión fue la posibilidad de derribar la barrera del idioma entre la ingeniería: el uso del mismo idioma en el servidor y en el lado del cliente mejoró la comunicación interna y la comprensión del producto en su conjunto. Según la empresa, su aplicación Node.js permite reducir en un 40% los archivos y un 33% el código. Así, el tiempo de respuesta se mejoró un 35%, lo que resultó en que las páginas se cargaban 200 ms más rápido.
- Uber: Node.js es ideal para sistemas distribuidos que llevan a cabo muchas solicitudes de red con el objetivo de procesar rápidamente grandes cantidades de información. Además dispone de un análisis de errores muy intuitivo y con rápido despliegue de código y dispone constante mejora tecnológica debido a la comunidad de código abierto. En consecuencia, Uber puede procesar más de 2 millones de llamadas de procedimiento remoto (RPC) por 1 segundo.
- LinkedIn: El backend de aplicación móvil de LinkedIn fue creado en Node JS, así lograron un rendimiento mucho mejor, con la aplicación ejecutándose hasta 20 veces más rápido y una mejor conexión entre el frontend y el backend.
- Ebay: Eligieron Node.js, todavía bastante inmaduro en ese momento, por la necesidad de hacer la aplicación de Ebay en tiempo real como sea posible. Además, querían que mantuviera una conexión activa con los servidores ya que, su objetivo principal era organizar una gran cantidad de servicios específicos de eBay que mostraban información en la página, por ejemplo, el manejo de operaciones enlazadas a E / S.

2. Herramientas que potencian tu perfil: **Microservicios**. Con el pasar del tiempo, cada vez son más las empresas que requieren que sus desarrolladores back end manejen Microservicios.

Esta herramienta permite separar las “capas” de la página web en distintos servicios en lugar de tenerlo en uno sólo (monolithic). Sirven para desarrollar softwares de gran calidad y robustez de una manera más rápida y eficiente a fin de conseguir ciclos de desarrollo más cortos y por ende, lograr que las aplicaciones estén en condiciones de salir al mercado en un plazo más corto de lo habitual.





## Nuestro primer objetivo - parte 2

### Entrevista Técnica para un Desarrollador de Microservicios:

Compartimos algunas posibles preguntas para atravesar un proceso de selección como desarrollador de Microservicios:

- Defina la arquitectura de Microservicios: cuando hace esta pregunta piden una definición exacta de la misma. Es importante prepararse para responder esta pregunta con documentación oficial.
- ¿Cuáles son las ventajas de implementar la arquitectura de microservicios?
- ¿Cuáles son los retos que trae implementarla?
- En la arquitectura ¿Cómo puede resolverse el problema de los parámetros de configuración de microservicios y sus instancias?
- En una arquitectura microservicios son comunes las llamadas servicio ? servicio ¿Qué mecanismos pueden implementarse para evitar posibles fallos? Lo que buscan evaluar con esta pregunta es la capacidad de garantizar la resiliencia.
- Explique cómo funciona un servicio de registro y descubrimiento así como la importancia para la infraestructura.
- Al utilizar esta herramienta, la escalabilidad es una característica deseada, como resultado de su aplicación pueden estar operativas múltiples instancias de varios microservicios. Suponga que una instancia del microservicio A debe llamar a la otra instancia de microservicio B ¿Qué estrategias serían las correctas seguir para que una instancia A llame de forma “adecuada” a la B? Buscan complementar la pregunta anterior,
- La interacción de los servicios de infraestructura con componentes externos (llamemos legacy) tiene lugar en muchos escenarios. Es común que los sistemas legacy estén desarrollados con tecnologías obsoletas o poco eficientes ¿Qué solución técnica podría aplicarse para evitar implementar en microservicios dichos sistemas? Lo que se busca en esta pregunta, es evaluar la capacidad de aplicar Capa de Interrupción.
- Explique cómo funciona Bulkhead y su relación con la resiliencia. Una guía para responder esta consigna, es pensar la infraestructura con y sin Bulkhead ¿Que diferencia hay entre uno y otro? ¿Dónde se pone en práctica este patrón?
- ¿Qué es la transaccionalidad distribuida y que relación tiene con los microservicios? Conceptos disparadores para una correcta respuesta: unidad indivisible - atómica - consistente - fuente de datos - acciones compensatorias - Sagas.

---

¿Te interesa explorar otros caminos además de los ya mencionados?

Te contamos qué otros puestos puedes ocupar aparte de Desarrollo (“Hands On”)

➤ **Arquitectura:** Quien desempeña este rol, traslada la idea del cliente al lenguaje técnico.

La definición de la arquitectura se trata de la introducción de la estructura, directrices de los aspectos técnicos de un proyecto. Por lo tanto se requiere de una figura dedicada a pensar en estos aspectos, es decir, alguien tiene que asumir la propiedad del proceso de definición de la arquitectura y esto es sin duda, parte de las competencias del Arquitecto.

Quien se desempeñe en este puesto, será un programador senior (más de 5 años de experiencia) especializado en algún lenguaje de programación. Es quien toma las decisiones técnicas, es decir, ya que son ellos los que deciden qué lenguaje de programación aplicar y están para resolver cualquier tipo de problema técnico.

➤ **Analista QA:** Muchos desarrolladores eligen apuntar su carrera al área de calidad, ya que los testeos también forman parte de sus tareas. En este caso, estarían dedicados de forma full time al desarrollo y ejecución de casos de prueba.

Cuando las empresas definen una vacante para este perfil, también evalúan si necesitan un profesional con experiencia previa en

programación. En caso que así sea, también van a delimitar su búsqueda a perfiles con manejo de algún lenguaje de programación en particular, ya que los testeos con los que va a trabajar también estarán basados en estas herramientas. Por ejemplo, si necesitan un Analista QA para crear casos de prueba con Java, probablemente busquen a un desarrollador que maneje ese lenguaje.

Por último, deberías evaluar si querés desempeñarte como Analista QA Manual o en la automatización de pruebas. De acuerdo a los conocimientos adquiridos en el track 1, los profesionales del área de QA recomiendan que te formes en las siguientes herramientas, dependiendo de la especialización que elijas seguir:

### **1. QA Manual - Testing I:**

TestRail, TestLink, XRail, ADO, Jira (todas herramientas de gestión). Postman, Swagger, SoapUI, Git, Kibana, MySQL, MongoDB (testing de microservicios).

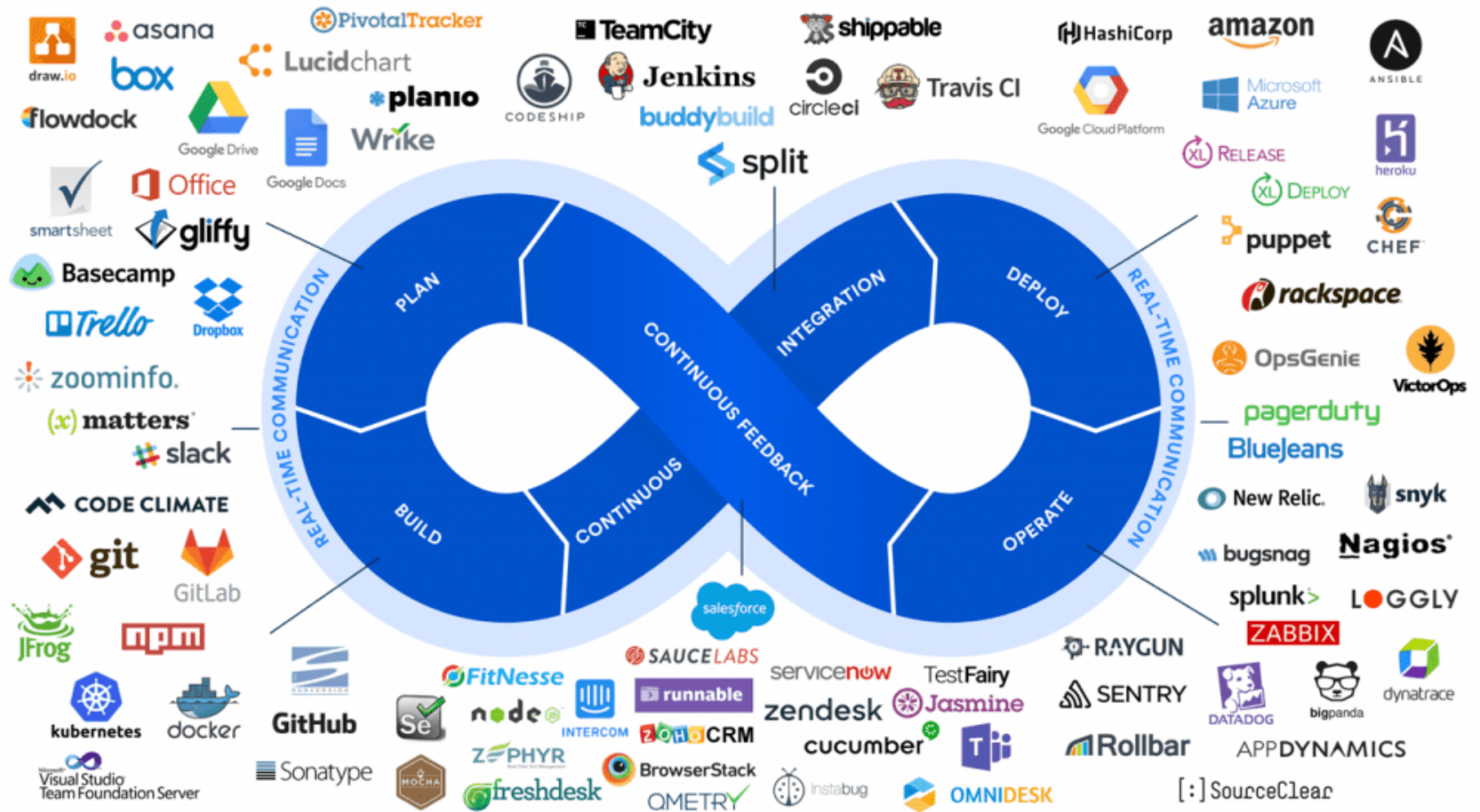
### **2. QA Automation - Testing II:**

- Algún lenguaje de programación (Java, .net, python)
- Selenium
- Appium
- Serenity
- Rest Assured
- Karate framework
- Git
- Jenkins
- Docker

➤ **Infraestructura:** : Los conocimientos adquiridos en esta carrera, también pueden brindarte la posibilidad de especializarte en esta área. Como es de público conocimiento, la tecnología y herramientas de trabajo IT evolucionan constantemente; por ese motivo es importante estar al tanto de los avances y cambios en esta área, ya que a pesar que te has capacitado en las tecnologías más recientes y modernas, necesitarás adaptarte a la dinámica del cambio y sus nuevas incorporaciones.

Entonces ¿Cómo puedo insertarme en el mundo de Infraestructura? El puesto en el que podrías desempeñarte a corto plazo, es como DevOps Junior ¿Cómo? Podrías participar en una de las facetas del proyecto, principalmente en las que corresponden a Desarrollo. Para profundizar tu experiencia, sugerimos que de a poco vayas incorporando experiencia en cada faceta; es decir, que cuanto más etapas conozcas más te acercas a ser un perfil semi-senior y senior.

A continuación dejamos una imagen que representa las etapas del proyecto, y muestra los conocimientos que necesitas para cada una:



Es importante tener en cuenta, que para insertarse en este puesto vas a implementar a diario las siguientes herramientas y tareas:

- Docker.
- Kubernetes.
- Git.
- Jenkins.
- Automatizar, programar y planificar procesos.

- Troubleshooting de errores.
- Arquitectura virtual: No es requisito excluyente manejar servidores de Cloud, aunque las posibilidades de inserción son más amplias cuando manejamos esta tecnología. Recomendamos profundizar sobre lo aprendido durante la carrera y aclararlo en tu CV y perfil de LinkedIn para concretar oportunidades laborales de tu interés.

Dejamos información adicional al pié de página, donde vas a encontrar la diferencia entre Diferencias entre Site Reliability Engineer vs. Ingeniero de software vs. Ingeniero en la nube vs. Ingeniero DevOps ¡Esperamos que sea de utilidad!

---

También existen posiciones en las que podés desarrollar habilidades blandas como liderazgo, comunicación, capacitación y planificación estratégica. Muchos profesionales que trabajan con hard skills, a veces buscan dar un giro a su profesión y optan por adquirir otras experiencias como estas.

Antes de entrar en detalle con cada uno, es importante destacar que todos tienen en común la implementación de metodologías ágiles (la más utilizada es Scrum), ya que independientemente de las tareas que se le asigne, todos ellos buscan la manera de administrar y optimizar los tiempos del proyecto.

#### **A continuación detallamos:**

- ⇒ **Project Leader:** También conocido como Developer Team, es quien representa al equipo de desarrollo. Es responsable de comunicar a otras áreas los avances o necesidades del proyecto ya que es un perfil que cuenta con amplia experiencia en su área. En la mayoría de los casos son programadores que participaron de varios proyectos y que por ese motivo, se encuentran en condiciones de representar un sector tanto para áreas externas como también para asesorar sus colegas con menor seniority y coordinar un grupo de trabajo.

- ⇒ **Product Owner:** Es quien representa al cliente dentro de la empresa y optimiza el valor del producto. Es decir, que es responsable de definir las características del producto, definir fechas de lanzamiento y de garantizar la mejora continua a fin de obtener un producto final que le de auténtico valor al proyecto.
- ⇒ **Project Manager:** Quien ocupe este puesto debe tener capacidad de liderazgo, de organización y de detección de problemas, ya que es el encargado de controlar y administrar los recursos necesarios para llevar adelante uno o más proyectos.

El objetivo de su función, es lograr que cada proyecto cumpla con sus expectativas finales, logrando la mayor rentabilidad posible y para eso debe proveer de los recursos necesarios para dicho fin. Por ese motivo, también es el encargado de hacer un análisis de mercado para definir un presupuesto en el cual se deberá invertir, tanto en profesionales que participan del proyecto como también, las herramientas/software necesarios para llevarlo a cabo.

---

#### [Differences between Site Reliability Engineer Vs. Software Engineer Vs. Cloud Engineer Vs. DevOps Engineer](#)

⇒ **Analista Funcional** Es el nexo entre el proyecto (área técnica) y el cliente. Elabora, planifica y hace un seguimiento del proyecto.

Releva las necesidades del cliente, está pendiente de los detalles técnicos y de la falta de recursos, como así también evalúa la viabilidad técnica y económica del desarrollo de las aplicaciones que se ejecutan. Utiliza herramientas para organizar la información como, entre ellas, base de datos.

Presta apoyo a los distintos usuarios; es decir, realiza una labor de asesoramiento y capacitación, con el fin de evitar cualquier problema que pueda surgir con los programas y obtener así el máximo rendimiento de los mismos.

Como podemos observar, existen algunas similitudes entre las posiciones mencionadas. Es común que se superpongan las características del puesto, ya que las habilidades blandas forman parte de los mismos. También puede ocurrir que algunas empresas fusionen más de un puesto en un sólo rol, aunque eso va a depender de su estructura y organigrama.

