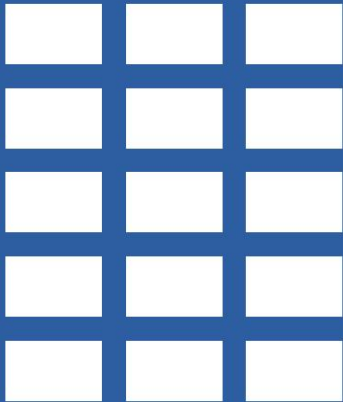


equipos completos,
No ingenieros





25 NO 23rd PI
suite 6314
Portland, Oregón 97210

Equipos Full Stack, no ingenieros

Este trabajo tiene licencia Creative Commons Atribución-No Comercial

-Licencia Internacional ShareAlike 4.0. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by-nc-sa/4.0/> o envíe una carta a Creative Commons, PO Box 1866, Mountain View, CA 94042, EE. UU.

Al compartir este contenido, por favor notifique
IT Revolution Press, LLC, 25 NW 23rd PI, Suite 6314, Portland, OR 97210

Producido en los Estados Unidos de América.

Diseño de portada e interior por Devon Smith.

Para obtener más información sobre IT Revolution, estas y otras publicaciones, descuentos especiales para compras de libros al por mayor o información sobre cómo reservar autores para un evento, visite nuestro sitio web en ITRevolution.com.



El Foro Empresarial DevOps 2019 fue patrocinado por XebiaLabs.

Prefacio

En mayo de este año, se celebró el quinto Foro Empresarial DevOps anual en Portland, Oregón. Como siempre, los líderes y expertos de la industria se reunieron para discutir los temas de vanguardia de la comunidad DevOps Enterprise y para elaborar orientación que nos ayude a superar y superar esos obstáculos.

Este año, el grupo profundizó en los problemas que recién habíamos comenzado a analizar en años anteriores, brindando orientación paso a paso sobre cómo implementar el paso del proyecto al producto y cómo hacer que DevOps funcione en entornos cibernéticos a gran escala. -físico sistemas, e incluso una mirada más detallada a la realización de Dojos en cualquier organización. También abordamos cambios culturales y de procesos, como romper con los viejos procesos de gestión de cambios y desacreditar el mito del ingeniero completo. Y, por supuesto, nos sumergimos en la cuestión constante sobre la seguridad en las tuberías automatizadas.

Como siempre, los temas de este año se esfuerzan por abordar los problemas, inquietudes y obstáculos que son más relevantes para las organizaciones de TI modernas en todas las industrias. Después de todo, toda organización es una organización digital.

Los artículos del Foro de este año (junto con nuestro archivo de artículos de años anteriores) son un activo esencial para la biblioteca de cualquier organización, ya que fomentan el aprendizaje continuo que es esencial para el éxito de una transformación de DevOps y ganar en el mercado.

Un agradecimiento especial a Jeff Gallimore, nuestro coanfitrión, socio y cofundador de Excella, por ayudar a crear una estructura para los dos días y las semanas siguientes para ayudar a todos a mantenerse concentrados y productivos. Un agradecimiento adicional al patrocinador del Foro de este año, XebiaLabs. Y lo más importante, un enorme agradecimiento a los participantes del Foro de este año, que aportan su valioso tiempo y experiencia y siempre hacen todo lo posible para reunir estos recursos para que toda la comunidad los comparta y aprenda de.

Lea, comparta y aprenda, y se ayudará a guiarse a sí mismo y a su organización. ción al éxito!

—Gene Kim

junio 2019

Portland, Oregon

Colaboradores

Jason Cox

Director, Plataforma y

SRE, Disney,

@jasonacox

Christian Posta

CTO de campo, Solo.io,

@cristianoposta

Cornelia Davis

Vicepresidente de Tecnología,

Pivotal, @cdavisafc

Dominica DeGrandis

Directora Digital

Transformación, Tasktop,

@dominica

Jim Stoneham

SVP Crecimiento, Nueva Reliquia,

@jimstoneham

Thomas A. Limoncelli

Gerente SRE,

Desbordamiento de pila, Inc.,

@sithattom

Introducción

Randy es un ingeniero completo. Conoce el software de la empresa mejor que nadie. Tiene experiencia en todos los aspectos de los servicios de producción de la empresa: literalmente "desde chips hasta CSS". Escribe código de servidor y código de interfaz, toma decisiones sobre la interfaz de usuario, administra la base de datos, configura el sistema CI/CD y maneja las interrupciones. Por todo esto le pagan generosamente.

Hay dos cosas que Randy no sabe. Primero, está a punto de agotarse. Nadie puede mantener su nivel de estrés por mucho tiempo. Su salud está empezando a verse afectada y no puede entender por qué trabajar más horas no le permite adelantarse a todos los proyectos que deben completarse. Hace años que no tiene vacaciones.

La segunda cosa que Randy no sabe es que se está convirtiendo en un cuello de botella que está perjudicando a la empresa. Tiene tanta responsabilidad que casi todos los proyectos dependen de él; por lo tanto, casi todos los proyectos se estancan mientras lo esperan. La frustración de los equipos es palpable.

El manager de Randy lo sabe. La empresa intentó contratar a un ingeniero completo adicional durante dos años, pero fracasó. Los ingenieros completos son tan raros como los dientes de gallina. En los raros casos en los que existen, se trata de una situación poco saludable para ellos y para la empresa.

Encontrar a Randy fue un accidente afortunado. Ahora es un único punto de fracaso. La empresa está a un accidente de bicicleta o a un billete de lotería del desastre.

Entonces, ¿qué debe hacer una empresa?



En este artículo, analizaremos el mito del desarrollador full-stack y describiremos un paradigma más sostenible llamado equipo full-stack, también conocido como equipo equilibrado o equipo multifuncional.

Un equipo full-stack es un equipo equilibrado. Tiene mucha gente, cada una con un conjunto básico de habilidades y diferentes especialidades. En su conjunto, el

El conocimiento del **equipo cubre todo el stack**. Los miembros del equipo siempre buscan educar y empoderar a sus compañeros de trabajo para ayudarlos a mejorar sus habilidades. Esto crea una división del trabajo y un sistema saludable de especialización, y también fomenta el aprendizaje y el crecimiento de todo el equipo.

Primero, comprendamos qué es un desarrollador full-stack, así como la historia y mitología detrás del papel.

Parte 1: El origen del desarrollador Full-Stack

El auge del desarrollador full-stack

Entonces, ¿dónde se originó el término "desarrollador full-stack" o "ingeniero full-stack"? Las organizaciones recurren cada vez más a la tecnología como catalizador de nuevas oportunidades de negocio, así como como medio para lograr eficiencia y velocidad a través de la automatización. La necesidad de ir más rápido y seguir siendo competitivo está en su punto más alto. Las nuevas funciones, productos y servicios deben enviarse rápidamente, con mayor agilidad y mayor calidad que nunca. Con ese objetivo en mente, algunas organizaciones naturalmente buscan simplificar sus operaciones, evitar bucles transaccionales lentos y optimizar tecnologías tradicionales complejas y difíciles de operar. Están buscando formas de comprimir su ciclo de entrega, eliminar la latencia creada por equipos aislados y concentrar el conocimiento y la experiencia en grupos cada vez más pequeños para acelerar la acción.

Como resultado, las **empresas han comenzado a contratar desarrolladores e** ingenieros que poseen **un mayor conjunto de habilidades** que **abarcan múltiples disciplinas para reducir la coordinación** necesaria entre equipos especializados y, en última instancia, entregar productos más rápido. Si una organización pudiera contratar a una persona, como Randy en nuestro ejemplo anterior, que sea un experto en tecnologías de desarrollo front-end, middleware y bases de datos back-end, configuraciones de redes y almacenamiento, y preocupaciones de seguridad, ¿por qué querrían contratar a varias personas? ¿Algunos desarrolladores con habilidades limitadas en estas áreas?

Algunas **organizaciones creen** que **contratar desarrolladores o ingenieros completos** es una forma **de implementar los principios y prácticas de DevOps** en la organización. Si bien los objetivos de las prácticas de DevOps y los de los desarrolladores full-stack pueden ser muy similares, existen claras diferencias. **DevOps consiste en permitir que diferentes equipos se comuniquen y trabajen mejor juntos,** **no en eliminar la necesidad de comunicación** entre equipos **mediante la fusión de equipos.**

o hacer a una persona responsable de todo. Las prácticas de DevOps ayudan a las organizaciones a acelerar el desarrollo, la innovación y la entrega de productos y soluciones de alta calidad.

DevOps no es una estructura organizativa; más bien, define una forma de organizar equipos independientes (multifuncionales o “full-stack”), una cultura (humana y basada en resultados),

un conjunto de principios Lean (retroalimentación rápida, incluida la retroalimentación de la producción) y un conjunto de prácticas (entrega continua y altamente automatizada).¹ Parece que estos esfuerzos de DevOps para organizar equipos independientes multifuncionales han dado como resultado que algunas organizaciones piensen que las personas en esos equipos, similar a Randy en el ejemplo anterior, deben incorporar experiencia de manera singular en cada área como desarrollador full-stack.

Como veremos en este documento, el deseo de avanzar hacia desarrolladores full-stack tiene un efecto cíclico y perjudicial en los esfuerzos de reclutamiento de las organizaciones, así como en las capacidades de los desarrolladores para avanzar en sus carreras. Los desarrolladores full-stack son habitualmente anunciados como “unicornios” o “estrellas de rock” y reciben una remuneración acorde con esa ilusión. Esto puede hacer que los tecnólogos tengan expectativas poco realistas sobre sus propias habilidades y puede frenar el impulso de otros que no creen con confianza que cumplen con los requisitos de un desarrollador full-stack.

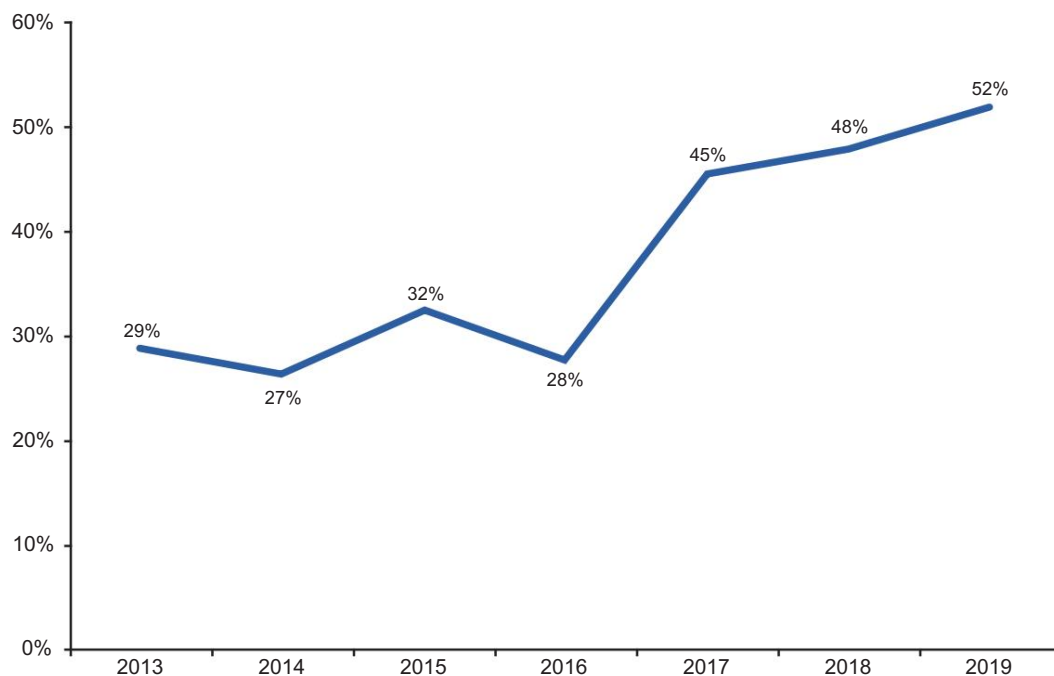


Figura 1: Desarrollador Full-Stack a lo largo del tiempo

Stack Overflow encuesta anualmente a los desarrolladores que utilizan su sitio y luego publican los resultados. Según datos de 2013 a 2019, hubo un aumento en el número de desarrolladores que se identificaron a sí mismos como “desarrolladores full-stack”.²

¿Qué es un desarrollador Full-Stack?

El **desarrollador full-stack**, también conocido como **ingeniero full-stack**, tiene una amplia gama de experiencia. Esas habilidades pueden diferir según el contexto, pero la intención es que el **ingeniero tenga experiencia laboral** en cada **capa necesaria para construir, enviar y ejecutar los productos en cuestión**. Como se ilustra en el ejemplo anterior con Randy, un **desarrollador full-stack** de una empresa puede ser **responsable de cada capa** de la **tecnología que impulsa el negocio**, producto o servicio. El desarrollador comprendería no solo la experiencia del usuario final, sino también los servicios API que respaldan el producto, el proceso de desarrollo, la ingeniería de calidad, la arquitectura de seguridad y la infraestructura necesaria para construir y ejecutar la aplicación. Como se menciona en la historia de Randy, el ingeniero de pila completa escribe el código, lo ejecuta, administra los datos y es el líder designado para manejar las interrupciones.

Muchas **organizaciones están contratando desarrolladores full-stack**. A continuación se muestra una plantilla de descripción de trabajo de SkillGigs.com que muestra el conjunto típico de habilidades y responsabilidades que se esperan de los desarrolladores full-stack:3

- **Diseño de la arquitectura general de la aplicación web.**
- Mantener **la calidad y garantizar la capacidad de respuesta de las aplicaciones.**
- Colaborar **con el resto del equipo de ingeniería para diseñar y lanzar nuevos** **características.**
- Mantener **la integridad y organización del código.**
- Experiencia **trabajando con diseñadores gráficos** y convirtiendo **diseños a visuales.** **elementos.**
- Comprender **e implementar la seguridad y protección de datos.**
- Tener mucha experiencia con lenguajes de **programación back-end** (por ejemplo, PHP, Python, Ruby, Java, .NET, JavaScript).
- **Experiencia competente en el uso de frameworks front-end** (p. ej., **AngularJS**, KnockoutJS, BackboneJS, **ReactJS**, DurandalJS).
- **Experiencia en desarrollo tanto para dispositivos móviles como de escritorio.**
- Comprensión **de los lenguajes del lado del servidor** (por ejemplo, Jade, EJS, Jinja).
- **Experiencia con API de mensajes en la nube y uso de notificaciones push.**
- Conocimiento de herramientas de control de versiones de código (por ejemplo, Git, Mercurial o SVN).

- Asegúrese de mencionar cualquier marco, biblioteca u otra tecnología adicional relevante para su proyecto/empresa.

Aquí hay otra plantilla para un puesto de desarrollador full-stack de BetterTeam.com:4

- Responsabilidades:

- 0 Desarrollar la arquitectura del sitio web front-end.
- 0 Diseñar las interacciones de los usuarios en las páginas web.
- 0 Desarrollar aplicaciones web back-end.
- 0 Crear servidores y bases de datos para funcionalidad.
- 0 Garantizar la optimización multiplataforma para teléfonos móviles.
- 0 Garantizar la capacidad de respuesta de las aplicaciones.
- 0 Trabajar junto con diseñadores gráficos para funciones de diseño web.
- 0 Ver un proyecto desde su concepción hasta el producto terminado.
- 0 Diseñar y desarrollar API.
- 0 Satisfacer las necesidades técnicas y de consumo.
- 0 Manténgase al tanto de las novedades en aplicaciones y programación web idiomas. •

Requisitos: 0

Licenciatura en Informática.

- 0 Fuertes habilidades organizativas y de gestión de proyectos.
- 0 Competencia con lenguajes front-end fundamentales como HTML, CSS, y JavaScript.
- 0 Familiaridad con marcos de JavaScript como Angular JS, React y Ámbar.
- 0 Competencia con lenguajes del lado del servidor como Python, Ruby, Java, PHP, y .Net.
- 0 Familiaridad con la tecnología de bases de datos como MySQL, Oracle y MongoDB.
- 0 Excelentes habilidades de comunicación verbal.
- 0 Buenas habilidades para resolver problemas.
- 0 Atención al detalle.

¿Qué es la pila completa?

Un **desarrollador full-stack** es aquel que es un **experto en cada parte de la tecnología** involucrada en la implementación y **las operaciones de algún software**. Como puede ver **en las descripciones de trabajo** de la **plantilla anterior**, esto a menudo **significa código** que forma **tanto la interfaz de usuario** (a menudo denominada interfaz), **un conjunto de servicios**

o API (a menudo denominadas back-end), sistemas de **almacenamiento de datos** (como **bases de datos y/** o dispositivos de almacenamiento de nivel inferior) y **herramientas de integración** (como buses de mensajes).

Con el uso cada vez mayor de arquitecturas de microservicios, el desarrollador de pila completa también debe estar bien versado en una gran variedad de herramientas que respaldan su uso, como disyuntores, servicios de configuración y descubrimiento de servicios y rastreo distribuido. Por supuesto, también necesitarán **comprender una gran cantidad de herramientas** para **respaldar los procesos de desarrollo de software**, como **sistemas de control de código fuente** y **sistemas CI/CD**, y **ser expertos en las prácticas involucradas** en la **creación, lanzamiento y ejecución de aplicaciones en producción**.

Pero incluso estas plantillas de descripción de puestos hacen varias referencias a cosas que se pueden agregar o eliminar para que la descripción sea relevante para su proyecto o empresa.

Las habilidades que necesita el desarrollador full-stack dependen en gran medida de la tecnología que requiere el producto, así como de los procesos que se siguen para llevar el producto de la idea a la producción. ¿Qué quieren decir los tecnólogos y las empresas cuando dicen “full stack”?

La **Figura 2** sugiere una definición de “pila completa” e ilustra que **la experiencia necesaria para un puesto de desarrollador de pila completa abarca un amplio espectro**. En el centro está la pila arquitectónica, comenzando con la **infraestructura en la base y agregando middleware y numerosas capas del software en sí**. A la **izquierda de esta pila hay otra pila de herramientas y prácticas en las que el desarrollador de pila completa podría necesitar tener experiencia para implementar el ciclo de vida de entrega de software (SDLC)**. Y a la **derecha están las prácticas operativas necesarias para mantener el software funcionando bien en producción**.

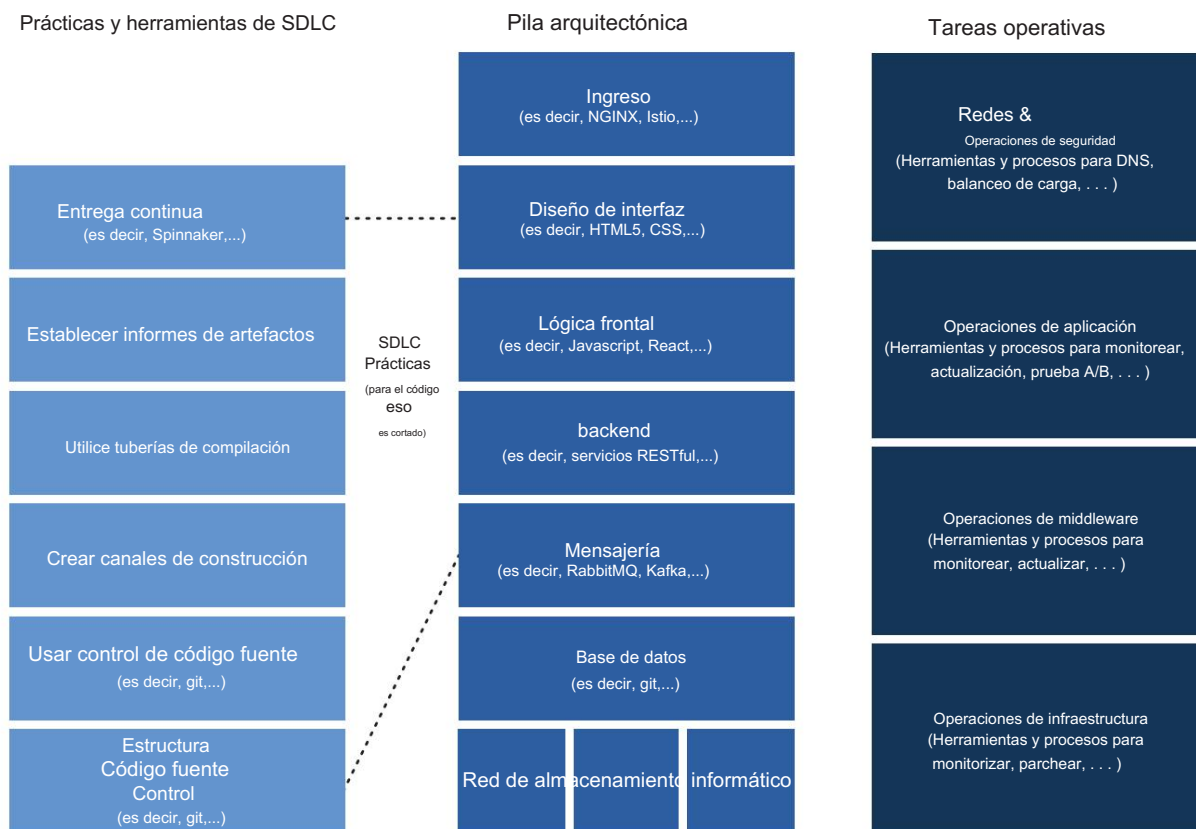


Figura 2: Ejemplo de ingeniero de pila completa que desarrolla nuevo software

¿Se espera realmente que el desarrollador full-stack haga todas estas cosas? Exploraremos esta cuestión más a fondo en secciones posteriores de este artículo; sin embargo, por el momento queremos volver a esta pregunta: ¿Qué es “la pila completa”? Dependiendo del entorno en el que se desarrolle y ejecute el software, algunos de los elementos anteriores pueden cambiar.

Por ejemplo, con la **creciente popularidad de la nube** y las plataformas en la nube, como **AWS, GCP y Azure**, algunos de los **requisitos operativos** pueden ser manejados por la plataforma, **eliminando o reduciendo** el **nivel de experiencia** requerido por el desarrollador de pila completa. Por ejemplo, si se utilizara **Amazon RDS** (servicio de **base de datos relacional**) para la **base de datos**, el desarrollador de pila completa **no sería responsable de actualizar sus instancias de MySQL**. Pero aun así, es probable que el desarrollador **full-stack** aún **necesite cierto nivel de experiencia en bases de datos**.

Por el lado del **SDLC**, no es inusual que las **organizaciones tengan un grupo que proporcione** herramientas **de CI y CD** como servicio a sus desarrolladores. En este caso, el desarrollador full-stack

no sería responsable de tomar decisiones sobre algunas de las prácticas específicas de **CI y CD**. Sin embargo, seguirían siendo responsables de seguir las **prácticas establecidas** y aún **necesitarían tener cierto nivel de experiencia en el uso de esas herramientas**.

La Figura 3 muestra una pila alternativa que refleja este tipo de escenarios.

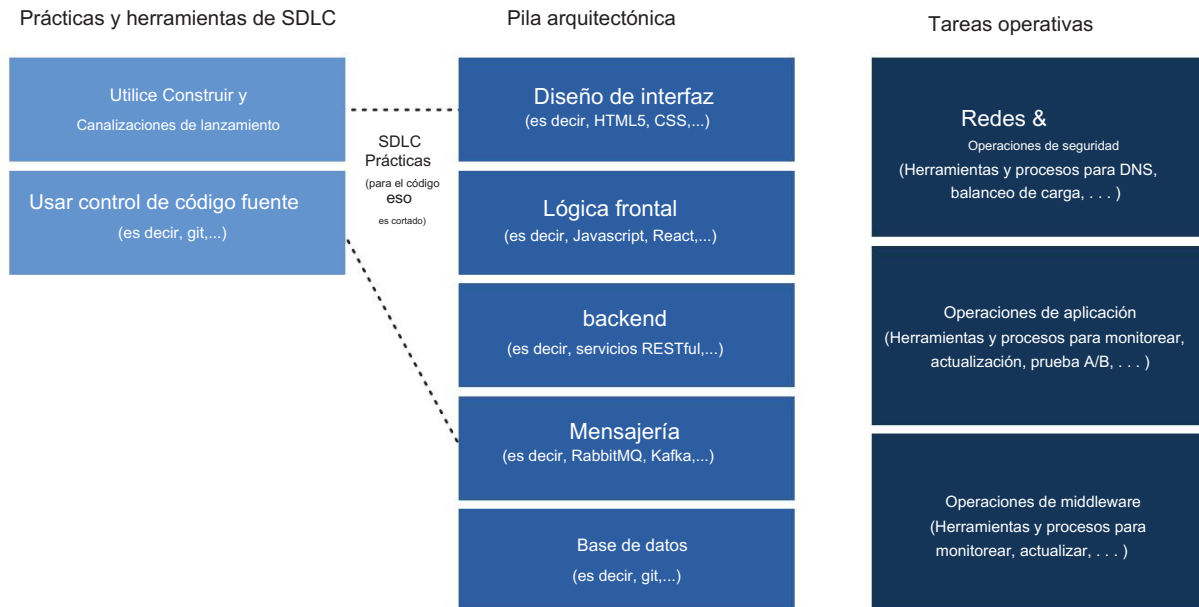


Figura 3: Ejemplo de pila reducida de desarrollo de nuevo software donde hay una plataforma

El equipo proporciona herramientas de CI y CD como servicio

El sombreado más claro indica una menor experiencia necesaria.

Claramente, diferentes **definiciones de "la pila"** darán lugar a diferentes requisitos para el desarrollador de pila completa. Además, estos ejemplos dan una idea del meollo de la cuestión: ¿**quién hace qué? Hay muchas cosas que intervienen en el desarrollo y ejecución de software en producción**. La forma en que se distribuyen esas responsabilidades es el centro de nuestra discusión.

Una de las **principales palancas que tenemos para formar prácticas eficientes de DevOps es cómo definimos la pila**.

Parte 2: Los problemas con el desarrollador Full-Stack

El mito del desarrollador full-stack

Si bien el desarrollador full-stack va en aumento, ¿es siquiera realista encontrar ingenieros de este tipo?

Dediquemos un minuto a analizar lo que significa ser un desarrollador o ingeniero "full-stack".

Como se sugirió anteriormente, un paso importante es definir su "pila completa". ¿Estás hablando de una **gama que va desde chips hasta CSS?*** ¿O algo más pequeño? Algunos equipos incluyen la red en su conjunto completo, mientras que otros equipos aprovechan los proveedores de plataformas y, por lo tanto, no requieren esa experiencia.

La plantilla de descripción del trabajo que presentamos anteriormente define implícitamente la pila a través de la lista de tecnologías que el desarrollador debe dominar con la expectativa (y sugerencia) de que será necesario que haya extensiones para las plataformas, marcos y requisitos únicos específicos de la empresa. . **Reclutar para estos roles puede ser extremadamente difícil.** **Encontrar grandes talentos** ya es una propuesta difícil. **Al agregar los requisitos completos**, el grupo de candidatos se reduce rápidamente. Estamos observando esto en la industria. Como ejemplo, a uno de los autores de este artículo se le asignaron puestos completos durante dos años para su equipo de crecimiento y nunca ocupó ninguno de esos puestos con una persona con todas las habilidades. Eventualmente contrataron personas que, con el tiempo, mejoraron sus habilidades para poder trabajar en todos los niveles, pero el criterio completo era problemático.

ático cuando se utiliza para seleccionar candidatos.

Como comentamos anteriormente, **la definición de "pila" puede variar mucho**. En consecuencia, la definición **del rol de desarrollador full-stack varía mucho**, tanto según quienes anuncian el empleo como según quienes buscan empleo. Esto ha dado lugar a hilos confusos en las bolsas de trabajo con empresas y candidatos que intentan lidiar con percepciones desconectadas. Además, se están **formando campamentos de entrenamiento para ayudar a las personas a obtener credenciales como desarrolladores completos**, pero no está claro si estos campamentos de entrenamiento están ayudando a cerrar la brecha.

Como anécdota, **parece haber un problema de contratación impulsado por la escasez de talento completo**. Sin embargo, ¿**podemos encontrar datos que respalden esa afirmación?** Para el análisis, vamos

* "Chips to CSS" se refiere a las capas de tecnología. En la capa más baja se encuentran los propios chips de computadora. Además de eso, está el sistema operativo, los marcos de software, la aplicación en sí y el código CSS (parte de HTML) que define la apariencia de la aplicación.

utilice la descripción del trabajo de SkillGigs.com de la sección anterior y compárela con un conjunto de datos que represente el mercado laboral. Utilizaremos la Encuesta para desarrolladores Stack Overflow de 2018,5 que es una sólida encuesta global de desarrolladores de software, para ver cómo la descripción del desarrollador full-stack coincide con la realidad del talento en el mundo real.

En la encuesta, aproximadamente el 48% de los encuestados se identificaron como desarrolladores full-stack. Con base en los datos de la encuesta, deberíamos poder reducir la población (N) a aquellos que realmente califican para la descripción de trabajo básica de desarrollador full-stack discutida anteriormente. La Tabla 1 ilustra los pasos.

	Notas	Desarrolladores
Financiación de hitos predefinidos en el alcance del proyecto. Nuevo presupuesto discrecional significa la creación de un nuevo proyecto.		N = 98.850 desarrolladores
Filtrar por ubicación: Estados Unidos.	Una ubicación más estrecha, como ciudad o estado, no estaba disponible.	norte = 20.300
Filtre los candidatos que se describen a sí mismos como ingenieros completos.		norte = 10.400
Diseño de la arquitectura general de la aplicación web.	La "arquitectura" no se mide directamente en la encuesta, por lo que la omitiremos por ahora.	norte = 10.400
Mantener la calidad y garantizar la capacidad de respuesta de las aplicaciones.	Nuevamente, esto no se mide directamente, por lo que omitir ahora.	norte = 10.400
Colabora con el resto del equipo de ingeniería para diseñar y lanzar nuevas funciones.	Esto es algo que tendrás que buscar manualmente en el currículum de un candidato. Saltando por ahora.	norte = 10.400

Mantener la integridad y organización del código.	“Usar control de versiones sistema regularmente” se utilizó como nuestro proxy para esto. (Usando cosas como Git y Subversion). No es perfecto, pero es un lugar para comenzar.	norte = 8.432
Experiencia trabajando con diseñadores gráficos y convirtiendo diseños en elementos visuales.	No hay disponible datos sobre el trabajo con diseñadores.	norte = 8.432
Comprensión e implementación de la seguridad y protección de datos.	No hay datos disponibles sobre seguridad y protección de datos.	norte = 8.432
Alta experiencia con lenguajes de programación back-end (por ejemplo, PHP, Python, Ruby, Java, .NET, JavaScript, etc.).	Observamos que un lenguaje como Go podría (y algunos argumentarían que debería) incluirse, pero un reclutador que se ciñe estrictamente a los idiomas enumerados aquí no lo incluiría. Es decir, aunque exista un “etc.” al final de esa lista, un reclutador no técnico no sabría incluir Go. También filtramos por cualquier exposición a los idiomas frente a “alta experiencia”.	norte = 44

Tabla 1: Ofertas de empleo completas frente a ingenieros disponibles

Esta tabla muestra la población que participa en la encuesta filtrada por un ejemplo de requisitos completos de publicación de empleo.

La columna de la izquierda muestra el requisito y la columna de la derecha muestra el número restante de la población.

A partir de 98.850 participantes, el grupo de candidatos que cumplen con el requisito completo se reduce a 44.

Detengámonos aquí. **Todavía hay cinco calificaciones más enumeradas** en la descripción del trabajo, incluidos marcos, desarrollo en dispositivos móviles y de escritorio, idiomas adicionales y API en la nube, y nos saltamos cinco de los requisitos anteriores porque no teníamos datos. Solo filtramos según cuatro de los catorce requisitos enumerados (aunque agregamos un requisito geográfico).

Cuarenta y cuatro. Todas las empresas de América del Norte compiten por cuarenta y cuatro ingenieros completos representados en la Encuesta de desarrolladores de Stack Overflow de 2018. Esto es sólo el **0,04% de la población encuestada** que coincide con el perfil deseado para un desarrollador full-stack. ¡Imagínese reclutar con este perfil y las compensaciones necesarias para cubrir los puestos! Esto nos ayuda a comprender por qué las organizaciones que contratan desarrolladores full-stack tienen tantas dificultades para desempeñar esos roles y retener ese talento. El desafío se vuelve aún más difícil si su objetivo es formar un equipo con diversidad de género, ya que solo el 7,1% de los encuestados se identifican como no hombres.

Las organizaciones que intentan **mejorar su diversidad e inclusión** deben tomar nota: la **publicidad de un desarrollador full-stack puede disuadir a las mujeres de postularse**. Los estudios muestran que los hombres solicitan un empleo si sienten que cumplen con el 60% de la descripción del puesto.⁶ Las mujeres, por otro lado, no solicitarán un empleo a menos que sientan que cumplen con todos los requisitos laborales. Dado que la típica descripción del puesto de desarrollador full-stack representa el pináculo de las habilidades de ingeniería, es muy probable que las organizaciones, sin saberlo, estén disuadiendo a las mujeres de postularse. Si tiene dificultades para atraer mujeres para postularse para su organización de ingeniería, es posible que esté utilizando términos o descripciones de trabajo que impliquen niveles de habilidad exagerados. También es posible que desees eliminar términos exagerados como **"full stack", "rock star" o "ninja"**.

Por qué el **ingeniero Full Stack es problemático**

Anteriormente, definimos el rol del desarrollador full-stack y analizamos lo difícil que es contratar para este rol. En esta sección, veremos el rol en sí y preguntaremos

si las personas **que desempeñan estos roles se encuentran en una posición sostenible o en riesgo de agotamiento**. No permitimos que nuestros servidores alcancen el 100% de utilización de su capacidad. ¿Por qué se lo permitimos a nuestra gente?

¿Es posible que seas bueno en todo? En la era digital, la gran cantidad de herramientas, marcos, lenguajes de programación y métodos/modelos abruma el cerebro. Las **tecnologías van y vienen, y aprender una nueva tecnología a menudo implica intensos**

esfuerzo. Aprender un nuevo entorno y conocimientos de dominio lleva tiempo. Cuando se pregunta a los ingenieros cuánto tiempo le toma a un nuevo empleado ponerse al día, una respuesta común es de seis a doce meses.

Se necesitan años de arduo trabajo para convertirse en un maestro en cualquier cosa, y las habilidades de DevOps no son diferentes. Kubernetes, por ejemplo, es famoso por su complejidad y la relativa facilidad para que los desarrolladores junior cometan errores de manera impredecible. Si está bebiendo de la manguera contra incendios mientras aprende nuevos conocimientos de dominio, es posible que también esté inmerso en el aprendizaje de un nuevo lenguaje como Clojure, utilizando nuevas API en la nube y aplicando Big-O notación hasta nuevos algoritmos y dominio del aprendizaje automático y las funciones sin servidor. Y eso apenas toca la superficie de una tecnología que continúa acelerándose y expandiéndose. ¿Cómo piensa mantenerse al día con la creciente pila? Dominar una habilidad de forma predeterminada significa que ignorarás intencionalmente otras habilidades para concentrarte en la que deseas dominar. El cerebro sólo puede contener una cantidad determinada. Nuestra memoria de trabajo humana es vulnerable a la sobrecarga, lo que ocurre a medida que estudiamos temas cada vez más complejos y realizamos tareas cada vez más complejas.

La sobrecarga cognitiva puede resultar en menor calidad, actos heroicos, agotamiento y problemas de salud.

Una de las principales quejas de los ingenieros son las constantes interrupciones que impiden que las personas completen el trabajo durante el día. Si no puede realizar su trabajo más importante durante el horario comercial, ¿cuándo termina su trabajo? La presión de trabajar durante las primeras horas de la noche además del trabajo diurno habitual es fuerte. Trabajar muchas horas puede conllevar derechos de fanfarronería de moda, lo que implica fuerza y poder, pero como dice la escritora de salud de The Wall Street Journal, Melinda Beck, "las personas genuinas que duermen poco sólo necesitan cuatro horas de sueño por noche, pero comprenden sólo del 1% al 3%. % de la población total". 7 Entonces, para el 97% al 99% de nosotros que no dormimos poco, trabajar hasta altas horas de la madrugada conlleva falta de sueño y errores; ambos contribuyen al agotamiento.

El agotamiento es más que sentirse triste. Es un estado crónico de desincronización en el trabajo, y puede ser un problema importante en su vida, incluidos los siguientes impactos:

- Pérdida de energía: el individuo agotado se siente abrumado, estresado y exhausto.

• **Pérdida de entusiasmo**: la pasión por el trabajo se ha desvanecido; el trabajo "te molesta"; se siente como una carga o una tarea. • **Pérdida de confianza**: sin energía y participación activa en el trabajo, es difícil mantenerse motivado.

La profesora de Psicología, Dra. Christina Maslach, explica que el agotamiento tiene muchas consecuencias para las personas, incluidas enfermedades físicas, sentimientos de desesperanza, irritabilidad, impaciencia y malas relaciones.⁸ En casos graves, el agotamiento puede afectar negativamente el funcionamiento ejecutivo, la atención y la memoria. . El Inventario de Burnout de Maslach (MBI) se considera el "estándar de oro" para medir el burnout. Un estudio interno de cuatrocientos ingenieros de Infosec compartido de forma anónima con nosotros indicó que el 47% tiene un riesgo significativo de agotamiento principalmente debido a altos niveles de cinismo, una dimensión clave del estudio porque captura la dificultad de tratar con otras personas y actividades en el mundo laboral.

El riesgo de agotamiento por sobrecarga de información es real y conduce a graves problemas de salud:

Demasiada información pone en peligro la salud de nuestro cerebro, lo que resulta en una sobrecarga de información. Con el tiempo, la exposición a múltiples fuentes de datos conduce a una sobreestimulación del cerebro. Las neuronas se sobrecargan con datos, números, plazos, objetivos que cumplir y proyectos que completar, y toda esta información innecesaria puede, en última instancia, destruirlas. En consecuencia, un cerebro estresado y sobrecargado tiene un alto riesgo de sufrir demencia y otros trastornos neurodegenerativos (enfermedades de Parkinson y Alzheimer).⁹

Esperar que las personas, incluyéndonos a nosotros mismos, sean hábiles en todo el proceso se suma a la carga cognitiva continua que se debe llevar a cabo para realizar el trabajo. Esto puede provocar agotamiento e incluso, en algunos casos, problemas de salud más graves. Pero necesitamos ser Capaz de respaldar todo el conjunto para ayudar a que nuestras empresas sigan siendo relevantes y sigan avanzando. ¿Qué podemos hacer? En la siguiente sección, veremos algunas opciones para ayudar a respaldar todo el conjunto y, al mismo tiempo, brindar un entorno de trabajo saludable y humano para nuestros equipos y para nosotros mismos.

Parte 3: ¿Cuál es el verdadero camino a seguir?

Como se mencionó anteriormente, el desarrollador o ingeniero full-stack es extremadamente difícil de encontrar, desarrollar y mantener. En los casos en que los individuos son capaces de cumplir estos roles, corren el riesgo de sufrir una sobrecarga cognitiva y los riesgos para la salud relacionados.

Además, la aceleración y evolución de la tecnología significa que el dominio es un objetivo móvil que exige una inversión continua en aprendizaje en múltiples dominios. Sin embargo, nuestras empresas necesitan una funcionalidad completa para seguir siendo relevantes, eficientes y ágiles en el mercado.

¿Cómo satisfacemos esa demanda sin sobrecargar a las personas?

El equipo completo

Un equipo completo tiene las habilidades combinadas de sus miembros para diseñar, construir, implementar y operar software de manera efectiva durante todos los ciclos de desarrollo de sus entregables.

Pasar a equipos full-stack nos ayuda a ofrecer la ventaja full-stack a nuestras organizaciones sin los desafíos de reclutar, desarrollar y mantener desarrolladores o ingenieros full-stack. A continuación se muestran algunos ejemplos de equipos full-stack:

- **Ingeniero rotacional:** este equipo ofrece un panel interno de salud del cliente que utilizan todos los equipos de atención al cliente. Tiene dos ingenieros de back-end, un ingeniero de front-end, un científico de datos, un analista de datos, un diseñador y un gerente de producto. Cada miembro del equipo es esencial, pero se recomienda encarecidamente la rotación entre roles y la enseñanza dentro del equipo. Cada rol de especialista también está activo en comunidades de práctica entre equipos para compartir y desarrollar una experiencia más profunda.
- **Ejemplo de matriz:** se puede formar un equipo completo incorporando expertos funcionales en los equipos de productos. Los miembros del equipo tienen dos relaciones jerárquicas, la principal es el líder del equipo de producto y la secundaria es el gerente del área funcional encargado de garantizar las habilidades funcionales y el desarrollo profesional. Estos equipos integrados son responsables de aprender del resto del equipo, asumiendo la responsabilidad de la entrega general y la confiabilidad operativa del producto, pero también de garantizar que el resto de su equipo también comprenda su área funcional.

A continuación se presentan algunas prácticas recomendadas para ayudar a crear equipos completos:

- **Miembros del equipo en forma de T:** todos son expertos en una o algunas habilidades (la barra vertical de la T), pero hay una comprensión básica y empatía por muchas o todas las demás habilidades necesarias para construir y ejecutar el producto (la barra horizontal).

Considere rotar a los miembros del equipo a través de diferentes especializaciones para ayudar a crear más miembros del equipo en forma de T. Rote/reorganice a los miembros del equipo y/o enseñe a otros miembros del equipo para garantizar que los expertos puedan tomarse unas vacaciones y reducir el riesgo en caso de que alguien abandone la empresa. • Todos enseñan: al enseñar, usted mismo comprende mejor el tema y al mismo tiempo genera empatía por los demás. Un vicepresidente de ingeniería con el que hablamos dijo que la medida de un ingeniero senior es cuánto enseña, no cuánto codifica.

- **Mentalidad de crecimiento:** contrate personas con mentalidad de crecimiento mientras integra el personal de sus equipos, de modo que esos miembros estén abiertos al crecimiento. Está bien tener curiosidad y asumir roles adicionales para avanzar mediante la rotación.¹⁰
- **Comunidades de práctica:** cree comunidades de prácticas para que los miembros de su equipo. Los miembros pueden desarrollar habilidades más profundas y aprender las mejores prácticas de otros.
- **Sin nuevos silos:** no cree nuevos silos funcionales necesarios para enviar valor. La intención es crear equipos de especialistas de alto rendimiento con altos grados de comunicación, empatía e intercambio de conocimientos. La creación de silos puede inhibir gravemente la velocidad, la comunicación, la empatía y el intercambio de conocimientos.

Partición “La Pila”

En una sección anterior, sugerimos que uno de los desafíos con el término desarrollador full-stack era que no estaba claro qué constituía el full stack y, por lo tanto, era difícil incluso saber el conjunto preciso de habilidades que se necesitaban del full stack.

-Desarrollador de pila. Por supuesto, mientras tanto, primero hemos desacreditado el mito del desarrollador full-stack y hemos sugerido una solución: el equipo full-stack (FST). Pero incluso con el equipo completo, la pregunta sigue siendo: ¿De qué es exactamente responsable el equipo colectivo? ¿Es, por ejemplo, todo lo que se muestra en la Figura 2, todo lo que se muestra en la Figura 3 o algo completamente distinto?

Lo que presentamos aquí es la idea de que mediante una cuidadosa partición del conjunto de todo lo necesario para desarrollar y ejecutar alguna oferta digital, se puede optimizar el desarrollo general y las operaciones de la misma. Y ya vemos amplia evidencia de que esto funciona en la última década en TI, en particular con la asombrosa adopción de servicios ofrecidos por empresas como Amazon Web Services (AWS), Google Cloud Platform (GCP), Micro-soft Azure y otros. Las figuras 4a, 4b y 4c ilustran la evolución de la plataforma en la nube observada en las últimas décadas. Cuando se iba a poner en línea una nueva aplicación, la organización de desarrolladores era responsable de todo, desde el montaje y apilamiento del hardware hasta el código de la aplicación (Figura 4a). Luego, con la llegada de servicios como AWS, GCP y Azure, la organización podría obtener computación, almacenamiento y red virtualizados de uno de esos proveedores de infraestructura como servicio (IaaS) (Figura 4b). Gracias a esto, la “pila completa” se redujo efectivamente, de modo que el equipo que puso la aplicación en línea era responsable y necesitaba experiencia en mucha menos tecnología.

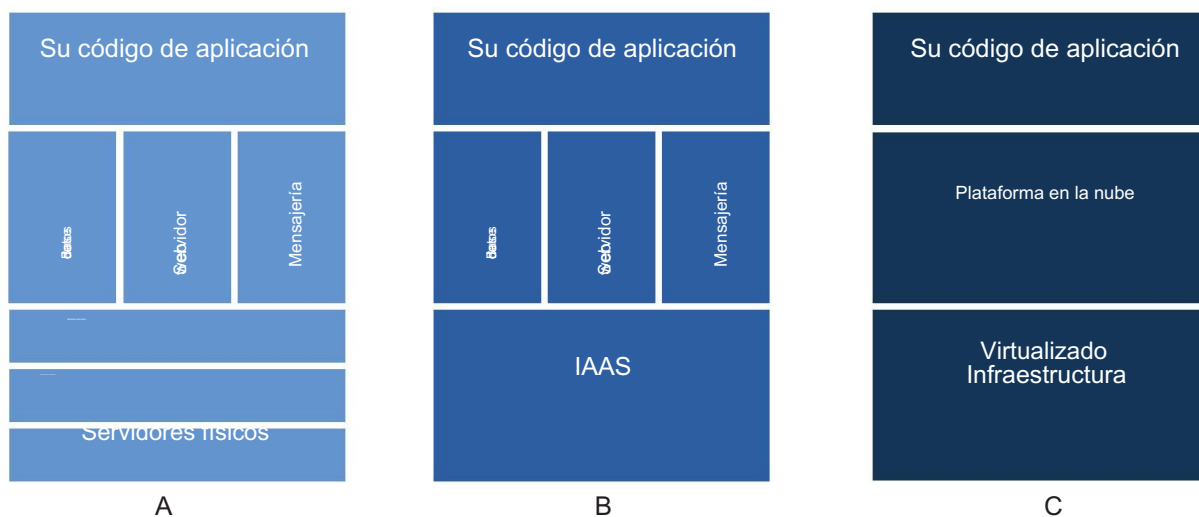


Figura 4: La evolución de la plataforma en la nube

En los últimos años, han surgido abstracciones de nivel aún más alto, llamadas “plataforma en la nube”. La plataforma en la nube puede adoptar diferentes formas, pero a menudo incluye muchos middleware y otros servicios, como programación y tiempos de ejecución de cargas de trabajo, bases de datos, servicios de mensajería, enrutamiento y más. Estas capacidades se ofrecen, generalmente como un servicio, en abstracciones de nivel más alto que IaaS, lo que efectivamente reduce aún más el conjunto de cosas en las que un equipo full-stack debe tener experiencia (Figura 4c). Algunas de las formas que brindan estos servicios de plataforma en la nube son:

- **Servicios de contenedor:** permite al equipo completo proporcionar simplemente imágenes de contenedor, como archivos Docker en los que han encapsulado la implementación de su aplicación, así como las dependencias del tiempo de ejecución, y la plataforma toma se encarga de aprovisionar todas las necesidades de soporte informático, almacenamiento y redes. Ejemplos de este tipo de plataforma basados en Kubernetes son Google Kubernetes Engine (GKE), Amazon Elastic Container Service for Kubernetes (EKS), Azure Kubernetes Service (AKS) y Pivotal Container Service (PKS).
- **Servicios de aplicaciones:** permite que el equipo completo proporcione simplemente la implementación de la aplicación mientras la plataforma proporciona el tiempo de ejecución para la aplicaciones, servicios de enrutamiento y más. Ejemplos de estas plataformas incluyen Cloud Foundry, Heroku, Google App Engine, AWS Elastic Beanstalk y Azure App Service.
- **Servicios de funciones:** permite al equipo completo proporcionar simplemente implementaciones de funciones de bajo nivel y una especificación de cómo los eventos unen esas funciones para definir el comportamiento de la aplicación. Ejemplos de estas plataformas incluyen AWS Lambda, Google Cloud Functions y Azure Functions.

Estas diferentes abstracciones definen efectivamente el conjunto de habilidades de las que es responsable el equipo completo y, como sugiere la Figura 5, pueden aumentar o disminuir el tamaño de ese conjunto. En resumen, la selección cuidadosa de la plataforma utilizada por el equipo full-stack para desarrollar y llevar sus aplicaciones al consumidor juega un papel central en lo que el equipo full-stack es responsable y en lo que debe poseer habilidades.



Figura 5: Ejemplo de cómo la abstracción de infraestructura reduce el Complejidad del Stack que debe poseer el equipo

Por supuesto, existe un equipo que es responsable de proporcionar la plataforma al equipo completo. Proporcionan esta plataforma como un producto, son ellos mismos responsables del desarrollo y las operaciones de este producto de software y, como equipo, tienen un conjunto específico de habilidades que se les exige; A este equipo lo llamamos equipo de plataforma y al equipo que consume la plataforma equipo de aplicación.

Considerando esto, ofrecemos una observación más interesante: lo que hemos descrito aquí es una transferencia de responsabilidad para ciertas partes de la pila a diferentes equipos. Esta transferencia de responsabilidad es efectiva solo cuando existe un protocolo bien establecido para que el equipo de la aplicación utilice los servicios ofrecidos en la plataforma por el equipo de la plataforma. Advertiríamos que no se recomiendan otros intentos de transferir la responsabilidad, como permitir a los desarrolladores escribir código y al mismo tiempo hacer que el control de calidad (QA) sea responsable de su calidad. Uno de los elementos más importantes del equipo full-stack es que todos los miembros están incentivados para lograr el mismo resultado: brindar valor al consumidor a través de su software.

Epílogo

Eran las cinco en punto. Jennifer y su equipo se marchaban por el día. Jennifer hizo una pausa para reflexionar sobre la buena semana que había sido. Se lanzó una nueva característica importante y todos estaban de humor para celebrar.

Qué diferente, pensó Jennifer, que hace un año.

Hace un año, la empresa perdió a un desarrollador full-stack llamado Randy. Randy se había agotado, se volvió tóxico y terminó dejando la empresa. Fue difícil reemplazarlo, pero al darse cuenta de que es casi imposible encontrar desarrolladores full-stack, Jennifer tomó un camino diferente. No quería recrear la situación que tuvieron con Randy, donde el éxito o el fracaso de cada lanzamiento dependía en gran medida de él.

En cambio, decidió formar un equipo completo. Su estrategia de contratación cambió para centrarse en contratar personas inteligentes a las que les gustara compartir y aprender. Lenta y metódicamente construyeron un equipo de ingenieros en forma de T; cada uno era un experto superficial en todo y un experto profundo en una o más áreas de enfoque. Cada miembro asumió la responsabilidad de educar a otros miembros del equipo y ayudarlos a "subir de nivel".

El resultado es un equipo multifuncional que siempre comparte y aprende. La gente hace "simulacros de incendio" y "días de juego" para ganar experiencia y entrenarse. Trabajan

colaborativamente. Se agolpan en los problemas. Ninguna persona es un cuello de botella. Los proyectos se realizan más rápido, a tiempo y las partes interesadas quedan impresionadas con los resultados.

Si bien los resultados comerciales son importantes, Jennifer está muy contenta con el equilibrio entre el trabajo y la vida personal que disfruta. Trabaja no más de cuarenta horas a la semana, espera trabajar todos los días y se toma vacaciones sin sentirse culpable. Le encanta trabajar aquí y no se imagina irse.

Lo más importante es que su equipo se ha convertido en un modelo. Los ingenieros de otros equipos quieren unirse a su equipo; Las áreas de productos quieren trabajar con su equipo. Otros equipos buscan consejo en el equipo de Jennifer. El éxito genera imitación y el equipo de Jennifer fomenta esto organizando charlas quincenales a la hora del almuerzo.

Mientras cerraba la puerta de su oficina, pensó: ¿Por qué trabajaríamos de otra manera?

¿Cómo consigo individuos en forma de T?

Un individuo en forma de T es un ingeniero que tiene al menos un conocimiento básico de todas las áreas funcionales (desarrollo, sistemas, calidad, producto, sistemas, seguridad, etc.) y también tiene una profunda experiencia en una de esas áreas, que pueden contribuir a la capacidad completa del equipo general.

La forma en que esto se desarrolla puede adoptar muchas formas.

En todos los casos, una "mentalidad de crecimiento" versus una "mentalidad fija" es esencial para construir la barra horizontal multifuncional. Las organizaciones pueden ayudar a desarrollar estas áreas fomentando el aprendizaje continuo, permitir que las personas se muevan entre equipos y roles y promover un entorno abierto y colaborativo. La creación de comunidades de tecnología o gremios de expertos puede ser un diseño organizacional que ayude a fomentar este crecimiento.¹¹

Carol Dweck escribe:

Cuando empresas enteras adoptan una mentalidad de crecimiento, sus empleados afirman sentirse mucho más empoderados y comprometidos; también reciben un apoyo organizacional mucho mayor para la colaboración y la innovación.

Por el contrario, las personas que trabajan principalmente en empresas con mentalidad fija informan más de una sola cosa: trampas y engaños entre los empleados, presumiblemente para obtener una ventaja en la carrera de talento.¹²

Estoy en un rol de "ingeniero completo". ¿Qué debo hacer?

No hay nada de malo en tener experiencia en múltiples áreas. Sin embargo, existen peligros a los que hay que prestar atención, como hemos identificado anteriormente:

- Eres un SPOF (punto único de falla): en algunos casos, el full-stack ingeniero es el único ingeniero que respalda la plataforma. Esto pone al equipo y a la organización en riesgo de tener un único punto de falla. Más importante aún, afecta drásticamente su capacidad para disfrutar del tiempo libre o del tiempo para invertir en usted mismo. A nivel profesional, esto a menudo resulta en un callejón sin salida: usted es demasiado crítico en su función para ascender.

- Tienes una mentalidad de héroe: sé estratégico a la hora de posicionarte. Es tentador querer ser un héroe, pero como se mencionó anteriormente, eso tiene consecuencias humanas para la salud, el equilibrio de la vida y la longevidad. • La calidad de su trabajo disminuye: Vemos una mejora continua en todas las áreas funcionales. La capacidad de seguir siendo relevante al mismo tiempo a un nivel de experiencia en todas estas áreas puede ser onerosa. En cambio, centrarse en un área para aprender, perfeccionar y entregar puede agregar un mayor grado de calidad al equipo en general. También presenta una poderosa oportunidad para compartir ese aprendizaje asimilado con el resto del equipo para aumentar su aptitud horizontal.

¿Cómo puedes salir de esta situación?

- Centrarse en la enseñanza. No arregles los problemas; en su lugar, enseñe a otras personas a solucionar el problema. Tomará más tiempo solucionar los problemas, pero siéntase cómodo con eso. Esta es una inversión en la salud a largo plazo de usted y de la empresa.
- Desarrolle un estilo de trabajo en el que documente sobre la marcha. No documente nada una vez finalizado el proyecto; Para entonces estarás trabajando en otro proyecto. No intente documentarlo todo. Suponga que la audiencia es alguien en quien confía para que se haga cargo de esta función, pero que puede llamarlo si tiene preguntas.
- Al solucionar un problema, anime a otras personas a "tomar el teclado" y Guíelos para solucionar los problemas en lugar de solucionarlos usted mismo.
- Enumere los múltiples trabajos que realiza y trabaje con la gerencia para delegar cada trabajo a otra persona con usted como asesor. Cada nuevo empleado debe ser alguien que pueda hacerse cargo de parte de su trabajo para que usted pueda especializarse.
- Planee tomarse unas vacaciones de tres meses. La planificación obligará a tu organización desarrollar cobertura y formar a otras personas de la empresa.

Tengo una empresa/organización pequeña y no puedo permitirme contratar un equipo grande, por lo que debo contratar a un ingeniero que pueda hacerlo todo.

Muchos de nosotros hemos trabajado en nuevas empresas y pequeñas organizaciones que requieren que todos los miembros del equipo desempeñen muchas funciones. Este es un ejemplo perfecto en el que una sola persona debe hacerlo todo y convertirse en un experto en todo el conjunto. Sin embargo, como se ha abordado anteriormente, todavía existen riesgos y limitaciones para continuar operando en este modelo, incluida la creación de puntos únicos de falla, agotamiento y eventual reducción de

calidad. ¿Qué sucede cuando ese único ingeniero full-stack necesita tiempo libre por un evento de la vida, vacaciones o por razones de salud? ¿Cómo se mitiga el agotamiento de trabajar las veinticuatro horas del día, los siete días de la semana? ¿Cómo puede garantizar que todas las áreas puedan escalar con la calidad necesaria para que su negocio prospere?

En la práctica y con razón, los ingenieros full-stack establecidos, o los ingenieros colocados en roles en los que deben ser expertos en el full-stack, pueden exigir una compensación significativa. Por los mismos costos, los equipos y las organizaciones pueden contratar varios ingenieros en forma de T que tengan una mentalidad de crecimiento y estén dispuestos a trabajar juntos para construir un equipo completo.

Referencias

"Diez síntomas de sobrecarga de información y cómo afecta el cerebro y el cuerpo"

Learning Mind, consultado el 21 de junio de 2019, <https://www.learning-mind.com/síntomas-de-sobrecarga-de-información/>.

"The Science", Mindset Works, consultado el 21 de junio de 2019, <https://www.mindsetworks.com/science/>.

Beck, Melinda. "The Sleepless Elite", The Wall Street Journal, actualizado el 5 de abril de 2011, <https://www.wsj.com/articles/SB10001424052748703712504576242701752957910>.

Cox, Jason, Chivas Nambiar y Paula Thrasher. Aprendizaje continuo: creación de una carrera profesional inspirada en DevOps. Portland, Oregón: Revolución de TI, 2017. <https://itrevolution.com/book/continual-learning/>.

Dweck, Carol. "Lo que realmente significa tener una 'mentalidad de crecimiento'". Negocios de Harvard Revisión, 13 de enero de 2016. <https://hbr.org/2016/01/what-having-a-growth-mentalidad-realmente-significa>.

Evko, Tim. "No existe un desarrollador full stack". Desarrollador, 12 de febrero de 2018. <https://dev.to/tevko/theres-no-such-thing-as-a-full-stack-developer-1m3i>.

Fienen, Michael. "El mito del desarrollador Full Stack". El blog de Aquent, noviembre 12 de diciembre de 2015. <https://aquent.com/blog/full-stack-developer-myth>.

"Desarrollador Full Stack". SkillGigs.com, consultado el 21 de junio de 2019. <https://www.skillgigs.com/full-stack-developer/?nabe=4589400386961408:0,5551131242266624:1>.

"Descripción del puesto de desarrollador Full Stack". Betterteam.com, consultado el 21 de junio de 2019. <https://www.betterteam.com/full-stack-developer-job-description>.

Gerasimov, Alexey. "¿Son los equipos Full Stack una utopía de TI?" The Doppler, 8 de agosto de 2018. <https://www.cloudtp.com/doppler/full-stack-teams/>.

"¿Es posible convertirse en desarrollador full stack en 1 año?" Quora. Última modificación el 18 de junio de 2018. <https://www.quora.com/Is-it-possible-to-become-a-full-stackz-developer-en-1-año>.

Katrompas, Alejandro. "La dura verdad sobre el desarrollador full stack. Mitos y mentiras". Medio, 9 de julio de 2018. <https://medium.com/@alexkatrompas/the-hard-truth-acerca-de-los-mitos-y-mentiras-del-desarrollador-full-stack-945ffadeeb8c>.

- Liu, Trista. "Seis consejos esenciales sobre cómo convertirse en un desarrollador Full Stack". *tarde-hack*, 24 de diciembre de 2017. <https://hackernoon.com/6-essential-tips-on-how-to-conviértete-en-un-desarrollador-full-stack-1d10965aaead>.
- Maslach, Christina, Jackson, Susan E. y Leiter, Michael P. *Maslach Burnout Inventor*. 3ª edición. Palo Alto, CA: Consulting Psychologist Press, 1996.
- Maslach, Christina y Michael P. Leiter, "Comprensión de la experiencia del agotamiento: investigaciones recientes y sus implicaciones para la psiquiatría", *World Psychiatry* 15, no. 2 (2016): 103–111, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4911781/>.
- Schwartz, Mark, Jason Cox, Jonathon Snyder, Mark Rendell, Chivas Nambiar, Mustafa Kapadia y Alyson Hoffman. *Entornos de pensamiento: evaluación Modelos organizativos de DevOps para acelerar los negocios y empoderar a los trabajadores*. Portland, Oregón: Revolución de TI, 2016. <https://itrevolution.com/book/thinking-ambientes/>.
- Shafer, Andrés. "No hay escasez de talento". Compartir diapositivas. Publicado por Andrew Shafer, 23 de octubre de 2013. <https://www.slideshare.net/littleidea/there-is-no-talent-escasez-de-velocidad-2013>.
- Sharma, Gaurav. "¿Qué empleado con forma de letra es usted?" LinkedIn.com. 7 de mayo de 2018. <https://www.linkedin.com/pulse/what-letter-Shaped-employee-you-gaurav-sharma/>.
- Shora, Andy. "El mito del desarrollador Full-Stack". *Andyshora.com* (blog), consultado el 21 de junio de 2019. <https://www.andyshora.com/full-stack-developers.html>.
- Desbordamiento de pila. Encuesta anual para desarrolladores de Stack Overflow: 2019. Consultado el 21 de junio. 2019. <https://insights.stackoverflow.com/survey>.

Notas

1. Mark Schwartz, et al., Entornos de pensamiento.
2. Stack Overflow, Encuesta anual para desarrolladores de Stack Overflow: 2019.
3. "Desarrollador Full Stack".
4. "Descripción del puesto de desarrollador Full Stack".
5. Stack Overflow, Encuesta anual para desarrolladores de Stack Overflow: 2019.
6. Esta estadística se basa en el estudio interno de Hewlett Packard de sus propios registros de empleados.
Encontrado en "Por qué las mujeres no solicitan empleos a menos que estén 100% calificadas" de Tara Sophia Mohr, Harvard Business Review, 25 de agosto de 2014, <https://hbr.org/2014/08/why-women-dont-apply-for-jobs-unless-they-re-100-qualified>.
7. Beck, "La élite insomne".
8. Maslach y Leiter, "Comprensión de la experiencia del agotamiento".
9. "Diez síntomas de sobrecarga de información", Learning Mind.
10. "La ciencia", Mindset Works.
11. Un buen documento de referencia o diseño de organización para promover entornos de pensamiento.
ments es Thinking Environments de Mark Schwartz, et al.
12. Dweck, "Lo que realmente significa tener una 'mentalidad de crecimiento'".

Un agradecimiento especial a nuestro patrocinador

Nuestra misión para el Foro es reunir a líderes tecnológicos de muchas industrias y facilitar un diálogo que resuelva problemas y supere obstáculos en el movimiento DevOps. Durante tres días en este evento privado, reunimos a 50 de los mejores pensadores y emprendedores en el espacio DevOps para abordar los desafíos más difíciles de la comunidad. Les pedimos a estos líderes de opinión que colaboren y generen una guía con sus mejores soluciones a los desafíos.

Nos gustaría agradecer a todos nuestros asistentes y a nuestros amigos de XebiaLabs por ayudar a que el Foro de este año fuera un gran éxito.

