



# Introducción a DevOps y DevSecOps

Un guía para entender la revolución en IT



**DEVSECOPS**  
Argentina

# ¿Que veremos?

- Autores
- ¿DevOps?
- ¿Cómo empezó todo?
- DevOps Manifest
- Analizando INFRA y DEV
- Un poco de realidad
- Cambios necesarios
- Infraestructura ágil
- Cultura DevOps
- Aplicando la cultura
- El especialista DevOps en el mundo corporativo
- El especialista queda en INFRA o en DEV?
- ¿Cuáles son los beneficios de adoptar la cultura DevOPs?
- Atando nudos
- DevOps en la era cloud
- DevOps y seguridad
- DevOps + SecOps = DevSecOps
- Links recomendados
- Referencias
- Conclusiones



# Sobre los autores

## Jose Augusto Carvalho - Guto



<https://www.facebook.com/gutocarv>



<https://www.linkedin.com/in/gutocarv/>



<https://twitter.com/gutocarvalho>



Soy una persona simple, amiga, fan de la tecnología que hacen la vida más fácil y un entusiasta del movimiento de código abierto. Creo que compartir el conocimiento es una de las maneras más eficaces para construir una sociedad más justa, coherente y evolucionada, yo creo que el conocimiento debe ser libre y que todos deben tener acceso a el.

Tengo 34 años (12 / mayo), Estoy casado y tengo dos hijos, nací en Campo Grande / MS Brasil

Tengo un profundo respeto por la naturaleza, Me gusta la música de calidad como MPB, la bossa nova, el choro, el jazz, la música clásica e instrumental, también la literatura, la poesía y sobre todo películas y series.

Actualmente vivo en Brasilia / DF, soy sysadmin - por la auto-enseñanza - y mi foco se centra en las tecnologías de código abierto. En los últimos años me he especializado en la automatización de la infraestructura, especialmente la automatización utilizado Puppet.

Consultor y instructor experto en TI con un enfoque en ambientes UNIX y LINUX Cuenta con más de 14 años de experiencia en la gestión de redes informáticas, servidores y servicios. Trabajó en varios proyectos de software libre.

Profesor de eventos especializados como FISL, CONSEGI, ENCOMP, FLISOL, PGDAY, FSD, el Latinoware y CLOUDCONF.  
Blogger, microblogueiro y mantenedor de wikis técnicas públicas.



# Sobre los autores

## Luciano Moreira – Lucho



<https://www.facebook.com/luciano.m.cruz>



<https://www.linkedin.com/in/lucianomoreiradacruz/>



[https://twitter.com/Luciano\\_m\\_cruz](https://twitter.com/Luciano_m_cruz)



Soy una persona apasionada por la tecnología y sobre todo por la seguridad de la misma en todos sus estados. Como Guto también creo que el conocimiento debe ser libre y que las personas deberían buscarlo todo el tiempo.

Tengo 35 años (9/ mayo) Estoy casado, nací en Curitiba Brasil pero vivo en Buenos Aires hace 15 años.

Me gusta la buena música (aficionado en violín y en un futuro piano), Me gusta el futbol, Squash y patinar con amigos, leo y escribo mucho sobre tecnología en general, me gusta mirar películas y sobre todo muchas series.

Soy vicepresidente del capítulo argentino de la CSA y fundador del capítulo argentino de DevSecOps, tengo 15 años de experiencia en IT, en las áreas de Infraestructuras híbridas y Seguridad de la Información y desarrollo.

Trabaje en una docena de proyectos en todas las capas de seguridad. Responsable del diseño, implementación y gestión de proyectos con actividades como el análisis de riesgos, análisis de las deficiencias, cumplimiento, desarrollo de políticas de seguridad, seguridad en desarrollo, recuperación de desastres, plan de continuidad de negocio y seguridad en cloud.

Sólidos conocimientos y experiencia en proyectos de implementación y mejora de sistemas de gestión de la calidad y de seguridad de la información, auditoría y control de cumplimiento de normas y estándares naci. Instructor de diversas charlas y cursos en distintos centros de entrenamiento y empresas

**Elegido Cybersecurity Consultant of the Year en los premios Cybersecurity Excellence Awards 2016 y 2017**

Miembro de ISSA, ISACA, OWASP. Así como de sus capítulos locales.

Miembro del comité académico de los eventos E-gisart y Cyber de ISACA

Miembro del comité científico en Ciberseguridad del evento IEEE ARGENCON 2016



# Sobre los autores

## Leonardo Rosso – Leo



 <https://www.linkedin.com/in/lrosso/>

 [https://twitter.com/Leonardo\\_Rosso](https://twitter.com/Leonardo_Rosso)

Me gusta describirme como un buscador de soluciones en todos los ámbitos laborales y personales. Estoy en la misma situación que Lucho, Guto y Chris me gusta mucho lo que hago y sobre todo el lado académico de esta profesión.

Tengo 39 años (25/ septiembre) Nacido en Argentina, me gusta decir que soy un esposo de la mas hermosa mujer, un padre baboso de dos hermosas señoritas y un apuesto señorito.

Un lector de cuanto material escrito se me ponga al alcance de la mano, pero pésimo blogger. Un opositor a la política de Facebook de abusar de manera constante de la privacidad de sus usuarios. Un Supreme Nerd si tomamos NerdTests.com en serio [http://www.nerdtests.com/ft\\_nq.php](http://www.nerdtests.com/ft_nq.php)

Actualmente soy Presidente del capítulo argentino de la CSA, Soy un profesional de IT desde fines de los años 90, y de seguridad de la información desde el año 2001.

Especialista en seguridad de infraestructura tecnológica, políticas de seguridad, cumplimiento regulatorio, auditoria de seguridad y concientización de usuarios.

Experiencia en múltiples proyectos de implementación de tecnologías, mecanismos de seguridad lógica y consultoría en seguridad de la información, durante los últimos años con foco en seguridad en desarrollo y arquitectura de seguridad en entornos de computación en la nube.

Soy miembro activo de las asociaciones ISSA, ISACA, ISC2, OWASP, CSA, DevSecOps y FSF.

Reconocimiento: **CISA Geographic Excellence Award**



# Sobre los autores

## Christian Ibiri – Chris



 <https://www.facebook.com/Christian.Ibiri>

 [www.linkedin.com/in/christian-ibiri](http://www.linkedin.com/in/christian-ibiri)

 <https://twitter.com/Christianibiri>

Soy una persona apasionada por la tecnología y sobre todo las disruptivas o las que transforman la forma en que estamos acostumbrados de hacer las cosas, como computación en la nube, metodologías Agile, o Infraestructuras Agiles.

Me gusta mucho estudiar y escribir sobre estas temática en diferentes medios relacionados con la tecnología.

Tengo 25 años (7/ mayo) soltero, nacido en Argentina.

La verdad me encanta la época que estamos viviendo hoy por hoy donde uno puede levantar varias instancias para correr lo que se necesite sin tener que contar con un parque computacional enorme, mucho mas si me acuerdo de mis principios donde teníamos que desplegar maquinas físicas, armar clusters, “virtualización si teníamos suerte”.

Actualmente formo parte de la comisión directiva del capítulo argentino de la CSA y DevSecOps, tengo 8 años de experiencia en IT, cuales los últimos 4 años en las áreas de Infraestructuras hibridas, cloud, DevOps tooling.

Me gusta la buena lectura, leo mucho blogs de tecnología en mi tiempo libre me gustar ver películas y series.

Participo en varios proyectos de infraestructura, networking, migración y implementación de clouds privadas y publicas, automatización, comunicaciones unificadas y colaboración, acompañando a las demás partes involucradas en los mismos, desde la etapa de requerimientos hasta la implementación y pos-implementación. Creo en la interoperabilidad y que el mundo opensource puede tranquilamente coexistir con tecnologías propietarias sin ningún tipo de limitaciones.

Entusiasta de DevOps, y infraestructuras agiles o infraestructuras como código.

**Microsoft**  
**CERTIFIED**  
Solutions Expert

SharePoint  
Server Infrastructure  
Cloud Platform and  
Infrastructure

Communication  
Private Cloud  
Productivity

# ¿Devops?

Este e-book es la primer entrega de DevSecOps Argentina con la colaboración de [Cloud Legion](#) vamos hablar sobre la palabra de moda **DevOps**.

En este ultimo tiempo varios clientes y conocidos me están preguntando mucho sobre este tema y normalmente buscan respuestas a las siguientes preguntas:

- ¿Qué DevOps?
- ¿DevOps es un movimiento?
- ¿DevOps es una filosofía, un concepto o una cultura?
- ¿DevOps es una metodología?
- ¿DevOps es algún tipo de ambiente o grupo de herramientas?
- ¿DevOps es un especialista en desarrollo que entiende de infraestructura?
- ¿DevOps es un especialista Sysadmin que entiende de desarrollo?
- ¿DevOps es una posición? es un sector o un departamento?
- ¿DevOps funciona sólo en Startups o sirve para el ambiente corporativo?
- ¿DevOps es algo nuevo?

A lo largo del texto la idea es responder a todas las cuestiones, pero para responderla primero necesitamos entender algunas cosas previamente.



# ¿Cómo empezó todo?



Para que podamos comprender totalmente, tenemos que ir al corazón de esta historia. El movimiento **DevOps** no comenzó en un solo lugar, hay muchos lugares que dan pistas sobre el origen del término, pero al parecer la información más **concreta** sobre el origen de este movimiento nos lleva al año 2008. En este año, empezaron a utilizar el término **Infraestructura Ágil** en algunos foros de debates con foco en el desarrollo ágil, y en la misma época durante el evento Agile 2008, surgieron charlas que abordaban el tema metodología ágil para la administración de infraestructura, inspirada en el modelo ágil de desarrollo, luego fue el foro de debate europeo con el nombre **agile-sysadmin** que empezó a abordar el tema con propiedad, con eso ayudaron a colocar los primeros ladrillos en el puente que haría la conexión entre los developers y sysadmins. Un participante de este foro **Patrick Debois** (@patrickdebois) **the godfather**, era uno de los más activos, también un gran entusiasta del tema.

El término **DevOps** fue creado solamente de hecho en 2009 durante la Conferencia Velocity da O'Reilly, en esta conferencia John Allspaw (Etsy.com) y Paul Hammond (Typekit) presentaron el trabajo **10+ Deploys Per Day: Dev and Ops Cooperation at Flickr**, ver en el link abajo los slides de la presentación

[10+ Deploys Per Day: Dev and Ops Cooperation at Flickr](#) from [John Allspaw](#)



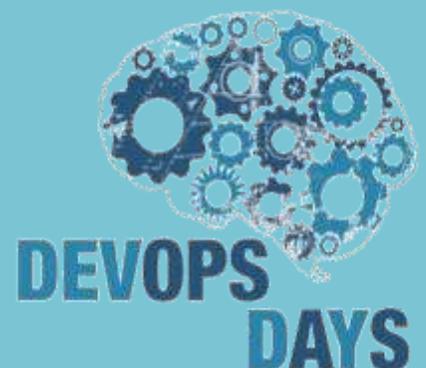
# ¿Cómo empezó todo?

El encuentro dejó a **Patrick Debois** muy emocionado, entonces tuvo la idea de crear una reunión/evento llamado **DevOpsDay**, este evento tuvo lugar en Ghent a finales de 2009, fue un encuentro de dos días y al parecer es donde todo comenzó.

Desde entonces, Patrick Debois, Gildas Le Nadan, Andrew Clay Shafer, Kris Buytaert, Jez Humble, Lindsay Holmwood, John Willis, Chris Read, Julian Simpson, R.I. Piennar (mcollective) y muchos otros llevaron el evento a varios lugares, incluyendo:

<b>2009</b>	<b>2013</b>	<b>2014</b>	Denver	Vancouver	Raleigh
Ghent	New York	Nairobi	New York	London	Boise
<b>2010</b>	(Winter)	Ljubljana	Austin	Denver	Singapore
Sydney	New Zealand	Austin	Toronto	Atlanta	Detroit
Mountain View	London (Spring)	Pittsburgh	Washington, DC	Austin	Havana
Hamburg	Paris	Amsterdam	Amsterdam	Kiel	Kansas City
Sao Paulo	Austin	Silicon Valley	Minneapolis	Seattle	Philadelphia
<b>2011</b>	Berlin	Minneapolis	Minneapolis	Toronto	Ghent
Boston	Amsterdam	Brisbane	Melbourne	Istanbul	Ohio
Melbourne	Silicon Valley	Boston	Pittsburgh	Washington, DC	Madison
Mountain View	Downunder	Toronto	Chicago	DC	Bangalore
Bangalore	Tokyo	New York	Bangalore	Salt Lake City	Cape Town
Göteborg	Tel Aviv	Warsaw	Boston	Silicon Valley	Nashville
Manila	Atlanta	Chicago	Tel Aviv	Amsterdam	Berlin
<b>2012</b>	Barcelona	Berlin	Singapore	Porto Alegre	Brasília
Austin	New York (Fall)	Belgium	Berlin	Minneapolis	Warsaw
Tokyo	Helsinki	Helsinki	Charlotte	Portland	Paris
Mountain View	Vancouver	Vancouver	Silicon Valley	Boston	Sydney
Delhi	Tel Aviv	Tel Aviv	Detroit	Chicago	Tel Aviv
Rome	Bangalore	Bangalore	Ohio	Oslo	
	London (Autumn)	<b>2015</b>	Warsaw	Dallas	
	Bangalore	Ljubljana	<b>2016</b>	New York	
		Paris	Los Angeles	Wellington	

Es importante mencionar que, por llevar el evento a varios países, estas personas fueron los encargados de difundir la cultura DevOps alrededor del globo, directa o indirectamente, se han convertido en la fuerza motriz de una revolución en el mundo de IT.



# DevOps Manifest

Aunque se ha organizado los DevOpsDays en varios países, no fue establecido todavía un manifiesto para este tema, existen muchas interpretaciones con relación a este término, muchas basadas en el Manifiesto de Agile, distintos profesionales creando sus propias versiones en base a sus experiencias, pero nada consensuado y validado entre la mayoría

Pero antes de argumentar sobre el posible contenido de un manifiesto DevOps, primero tenemos que entender la dinámica en la relación entre las áreas de infraestructura (**INFRA**) y el desarrollo (**DEV**).



# Analizando a INFRA y a DEV

Para entender mejor lo que **DevOps** significa, necesitamos analizar de forma práctica y directa la vida de los sysadmins, desarrolladores y la vida cotidiana de estas áreas, esperamos que esto los ayude a entender mejor el tema.

...Imaginemos hipotéticamente una empresa que desarrolla aplicaciones web, en su mayoría portales de noticias y en algunos casos también realizan aplicaciones web internas (recursos humanos, finanzas, administrativas), en esta compañía el desarrollador trabaja con PHP, PYTHON, RUBY y JAVA.

Para una mejor comprensión del ejercicio, consideremos las dos características abajo como cotidianas en esta empresa ficticia:

1. **DEV** empieza a trabajar con metodologías ágiles (proactivo, evolutivo y continuo).
2. **INFRA** continúa trabajando en el modelo tradicional de administración (manual, caótica y reactivo).



# Analizando a INFRA

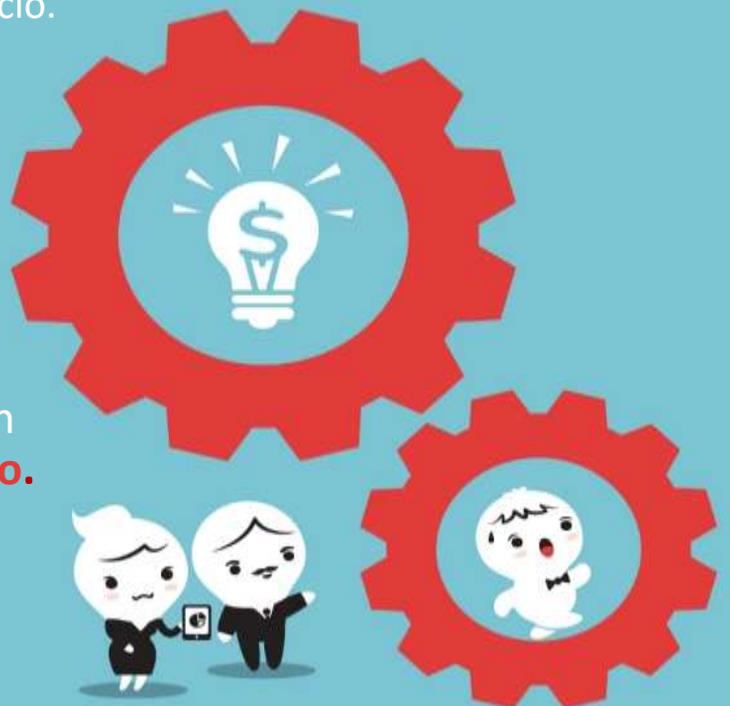
INFRA está compuesta en parte por los sysadmins, estos chicos y chicas tienen la misión del mantenimiento de los sistemas y su normal funcionamiento, hacen los despliegues y las rollbacks de las versiones anteriores de las aplicaciones que desarrolla DEV, es su responsabilidad mantener intacto el entorno de producción.

Los sysadmins tienen que ejecutar las aplicaciones, supervisar la operación, funcionamiento, evaluar y proponer mejoras para mantener las aplicaciones bajo su cuidado a todo vapor, rápida y estable, además, deben planear los cambios con cautela, tratando de minimizar los riesgos.

Les preocupa sobre todo la seguridad, estabilidad y el acuerdo de nivel de servicio (SLA) para cada producto bajo su responsabilidad, esta preocupación es fundamental para el negocio.

Entienda que si una aplicación deja de funcionar esto impactará en el SLA, que puede significar importantes pérdidas financieras relacionadas con el producto, después de todo un producto sin funcionar significa cliente insatisfecho trayendo perjuicios, ya sea financieros, institucionales, o en la sustentabilidad del producto, esto significa que el proveedor de este servicio, o sea el empleador de los sysadmins, puede ser multado. Lo que ocurre directamente es disminución en el valor del negocio.

En Resumen, INFRA  
(sysadmins) se preocupa en  
**proteger el valor del negocio.**



# Analizando a DEV

DEV se compone en parte por los desarrolladores estos chicos y chicas trabajan con lógica y creatividad, pasan la mayor parte de su tiempo codificando soluciones y centran su trabajo en los requisitos que el analista funcional pudo mapear junto con el cliente.

Los desarrolladores están constantemente creando y mejorando sus aplicaciones con nuevas versiones de esta y necesitan estar disponibles, así los clientes pueden tomar ventaja de las capacidades solicitadas.

Nueva versión significa desplegar de nuevo y si ocurre un problema, esto va a exigir un rollback, ambos procedimientos implican trabajo no estimado de los equipos de infraestructura.

En Resumen, podemos decir que DEV se preocupa en **aumentar el valor de negocio.**



# ¿Dónde está el conflicto?

Los desarrolladores normalmente quieren poner sus aplicaciones en el aire lo antes posible, sin embargo, los sysadmins quieren asegurarse de que la aplicación es lo suficientemente estable como para entrar en producción sin generar incidentes.

En los últimos años este conflicto fue latente el mundo de IT, algunas compañías tenían reglas tan rígidas que sólo permitían desplegar una vez a la semana, en casos más rígidos sólo una vez al mes, todo esto pensando en la protección del negocio.

Con DEV cambiando de metodología, ni que decir, este método de deploys de una vez por semana no encaja con el desarrollo ágil, INFRA tuvo que evolucionar a fórceps, y quien antes hacía deploys una vez a la semana, tuvo que aprender a hacerlo varias veces al día.

Por supuesto, INFRA trabajando con los métodos que usaba (desplegar 1 vez por semana y manual) no podía fluir con las demandas, y también es obvio que DEV no tenía una infraestructura adecuada para el desarrollo de forma continua.



# ¿Dónde está el conflicto?

Además de todo esto, por lo general DEV no conoce y no tiene como prever aspectos importantes relacionados a la infraestructura que se encuentra de cara al cliente, así que cuando la aplicación entra en producción, generalmente se producen constantes y pequeños incidentes que causan una gran pérdida de valor en el negocio. Traducción, son los típicos ajustes en la aplicación que necesitan ser hecho en el último minuto porque el ambiente de desarrollo es completamente diferente al de producción.

El cliente a su vez se queja y con razón, luego la gerencia de IT queda tratando de encontrar el dueño del problema (caza de Brujas), por otro lado, DEV se excusa diciendo que INFRA es rígida, lenta y que no ofrece un ambiente adecuado para desarrollar sus aplicaciones, en el otro lado INFRA se excusa diciendo que DEV hace un código malo e inestable y que no es su culpa si la aplicación no funciona.

Soy de infra/Seguridad hace muchos años, pero tengo que decir que infraestructura debido a las culturas arcaicas de administración y herencia de tiempos de mainframes, tiene mucha culpa en este escenario, pero DEV también tiene sus problemas, después de todo, como están empezando a aplicar metodologías ágiles, y aún están creando la cultura de pruebas y aseguramiento de la calidad (QA).



# Incidentes

Cuando un incidente ocurre, normalmente escucharas desde INFRA hablando a DEV que el problema no es en sus máquinas, sino el código, y ciertamente DEV hablará a continuación que el problema no es el código, y que son las máquinas y probablemente todavía van a decir que el sistema está corriendo ok en su notebook **“lamentablemente esto es algo cotidiano”**.

Espero que en este punto ustedes ya puedan visualizar el problema.

Uno debe entender que INFRA y DEV trabajan en nichos, cada uno en su cuadrado, cada uno en su realidad y ninguno de ellos están muy dispuestos a cambiar su cultura, ninguno de ellos está dispuesto a ceder. INFRA no conoce a DEV e no sabe cómo cambiar para ayudarlo, e DEV no conoce a INFRA e no sabe cómo pedir lo que necesita.

Al final de cuentas, las personas no pueden establecer una forma sana y eficiente de comunicación y con esto, no hay colaboración entre estas dos áreas. ¿pero normalmente que produce ese conflicto?



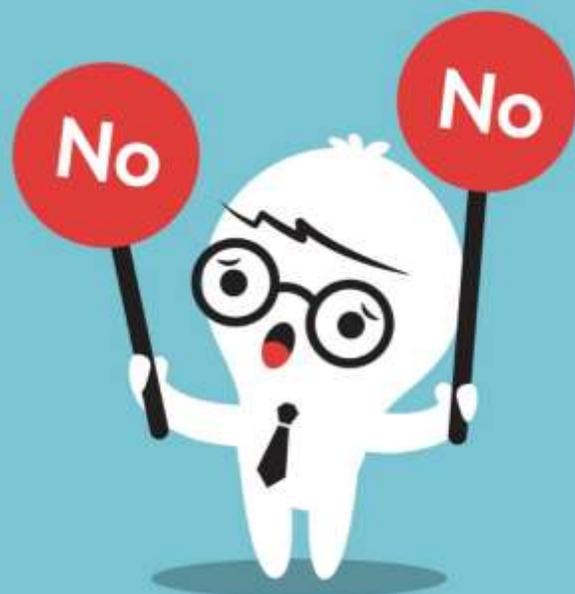
# El combustible de los conflictos

Antes presenté el conflicto común entre áreas, pero existe el combustible que lo mantiene prendido, ese combustible es el comportamiento del sysadmin típico, por lo tanto, de hecho, hay una razón para tener tanto odio hacia ellos, veamos:

- Ellos dicen no
- Ellos dicen no por segunda y tercera vez
- Ellos dicen no todo el tiempo por varias razones, a varios pedidos
- Ellos toman, retrasan y pierden tiempos de pedidos de DEV
- Se niegan a cambiar/romper cosas, incluso si es para encontrar el problema
- Ellos se preocupan con el UPTIME y no con el negocio
- No hacen el esfuerzo para ayudar a DEV a encontrar el problema
- Ellos piensan que el problema de DEV no es su problema
- No ven el negocio y no logran ver que INFRA y DEV son parte de un todo

Todo esto forma parte de ese comportamiento arcaico que hemos mencionado antes, queríamos fortalecer porque es bueno mostrar las raíces de un problema para ayudar en la reflexión de que necesitamos cambiar.

Podemos ver que esa conducta es inaceptable e incompatible con el mundo de ágil hoy, Igualmente todavía es muy común encontrar personas que tienen este tipo de actitud y perfil.



# Un poco de realidad

¿Recuerden que dijimos que INFRA se preocupa en proteger el negocio y DEV se preocupa por las formas de agregar valor al negocio?

**Olvida lo que dijimos, eso funcionaba en los años 70/80/90, pero hoy en día no es suficiente.**

INFRA debe entender que su obligación es proporcionar los medios para hacer que el negocio fluya, y esto también es el papel de DEV.

Ambos equipos necesitan cambiar la forma de pensar y de manejarse, pero tienen que ser conscientes de que los cambios están asociados con posibles problemas, un cambio puede romper su producto y afectar a su negocio.

**¿Entonces, cuál es la receta mágica?**

**¿Cómo cambiar sin afectar el negocio?**



# Cambios necesarios

INFRA necesita evolucionar y necesita hacerlo rápidamente.

DEV necesita tener control de todas las etapas de deploy.

INFRA necesita comenzar a trabajar de forma automatizada y dinámica, necesita ser más veloz en provisionar nuevos entornos o incluso reconstruir/duplicar los entornos existentes para satisfacer las necesidades de DEV, no se puede más trabajar manualmente y utilizar las mismas metodologías de la época de los mainframes.

DEV tiene que pasar sus necesidades a INFRA y tiene que esforzarse por hacer de forma clara, que INFRA entienda, aunque cueste la primera vez.

## **búsqueda de soluciones**

Fue la búsqueda de soluciones a estas necesidades que motivó debates importantes en la informática mundial, donde entonces empezaron a hablar de '**Infraestructura ágil**' en el año 2008. entendamos de que se trata.



# Infraestructura ágil

## Infraestructura como código

La discusión sobre **infraestructura ágil** cobró impulso con el crecimiento de dos tendencias, la **Virtualización y el cloud computing**. Puesto que 2003 las empresas empezaron a vivir con los entornos virtualizados, luego un parque con pocas máquinas físicas podría convertirse en un parque con decenas máquinas virtuales, y después de la reciente llegada de la nube, docenas de máquinas virtuales se pueden hacer cientos o miles de instancias a ser administradas en la nube.

No había más espacio para trabajar en infraestructura de la manera tradicional, era necesario dar un paso adelante y pensar en **infraestructura como código**, especialmente cuando nos detenemos a analizar el reciente auge de los **Startups**, empresas pequeñas con amplia gama de productos, que se ejecutan en cientos de instancias en la nube, brindando servicio a millones de usuarios, y todo esto es administrado por equipos mínimos.

El objetivo es desplegar no sólo aplicaciones, pero desplegar infraestructura de forma rápida y controlada.

Esto significa que si usted necesita subir un ambiente JBOSS lo vas hacer en minutos en lugar de días.

Infraestructura Ágil es esencialmente un conjunto de buenas practicas a ser utilizadas dentro de su equipo de IT, partiendo desde la visión de los sysadmins conectados a los modelos agiles.



# Infraestructura ágil

## herramientas de infraestructura ágil

Cuando se trata de infraestructura ágil lo que viene a la mente son herramientas, y de hecho forman parte de él.

Básicamente tenemos tres tipos de herramientas, las mismas son:

1. Orquestadores
2. Herramientas para la gestión de la configuración
3. Herramientas de provisioning y bootstrapping

**Orquestadores** son herramientas que nos permiten ejecutar comando y control de nodos/instancias de nuestro parque en tiempo real. Algunos de estas son, Fabric, Capistrano, Func y Mcollective, etc...

**Herramientas de gestión de la configuración** típicamente controlan los estados de su sistema, ayudan a centralizar todas las configuraciones y facilitan la administración y creación de nuevos entornos. Algunos de ellos son Puppet, Chef, Cfgine, Salt, etc...

**Herramientas de bootstrapping** son las que ayuda a instalar un sistema operativo en una máquina física, en una máquina virtual, ya sea en una instancia en la nube, entre ellos tenemos algunos proveedores de nube como AWS, Azure y Rackspace que ya ofrecen este nativamente, son también herramientas como Kickstart y Cobbler que actúan en este segmento, entre otras.

La combinación de estos tres tipos de herramientas nos permite tener lo que llamamos **infraestructura ágil**.

Pero a pesar de la calidad y beneficio en el uso de estas tecnologías, las herramientas solas no resolverán sus problemas, necesitaras cambiar la cultura y la forma de trabajar de infraestructura.



# Equipo de Infraestructura ágil

Los equipos que trabajan con una infraestructura ágil también necesitan un método diferenciado de la organización, normalmente estos equipos están trabajando a raíz de estos ejes:

1. **Control de versiones de código y archivos de configuración (git)**
2. **Organización de actividades en formato visual (tablero KANBAN)**
3. **Trabajar en pares**
4. **División de actividades en sprints**
5. **Reuniones diarias ágiles (reunión de standup 10 minutos en pie)**
6. **Reuniones periódicas ágiles (retrospectiva y la planificación de Sprint).**

**Parece SCRUM pero no es, pero está fuertemente inspirado en él y sobre todo Kanban.**



# DevOps e infraestructura ágil, entonces son la misma cosa?

**No, infraestructura Ágil es parte de la cultura de DevOps.**

DevOps depende de la infraestructura ágil, pero debería ser mucho más.

Infra Ágil e DevOps son movimientos con principios comunes pero con focos diferenciados.

DevOps es una cultura que propone un proceso de transformación que alcanza toda una organización a través de la integración, colaboración y retroalimentación.

Infraestructura Ágil tiene foco en transformar la forma de como el equipo de operaciones se relación entre si y como este equipo maneja la infraestructura de su organización. Para esto deberíamos partir de los mismos principios CAMS y ICE de DevOps aplicando al equipo INFRA, con eso el equipo ganara madurez y empezara a caminar en dirección a la cultura DevOps

A pesar de la evolución de la infraestructura, todavía falta algo que conecta las dos áreas, necesitamos un agente de cambio principalmente en el medio corporativo.



# Cultura DevOps

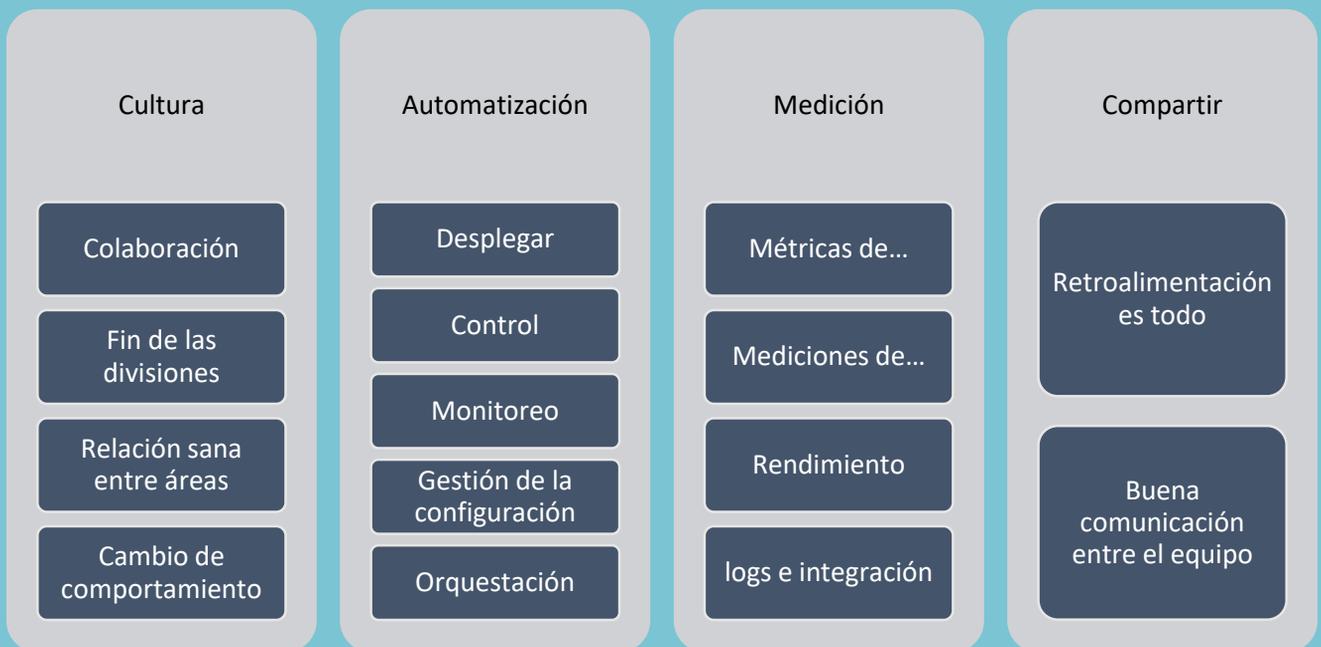
Es el momento para entrar en este tema, ahora vamos a profundizar nuestros estudios sobre la cultura de DevOps.

## características de la cultura DevOps

Para entender la cultura de DevOps sin hacer un texto muy largo, vamos a apuntar sus principales características.

## Con relación a las características de la cultura

Patrick Debois (quien acuñó el término) nos dice que es esencialmente una cultura, DevOps y describe el uso de 4 ejes, los mismos son:



# Cultura DevOps

## Con relación a las características técnicas

Un entorno de DevOps debe tener/poseer/proveer/permitir:

- Infraestructura como un código
- Orquestación de servidores
- Gestión de configuración
- Aprovisionamiento dinámico de los ambientes
- Control de versiones entre infraestructura y desarrollo
- Entorno de desarrollo, prueba y producción (mínimo)
- El ambiente debe permitir TDD
- INFRA debe participar desde el principio **[1]**
- INFRA debe participar en las reuniones de desarrollo **[2]**
- DEV debe asistir a las reuniones de INFRA **[3]**
- Ambiente de entrega continua **[4]**
- Los desarrolladores deben ser capaces de hacer deploy sin interferencias de INFRA **[5]**
- Supervisión efectiva con el procesamiento adecuado de eventos y mediciones
- Capacidad de respuesta rápida a incidencias y problemas
- Backup y restore confiables



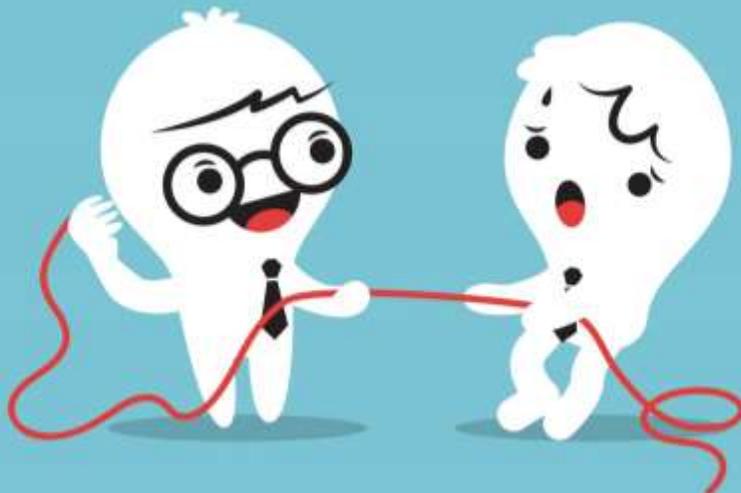
# Cultura DevOps

## Con relación a las características técnicas

**[1]** DEV debe involucrar a INFRA en los proyectos desde el comienzo, participarlos en reuniones técnicas o SCRUM, de todas formas, sin infraestructura no tenemos proyecto, además, cuantos más problemas se resuelvan durante el proyecto con la ayuda de INFRA, menos problemas se expondrá a los clientes.

**[2]** INFRA también tiene que observar cuales son las metas a largo plazo de la empresa, especialmente aquellas relacionadas con DEV porque viendo donde quieren llegar, puede planear mejor para asegurarse de que la infraestructura tecnológica estará disponible para atender cuando llegue ese momento.

**[3]** INFRA necesita involucrar a DEV en sus reuniones técnicas para que puedan comprender y ser conscientes de la realidad de INFRA, por lo que podrán ver sus calificaciones, asignaciones, planes de mejoras, actualizaciones programadas, horarios de mantenimiento, conocer los recursos disponibles y también descubrir las limitaciones del equipo, ya sean técnicas o materiales. Además, DEV puede ser un gran aliado de INFRA en la solución de problemas, después de todo el conocimiento que trae DEV puede ayudar a mejorar la forma en que administran su entorno, haciendo este proceso más eficiente. (una visión externa a veces es la mejor opción)

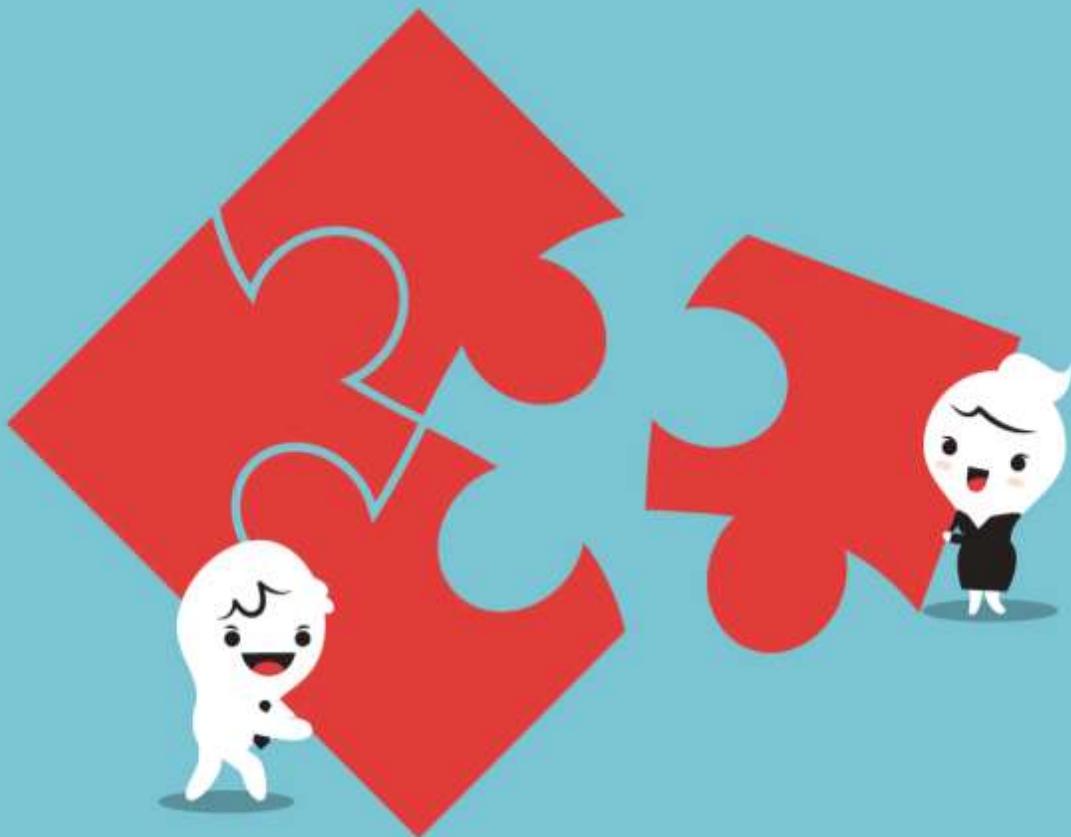


# Cultura DevOps

## Con relación a las características técnicas

**[4]** DEV necesita adoptar cualquier método de entrega o desarrollo continuo e INFRA necesita entender este proceso para que juntos puedan crear los ambientes con las herramientas adecuadas.

**[5]** INFRA tiene la necesidad de ceder un poco y evolucionar, necesita proporcionar un ambiente adecuado donde DEV son el propietario del producto, donde DEV pueda realizar todo el ciclo de desarrollo directamente y pueda ser capaz de generar y administrar el código, necesita comprometerse a la seguridad, necesita hacer el Build, testear el mismo, validar la aplicación y entregar las nuevas versiones de una manera transparente sin que deban pasar por un burocrático y rígido proceso de cambio.



# Los valores humanos

Para la adopción de la cultura, el equipo debe observar y ejercitar los siguientes valores:

- Confianza en la labor de su equipo
- Respeto personal y profesional para todos en el equipo
- Sinceridad acerca de incidentes y eventos
- Honestidad sobre las causas de los incidentes (no ocultar nada de su equipo)
- Entendiendo que el problema es responsabilidad de todos
- Comprensión de que la solución es responsabilidad de todos
- Comprensión de que los resultados son el reflejo del trabajo de todo el equipo
- Dinámica y comunicación efectiva
- Postura constructiva siempre
- Espíritu de colaboración

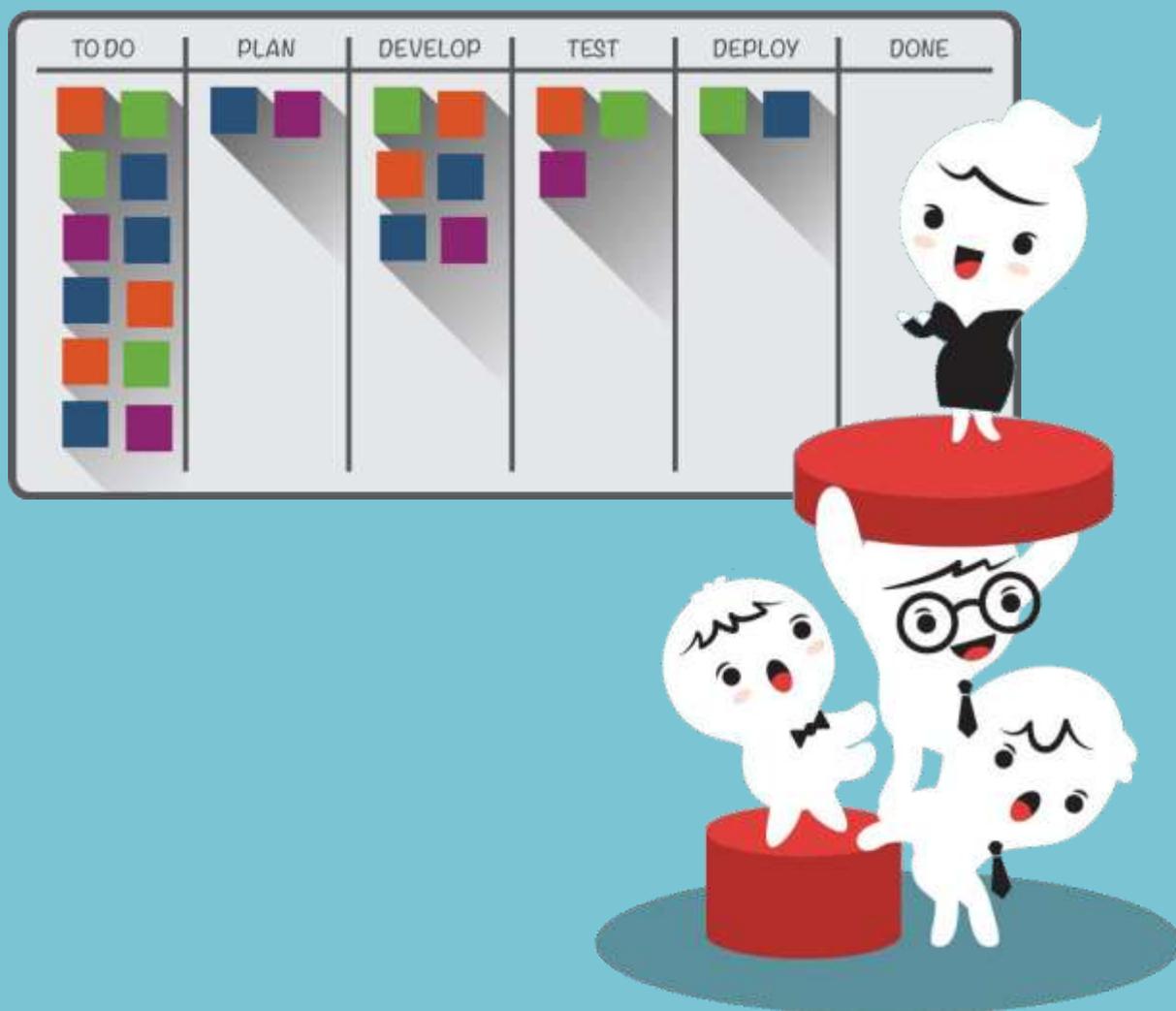


# La forma de trabajo

## Es recomendable que el equipo:

- Asimilar y adaptar metodologías ágiles como SCRUM y KANBAN para su día a día
- Profundizar estudios en entrega continua
- Profundizar estudios en gestión de la configuración y orquestación

Creo que es eso, características técnicas, valores humanos y forma de trabajar, espero que les sirva a ustedes.



# Aplicando la cultura

Después de observar las principales características de este movimiento, normalmente pensamos en cómo aplicar esto en nuestro mundo. Para ayudarlos en la reflexión vamos a evaluar el mundo Startup y corporativos.



# Aplicando la cultura

## la realidad en el entorno de Startup

La cultura **DevOps** encaja muy bien con las startups, en estos lugares generalmente ya se trabaja metodología desarrollo ágil, fue justamente en este nicho en el que comenzaron a discutir infraestructura ágil el movimiento precursor del **DevOps**, por lo tanto, la gente de este medio puede absorber los conceptos y la cultura de **DevOps** sin demasiada dificultad, pueden entender los principios de colaboración y retroalimentación porque esto hace parte de su vida diaria.

Quienes están en una startup no tiene las limitaciones y vicios del mundo corporativo, esto es un gran facilitador y no requiere ningún tipo de intervención a internalizar la cultura, a partir de la estimulación de un líder la gente empieza a estudiar y aplicar **DevOps** naturalmente.

En las startup normalmente no existen las divisiones, departamentos, todos trabajan juntos y esto también es un facilitador, después de todo no existen barreras para comunicarse.



# Aplicando la cultura

## la realidad en el ambiente corporativo

La corporación no funciona como una startup, hay burocracia y el uso vicioso de métodos anticuados, por lo que no será suficiente el estímulo de la alta dirección hacia los equipos de infraestructura y desarrollo para que comiencen a experimentar la cultura DevOps, en este tipo de entorno, en mi opinión personal y profesional, es necesario intervenir quirúrgicamente para lograr la internalización de la cultura de DevOps.

En Resumen, usted necesita a alguien externo que sepa de Infraestructura Agil y DevOps para que esta persona pueda ir contagiando a otros.

Debe ser un apoyo tanto en los roles de sistemas como de programación. Debe ayudarte a diseñar nuevos modelos operativos teniendo en cuenta los riesgos, oportunidades y necesidades del negocio. Lo más importante, debe conocer tu actividad, tu sector y tu gente para ser capaz de diseñar una solución a la medida exacta de tus circunstancias y de su evolución.

Este proceso es lento, pero si el experto tiene los medios y el apoyo de los mandos altos, pueden ocurrir cambios fantásticos.



# El especialista DevOps en el mundo corporativo

Normalmente buscado para actuar como un agente de cambio, él tiene que contagiar a las áreas y mostrar que la cultura de DevOps funciona.

## Característica de un experto DevOps en 2016

En la casa de entre 30 años o más

Es un profesional senior en infraestructura

Tiene una buena base en el desarrollo o código

Tiene una buena formación en metodologías ágiles

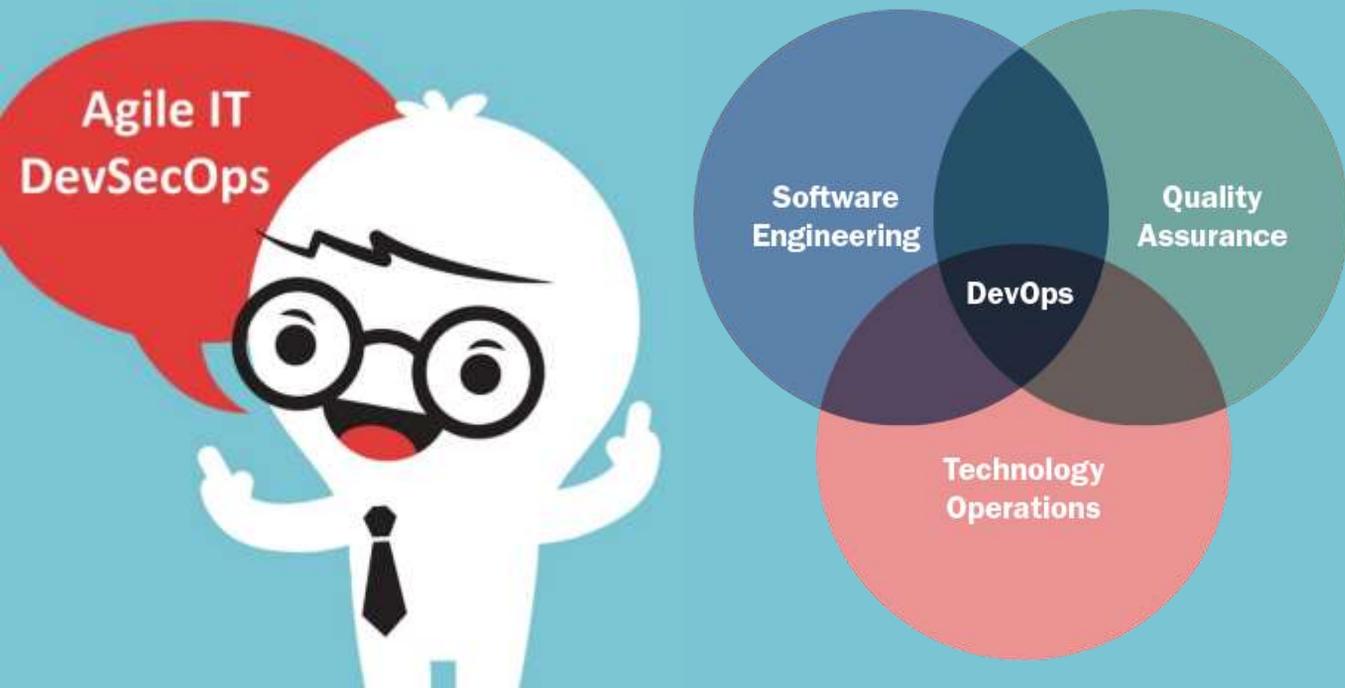
Tiene conocimientos sólidos de soluciones opensource y similares

Trabaja intensamente con automatización e infraestructura como código

## donde actúa este experto?

Este especialista en **DevOps** normalmente tendrá un pie en infraestructura y otro en desarrollo, en algunos casos también tienen un pie en las áreas de seguridad y aseguramiento de calidad (QA).

**Es el pegamento que faltaba para unir infraestructura, desarrollo, Seguridad y calidad.**



# El especialista DevOps en el mundo corporativo

## Cómo actúa este especialista?

Él será el puente, embajador y a la vez como evangelista entre las áreas de infraestructura y desarrollo, conoce la infraestructura a fondo y comprender ampliamente los procesos de desarrollo ágil.

Se encarga de alinear el desarrollo con las necesidades funcionales, operativas y de calidad. Esta labor de negociación y convicción requiere un alto nivel de conocimiento práctico en las tres áreas, junto con una visión clara de las necesidades del negocio .



# El especialista DevOps en el mundo corporativo

## pie en desarrollo

Participa en los proyectos de desarrollo desde su nacimiento, su enfoque es proporcionar las capacidades para que los desarrolladores puedan trabajar de la manera más eficiente, además, con su punto de vista de infraestructura toma todas las precauciones en los aspectos de seguridad, control, eficiencia y escalabilidad y que sean observados desde el comienzo del proyecto.

El DevOps estudiará todo el proceso de desarrollo integral y definirá junto con el equipo de DEV las herramientas que permitan un proceso de desarrollo y entrega continua. Después que se defina el instalara y mantendrá esta infraestructura.

Algunos DevOps pueden incluso evaluar el código del producto y ver problemas de performance, este tipo de visión sistémica y pensamiento rápido es un diferencial importante para una entrega con más calidad.



# El especialista DevOps en el mundo corporativo

## pie en Infraestructura

En infraestructura es el principal agente del cambio, es el que se compra el tema para iniciar la implementación de una infraestructura ágil, domina las herramientas de orquestación, gestión de configuración y aprovisionamiento y usará ese conocimiento para que su equipo pueda trabajar en una infraestructura como código.

Este profesional también les ayudará a cambiar su comportamiento y su cultura, les guiará en métodos ágiles de ejecución de las actividades inspiradas en el SCRUM y KANBAN.



# El especialista DevOps en el mundo corporativo

## ¿Este especialista queda en INFRA o en DEV?

Para internalizar DevOps, normalmente estas barreras no existen, INFRA y DEV debe habitar el mismo espacio, sin muros, sin particiones, todos en la misma habitación, es fundamental para extinguir los nichos y crear un equipo más unido y cohesionado.

Respondiendo a la pregunta, él debe estar junto con los dos equipos a la medida de lo posible, este es el mejor de todos los mundos.

Si no es posible que estén todos junto, el experto debe esforzarse para interactuar con ambos equipos en una base diaria.

El Será el agente de cambio hasta que INFRA y DEV comience a entender y adoptar la cultura de una manera natural.



# ¿Cuáles son los beneficios de adoptar la cultura DevOps?

Evaluaremos la perspectiva de cada área que mejora con la adopción de la cultura de DevOps.

## Beneficios para infraestructura

- Infraestructura como código (El equipo deja de administrar y pasa a desarrollar infraestructura)
- Infraestructura más eficiente y rápida utilizando métodos ágiles
- Equipo de infraestructura más organizado
- Equipo de infraestructura se comunica mejor
- logra hacer más en menos tiempo con menos gente
- Entornos de gestión de configuración, orquestación y provisioning desplegados
- Despliega infraestructura (nueva), rápida y segura = > entrega rápida
- Ambientes bajo control y estandarizado
- Retroalimentación en todas las actividades

## beneficios para desarrollo

- Tiene ambientes más adecuados para trabajar (dev/prueba/prod)
- Pasa a contar con entornos de desarrollo continuo
- Pasa a contar con pruebas automatizadas
- Implementa aplicaciones (nuevas versiones) más rápidas y seguras = > entrega rápida
- Rápida retroalimentación en todas las etapas de desarrollo



# ¿Cuáles son los beneficios de adoptar la cultura DevOps?

Evaluaremos la perspectiva de cada área que mejora con la adopción de la cultura de DevOps.

## **mutua ganancia para INFRA/DEV**

- Terminan las divisiones de infraestructura vs Desarrollo (fin de la guerra)
- INFRA participa en los proyectos y sigue de cerca todo lo que pasa
- INFRA participando resulta en mejor planificación del entorno de producción
- INFRA participando resulta en un control más efectivo de la aplicación
- DEV comienza a comprender mejor INFRA y esto se traduce en un mejor producto
- Equipos que trabajan juntos para aumentar el valor de la empresa

## **ganancias para la empresa**

- Mejor comunicación entre desarrollo e infraestructura (disminución de los conflictos)
- Soluciones corriendo con mayor estabilidad y rendimiento
- Entregas más rápidas
- Menor tiempo de paradas
- Disminución de incidentes
- Reducción de costes
- Disminución de riesgos
- Aumento de valor para el negocio



# Atando nudos



**Directo al grano (y respondiendo a las preguntas del principio)**

Respondiendo sobre el término DevOps:

1. DevOps está directamente relacionado con un mejor feedback entre las áreas de IT.
2. DevOps es un movimiento, es un concepto, es una cultura y una filosofía, y no existe ninguna explicación fácil.
3. DevOps permite la reducción de riesgos de cambios mediante el uso de una herramienta adecuada y la adopción de una cultura específica.
4. DevOps busca entregar sistemas mejores, con menor costo, de forma más rápida y con menos riesgo.
5. DevOps implica las áreas de infraestructura y desarrollo principalmente.
6. DevOps es pura metodología ágil tanto en infraestructura y desarrollo.
7. DevOps sólo funciona si las áreas de infraestructura y desarrollo están dispuestas a ceder, cambiar su cultura y su método de trabajo.
8. DevOps no es una posición, un sector o departamento, es una cultura.
9. DevOps no se limita solo a startups, se puede utilizar este tipo de cultura en mundo corporativo.
10. DevOps no es algo nuevo, las mejores prácticas estuvieron siempre, podemos decir que este 'mezclado' de prácticas no es nada nuevo para muchas personas, pero para algunos sirve como una buena referencia para implementar los cambios necesarios.

# Atando nudos

## Responder sobre especialista DevOps:

1. Especialista DevOps hoy es típicamente alguien que sabe mucho sobre infra y tiene una base sólida de desarrollo.
2. Especialista DevOps también puede ser alguien que vino del desarrollo y que tiene una base sólida de Infraestructura (esto es generalmente más complejo de encontrar).

## yo quiero ser un DevOps y no sé por dónde empezar

No hay ningún tutorial para esto, mi recomendación es que estudies métodos ágiles y tratar de entender el proceso de desarrollo de donde usted trabaja, aprende herramientas para el desarrollo continuo, estudia las herramientas de gestión de la configuración, provisioning y orquestación, al hacer esto puede dar tus primeros pasos como entusiasta de la cultura DevOps y será capaz de actuar en tu entorno, puedes actuar como un puente entre las áreas de desarrollo e infraestructura y principalmente, podrás modernizar procesos y el entorno de infraestructura.



# Atando nudos

## ¿trabajo con infraestructura ágil, soy un DevOps?

No hay ninguna entidad de certificación reconocida actualmente (Pero si existen varias certificaciones nuevas saliendo todo el tiempo), o alguien que le puede conceder este título, no existe un manual de conducta para decir si eres un DevOps o no, pero en 2016 es muy común encontrar profesionales que vienen del mundo de infraestructura ágil y son considerados DevOps.

Si usted está buscando mejorar su entorno, entregar más rápido, ofrecer algo con más calidad, algo que minimiza los costos, algo que disminuyen los riesgos, algo que implica automatización, si trabajas con infraestructura como código, si usted está actuando como un agente de cambio en su equipo, creo que DevOps es un término conveniente para describir su trabajo, aunque no estés directamente involucrado con un equipo de desarrollo.



# DevOps en la era Cloud

## El cloud lo cambia todo

Pruebas de concepto, testing, QA, staging, despliegue, posproducción, monitorización, analítica... Todo esto puede realizar ya sobre una infraestructura cloud.

El cloud es una solución directa a esta necesidad de rapidez en el testeo y el despliegue. Sin embargo, el entendimiento entre desarrollo y operaciones sigue siendo lo más importante.

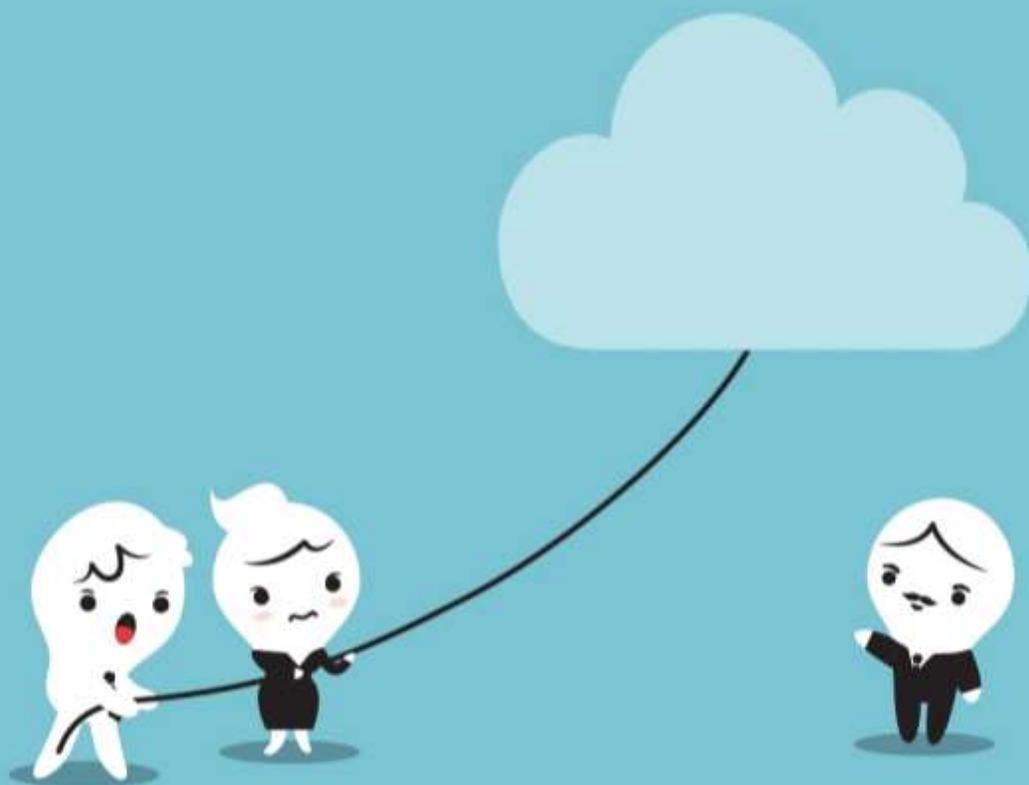


# DevOps en la era Cloud

Un proveedor cloud experimentado puede eliminar esa tensión entre DEV y INFRA ¿Pero cómo?

1. Construyendo infraestructuras bajo demanda, definidas por software y con gestión del rendimiento a nivel de aplicación a tiempo real, para pruebas y para despliegues.
2. Programándola para que se expanda y se reduzca automáticamente, minimizando costes.
3. Ayudándote a optimizar tu código para lograr un rendimiento óptimo de tu aplicación sobre la plataforma.
4. Garantizando la disponibilidad y la seguridad de tu aplicación a un nivel que no puedes lograr solo.
5. Configurando contigo herramientas que te permitan desarrollar y desplegar código de manera fácil y rápida.

Con una buena gestión, la infraestructura adecuada puede conducirte a DevOps con éxito.



# DevOps = + Seguridad

## ¿DevOps promueve la seguridad de la información?

- Como expertos en seguridad, percibimos mucho rechazo hacia el movimiento DevOps. “Sugerimos lo contrario, ya que este método permite una integración de la infraestructura y desarrollo, causando un gran impacto en el entorno de seguridad”
- DevOps promueve el análisis desde el principio del proceso de desarrollo
- Con DevOps, el equipo de desarrollo aprecia los factores relacionados con la fiabilidad, seguridad, rendimiento y análisis desde las primeras etapas.
- DevOps permite un monitoreo más preciso de todo lo que implica los esfuerzos de desarrollo y equipos de test.
- DevOps requiere pruebas en todas las etapas de desarrollo
- En lugar de comenzar a hacer pruebas sólo cuando el producto está terminado, con DevOps, los equipos hacen todo tipo de pruebas de la aplicación etapa por etapa. Esto hace que sea más fácil de identificar las fallas, los desajustes, las solicitudes incompletas. Podemos decir que el software se corrige de forma instantánea, permitiendo que las siguientes etapas tengan un mayor grado de confianza.



# DevOps = + Seguridad

- DevOps promueve la sincronización de los equipos que permite múltiples mecanismos de autenticación y validación.
- Podemos resumir que DevOps trae más seguridad al código (pruebas en todo el desarrollo), las correcciones en el camino (en el proceso tradicional, las pruebas se realizan al final), de esta forma colocamos los profesionales de seguridad en el medio del proceso de desarrollo, que también mejora la detección de vulnerabilidades y correcciones, y promueve la cultura de prevención.
- “Esto permite un entendimiento más profundo de lo que la aplicación hace y dónde puede ser más débil ante amenazas de seguridad. Con la proliferación de las brechas de seguridad (como Heartbleed o Shellschok) este entendimiento del desarrollo de aplicaciones permite tomar decisiones más informados”.
- Por lo tanto, DevOps no sólo es una cultura de desarrollo ágil y la integración de equipos de desarrolladores con la operación de TI, pero también es un salto en el campo de la seguridad de la información. Es un método que hace que la seguridad deje de ser un punto aislado, empezando a ser parte de todo el desarrollo de la aplicación.



# DevOps + SecOps = DEVSECOPS

Hoy en día el departamento de seguridad información debe trabajar de la mano con otros departamentos, es decir el equipo de seguridad ahora está invirtiendo más tiempo en ayudar a la próxima iniciativa de negocio en lugar de apagar incendios todo el tiempo.

La seguridad siempre es mejor cuando no se “hermetiza” y adoptar este enfoque normalmente significa que el ruido de los falsos positivos y la falta de contexto se reducen considerablemente. El resultado es un equipo de seguridad que ya no está buscando entre los archivos de registro las 24 horas, los 7 días de la semana o realizando otras tareas de seguridad.

Como sabemos La seguridad de la información tiene tres objetivos principales: **confidencialidad, integridad y disponibilidad.**

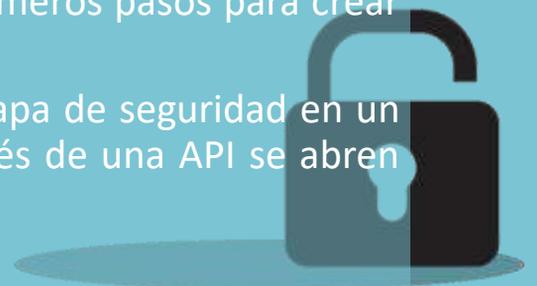
La manera sencilla de relacionar estos objetivos es que la seguridad trabaja para garantizar que sólo la gente que usted quiere reciba (confidencialmente) los datos correctos (integridad) cuando los necesitan (disponibilidad).

Y aunque invertimos mucho tiempo preocupándonos y hablando sobre la confidencialidad y la integridad, a menudo ignoramos la disponibilidad y dejamos a otros equipos satisfacer este requerimiento.

Ahora con la funcionalidad disponible en la nube podemos utilizar realmente los aspectos de disponibilidad para elevar nuestra seguridad.

Aprovechando características como Amazon SNS, AWS Lambda y Auto Scaling, podemos crear escenarios de reacción automáticos. Esta “respuesta continua” es uno de los primeros pasos para crear cargas de trabajo de auto-recuperación.

Cuando usted comienza a automatizar la capa de seguridad en un entorno en el que todo es accesible a través de una API se abren algunas posibilidades interesantes.



**“El cambio a la cultura DevSecOps permite a sus equipos concentrarse en las tareas que mejor hacen”**

# DevOps + SecOps = DEVSECOPS

Nueva filosofía generada por la fusión de DevOps y SecOps.

Integra el enfoque de la seguridad en el ciclo de desarrollo y explotación de aplicaciones de una manera sistemática. Al igual que las DevOps, operaciones de soporte al desarrollo, la seguridad se debe poder automatizar o sistematizar en gran medida.

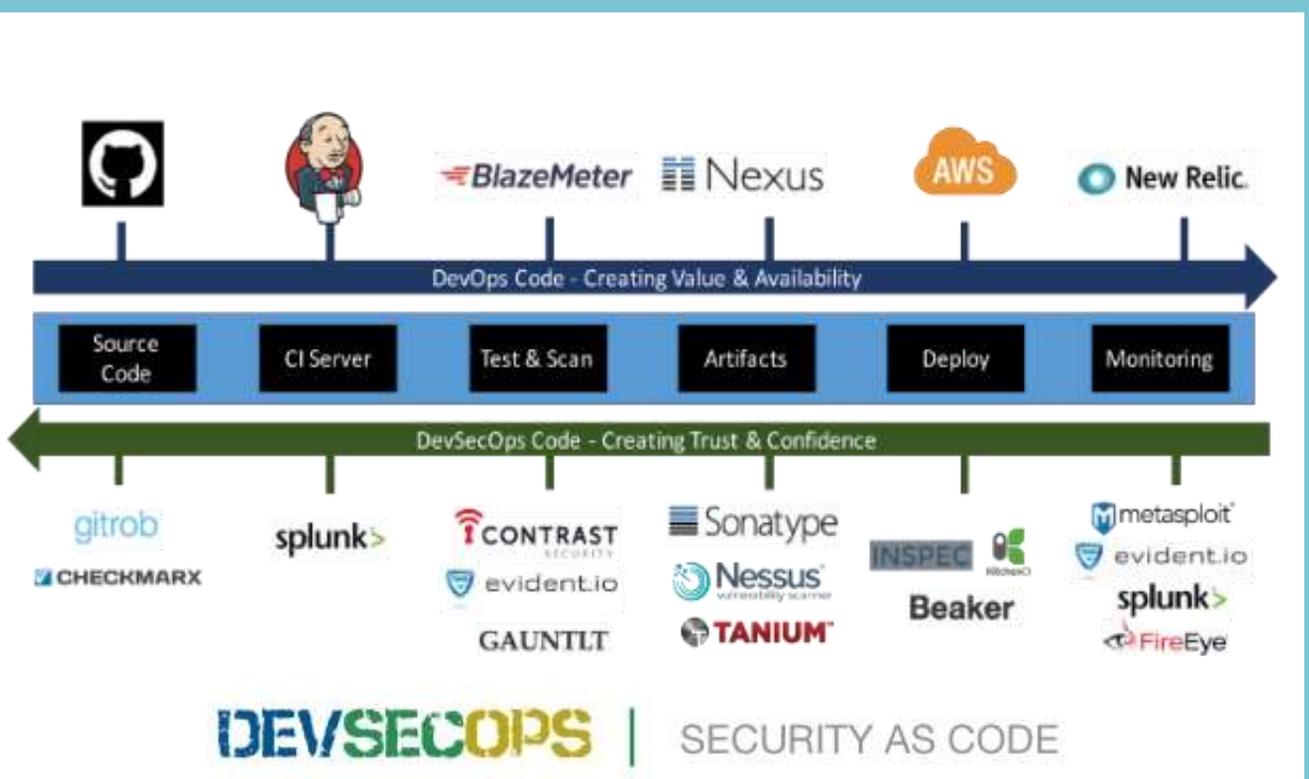
El objetivo final es poder pasar a producción de manera automática una aplicación razonablemente segura en cuestión de minutos. Considerando siempre que la seguridad total no existe, lo que se minimiza es el riesgo.

[DevSecOps Manifiesto \(2016\)](#)

[DevSecOps Presentations](#)

No son ideas nuevas, simplemente es una reformulación de una necesidad.

Se compone de varias ideas, una de ellas es la consideración de la seguridad integrada en el proceso de desarrollo.



# DevOps + SecOps = DEVSECOPS

## Situación habitual en el desarrollo de aplicaciones

Vemos que el enfoque actual es que la seguridad se contempla fuera del ciclo de desarrollo de una aplicación. Inicialmente con pruebas que dependerán de la experiencia del equipo de desarrollo. Posteriormente mediante tests de penetración o hacking ético, tratando de detectar vulnerabilidades de manera interna. Para ello se utilizan herramientas como [OWASP Zed](#) (orientada a entorno web) incluidas en distribuciones como [Kali Linux](#).

## Seguridad en el código

Otro nuevo enfoque, bajo la filosofía DevSecOps, es recopilar información dentro de la aplicación para saber si se está bajo un ataque. Para ello es muy útil el enfoque del siguiente proyecto [OWASP AppSensor](#). No solo se trata de enterarse de los ataques por recopilación y correlación de logs sino de contemplarlos y actuar directamente desde la aplicación siguiendo unas sencillas pautas en el desarrollo:



**Controlar el número de intentos de acceso fallidos de un usuario**

**Reducir el número de servicios expuestos.**

**Reducir el número de protocolos y operaciones admitidas. Si realiza llamadas extrañas incorporarlo a un log.**

**Comprobar periódicamente datos relacionados con la identidad del usuario, por ejemplo si ha cambiado su IP.**

# DevOps + SecOps = **DEVSECOPS**

Un ejemplo de aplicación es llevar un contador de operaciones sospechosas, cuando se realizan 3 en un minuto se bloquea al usuario y se lanza una alerta. Se puede correlacionar el usuario con eventos en otras aplicaciones.

[OWASP AppSensor Intro for Developers](#)

[OWASP AppSensor Guide v2](#)

[AppSensor Detection Points](#) (Listado de eventos a contemplar y Logs)

## **Siempre dentro de una organización segura**

**DevSecOps** implica a toda la organización, difícilmente se podrán implantar de manera aislada, es necesario que la organización contemple la seguridad. No tiene sentido desarrollar aplicaciones seguras si todo el entorno en el que se ejecutan no lo es, es un desperdicio de esfuerzos para obtener un sistema de información que, en su conjunto, tiene vulnerabilidades. Una organización razonablemente segura necesitará:

**Securizar y monitorizar el tráfico de red.**

**Securizar servidores y servicios, con especial interés en los principales servicios que proporcionan datos a las aplicaciones, bases de datos y sistemas de ficheros, junto con los servidores de aplicaciones.**

**Almacenamiento y correlación de logs.**

**Monitorización continua y sistemas de alerta temprana.**



Pero por supuesto formación específica en seguridad a todo el personal junto con perfiles especializados en seguridad.

En resumen, son ideas que las empresas llevan tiempo aplicando, simplemente que ahora existen unas siglas que las representan. No parece que sea una moda pasajera, ya existen conferencias al respecto:

[DevSecCon.](#)

# Links recomendados

## **Automatización (debemos automatizar para tener control, visión y productividad)**

### Teorías

[https://en.wikipedia.org/wiki/Infrastructure\\_as\\_Code](https://en.wikipedia.org/wiki/Infrastructure_as_Code)

[https://en.wikipedia.org/wiki/Software\\_configuration\\_management](https://en.wikipedia.org/wiki/Software_configuration_management)

[https://en.wikipedia.org/wiki/Version\\_control](https://en.wikipedia.org/wiki/Version_control)

[https://en.wikipedia.org/wiki/Cloud\\_computing](https://en.wikipedia.org/wiki/Cloud_computing)

<https://en.wikipedia.org/wiki/Virtualization>

### Mark Burgess

<http://markburgess.org>

[https://en.wikipedia.org/wiki/Mark\\_Burgess\\_\(computer\\_scientist\)](https://en.wikipedia.org/wiki/Mark_Burgess_(computer_scientist))

[http://www.iu.hio.no/~mark/papers/cfengine\\_history.pdf](http://www.iu.hio.no/~mark/papers/cfengine_history.pdf)

### GCONF Apps

<https://en.wikipedia.org/wiki/CFEngine>

[https://en.wikipedia.org/wiki/Puppet\\_\(software\)](https://en.wikipedia.org/wiki/Puppet_(software))

[https://en.wikipedia.org/wiki/Chef\\_\(software\)](https://en.wikipedia.org/wiki/Chef_(software))

[https://en.wikipedia.org/wiki/Salt\\_\(software\)](https://en.wikipedia.org/wiki/Salt_(software))

### Orquestadores Apps

<https://get.fabric.io>

<http://capistranorb.com>

[https://en.wikipedia.org/wiki/Ansible\\_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))

<https://docs.puppet.com/mcollective/>

### Aprovisionamiento Apps

<http://vagrantup.com>

<https://www.packer.io>

<http://www.ubuntu.com/cloud/juju>

<https://theforeman.org>

<https://www.terraform.io>

### Distribución con foco en containers

<http://www.projectatomic.io>

<https://coreos.com>

### Cluster de containers

<http://mesos.apache.org>

<http://aurora.apache.org>

<https://www.nomadproject.io>

<http://kubernetes.io>

### Herramientas de nube (autoservicio)

<https://en.wikipedia.org/wiki/Openstack>

[https://en.wikipedia.org/wiki/Apache\\_CloudStack](https://en.wikipedia.org/wiki/Apache_CloudStack)

<https://en.wikipedia.org/wiki/OpenNebula>



# Links recomendados

## **Entrega (Foco en una entrega con calidad, automatizada y rápida)**

### Teorías

[https://en.wikipedia.org/wiki/Continuous\\_delivery](https://en.wikipedia.org/wiki/Continuous_delivery)

[https://en.wikipedia.org/wiki/Continuous\\_integration](https://en.wikipedia.org/wiki/Continuous_integration)

[https://en.wikipedia.org/wiki/Test-driven\\_development](https://en.wikipedia.org/wiki/Test-driven_development)

[https://en.wikipedia.org/wiki/Acceptance\\_test-driven\\_development](https://en.wikipedia.org/wiki/Acceptance_test-driven_development)

[https://en.wikipedia.org/wiki/Behavior-driven\\_development](https://en.wikipedia.org/wiki/Behavior-driven_development)

[https://en.wikipedia.org/wiki/Load\\_testing](https://en.wikipedia.org/wiki/Load_testing)

[https://en.wikipedia.org/wiki/Stress\\_testing](https://en.wikipedia.org/wiki/Stress_testing)

[https://en.wikipedia.org/wiki/Security\\_testing](https://en.wikipedia.org/wiki/Security_testing)

### SW Repo Apps

<https://archiva.apache.org>

<http://www.sonatype.org/nexus/>

<https://www.jfrog.com/artifactory/>

### VCS-SCM Apps

<https://en.wikipedia.org/wiki/Git>

<https://gitlab.com>

<https://github.com>

<https://bitbucket.com>

### Build Apps

<http://buildbot.net>

<https://jenkins.io>

### Testeo de Integración

<http://kitchen.ci>

<https://about.gitlab.com/gitlab-ci/>

<https://travis-ci.org>

<https://www.go.cd>

### Web Tests

<https://saucelabs.com>

<http://www.seleniumhq.org>

<https://www.browserstack.com>

<http://casperjs.org>

<http://phantomjs.org>

<https://slimerjs.org>

### Testeo de carga

<http://jmeter.apache.org>

<https://www.blazemeter.com>

### Puppet+Git (Deploy simples)

<https://github.com/puppetlabs/puppetlabs-vcsrepo>



# Links recomendados

## **Métricas (No existe gestión o administración de sistemas sin métricas)**

### Logs Apps

<https://www.elastic.co/products/logstash>

<http://www.fluentd.org>

<https://flume.apache.org>

<https://splunk.com>

### Status Apps

<https://statuspage.io>

<https://status.io>

<https://cachethq.io>

### Sites Metric Crash Report

<https://getsentry.com>

<https://bugsnag.com>

<https://raygun.com>

### Incident Metrics

<https://www.pagerduty.com>

<https://victorops.com>

<https://bigpanda.io>

### Sites Metric APM

<https://newrelic.com>

<https://github.com/naver/pinpoint>

<https://glowroot.org/>

### Monitoreo amplío

<https://www.datadoghq.com>

<http://www.logicmonitor.com>

<https://www.zabbix.com>

<https://www.nagios.org>

### Graphite Stack

<https://collectd.org>

<https://github.com/etsy/statsd>

<http://graphiteapp.org>

<http://grafana.org>

### ELK Stack

<https://www.elastic.co/products/elasticsearch>

<https://www.elastic.co/products/logstash>

<https://www.elastic.co/products/kibana>



# Links recomendados

**Personas (Las personas deben entender, ser parte del equipo y integrarse al mismo)**

<https://en.wikipedia.org/wiki/Kanban>

[https://en.wikipedia.org/wiki/Coding\\_Dojo](https://en.wikipedia.org/wiki/Coding_Dojo)

[https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))

<https://en.wikipedia.org/wiki/Hackerspace>

<https://en.wikipedia.org/wiki/Hackathon>

<https://en.wikipedia.org/wiki/Gamification>

**Formación Ops (Un Ingeniero OPS necesita un programa integral )**

<http://www.opsschool.org>

<https://www.codecademy.com>

<https://www.codeschool.com>



# Referencias

## Sites:

<http://www.kartar.net/2010/02/what-devops-means-to-me/>

<http://itrevolution.com/the-convergence-of-devops/>

<http://devopsdays.org/events/>

<http://devopsweekly.com>

<http://www.devsecops.org/>

<https://www.owasp.org>

## Presentaciones:

<http://www.devsecops.org/presentations/>

<http://www.slideshare.net/jallspaw/10-deploys-per-day-dev-and-ops-cooperation-at-flickr>

<http://www.slideshare.net/KrisBuytaert/devops-the-future-is-here-its-just-not-evenly-distributed-yet>

<http://www.slideshare.net/jedi4ever/devops-is-a-verb-its-all-about-feedback-13174519>

<http://www.slideshare.net/jedi4ever/devops-the-war-is-over-if-you-want-it>

<http://www.slideshare.net/jedi4ever/devops-tools-fools-and-other-smart-things>

<http://lanyrd.com/2012/berlindevops-july/swzgt/#link-kryx>

<http://www.slideshare.net/wickett/the-devops-way-of-delivering-results-in-the-enterprise>

## Imágenes:

<http://www.freepik.com/kraphix> "Diseñada por kraphix"



# Conclusiones

**no existe ninguna receta mágica o vía rápida**

Como hemos dicho no es ninguna metodología clara, DevOps es todavía un movimiento en constante construcción y definición, donde cada día se suman mas y mas practicas, comentamos una serie de mejoras que forman parte de la cultura de DevOps, depende de cada organización o cada profesional estudiar y averiguar la mejor manera de combinar estas pequeñas recetas técnicas y aplicar en su entorno.

Esperamos que este **E-book** haya ayudado a comprender mejor la cultura de DevOps y DevSecOps.



<https://www.linkedin.com/groups/12033278>



[https://twitter.com/devsecops\\_ar](https://twitter.com/devsecops_ar)

