

# Infraestructuras Agiles

El corazón de DevOps en IT OPS



# ¿Que veremos?

- Autores**
- Concepto**
- Ejes**
- Resultados**
- Transición**
- Dinámica de trabajo**
- Kanban para IT OPS**
- Links recomendados**
- Referencias**
- Conclusiones**



# 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 36 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 16 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 y infraestructura. 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. Instructor de diversas charlas y cursos.

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 científico en Ciberseguridad del evento IEEE ARGENCON 2016



CERTIFIED  
INTEGRATOR

# 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

## 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 26 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 híbridas, 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.

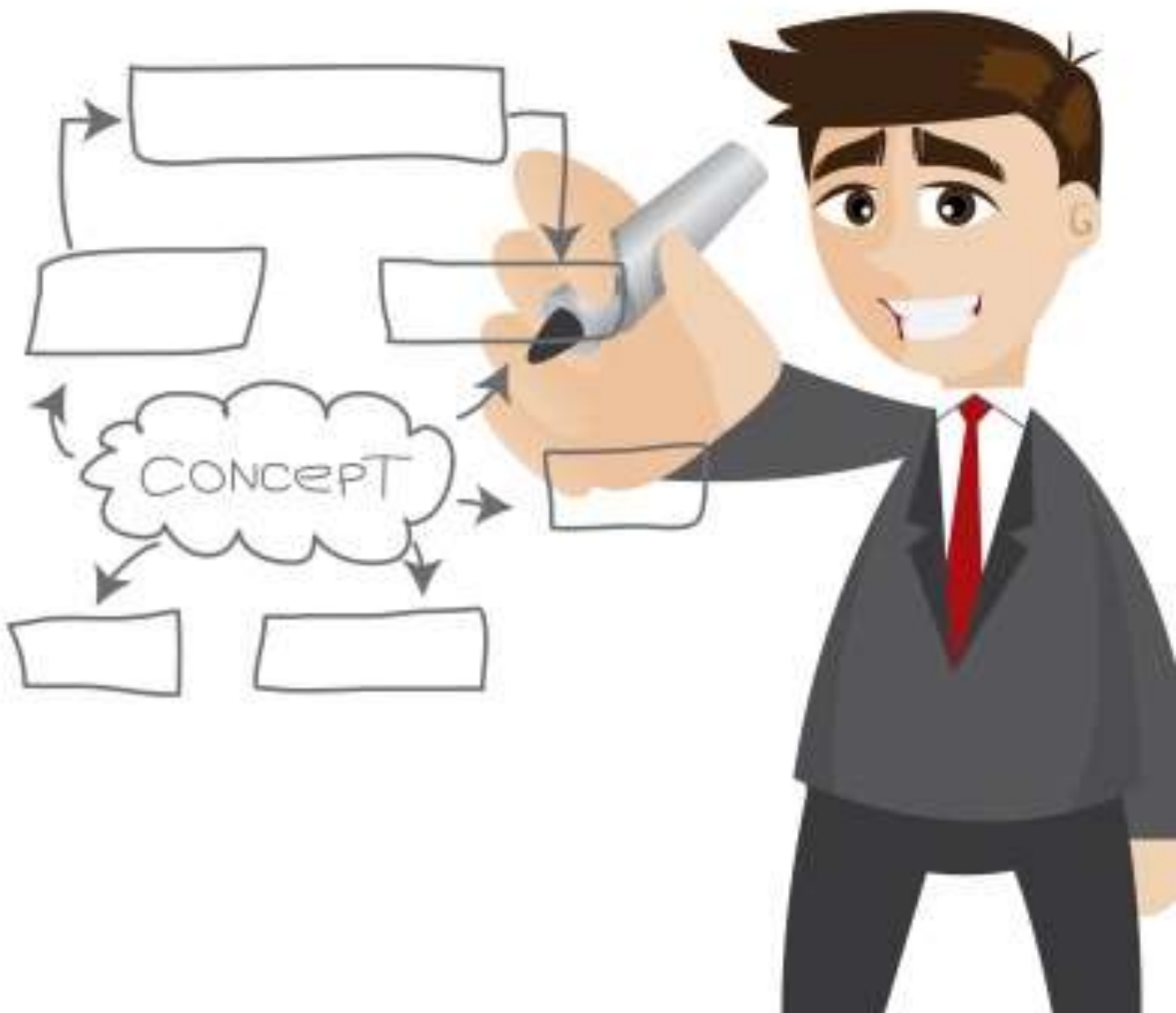
Entusiasta de DevOps, y infraestructuras agiles o infra como código.



Private Cloud	Server Infrastructure
Productivity	SharePoint
Cloud Platform and Infrastructure	Communication



# Concepto



# Concepto

Este e-book es la segunda entrega de DevSecOps Argentina con la colaboración de [Cloud Legion](#)

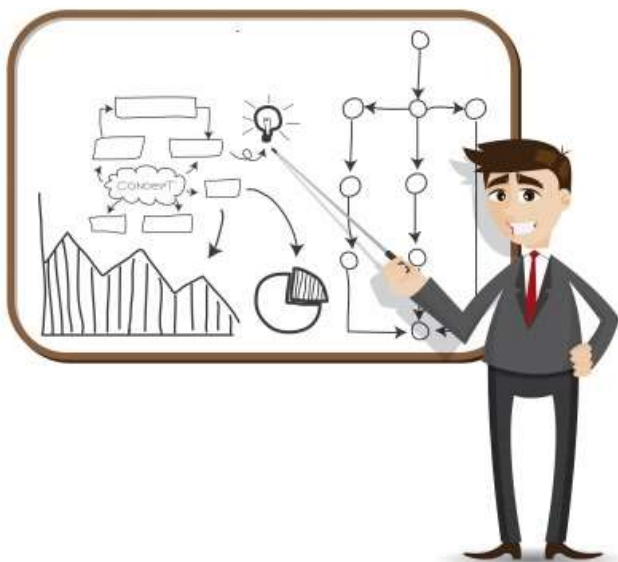


El termino **INFRAESTRUCTURA ÁGIL** viene tomando peso dentro de las comunidades técnicas y se nota cada vez mas presente en varias conferencia de IT en Argentina y en el resto de mundo.

Pero que es **Infraestructura Ágil** y cual es su relación con la cultura **DevOps**?

Tanto **DevOps** como **Infraestructura Ágil** tienen como valores fundamentales el CAMS (Culture, Automation, Measurement and Sharing) y el ICE (Inclusion, Complex Systems and Empathy)

Se puede decir que **Infraestructura Ágil** y **DevOps** son movimientos con principios comunes pero con focos diferenciados.





# Concepto

Podemos decir que hoy por hoy una infraestructura de TI tradicional es generalmente bien heterogénea y contiene diversos roles bien segmentados (Sysadmins, Analistas de red, DBA, etc).

Normalmente encontramos equipos de administración de sistemas separados del equipo de base de datos, backup, red, virtualización y almacenamiento, hay también escenarios en los que tenemos un equipo monolítico que interactúa con diversos proveedores, a veces un proveedor para cada role del equipo.

En escenarios como este, implementar **DevOps** puede ser un desafío, porque hay una gran dificultad de comunicación entre los equipos de infraestructura y prácticamente no hay cualquier comunicación con equipos de desarrollo y calidad.



# Concepto

**Infraestructura Ágil** es una guía de adopción de buenas prácticas creada y organizada por el colectivo de sysadmins a partir de experiencias que implican la automatización y el uso de métodos ágiles.

La idea principal es que este modelo ayude a los equipos de operación a realizar una transición segura dentro de su infraestructura para un modelo más eficiente y autónomo.

Los principales fundamentos de la **Infraestructura Ágil** son la automatización, las métricas y el uso de métodos específicos para el trabajo en equipo

En este ebook, usted encontrará un modelo organizado de forma coherente y sencilla para facilitar la adopción e implementación en su empresa.



# Infraestructura Ágil

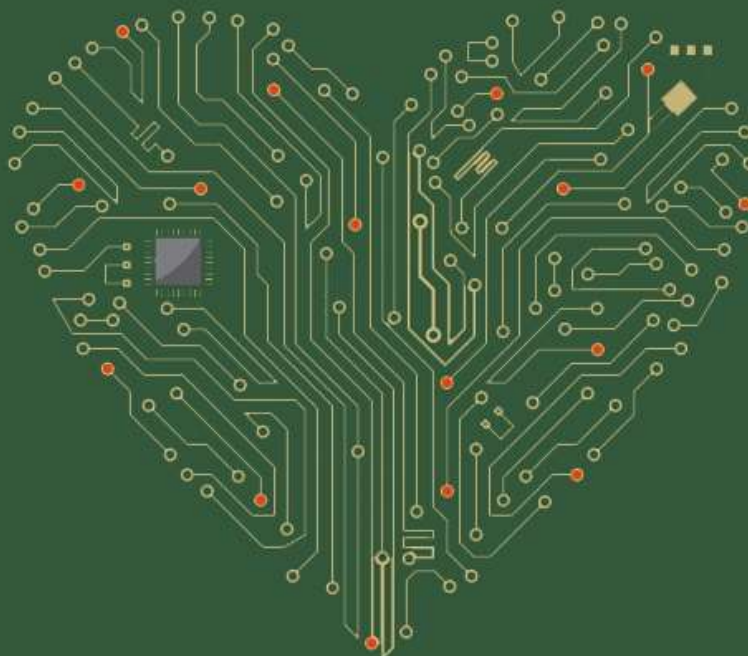
## El corazón de DevOps en IT.

**DevOps** es esencialmente una cultura, que propone un proceso de transformación para alcanzar **toda** organización, haciendo esto a través de la integración, la colaboración y la retroalimentación entre los equipos.

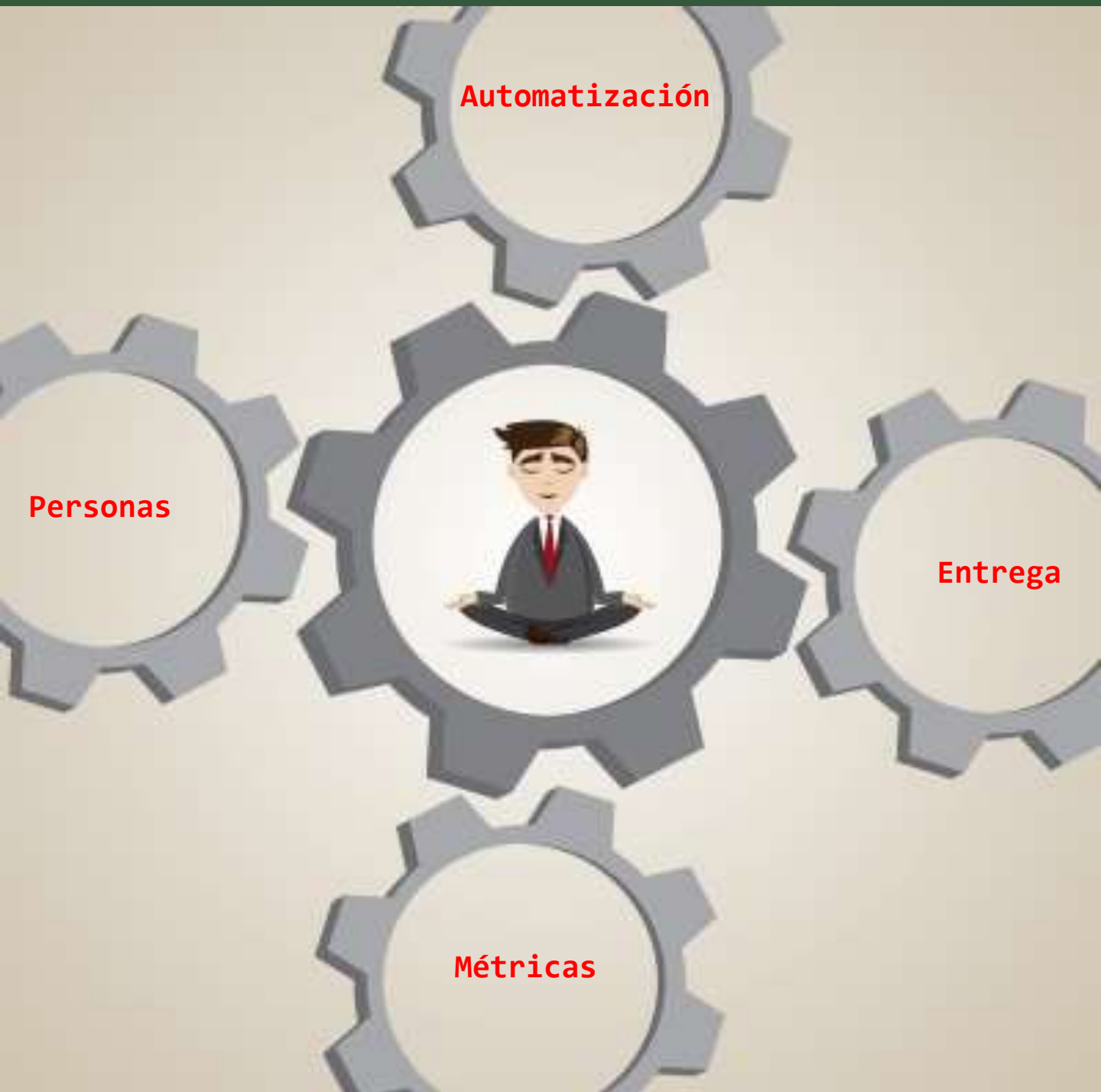
**Infraestructura Ágil** propone los mismos valores enfocándose en la infraestructura / operación de TI de una organización.

Mejorar la productividad de infraestructura de una organización es el foco de **Infraestructura Ágil** a través de la Automación, Entrega, Métricas y Personas. Para ello, se utilizan las diversas prácticas de Agile (Kanban, Infrastructure as Code, etc.) sin menospreciar cosas relacionadas a SLA, tickets, etc

A partir de **Infraestructura Ágil** el equipo ganara madurez y empezara a caminar en dirección a la cultura **DevOps**



# Ejes de una Infraestructura Ágil



Entendamos los principios que contemplan **Infraestructura Ágil**

# Automatización

No existen más sistemas simples, todos los sistemas son complejos e interdependientes.

Hoy en día mantener un simples blog significa mantener servicios como servidor de aplicación, base de datos, caché, autenticación, estadísticas, dns, firewall, copia de seguridad entre otros componentes.

La administración de todo esto manualmente se vuelve inviable porque un cliente puede rodar cientos o miles de sitios, servicios y sistemas con características similares a éstas.

Dentro de este contexto, podemos decir que la automatización es uno de los principales pilares de **Infraestructura Ágil** pues no hay más escenarios que permiten administrar sistemas y servicios en el modelo artesano.

La automatización se vuelve obligatoria principalmente en entornos que utilizan virtualización, nube o contenedores.

No podemos llevar días, semanas o meses para atender las demandas de nuestros clientes, tenemos que conseguir atender estas demandas en minutos a través de la automatización.



# Automatización

Gestión de la  
configuración

Aprovisionamiento

Orquestación



La etapa de automatización es compuesta por las siguientes actividades



# Gestión de la configuration

Actualmente, cuando hablamos de automatización a gran escala, el principal modelo que encontramos en estos proyectos es la “gestión de las configuraciones”

(**Configuration Management \*\* CM \*\***).

El concepto de CM para sistemas computacionales fue creado por el profesor **Mark Burgess** durante el período en que hacía un post-doctorado en la universidad de Oslo en Noruega en 1993.

En esa época él ayudaba en la administración del laboratorio UNIX del departamento de física teórica. Durante ese período se enfrentó a algunos problemas con el ambiente del laboratorio, para resolver estos problemas escribió la primera versión de la herramienta **CFEngine**.

El **CFEngine** era un proyecto que reunía conceptos de Configurations Management encontrados en la ingeniería de software con los principios de modelado científico de la ciencia de computación, aplicando esto para la gestión de sistemas computacionales.

Este enfoque permitió la creación de la noción de operaciones convergentes.



# Gestión de la configuration

Los sistemas de automatización con operaciones convergentes logran converger a un estado final definido por el usuario, con lo que si la herramienta encontraba alguna característica que divergía de lo que el usuario definió, el sistema se convertía al estado definido.

Con el **CFEngine** era posible definir estados para diferentes sistemas operativos, ocultando sus diferencias a través de un DSL (lenguaje de dominio específico). Este lenguaje declarativo permitía que el usuario trabajara con un alto nivel de abstracción que tenía foco en el estado final que el sistema debería tener.

Entre 1993 y 2003 el **CFEngine** y el modelo CM evolucionaron y adquirieron otras características como **self-healing**, detección de anomalías y comunicaciones seguras.

En 2004 **Mark Burgess** organizó todo ese aprendizaje con el **CFEngine** y CM en la teoría Promise que es un modelo distribuido de automatización **self-healing**.

Después de 2004, muchas herramientas surgieron influenciadas por esta teoría, entre ellas podemos destacar el **Puppet y Chef** que heredan muchas características del **CFEngine**.



# Gestión de la configuration

Las herramientas de gestión de configuraciones se utilizan para mantener la estandarización y el cumplimiento en sistemas y servicios a partir del control automatizado de estados del sistema.

## **Código**

Su herramienta de gestión de configuración debe ofrecer el concepto de documentación ejecutable

## **La vida**

Acompañe todo el ciclo de vida de cada servidor de su parque

## **Cambio**

Tenga las condiciones de seguir y entender los cambios ocurridos en cada servidor



# Gestión de la configuration

## Estados

La gerencia de estados es la principal característica de herramientas que gestionan configuraciones: con ellas, es posible definir los estados deseados para el sistema. Ellas mantienen la configuración según lo declarado por el sysadmin.

## Convergencia

Si la herramienta encuentra una divergencia entre el estado declarado y el presente en el sistema operativo, realizará la convergencia al estado correcto. Es decir, corrige la configuración de su sistema de manera automatizada.

## Informe

Cuando alguna inconsistencia es detectada por la herramienta, se realiza el drift y el sistema converge al estado declarado anteriormente. Todos los eventos y cambios ocurridos en este proceso se registran en un sólido sistema de informes.



# Gestión de la configuration

Con la gestión de configuraciones la organización entra en una parte del mundo de la infraestructura como código y pasa a utilizar también herramientas de control de versiones para gestionar el código que representa su infraestructura.

Su infraestructura pasa a ser desarrollada como si fuera un software, eso posibilita el uso de diversos métodos y herramientas para pruebas de código. Todo este control ayudará a garantizar la calidad del código que se utilizará en la organización.

En **Infraestructura Ágil** el uso de gestión de configuraciones es obligatorio para tener un completo proceso de automatización.



# Gestión de la configuration

Para un proyecto de **Infraestructura Ágil**, se recomienda utilizar herramientas CM con las siguientes características:

- - Arquitectura Master / Agent;
- - Secure Communications;
- - Capa de abstracción;
- - Sistema de convergencia;
- - Sistema de control de estados;
- - Self-healing;
- - Enforcing;
- - Idempotencia;
- - Lenguaje declarativo;
- - Sistema de informes;
- - Sistema de procesamiento de informes;
- - Visualización de eventos y cambios;
- - Report Processors;
- - APIs.



Estas son algunas herramientas que ofrecen estos recursos.





# Orquestación

Si vamos a disfrutar de una orquesta podemos observar que cada músico tiene consigo el "código" que necesita ejecutar de forma autónoma. En este ambiente el maestro actúa como un "sistema" de apoyo para armonizar y facilitar la integración, colaboración y el trabajo del grupo.

La orquestación en esencia está mucho más cerca de las herramientas de CM, pero en el mundo de IT, este es un término con diferentes concepciones y entendimientos. A pesar del origen del término venir de una orquesta, en IT el término ganó otro punto de vista.

En el entendimiento común la orquestación es un mecanismo central que dispara acciones en paralelo para sistemas operativos de un centro de datos.

Para nosotros, dentro del modelo **Infraestructura Ágil**, vamos a considerar que la orquestación consiste en ejecutar algo de forma paralela o no, en tiempo real, en un sistema operativo o en un grupo de sistemas operativos.



# Orquestación

A diferencia de los sistemas de gestión de configuración, los orquestadores se preocupan en girar lo que usted determinó en los nodos de su parque, pero no hay una preocupación con la garantía de estado o con la garantía de ejecución, idempotencia, registro riguroso de eventos o self-healing.

El enfoque de las herramientas es más objetivo y simple, lo que por un lado es bueno pues ofrece una pequeña curva de aprendizaje, sin embargo, por otro lado, la visibilidad de los cambios es muy baja, dejándote sin estar seguro si el comando realmente funcionó en la punta, después de la ejecución del mando, la máquina pasa a tener un estado indefinido.

Los orquestadores son muy utilizados para la administración de sistemas, agregan valor complementando y apoyando las herramientas de CM.

Por lo general se utilizan para hacer deploy de aplicaciones, mantenimientos puntuales y de emergencia en datacenters. También se utilizan para hacer mantenimiento y rollout de agentes CM como Puppet y Chef.



# Orquestación

Dentro de la **Infraestructura Ágil** hemos segmentado orquestadores en dos tipos, aquellos que hacen PUSH y los que usan MQ.

## **\*\* Orquestación en arquitectura PUSH \*\***

Las herramientas de orquestación en arquitectura PUSH normalmente utilizan el protocolo SSH para conectarse a los nodos y realizar acciones, entre ellas podemos citar Ansible, Capistrano y Fabric.

Ellas dependen íntegramente de un protocolo externo, no usan agentes locales y en su diseño la gerencia de estados, convergencia, abstracción, visibilidad plena e idempotencia no son parte nativa de sus premisas.



# Orquestación

## **\*\* Orquestación en Arquitectura MQ \*\***

Las herramientas de orquestación en arquitectura MQ utilizan sistemas de mensajería, en este modelo la herramienta consume una cola o tópico que puede tener comandos inyectados por el sysadmin.

Una vez que los comandos se inyecten en la cola, el agente de orquestación recibe, valida y se ejecuta en el nodo. Marionette Mcollective es un ejemplo de herramienta de este tipo, utiliza ActiveMQ como MQ, otro ejemplo conocido es Salt, que tiene un subsistema de orquestación que utiliza el proyecto ZeroMQ para orquestar sus Minions.

Estos orquestadores dependen del sistema MQ, son extremadamente sensibles en cuanto a la sincronización de fecha y hora entre nodos y servicio MQ, no usan agentes, y como en los orquestadores PUSH su diseño no acoge la gestión de estados, convergencia, abstracción, visibilidad plena e idempotencia.



# Orquestación

En **Infraestructura Ágil** los orquestadores son herramientas importantes para apoyar su sistema de gestión de configuraciones y para agregar valor al proceso de automatización.

Algunas de la herramientas que ayudan a implementar los conceptos de Orquestación son:



# Provisionamiento

Para la **Infraestructura Ágil**, provisionar significa tener una capa de inteligencia entre su necesidad y los recursos disponibles.

La creación de nuevos entornos debe ser un proceso rápido y fácil.

También debe tomar como máximo algunos minutos para ser realizado y el usuario / consumidor debe tener acceso al recurso de manera directa (auto-servicio).

En la mayoría de las organizaciones el aprovisionamiento es un punto muy sensible y polémico.





# Provisionamiento

Hay organizaciones que les lleva hasta 90 entregar una simple VM para que un equipo pueda hacer pruebas.

Esto ocurre debido a procesos lentos, burocráticos, ineficientes y por el uso de método artesano para la creación y configuración de estos ambientes al final del proceso.

En **Infraestructura Ágil** la creación de nuevos ambientes debe ser un proceso simple y debe llevar minutos en lugar de horas, días, semanas o meses.

Para ello se combinan algunos factores:

- Autoservicio
- Cuotas
- Automatización

El uso de herramientas de autoservicio permite que el cliente pueda crear su ambiente directamente.



# Provisionamiento

Cuando hablamos de un equipo de IT, crear VMs es algo natural, los profesionales de IT están familiarizados con la creación de máquinas virtuales o instancias en la nube para pruebas o estudios, por lo que los procesos de creación de VM en el hipervisor de la empresa, la creación de instancias en la nube interna o la creación de contenedores deben ser procesos naturales y simples.

No tiene sentido burocratizar una demanda cotidiana e insertar personas entre la necesidad de un equipo o de un profesional y el recurso de IT a ser utilizado.

Hay varias herramientas de autoservicio que se pueden utilizar y cada equipo puede tener sus cuotas de uso del hipervisor, nube, contenedor y almacenamiento. El control a través de las cuotas garantiza el uso simplificado y al mismo tiempo protege, controla y condiciona a los equipos a usar racionalmente los recursos de TI de su organización.



# Provisionamiento

Para llegar a este resultado, las herramientas de automatización se integran al sistema de autoservicios para crear máquinas virtuales, instancias y contenedores de forma automatizada. Esta acción convierte todos los requisitos en un solo proceso que hará el arranque del sistema operativo, configuraciones estructurantes de red, reglas de firewall, configuración del sistema operativo, instalación y configuración de la aplicación, entre otras características.

En **Infraestructura Ágil**, el aprovisionamiento de ambientes debe ser accesible a todos del equipo de IT de forma inteligente, rápida y segura.



# Entrega



# Entrega

Podemos resumir que su principal foco es en una entrega con calidad, automatizada y rápida.

- Control de versiones
- Repositorio de artefactos
- Estera
- Pruebas
- Rollback

Es importante entender lo que no es entrega:

Recibir archivos adjuntos vía e-mail para publicar en N nodes;

Editar archivos en el servidor directamente;

Publicar algo que no ha pasado por pruebas mínimas;

Publicar algo que no tenga plan de rollback.



# Entrega

En la **Infraestructura Ágil**, todo el código que estará disponible para el usuario debe estar versionado y almacenado en un lugar adecuado.

Esto incluye tanto el código que representa la infraestructura, como el código de las aplicaciones, que deben estar preferentemente en el mismo sistema de control de versiones.

Así, todos los archivos están disponibles y accesibles para todos los miembros del equipo de TI.





# Entrega

Para que se publique algo, antes debe ocurrir un riguroso proceso de pruebas, después de estas pruebas - si la aplicación se aprueba en todas las pruebas, el artefacto debe almacenarse en un lugar adecuado para permitir la automatización del paso final de publicación y del rollback a una versión anterior si es necesario - también disponible en este mismo repositorio.

La entrega de software en **Infraestructura Ágil** debe ser automatizada a través de software especializados, es esencial para el proceso de desarrollo de productos, ya que van a interactuar con sistemas de control de versiones y el repositorios de artefactos.

Es recomendable la construcción de una cinta de publicación con pruebas integradas (Contemplando también las pruebas de seguridad) para eso les recomiendo ver el e-book **“Introduccion a Devops y DevSecOps”** y buscar mas información en [DevSecOps.org](https://DevSecOps.org)



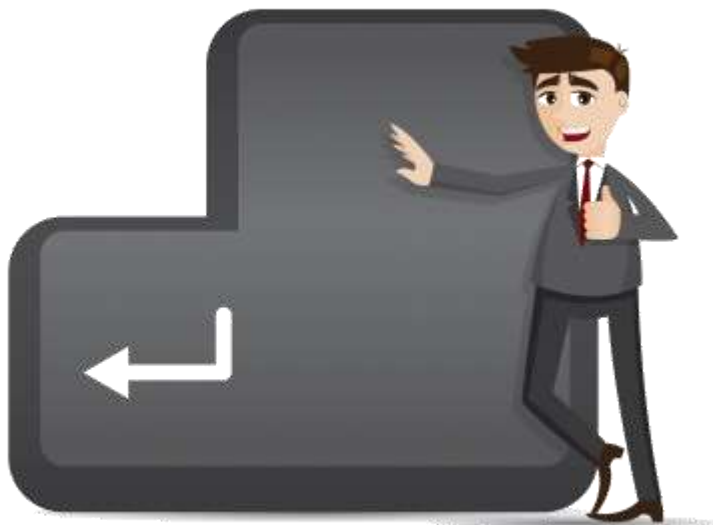
# Entrega

El equipo de sysadmins debe entender cuáles son los requisitos para que una aplicación entre en producción y proporcionar la automatización de proceso a través de herramientas adecuadas.

El negocio debe entregar premisas claras para la publicación, tales como url del repositorio, versión, branch, tag, etc.

No es papel del equipo de infraestructura autorizar o decidir si algo debe ser publicado o no, esa competencia no cabe a ese equipo, quién debe decir si algo debe o no ir a producción son los resultados de las pruebas y el equipo de negocio.

En **Infraestructura Ágil** no existe llamados o tickets de deploy pues el proceso debe ser automatizado y el control de él - su gatillo, queda con el área de negocio.



# Entrega

Las pruebas que normalmente caben al equipo de infraestructura son responsables de indicar si una aplicación está listo o no para producción.

Si una nueva versión ha sido aprobada por todas las pruebas unitarias, de aceptación, funcionales, carga, rendimiento y seguridad. Esto significa que el software está listo para ser utilizado.

Lo ideal es que toda esta automatización en algún momento facilite la integración y la cooperación entre los equipos Ops y Devs.

En **Infraestructura Ágil** debemos hacer la entrega de nuevas versiones de una aplicación de forma segura, eficiente y con bajo riesgo, para eso les recomiendo ver el e-book “**Introduccion a Devops y DevSecOps**” y buscar mas información en [DevSecOps.org](https://DevSecOps.org)



# Métricas



# Métricas

No hay administración o administración de sistemas de infraestructura sin métricas.

- Generar datos
- Recopilar datos
- Procesar datos
- Almacenar datos
- Ver datos
- Transformar dado en información
- Consumir información

Para mejorar y evolucionar la infraestructura es necesario tener métricas disponibles para que podamos acompañar su funcionamiento y su desempeño.

Si no es posible medir su desempeño, no tenemos una referencia para realizar cambios y mejorarla.

Es necesario recopilar, procesar y transformar datos en informaciones relevantes que puedan ser efectivamente utilizadas por los equipos. Serán útiles para ayudar a evolucionar y mejorar la infraestructura y los servicios ofrecidos.



# Métricas

Los Sysadmins normalmente se centran mucho en el monitoreo de la infraestructura básica y se olvidan de las métricas más importantes para el negocio de sus clientes.

Monitorear es importante y esencial, sin embargo, no siempre podemos monitorear lo que necesitamos porque ciertos datos no existen y necesitan ser generados para que podamos entenderlos, estudiarlos y tomar acciones a partir de ellos.

¿Cómo usted sabe si el rendimiento de la aplicación principal de su cliente ha mejorado o ha empeorado después de la última publicación?

¿Cómo puedes medir la cantidad real de clientes utilizando su aplicación?

¿Cómo puedes medir la salud general de su aplicación y de su ambiente?

Estas son algunas de las preguntas que deberían empezar hacerse los sysadmins...





# Métricas

Podemos observar que esto va mucho más allá de chequear, puerto en servidor o cadena de retorno de alguna página.

- Determinadas métricas van a existir si las creamos;
- Determinadas informaciones sólo podrán ser consumidas si hay métricas para sostenerlas;
- Métricas deben ser generadas, recogidas y almacenadas en un lugar adecuado;

Existiendo tales métricas podemos crear información realmente útil sobre nuestras aplicaciones e infraestructura.



# Métricas

Es muy importante que la aplicación desde su concepción ya piense en métricas e informaciones para que sea posible medir su salud e información estructural, pero si tales datos no existen, deberán ser creados.

Las métricas claras permiten incluso la integración de su monitoreo con su automatización y aprovisionamiento, sus métricas pueden ser el gatillo para procesos internos de escalabilidad o reducción de recursos en el caso de poca demanda.

En **Infraestructura Ágil** las métricas son los fundamentos de la gestión de su infraestructura, ellas van a posibilitar planificación, van a permitirle analizar escenarios e incidentes y van a proporcionar respuestas importantes para sus clientes y usuarios.



# Personas



# Personas

Lo primero y fundamenta es que la gente tiene que sentirse parte del equipo e integrarse a él.

- Aplicación de métodos ágiles
- Integración del equipo
- Nivelación y uso compartido
- Definición de valores
- Definición de objetivos

Después de hablar de tantos modelos y tecnologías necesitamos hablar de personas, al final no es posible tener tantos modelos, tecnologías y continuar funcionando en una estructura de equipo clásica.



# Personas

Tal vez la parte más sensible de una iniciativa de **infraestructura Ágil** sea el cambio en la forma en que el equipo se comunique y trabaje.

En este modelo las personas necesitan comunicarse, expresarse, necesitan tener libertad creativa, necesitan direccionamiento y principalmente de foco.

En un equipo ágil necesitamos:

- Fomentar el intercambio de conocimiento
- Fomentar la integración de su equipo
- Ayudar a su equipo a encontrar lo que necesita para motivarse
- Ofrecer los medios para que tengan dirección y conciencia del trabajo
- Fomentar la creatividad y la innovación



# Personas

Para ayudar a su equipo, haga una planificación colaborativa de corto, mediano y largo plazo, trate de responder juntos a las siguientes preguntas:

¿Cuál es el estado deseado de nuestra infraestructura dentro de 3, 6, 9 y 12, 24 meses?

¿Qué está buscando nuestra organización en 24 meses?

¿Qué resultado buscamos como equipo?

Ayúdalos a ver el camino a recorrer y todo será más fácil.

Recuerde que si las personas están motivadas, si tienen libertad creativa, incentivo, direccionamiento y los recursos necesarios para hacer su trabajo, esto se reflejará en resultados positivos para el profesional, para el equipo y para la organización.





# Personas

Métodos que pueden ser utilizados y adaptados para equipos de infraestructura:

- Kanban
- Scrum
- Dojos
- Gamification
- Hacklabs
- Hackatons
- Cualquier método ágil
- Práctica de deportes colectivos
- Prácticas colectivas recreativas

Adapte los métodos a su organización ya su equipo.

La integración de su equipo debe ocurrir dentro y fuera del lugar de trabajo, el entretenimiento externo es importante.

A pesar de todos los métodos y tecnologías, lo importante son siempre las personas.



# Resultados



# Resultados

El uso de infraestructura ágil traeran los siguientes resultados y ventajas empresariales

## Obtención de valor más rápido

Al eliminar completamente los lentos procesos manuales de adquisición y construcción, los desarrolladores y operadores pueden centrarse en el desarrollo para completar sus procesos mucho más rápido que en el pasado. Una obtención de valor más rápida significa que las ventajas operativas y económicas del proceso de desarrollo pueden materializarse mucho antes.

## Aumento de la productividad del personal

**DevOps** es una mentalidad cultural donde la productividad es el resultado clave.

Al no tener la necesidad de construir manualmente la infraestructura, los desarrolladores y operadores pueden ser mucho más productivos que en el pasado. La infraestructura que se alinea con la forma en la que colaboran los equipos crea una sinergia que no puede lograrse con otras arquitecturas



# Resultados

## Flexibilidad y agilidad

Mediante el uso de la **Infraestructura Ágil**, las organizaciones consiguen la capacidad de permitir que el software construya su propio entorno operativo en función de las necesidades del momento. **DevOps** y la **Infraestructura Ágil** no solo se centran en el desarrollo, sino que también tienen que ver con lo que pasa en un entorno de producción.

Imagine un escenario en el que una aplicación de producción pueda detectar que la granja de servidores web está a punto de alcanzar su capacidad máxima, implementar proactivamente más servidores web y reconfigurar el equilibrador de carga para agregar estos nuevos recursos, todo ello sin la intervención de un desarrollador o un operador.

Ésa es la magia de la **Infraestructura Ágil** en todo su esplendor.



# Resultados

## Costes reducidos

La **Infraestructura Ágil** puede tener un impacto positivo en la estructura de costes de una organización de varias maneras:

Costes de personal reducidos o redirigidos. Incluso si no se reduce el personal con el modelo **DevOps**, se destinan más esfuerzos a las características que añaden valor que a las pesadas tareas manuales.

- Mayor aprovechamiento de la infraestructura. Ya no es necesario sobreaprovisionar los recursos. Las empresas pueden habilitar al entorno de software para aprovechar la infraestructura existente hasta el máximo de su potencial. Los nichos que solían separar los elementos de la infraestructura pueden eliminarse.

Menos defectos. Los defectos en el software tienen un coste enorme. Los entornos y procesos de desarrollo tradicionales tienen una gran carga manual y son propensos a errores. Con la automatización, las empresas pueden reducir e impedir proactivamente la aparición de errores, lo que suaviza el impacto de los defectos potenciales.



# Resultados

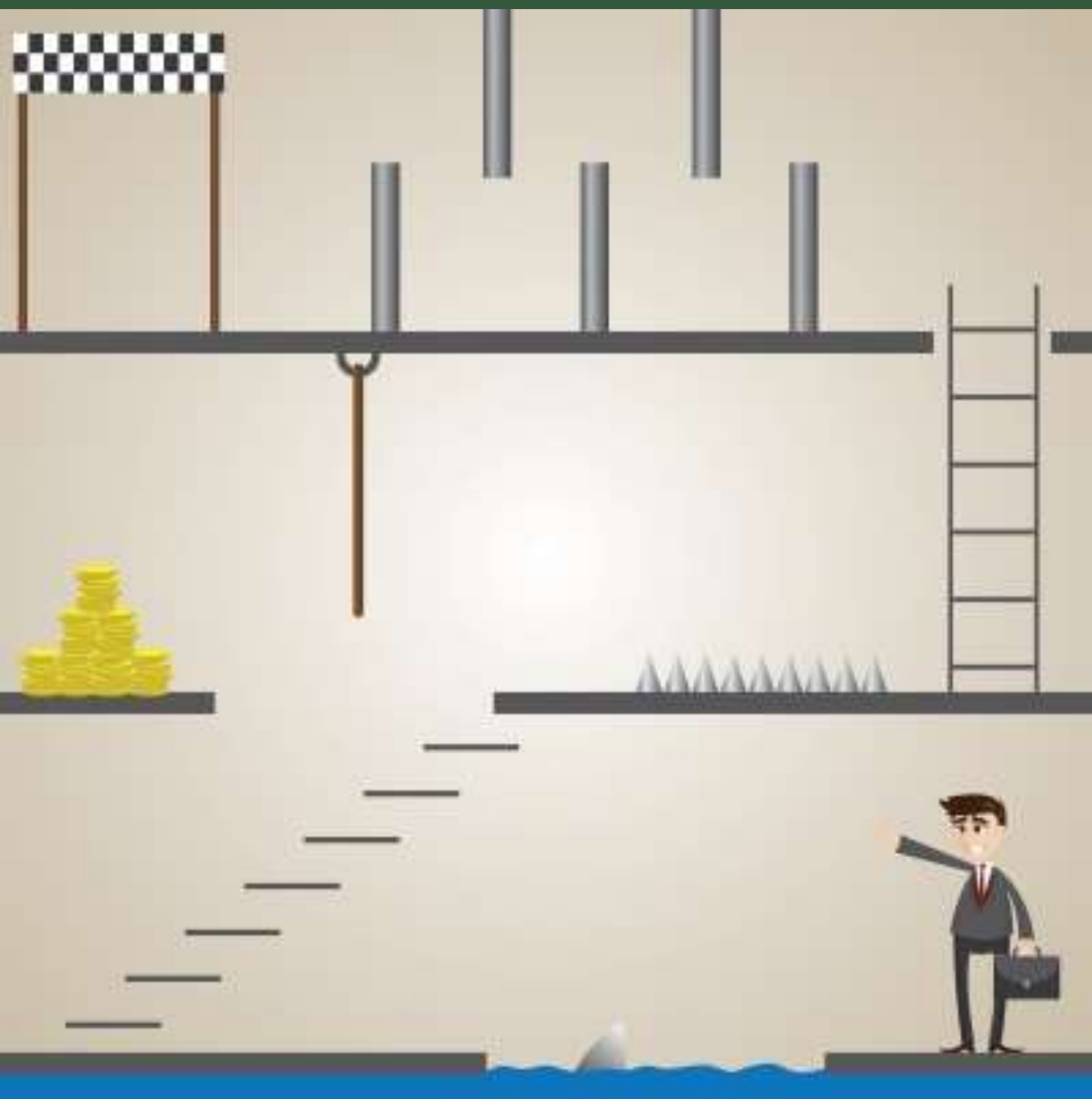
En última instancia, la **Infraestructura Ágil** resulta idónea para las necesidades de la comunidad **DevOps**. Esto le permite:

- Ejecutarlo todo: optimice cualquier aplicación y almacene todos los datos
- Actuar con mayor rapidez: acelere la entrega de aplicaciones y servicios
- Trabajar de manera eficaz: reduzca los esfuerzos y el coste operativo
- Liberar el valor: aumente la productividad y el control





# Transición



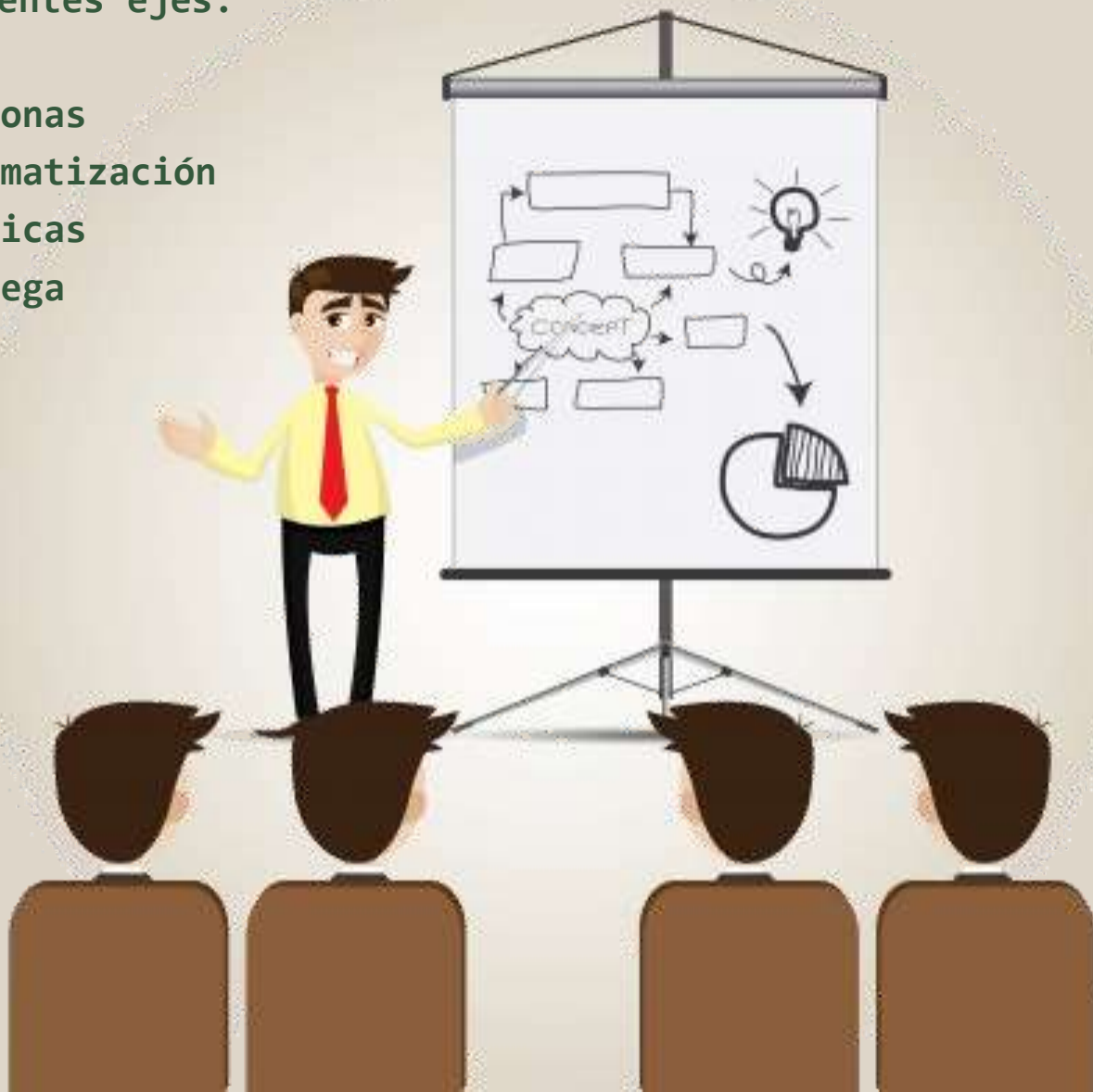
# Transición

Toda la iniciativa de **Infraestructura Ágil** comienza por la transformación de la cultura:

Charlando con su equipo, presente los conceptos, proporcione información, el camino que desea seguir, haga una planificación colaborativa, traiga y adapte método Agiles y elija las herramientas para realizar la automatización.

Se recomienda iniciar el proceso desde los siguientes ejes:

- Personas
- Automatización
- Métricas
- Entrega



# Transición

También recomendamos para el inicio de su transición a **Infraestructura Ágil** buscar un proveedor de servicios que este familiarizado con el termino **Infraestructura Ágil**, con foco en los siguientes servicios:

Consultoría sobre automatización y monitoreo de infraestructura

Elaboración de proyectos para atender a sus expectativas y transformar su estrategia de la infraestructura, dejando esta más robusta y eficiente a partir de la automatización

Implantación de tecnologías de automatización, orquestación, métricas, deploy y gerencia de configuración

Desarrollo de módulos para automatizar sus ambientes y sus necesidades aplicaciones

Seminarios de **Infraestructura Ágil**

Pero sobre todo alineado con la cultura **DevOps**, **DevSecOps** y las metodologías Agiles.



# Dinámica de trabajo

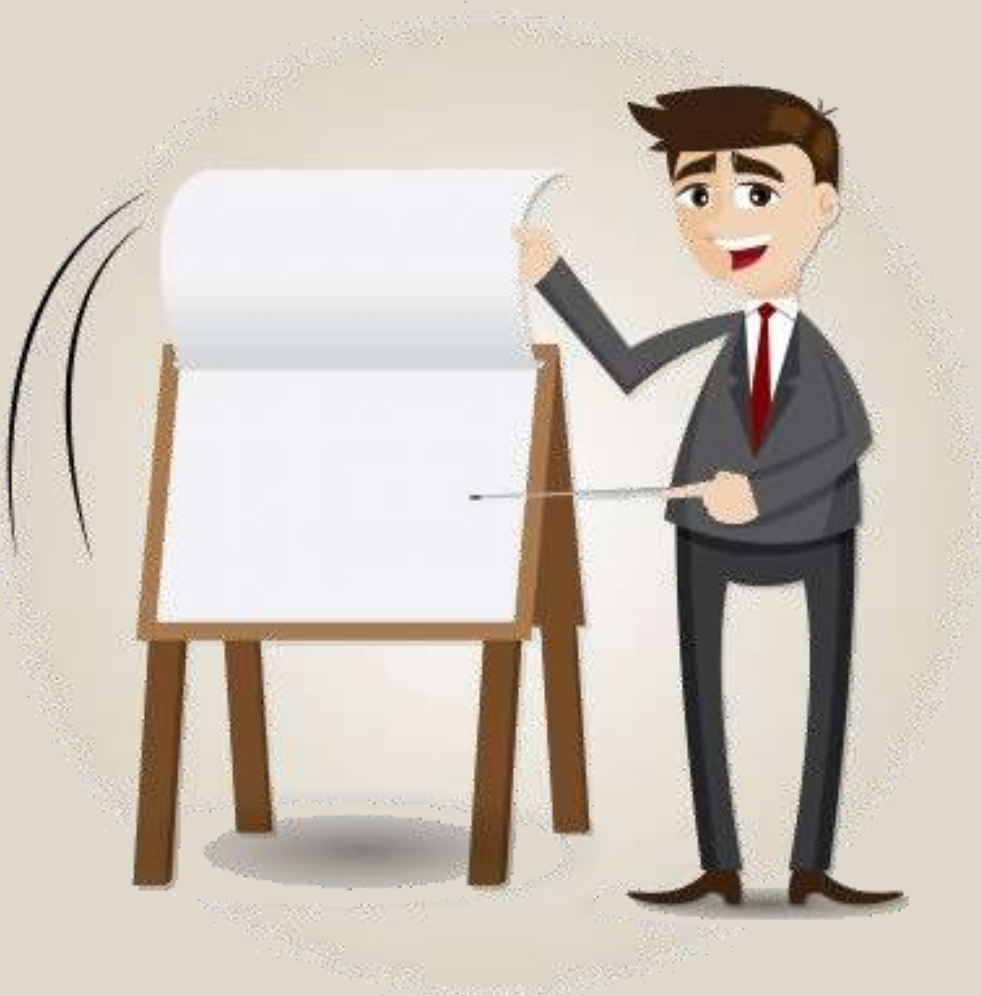


# Dinámica de trabajo

Iniciar un ciclo de encuentros ágiles con el objetivo de mejorar la comunicación, planificación, seguimiento y ejecución de actividades.

Este método se divide de la siguiente manera:

- Definición de actividades por los equipos
- Planificación y aprobación de actividades
- Retrospectiva de ejecución de actividades
- Cierre y evaluación



# Dinámica de trabajo

## Definiendo las actividades

Por ejemplo los equipos Windows, Linux y BBDD deben definir las actividades que desean realizar en los próximos días.

Las actividades pueden ser mejoras, correcciones, actualizaciones o implantación de nuevas tecnologías.

Este proceso de definición es hecho por cada equipo y no necesita reunión, basta que los miembros conversen y definan las actividades, éstas serán utilizadas en el encuentro de planificación.

Las actividades propuestas se pueden colocar en un wiki.





# Dinámica de trabajo

## Encuentro de planificación

Lo ideal es que se haga un encuentro siempre el lunes.

La planificación tiene el objetivo de evaluar las propuestas y aprobar las actividades que se ejecutarán en la próxima semana (s).

Duración: El encuentro deberá ser rápido, 5 minutos para cada equipo, totalizando 15 minutos para 3 equipos.

Los gestores tendrán más 5 minutos para aprobar, rechazar, dirigir o priorizar actividades.

Estos 5 minutos se deben medir en el reloj o en el teléfono.

Si es posible utilizar un cronómetro visible a todos del encuentro.

Alguien debe ser el facilitador conduciendo el encuentro.



# Dinámica de trabajo

## Encuentro de planificación

El facilitador debe:

- Dar la palabra a cada equipo, leer las propuestas presentadas
- Controlar el tiempo de cada equipo para quedar dentro de los 5 minutos
- Controlar el tiempo del gestor para permanecer dentro de los 5 minutos
- Consolidar lo que se aprobó en el wiki
- Evitar que los asuntos fuera de las actividades presentadas sean abordados
- Direccionar la planificación para que se apague rápidamente
- La planificación no debe durar más de 25 minutos.



# Dinámica de trabajo

## Retrospectivas

Las retrospectivas son un concepto completamente nuevo para los equipos de IT. Cada equipo del departamento debe realizar una retrospectiva cada dos semanas.

¿Lo que funciona bien se puede olvidar?

¿Qué debemos hacer distinto la próxima vez?

¿Qué nos sigue rompiendo la cabeza?

Todos los equipos deben mantener a sus bloqueadores una vez desbloqueados y revisar en cada retrospectiva para buscar causas comunes de retraso.

Después de la primera retrospectiva de cada equipo un comentario común es“

¿Cada dos semanas no es demasiado frecuentes?

¿Tendremos algo que discutir en la próxima retro?

“ Pasaran meses y veras que todos todavía seguirán llegando con mejoras. Una vez que los equipos adquieran el hábito de las retrospectivas, se les anima a "doblar las reglas" y no esperar hasta la próxima retrospectiva para actuar sobre las mejoras.



# Dinámica de trabajo

## Cierre

En el cierre, las actividades se actualizan de nuevo en el wiki.

Los gestores harán un análisis de las actividades cerradas en los últimos 15 días.

Los gestores darán feedback a los equipos.

Los equipos podrán posicionarse sobre el análisis de los gestores.

El facilitador debe controlar el tiempo y registrar los cambios en el wiki.



# Dinámica de trabajo

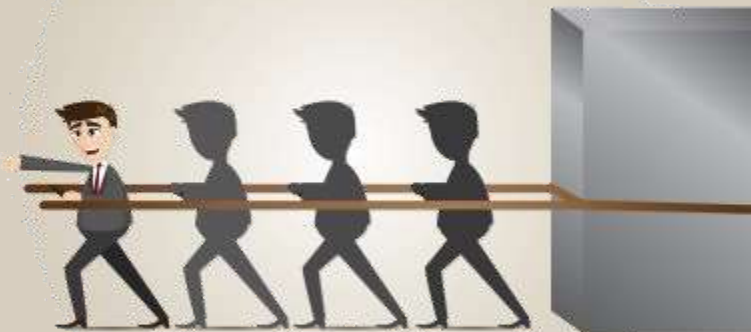
## Reunión diaria de Pie

Cada equipo hace una reunión diaria parados frente al tablero de control, se habla del mismo de derecha a izquierda, concentrándose donde sea necesario y enfocándose en desbloquear trabajo.

El SCRUM tiene una técnica de encuentros diarios que es muy interesante, ocurre normalmente en pie y no dura más de 10 minutos.

En este encuentro las personas del equipo intercambian información rápida sobre lo que está sucediendo, normalmente para agilizar a cada persona responde tres preguntas:

¿Qué hice ayer?  
¿Qué estoy haciendo hoy?  
¿Qué haré mañana?  
Y si alguien tiene dificultad, pedirá ayuda, quien pueda ayudar, lo hará después del encuentro.



Es un encuentro muy rápido, cada uno debe hablar como máximo 3 minutos (el ideal es 1 minuto).

# Dinámica de trabajo

## Reunión Diaria de Pie

El facilitador debe controlar el tiempo apenas - anotar no es obligatorio.

El objetivo principal es hacer que el equipo se comunique mejor.

El gestor oyendo al equipo puede ayudar a quien tiene dificultades.

El gestor oyendo el equipo tendrá una idea precisa del progreso de las actividades.

El equipo pasa a valorar el trabajo de cada miembro.



# Dinámica de trabajo

## Cuadro de control

Si es posible, el equipo puede y debe montar un Kanban Board con las actividades.

El Kanban Board permite visualizar las actividades de todos los equipos claramente y de forma fácil.

Cada miembro debe mover las actividades en el cuadro según su evolución.

El gestor no necesita preguntarse sobre el progreso de las cosas, basta mirar el cuadro.

Se puede utilizar un cuadro blanco y escribir en él.

Se pueden utilizar post-it's coloreados y fijar éstos en el cuadro (preferible).

Se puede utilizar una herramienta WEB de KANBAN como el Trello cuando se tienen personas externas.





# Kanban para IT OPS



# Kanban para IT OPS

Junto con “podríamos introducir Kanban a nuestra área de IT”, la primera pregunta siempre debe ser ¿Por qué?

Bien, aquí está el porqué:

- Muchos equipos (DBA, redes, administradores de sistemas, soporte técnico, equipo de deploy) a menudo de forma inadvertida trabajan entre ellos.
- Todas las comunicaciones son hechas a través de un software de helpdesk o mails, Incluso cuando los individuos se sientan uno al lado del otro resultando en muy poca colaboración y trabajo en equipo.
- Gran carga de horas extraordinarias con muchas personas que trabajan regularmente unas 70 horas a la semana.
- Muchos proyectos en vuelo - priorización por "el que grita más fuerte".
- Los gerentes de Proyectos van de cabeza a cabeza sobre recursos compartidos y aplicando una presión injusta a los equipos de entrega.
- Reducción de tiempos del equipo entre los proyectos, áreas luchando por gestionar a muchos interesados y lograr un equilibrio saludable entre proyectos, incidentes y BAU.
- Plazos regularmente no cumplidos, capacidad desperdiciada - particularmente alrededor de las transferencias, el tiempo de espera, y mucho de trabajo hecho parcialmente.

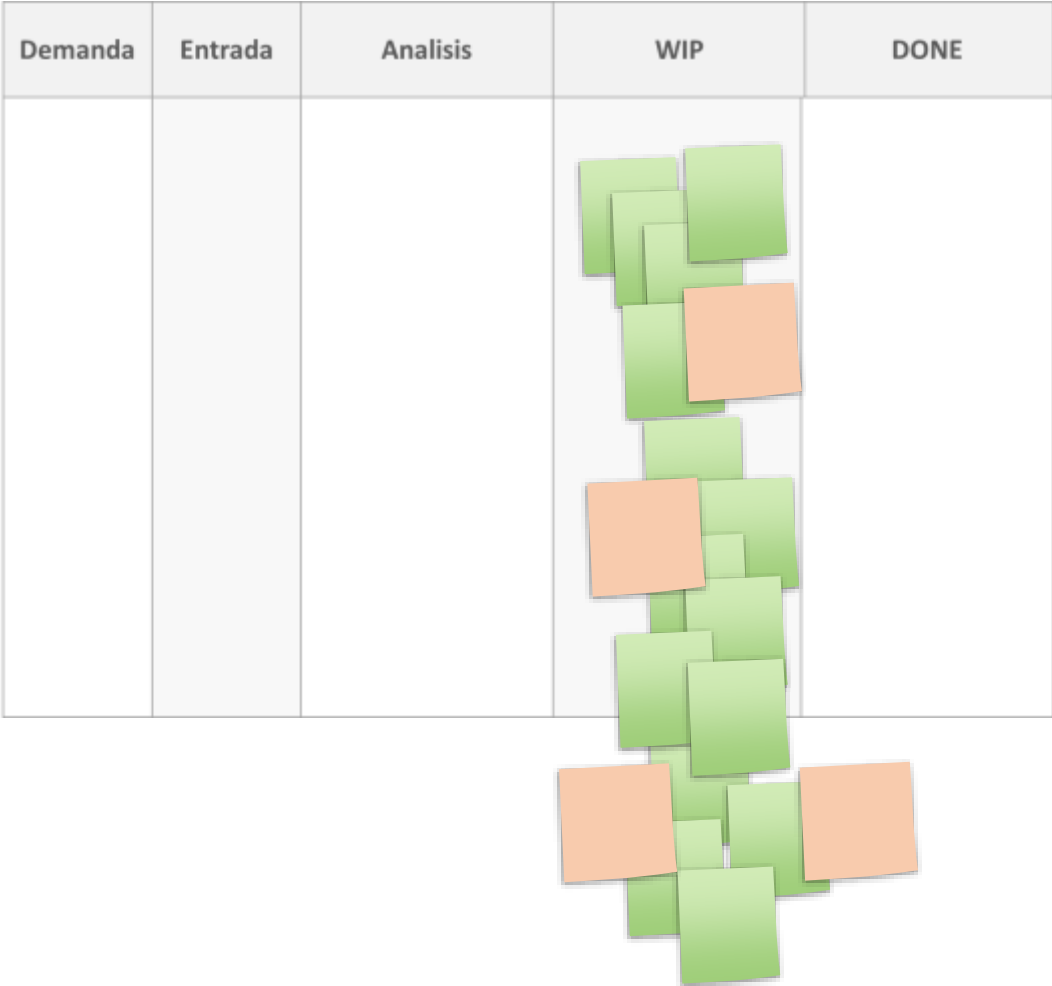


# Kanban para IT OPS

El primer paso es simplemente tomar su carga de trabajo existente y visualizarla en un tablero.

Es un patrón repetitivo ya visto en cada equipo que implementa tableros de control que cuando visualizan su trabajo - el estilo de Kanban el mismo demuestra constantemente mucho trabajo en progreso.

Los equipos en primera estancia debe centrarse en reducir este WIP y mantener un flujo saludable.



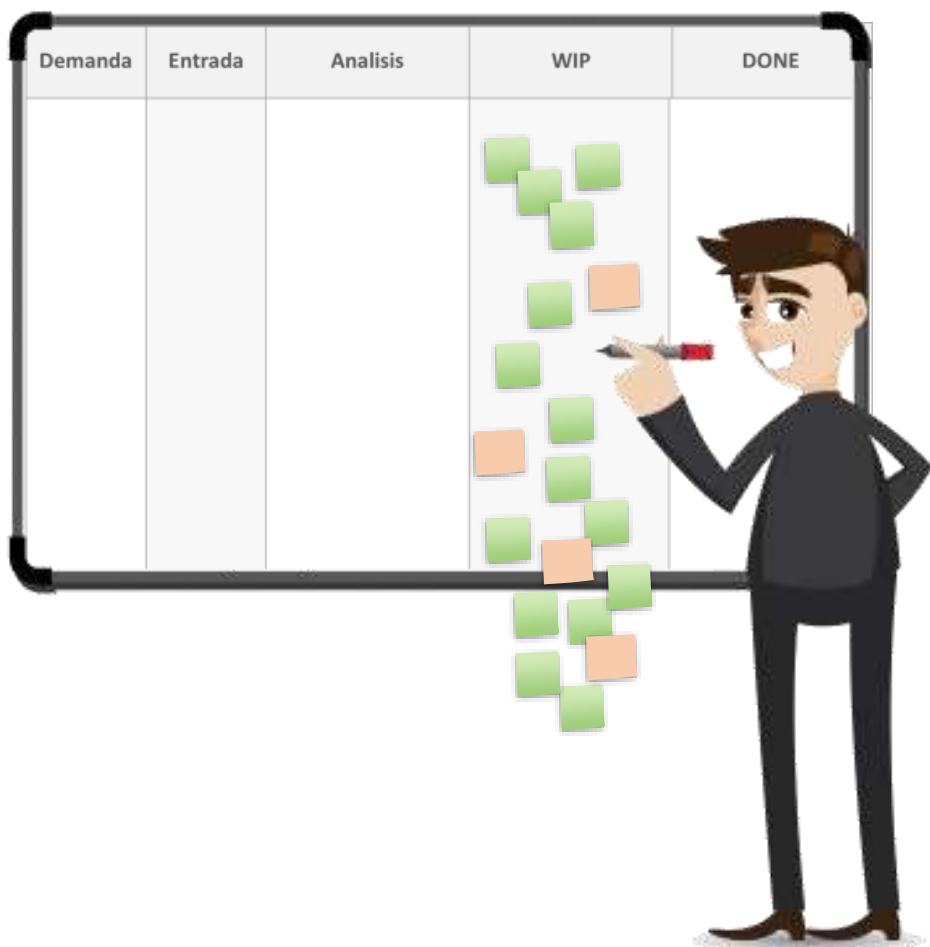
# Kanban para IT OPS

Recomendamos que cada equipo opere un sistema de dos tableros.

Un tablero para la demanda y otro para la entrega.

Sería bueno tener toda la cadena de valor representada en un solo tablero/pared, pero normalmente la logística hace que esta configuración sea necesaria.

La mayoría del retraso se encuentra en el tablero de demanda y organizado por las diversas partes interesadas, cada tablero de entrega tiene una parte del tablero de demanda en él para actuar como un embudo para el trabajo entrante.



# Kanban para IT OPS

## Tablero de Administrador de Sistemas

Demanda				Entrada	WIP	Sign OFF	DONE
clientes							
Other							
BAU							
.....							
Programado para despliegue							
indicadores	vacaciones	reuniones	licencias	Acelerar →			



# Kanban para IT OPS



## Tablero de Administrador de Sistemas

### Columna de Demanda:

Se utiliza para visualizar a los diversos actores y para hacer que piensen en las prioridades.

### Columna de Entrada:

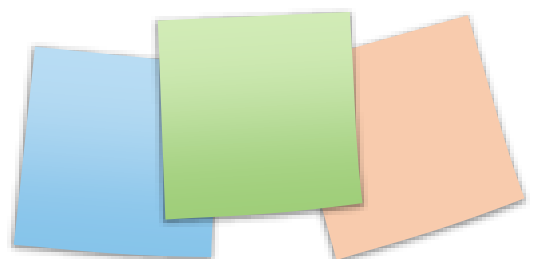
Esta es posiblemente la columna más importante en el tablero. Esta columna promueve la conversación de priorización con las partes interesadas y proporciona protección para el equipo de entrega. Obliga a las conversaciones que solían ser evitadas y a menudo empujadas al equipo de entrega para priorizarlas.

### Columna de WIP:

Los miembros del equipo extraen el trabajo de la columna de entrada **sólo cuando están listos para hacerlo**. Si se bloquean en un trabajo, primero se concentran en desbloquear el problema o ayudar a otros miembros del equipo con sus tareas.

### Columna de Programación:

Una vez que un trabajo se ha hecho a menudo tiene que esperar una ventana de mantenimiento programado (a menudo fuera de horas) antes de que pueda ser desplegado y poner en vivo. Esta columna es un tipo de espera para la columna de liberación y se utiliza para planificar las ventanas de mantenimiento.



# Kanban para IT OPS



## Tablero de Administrador de Sistemas

### Sobre de residuos:

Todos los desechos incurridos por el equipo fueron capturados, cuantificados y almacenados en el sobre de residuos para su discusión durante las retrospectivas.

### Sobre de Bloqueos:

Todas las notas post-it magentas (bloqueadores) se mantienen una vez que se ha eliminado el bloqueo. Estos bloqueadores se analizan durante las retrospectivas para buscar tendencias, bloqueos comunes o cualquier otra información para ayudar en la mejora del proceso.

### Métricas:

Crear un formulario basado en papel a la derecha del tablero para que simplifique la generación de un diagrama de flujo acumulativo (CFD). Normalmente es más fácil escribir el número de cartas para cada columna en el formulario y luego actualizar el gráfico electrónico periódicamente - por ejemplo, al final de cada semana.

### La Tarjeta de Infra:

Cada día se inserta una nueva tarjeta directamente en la columna WIP (se reserva una ranura para no sobrepasar el límite wip). Cada día entre las 13:30 y las 15:00 el equipo impacta las filas de Infra para que su equipo complete lo que puede dentro de la caja de tiempo. El número de trabajos terminados / tickets en infra se agrega a la tarjeta y luego colocado en la columna de terminados. El número de trabajos / tickets completados se agrega al valor del CFD terminado.



# Kanban para IT OPS

## Tablero de Aplicaciones de Negocio



# Kanban para IT OPS



## Tablero de Aplicaciones de Negocio

### Columna de Demanda:

Se utiliza para visualizar a los diversos actores y para hacer que piensen en las prioridades.

### Columna de entrada:

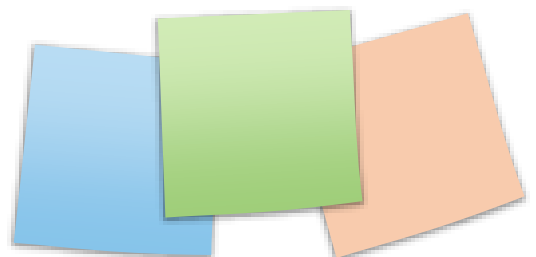
Esta es posiblemente la columna más importante en el tablero. Esta columna promueve la conversación de priorización con las partes interesadas y proporciona protección para el equipo de entrega. Obliga a las conversaciones que solían ser evitadas y a menudo empujadas al equipo de entrega para priorizarlas.

### Columna WIP

Los miembros del equipo extraen el trabajo de la columna de entrada **sólo cuando están listos para hacerlo**. Si se bloquean en un trabajo, primero se concentran en desbloquear el problema o ayudar a otros miembros del equipo con sus tareas.

En este equipo en particular a veces se opta por un diseño basado en grilla para incluir sus nombres individuales para que todos puedan ver quién trabaja en qué. Puede que inicialmente, se encuentren con algunos comportamientos erróneos de la administración de ver una ranura vacía y tratando de querer llenarlo.

Este comportamiento es incorrecto y se resuelve haciendo una política explícita en torno al propósito del tablero y asegurando que todos los compromisos fueron comprados y honrados.



# Kanban para IT OPS



## Tablero de Aplicaciones de Negocio

### Columna de Programación:

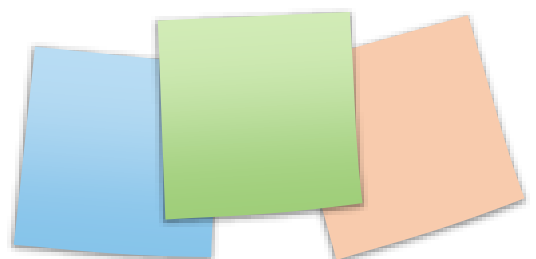
Una vez que un trabajo se ha hecho a menudo tiene que esperar una ventana de mantenimiento programado (a menudo fuera de horas) antes de que pueda ser desplegado y poner en vivo. Esta columna es un tipo de espera para la columna de liberación y se utiliza para planificar las ventanas de mantenimiento.

### Bloqueadores de Largo Plazo:

Esta sección del tablero que normalmente tiene un montón de post-it magenta en él es el área de los bloqueadores a largo plazo. Este equipo depende muchas veces de terceras partes y regularmente se bloquean mientras espera a la tercera parte para completar algo. Desafortunadamente, los contratos con estas terceras partes no proporcionan suficiente influencia para que respondan más rápido, por lo que el equipo solo puede hacerlas visibles. Hacerlos visibles permite a la alta dirección ver exactamente quién los esta bloqueando y perseguir a los proveedores a diario, aplicando presión cuando es posible.

### Sobre de Residuos:

Todos los desechos incurridos por el equipo fueron capturados, cuantificados y almacenados en el sobre de residuos para su discusión durante las retrospectivas.



# Kanban para IT OPS



## Tablero de Aplicaciones de Negocio

### Sobre de Bloqueadores:

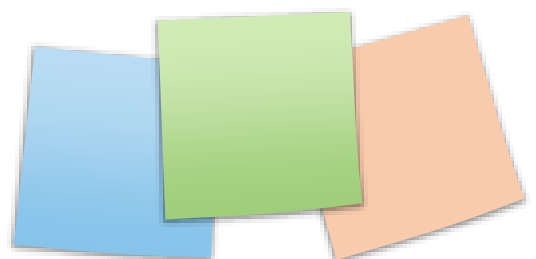
Todas las notas post-it magentas (bloqueadores) se mantienen una vez que se ha eliminado el bloqueo. Estos bloqueadores se analizan durante las retrospectivas para buscar tendencias, bloqueos comunes o cualquier otra información para ayudar en la mejora del proceso.

### Métricas:

Crear un formulario basado en papel a la derecha del tablero para que simplifique la generación de un diagrama de flujo acumulativo (CFD). Normalmente es más fácil escribir el número de cartas para cada columna en el formulario y luego actualizar el gráfico electrónico periódicamente - por ejemplo, al final de cada semana.

### La Tarjeta de Infra:

Cada día se inserta una nueva tarjeta directamente en la columna WIP (se reserva una ranura para no sobrepasar el límite wip). Cada día entre las 13:30 y las 15:00 el equipo impacta las filas de Infra para que su equipo complete lo que puede dentro de la caja de tiempo. El número de trabajos terminados / tickets en infra se agrega a la tarjeta y luego colocado en la columna de terminados. El número de trabajos / tickets completados se agrega al valor del CFD terminado.



# Kanban para IT OPS

## Tablero DBA



# Kanban para IT OPS



## Tablero DBA

### Columna de Demanda:

Se utiliza para visualizar a los diversos actores y para hacer que piensen en las prioridades.

### Columna de Entrada:

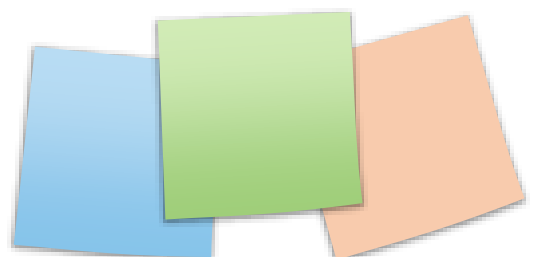
Esta es posiblemente la columna más importante en el tablero. Esta columna promueve la conversación de priorización con las partes interesadas y proporciona protección para el equipo de entrega. Obliga a las conversaciones que solían ser evitadas y a menudo empujadas al equipo de entrega para priorizarlas.

### Columna WIP:

Los miembros del equipo extraen el trabajo de la columna de entrada **sólo cuando están listos para hacerlo**. Si se bloquean en un trabajo, primero se concentran en desbloquear el problema o ayudar a otros miembros del equipo con sus tareas.

### Columna de Programación:

Una vez que un trabajo se ha hecho a menudo tiene que esperar una ventana de mantenimiento programado (a menudo fuera de horas) antes de que pueda ser desplegado y poner en vivo. Esta columna es un tipo de espera para la columna de liberación y se utiliza para planificar las ventanas de mantenimiento.



# Kanban para IT OPS



## Tablero DBA

### Bloqueadores de Largo Plazo:

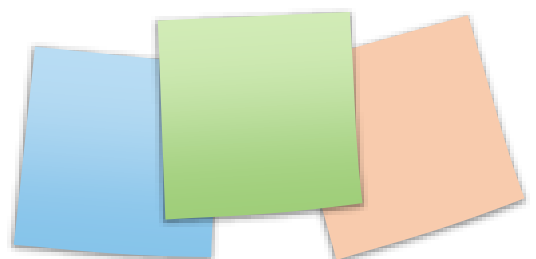
Esta sección del tablero que normalmente tiene un montón de post-it magenta en él es el área de los bloqueadores a largo plazo. Este equipo depende muchas veces de terceras partes y regularmente se bloquean mientras espera a la tercera parte para completar algo. Desafortunadamente, los contratos con estas terceras partes no proporcionan suficiente influencia para que respondan más rápido, por lo que el equipo solo puede hacerlas visibles. Hacerlos visibles permite a la alta dirección ver exactamente quién los esta bloqueando y perseguir a los proveedores a diario, aplicando presión cuando es posible.

### Sobre de Residuos:

Todos los desechos incurridos por el equipo fueron capturados, cuantificados y almacenados en el sobre de residuos para su discusión durante las retrospectivas.

### Sobre de Bloqueadores:

Todas las notas post-it magentas (bloqueadores) se mantienen una vez que se ha eliminado el bloqueo. Estos bloqueadores se analizan durante las retrospectivas para buscar tendencias, bloqueos comunes o cualquier otra información para ayudar en la mejora del proceso.





# Kanban para IT OPS

## Tablero DBA

### Métricas:

Crear un formulario basado en papel a la derecha del tablero para que simplifique la generación de un diagrama de flujo acumulativo (CFD). Normalmente es más fácil escribir el número de cartas para cada columna en el formulario y luego actualizar el gráfico electrónico periódicamente - por ejemplo, al final de cada semana.

### La Tarjeta de Infra:

Cada día se inserta una nueva tarjeta directamente en la columna WIP (se reserva una ranura para no sobrepasar el límite wip). Cada día entre las 13:30 y las 15:00 el equipo impacta las filas de Infra para que su equipo complete lo que puede dentro de la caja de tiempo. El número de trabajos terminados / tickets en infra se agrega a la tarjeta y luego colocado en la columna de terminados. El número de trabajos / tickets completados se agrega al valor del CFD terminado.



# Kanban para IT OPS

## Tablero de Redes e Infraestructura



# Kanban para IT OPS



## Tablero de Redes e Infraestructura

### Columna de Demanda:

Se utiliza para visualizar a los diversos actores y para hacer que piensen en las prioridades.

### Columna de Entrada:

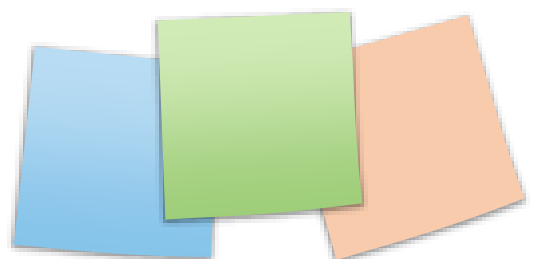
Esta es posiblemente la columna más importante en el tablero. Esta columna promueve la conversación de priorización con las partes interesadas y proporciona protección para el equipo de entrega. Obliga a las conversaciones que solían ser evitadas y a menudo empujadas al equipo de entrega para priorizarlas.

### Columna WIP:

Los miembros del equipo extraen el trabajo de la columna de entrada **sólo cuando están listos para hacerlo**. Si se bloquean en un trabajo, primero se concentran en desbloquear el problema o ayudar a otros miembros del equipo con sus tareas.

### Columna de Programación:

Una vez que un trabajo se ha hecho a menudo tiene que esperar una ventana de mantenimiento programado (a menudo fuera de horas) antes de que pueda ser desplegado y poner en vivo. Esta columna es un tipo de espera para la columna de liberación y se utiliza para planificar las ventanas de mantenimiento.



# Kanban para IT OPS



## Tablero de Redes e Infraestructura

### Bloqueadores de Largo Plazo:

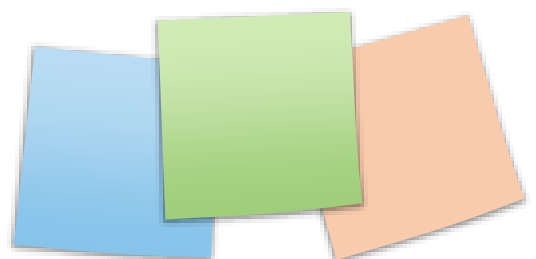
Esta sección del tablero que normalmente tiene un montón de post-it magenta en él es el área de los bloqueadores a largo plazo. Este equipo depende muchas veces de terceras partes y regularmente se bloquean mientras espera a la tercera parte para completar algo. Desafortunadamente, los contratos con estas terceras partes no proporcionan suficiente influencia para que respondan más rápido, por lo que el equipo solo puede hacerlas visibles. Hacerlos visibles permite a la alta dirección ver exactamente quién los esta bloqueando y perseguir a los proveedores a diario, aplicando presión cuando es posible.

### Sobre de Residuos:

Todos los desechos incurridos por el equipo fueron capturados, cuantificados y almacenados en el sobre de residuos para su discusión durante las retrospectivas.

### Sobre de Bloqueadores:

Todas las notas post-it magentas (bloqueadores) se mantienen una vez que se ha eliminado el bloqueo. Estos bloqueadores se analizan durante las retrospectivas para buscar tendencias, bloqueos comunes o cualquier otra información para ayudar en la mejora del proceso.



# Kanban para IT OPS



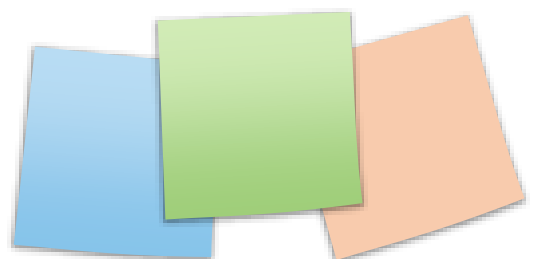
## Tablero de Redes e Infraestructura

### Métricas:

Crear un formulario basado en papel a la derecha del tablero para que simplifique la generación de un diagrama de flujo acumulativo (CFD). Normalmente es más fácil escribir el número de cartas para cada columna en el formulario y luego actualizar el gráfico electrónico periódicamente - por ejemplo, al final de cada semana.

### La Tarjeta de Infra:

Cada día se inserta una nueva tarjeta directamente en la columna WIP (se reserva una ranura para no sobrepasar el límite wip). Cada día entre las 13:30 y las 15:00 el equipo impacta las filas de Infra para que su equipo complete lo que puede dentro de la caja de tiempo. El número de trabajos terminados / tickets en infra se agrega a la tarjeta y luego colocado en la columna de terminados. El número de trabajos / tickets completados se agrega al valor del CFD terminado.



# Kanban para IT OPS



## Desafíos

### Gestiones de Proyecto

Si pensamos hoy por hoy , los gerentes de proyecto dan un plazo limite a los equipos y "administradores" para entregar en fecha o tan cerca como sea posible, con el uso de la metodología e priorización del que grita más fuerte.

En el período de transición del viejo mundo al nuevo mundo (Ágil), los gerentes de proyecto se ven obligados a ir cabeza a cabeza en las tareas de priorización - en lugar de culpar al equipo de entrega.

La ausencia de alguien que va cabeza a cabeza es algo malo.




Este es un problema específico del sistema y sólo podría tratarse en el nivel del porfolio.


Consulte la sección Gestión del porfolio para obtener información sobre cómo se soluciona este problema en particular (sí, ¡es posible!).

El mejor resultado para los equipos es la valla de seguridad que Kanban les proporciona para concentrarse en hacer su trabajo correctamente y con orgullo.

# Kanban para IT OPS

## Gestión del porfolio

Backlog	Escope	WIP Abstracto	WIP	WIP 3ra Partes	DONE
			Sysadmin 	 	
			BAE 		
			DBAs 		
			Networking 		
			Field		
			Service Desk		





# Kanban para IT OPS

## Gestión del portfolio

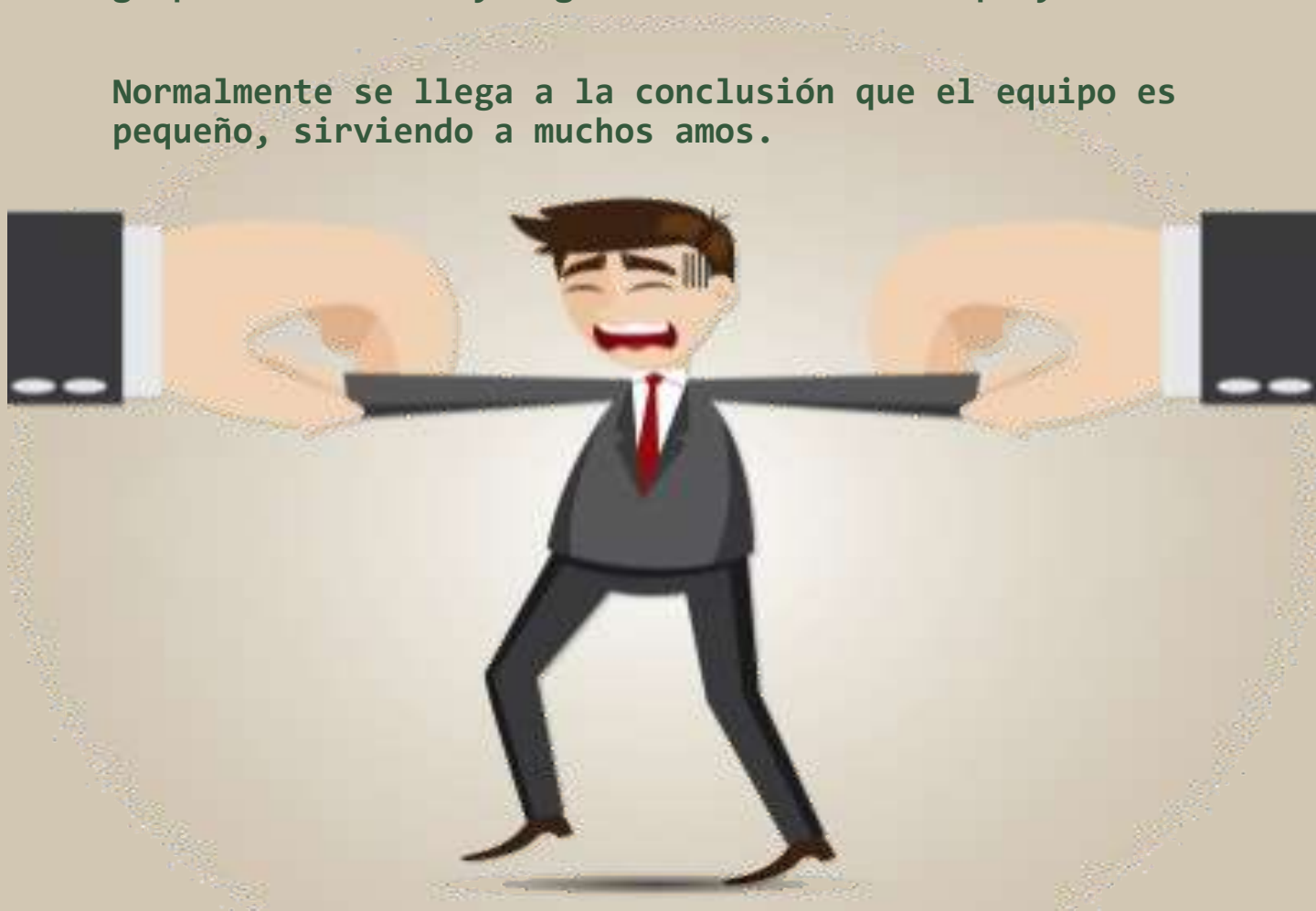
El tablero de gestión del portfolio se utiliza para visualizar simplemente el WIP actual.

Normalmente queda evidente que el departamento tiene demasiados proyectos en vuelo simultáneamente.

La codificación de color se utiliza para mostrar la relación de contención entre los distintos equipos.

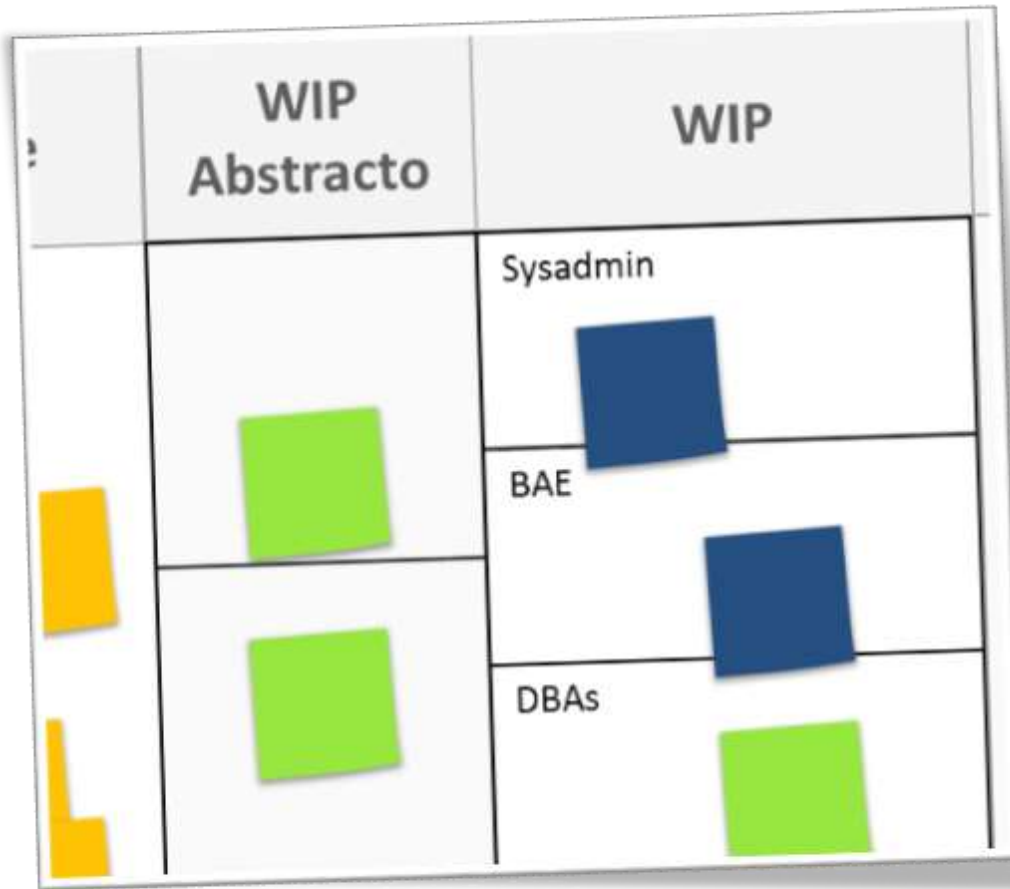
Brindando visibilidad, así los gerentes de proyectos pueden colaborar restableciendo las expectativas de los grupos de interés y organizando la cola de proyectos.

Normalmente se llega a la conclusión que el equipo es pequeño, sirviendo a muchos amos.



# Kanban para IT OPS

Gestión del portfolio



# Kanban para IT OPS

## Gestión del portfolio

### WIP Abstracto y WIP Actual

Una vez que hayamos introducido Kanban en la operación de los diversos equipos de entrega y, por lo tanto, limitado su trabajo en curso, se hará evidente que, aunque los gerentes de proyecto tenían proyectos en curso, en realidad no estaban avanzando.

Para representar esto, se usan dos columnas para visualizar el nivel del portfolio WIP. **WIP abstracto y WIP real.**

### ¡Importante!

Un gran punto de aprendizaje de este ejercicio es la importancia de no mirar el portfolio de forma aislada. Tradicionalmente, los portfolios se consideran la única fuente de la verdad, en este nuevo mundo, el portfolio sólo se utiliza para visualizar el estado general de los proyectos.

La única fuente de verdad es el equipo de entrega individual. Los tableros de entrega muestran no sólo el trabajo del proyecto, sino que también muestran las otras clases de demanda: incidentes, BAU y solicitudes de línea de servicio.



# Links recomendados



# 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>





# 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.freedigitalphotos.net/images/view\\_photog.php?photogid=5903](http://www.freedigitalphotos.net/images/view_photog.php?photogid=5903) "Diseñada por iosphere "



# Conclusiones



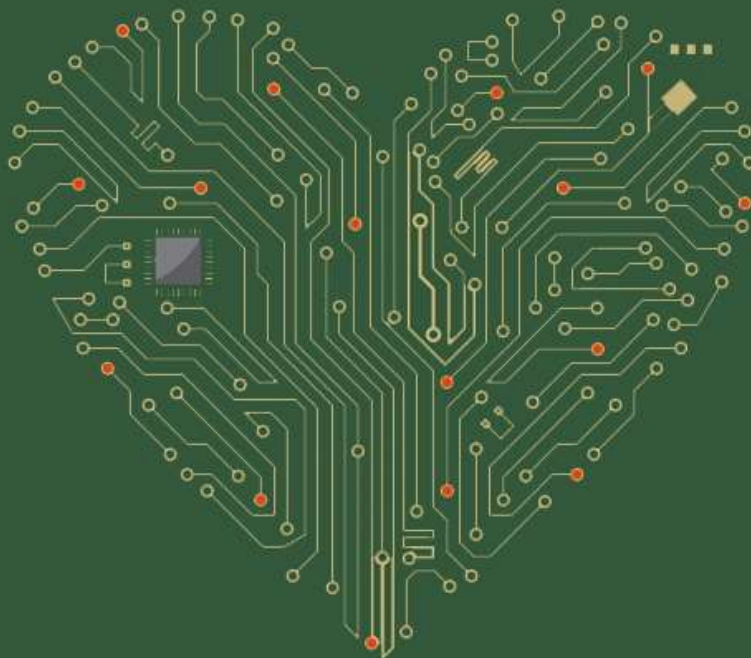
# Conclusiones

no existe ninguna receta mágica o vía rápida

Como hemos dicho no es ninguna metodología clara, Tanto **Infraestructura Ágil** y **DevOps** son todavía movimientos y cultura en constante construcción y definición, donde cada día se suman mas y mas practicas, comentamos una serie de mejoras que forman parte de esa cultura, 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 **Infraestructura Ágil**, **DevOps** y **DevSecOps**.





# Infraestructuras Agiles

El corazón de DevOps en IT OPS



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



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

