

# SysAdmin vs DevOps

Primeramente, en tengo que destacar los diferentes conceptos que conciernen a esta rivalidad. Rivalidad...? o son hermanos?, seran lo mismo pero diferente?, quien es mejor?; interrogantes que he de explicar en esta ocasión.

SysAdmin, como se suele intuir ADMINISTRADOR DE SISTEMAS, persona o conjunto coordinado que es encargado como la palabra lo dice “*administrar*”; en diversos textos y en escencia se **Sysadmin: El Arte de ser un Administrador de Sistemas y Servidores** define como: “”, tanto asi que el SysAdmin su principal caracterisiticas es su profesionalismo, su experiencia y su seguridad al ejecutar su trabajo, porque a lo largo de su carrera de estudio se ha desarrollado en su campo o especialidad; por ejemplo, programacion, pruebas, desarrollo, soporte, servidores, BD. Pero especializados en areas especificas, ante estos se levanto un enemigo. Enemigo?, no un sustito un nuevo modelo de gestión que supone a los empresarios ahorro de costos y lograr ver resultados en poco tiempo, DevOps.

DevOps, puedo decir en mi opinion que este modelo es hijo y sucesor del DESARROLLO AGIL, citando:

*“La mejor manera de definirlo en profundidad es utilizar un método paralelo a la definición de un término igualmente complejo, desarrollo ágil”*

Como dice el texto, la mejor manera de entender este concepto su ADN, entiendase que el objetivo del desarrollo agil es dar resultados en la mejor cantidad de tiempo posible, probarlos y si es necesarios volver a trabajar a en el, de modo que el cliente vea posibles resultado, explico esto a groso modo, porque si ahondo mas en el desarrollo agil surge una complejidad de técnicas, valores, diagramas, entre otras cosas.

Volviendo al sucesor del desarrollo agil, DevOps, va mucho mas alla de manera que su principal caracteristica no es el desarrollo agil, en mi opinion sino que a simple vista busca a personas no especializadas, con poca experiencia pero que realicen tareas segmentadas, creo la mejor manera de verlo es como un “*FORDISMO INFORMATICO*”, que busca reducir costos en la minimizacion de tareas. No obstante, la disparidad de esto es que tambien el personal que se mueve bajo esta metodología sea capaz en algun momento ser “Maestro de nada” y “Sirviente de todo”, al conocer lo anterior vemos “*todologos*” aludiendo a una expresion mas familiar.

Agregando a la idea de DevOps, “Un DevOps puede ser un SysAdmin, pero un SysAdmin no puede ser un DevOps”, aquí puedo decir, esta la parte medular de esta rivalidad, sin embargo, tienen mucho en comun. Se pueden ver en la siguiente lista:\

**Los Sysadmin tienden a abarcar las siguientes funciones y actividades:**

1. Implementar nuevos o remover obsoletos
2. Realizar respaldos
3. Monitorizar el rendimiento
4. Administrar los cambios de configuración

5. Operar las Aplicaciones y los Sistemas Operativos
6. Gestionar las cuentas de usuarios
7. Vigilar la seguridad informática
8. Afrontar los fallos y las caídas
9. Atender los requerimientos de los usuarios
10. Reportar a los niveles directos responsables de la Organización
11. Documentar las actividades informáticas del Sistema y la Plataforma

**Y ha tener ciertos conocimientos de:**

1. Programación
2. Bases de Datos
3. Seguridad Informática
4. Redes
5. Sistemas Operativos

**Los DevOps tienden a dominar una gran variedad de lenguajes de programación, además de poseer capacidades técnicas y habilidades de gestión. Un DevOps suele ser también una mezcla de Desarrollador de Software y Sysadmin** cuya función suele verse como una eliminación de las barreras entre ambos perfiles. Por lo que es de esperar que un DevOps posea conocimientos tanto del Software como del Hardware (Infraestructura/Plataforma) de la Organización donde se desempeñe.

**Por ende, los DevOps suelen ser capaces de:**

1. Escribir código y realizar la función de un Programador.
2. Administrar Servidores Multi-Plataformas y realizar la función de un SysAdmin.
3. Gestionar Redes y realizar la función del NetAdmin.
4. Manejar una base de datos (BD) y realizar la función de un DBA.

En los DevOps hay ciertas etapas

## **1- Gestión y planificación**

Todo proyecto necesita una visión que indique a los participantes -sean directos o indirectos- el motivo y fin último del trabajo a realizar; **definiendo un conjunto mínimo de funcionalidades que permitan aportar Valor en cada iteración**, los criterios de aceptación a cumplir y la definición de acabado; para cada una de las fases y en el conjunto del proyecto.

## **2- Desarrollo, construyendo código**

Esta fase es en donde se construye. Sea picando código, diseñando infraestructura, automatizando procesos, definiendo las pruebas o implantando la seguridad.

Es en donde, en la actualidad, se está realizando el esfuerzo más importante en la automatización de las acciones repetitivas o complejas; y que debiera ser uno de los primeros peldaños que escalar para implantar DevOps en una organización.

Si tuviera que **resumir en una sola palabra el concepto más importante de esta fase, esta sería “pruebas”**.

Ya sea en una aplicación de gestión, operaciones con datos o el despliegue de infraestructura virtual; siempre voy a trabajar en código – ya sea con un lenguaje de programación o de scripting; el cual debe ser almacenado en un gestor de código que me permita operaciones básicas como históricos, ramas, versionado, etc.

### **3-Integración continua, o cómo dormir tranquilo**

La integración continua es automatizar el mecanismo de revisión, validación, prueba y alertas del valor construido en las iteraciones, desde un punto de vista global.

Es decir, mi singular funcionalidad o característica, que he construido en mi entorno de desarrollo, junto con las pruebas automáticas que aseguran su correcto funcionamiento, **son publicadas en un servicio que la integra con el resto de la aplicación.**

### **4-Despliegue automatizado**

Desplegar, en las organizaciones clásicas, *siempre ha sido un dolor*. Dos roles (Dev e IT) con objetivos e intereses divergentes se encuentran en una batalla de incomunicación y recelo mutuo para publicar la aplicación en los diferentes entornos de trabajo: desarrollo, integración, calidad/test, preproducción, producción, etc.

Como en toda cadena, es fácil romper por el eslabón más débil, y **cuantos más pasos existan en los procesos de despliegue, más posibilidades de fallo humano se suman.**

### **5-Operaciones, velando por el buen funcionamiento**

Es una minoría las aplicaciones que son puestas en producción y no requieren de un trabajo constante en su optimización, evolución, o soporte. Pero, además, debo tener en cuenta todas las operaciones relacionadas con su funcionamiento que deben realizarse de forma continuada durante toda la vida del software.

En resumen, vemos que cada uno de estos modelos son altamente efectivo, pero que cada uno tiene diferentes beneficios, de lado del experimentado, especialista en alguna rama de la informática surge un descenso en su utilidad pero al empresario le es una reducción de costos al no contratar servicios especializados, sino a posible recién graduados que no tiene la suficiente experiencia para un campo específico. Definimos los conceptos de Devops y SysAdmin, por tanto, vierto mi opinión que según la investigación realizada puede llevar un momento donde se estableciera con mayor auge los DevOps.