

## Thorsten Prumbs

**Arquitecto de soluciones en atSistemas**

**Thorsten es un reputado profesional, con dilatada experiencia en el diseño de arquitecturas corporativas y la consultoría de seguridad y rendimiento de entornos distribuidos de alta disponibilidad, principalmente para el sector financiero y asegurador.**

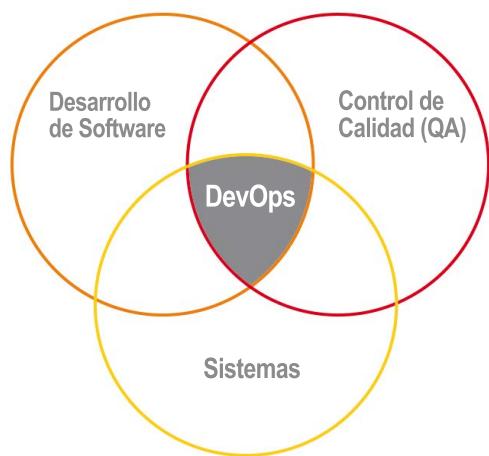
**“Cuanto más cerca se está de un software a nivel funcional y no funcional, más claramente se percibe el abismo existente entre los tres departamentos.”**

## ¿Qué es un DevOps?

### Introducción

DevOps es el acrónimo de Development & Operations, que en castellano podemos traducir como Desarrollo y Sistemas. Con este nombre, designamos un rol que se está introduciendo, poco a poco, en nuestro entorno tecnológico.

La persona o equipo que desarrolla este papel debe reunir conocimientos horizontales y una visión muy clara de los siguientes mundos tecnológicos:



En este sentido, un DevOps actúa como puente, como embajador y, a la vez, como evangelista entre los equipos de Desarrollo, Sistemas y Control de calidad.

### ¿Responde a una necesidad real o es una moda?

La respuesta a esta pregunta sería: depende de a cuál de los tres departamentos anteriores se le realiza la pregunta. Cuanto más cerca se está de la situación real de un software a nivel funcional y no funcional, más claramente se percibe el hueco, o más bien el abismo, existente entre los tres departamentos.

### Hablemos de la realidad

En la mayoría de las empresas que invierten en el desarrollo de productos software, la realidad es que los tres departamentos de Desarrollo, Calidad y Sistemas están separados y aislados el uno del otro. Con el avance de la externalización (Servicios de Operaciones, Software Factories), los miembros de cada departamento ni conocen a los integrantes de los demás departamentos, ni tienen una forma de comunicación directa con ellos.

El resultado es lo que se podría llamar una “guerra de trincheras”, donde la patata caliente se pasa de un lado a otro empleando técnicas propias de una guerra como la distracción, el engaño, el ataque y la defensa.

## ¿Qué es un DevOps?

**El DevOps es, por lo tanto, un embajador entre tres mundos que suelen estar demasiado separados en la realidad actual de las empresas.**

**En proyectos ágiles con continuas puestas en producción, se percibe mucho antes la necesidad de un DevOps.**

En este contexto, se dan pocos cambios con la metodología de calidad o de desarrollo que se emplee. La diferencia está en la rapidez con la que un usuario puede cuestionar o justificar con su experiencia la inversión ya hecha.

Con un modelo cascada, los problemas de comunicación y conocimiento entre los departamentos están camuflados por los problemas de desviación de la Triple Restricción: requisitos, fecha de entrega y presupuesto.

Con un modelo ágil, la retroalimentación del usuario se da mucho antes. Además, como los períodos de evolución son más cortos, los problemas de comunicación entre los departamentos se dan antes y con más frecuencia. El resultado es que todos los departamentos ven la necesidad de automatizar para eliminar fallos humanos y para poder llevar a cabo su trabajo con el tiempo y los recursos que tienen.

Al final, con las mejoras y evoluciones continuas, todos los departamentos se dan cuenta de los problemas de mantenimiento:

- **Desarrollo** tiene que reimplementar en vez de reutilizar.
- **Calidad** tiene que volver a probar todo porque no se pueden fiar de la estabilidad y funcionalidad del software.
- **Sistemas** tiene que aprender de nuevo, con casi cada pase a producción, cómo analizar trazas, cómo dimensionar, cómo monitorizar, etc.

Como consecuencia, los proyectos de desarrollo ágil emplean pocos recursos con un alto nivel de conocimiento y técnicas como TDD y Pair Programming.

Éstas son las razones por las que, en proyectos ágiles con continuas puestas en producción, se percibe mucho antes la necesidad de un DevOps.

### Hablemos de proyectos ágiles

La parte de gestión de requisitos con sus historias de usuario y su plan de releases ya acerca, por fuerza, dos partes: el usuario y el técnico que lo implementa.

Con la planificación a corto plazo dentro de un sprint, llegan las funcionalidades ya “modularizadas” y “pre-pensadas”. Por tanto, resulta más fácil y directo pensar en módulos funcionales y su integración.

Junto con las mencionadas técnicas de desarrollo, se implementan pruebas unitarias de cada módulo funcional y de cada integración. Si el desarrollo conlleva historias ya completas, se hacen pruebas de humo de la funcionalidad básica.

## ¿Qué es un DevOps?

Esta forma de desarrollo debe ir acompañada de un alto nivel de automatización de casi todas las fases del ciclo de vida: gestión de dependencias (Maven, Gradle), compilación, empaquetado, pruebas unitarias, despliegue, pruebas de humo, revisión estática de código, .... En este contexto, las pruebas funcionales de aceptación y las no funcionales se convierten en el cometido principal del Departamento de Calidad.

Como los despliegues y las pruebas de humo están automatizados, y se hicieron ya pruebas de rendimiento, robustez y seguridad (no funcional), el Departamento de Sistemas tampoco se encuentra sorpresas en el momento de dimensionar, monitorizar y operar en producción, ya que el mantenimiento y las evoluciones suelen ser pasos pequeños e incrementales. Como resultado tampoco hay sorpresas en el día a día.

El problema de los proyectos ágiles es que requieren un alto nivel de conocimiento y experiencia de todos los participantes. Si el sistema a construir tiene un alto nivel de complejidad (muchas integraciones a medida, tecnologías heterogéneas, etc.), la agilidad y mejora continua se ven afectadas por los problemas técnicos (Technical Debt – deuda técnica).

La Deuda técnica - no tanto a nivel funcional sino más a nivel no funcional y a medio/largo plazo - se mitiga con DevOps por su alto nivel de experiencia y su enfoque de entrega continua con sus implicaciones en los tres departamentos: Desarrollo, Calidad y Sistemas.

## ¿Cómo llegar a ser un buen DevOps?

Desde los primeros momentos de un proyecto, el DevOps 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 necesidades y posibilidades futuras.

Según el contexto de cada proyecto y empresa, el DevOps debe conocer de antemano la empresa y su forma de trabajar. Es decir, debe tener una idea muy clara y concreta de las formas de trabajo de cada uno de los tres departamentos: Desarrollo, Calidad y Sistemas. Durante todo el proyecto, es de gran ayuda si el DevOps sabe organizar y documentarse bien. Es decir, hay que tener una visión para los problemas y soluciones esenciales y no liarse con el “ruido” del trabajo o problemas diarios.

**Desde los primeros momentos de un proyecto, el DevOps se encarga de alinear el desarrollo con las necesidades funcionales, operativas y de calidad.**

## ¿Qué es un DevOps?

Pero todas las habilidades y conocimientos no son suficientes si el contexto organizativo - la empresa con sus departamentos - persigue intereses ocultos y opuestos. Es decir, con la organización formal no se consigue nada si no se tiene en cuenta la informal. La organización formal se suele desviar mucho más de la informal en contextos jerárquicos con un alto nivel de externalización que en organizaciones planas.

### ¿Cómo me pueden permitir ser un buen DevOps?

Como siempre, depende del contexto empresarial, organizativo y tecnológico pero, en líneas generales, un DevOps sólo puede convencer y alinear los tres departamentos (Desarrollo, Calidad y Sistemas) si los conoce.

El DevOps debe considerar imprescindible hacer “reuniones” informales con miembros claves de cada departamento. En general, esto se consigue con asignaciones directas al proyecto en curso de personas de cada departamento, con nombres y apellidos. Dicho de forma rápida: tomando un café con gente de cada departamento, un DevOps se entera de asuntos de igual o más importancia que leyendo las actas del PMO.

La información e impresión obtenida ayuda al DevOps a hacer preguntas de la forma más efectiva posible. Dicho de otra manera: con el conocimiento de los atajos y trucos habituales de un departamento, se relativizan las historias de éxito de los demás departamentos.

Sin embargo, el DevOps debe contrastar simultáneamente este conocimiento con la documentación existente, por un lado, y con el código más configuración, por el otro. En este sentido, todo conocimiento empieza y acaba en la parte de Sistemas. Sí, en la parte de Sistemas y no de Desarrollo, porque en un contexto organizativo difícil, un DevOps no se puede fiar de nadie y si no hay mecanismos automatizados, no se puede fiar de que el código fuente de Desarrollo o de Calidad coincida con el compilado en ejecución en un entorno concreto.

En este sentido, es de ayuda si el DevOps emplea técnicas de Ingeniería Inversa. Por ejemplo, puede tener que descompilar binarios obtenidos de Sistemas para poder demostrar un problema de código (es decir, de Desarrollo y a lo mejor también de Calidad). Esta información suele ayudar a la hora de establecer confianza en el conocimiento y la capacidad del DevOps. Cualquier análisis que se haga tiene que hacer énfasis en el problema, su impacto y su posible solución, pero no en su causa o responsable.

*Cualquier análisis que se haga tiene que hacer énfasis en el problema, su impacto y su posible solución, pero no en su causa o responsable.*

## ¿Qué es un DevOps?

El DevOps debe buscar siempre la mejora para todos los afectados (Usuario y Desarrollo -> funcional, Calidad -> verificación, Sistemas -> no funcional) sin escalar a los niveles de gestión porque suele destruir niveles de confianza ya obtenidas. Si, en un momento dado, el DevOps sabe dar una solución cuya mejora afecta a todos los tres departamentos, suele crear a la vez un clima de cooperación, aunque sea temporal. Esta cooperación hace hueco para aquellas mejoras del DevOps cuyo valor todavía no es percibido como necesario.

En términos generales, hace falta automatizar tareas como parte del proyecto donde no hay conocimiento o voluntad en uno o varios departamentos.

Además, suele ocurrir que el valor del trabajo de un DevOps se pone de manifiesto de forma indirecta a medio y largo plazo.

El DevOps es, por lo tanto, un embajador entre tres mundos que suelen estar demasiado separados en la realidad actual de las empresas, tanto si se le emplea como un servicio horizontal o como si sólo interviene en situaciones de crisis a modo de negociador.



atSistemas es una compañía de consultoría, servicios de IT y desarrollo de software. Con un crecimiento sostenido desde su fundación en 1994 y una plantilla que supera los 450 empleados, los servicios de atSistemas se caracterizan por la flexibilidad y la agilidad, lo que nos permite ayudar a grandes empresas de todos los sectores, aportando conocimiento y experiencia sobre el más amplio abanico de tecnologías. Nuestra cartera de clientes incluye más de 200 de las principales empresas del país, con representación de todos los sectores de actividad, a los que prestamos servicio desde nuestras oficinas de Madrid, Barcelona, Cádiz y Zaragoza.

Nuestro portfolio de servicios abarca desde el desarrollo de software a medida hasta la integración de grandes soluciones de software empresarial, en áreas que van desde la más compleja arquitectura de sistemas hasta las soluciones más novedosas de comercio electrónico o aplicaciones móviles.

Valle de Alcudia, 3  
28230 Las Rozas, Madrid  
+34 91 640 76 20

Passeig de Gràcia 55, 8º - 4<sup>a</sup>  
08007 Barcelona  
+34 93 209 66 61



[atsistemas.com](http://atsistemas.com)



[info@atsistemas.com](mailto:info@atsistemas.com)