

**ANDRÉS AZNAR LÓPEZ**

Subdirector de Sistemas y Tecnologías de la información y las Comunicaciones  
Comisión Nacional de los Mercados y la Competencia

## La Cosa que Nadie Más Conoce

---

El presente artículo expone algunas prácticas de integración continua en el desarrollo de software que actualmente se llevan a cabo en la Comisión Nacional de los Mercados y la Competencia. Este organismo público es joven, pues nació el 7 de octubre de 2013 gracias a la Ley 3/2013, y tiene unos 550 empleados propios repartidos en 3 sedes. Se trata de un organismo supervisor, apodado a veces por la prensa como “El Superregulador”, que tiene encomendada la tarea de velar por el funcionamiento eficiente de los mercados y por la existencia de una competencia efectiva, siempre en beneficio de los consumidores. También participa en el diseño y definición de las políticas de actividades económicas

**C**uando uno oye hablar de la factura de la luz, de si se ha sancionado a Telecinco debido a su programa Sálvame Naranja, de si se autoriza la adquisición de Eroski por parte de Carrefour, de dónde se consolidan los números de teléfono que se envían a las fuerzas de seguridad, del efecto lunes en los precios de la gasolina, de la multa de 5,64 millones de euros a cuatro empresas y a varios directivos por repartirse concursos públicos de ADIF, del intenso debate generado sobre la economía colaborativa (UBER y los taxistas entre otros), del primer registro público de lobbies en España ... sobre todas estas cuestiones se proyecta la sombra de la CNMC. No es el único ac-

tor que interviene en estas materias, existen otros que también participan como el Ministerio de Industria, Turismo y Comercio, o el Instituto de Cinematografía y de las Artes Escénicas (ICAA), por ejemplo.

### **Desarrollar software es difícil**

En realidad, desarrollar software es muy difícil, y hacerlo en la Administración es una tarea heroica pues se añaden ciertos ingredientes necesarios, derivados del principio de servicio público, transparencia, y del marco específico de contratación, que elevan la complejidad de la gestión de los proyectos. Desde siempre ha sido complicado: Ambient Consulting elaboró un White Paper en 2005 en

el que determinó que más del 80% de los proyectos software fracasan significativamente, ZDNet mostró en un estudio de 2009 que el éxito en el 68% de los proyectos es “improbable”, y el propio IEEE en su revista IEEE Spectrum publicó un artículo “Why Software Fails” en el que indica que entre el 5% y 15% de los proyectos IT será abandonado poco después de nacer por ser inadecuados.

Pero cuando un proyecto de desarrollo de software sale bien, aporta tanto valor...

Sinceramente, creo que desde hace unos años la cosa está cambiando, y de hecho que ahora estamos viviendo una pequeña revolución y edad dorada en el desarrollo de soft-

ware, gracias a nuevos paradigmas. El pasado 2 de agosto Bloomberg publicaba que por primera vez las cinco empresas más valiosas de la bolsa a nivel mundial son tecnológicas (y estadounidenses). Cuando hablamos de tecnología el recurso más valioso es el capital humano.

### **La magia sí existe**

Sólo comentar de forma preliminar que es el cuerpo TIC el que tiene en sus manos la capacidad de elevar a la Administración a un nivel superior, de convertirla en “otra cosa”. Hay usuarios que califican como magia cosas incomprensibles para ellos, pero que detrás han sido levantadas con recursos muy diferentes de la magia: esfuerzo, dedicación, tecnologías modernas, paciencia y, en muchas ocasiones, buenas dosis de trabajos manuales y pesados. Un TIC es un gestor que tiene un importante “bonus”, a saber, la tecnología.

La tecnología define drásticamente la capacidad de una administración, al igual que el disponer de aviones define drásticamente la capacidad de un ejército. Y esta tecnología se materializa en su última expresión a través del software que se encuentra disponible para los ciudadanos, del que utilizan los empleados públicos y, por supuesto, de la infraestructura tecnológica necesaria.

### **Póngame con el encargo**

Hace pocos años surgió el término “Devop”, como una simbiosis de “Developer” y “Operator” para referirse a los programadores que además de los conocimientos necesarios para desarrollar software, también tienen una buena capacidad para montar y configurar los servidores, bases de datos, almacenamiento y redes sobre los que se ejecutan sus aplicaciones. Esta polivalencia dota a estas personas de una visión global que permite entender las consecuencias de



las líneas de código que escriben en términos de rendimiento, seguridad, arquitectura de la aplicación y problemas derivados de los entornos y servidores. Por poner algunos ejemplos, permite saber por qué hay que vigilar el uso de variables de sesión cuando se utiliza balanceo de carga en servidores web, por qué un índice en cierta tabla de base de datos puede cambiar radicalmente la velocidad de tu aplicación, por qué una sentencia SQL no se debe construir concatenando strings, o por qué hay que erradicar el HTTP en favor del HTTPS.

Hay buenas aplicaciones que han sido construidas por unas pocas personas, incluso a veces por una sola, lo

cual pone de manifiesto que los buenos programadores suelen valer su peso en oro... pero no de forma clara en España. Este junio se publicaba en Xataka el artículo “Soy programador y tengo 50 años”, que exponía, citando un estudio de la Universidad de Carolina del Norte basado en el análisis de StackOverflow.com (4,7 millones de programadores), que se es mejor programador conforme se cumplen años, en contra de lo que a veces se piensa. En EE.UU. los buenos devops están muy cotizados, no siendo raro que perciban sueldos superiores a los de sus jefes, y es una tendencia que va a más, pero sin embargo en España existe una cultura en la que

el programador es considerado el escalón más bajo, e incluso a veces se piensa que pueden ser sustituidos o intercambiados fácilmente.

Intentamos que no sea así en la CNMC, donde se está impulsando una organización que otorga al programador la capacidad de iniciar un despliegue a producción, con objeto de que pueda controlar la aplicación en la que trabaja en todos sus extremos. No existen autorizaciones previas para un miembro permanente del equipo de una aplicación, lo cual no significa en absoluto que esté solo. De hecho, esto hace que su trabajo deba ser más colaborativo que nunca pues el compromiso y la responsabilidad son muy elevados. De este modo, para alcanzar los objetivos tiene a su disposición a mucha gente: a sus compañeros en la aplicación, a los de sistemas, a la Oficina de Calidad, a los jefes. Este planteamiento implica que el programador debe estar cerca de estos colaboradores necesarios. El modelo de software factory puede resultar interesante en ciertos casos, pero ha de valorarse concienzudamente y sólo bajo ciertas circunstancias. En la CNMC se ha utilizado en varios proyectos este año, de forma controlada.

## **A producción, cuesta abajo y sin frenos**

En realidad, ese es el objetivo: que subir a producción sea tan normal y tan rápido que resulte tremendamente aburrido. Ningún factor bloquea una subida a producción por defecto, sino que es necesario realizar un esfuerzo para pararla, en lugar de hacer un esfuerzo para hacerla. Por supuesto las subidas se pueden parar, pero para ello hay que solicitar de forma expresa que no se lleve a cabo. Es importante recalcar que este modelo se aplica en la CNMC con aplicaciones web donde no existe el concepto de “versión”, ni de “releases”, “alpha” o

conceptos parecidos, sino que la aplicación siempre está viva y siempre se está usando, evolucionando con microcambios permanentes.

Una consecuencia muy interesante de esta forma de actuar es el concepto de “hecho”. Cerrar una tarea en Jira (herramienta que usamos para la gestión de tareas) significa que como muy tarde, al día siguiente estará en producción y funcionando, entendiéndose que ha sido probada adecuadamente.

Pero “sin frenos” no significa sin control. Cada letra que alguien escribe en el código fuente, o cada cambio en la configuración de base de datos, queda versionado con trazabilidad absoluta (en la CNMC utilizamos GIT). Cada subida puede ser notificada de forma automática a todo el equipo, y en general cada movimiento de cierta importancia. Además, existen muchas herramientas para analizar y supervisar la calidad del código, el linting, los posibles agujeros de seguridad, la accesibilidad, el rendimiento, la usabilidad, los code smells o estilos de programación, los test unitarios y otros tipos de test, por ejemplo. Estas herramientas son muy didácticas y cuando llevan un tiempo implantadas uno acaba preguntándose cómo se podría vivir sin ellas. Durante 2016 en la CNMC hemos puesto en marcha varios puntos de control, entre los cuales se encuentra por ejemplo la herramienta de análisis estático de código fuente SonarQube y estamos en proceso de implantar Asqatasun, con el fin de afrontar el desafío organizativo de alcanzar el grado deseado de supervisión y control sin generar embudos.

El pasado 2 de junio se celebró en Madrid el evento de Amazon AWS Summit, en el cual nos resultó grato descubrir que su modelo de desarrollo estratégico va en esta misma dirección. Se facilitaron algunos datos sorprendentes, como por ejemplo

que contando todos sus equipos de desarrollo, efectúan unos 50 millones de pases a producción cada año, o que el día que incorporan a un nuevo miembro tras contratarlo intentan que suba algo a producción para que vaya acostumbrándose. Hay que decir que trabajan con un enfoque de microsistemas, de forma que hay un equipo de desarrollo independiente para cada pequeña parte de su web. Estas cifras significan que, cuando alguien entra a comprar un reloj en su tienda online, durante los escasos minutos que pasan desde que se elige el producto hasta que se compra resulta casi imposible que no se haya producido una subida a producción en alguno de sus sistemas.

## **Excepción no manejada**

Para que el sistema de subidas a producción continuas funcione, es fundamental diseñar una gestión de errores adecuada. El equipo que introduce nuevo código en producción ha de saber qué está ocurriendo en todo momento. El objetivo es que todos aquellos involucrados en el desarrollo de la aplicación conozcan en tiempo real cualquier excepción no manejada (las conocidas “unmanaged exception”) que pueda surgir, lo cual puede convertirse en un infierno si se tienen muchos fallos, pero hay que aspirar a no tenerlos.

Existen varias alternativas para alcanzar esta meta. En la CNMC se utiliza un método bastante simple que consiste en enviar un e-mail cada vez que se produce un error, pero existen otras formas de conseguirlo, por ejemplo mediante Slack, una herramienta interesante.

En todo caso, la clave para solucionar problemas es un llevar a cabo un ejercicio permanente de proactividad. Al preguntar hace dos años su opinión a una usuaria que llevaba unos meses usando el gestor de expedientes de la CNMC, respondió que

por encima de todo había una cosa que le había sorprendido positivamente: en cierta ocasión la aplicación le dio un error al que no prestó mayor importancia, y volvió a entrar con el navegador para seguir trabajando. A los pocos minutos sonó su teléfono y alguien de la subdirección TIC le preguntó qué texto había introducido exactamente en el formulario X del expediente E, indicándole que eran conscientes de que había fallado y que en breve estaría solucionado. Esto es práctica habitual y rutinaria desde hace años en una aplicación concreta, consiguiendo que los usuarios tengan una sensación positiva de protección y vigilancia.

### **Producción es mi prototipo**

Tal es el énfasis puesto en conseguir que cualquier desarrollo cuanto antes en producción y en evitar la acumulación de cambios, que incluso el código de las funcionalidades a medio desarrollar ya se sube a producción, aunque éstas se mantienen ocultas para los usuarios. Las fronteras entre desarrollo, preproducción y producción son muy livianas, no hay prácticamente diferencias, tanto a nivel de código fuente como de estructura de base de datos (tablas, columnas, índices... en ningún caso se refiere a los datos), lo cual produce una gran relajación.

El día que se decide subir una nueva y rimbombante funcionalidad a producción, suele ser un día tranquilo, pues en realidad lo único que se hace es abrirla a todos los usuarios, tras llevar meses en producción funcionando.

### **Pues los he visto más rápidos...**

Cierto, ¿has oído hablar de los “contenedores”? Docker, Vagrant, Kubernetes, Proxmox... Actualmente, subir a producción en la CNMC consiste en desplegar nuevo código en un servidor existente, pero en un futuro po-

dría consistir en crear en unos pocos segundos y mediante configuración, un nuevo servidor totalmente operativo y limpio desde cero. Bueno, un servidor no, un contenedor. Y después esperar a que el anterior finalice su última llamada HTTPS para destruirlo mientras el nuevo ya está dando servicio.

En el repositorio de código fuente, además del código y de la estructura de base de datos, se incluirían y versionarían también los ficheros de configuración relativos a la configuración de los servidores.

### **El cuadrado mágico de Luxemburgo**

En 2004 el Real Madrid fichaba a un nuevo entrenador (para variar), un brasileño llamado Vanderlei Luxemburgo que al poco de llegar ya predicaba las maravillas de su sistema táctico: el Cuadrado Mágico en el centro del campo. Dicha estrategia tenía grandes virtudes y solucionaría todos los problemas de cualquier equipo, pero quizás Luxemburgo se estaba precipitando... ¿acaso había analizado la plantilla que tenía a su disposición, para valorar si las personas encajaban con las características necesarias para ejecutar su mágico cuadrado?

Esta pequeña historia sirve para introducir una aclaración: no existe el sistema perfecto de forma general, sino que todo sistema depende de las características del entorno, de la unidad y de las personas que han de llevarlo a cabo.

Por otro lado, se encuentra la gestión en lo referente a la contratación. En la Administración Pública estamos sujetos a unas reglas del juego que han de garantizar oportunidades a todos los competidores con transparencia. Esto hace que uno deba asumir desde el principio que el personal externo de apoyo, los programadores, van a cambiar en algún momento.

El conocimiento de las aplicaciones debería permanecer y residir en la subdirección. Para ello, lo deseable es que los proyectos de desarrollo cuenten con un porcentaje de personal propio, de forma que no sean totalmente dependientes de la contratación, que el know-how se mantenga en la administración, y que además se controle y se conozca lo que de verdad está ocurriendo en el equipo.

### **Conclusión**

Enfocar un sistema de trabajo en el que las subidas a producción de cada aplicación se hagan todos los días o incluso en horas, reduciendo al máximo el tiempo que transcurre desde que se toca el código fuente hasta que ese código está ejecutándose en producción, tiene ventajas muy interesantes no solo desde el punto de vista del usuario sino también para el equipo de desarrollo.

Para ello, es muy recomendable dotar a los desarrolladores de la capacidad de controlar los sistemas, y potenciar un entorno en el que esos desarrolladores los conozcan y manejen.

Un sistema de autorizaciones o de compartimentaciones suele perjudicar este enfoque, pues la idea es que cuando sea necesario que algo esté en producción inmediatamente, un miembro del equipo de desarrollo pueda conseguirlo por sus propios medios, y que de hecho sea tan rutinario que le resulte soporífero. Pero no hay que olvidar que existen todo tipo de controles y herramientas muy adecuadas y normalmente automatizadas para supervisar el desarrollo, realizar los despliegues, garantizar la calidad, y asegurar la trazabilidad de cualquier modificación o cambio que se suba.

Todo despejado en esta vía libre a producción. \*