

"Cómo un banco europeo adopta el código abierto" por Ben Rometsch

Este artículo en específico se rige por la licencia **CC-BY** que fue asignada por su correspondiente(s) autor(es) o autora(s). Dicha licencia permite así la propagación de este material y su traducción a cualquier idioma. Usted podrá encontrar un resumen de tal licencia en [este enlace \(en idioma castellano\)](#) y la licencia en sí misma de forma completa y totalmente traducida al español en [este otro enlace](#).

This article is under [License Creative Commons Attribution 4.0 International](#).

Traducción del artículo publicado en OpenSource.Com titulado "How one European bank embraces open source" escrito por Ben Rometsch:

<https://opensource.com/article/22/7/european-bank-open-source>

Resumen:

Entrevista con el Jefe de Desarrollo y Jefe de Ingenieros de "Komer?ní banka" donde revela cómo están aprovechando las tecnologías de código abierto.

Me senté a hablar sobre código abierto y desarrollo con Jindrich Kubat, Jefe de Desarrollo y Jefe de Ingenieros (COE) en *Komer?ní banka* (**KB**) en la República Checa.

Ben Rometsch: ¿Cuál es su función oficial en **KB**?

Jindrich Kubat: Mi función oficial es Jefe de Desarrollo, COE (centro de experiencia) en *Komer?ní banka*.

Ben: ¿Cuál es exactamente su área de responsabilidad y su día a día?

Jindrich: Como líder del COE de desarrollo, considero la experiencia, la estructura y las capacidades necesarias para que el desarrollo de software sea exitoso entre cientos de desarrolladores. Tenemos aproximadamente 600 personas en el equipo de desarrollo más amplio, pero no soy directamente responsable de administrarlas. Soy responsable de definir cómo harán su trabajo. Entonces, para mejores prácticas de desarrollo, unificación de estándares, etcétera.

Ben : ¿Cuál es su enfoque principal hoy en **KB**?

Jindrich: Nuestro proyecto de transformación. Tenemos el banco actual, que se basa en sistemas bancarios heredados típicos que son enormes sistemas de arquitectura monolítica distribuida. Ahí es donde la mayoría de nuestros desarrolladores (alrededor de 400) trabajan día a día, ya que genera la mayor parte de los ingresos del banco.

Junto con las aplicaciones heredadas, decidimos modernizarnos por completo a algo que llamamos "*Digital Hub*" que está reemplazando la infraestructura actual del banco. Las aplicaciones *front-end* son aplicaciones móviles y web nativas.

El *Digital Hub* y las nuevas aplicaciones *front-end* se basan completamente en microservicios, por lo que tenemos que rediseñar muchas cosas e introducir nuevas pautas para las personas. Todo esto en conjunto es lo que llamamos el "Nuevo Banco Digital". ¡Es un proyecto enorme!

Ben: ¿Esta transformación se detiene en algún momento?

Jindrich: Pensamos en la transformación en dos partes principales:

1. **La iniciativa** "El nuevo banco digital" está prevista para 5 años y ya llevamos unos 2 años y medio. Comenzó pequeño con unos pocos equipos que estaban listos para adoptar los enfoques de desarrollo más modernos y está creciendo cada año. Hoy tenemos alrededor de 180 ingenieros divididos en 50 equipos. Esos equipos han sido contratados externamente o transferidos desde los sistemas heredados. Están dedicados al 100% al esfuerzo de "*The New Bank*".
2. **El otro lado es la transformación cultural.** Llamamos a esto "Agile 2.0" que afecta a todo el equipo (tanto los sistemas heredados como los equipos de sistemas nuevos). Adoptar *Agile* y *DevOps* se trata de adoptar una nueva forma de trabajar. Este cambio es algo que también estamos impulsando con nuestros equipos de sistemas heredados. Si bien el edificio llegará a su fin en algún momento, la esperanza es que la nueva cultura se arraigue y sea la nueva forma de trabajar en **KB**.

Ben: Más allá de (la metodología) ágil, ¿han adoptado algún enfoque nuevo?

Jindrich: Sí, hay tres nuevos enfoques que adoptamos:

1. Pasar a [OKR](#) que se alinea bien con *Agile*.
2. Adoptamos una aplicación de chat colaborativo para todas las reuniones, lo que ha ayudado con el trabajo remoto.
3. Tenemos "tribus" funcionales que contienen personas y habilidades que cada tribu necesita para desarrollar y ejecutar sus propios proyectos o sus propias aplicaciones. Esto se alinea bien con la arquitectura de microservicios hacia la que nos estamos moviendo.

Ben: ¿Qué pasa con las herramientas, los idiomas y la infraestructura? ¿Cómo manejas eso entre tantas personas y equipos?

Jindrich: Como COE, somos los principales responsables de la unificación de cómo trabaja la gente. Somos bastante rigurosos en esto. Desarrollamos nuestro propio marco **KB**, al que llamamos *Speed*, que se divide en tres partes:

1. El primero es un [SDK](#) de Java®, creado sobre *Spring Boot*, que es un marco para crear aplicaciones en la nube.
2. Luego, las canalizaciones de [CI](#) y CD se utilizan para automatizar la implementación. Esto se basa en [Jenkins](#) y [Argo CD](#).
3. [Kubernetes](#) se usa donde implementamos y ejecutamos nuestras aplicaciones.

Construimos nuestra propia infraestructura en la nube, por lo que todo debe estar en las instalaciones hoy. Si un equipo necesita algo, debe pedirlo y mi equipo lo revisa. Si tenemos algo que resuelve su problema, entonces les pedimos que lo usen. Si no, estoy abierto a adoptar una nueva tecnología.

Comenzamos con [Speedy Framework](#) hace dos o tres años y todavía está evolucionando. La primera versión fue construida para aplicaciones monolíticas y obras monolíticas. Estamos en la versión 4 ahora. El enfoque de microservicio se implementó en la versión 3.

La desventaja es que tenemos que seguir algunas actualizaciones de productos y ciclos de vida, lo que es realmente doloroso para nosotros. Kubernetes, por ejemplo, tiene tres versiones principales por año.

En el futuro, queremos hacerlo más independiente de estos ciclos y automatizar las actualizaciones en el futuro. Curiosamente, nuestra empresa matriz adoptó [OpenShift](#), que estamos vigilando. Veremos si unificamos los marcos o no en el futuro.

Ben: Parece que está utilizando muchas tecnologías de código abierto en la nueva plataforma. ¿Siempre han hecho eso?

Jindrich Kubat: No, usamos muchos de los productos típicos como Oracle® y Sun Microsystems®. Para el nuevo banco digital, decidimos adoptar más tecnologías de código abierto como *Kafka* y *Flask*. Esto es genial porque son gratuitos, pero somos responsables de mantenerlo en funcionamiento. Esta es una mentalidad completamente nueva para los equipos.

Ben: ¿Tuvo que luchar para usar esos productos internamente?

Jindrich: No, simplemente lo aceptaron porque conocen las razones. Si les mostramos el contrato con Oracle, lo caro que es, hay un gran argumento comercial para adoptar el código abierto.

Incluso si tiene que tener algunas personas más para administrar la infraestructura, tiene un impacto positivo en nuestros costos.

Si les mostramos el contrato con Oracle®, lo caro que es, hay un gran argumento comercial para adoptar el código abierto.

Jindrich Kubat, Jefe de Desarrollo y Jefe de Ingenieros (COE) en *Komer?ní banka (KB)* en la República Checa.

Ben : ¿Qué pasa con las banderas de características? ¿Era algo que sabías que necesitabas antes de empezar?

Jindrich : Esa es una buena pregunta porque, ya sabes, incluso en los bancos actuales (sistemas heredados), las personas usan indicadores de funciones, pero han desarrollado sus propios sistemas. Son sistemas muy sencillos que gestionan el archivo de configuración. Y fue solo porque el sistema heredado solo se implementó 3 veces al año. Entonces puedes imaginar lo difícil que fue cumplir todas las promesas de otros equipos. Así que en realidad solo estaban allí para mantener la gestión y las pruebas de versiones empresariales a tiempo y coordinadas. Fue fácil activar o desactivar algunas funciones que no estaban listas para su implementación.

Pero para el Nuevo Banco Digital tenemos microservicios e integración continua. Realizamos implementaciones a diario en entornos que no son de producción. Curiosamente, solo tenemos dos entornos que no son de producción. Uno de ellos es solo para pruebas durante la compilación y el otro es el entorno final que no es de producción (puesta en escena).

Ben : ¿Fue una decisión consciente tener menos ambientes?

Jindrich : Sí, este es un enfoque que traje al banco donde rediseñamos por completo algunos entornos y cómo los implementamos. El objetivo es llegar a la producción lo más rápido posible para mantener la producción y la no producción lo más cerca posible.

Ben : El "santo grial" es tener un solo entorno: producción. ¿Seguramente eso no es posible en su negocio?

Jindrich : Tengo algo de experiencia con eso, donde simplemente pruebas todo en producción. Pero en la banca, eso es imposible. Muchos de nuestros desarrolladores ni siquiera pueden acceder al entorno de producción. Hay regulaciones. Por ejemplo, una regulación dice que no puede tener una cuenta de prueba en producción porque está operando con dinero real. Todo debe ser "real" en la producción.

Ben : ¿Qué haces para mitigar eso?

Jindrich : Tiene la menor cantidad posible de entornos que no son de producción.

Ben : Cuando estaba seleccionando un sistema de marcado de funciones, ¿consideró los servicios SaaS o fue algo que descartó de inmediato?

Jindrich : Sí, decidimos evaluar tres sistemas. [Flagsmith](#) , [LaunchDarkly](#) y construyéndolo nosotros mismos. Hicimos tres “pruebas de conceptos”. Primero fue con LaunchDarkly, luego Flagsmith y luego analizamos nuestro propio sistema construido en casa. Nos decidimos por Flagsmith no solo por la flexibilidad del sistema, sino también por el gran soporte. El hecho de que ustedes sean de código abierto y la excelente documentación también ayudaron en nuestra decisión.

Ben : ¿Cómo van las cosas hoy con las banderas de características?

Jindrich : Hemos estado usando indicadores de funciones durante un año en nuestros entornos que no son de producción, lo que ha funcionado muy bien. En cuanto al entorno de producción, tomó un tiempo desde una perspectiva de seguridad implementar indicadores de características en nuestro entorno de producción. La razón por la que tomó tanto tiempo es porque la persona originalmente responsable de traer indicadores de características a KB se fue. Luego, el equipo de seguridad cambió sus criterios de evaluación. ¡Finalmente, en la tercera vez, lo pusimos en producción después de pasar todas las pruebas de penetración y los requisitos!

Ben : ¡Hacer cambios en un banco no es para los débiles de corazón!

Jindrich : Sí, debido a los estándares de seguridad, toda la industria bancaria es muy sensible a esto. Ahora estamos en funcionamiento, centrándonos en los modelos de permisos y registrando todo para que sea auditable.

Ben : ¿Ha cambiado tu flujo de trabajo?

Jindrich : ¡Sí! Decidimos comenzar con la interfaz, especialmente los equipos móviles. Las personas han estado esperando esta capacidad porque han usado indicadores de funciones en funciones anteriores. Tenemos gente en el equipo presionando para adoptar porque sin un sistema de indicadores de características, todo tiene que esperar hasta que esté completamente terminado. Ahora pueden implementarse en producción y más equipos se están incorporando a The New Bank. Para aquellos equipos en una configuración de microservicio, es crucial que los equipos sean más independientes.

Hasta el momento, tenemos alrededor de 85 ingenieros trabajando con Flagsmith en la producción. ¡Eso ya es casi la mitad de The New Digital Bank!

Ben : ¿Estás usando banderas en el lado del servidor también o solo en la interfaz?

Jindrich : Hasta ahora solo en el frontend, pero nos estamos preparando para implementarlo con

KS7000+WP

KS7000 migra a GNU/Linux y escoge a WordPress para registrar el camino.

<https://www.ks7000.net.ve>

los sistemas backend que aprovechan los microservicios. ¡Eso será muy pronto!

Ben : ¿Hay alguna forma en la que quieras usar Flagsmith en el futuro?

Jindrich : Sí. Estamos buscando integrar la API de Flagsmith en nuestro rastreador de errores, poder impulsar las cosas y mantener a los equipos de control de calidad alineados con los propietarios de productos.