

Desafío Spike
Jorge Valencia G.
18 de abril de 2021

Proyecto Rose

every rose has its thorn

Primero que todo he decidido llamar al proyecto con el nombre código Rose, ya que dentro del mundo tecnológico, los productos no siempre comienzan con un nombre desde el inicio. Si bien puede que no corresponda al nombre comercial, al asignar un nombre clave, este tiene un impacto en el compromiso del equipo y eso es algo deseable en cualquier compañía.

Respuestas cuestionario

1. En mi opinión para el prototipo funcional necesitamos de al menos:
 - 1.1. Aplicación y sitio web, ideal contar con ellos y no duplicar los esfuerzos, hay numerosos frameworks que permiten portabilidad de código desde un ambiente a otro: React Native, Ionic, Flutter.
 - 1.2. Un backend de autenticación, autorización y contabilidad (AAA)
 - 1.3. Un backend para procesamiento de las operaciones GIS
 - 1.4. Un backend encargado de las operaciones CRUD
 - 1.5. Aplicación de versionamiento (Github/Gitlab, etc.)
 - 1.6. Componente CI/CD, Jenkins o Argo.
 - 1.7. Herramienta de comunicación con el equipo, Slack o Discord.
 - 1.8. Servicios Cloud

2. Lo conveniente es utilizar un entorno de microservicios, economizando en infraestructura. Es decir, aprovechemos las ventajas de los servicios cloud en la rapidez que podemos tener el producto (“time to market”), y cambien el coste inicial en ellos. No podemos esperar tener un producto finalizado desde los primeros meses, ni menos retornos de inversión el primero año sin invertir al menos el 50% del presupuesto en publicidad, entonces ahorrémonos estos costos fijos y enfoquemos esfuerzos en la venta.

3. Suponiendo que es un equipo nuevo y que no ha trabajado anteriormente en otros proyectos, recomendaría una metodología agile, pero híbrida, al menos durante las fases iniciales en que se construirá el prototipo. La metodología que propongo es Scrumban, teniendo solo las daily meeting y reuniones de retrospectiva semanal del mundo Scrum, con un fuerte manejo vía kanban de los avances de historias. Idealmente contar con métricas desde un inicio para identificar los puntos de bloqueo y que el líder técnico pueda destrabar a tiempo. A medida que el equipo en si se afiance es bueno que estas decisiones vayan pasando de a poco al equipo y lograr que se involucren en la toma de decisiones.

4. Debemos plantearnos si abordaremos completamente DevOps o si además agregaremos de inmediato el componente de seguridad DevSecOps. Lo segundo, irremediabilmente hará más largas cada una de las fases, pero es un riesgo no invertir en seguridad. Así que mientras determinamos cuanto riesgo queremos correr, una aproximación para comenzar sería ir agregando las capas de seguridad en forma paulatina, para no ir retrasando el avance del proyecto. Así que al menos el flujo GIT contaría de 2 ramas:

4.1.Rama de desarrollo, usualmente dev. Asociada a infraestructura de testing.

4.2.Rama de producción (main / master). Asociada a infraestructura productiva.

4.3.El uso de tags sería aplicado para el versionamiento, pero además para detonar el lanzamiento de revisión de código en forma estática por aplicaciones que busquen vulnerabilidades de seguridad como lo son SonarQube o Veracode.

5. Inicialmente el equipo parece completo, pero es porque el líder además de ese rol toma el rol de desarrollador backend, si eso no ocurre estamos en aprietos y se debe agregar al menos 1 desarrollador backend. Al Impact lead le pondría a trabajar con el Data Scientist y

el ingeniero ML, con el objeto de generar las directrices del proyecto. Primero en reuniones creativas para encontrar aquellas ideas que son realizables tanto económicamente como tecnológica en esta fase del proyecto, el prototipo. Y luego de finalizado el prototipo, siguen trabajando en conjunto en las ideas que fueron plasmadas en este desarrollo, con objeto de medir adopción de la aplicación y que posibles mejoras son necesarias para un crecimiento de ella. Dependiendo de estas ideas, del crecimiento de la aplicación y del feedback de usuarios, recomendaría crecer en 1 desarrollador backend y 1 frontend. Sería ideal construir una métrica de equivalencia para conocer el número exacto, ejemplo, cada 10000 descargas, cada 1000 usos, cada 10 feedback negativos, etc.

6. Como lo mencioné anteriormente, debería adoptarse un enfoque full DevSecOps y también fortalecer la infraestructura haciéndola tolerante ante los fallos que puedan presentar los proveedores cloud o los productos que estemos contratando ellos. Por el lado del desarrollo, debemos ser capaces de crear inteligencia de negocio dentro del mismo equipo, procesos de capacitación y formación de nuevos líderes para que en la medida que crezcamos seamos capaces de manejar mayor cantidad de células. Para esta altura, si anteriormente pudimos ser menos agile, ahora lo debemos ser 100%, de lo contrario la competencia nos ahogará. También creo esencial contar con un buen departamento de postventa que recoja el sentimiento de los usuarios, una práctica que recomiendo es que los desarrolladores trabajen en esta área para conocer de primera mano las sensaciones, en mi opinión ayuda y da una visión que el desarrollador no tiene generalmente.