

Cuéntanos qué piezas de software crees que sea necesario desarrollar para el prototipo funcional y cómo se relacionan estas. Llamamos pieza de software a cada aplicación (web, móvil o de escritorio), cada API, cada proceso batch que se puede desplegar de forma independiente. Apóyate con un diagrama si crees necesario.

R: Como prototipo funcional partiría por conversar con el Impact Lead para entender el valor que le queremos entregar al usuario, para poder tener una priorización en cuanto a valor. Si ir diseñando un prototipo que pueda entregar el mayor valor con menor esfuerzo (es decir lo antes posible) para luego ir iterando hasta obtener un producto maduro. Como equipo nos apoyaríamos en la persona de DevDesign también para tener una idea de las pantallas que podría tener este prototipo y con la ayuda de nuestra persona de Datos y Machine Learning quizás podríamos revisar dentro de la compañía o si hay investigación de por ejemplo cuál de las 3 estrategias tomar para el prototipo (si hacemos una app web, mobile o de escritorio)

Para efectos de poder responder vamos a asumir que nos decidimos por una app mobile comenzando por iOS, que tiene distintos tipos de usuarios con roles y pantallas específicas que podrán o no compartir ciertos componentes. Estos roles serían el cliente, driver, tienda y administrador. La arquitectura sería de microservicios ya que podemos tener distintos servicios que cumplan un objetivo particular para así tener distintas estrategias para cada feature.

Diseño para prototipo:
App mobile para iOS

Backend:
Una API Orquestadora que maneje los casos de uso.
Una API de usuarios
Una API de pedidos
Una API de búsqueda
Una API de geolocalización
Una API de pagos

Frontend:
Diseño de las pantallas para los distintos roles con un login y registro de cuenta, también:
A) Cliente: ingreso de dirección, listado de tiendas donde elegir, productos de la tienda, pago y seguimiento del pedido
B) Driver: Solicitud de un driver para aceptar o rechazar, seguimiento hacia la tienda, seguimiento hacia el cliente
C) Tienda: ingresar los productos y sus valores, pagina de pedidos para aceptar o rechazar, driver que va a buscar el pedido y cambio de estado a entregado
D) Administrador: ingresar nuevos roles

Infraestructura:
Una base de datos (Me inclinaría por una base de datos relacional)
Asumiría que vamos utilizar la nube para hostear por su facilidad en la escalabilidad para cuando vayamos ampliando nuestro target.

Transversal:

Métricas para poder tomar decisiones de los siguientes pasos

También habría que tener algún plan de soporte dentro de la aplicación con un chatbot (verlo con Machine Learning)

Mejoras al prototipo:

Lo ideal sería hacer cada componente lo más simple posible para el prototipo e ir iterando desde ahí.

Consultaría con Machine learning algún algoritmo para la prioridad del listado de las tiendas cerca del cliente, para entregarle la mejor experiencia

Consultaría al Cientista de datos que analizaremos las métricas para ver de agregar distintos tipos de cache (por ejemplo desde la aplicación las distancias de las tiendas más frecuentes de un cliente)

Algunos casos de uso evaluaría hacerlos asíncronos, como por ejemplo la actualización de productos de las tiendas, utilizar alguna cola para que esa información se vaya actualizando sin tener que esperar el resultado en línea.

En cuanto al equipo sería ideal contar con al menos un experto en iOS y otro en APIs

Cuéntanos sobre el tipo de arquitectura que elegiste para la pregunta (1). ¿Monolítica? ¿Micro-servicios? ¿Algún intermedio? ¿Otra? Comenta en qué te basaste para tomar esta decisión

R: Utilizaría distintos micro-servicios dentro de las APIs a utilizar de esta manera la escalabilidad y disponibilidad de las distintas feature de nuestra aplicación no estén acopladas, esto mejoraría la mantenibilidad también de la aplicación.

Describe la metodología de trabajo que usarías para el desarrollo. Puede ser alguna metodología conocida (Scrum, XP, RUP), una adaptación, o una mezcla entre varias metodologías. Lo que sea que tu experiencia te haya mostrado que funciona. Cuéntanos por qué crees que esta forma es adecuada para nuestro problema.

R: Utilizaría alguna metodología agile, dado a que es un proyecto que parte de cero me inclinaría por hacer SCRUM con un ciclo de desarrollo de 2 semanas entendiendo que esto debería ser acordado con todo el equipo y sus roles pero generalmente he trabajado de esta forma y así ir iterando y entregando valor cada 2 semanas
Ya cuando la aplicación esté madura podría ser una mejor opción utilizar kanban para el mantenimiento, incidentes, deuda técnica o nuevas features.

Describe el workflow que usarías para colaborar usando Git. Al igual que con (3), puedes usar algo conocido o una adaptación.

R: Creo que gitflow funciona bien y es un estándar bastante conocido por lo que la curva de aprendizaje del equipo es baja y hay mucha documentación. Esto significa tener un master con la versión de código que está en producción, tener un branch de develop que debería tener lo mismo de master más lo que se va desarrollando en branches de cada nueva feature. Luego de tener tu feature mergeada con develop se crea una branch de release para desplegar una nueva versión de la aplicación. Para arreglos de bugs se utiliza otro

proceso donde se saca una branch directo de master como hotfix que luego de mergearlo con master y sacar un release se mergea con develop para así siempre contar con un ambiente de desarrollo actualizado.

¿Crees que sea necesario agregar algún integrante extra al equipo durante el desarrollo del prototipo? ¿Cuál sería su rol? ¿Crees que sería necesario agregar nuevos integrantes después de la fase de prototipo? ¿Cuándo y por qué?

R: Me gustaría contar con algún dev experto en iOS y otro en Microservicios para tener un código con la mejor calidad posible

¿Qué otras consideraciones tendrías para hacer el proceso de desarrollo robusto y eficiente?

R: Consideraría liberar la aplicación en un subconjunto de usuarios para ir tomando decisiones rápidas en el ciclo de desarrollo de la aplicación sin afectar la reputación de nuestra aplicación. Hacer pruebas para validar hipótesis (ya sea A/B testing o encuestas en la misma aplicación) para garantizar que estamos entregando el valor que queremos entregar.

En lo técnico consideraría tener un pipeline que ejecute validaciones como por ejemplo los test, cobertura de test, seguridad, etc para entregar software de calidad