

Tarea 1 – Ingeniería de Software – Magíster en Ciencias de Datos UC

Nombre del estudiante: Luciano Davico

1. Descripción general de un proyecto en que hayas participado en lo posible recientemente (preferiblemente, un proyecto de desarrollo de software). Descríbelo resumidamente desde el punto de vista del dominio (o ámbito), propósito, usuarios, equipo de desarrollo (¿roles?), tiempo de desarrollo (estimado, real), resultado (entre exitoso, ..., y hasta fracaso) y por qué lo evalúas así.

Un proyecto en el que trabajé recientemente fue desarrollar un framework para poder entrenar y hacer inferencia sobre modelos de Machine Learning de forma automática y estándar, utilizando servicios cloud. Esto es en el ámbito interno de un área de Data y Analytics de una empresa bancaria.

El propósito del proyecto era agilizar las operaciones de MLOps y tener una puesta en producción más ágil de activos analíticos, haciendo más escalable la operación.

Los usuarios que utilizarán el framework son data engineers y data scientist del área de datos de la misma empresa, por lo que es un cliente interno.

El equipo de desarrollo fui únicamente yo, con asistencia de un data engineer en colaborar con piezas de código más concretas, o bien, en velar por mantener un diseño del código que sea poco acoplado y altamente desacoplado.

El tiempo estimado de desarrollo era de tres meses y el tiempo real fue de seis meses. Si consideramos tiempo efectivo de desarrollo, fue de cuatro meses, ya que hubo una ventana de tiempo entre dos y tres meses en que se congeló el proyecto dado otras prioridades que salieron en la empresa.

El resultado del proyecto fue exitoso y con el framework ya se han productivizado ciertos modelos de un ámbito específico relacionado al área de créditos. Lo evaluo de forma positiva dado que se redujo mucha manualidad y fricción para productivizar los modelos de Machine Learning, lo que ahora se puede lograr de forma mucho más simple y rápida para el usuario que opere con el software.

2. Supuestos (dos o tres) a partir de los cuales se llevó adelante el proyecto; para cada uno para cada supuesto:

- **cuál fue el supuesto y por qué se hizo (p.ej., había alguna evidencia / se le preguntó la opinión a un experto / si no se hacía el supuesto, entonces el proyecto no se podía llevar a cabo / etc.)**

- ese supuesto, ¿resultó cierto? En caso afirmativo, ¿de qué manera se benefició el proyecto gracias a que el supuesto era cierto? En caso negativo, ¿qué implicaciones tuvo en el desarrollo/resultado del proyecto el hecho de que el supuesto no se cumplió?

Supuestos:

- **La implementación del software no sería más caro en comparación a lo que ya existe:** el supuesto se hizo ya que, en el momento en que se ideó el proyecto se tiene como vertical de la empresa no tener aumento descontrolado de costos y la operación en tal momento todavía escalaba de tal forma que se puede sobrevivir sin implementar un nuevo software automático de Machine Learning. Se conversó con personas que habían hecho uso anteriormente de los servicios cloud nuevos a implementar en término de costos y comentaron que manteniendo un buen uso del servicio y de forma minimalista, no tendríamos que tener un aumento de costos. El supuesto resultó cierto y de hecho a largo plazo genera economías de escala, ya que facilita realizar un testeo de los modelos pre productivamente a costo cero, por lo que ahora operamos a menor costo y de forma más ágil. Cabe destacar que si aumentaban significativamente los costos, probablemente el proyecto no terminaría concretándose.
- **La curva de aprendizaje de las tecnologías a utilizar no sería alta:** este supuesto se consideró debido a que anteriormente se realizó una prueba de concepto por parte de un practicante en la empresa de lo que sería la antesala del proyecto y la idea era partir desde ahí. Lo anterior pese a que ayudo, no logró disminuir la curva de aprendizaje, dado que la implementación respondía más a un código externo y no se consideraban detalles realmente acordes a la arquitectura de datos que actualmente se maneja en el área. Las implicancias de lo anterior fue un aumento del tiempo que tomaría la implementación técnica del framework, dado que primero hubo que leer documentación del servicio en cuestión (Sagemaker), casos de uso concretos, cómo es la arquitectura de contenedores detrás (por medio de Docker) y entender el funcionamiento del SDK en Python. Lo anterior retrasó el proyecto entre uno y dos meses, pero no afectó en la calidad de la solución final, sino que solo la retrasó.

3. Requisitos del proyecto:

- ¿cómo fue el proceso de levantamiento de los requisitos?
- ¿cómo resultó la experiencia de definición de requisitos con sus usuarios?
- ¿cómo documentaron los requisitos?
- ¿cambiaron los requisitos al transcurrir el tiempo?

- El levantamiento de requisitos se realizó por medio de reuniones con los stakeholders del proyecto. En primer lugar esboqué una arquitectura de alto nivel de los componentes que actuarían en el proceso de Machine Learning y discutimos sobre dicha arquitectura. Ahí hubo un cierto consenso y se establecieron las épicas principales que se tenían que cumplir. Luego, a lo largo de dichas reuniones se estableció lo que debía cumplir el MVP y también una priorización de lo primero que debíamos abordar.
- La experiencia de definir requisitos fue más bien cómoda. En cierto punto, desde el punto de vista técnico yo tenía más visibilidad que los demás, por lo que los stakeholders más bien establecieron sus visiones y los criterios de aceptación. En el proceso se respetó nuestra priorización y se nos entregó a mí y al otro data engineer plena autonomía para definir la arquitectura y el diseño inicial.
- No fueron documentados de manera formal. Sin embargo, se establecieron reuniones de avance todos los viernes y se grababan dichas reuniones. Luego, se generaban tareas técnicas en un backlog que teníamos e íbamos abordando las tareas, estableciendo una descripción y criterios de aceptación en cada una. Dichas tareas se entregaban a los stakeholders solo a modo de validación para alinear las partes. Además, las tareas quedan almacenadas en la historia del tablero Kanban, por lo que se pueden consultar.
- Sí cambiaron. En la mitad del proyecto se generó un MVP y se presentó a los stakeholders. En tal caso, se comentaron posibles mejoras de la arquitectura y se redefinieron ciertos componentes que actuaban en el entrenamiento e inferencia de los modelos. En particular, se pidió revisar la forma de crear trabajos (*jobs*) para que estos sean más genéricos y así facilitar la operación con el framework, a modo que el usuario final no deba preocuparse de aspectos que no sean directamente relacionados con el análisis de datos. Por otra parte, se realizó una repriorización de tareas y además, se consideró que el software debía montarse en un orquestador de datos diferente al pactado en un inicio, pero luego con el pasar de las semanas se volvió al diseño inicial en este último aspecto. Pese a este cambio de requisitos, el adoptar una metodología iterativa-incremental facilitó que la adopción de los cambios sea simple.