Cómo transformar una idea en una startup

Julián Mayorga

Cómo transformar una idea en una startup

Julián Mayorga

Índice general

Desarrollo de Ingeniería

Para completar los conceptos sintetizados en la sección anterior, en las siguientes hojas detallaré cómo usar técnicas de Programación Extrema y de Desarrollo de Clientes para transformar una idea inicial en una startup.

Utilizaré el proceso descripto en la síntesis del Marco Teórico, aclarando con ejemplos cómo conseguir los objetivos de las dos grandes etapas de una startup: Descubrimiento de Clientes y Validación de Clientes.

El esfuerzo de ingeniería de software es grande para llevar adelante una startup innovadora. El equipo de ingeniería debe desarrollar y mantener una infraestructura que lograr dos objetivos: lograr una fácil adaptación a los cambios y realizar entregas frecuentes del producto/servicio. Para lograr la capacidad de adaptarse a cambios utilizaré una **arquitectura modular**, **orientada a servicios**; y para realizar entregas frecuentes del producto usaré **Programación Extrema**.

Diseñaré una arquitectura orientada a servicios para lograr un bajo acoplamiento entre cliente y servidor, y así mejorar la capacidad de adaptación del producto desarrollado. Utilizaré la arquitectura cliente-servidor, con una API REST como servidor y una aplicación con tecnologías modernas como cliente.

Para lograr las entregas frecuentes que requiere una startup utilizaré técnicas de XP. Configuraré el desarrollo para posibilitar una puesta en producción continua, y utilizaré Desarrollo Guiado por Pruebas (TDD en inglés), escribiendo pruebas automatizadas tanto unitarias como de aplicación.

El Desarrollo de Ingeniería comienzará con el desarrollo práctico de la etapa Descubrimiento de Clientes.

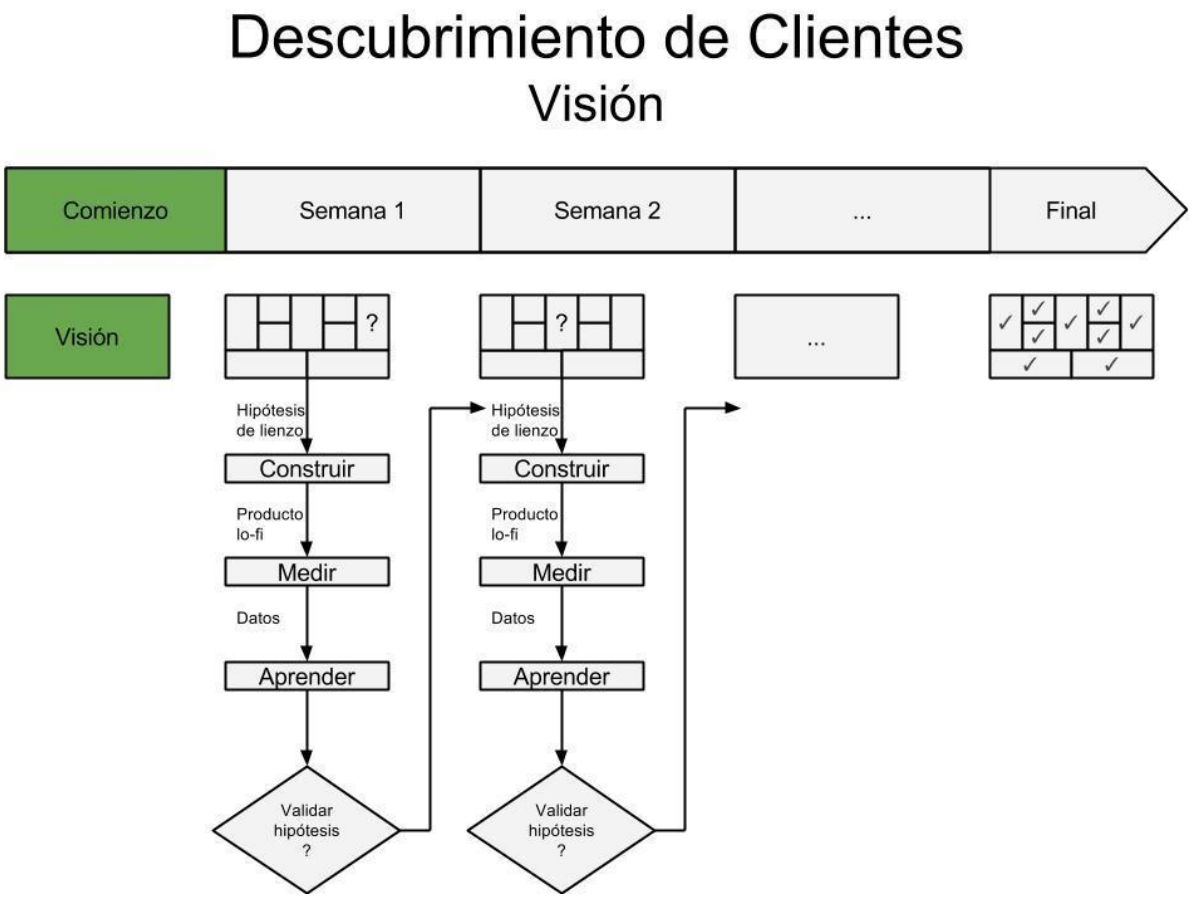
Descubrimiento de Clientes

Ahora comienza la etapa de Descubrimiento de Clientes, en la que se busca validar un modelo de negocios. En esta fase hay que realizar experimento tras experimento hasta llevar a cero la gran lista de incógnitas con la que comienzan todas las startups.

Por cuestiones de tiempo y alcance del presente trabajo realizaré dos iteraciones de Descubrimiento de Clientes, las suficientes para aclarar cada elemento del proceso de creación de startups. En la realidad pueden llegar a ser necesarios meses o años llevar a cabo los experimentos necesarios para conseguir un modelo de negocios estable.

A continuación introduciré la idea sobre la cual construir una startup, y formularé una visión a partir de ella.

Visión



Vision

En este capítulo determinaré la visión que guiará el camino de la startup a partir de una idea. Empezaré por el planteo del problema a resolver, después construiré una visión a partir de lo planteado y sólo al final propondré una solución.

El proceso de planteo de visión propuesto contrasta con la forma de pensar que tienen en su mayoría los ingenieros. Al ser personas encargadas de crear soluciones, generalmente

empezamos la creación de startups con una solución ideal en nuestra mente. El problema con este método es, irónicamente, que no haya problema que se ajuste a nuestra solución.

A continuación describiré el problema a resolver.

Problema

Siempre he pensado que para ser un buen programador no sólo hay que escribir mucho código, sino también leer mucho código. Los buenos escritores se forjan tanto escribiendo obras propias como leyendo obras ajenas, entonces ¿por qué no forjar buenos programadores de la misma manera? Pienso que leer código de calidad es una parte importante del proceso de aprendizaje de programación.

Creo que mejorar la forma de leer código puede significar beneficios en ámbitos laborales, no sólo de aprendizaje. Si bien es cierto que es raro que un programador lea código ajeno en sus momentos ociosos con el único objetivo de leer código de calidad, hay muchas situaciones laborales que pueden ser beneficiadas por herramientas que faciliten la lectura, como por ejemplo:

- Aprender sobre librerías ajenas, lo cual puede ser necesario debido a mala documentación de las mismas
- Encontrar bugs, debido que todos usamos código escrito por terceros en nuestros proyectos
- Agregar features a proyectos ajenos
- Volver a trabajar sobre proyectos propios antiguos, debido a que al pasar el tiempo puede ser dificil comprender hasta el código propio
- Hacer Code Review
- Buscar material para reusar en código ajeno

Sin embargo, una cosa es segura: **Leer código es dificil**. Por eso voy a encarar este problema como inicio para la startup que detallaré en este Trabajo Final.

Visión

Una visión tiene que ser amplia porque todavía no hay nada definido en el modelo de negocios. Si definimos un objetivo muy específico seguramente el mismo va a cambiar en el ciclo de vida de una startup, sin embargo una visión a largo plazo seguirá vigente.

La visión que tengo es la siguiente:

> *Aumentar la capacidad de entender código*

Hay que empezar por un objetivo más alcanzable que la visión propuesta debido a la magnitud de la visión y al carácter iterativo de la metodología de Desarrollo de Clientes. Por este motivo a continuación mostraré cómo deconstruir la visión en una solución al problema de lectura de código.

Solución

En el planteo de problema enumeré motivos por el cuál leer código. Para proponer una solución al problema planteado voy a pensar en cómo se lee código actualmente. Finalmente enumeraré los features que la solución ideal debería tener.

A la hora de pensar cómo leer codigo, se puede armar una extensa lista de posibilidades. Siempre depende del objetivo que cada uno tenga para leer código, pero a nivel general algunas formas de leer son:

- Empezar la exploración por el punto de partida del programa, por ejemplo un archivo main
- Pensar en qué se quiere aprender, buscarlo con herramienta tipo grep y usarlo como punto de partida de la exploración del código
- Buscar palabras clave del lenguaje en el que el código está escrito
- Buscar declaraciones de funciones u objetos interesantes
- Ver estructura del código
- Encontrar comentarios que ayuden a entender fragmentos
- Analizar el código con un profiler y ver donde tarda mas, y usarlo como punto de partida

Teniendo en cuenta la forma en la que se lee el código actualmente, a continiación daré una lista de historias de usuario para la solución propuesta.

Como usuario, quiero:

- Seleccionar código y hacer preguntas sobre él en stack overflow
- Subrayar código que me parezca importante
- Tomar notas sobre fragmentos de código y hacerlas visibles para los otros usuarios
- Hacer diagramas sobre fragmentos de código
- Correr fragmentos de código en intérpretes del mismo y ver el resultado
- Buscar el uso de palabra clave en el código
- Seleccionar código y ver la documentación asociada
- Hacer guías paso a paso sobre el código para que otros vean
- Escribir tests para fragmentos del código
- Refactorizar fragmentos de código
- Armar versiones simplificadas del código
- Remover una implementación y reimplementarla hasta que los tests pasen de nuevo

Es oportuno aclarar que las previas historias de usuario son para la versión ideal de la aplicación. Esta solución ideal se plasmará en la sección de Propuesta de Valor del Lienzo de Modelos de Negocios que se construirá en la próxima sección, para después ser validada o rechazada a través del feedback de clientes.

Una vez que esté determinada la visión, problema y solución de la startup hay que realizar la construcción del Lienzo de Modelos de Negocios, lo cual es el objetivo del siguiente capítulo.