

# **Cómo transformar una idea en una startup**

Julián Mayorga

# **Cómo transformar una idea en una startup**

Julián Mayorga

©2013 - 2014 Julián Mayorga

# Índice general

Agradecimientos . . . . .	1
Resumen . . . . .	2
<b>Introducción . . . . .</b>	<b>3</b>
Planteo del problema . . . . .	4
Contexto del trabajo final . . . . .	7
Metodología usada . . . . .	8
Limitaciones . . . . .	9
Que pretendo dejar al lector . . . . .	10
<b>Marco Teórico . . . . .</b>	<b>11</b>
Ideas innovadoras . . . . .	12
¿Qué es una startup? . . . . .	14
La gran mayoría de las startups fallan . . . . .	17
Lienzo de Modelos de Negocios . . . . .	18
Desarrollo de clientes . . . . .	22
Lean Startup . . . . .	27
Síntesis: Cómo transformar una idea en una startup . . . . .	36
<b>Desarrollo de Ingeniería . . . . .</b>	<b>41</b>
Descubrimiento de Clientes . . . . .	42
Validación de Clientes . . . . .	69
<b>Conclusión . . . . .</b>	<b>75</b>
<b>Bibliografía . . . . .</b>	<b>76</b>

# **Agradecimientos**

# Resumen

La innovación es el motor que impulsa la humanidad hacia adelante. El problema que plantearé en este trabajo final es que la mayoría de las startups innovadoras fracasan, y exploraré soluciones en ingeniería y administración que ayuden a reducir la probabilidad de fallas.

En el trabajo trataré la creación de una startup desde su etapa más temprana, la cual es la creación de una idea innovadora.

Expandiré los conceptos asociados al proceso de creación de startups. Trataré temas como generación ideas innovadoras, qué es una startup, tasa de startups fallidas, modelos de negocios, Desarrollo de Clientes, metodologías ágiles y Lean Startup.

Presentaré la parte práctica de la creación de startups con una descripción de la metodología propuesta. Detallaré paso a paso cómo formular una visión, plantear el problema encarado, proponer una solución, cómo armar las hipótesis sobre el modelo de negocios y finalmente cómo validarlas.

Presentaré un resumen propuesto de las metodologías, que ejemplificaré con un caso práctico. El ejemplo comienza en una idea que, luego de pasar por iteraciones y experimentos, culmina en una startup.

La transformación de la mentalidad del ingeniero que encara un proyecto innovador es la enseñanza principal que el presente trabajo pretende dejar. Propongo al ingeniero un enfoque en la experimentación mas que en la ejecución, y se presenta un marco de trabajo que coloca tanto a la experimentación como al aprendizaje como ejes fundamentales.

# Introducción

*Las compañías innovadoras y en expansión generan crecimiento económico y empleos que, sucesivamente, mejoran extremadamente la vida de las personas.*<sup>1</sup>

La innovación puede brindar incalculables beneficios para la sociedad. Productos novedosos generan empleos, reducen costos y amplían el alcance de soluciones para los sectores más desprotegidos de la sociedad.

El aporte de la ingeniería y la informática a la sociedad en los últimos años es innegable. Google, Apple, Wikipedia, Amazon y muchas otras compañías tecnológicas tienen un enorme nivel de crecimiento impulsado por innovación constante, el cual se traduce directamente en mejoras en el mundo que vivimos.

---

<sup>1</sup>Libro “Innovation and Growth” de David Ahlstrom, Academy of Management

## Planteo del problema

Como estudiante de Ingeniería en Informática siempre me he planteado cuál es el rol del Ingeniero en la innovación. ¿Será un rol técnico, como implementación de soluciones? ¿O también puede ser la resolución de problemas?

En el mundo de la programación existen muchas metodologías para manejar proyectos, dos de los métodos más utilizados son el desarrollo en cascada y las metodologías ágiles.

## Situación



## Metodología: cascada

El modelo en cascada es un proceso secuencial que se ajusta muy bien a situaciones en las que se cuenta con un problema conocido y una solución conocida. Un ejemplo en el que el modelo en cascada es adecuado es en el desarrollo de una nueva versión de un producto ya existente, por ejemplo el desarrollo de una nueva versión de Microsoft Word, o el desarrollo de una nueva versión del juego de fútbol FIFA.

# Situación



## Metodología: ágil

Las metodologías ágiles son métodos iterativos e incrementales que se ajustan muy bien a situaciones en las que se cuenta con un problema conocido y una solución desconocida. Un ejemplo en el que las metodologías ágiles son adecuadas es en proyectos en los que los requerimientos y la solución evolucionan con el tiempo, por ejemplo siempre que se desarrolla software a medida para clientes.

# Situación

**Problema**

X

**Solución**

X

## Metodología: ???

La pregunta es ¿qué metodología utilizar cuando tanto el problema como la solución son desconocidas? En proyectos innovadores no se conocen de entrada estas dos variables, por lo tanto la respuesta a la pregunta inicial es relevante en ámbitos de innovación. El objetivo del presente trabajo final es responder a esta pregunta.

## Contexto del trabajo final

El mejor momento para innovar es ahora, gracias a innovaciones e incentivos a nivel global y nacional.

Cloud computing redujo los costos de servidores al mínimo, y su elasticidad permite aumentar la capacidad de los mismos a niveles inimaginables previamente<sup>2</sup>.

No sólo hay bajos costos iniciales y alta escalabilidad gracias a Cloud Computing, sino que en Argentina hay incentivos para emprendedores, que proveen tanto financiamiento<sup>3</sup> como asesoramiento gratuito<sup>4</sup>.

---

<sup>2</sup><http://www.rackspace.com/blog/newsarticles/88-per-cent-of-cloud-users-point-to-cost-savings-according-to-rackspace-survey/>

<sup>3</sup><http://www.agencia.mincyt.gob.ar/frontend/agencia/fondo/fonsoft>

<sup>4</sup><http://www.industria.gob.ar/emprendedorismo/>

## **Metodología usada**

Para desarrollar el Trabajo Final realicé un estudio bibliográfico de diferentes ideologías en el ámbito de la innovación y la ingeniería de software.

Sinteticé los conocimientos adquiridos en una guía práctica que indica cómo avanzar cuando se quiere transformar una idea en una startup exitosa.

Para exemplificar el camino que se puede recorrer para desarrollar una idea hice un caso práctico, el cual reúne tanto las metodologías de administración como las técnicas de Ingeniería de Software esenciales para un proyecto innovador.

## Limitaciones

Este trabajo final se limita al proceso de creación de startups, no a la administración de empresas. Una startup no es lo mismo que una empresa, una startup se convierte en una empresa cuando tiene aceitados todos sus procesos.

En el trabajo se define una metodología de trabajo que reduce la probabilidad de fallar en una startup, pero es necesario aclarar que el uso de ninguna metodología **no** garantiza el éxito en el desarrollo de startups, debido a que entran en juego muchos factores, como habilidades de personas, comunicación de equipo, etc.

## Que pretendo dejar al lector

Si el lector se tiene que dejar un concepto del presente trabajo, el mismo sería el cambio de mentalidad sobre el rol de los Ingenieros en startups.

Los Ingenieros no sólo resuelven problemas correctamente, **los Ingenieros tienen que descubrir los problemas correctos** y resolverlos correctamente.

# **Marco Teórico**

## Ideas innovadoras

*Si siempre hacés lo que siempre hiciste, siempre vas a obtener lo que siempre obtenés*

– Albert Einstein

La innovación impulsa la humanidad hacia adelante con una rapidez incomparable. Pensar de manera diferente es lo que encendió la mente de muchísimos ingenieros a cambiar el mundo como lo conocemos. Ellos impulsaron la transición del papel y lápiz hacia máquinas de escribir, de máquinas de escribir hacia computadoras, y de computadoras hacia tablets y smartphones, y seguramente de smartphones hacia algo aún más novedoso. Innovación es lo que iluminó a Henry Ford e hizo que dejemos de lado los caballos para empezar a andar en autos.

La capacidad de innovar debería ser un utensilio infaltable en la caja de herramientas de todo ingeniero.

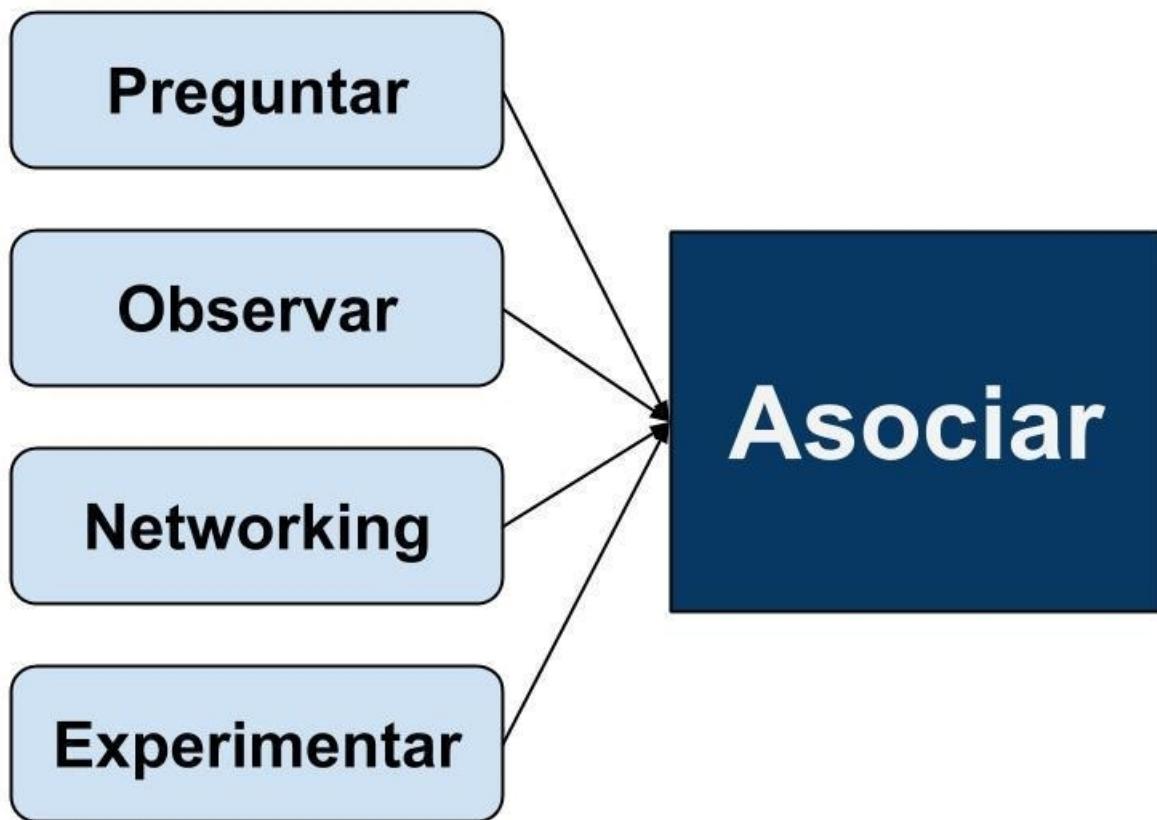
Los productos de la innovación son visibles diariamente, pero ¿qué hace a una idea innovadora?. ¿Innovar es tarea de unos selectos visionarios en todo el mundo?. ¿O todos podemos ser innovadores?. ¿Cómo podemos llegar a tener ideas que cambien a la realidad que conocemos? Estas preguntas han sido formuladas, estudiadas por años y respondidas en el libro *Innovator's DNA*<sup>5</sup>.

Después de ocho años de estudios sobre las personas más innovadoras del mundo—Creadores de compañías como Apple, eBay, Amazon o Salesforce—la conclusión del libro *Innovator's DNA* es que hay patrones recurrentes en personas innovadoras, habilidades que comparten estos innovadores, una especie de ADN en común.

Hay cinco habilidades principales que hacen a un innovador, llamadas habilidades de descubrimiento, las cuales son:

---

<sup>5</sup>“The Innovator's DNA”, por Clayton Christensen, Harvard Business Review Press.



- **Asociar:** Es la más importante de las cinco habilidades, el desarrollo de las otras cuatro es para mejorar el pensamiento asociativo. Asociar conecta ideas totalmente diferentes entre sí, e impulsa a la creación de ideas novedosas, formadas en la intersección de conceptos disímiles.
- *Preguntar:* Los innovadores están cuestionando el presente que conocemos. Preguntas como ¿Que pasaría si ...? o ¿Por qué pasa ...? impulsan nuevos descubrimientos.
- *Observar:* Se llega a nuevos descubrimientos observando el mundo que nos rodea, los sistemas dentro de supersistemas, las relaciones entre los elementos dentro del sistema. Los innovadores son observadores permanentes.
- *Networking:* Conectándose con diferentes personas y ámbitos es como los innovadores llegan a nuevas ideas. Se innova encontrando ideas a través de las conexiones con personas con un punto de vista diferente al propio, probando las ideas con estas personas.
- *Experimentar:* El campo de prueba de las ideas es la experimentación. Con experimentos se llega a conocimientos y enseñanzas imposibles de ver en el mundo de los pensamientos.

Un pensador se puede convertir en innovador a través de la aplicación de estas habilidades, lo cual significa que se puede trabajar para ser más innovador. La capacidad de innovar no es una característica genética, se puede mejorar a través de la práctica.

Tomarse el trabajo de aplicar una, dos o todas estas habilidades a la hora de crear una nueva idea puede significar grandes cargas de innovación adicionadas a la misma. Es bueno tener en cuenta que con un esfuerzo mínimo se le puede dar forma a las ideas aplicando estas cinco herramientas, y así sumar innovación disruptiva a nuestras ideas.

# ¿Qué es una startup?

*El valor de una idea yace en el uso de ella*

– Thomas Edison

Startup es un término de moda en las charlas de tecnología actuales, compañías como Facebook, Twitter o LinkedIn son ejemplos de startups exitosas. Muchas personas tienen definiciones diferentes a la hora de decir exactamente qué es una startup, así que construiré una definición que tome en cuenta los puntos más importantes de las diferentes asepciones.

Primero voy a mostrar la definición de startup según respetables diccionarios, después según referentes del mundo startup y antes de llegar a la definición final, voy a explorar las diferencias de startup con PyME y empresa.

## Definiciones

### Definición de diccionario

Para comenzar la búsqueda de una definición de startup voy a recurrir a las definiciones de diccionario. Se hace visible que se trata de un término novedoso cuando el mismo no está definido en el diccionario de la Real Academia Española (RAE).

*Definición de startup desconocida hasta ahora*

Al tratarse de un término novedoso, que está de moda en países de habla inglesa, es una buena idea buscar la definición en un respetable diccionario anglosajón. Afortunadamente Longman tiene una entrada para la palabra startup, y con la misma podemos llegar a la primera definición:

### Startup

*Una nueva pequeña compañía o negocio, especialmente aquellas cuyo trabajo involucra computadoras o internet: una startup de internet.*

– Diccionario Longman <sup>a</sup>

<sup>a</sup>[http://www.ldoceonline.com/dictionary/start-up\\_2](http://www.ldoceonline.com/dictionary/start-up_2)

Entonces una startup es una compañía nueva, que está metida en el mundo de las computadoras. Pero esta es una definición muy amplia, vamos a expandirla teniendo en cuenta las opiniones de los expertos en la materia.

## Definición basada en objetivo principal

Steve Blank es un reconocido emprendedor de Silicon Valley, que ha fundado ocho startups y ha trabajado mas de treinta años en tecnología. Es reconocido por crear la metodología llamada Desarrollo de Clientes (Customer Development), el cual comenzó el movimiento Lean Startup.<sup>6</sup> Su definición es la siguiente:

*Una startup es una compañía, una sociedad u organización temporal diseñada para buscar un modelo de negocios repetible y escalable.*

– Steve Blank<sup>7</sup>

Lo único que comparte con la definición de Longman es que una startup es una compañía, no limita startups a empresas nuevas, ni al sector tecnológico. Además, menciona el objetivo principal de las mismas: buscar un modelo de negocios repetible y escalable, lo cual es una característica única de las startups.

Hasta ahora podemos construir la definición de la siguiente manera:

## Startup

*Compañía diseñada para buscar un modelo de negocios.*

## Definición basada en innovación e incertidumbre

Eric Ries, al igual que Steve Blank, es un emprendedor de Silicon Valley y también un reconocido autor por comenzar el movimiento Lean Startup, una estrategia de negocios que promueve a las startups a distribuir sus recursos lo más eficientemente posible.<sup>8</sup> Propone la siguiente definición:

*una institución humana diseñada para crear nuevos productos y servicios bajo condiciones de incertidumbre extrema.*

– Eric Ries<sup>9</sup>

Aporta un nuevo objetivo de las startups y también presenta el contexto en el que se dan las mismas. Su objetivo es crear nuevos productos y servicios, y su contexto es uno de incertidumbre extrema. Agregando el punto de vista de Eric Ries a la definición de startup tendríamos que:

<sup>6</sup>[http://en.wikipedia.org/wiki/Steve\\_Bank](http://en.wikipedia.org/wiki/Steve_Bank)

<sup>7</sup><http://steveblank.com/2012/03/05/search-versus-execute/>

<sup>8</sup>[http://en.wikipedia.org/wiki/Eric\\_Ries](http://en.wikipedia.org/wiki/Eric_Ries)

<sup>9</sup>The Lean Startup, Introducción, sección The Lean Startup Method

## Startup

*Compañía diseñada para buscar un modelo de negocios, que crea nuevos productos y servicios y opera bajo condiciones de incertidumbre extrema.*

### Definición basada en crecimiento

Paul Graham es un emprendedor exitoso y unos de los fundadores de Y Combinator<sup>10</sup>, una reconocida inversora de startups. Su concepto de startup es:

*Una startup es una compañía diseñada para crecer rápido.*

– Paul Graham <sup>11</sup>

Según Graham, lo que hace a una startup es su capacidad de crecer rápido. Una startup no necesita ser nueva o involucrar computadoras, no necesariamente tiene que buscar un modelo de negocio o crear productos y servicios nuevos. Si una compañía está diseñada para un crecimiento vertiginoso puede considerarse como startup. Para finalizar la definición de startup, vamos a agregar el aporte de Graham.

## Definición final de startup

*Compañía diseñada para buscar un modelo de negocios y crecer rápidamente, que crea nuevos productos y servicios y opera bajo condiciones de incertidumbre extrema.*

### Conclusión

Una startup, a diferencia de una PyME o empresa, es una compañía diseñada para buscar un modelo de negocios y crecer rápidamente, que crea nuevos productos y servicios y opera bajo condiciones de incertidumbre extrema.

<sup>10</sup><http://ycombinator.com/>

<sup>11</sup><http://www.paulgraham.com/growth.html>

## La gran mayoría de las startups fallan

*No te preocupés por fallar, sólo tenés que acertar una vez*

– Drew Houston, fundador de Dropbox

Ahora que ya hablé de ideas innovadoras y startups exitosas que crecen vertiginosamente—el costado atractivo de las startups—voy a encargarme del costado no tan ameno: construir una startup es **muy difícil**.

Los números hablan por sí mismos, **nueve de cada diez startups no tienen éxito** <sup>12 13</sup>.

Es importante saber que la tasa de fallo es alta si uno quiere seguir el camino del emprendedurismo, y poder armarse de equipamiento y herramientas adecuadas para el mismo. Aprender sobre los casos de éxito es una buena forma de recolectar herramientas necesarias a la hora de construir una startup.

A continuación se detallarán diferentes metodologías que ayudarán al lector si decide emprender el camino de su propia startup.

---

<sup>12</sup><http://www.effectuation.org/paper/failing-firms-and-successful-entrepreneurs-serial-entrepreneurship-simple-machine>

<sup>13</sup><http://www.altassets.net/knowledge-bank/investor-performance-in-israeli-high-tech.html>

# Lienzo de Modelos de Negocios

*No se trata de ideas. Se trata de llevarlas a cabo*

– Scott Belsky, cofundador de Behance

La primera herramienta que todo emprendedor moderno debe conocer es el Lienzo de Modelos de Negocios, el cual le permite describir en una sola hoja cómo funciona su startup.

## Introducción

El término modelo de negocios describe cómo una organización crea, entrega y captura valor<sup>14</sup>. El concepto de modelo de negocios no es algo novedoso, pero en los últimos años surgió una nueva herramienta para describir, analizar y diseñar modelos de negocios, llamado Lienzo de Modelo de Negocios, o “Business Model Canvas” en inglés.

## Problema que soluciona

Un frecuente primer paso a la hora de comenzar un emprendimiento es describirlo a través de un plan de negocios de cientos de páginas. El problema de estos planes de negocios es que son difíciles de comprender al cien por ciento, debido a su longitud; y lo más probable es que sean tediosos de mantener actualizados a medida que evoluciona el negocio que describen.

El lienzo de modelos de negocios surge como solución al problema de complejidad que presentan los planes de negocios. Constituyen un lenguaje común para describir modelos de negocios.

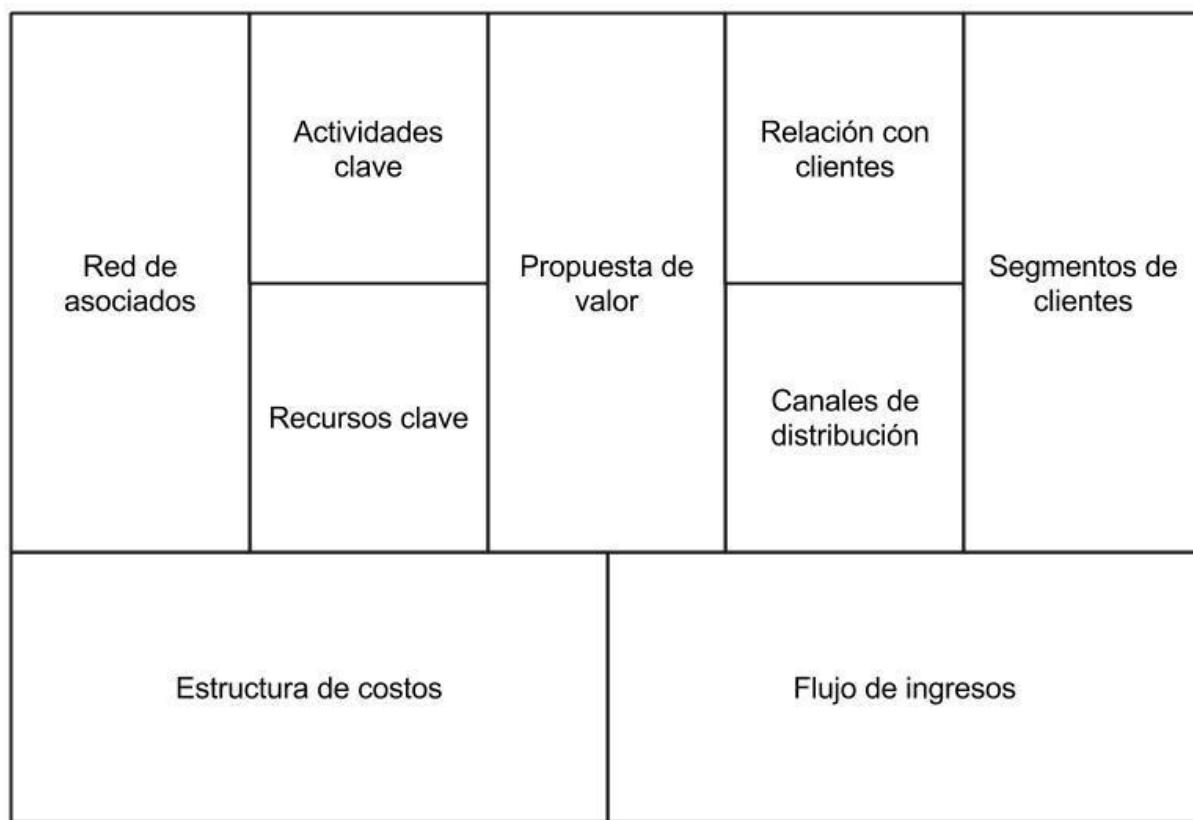
## Cómo está constituido

El lienzo provee nueve bloques de construcción que describen el modelo de negocios de una compañía.

Los nueve bloques están organizados en el lienzo de una manera particular, la cual permite que sea sencillo visualizar el modelo de negocios en un rápido vistazo.

---

<sup>14</sup>Libro “Business Model Generation”, capítulo “Canvas”, sección “Definition of a Business Model”; de Alexander Osterwalder



Canvas de modelos de negocios

1. Segmentos de clientes: Son los clientes de la organización
2. Propuestas de valor: Problemas solucionados y necesidades satisfechas por la compañía
3. Canales: Vías de comunicación, distribución y ventas por las que se entrega la propuesta de valor a los clientes
4. Relaciones con clientes: Relación que se mantiene con cada segmento de clientes
5. Flujos de ganancias: Se obtienen gracias a la propuesta de valor entregada a clientes
6. Recursos clave: Bienes requeridos para ofrecer servicios y generar productos
7. Actividades clave: Realizadas para obtener los otros elementos del modelo de negocios
8. Sociedades clave: Sociedades entabladas para realizar actividades y conseguir recursos externos a la empresa
9. Estructura de costos: Costos asociados al funcionamiento de los otros bloques de negocio

El objetivo de una startup es buscar su modelo de negocios, y el lienzo sirve para definir elementos del modelo de negocios de una startup. Por lo tanto el lienzo es una excelente herramienta para seguir la evolución en el tiempo de ésta búsqueda.

### Lienzo adaptado a startups

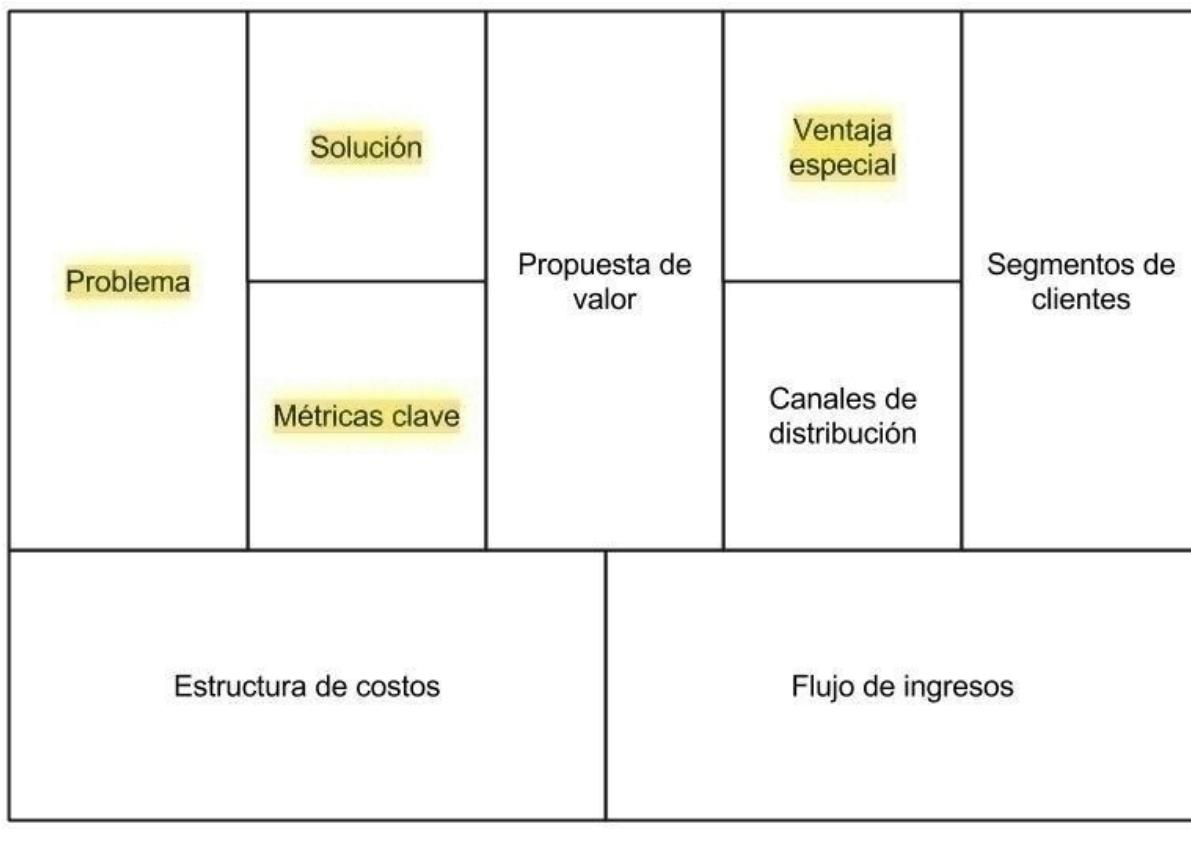
El lienzo de modelos de negocios se utiliza en todo tipo de compañías, no sólo en startups. Debido a que hay patrones que siempre se repiten, y bloques que no son tan necesarios en startups, Ash Maurya diseñó una versión adaptada del lienzo, llamado “Lean Canvas”.<sup>15</sup>

<sup>15</sup>Libro “Running Lean”, capítulo “Step 1: Document Your Plan A”, sección “Capture Your Business Model Hypotheses”; de Ash Maurya

La misma agrega un enfoque en el problema que se está tratando de resolver, mucho más que en la solución planteada. Esto se debe a que un emprendedor debe enfocarse mucho más en el problema a resolver que en la solución.

*A los clientes no les importa tu solución. Les importan sus problemas.*

– Dave McClure, 500 Startups



Lean canvas

Este canvas intercambia cajas del lienzo de modelo de negocios, a continuación describo los elementos adicionados y removidos al igual que los motivos de cada cambio<sup>16</sup>:

### Elementos agregados

- Problema: Lean canvas tiene una caja especial para problema para resaltar su importancia. Muchas startups fallan porque no entienden en problema de sus clientes
- Solución: Tiene una caja pequeña para enfatizar que la solución es algo sobre lo que hay que iterar, porque muchos emprendedores se ensañan con la primer solución construida
- Métricas clave: Hay que enfocarse en pocas métricas, y asegurarse que sean importantes. Muchas startups se ahogan en métricas insignificantes, desperdiциando el recurso más preciado con el que cuentan: su tiempo

<sup>16</sup><http://practicetrumpstheory.com/2012/02/why-lean-canvas/>

- Ventaja especial: También llamada ventaja competitiva o barrera de entrada. El objetivo de esta caja es que comience en blanco al inicio de la vida de una startup, e ir llenándola a medida que genera algo que no pueda ser fácilmente copiado, lo cual pasa a ser su ventaja competitiva

**Elementos removidos** Al trabajar en una startup web/mobile, el emprendedor notará que los siguientes elementos no cambian seguido, o tienen dentro certezas, no hipótesis a probar. Lo cual significa espacio dentro del lienzo que es mejor ocupado por los elementos adicionados en el Lean Canvas.

- Actividades clave y recursos clave: Generalmente tiene valores como desarrollo de software, desarrollo de clientes, desarrolladores o diseñadores
- Relaciones con clientes: La metodología de Desarrollo de Clientes recomienda tener contacto directo con clientes desde el día cero, y luego encontrar un canal para llegarles
- Red de asociados: En un comienzo puede ser difícil o improductivo conseguir socios en etapas tempranas de una startup, si bien es cierto que a futuro pueden haber sociedades clave importantes

## Cómo utilizarlo

En el siguiente capítulo se explorará la técnica de Desarrollo de Clientes, metodología que demuestra cómo utilizar el lienzo de modelos de negocios para encontrar un problema que valga la pena solucionar.

# Desarrollo de clientes

*Encontrás tu milésimo cliente solo después de tu centésimo y tu centésimo después del décimo.  
Tu décimo después del primero. Empezá ahí*

– Asif Mandozai

## Introducción

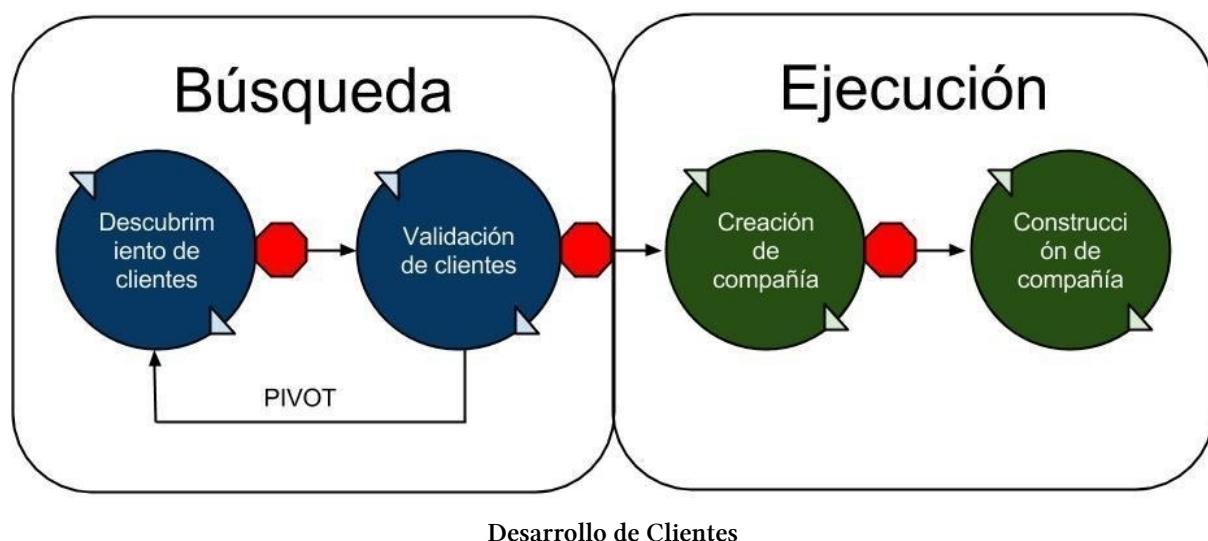
La segunda herramienta para emprendedores que hay que conocer es la técnica llamada Desarrollo de Clientes. En este capítulo exploraremos esta metodología. Primero veremos por qué fue ideado este método, luego cómo funciona el proceso y también cómo usarlo junto al lienzo de modelos de negocios para construir una startup exitosa.

### ¿Por qué fue creada?

La metodología es un patrón propuesto para **tener éxito en startups**, que ha sido creado por Steve Blank a partir del estudio de cómo han llegado al éxito varias startups.

### Proceso de Desarrollo de Clientes

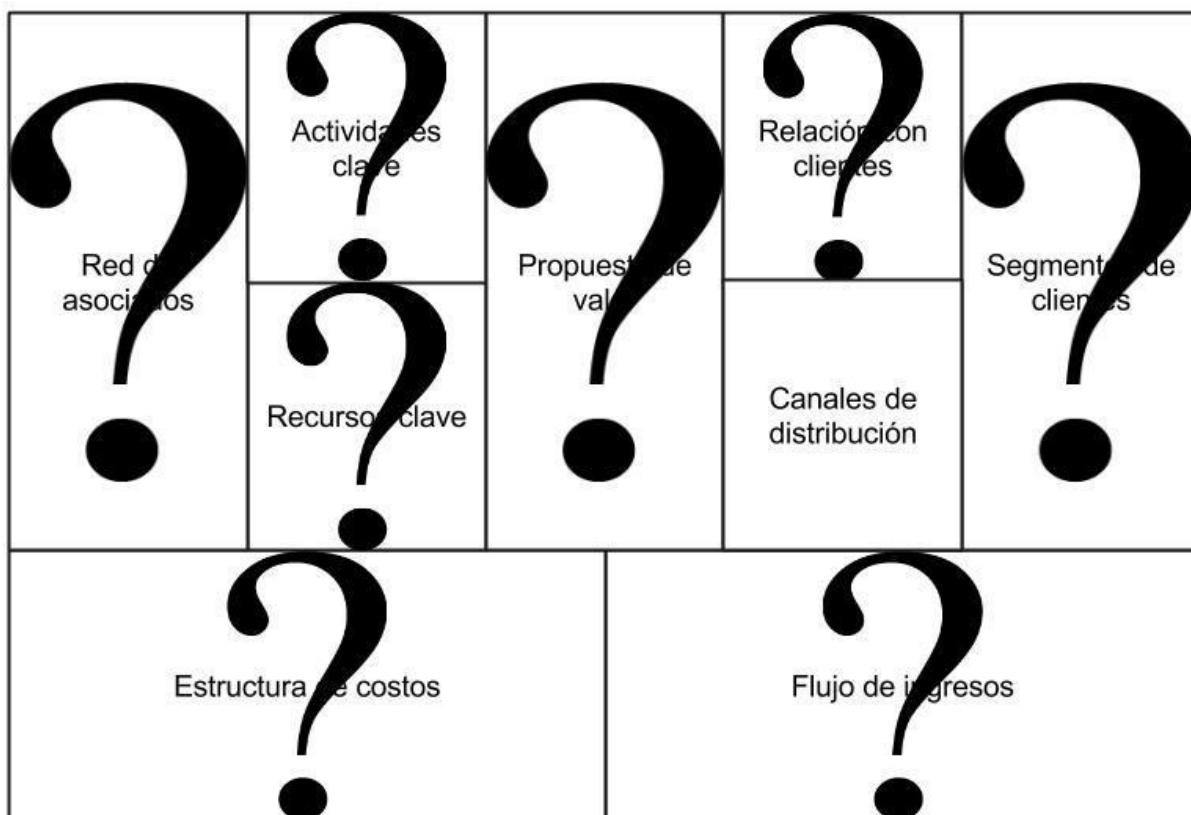
El proceso está enfocado en descubrir y validar un negocio. Separa la vida de una organización en dos grandes etapas, cada una dividida en dos procesos:



En la primera etapa se realiza la búsqueda de un modelo de negocios, y la segunda constituye la ejecución del modelo descubierto.

### Búsqueda

Toda startup permanece en esta etapa de búsqueda mientras tenga incógnitas en su modelo de negocios, es decir está buscando su modelo de negocios.



En un comienzo lo único que tiene una startup es un conjunto de hipótesis no validadas

La etapa de búsqueda consta de dos sub-etapas llamadas Descubrimiento de Clientes y Validación de Clientes, las cuales pueden sintetizarse de la siguiente manera:

- Un producto soluciona un problema para un grupo de usuarios identificables (Descubrimiento de Clientes)
- El mercado es vendible y suficientemente grande para que se pueda construir un negocio viable (Validación de Clientes) <sup>17</sup>

La “graduación” de una startup hacia una compañía es cuando su objetivo pasa desde la búsqueda hacia la ejecución.

**Desarrollo de Clientes** La relación de la metodología de Desarrollo de Clientes con el Lienzo de Modelos de Negocios resulta evidente en esta etapa. Para explicar la relación se citará a Steve Blank:

*Los ganadores reconocen su “visión” como una serie de hipótesis no probadas que necesitan “prueba de clientes”. Ellos buscan revelaciones implacablemente, y ajustan su camino en días o semanas, no meses o años, para preservar dinero y eliminar el tiempo perdido construyendo funcionalidad que los clientes no necesitan.*

– Steve Blank <sup>18</sup>

<sup>17</sup>Libro “Entrepreneurs Guide to Customer Development”, capítulo “Customer Development”, sección “What Customer Development is”

<sup>18</sup>Libro “The Startup Owner’s Manual”, capítulo “Introduction”, sección “A Repeatable Path” por Steve Blank

**El lienzo de modelos de negocios funciona como una planilla de avance para el Desarrollo de Clientes.** Es la herramienta que se utilizará para trackear la validación de cada bloque. A medida que avance el tiempo se contará con varios lienzos, los cuales formarán una especie de crónica del desarrollo de la startup en el tiempo.

Para validar cada bloque del lienzo hay que “salir del edificio”, realizar experimentos que prueben cada hipótesis y enfrentarlos a los clientes. Sólo entrando en contacto con los clientes se puede construir un producto que solucione problemas reales. Por eso es necesario “salir del edificio”, los problemas no se resuelven dentro de oficinas o garajes, se resuelven con clientes satisfechos.

### Pivot

*¿Qué hacer si se descubren hipótesis completamente equivocadas?*

Es normal tener suposiciones erradas, ningún plan de negocios comienza siendo perfecto. A diferencia de una empresa—donde no hay que cometer errores— el objetivo de una startup es equivocarse para poder aprender de los errores.

Por ejemplo, en el lienzo inicial la hipótesis de Segmento de Clientes puede ser “adolescentes de sexo masculino que viven en zonas rurales”. A la hora de recolectar datos sobre los usuarios del producto si descubrís que los usuarios son en su mayoría madres solteras entre 30 y 40 años que viven en zonas urbanas. En este caso es lógico tomar la decisión de enfocarse en este segmento, entender el valor que el producto les ofrece y hacer los ajustes necesarios.

Un cambio drástico a uno de los bloques del modelo de negocios es llamado pivot.

Se logra el éxito en esta etapa **validando qué tan bien se entendió el problema**. El objetivo del Descubrimiento de Clientes NO es vender, las hipótesis de venta se validarán en la próxima etapa de Búsqueda, llamada Validación de Clientes.

**Validación de Clientes** El único objetivo de esta etapa es plantear y validar las hipótesis de:

- Quién va a comprar
- Por qué
- A qué precio

¿Cómo validar los intentos de compra? Construyendo un MVP de alta fidelidad, y consiguiendo que los clientes paguen por este producto mínimo. El concepto de compra depende del producto o servicio que se esté desarrollando; puede ser dinero, una descarga, una registración o un click.

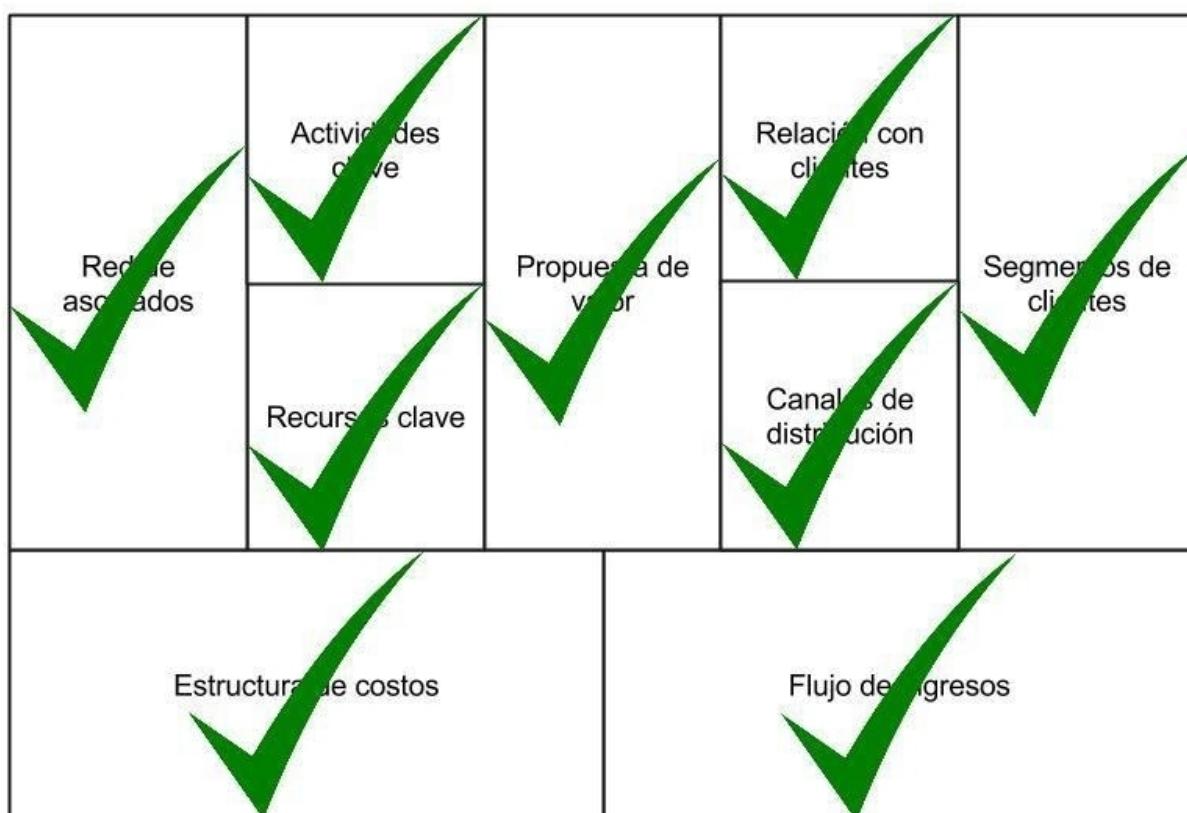
Hay que hacer experimentos sobre el lienzo de modelo de negocios en su totalidad, a diferencia de la etapa anterior en la que se testeaba cada bloque individualmente.

En esta etapa hay que plantear **hipótesis de crecimiento**, e iterar hasta encontrar el “motor de crecimiento”<sup>19</sup> de nuestra startup. Una startup puede tener tres posibles motores de crecimiento, los cuales son:

<sup>19</sup>Libro “The Lean Startup”, capítulo 10: Grow por Eric Ries

- Crecimiento viral
  - Diseñado para esparcirse rápidamente de persona a persona
  - Cada usuario nuevo trae más de un usuario adicional
- Crecimiento “sticky”
  - Diseñado para atraer y retener clientes a largo plazo
  - El crecimiento orgánico es mayor a la tasa de clientes perdidos
- Crecimiento pago
  - Diseñado para crear fondos destinados a publicidad
  - El valor de por vida de cada usuario es mayor al costo de adquisición por usuario

Si la solución encontrada luego de realizar experimentos no es vendible o suficientemente grande para construir un negocio es necesario volver a la etapa anterior de la metodología. Hay que realizar un nuevo Desarrollo de Clientes planteando nuevas hipótesis de problema y segmento de clientes, en otras palabras, hay que hacer un pivot.



Una startup se “gradúa” en una compañía una vez que valida cada elemento de su modelo de negocios

Luego de contar con un modelo de negocios validado, y tener certeza de que hay un mercado vendible y suficientemente grande hay que pasar a la siguiente etapa del Desarrollo de Clientes: la etapa de Ejecución.

El término venta es usado en un sentido amplio, no necesariamente una venta involucra dinero. Una venta en una startup web/mobile puede ser un click, un registro de usuario, una descarga

o dinero.

## Ejecución

En esta etapa la startup ya se ha convertido en una compañía, y cuenta con dos subetapas: Creación de Compañía y Construcción de Compañía:

- El negocio es escalable a través de ventas repetibles y una hoja de ruta de marketing (Creación de Compañía)
- Departamentos de la compañía y procesos operacionales son creados para posibilitar escala (Construcción de Compañía) <sup>20</sup>

Lamentablemente el desarrollo de esta etapa está fuera del alcance del presente Trabajo Final. Afortunadamente hay incontables libros de Administración a los cuales se puede recurrir para ejecutar sobre un modelo de negocios validado.

## Resumen

Este capítulo trató la metodología de Desarrollo de Clientes creada por Steve Blank. El origen de la misma fue la creación de un patrón para tener éxito en startups. Consta de dos etapas, búsqueda y ejecución.

El objetivo de la primera etapa es buscar un modelo de negocios, y el mismo se consigue realizando experimentos y enfrentando el producto con los clientes lo más temprano posible. El objetivo de la segunda etapa es desarrollar una compañía a partir de un modelo de negocios validado.

Nadie mejor que el creador de la metodología para resumirla en pocas palabras:

*El núcleo de Desarrollo de Clientes es dichosamente simple: Productos desarrollados por fundadores que salen a enfrentarse a los clientes temprano y seguido, ganan.*

– Steve Blank <sup>21</sup>

El próximo capítulo es sobre la técnica llamada Lean Startup, que muestra cómo mejorar el proceso de Desarrollo de Clientes con una metodología que resulta bastante familiar para los Ingenieros Informáticos.

<sup>20</sup>Libro “Entrepreneurs Guide to Customer Development”, capítulo “Customer Development”, sección “What Customer Development is”

<sup>21</sup>Libro “The Startup Owner’s Manual”, capítulo “Introduction”, sección “The Four Steps: A New Path” por Steve Blank

## Lean Startup

*¿Por qué usar el enfoque Lean? La vida es muy corta para construir algo que nadie quiere*

– Ash Maurya

### Introducción

En los últimos dos capítulos presenté cómo describir una startup utilizando el Lienzo de Modelos de Negocios y cómo el Desarrollo de Clientes lo utiliza para validar todos los aspectos del negocio.

En este capítulo presentaré el método Lean Startup, utilizado por cada vez más startups en todo el mundo. El mismo se basa en la filosofía de “salir del edificio” del Desarrollo de Clientes, agregandole prácticas que elevan la agilidad y el nivel de experimentación al máximo.

Comenzaré enumerando los beneficios de la metodología. Después mostraré los principios fundamentales de la misma, junto con ejemplos de casos reales para bajar los conceptos a la realidad.

Para describir cómo operar una Lean Startup describiré el método con tres conceptos fundamentales; la unión del Desarrollo de Clientes con las Metodologías Ágiles, el bucle Construir-Medir-Aprender y el Mínimo Producto Viable.

### Beneficios del método Lean Startup

Los beneficios principales que se obtienen al aplicar la metodología son metas que toda startup desea, ellos son:

- Ser mas innovador
- Dejar de desperdiciar el tiempo de las personas
- Ser mas exitoso <sup>22</sup>

### Principios de Lean Startup

La filosofía Lean Startup posee cinco principios, los cuales son:

1. Emprendedores hay por todos lados
2. Emprendedorismo es administración
3. Aprendizaje validado
4. Contabilización de innovación
5. Construir-medir-aprender

Para facilitar la comprensión de estos principios con casos reales contaré cómo diferentes startups han utilizado la metodología.

---

<sup>22</sup><http://theleanstartup.com/#benefits>

## Emprendedurismo es administración

Para exemplificar que el “*emprendedurismo es administración*” se citará al autor de Lean Startup, Eric Ries:

El emprendedurismo es una especie de administración. No, no leíste mal. Tenemos asociaciones salvajemente divergentes sobre esas dos palabras, *emprendedurismo* y *administración*. Últimamente, parece que una es *cool*, innovadora y emocionante y la otra es aburrida, seria y sosa. Es tiempo de dejar atrás esos preconceptos.

Dejame contarte la historia de una startup. Es 2004, y un grupo de fundadores recién han comenzado una nueva compañía. Su compañía anterior falló muy públicamente. Su credibilidad está en su nivel mas bajo. Tienen una visión enorme: cambiar la forma en la que la gente se comunica al usar una nueva tecnología llamada avatars (recuerden. esto era antes de la taquillera película de James Cameron). Están siguiendo a un visionario llamado Will Harvey, que pinta una imagen irresistible: personas conectándose con sus amigos, pasando el rato online, usando avatares para darles una combinación de conexión íntima y anonimidad segura. Aún mejor, en vez de tener que construir toda la ropa, muebles y accesorios que estos avatares necesitarían para anexar a sus vidas digitales, los clientes serían enlistados para construir esas cosas y venderlas unos a otros.

El reto de ingeniería ante ellos es inmenso: crear mundos virtuales, contenido generado por usuarios, un motor de e-commerce, micropagos y por último, pero no menos importante, la tecnología tridimensional de avatares que pueda correr en la PC de cualquiera.

Estoy en esta historia. Soy un cofundador y Director de Tecnología de esta compañía, que es llamada IMVU. En este punto de nuestras carreras, mis cofundadores y yo estamos determinados a cometer nuevos errores. Hacemos todo mal: en vez de gastar años perfeccionando nuestra tecnología, construimos un producto mínimo viable, un producto precoz que es terrible, lleno de errores y problemas de estabilidad trabamos-tu-computadora-si-en-serio. Después lo enviamos a producción a los clientes antes que esté listo. Y cobramos dinero por él. Después de asegurar los clientes iniciales, cambiamos el producto constantemente -demasiado rápido en comparación con los estándares tradicionales- enviando a producción nuevas versiones del producto decenas de veces cada día.

Realmente teníamos clientes en esos días iniciales -verdaderos pioneros visionarios, hablábamos con ellos seguido y les pedíamos feedback. Pero enfáticamente *no* hicimos lo que ellos decían. Vimos sus aportes como sólo una fuente de información sobre el producto y visión general. De hecho, era mucho más probable que corriéramos experimentos en nuestros clientes antes que satisfacer sus caprichos.

El pensamiento tradicional de negocios dice que este enfoque no debería funcionar, pero funciona, y no necesitás creerme. Como verás a través del libro Lean Startup, este enfoque que iniciamos en IMVU se ha convertido en la base para un nuevo movimiento de emprendedores alrededor del mundo. Se basa en varias ideas de administración, incluídas *lean manufacturing*, pensamiento de diseño, desarrollo de clientes, y desarrollo ágil. Representa un nuevo enfoque a la creación de innovación continua. Es llamado Lean Startup.<sup>a</sup>

<sup>a</sup>The Lean Startup (Crown Business). Copyright 2011 Eric Ries, 0307887898". Introducción"

En pocas palabras: Una startup es una institución, no sólo un producto, entonces requiere administración, una nueva clase de administración específicamente orientada a su contexto.

## Emprendedores hay por todos lados

Uno de los principios más importantes de Lean Startup es “*Emprendedores hay por todos lados*”. Para entenderlo con una historia de éxito mostraré la historia de una startup del área de servicios financieros personales llamada SnapTax:

En el 2009, una startup decidió tratar algo realmente audaz. Querían liberar a los que pagan impuestos de las costosas tiendas de impuesto al automatizar el proceso de recolectar información tipicamente encontrada en formularios W-2 (es el resumen de final de año que la mayoría de los empleados de USA reciben de sus empleadores, resume sus tasas de impuestos del año). La startup rápidamente se encontró con dificultades. A pesar que muchos clientes tenían acceso a impresoras/scanners en sus casas u oficinas, pocos sabían cómo usar esos dispositivos. Después de numerosas conversaciones con clientes potenciales, el equipo se iluminó con la idea de hacer que los clientes saquen fotos a los formularios directamente desde sus teléfonos. En el proceso de probar este concepto, los clientes preguntaron algo inesperado: ¿será posible terminar *el proceso entero de devolución de impuestos* desde el teléfono mismo?

Esa no era una tarea fácil. La preparación tradicional de impuestos requiere que los consumidores pasen con dificultad a través de cientos de preguntas, muchos formularios y mucho papelerío. Esta startup intentó algo novedoso al decidir sacar al mercado una versión inicial del producto que podía hacer mucho menos que el paquete completo de impuestos. La versión inicial funcionó sólo para consumidores con un resultado de impuestos muy simple, y sólo funcionaba en California.

En vez de hacer que los consumidores llenen un formulario complejo, permitieron que los consumidores usen la cámara del teléfono para sacar una foto de sus formularios W-2. De esa única foto, la compañía desarrolló la tecnología para compilar y entregar la mayoría del resumen de impuestos 1040 EZ. Comparado con el trabajo penoso del resumen de impuestos tradicional, el nuevo producto -llamado SnapTax- proveeeyó una experiencia mágica. Desde sus modestos comienzos, SnapTax creció hasta ser un importante caso de éxito de startup. Su lanzamiento nacional en el 2011 mostró que los clientes lo amaban, al ritmo de mas de 350000 descargas en sus primeras tres semanas.

Este es el tipo de increíble historia de innovación que esperarías de una startup nueva.

Sin embargo, el nombre de la compañía te puede sorprender. SnapTax fue desarrollado por Intuit, el productor más grande de América de herramientas de finanzas, impuestos y contabilidad para individuos y empresas pequeñas. Con más de 7700 empleados y ganancias anuales en los billones, Intuit no es una startup típica.

El equipo que construyó SnapTax tampoco se parece mucho a la imagen típica de emprendedores. Ellos no trabajan en un garaje o comen *ramen noodles*. A su compañía no le faltan recursos. Ellos reciben salario y beneficios. Ellos van a trabajar a la oficina todos los días. A pesar de eso

ellos son emprendedores.<sup>a</sup>

<sup>a</sup>The Lean Startup (Crown Business). Copyright 2011 Eric Ries, 0307887898". Sección "The snaptax story"

Resumiendo: No necesitas trabajar en un garaje para estar en una startup

## Descripción del método Lean Startup

Después de describir Lean Startup con historias, describiré los componentes que forman la metodología: Desarrollo de Clientes + Metodologías Ágiles, Bucle Construir-Medir-Aprender y Mínimo Producto Viable.

### Unión de Desarrollo de Clientes y Metodologías Ágiles

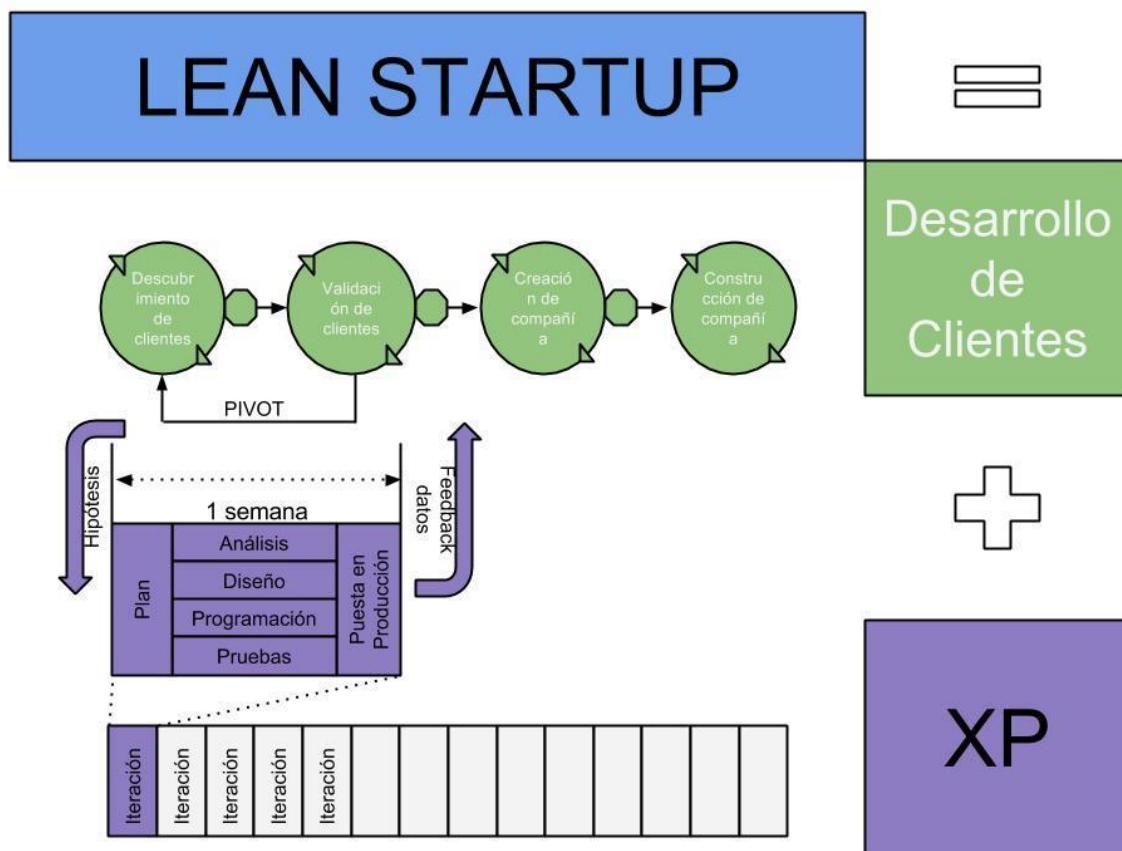
El gran aporte de Lean Startup a la metodología de Desarrollo de Clientes es agregarle el desarrollo del producto utilizando Metodologías Ágiles, más específicamente Programación Extrema (XP).

¿Por qué es un acierto tan grande agregar XP?

Para poner en producción software más frecuentemente.<sup>23</sup>

Enfrentar el software a los clientes es un beneficio clave. Recordemos que el objetivo del Desarrollo de Clientes es llevar el producto hacia los clientes lo más rápido posible para obtener feedback pronto. Para lograr este objetivo nada mejor que poner software en producción más seguido, para lo cual hay que usar métodos ágiles.

<sup>23</sup>Libro "The Art of Agile Development", capítulo 1 "Why Agile?", por J. Shore y S. Warden



Lean Startup = Desarrollo de Clientes + Metodologías Ágiles

¿Cómo hacer encajar Desarrollo de Clientes con Programación Extrema? El ciclo iterativo que se forma se puede enumerar de la siguiente manera:

1. Plantear hipótesis
2. Comenzar experimento
3. Realizar iteración de XP
4. Poner en producción producto para recolectar métricas
5. Analizar los datos obtenidos para validar o refutar hipótesis
6. Volver a plantear hipótesis para comenzar el ciclo de nuevo

**Programación Extrema** Ya mencioné que el beneficio principal de utilizar XP es entregar software más frecuentemente, pero ¿qué hay que hacer para lograr esto?

La base de XP es la realización de iteraciones semanales, no mensuales como otras metodología ágiles, ni mucho menos anuales como el desarrollo en cascada; hay que iterar semanalmente. En cada iteración se realizan todas las actividades de Ingeniería de Software: Análisis, Diseño, Programación, Pruebas y hasta Puesta en Producción.

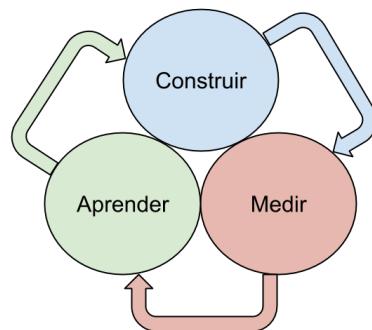
Hay varias técnicas propias de la programación extrema. A continuación se enumerarán las que más beneficio proveen a una Lean Startup:

- *Build* de diez minutos: Un build rápido y automatizado incrementa la productividad y reduce al mínimo los errores de producción que trae un proceso manual.
- Integración continua: El objetivo de la integración continua es poner en producción todo excepto las últimas horas de trabajo. Esto se consigue teniendo un servidor de integración continua que junta el código de los diferentes programadores, verifique que todo funciona corriendo las pruebas automatizadas y notifique al equipo si hay algún error.
- Desarrollo guiado por pruebas (TDD): Escribir las pruebas antes de escribir código lleva numerosos beneficios, como un mejor diseño, mayor cobertura de código probado y menos bugs. No es una práctica fácil de adoptar, pero vale la pena el esfuerzo.
- Retrospectivas: Retrospectivas frecuentes son excelentes para mejorar los procesos continuamente.

A continuación mostraré cómo resumir el proceso de Lean Startup en un bucle iterativo, llamado Construir-Medir-Aprender.

### **Bucle Construir - Medir - Aprender**

Un eje de toda Lean Startup es el bucle “*Construir-medir-aprender*”. El mismo consiste en un proceso iterativo que involucra construir un producto mínimo, tomar métricas clave al enfrentar el producto a clientes, aprender utilizando estas medidas para después comenzar de nuevo el ciclo.



**Construir-Medir-Aprender**

El bucle es utilizado para probar las hipótesis definidas sobre la startup. La compañía tiene que ser lo suficientemente ágil para correr permanentemente estos experimentos. Para dar una idea de la agilidad con la que hay que iterar citaré a Eric Ries contar la historia de cómo una conocida empresa americana llamada Intuit comenzó a correr cientos de experimentos para probar cada elemento de su visión.

Uno de sus productos insignia es TurboTax. Porque TurboTax hace la mayoría de sus ventas en época de impuestos de USA, solía tener una cultura extremadamente conservadora. Al transcurrir el año, los equipos de marketing y producto concebían una iniciativa importante que sería puesta en marcha justo a tiempo para época de impuestos. Ahora ellos prueban mas de quinientos cambios diferentes en una época de impuestos de dos meses y medio. El equipo puede hacer un cambio en vivo en su sitio un Jueves, correrlo todo el fin de semana, leer los resultados el Lunes y llegar a conclusiones al empezar el próximo Jueves; después construyen nuevas pruebas ese mismo dia y lanzan el próximo conjunto de pruebas a la noche.

Como Scott (fundador de Intuit) dijo,

“Hombre, la cantidad de aprendizaje que obtienen es inmenso ahora. Y lo que hace es desarrollar emprendedores, porque cuando tenés solo una prueba, no tenés emprendedores, tenés políticos, porque tenés que vender. A partir de cientos de buenas ideas, tenés que vender la tuya. Entonces construís una sociedad de políticos y vendedores. Cuando tenés quinientas pruebas corriendo, entonces las ideas de todos pueden correr. Y después creás emprendedores que corren y aprenden y pueden probar de vuelta y aprender nuevamente, todo lo contrario que una sociedad de políticos. Entonces estamos tratado de implementar eso a través de nuestra organización, usando ejemplos que no tienen nada que ver con tecnología de punta, como el ejemplo del sitio web. Hoy en día todo negocio tiene un sitio web. No necesitás estar en tecnología de punta para usar ciclos rápidos de testing.” <sup>a</sup>

<sup>a</sup>The Lean Startup (Crown Business). Copyright 2011 Eric Ries, 0307887898”. Sección “A seven thousand person lean startup”

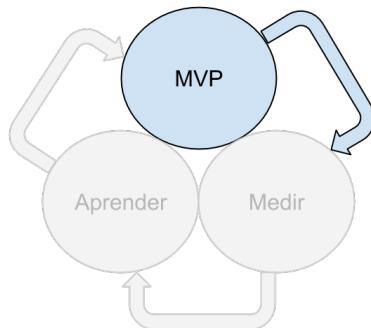
**Intuit invirtió en sistemas que incrementaron la velocidad en la que las pruebas podían ser construidas, puestas en producción, medidas y analizadas.**

El objetivo de cada iteración es el aprendizaje sobre la hipótesis planteada al comienzo, y mientras más rápido iteremos por el bucle más aprendizaje vamos a obtener.

Para acelerar el bucle se puede optimizar la etapa de construcción. Esto se logra construyendo el producto más simple que nos provea de los datos necesarios para analizar. A continuación se verá cómo optimizar la etapa de construcción realizando productos con funcionalidad mínima.

## MVP: Mínimo Producto Viable

En la etapa de construcción se ensambla lo mínimo para probar las hipótesis, un Mínimo Producto Viable. Un MVP debería tener la funcionalidad exacta que pruebe la hipótesis planteada. No debe poseer funciones redundantes que causen pérdida de tiempo; y tampoco debe tener una falta de comportamiento que ocasione poco material para aprender.



### MVP

En la etapa de Descubrimiento de Clientes el objetivo es validar un problema y un segmento de clientes que lo posea. No se trata de validar soluciones correctas, o canales de distribución adecuados.

Por lo tanto en esta etapa normalmente se construyen productos mínimos como una página de marketing para probar que existe un problema. Un ejemplo de esto es el servicio Dropbox. Para probar que hay un interés en su servicio se hizo un video de la funcionalidad deseada mucho antes de escribir una línea de código. Este video se viralizó, demostrando que hay un problema real y hay muchas personas interesadas en que se solucione.<sup>24</sup>

Otros ejemplos de MVPs que se pueden llegar a utilizar en la etapa de Descubrimiento de Clientes es una simple entrevista, un prototipo de papel y lápiz o encuestas masivas. El único requisito es que tengan la capacidad de probar una hipótesis de problema o segmento de clientes.

Un MVP en la etapa de Validación de Clientes puede llegar a ser mucho más complejo que en la etapa anterior debido a que si se quiere probar una solución o un canal de distribución, es necesario un esfuerzo de ingeniería mayor.

Este tipo de MVP pueden ser tan simples como una aplicación sin backend que pruebe interacciones con el cliente hasta una aplicación totalmente funcional. Pero no porque haya soluciones más complejas hay que perder el foco en minimalismo del MVP, tiene que tener la funcionalidad justa para probar una hipótesis, no más ni menos.

## Conclusión

Lean Startup propone una forma radical de desarrollar startups, que mejora el éxito de las mismas, la innovación generada y el aprovechamiento del tiempo.

A través de casos de éxito se exemplificaron los principios esenciales de la metodología. Luego se describió el funcionamiento de Lean Startup, cómo complementa al Desarrollo de Clientes con XP, el bucle principal que guía el avance del aprendizaje en una startup, y cómo se utilizan productos mínimos para probar las hipótesis planteadas.

<sup>24</sup><http://theleanstartup.com/casestudies>

Después de ver tantas metodologías puede ser difícil procesar los conceptos de cada una y llevarlos a la práctica. Con el objetivo de allanar el camino desde la generación de ideas hacia la creación de una startup y sintetizar los conceptos dados realicé una guía paso a paso del proceso, exemplificándolo con un caso hipotético. Este resumen será el objeto de la próxima sección del Trabajo Final.

## Síntesis: Cómo transformar una idea en una startup

En el transcurso del Trabajo Final presenté muchas herramientas para emprendedores que se interconectan entre sí: lienzo de modelos de negocios, desarrollo de clientes y lean startup. Debido a que estos métodos tienen muchos conceptos interrelacionados puede ser difícil comprenderlos en su totalidad.

Para simplificar el proceso creación de startups voy a resumirlo en dos grandes etapas: descubrimiento de clientes y validación de clientes. En esta última sección del Trabajo Final explicaré la motivación de cada etapa, cómo realizarla y qué componentes integran a cada una.

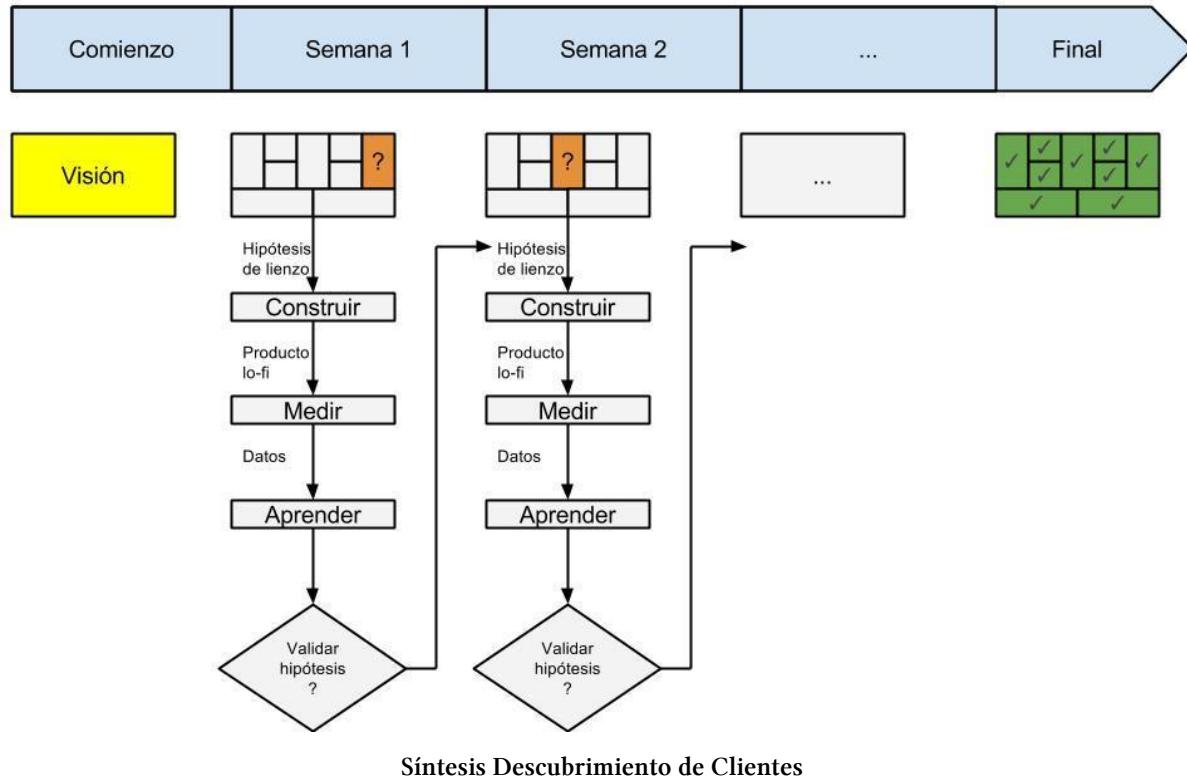
### Descubrimiento de Clientes

La primer etapa a completar a la hora de comenzar camino de idea hacia startup se llama Descubrimiento de Clientes. Resumiré la etapa gráficamente con un diagrama, después explicaré la motivación de la etapa—su objetivo principal—, luego detallaré cómo realizarla, y finalizaré con una descripción de cada componente que la integra.

### Diagrama

El siguiente gráfico sintetiza con simpleza la primera mitad de la creación de startups. Es esencial saber el punto de partida y de llegada en el Descubrimiento de Clientes, por eso destaco la atención en la visión inicial y el lienzo validado al finalizar la etapa.

# Descubrimiento de Clientes



## ¿Por qué seguir estos pasos?

Porque queremos resolver problemas específicos de personas reales. Este objetivo contrasta con suponer que existe un problema, lo cual puede llevar a miles de horas de trabajo perdidas en productos o servicios que no le interesan a nadie, que significaría un rotundo fracaso.

## Metodología

La forma de desarrollar el Descubrimiento de Clientes es:

- Elaborando hipótesis sobre elementos del lienzo de modelos de negocios
- Construyendo Productos Mínimos Viables, que tengan la funcionalidad justa y necesaria para probar una hipótesis
- Iterando semanalmente utilizando Programación Extrema
- Saliendo del edificio, juntando feedback de los clientes desde la primera semana
- Realizando pivots si fuera necesario, para aclarar las suposiciones sobre el modelo de negocios

## Elementos

**Comienzo: Visión** La innovación comienza formulando una visión a partir de una idea. Una visión es una imagen mental de cómo va a ser el futuro o cómo podría ser; tener visión es la habilidad de pensar acerca del futuro con imaginación o sabiduría <sup>25</sup>. Este es el tipo de pensamiento a gran escala requerido para innovar, y el punto de partida sugerido para toda startup.

Es fundamental buscar un problema específico de un segmento de clientes bien definido. Las dos primeras hipótesis a definir tienen como objetivo validar el Segmento de Clientes y Propuesta de Valor de una startup.

**Primer experimento** La primera iteración del bucle construir-medir-aprender siempre debe tener como objetivo validar la suposición realizada sobre el Segmento de Clientes al que va a apuntar la startup.

Cuando se confirme la hipótesis sobre los clientes hay que pasar al segundo experimento, sino hay que continuar iterando hasta encontrar un Segmento de Clientes.

**Segundo experimento** La segunda hipótesis a validar es la hipótesis de Propuesta de Valor. En este momento se cuenta con un grupo de personas con un problema concreto, y el objetivo es encontrar una solución adecuada para el mismo.

**Experimentos intermedios** El objetivo es validar **todas** las suposiciones del modelo de negocios planteado, por lo tanto hay que iterar hasta contar con un modelo de negocios probado.

**Final: Lienzo de modelo de negocios validado** Un modelo de negocios con cada bloque validado marca el fin de la etapa de Descubrimiento de Clientes. Recordemos que una startup es una organización que busca su modelo de negocios, la primer parada hasta encontrar un modelo de negocios es validar cada bloque del lienzo de modelo de negocios. El próximo paso en la vida de una startup es la Validación de Clientes.

## Validación de Clientes

La segunda etapa en la creación de startups es la Validación de Clientes. El análisis de esta etapa será equivalente al realizado con Descubrimiento de Clientes: resumiré el proceso en un diagrama, explicaré la motivación de la etapa, cómo realizarla y finalizaré con los componentes de la misma.

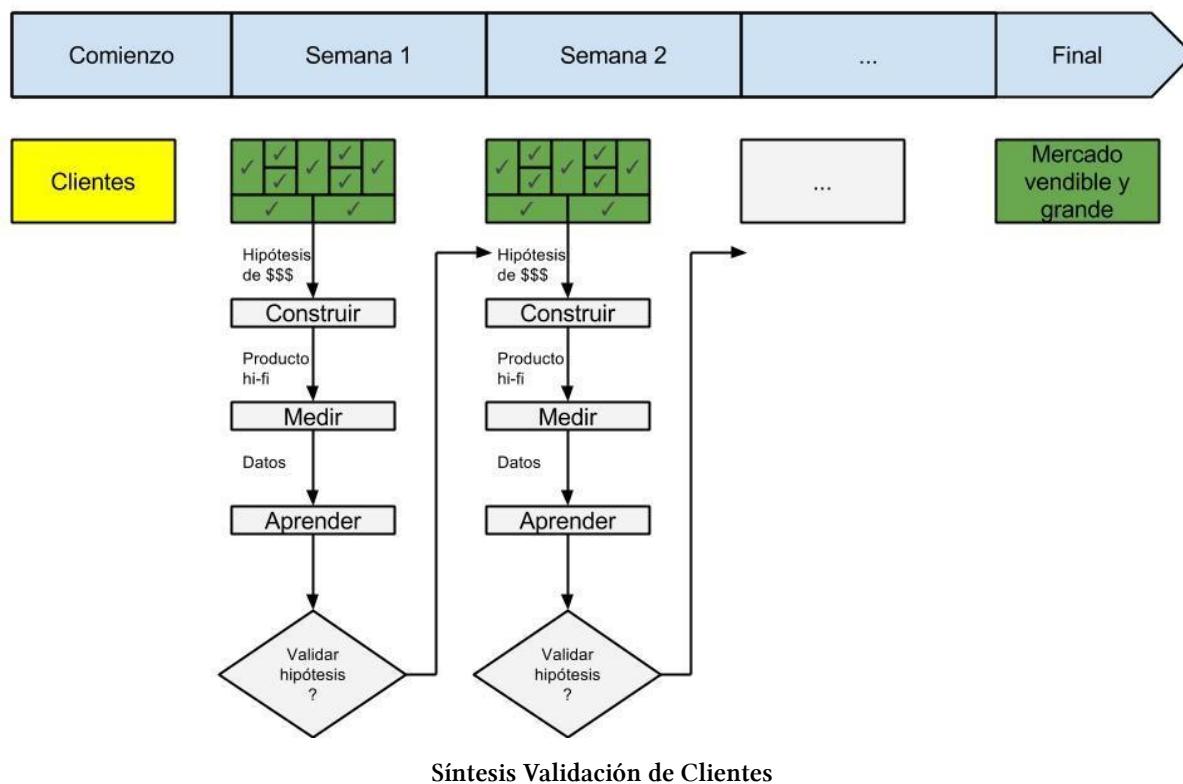
## Diagrama

Este diagrama determina con simpleza los pasos a seguir en la etapa de Validación de Clientes. El mismo comienza con clientes determinados, gracias al resultado del Desarrollo de Clientes.

<sup>25</sup><http://www.oxforddictionaries.com/definition/english/vision>

La etapa de Validación de Clientes culmina al poseer un mercado vendible y lo suficientemente grande para ser el objetivo de una compañía sustentable.

## Validación de Clientes



### ¿Por qué seguir estos pasos?

Porque hay productos que no generan ganancias, aunque solucionen problemas específicos de personas reales. Tener un producto útil, pero que no puede sustentar una compañía es una situación que—lamentablemente—no es sostenible en el tiempo, por eso en esta etapa el objetivo es encontrar una estrategia de ventas escalable y repetible.

### Metodología

Una estrategia de ventas escalable y repetible se encuentra de la siguiente manera:

- Elaborando hipótesis de crecimiento
- Construyendo Productos Mínimos Viables, más cercanos al producto final que los MVP de la etapa anterior porque ahora se cuenta con un mejor entendimiento de las necesidades del cliente
- Iterando semanalmente utilizando Programación Extrema

- Saliendo del edificio: Haciendo “ventas de prueba”, tratando que los clientes paguen por nuestro producto/servicio
- Realizando pivots si fuera necesario, para aclarar las suposiciones sobre el modelo de negocios

## Elementos

**Comienzo: Clientes - Modelo negocios validado** El punto de partida es un producto que soluciona el problema de un grupo de clientes, gracias a un modelo de negocios validado.

**Iteraciones** Las iteraciones en la Validación de Clientes tienen que comenzar con una hipótesis sobre la estrategia de ventas, porque el objetivo de esta etapa es encontrar el motor de crecimiento de la startup.

Las etapas de construcción, medición y aprendizaje dependen de las suposiciones sobre las hipótesis de crecimiento planteadas. Las métricas son especialmente importantes, qué tipo de métricas se plantea depende específicamente del motor de crecimiento buscado <sup>26</sup>.

**Final: Mercado vendible y grande** El fin de la Validación de Clientes se consigue luego de realizar las iteraciones sobre la estrategia de ventas, y validar el funcionamiento de un motor de crecimiento.

## Próximos pasos

El fin de la Validación de Clientes marca el fin de la vida de una startup. También marca la conclusión del objetivo propuesto por este Trabajo Final—detallar cómo transformar una idea en una startup—.

Completar la Validación de Clientes además determina el momento en que una startup entra en la etapa de ejecución, para convertirse en una compañía. Como ya mencioné previamente, la etapa de ejecución está fuera del alcance del presente Trabajo Final. Recomiendo al lector interesado en esta etapa referirse al libro “The Four Steps to the Epiphany” de Steve Blank, el cual es una excelente referencia para la etapa de Ejecución.

Para mejorar el entendimiento sobre la transformación de ideas en startups presentaré un caso práctico. La idea sobre la cual trabajaré en el Desarrollo de Ingeniería del Trabajo Final es:

*Aumentar la capacidad de entender código*

---

<sup>26</sup>Libro “The Lean Startup”, capítulo 10: Grow por Eric Ries

# 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 usará **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 comenzará con el desarrollo práctico de la etapa Descubrimiento de Clientes.

## Descubrimiento de Clientes

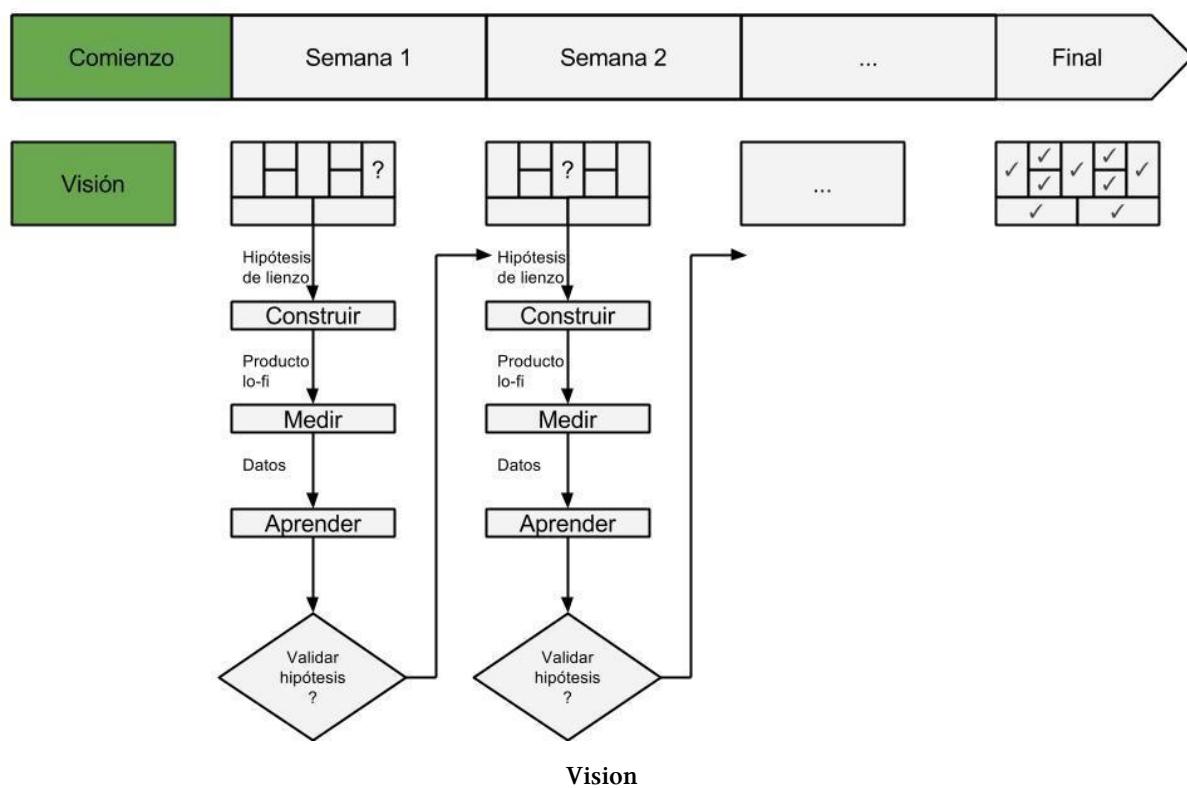
Ahora comienza la etapa de Descubrimiento de Clientes, en la que se busca validar un modelo de negocios. En esta fase hay que realizar experimento tras experimento hasta llevar a cero la gran lista de incógnitas con la que comienzan todas las startups.

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

## Descubrimiento de Clientes Visión



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 difícil comprender hasta el código propio
- Hacer Code Review
- Buscar material para reusar en código ajeno

Sin embargo, una cosa es segura: **Leer código es difícil**. 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 es la siguiente:

*Aumentar la capacidad de entender código*

También, después de incontables semanas de brainstorming, elegí el nombre de la startup que trataré por el resto del Trabajo Final:

[sonido de redoble de tambor]



“dip”

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 código, 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 continuación 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

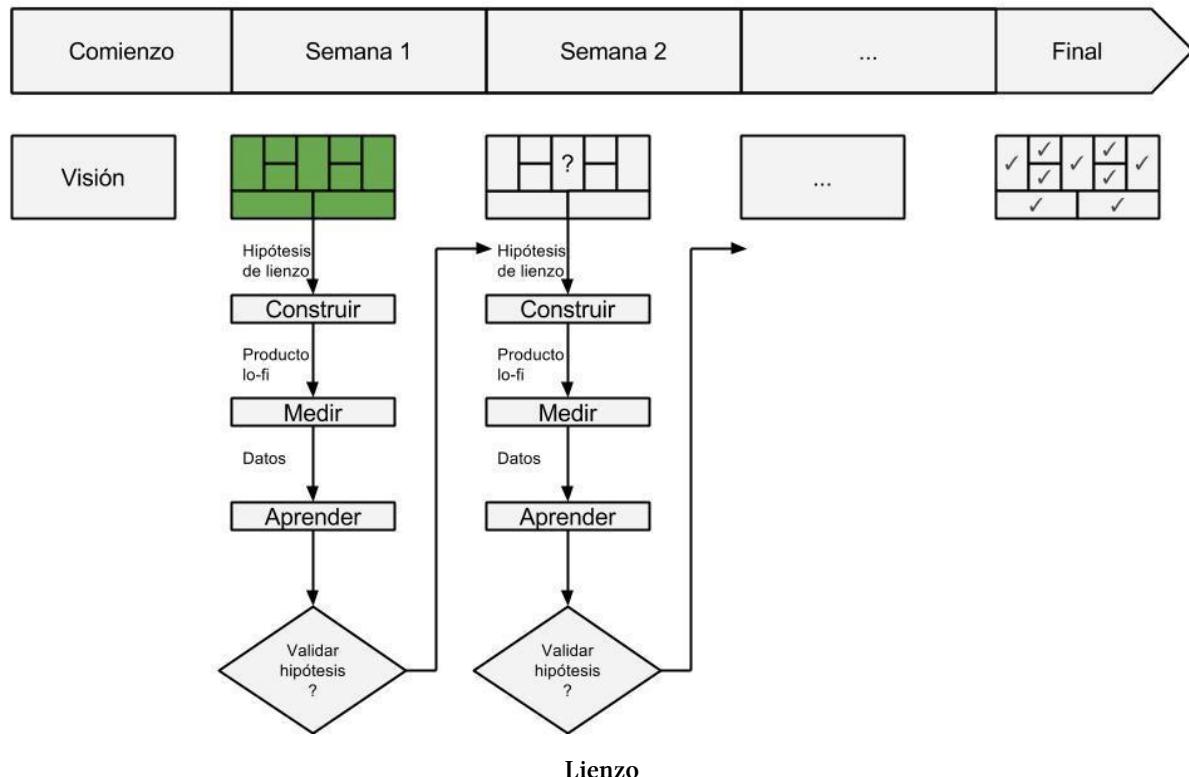
- Buscar el uso de palabra clave en el código
- Seleccionar código y ver la documentación asociada
- Hacer guías paso a paso sobre el código para que otros vean
- Escribir tests para fragmentos del código
- Refactorizar fragmentos de código
- Armar versiones simplificadas del código
- Remover una implementación 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.

## Lienzo

# Descubrimiento de Clientes Armado de Lienzo



Después de hablar sobre el Lienzo de Modelos de Negocios finalmente llegó la hora de dar un ejemplo de construcción de lienzo. En este capítulo deconstruiré la solución planteada

anteriormente en un modelo de negocios, descripto en las próximas páginas a través de sumario sobre cada elemento que lo compone.

Diseñaré el modelo de negocios partiendo del público objetivo. Para esto usaré una herramienta llamada Empathy Map, desarrollado por la compañía XPLANE<sup>27</sup> y referenciada en el libro “Business Model Generation”<sup>28</sup>. Partir del público objetivo permite construir un modelo de negocios mas fuerte porque el perfil del cliente guía el diseño de mejores Propuestas de Valor, Canales para llegar a él, y mejores Relaciones con Cliente.

La forma de diseñar el mapa de empatía de nuestro cliente objetivo es la siguiente:

1. Dar al cliente un nombre y características demográficas, como salario, estado civil, etc.
2. Refiriéndose al siguiente diagrama, usar un pizarrón o cartulina para construir un perfil del cliente recientemente nombrado preguntando y respondiendo las preguntas del diagrama.

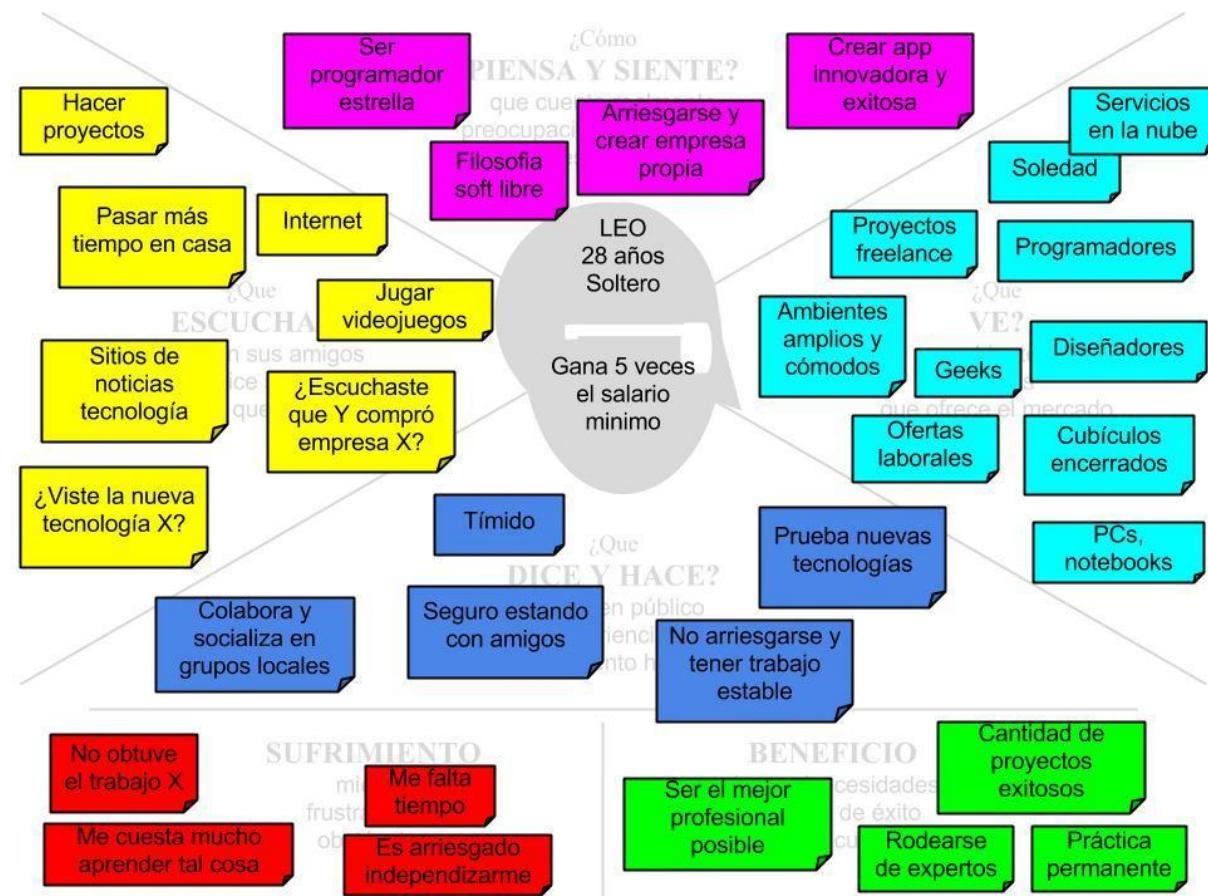


## Mapa de empatía de DIP

El mapa de empatía realizado para DIP quedó así:

<sup>27</sup><http://xplane.com/>

<sup>28</sup><http://www.businessmodelgeneration.com/>



Mapa de Empatía DIP

### Conocé tus usuarios como a vos mismo

Lo mejor que puede pasar a la hora de diseñar partiendo del público objetivo es ser parte del mismo. Por este y muchos otros motivos es fácil darse cuenta por qué es beneficioso construir una startup y a la vez ser el usuario objetivo de la misma.

Llegó el momento de describir los nueve bloques del modelo de negocios de DIP.

### Segmento de Clientes

DIP está orientado a programadores web, entre 18 y 30 años que quieran mejorar continuamente como profesionales. La mayoría de las personas a las que apunto son solteros, están trabajando y ganan 5 veces el salario mínimo.

### Propuesta de Valor

Usando DIP van a comprender fácilmente proyectos open source, así pueden contribuir a los mismos. Ofreceremos este entendimiento a través de guías digitales llamados “dips” que analizan código fuente de proyectos open source, y guían al usuario a través del código.

## Canales

La forma de distribución del servicio es a través de apps para smartphones y tablets, y a través del sitio dip.io.

## Relaciones con Clientes

Mantendremos la relación con nuestros clientes a través de un blog técnico que sea de interés para ellos, y también con la creación de comunidades de contribuidores de proyectos libres.

## Flujos de Ganancias

Utilizaré un modelo de suscripciones mensuales, el cual dará a los usuarios acceso completo a todos los dips.

## Recursos Clave

Los elementos claves que hacen funcionar al negocio son tres: los usuarios, los dips que les proveen valor y el hecho de que todo el software utilizado por la startup sea código libre.

## Actividades Clave

El armado de dips por especialistas, idealmente expertos en el código; y el desarrollo de las apps son las actividades clave.

## Sociedades Clave

Github es un socio clave porque prácticamente todo proyecto open source está hosteado allí, y proveen una API para interactuar con ellos.

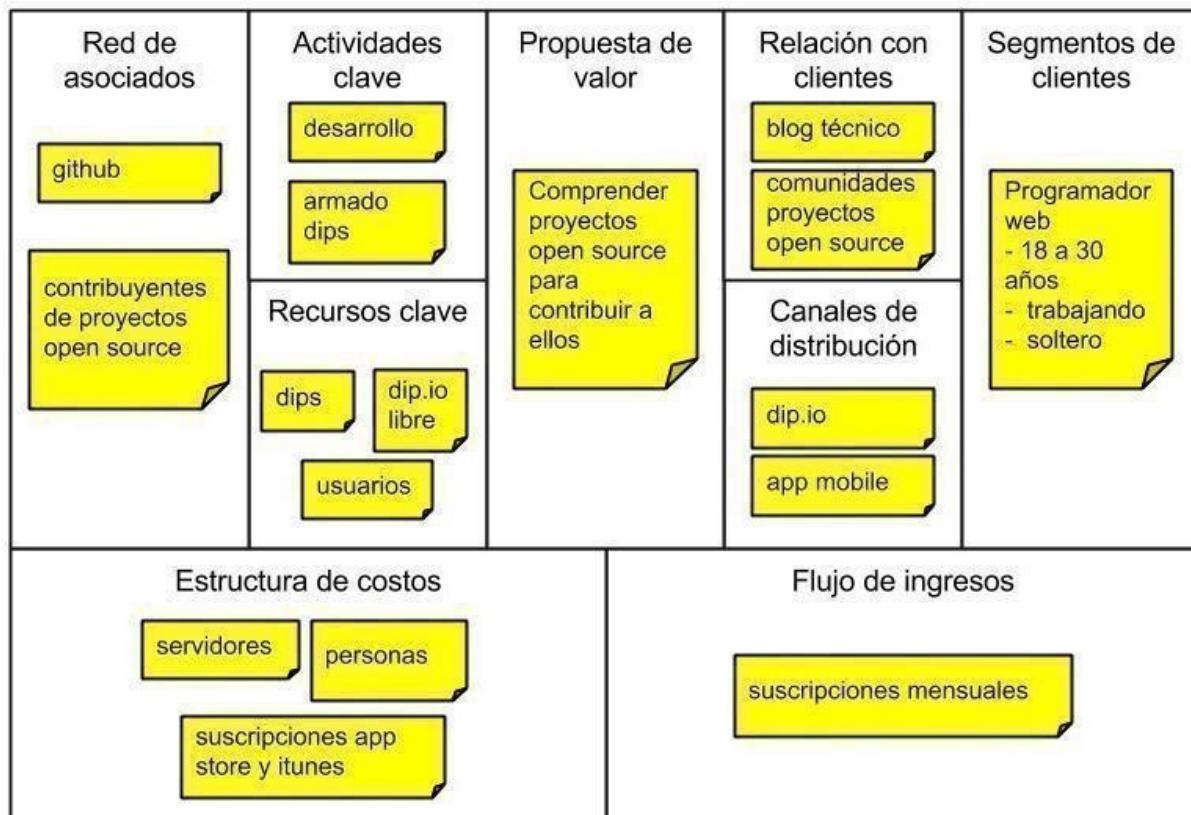
## Estructura de Costos

La startup cuenta con costos típicos de una empresa web/mobile: servidores, suscripción a la Play Store de Google y a iTunes de Apple. También hay que incluir a los salarios de personas y suscripciones a servicios y herramientas de desarrollo.

## Lienzo de DIP

Sintetizando el modelo de negocios de DIP en una sola imagen:

# DIP



Lienzo dip

Es importante recordar que este modelo de negocios es solo una primera aproximación llena de suposiciones, que pueden ser acertadas o erradas, y sólo sabremos eso con el Desarrollo de Clientes.

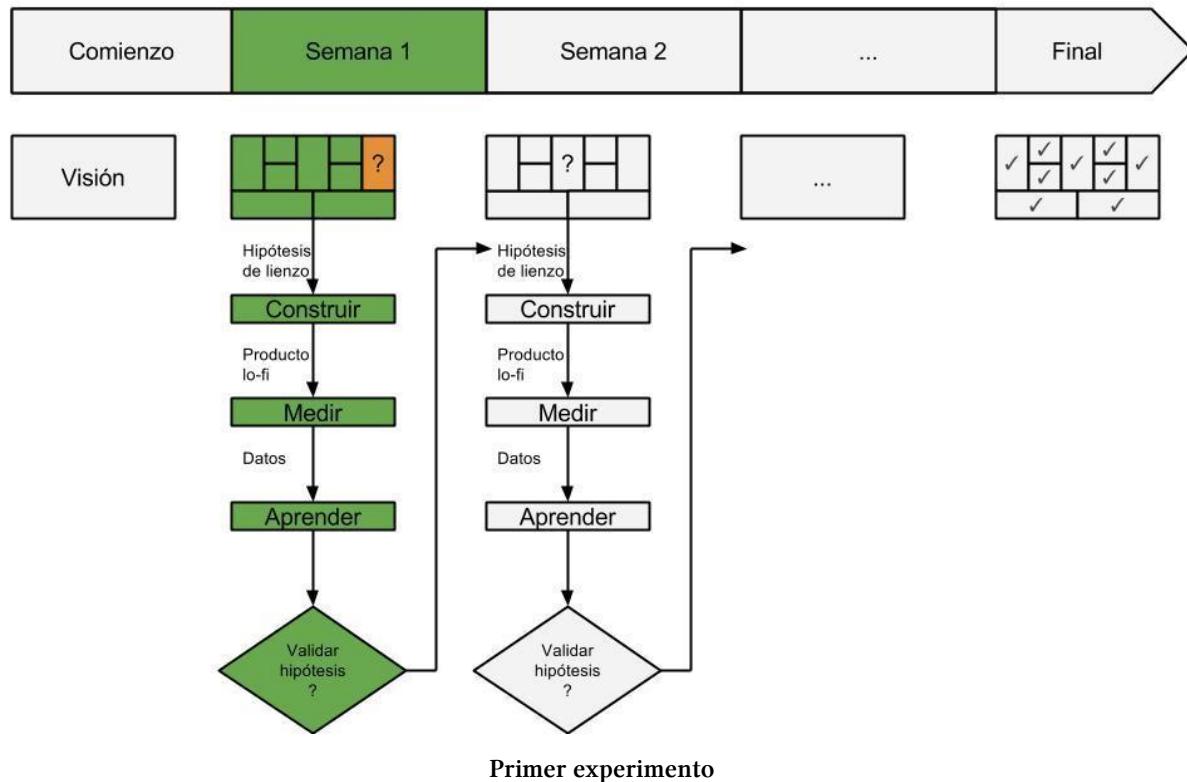
## Resumen

Con una visión formada y un lienzo de modelo de negocios listo, no queda nada que impida el comienzo de la parte mas emocionante: armar el producto y salir del edificio. En el próximo capítulo comenzaré con la primera fase de construcción.

## Primer Experimento

# Descubrimiento de Clientes

## Primer Experimento



Finalmente llegó el momento de empezar la validación del modelo de negocios, y convertir todas las suposiciones en certezas. Este proceso generalmente es largo y laborioso, a menos que seas parte de la minoría que saca la lotería de usuarios (Facebook, Instagram, Twitter, etc).

## Hipótesis

¿Por dónde empezar? Tenemos un lienzo de negocio lleno de hipótesis sin validar, sin embargo generalmente el punto de partida será siempre el mismo: hay que tener certeza que la startup encara un problema real de los clientes.

Por lo tanto la primera hipótesis en la que trabajaré es la del Segmento de Clientes:

A los programadores web les resulta difícil comprender código.

## Construir

¿Cómo validar esta hipótesis? Llegó la hora de construir un producto que valide la hipótesis planteada. Pero es importante tener en cuenta que se construirá un MVP—un producto que tenga la funcionalidad mínima y necesaria para probar la hipótesis—no hay que desperdiciar recursos en productos con funcionalidad innecesaria por ahora.

Generalmente los ingenieros tenemos la tendencia de construir soluciones y perfeccionarlas hasta el más mínimo detalle. Esta forma de pensar nos puede llevar a construir soluciones perfectas, pero que desafortunadamente nadie usa porque no solucionan ningún problema real.

Es bueno saber programar no es la única opción, podemos construir soluciones que no involucren código y también nos provean datos para validar o rechazar hipótesis, por ejemplo:

- podríamos armar un video que plantee el problema a solucionar y después medir su éxito
- podríamos relizar entrevistas con posibles usuarios y detectar si realmente cuentan con el problema planteado en la hipótesis
- podríamos hablar con potenciales clientes y así entender qué necesitan realmente

También está la opción de escribir sólo el código necesario, esto puede ser una simple pero efectiva página web que plantee el problema a validar y mida el éxito de la misma para validar o no la hipótesis. Ése es el MVP que voy a construir en esta etapa: una página de marketing.

**MVP** Construiré una página que despierte el interés del público objetivo y recolecte emails de interesados. Basta con que la misma plantea que **leer código es difícil**, sin detallar una solución específica al problema. El contenido final de la página será el siguiente:



### Decodificador de HTML, CSS y js

¿Necesitas leer varias veces el código fuente para entenderlo?

¿Querés saber el por qué de cada línea de código?

¿Te serviría leer interpretaciones ajenas para entender mejor el código?

Leé código en dip y convertite en un  
Shakespeare del frontend

¿Suena interesante? Dejanos tu email y sé de los primeros diperos.

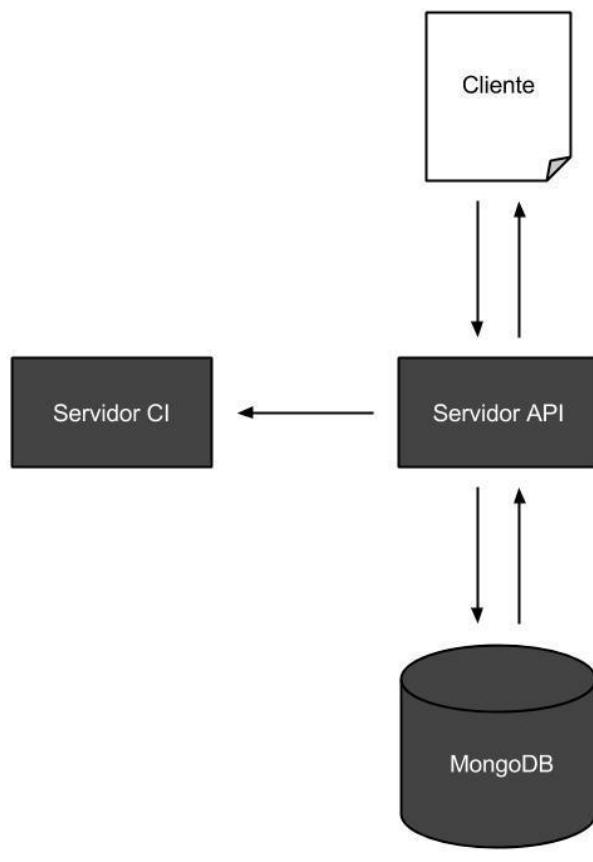
Tu dirección de email

Vamos

### Landing page de dip

La página cuenta con un cliente y un servidor. Utilicé un solo lenguaje para la construcción de todos los componentes de la misma—cliente, servidor y base de datos—javascript.

- Cliente: desarrollado con el framework AngularJS
- Servidor: API hecha en NodeJS
- Base de datos: MongoDB, ideal para startups por el hecho de no requerir esquemas estáticos

**Arquitectura MVP**

A pesar de la simpleza de la solución planteada, tiene un objetivo importante—recolectar emails de interesados— y hay que asegurar que haga esto perfectamente. Para esto armaré un conjunto de pruebas funcionales y unitarias que verifiquen que página hace lo que tiene que hacer.

**Pruebas automatizadas** Probaré la funcionalidad de la aplicación con dos tipos de pruebas, pruebas de integración (end-to-end o E2E en inglés) y pruebas unitarias. Las pruebas E2E verifican que la aplicación funciona desde el punto de vista del usuario, y las pruebas unitarias verifican que cada unidad (clase, función, etc) funciona.

**Pruebas E2E** La página cuenta con un elemento en el que el usuario ingresa su email y un botón que lo envía al servidor. Para verificar que la página muestra una confirmación al usuario cuando ingrese su email correctamente realizaré una prueba E2E.

Para probar esto utilizaré Selenium como herramienta de automatización, la cual puede realizar las mismas acciones que una persona a la hora de navegar sitios web. A continuación muestro cómo ingresar un email y apretar el botón “Vamos” a través de Selenium.

```

1 describe('Pagina', function() {
2   it('deberia mostrar confirmacion al recibir un email', function() {
3     browser.get('#/');
4
5     var entradaEmail = element(by.id('email'));
6     var botonVamos = element(by.id('vamos'));
7     var confirmacion = element(by.id('confirmacion'));
8
9     entradaEmail.sendKeys('nombre@example.com');
10    botonVamos.click();
11    expect(confirmacion.isDisplayed()).toBe(true);
12    expect(entradaEmail.isDisplayed()).toBe(false);
13    expect(botonVamos.isDisplayed()).toBe(false);
14  });
15 });

```

Con esta prueba queda verificada la funcionalidad desde el punto de vista del usuario, pero ¿cómo verificar que al apretar el botón “Vamos” se envía un pedido HTTP al servidor? Para eso están las pruebas unitarias.

**Pruebas unitarias** En una arquitectura modelo-vista-controlador un controlador tiene como responsabilidad ser el “pegamento” entre la vista y los modelos, es un mero intermediario. Por lo tanto una prueba unitaria de controlador debe verificar que el mismo llama de manera correcta a los modelos, y esto se puede verificar tranquilamente utilizando “mocks”—objetos temporales que tienen la funcionalidad de objetos reales—no hay necesidad de usar a los objetos reales porque queremos verificar el comportamiento de una unidad a la vez.

La prueba unitaria del controlador es la siguiente:

```

1 describe('Controller: MainCtrl', function () {
2   var MainCtrl,
3     scope,
4     ServicioEmailMock;
5
6   beforeEach(module('launchApp', function ($provide) {
7     ServicioEmailMock = jasmine.createSpyObj('ServicioEmail', ['enviar']);
8     $provide.value('ServicioEmail', ServicioEmailMock);
9   }));
10
11  beforeEach(inject(function ($controller, $rootScope) {
12    scope = $rootScope.$new();
13    MainCtrl = $controller('MainCtrl', {
14      $scope: scope
15    });
16  }));
17

```

```
18 it('deberia enviar email usando ServicioEmail', function () {
19   scope.email = 'nombre@example.com';
20   scope.enviarEmail(scope.email);
21   expect(ServicioEmailMock.enviar).toHaveBeenCalledWith(scope.email);
22 });
23 });
```

Para probar que el modelo de la aplicación—el servicio email—funciona voy a utilizar una nueva especie de objeto temporal, un backend falso. El mismo evita que la aplicación, al ser probada, haga pedidos HTTP innecesarios al servidor que ralenticen innecesariamente nuestras pruebas.

La forma de utilizar este backend falso para probar que el servicio email funciona correctamente es la siguiente:

```
1 describe('Service: ServicioEmail', function () {
2   var ServicioEmail,
3     httpBackend,
4     API_URL = 'http://dip-api.herokuapp.com';
5
6   beforeEach(module('launchApp'));
7
8   beforeEach(inject(function (_ServicioEmail_, $injector) {
9     ServicioEmail = _ServicioEmail_;
10    httpBackend = $injector.get('$httpBackend');
11  }));
12
13  afterEach(function() {
14    httpBackend.verifyNoOutstandingExpectation();
15    httpBackend.verifyNoOutstandingRequest();
16  });
17
18  it('deberia guardar email en el servidor', function () {
19    var email = 'nombre@example.com';
20    httpBackend.expectPOST(API_URL + '/email', { 'email': email }).respond(201, \
21    '');
22    ServicioEmail.enviar(email);
23    expect(ServicioEmail.estado).toBe(null);
24    httpBackend.flush();
25    expect(ServicioEmail.estado).toBe('exito');
26  });
27
28  it('deberia indicar si hubo un error', function () {
29    var email = 'nombre@example.com';
30    httpBackend.expectPOST(API_URL + '/email', { 'email': email }).respond(404, \
31    '');
32    ServicioEmail.enviar(email);
```

```

33     expect(ServicioEmail.estado).toBe(null);
34     httpBackend.flush();
35     expect(ServicioEmail.estado).toBe('error');
36   });
37 });

```

El resto del código del frontend se puede ver en la siguiente dirección:

[Frontend en github<sup>29</sup>](https://github.com/J JulianMayorga/dip-primer-mvp-frontend)

**Servidor** El servidor está hosteado en un dominio diferente al cliente por motivos de modularidad. Al hacer esto hay un desacoplamiento entre cliente y servidor, y una interfaz REST definida para la comunicación entre ambos.

La API REST es muy simple, y cuenta con sólo dos funciones:

1. responder a los pedidos realizados a “/email” con un código de confirmación
2. guardar en la base de datos el correo que viene en el cuerpo del pedido.

A continuación mostraré cómo probar estos dos requisitos:

```

1 var app = require('../index.js'),
2 request = require('supertest'),
3 expect = require('chai').expect,
4 mongodb = require('mongoskin');
5
6 var MONGODB_URL = process.env.MONGODB_URL ||
7   'mongodb://localhost:27017/dip-testdb';
8
9 describe('POST /email', function () {
10
11   var db = mongodb.db(MONGODB_URL, {safe: true});
12
13   after(function (done) {
14     db.collection('users').drop(done);
15   });
16
17   it('deberia responder con codigo 200', function (done) {
18     request(app)
19       .post('/email')
20       .send({email: 'nombre@example.com'})
21       .expect(200, done);
22   });
23

```

---

<sup>29</sup><https://github.com/J JulianMayorga/dip-primer-mvp-frontend>

```

24  it('deberia guardar email en la base de datos', function (done) {
25    var testEmail = 'miguel@example.com';
26
27    request(app)
28      .post('/email')
29      .send({email: testEmail})
30      .end(function (err, res) {
31        db.collection('users').findById(res.body._id, function (err, res) {
32          expect(res.email)
33            .to.equal(testEmail);
34          done();
35        });
36      });
37    });
38  });
39 });

```

El código fuente completo de la API, al igual que el del cliente, se encuentra hosteado en github. La dirección del mismo es la siguiente:

[API en github<sup>30</sup>](#)

## Medir

Una vez que se libera el MVP al mundo hay que analizar los resultados del experimento. Puede ser tentador medir cantidad de visitas al mes, tiempo total en la aplicación, tiempos de respuesta, etc. Trackear estas métricas puede ser divertido, pero no es el objetivo del experimento, lo único que interesa son la cantidad de personas que mostraron interés en la solución al dar sus emails.

En la etapa de medición hay que plantearse la pregunta ¿cuántos emails consideramos un éxito? ¿10 emails? ¿50? ¿1000? Estos números, obviamente, dependen de muchos factores propios de cada experimento pero es importante definir un número. Éste número va a ser la vara que tildará de éxito o fracaso al experimento.

## Aprender

En esta etapa hay que analizar los datos que surgieron del experimento y tomar una decisión con respecto a la próxima hipótesis a medir. ¿Es necesario realizar otro experimento para validar la hipótesis planteada? ¿O hay suficientes pruebas para decir que contamos con un problema real, y estamos listos para validar la solución?

## Paso siguiente

Para exemplificar el proceso de Desarrollo de Clientes, vamos a suponer que la página de marketing fue todo un éxito.

Ahora que sabemos que hay programadores interesados en la lectura de código vamos a armar una solución real: **Un editor de HTML, CSS y JavaScript con enfoque en lectura de código**.

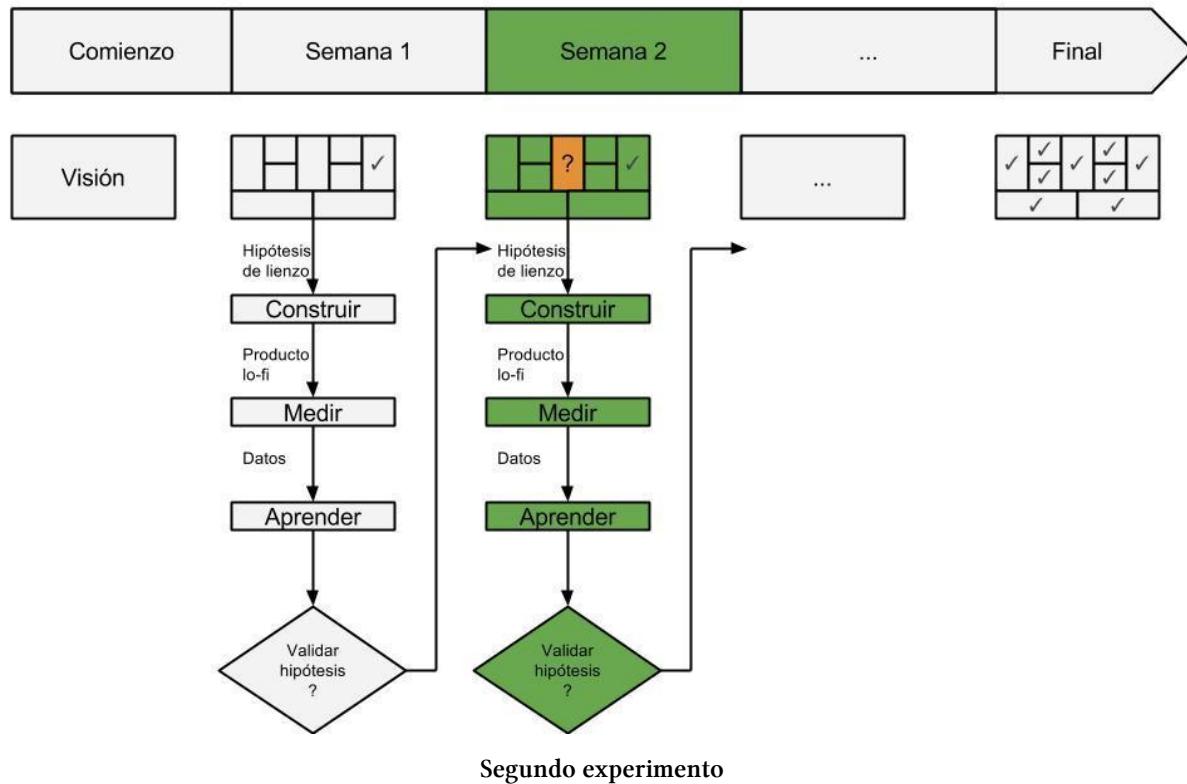
---

<sup>30</sup><https://github.com/J JulianMayorga/dip-primer-mvp-api>

## Segundo Experimento

# Descubrimiento de Clientes

## Segundo Experimento



El paso que le sigue a la validación del problema es la validación de la solución. En el experimento anterior definí un experimento y validamos el interés del segmento de clientes, ahora voy a diseñar un producto mínimo viable para validar la solución propuesta. Este es sólo el principio de un largo proceso de descubrimiento, en el que construiré y deconstruiré numerosas soluciones hasta encontrar una que encaje con las necesidades del cliente.

Los detalles de la hipótesis, construcción, medición y aprendizaje de este experimento están detallados a continuación.

### Hipótesis

Así como en el experimento anterior definí una hipótesis relacionada al Segmento de Clientes, en esta etapa voy a definir una hipótesis relacionada a otro bloque del Lienzo de modelos de negocio: la Propuesta de Valor.

Hay una hipótesis de propuesta de valor definida en el lienzo, pero en esta etapa voy a elegir un subconjunto de la misma para tener más precisión a la hora de validarla. La hipótesis actual es “El valor que ofrece la startup es comprender proyectos open source para contribuir a ellos”, y la misma puede deconstruirse en tres hipótesis de valor independientes:

1. Comprender proyectos representa valor para programadores web
2. Programadores web tienen interés en open source
3. Contribuir a proyectos open source les da valor a programadores web

La primer sub-hipótesis que voy a validar a través de una solución es la primera.

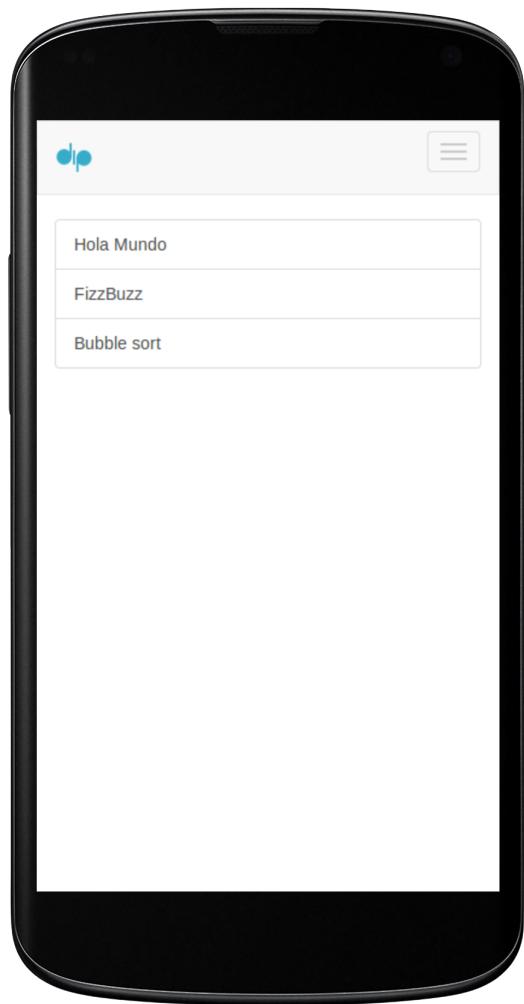
Comprender proyectos representa valor para programadores web

## Construcción

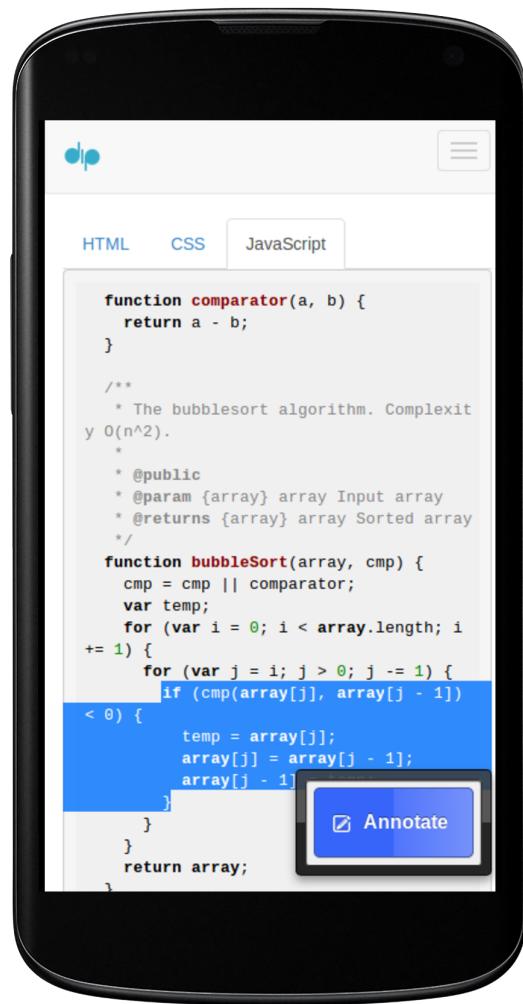
Código bien comentado es código de calidad, y unos buenos comentarios ayudan a entender código; pero la forma de encarar código, ya sea de calidad o no, varía dependiendo de la experiencia y habilidad de cada programador. No hay forma que los comentarios den información útil a todo el tipo de personas que pueden leerlo. ¿O si hay forma?

La solución que voy a construir es una capa adicional al código, que tenga comentarios creados por y para la comunidad para comprender mejor el código: preguntas, aclaraciones, enlaces, etc.

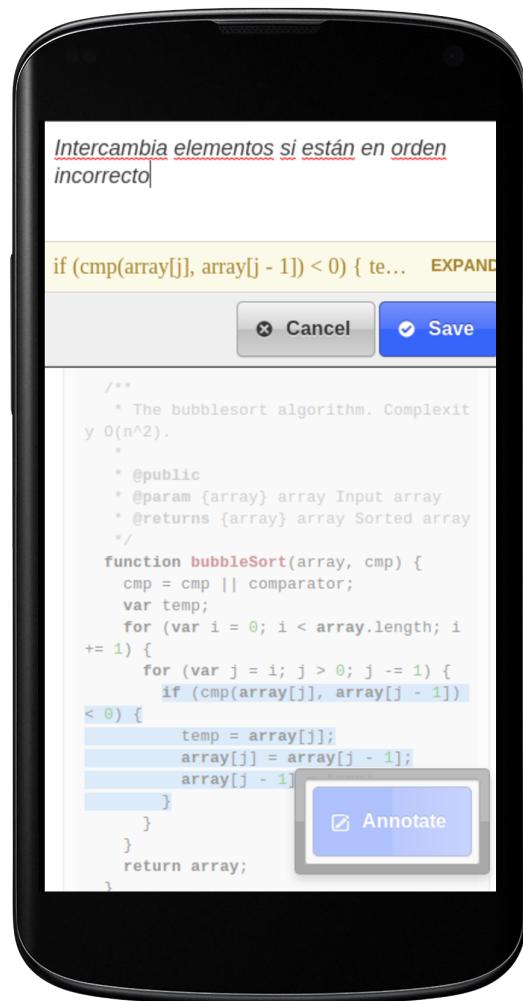
Dicen que una imagen vale más que mil palabras, así que espero que con cuatro imágenes se entienda perfectamente la idea de la aplicación:



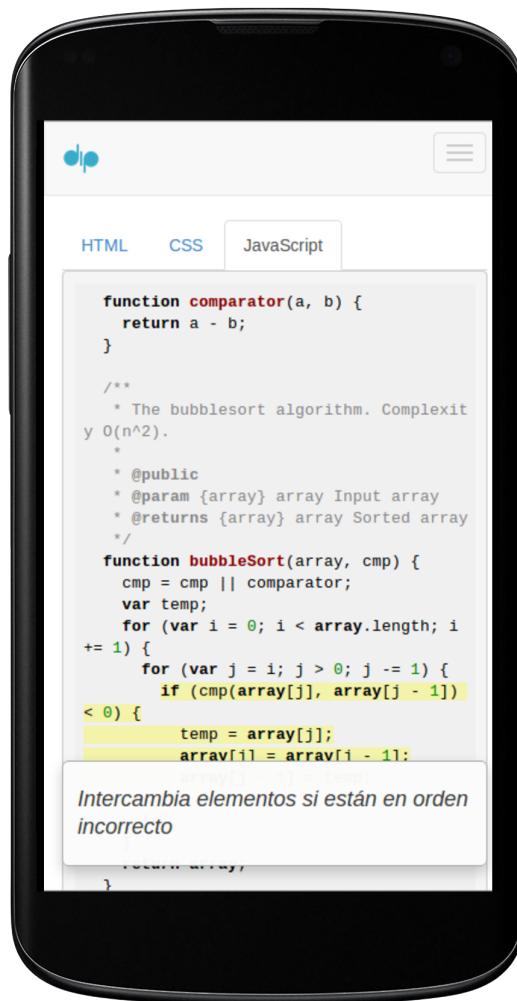
“Pagina inicial de la app: Lista de código fuente”



“El usuario selecciona el contenido a comentar”



“El usuario comenta el código”



“Al tocar el texto seleccionado se muestra el comentario”

La forma de construir el MVP actual es escribiendo las pruebas automatizadas antes que el código. Por motivos de espacio sólo voy a mostrar las pruebas a nivel de usuario, o E2E. El código fuente completo de la aplicación se encuentra [aquí<sup>31</sup>](#).

**Tests E2E** La página inicial de la app debería mostrar tres dips inicialmente y debería pasar a la vista de dip al cliquear uno, y eso es exactamente lo que voy a automatizar con el siguiente código:

---

<sup>31</sup><https://github.com/JulianMayorga/dip-segundo-mvp>

```
1 describe('App', function() {
2   beforeEach(function () {
3     browser.get('/#/');
4   });
5
6   describe('Pagina inicial', function () {
7     it('deberia mostrar tres dips inicialmente', function () {
8       var dips = element.all(by.css('.dips a'));
9       expect(dips.count()).toBe(3);
10    });
11
12    it('deberia pasar a la vista de dip al cliquear uno', function () {
13      var primerDip = element(by.repeater('dip in dips').row(0).column('{{dip.\n14 nombre}}'));
15      primerDip.click();
16      expect(browser.getCurrentUrl()).toContain('/dip/1');
17    });
18  });
19
```

En la vista de dip se debería poder hacer anotaciones sobre el código; debería tener tabs para HTML, CSS y JavaScript; y debería mostrar el código fuente de cada dip. El código para probar eso es el siguiente:

```
1 describe('Pagina de dip', function () {
2   var primerDip;
3
4   beforeEach(function () {
5     primerDip = element(by.repeater('dip in dips').row(0).column('{{dip.nomb\\
6 re}}'));
7     primerDip.click();
8   });
9
10  it('deberia inicializar las anotaciones en la vista de dip', function() {
11    var wrapper = element(by.css('.annotator-wrapper'));
12    expect(wrapper.isPresent()).toBe(true);
13  });
14
15  it('deberia tener tabs en la vista de cada dip', function () {
16    var dips = element(by.css('.nav-tabs'));
17    expect(dips.isPresent()).toBe(true);
18  });
19
20  it('deberia mostrar contenido del dip', function () {
21    var js = element(by.css('.js .hljs'));
22    expect(js.getText()).not.toBe('');
23  });
24
```

```
24     });
25 });


```

## Medir

El objetivo del experimento es medir la comprensión del código fuente, la cual es una métrica no tan exacta como el número de inscriptos, pero igualmente es medible. La forma que elegí para medir esto es a través de una simple pregunta hecha a cada usuario, la cual es ¿qué hace el código fuente que acabas de leer?

**Quiero resaltar la importancia de hablar cara a cara con los usuarios para aprender sobre ellos.** Para obtener respuestas a la pregunta sobre el código fuente que van a leer los usuarios podría hacer un simple formulario que envíe las respuestas al servidor, analizarlas una por una y llegar a una conclusión al final del experimento. Pero la forma de medir la comprensión de los clientes va a ser preguntándoles cara a cara, ya sea en persona o por skype. Haciendo esto no sólo voy a salir con una idea clara de la efectividad de la solución que estoy probando, sino que voy a sentir si la experiencia del usuario es la que yo deseo.

Hablando con los usuario en etapas tempranas del producto es una muy buena forma de ganar aprendizajes clave que ayuden a darle forma al producto en desarrollo.

## Aprender

Luego de realizar suficientes entrevistas con usuarios interesados y medir qué tanto la solución los ayuda a comprender mejor el código, es la hora de aprender sobre la hipótesis planteada. Más allá de la conclusión a la que llegue en este experimento, hay mucha funcionalidad posible que puede ayudar a programadores a comprender mejor el código. Por eso hay que seguir experimentando, planteando hipótesis y construyendo la funcionalidad mínima que provea aprendizajes clave.

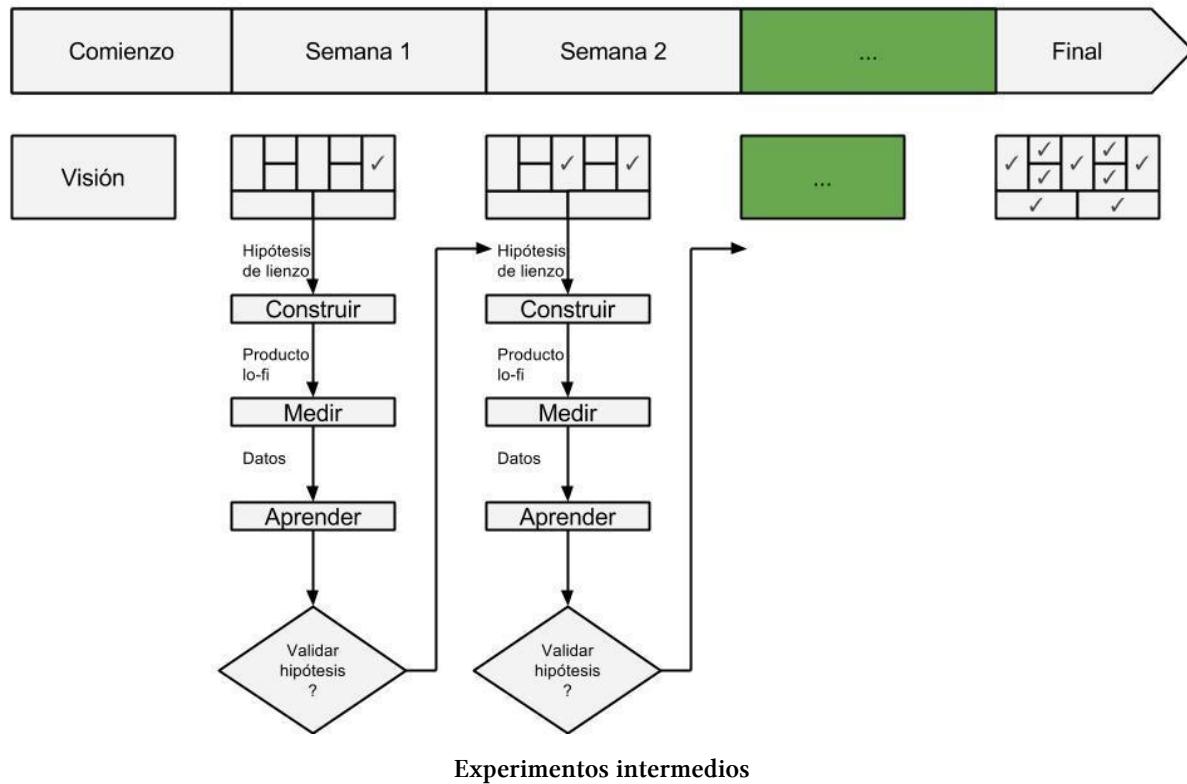
## Paso siguiente

Es necesario seguir iterando hasta encontrar una solución adecuada al problema planteado. Luego de eso hay que validar el resto de los bloques del lienzo de modelo de negocios, el cual es el objeto de la próxima sección.

## Experimentos intermedios

# Descubrimiento de Clientes

## Experimentos intermedios



Hay que hacer numerosos experimentos e iteraciones en la etapa de Descubrimiento de Clientes, siempre recordando que el objetivo no sólo es construir un servicio/producto, sino **encontrar un modelo de negocios**.

Iterar sobre cada bloque—realizar hipótesis, MVPs y mediciones—ayuda a mantener un ciclo de feedback con clientes permanente. Hay que *salir del edificio* para optimizar el uso de recursos y aprender más sobre los usuarios.

Para clarificar la búsqueda de modelo de negocios, voy a ejemplificar los aspectos del lienzo de dip que habría que validar.

### Hipótesis a validar

**Segmento de clientes** La primera iteración de dip estuvo enfocada en validar un problema del segmento de clientes, por lo tanto este aspecto debería estar validado.

**Propuesta de valor** La segunda iteración ejemplificada en el trabajo tuvo como objetivo encontrar una solución que se adecúe al problema del segmento de clientes, pero sólo validó una

parte de ella, es necesario continuar experimentando hasta encontrar un producto que encaje perfectamente con el problema encontrado.

Las hipótesis que falta validar son:

- Programadores quieren comprender proyectos completos, no sólo fragmentos de código
- Programadores quieren explorar issues de proyectos open source, armar dips para comprender el código a modificar y una vez comprendido contribuir al proyecto con pull requests

**Canales** Con respecto a los canales de distribución la principal suposición a validar es:

- Aplicaciones móviles y web son útiles para consumir contenido

**Relaciones con clientes** Hay diferentes formas de conectarse con programadores web, las suposiciones que planteé en el lienzo de dip y hay que validar son:

- Un blog técnico serviría para potenciar el armado de comunidad de desarrolladores que pretende dip
- Crear comunidades alrededor de proyectos open source ayudaría a conectarse con usuarios de dip

**Flujos de ganancias** Las suscripciones mensuales planteadas en el lienzo inicial son lo más adecuado? Es necesario probar que eso es cierto, y explorar otras opciones si fuese necesario.

- Quizás es mejor usar una estrategia de publicidad para generar ingresos
- O quizás armar un mercado de desarrolladores y personas que ofrecen trabajo sería mejor

**Recursos clave** A medida que vamos validando otras partes del negocio vamos a identificar y rechazar diferentes recursos clave. Hay que estar atentos a estos cambios para identificar qué elementos le dan a la startup una ventaja competitiva.

**Actividades clave** El desarrollo de aplicaciones es una actividad clave debido a la naturaleza de las startups web/mobile, pero es necesario contemplar otras actividades esenciales.

- Armar dips que ayuden a entender proyectos es una forma de incentivar a la comunidad a crear sus dips propios
- Armar exploraciones de proyectos open source y venderlos como libros quizás sea una actividad clave, hay que probar esto

## Sociedades clave

- Integrar con github o no?
- Alguna otra sociedad sería buena?

## Estructura de costos

- Que servicios desarrollar como propios y que servicios alquilar?
- Utilizar servidores propios o la nube?
- Utilizar un backend propio o un servicio como firebase, para poder concentrarse sólo en la experiencia de usuario?

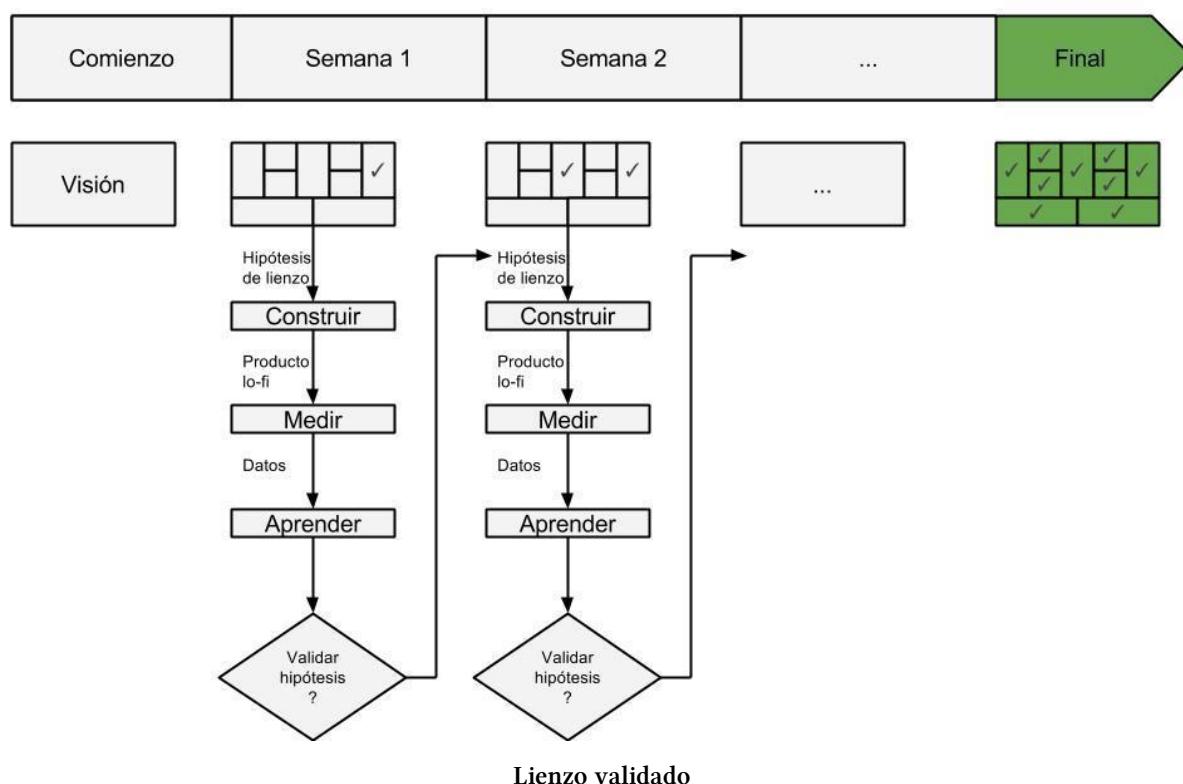
## Próximos pasos

Una vez que haya certezas sobre cada bloque del modelo de negocios es el momento de analizar el lienzo completo y validado, lo que trataré en la próxima sección.

## Lienzo Validado

# Descubrimiento de Clientes

## Lienzo validado



Encontrar un modelo de negocios que funcione es un logro importante en la vida de una startup, ya que marca el final de la etapa de Descubrimiento de Clientes.

Tener usuarios que encuentran valor en el producto ofrecido es algo que no todas las startups pueden tener, por lo tanto hay que aprovechar esto y trabajar sobre ellos en la próxima etapa: la Validación de Clientes, en la cual hay que iterar hasta encontrar una estrategia de ventas que permita obtener ganancias de estos usuarios satisfechos.

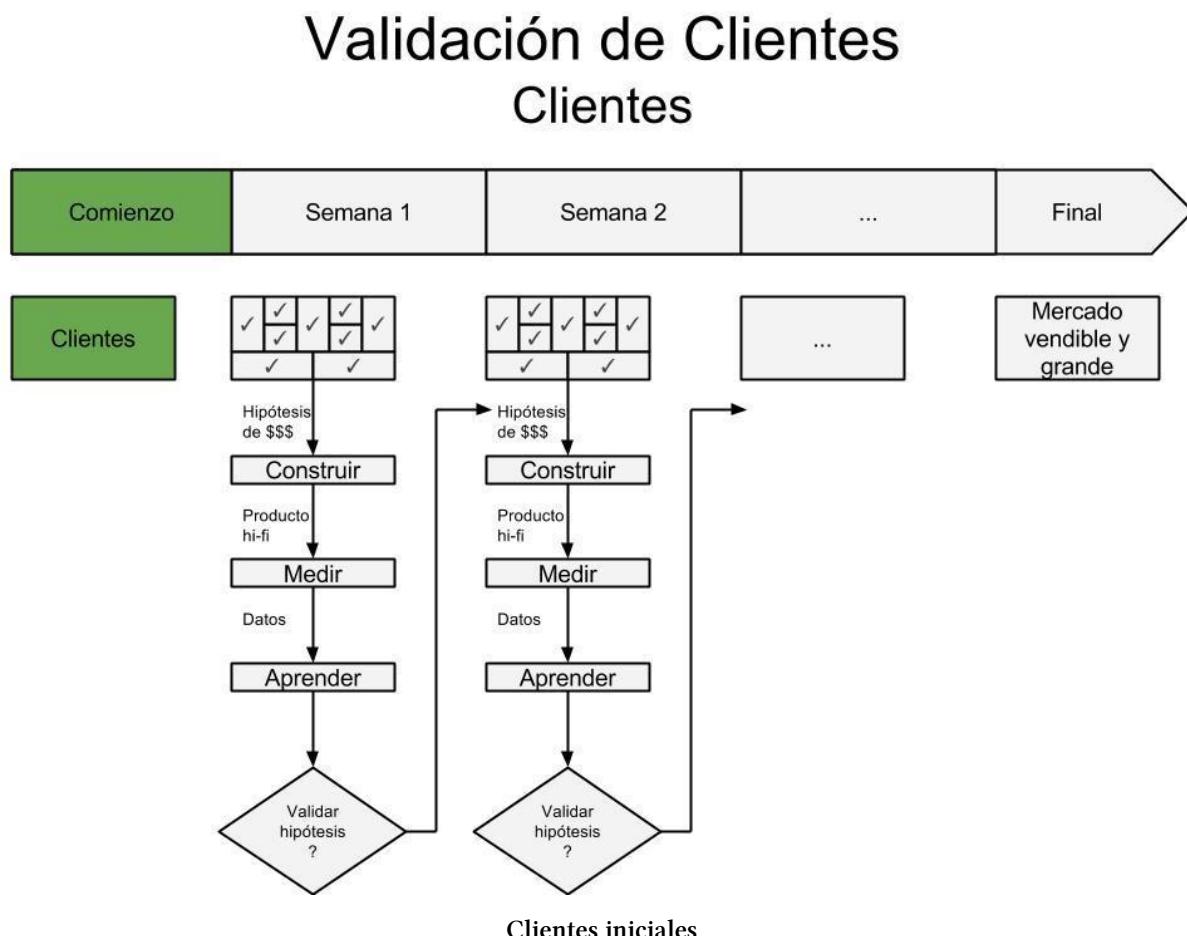
## Validación de Clientes

En la Validación de Clientes se realiza el proceso que busca tener una estrategia de ventas repetible y escalable. Esto significa que no sólo hay que encontrar la forma correcta de vender el producto/servicio a los clientes, sino que hay que asegurarse que ésta venta se pueda hacer a gran escala a futuro, cuando la startup sea una compañía.

La manera de encontrar esta estrategia de ventas es la misma que se utilizó para encontrar un modelo de negocios: planteando hipótesis, realizando experimentos y saliendo del edificio para validar las suposiciones.

En las próximas secciones voy a mostrar con ejemplos las estrategias de venta posibles para startups. Al finalizar la etapa de Validación de Clientes finaliza el ciclo de vida de una startup, la cual se transforma en una compañía con procesos aceptados, lista para ser llevada adelante a través de técnicas de management tradicionales.

### Clientes iniciales



Partimos esta etapa con el resultado de la anterior: un conjunto de clientes satisfechos por un modelo de negocios validado. Ahora el objetivo es **crecer la startup**, ya sea con una o muchas de las siguientes opciones:

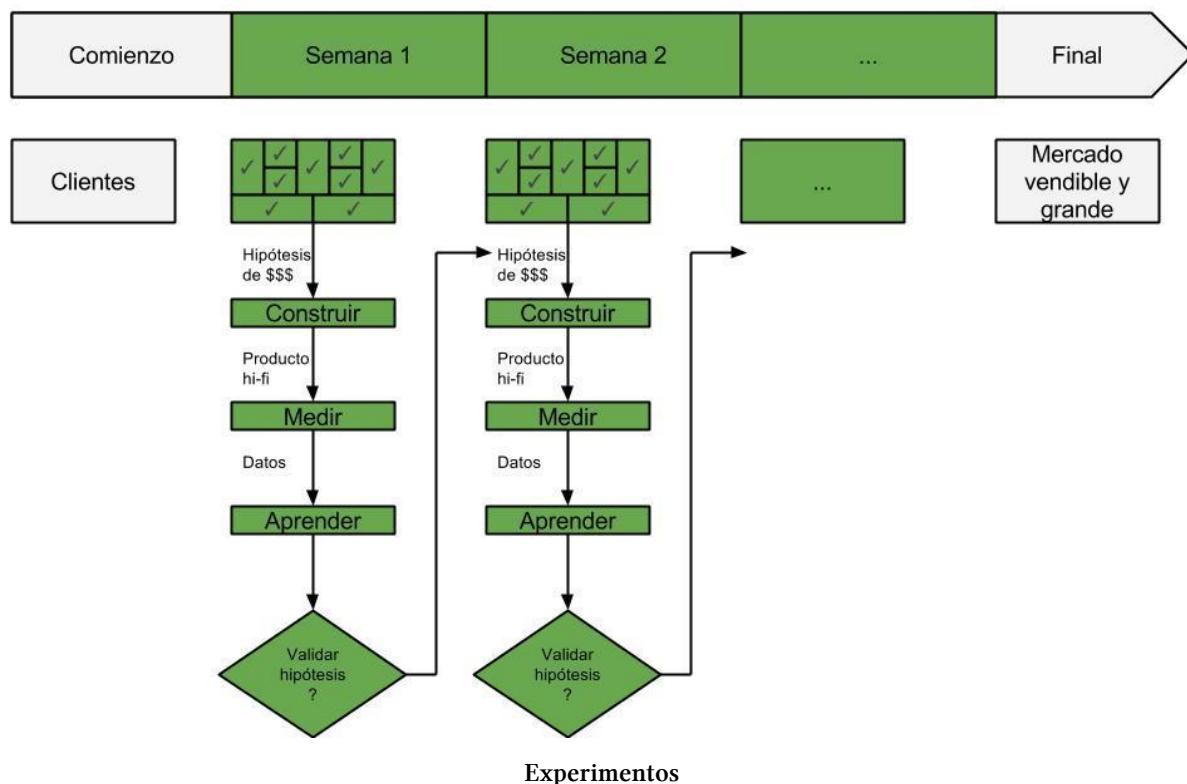
- consiguiendo más clientes
- haciendo que paguen más
- reteniéndolos mejor
- bajando costos de adquisición

Hay que plantear **hipótesis de crecimiento** y validarlas a través de experimentos, voy a describir la forma de realizar esto a continuación.

## Experimentos

# Validación de Clientes

## Experimentos



Hay que hacer experimentos enfocados en la venta del producto o servicio ofrecido, y seguir experimentando todo lo que sea necesario hasta finalmente encontrar una estrategia de ventas definida.

La estrategia de venta debe ser escalable y repetible, esto es importante debido a que hay cierto tipo de acciones que generan crecimiento temporal, el cual no aporta a la sustentabilidad del negocio. Por ejemplo comprar publicidad gracias a dinero de inversores seguramente proveerá un aumento de usuarios, pero si los usuarios no pueden pagar los próximos avisos publicitarios eventualmente no va a haber más dinero para continuar creciendo con publicidad.

Específicamente el objetivo de esta etapa es saber:

- Cuál es el motor de crecimiento que hace avanzar a la startup
- Quién va a comprar
- Por qué
- A qué precio

A continuación voy a mostrar diferentes elementos para incluir en los experimentos de venta:

## Experimentos de venta

Así como en la etapa de Descubrimiento de Clientes había que plantear hipótesis sobre el modelo de negocios, la hipótesis clave a plantear en la Validación de Clientes es sobre el motor de crecimiento de la startup.

**Hipótesis de crecimiento** Como mencioné en el marco teórico, Eric Ries menciona que hay tres motores de crecimiento posibles para una startup: crecimiento pegajoso, crecimiento viral y crecimiento pago. Por lo tanto hay que elegir uno de estos motores de crecimientos, adaptarlos al funcionamiento y procesos de la startup y realizar experimentos para verificar si el modelo elegido produce crecimiento o no.

**Qué construir: MVPs de alta fidelidad** En el Descubrimiento de Clientes se construyen prototipos como MVPs, productos con la funcionalidad mínima para probar hipótesis; en la Validación de Clientes también se construirán MVPs, la diferencia es que estos son mucho más que prototipos por dos motivos principalmente:

1. Hay un modelo de negocios validado, el cual dicta la funcionalidad esperada por los clientes, por lo tanto va a haber un producto mucho más aceptado en esta etapa
2. Un producto mínimo para obtener ventas tiene que ser mucho más robusto que un producto mínimo para probar una hipótesis de negocio

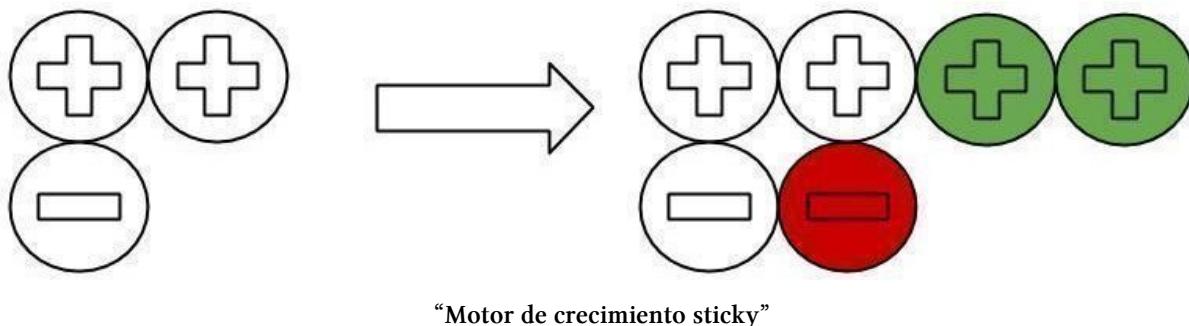
**Cómo medir: Ventas de prueba** Los intentos de compra de los clientes son medidos a través de las llamadas “ventas de prueba”. La forma de realizar una de estas ventas depende del modelo de negocios de cada startup. Un intento de ventas puede ser un cliente entregando dinero a través de un producto, o consiguiendo muchos clientes para que un anunciante pague por estar en el producto/servicio.

**Qué medir: Coeficientes de crecimiento** Las llamadas “métricas de vanidad” pueden dar información errada sobre estado actual de startup, por ejemplo puede darse el caso que una empresa tenga un 40% de nuevos clientes cada mes, pero esté en un estado de caída porque pierde 50% de los clientes a medida que pasan los meses.

Por este motivo hay que concentrarse en las métricas adecuadas para cada startup, y las mismas dependen del motor de crecimiento de cada una. Dependiendo del motor de crecimiento elegido hay que concentrarse en medir y optimizar un conjunto diferente de métricas, a continuación se enumera las mismas:

**Métricas de motor sticky** En este tipo de motor de crecimiento es clave la capacidad de retener clientes, mientras la cantidad de clientes perdidos sea menor que la cantidad de clientes nuevos por mes, se produce crecimiento.

$$\text{Tasa de crecimiento} = \text{Clientes nuevos por mes} - \text{Clientes perdidos por mes}$$



Este es un caso en el que se muestra con claridad la importancia de enfocarse en medir lo correcto, debido a que hay otros números que pueden dar información errónea. Hay métricas de “vanidad” que no indican con certeza el estado actual de una startup con motor sticky:

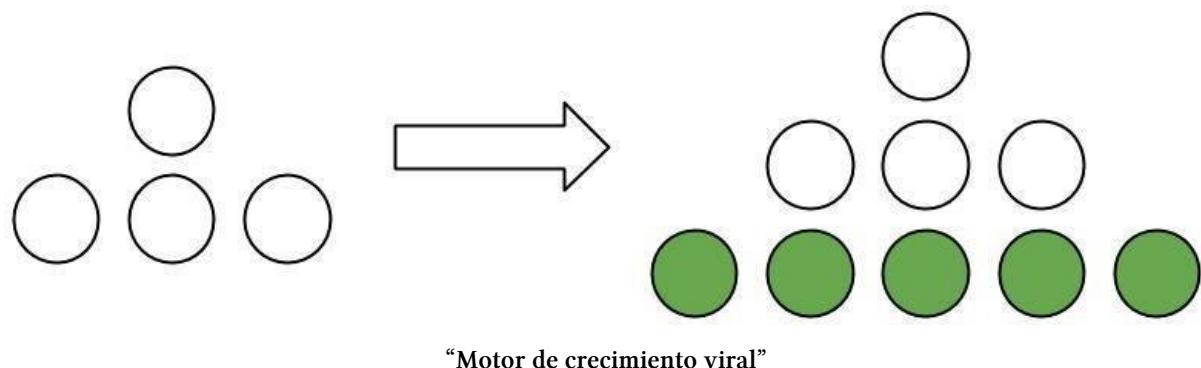
- Número de clientes
- Tasa de activación (usuarios que se transforman en clientes)
- Ganancia por cliente

Una startup puede tener gran cantidad de clientes, puede tener una gran estructura para transformar visitantes en usuarios, o ganar mucho por cliente; pero igual estará destinada a la desaparición si tiene una tasa de crecimiento estancada o negativa.

El enfoque en este tipo de empresas tiene que ser en mejorar retención de clientes, no en ventas o marketing como generalmente sucede.

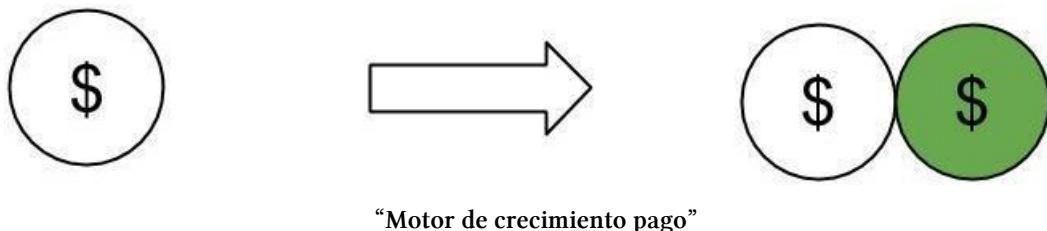
**Métricas de motor viral** Muchas startups conocidas actualmente crecen debido a un motor viral, como Facebook, Twitter o LinkedIn. El concepto de crecimiento es simple, mientras cada cliente nuevo traiga a uno o más amigos, entonces va a haber crecimiento.

$$\text{Coeficiente viral} = \text{Clientes nuevos como consecuencia de cada cliente que se registra}$$



**Métricas de motor pago** El motor de crecimiento pago se logra cuando un cliente significa más dinero que el costo de adquirir uno, el margen de ganancias es utilizado para generar fondos de publicidad, los cuales realimentan el ciclo de adquisición de clientes.

$$\text{Crecimiento} = \text{Valor de vida del cliente} - \text{Costo por adquisición}$$



### Pivots de motor de crecimiento

Los pivots de motor de crecimiento pueden ser necesarios, después de todo hay que recordar que todo comienza como una suposición, y sólo se convierte en certeza al validarla con experimentos y saliendo del edificio.

¿Qué pasa si una startup puede utilizar varios motores de crecimiento? La recomendación de Eric Ries es elegir un sólo motor de crecimiento y optimizarlo, debido a que es muy complejo trackear y optimizar efectivamente varios modelos de crecimiento.<sup>32</sup>

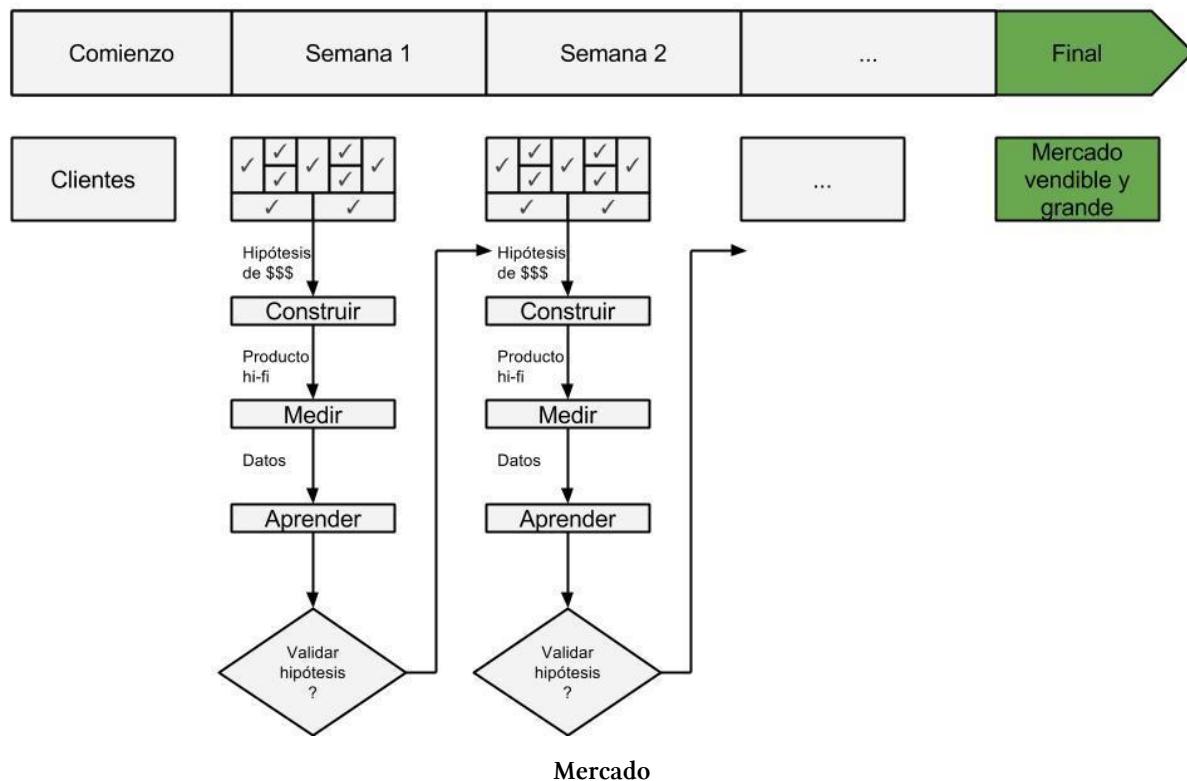
### Paso siguiente

Una vez que la startup encuentre cómo crecer sustentablemente, la misma tiene acceso a un mercado grande y vendible, lo cual es el objeto de la próxima sección.

<sup>32</sup>Libro "The Lean Startup", capítulo 10: Grow, sección "A Technical Caveat" por Eric Ries

## Mercado

# Validación de Clientes Mercado



El momento en que una startup termina de iterar sobre sus hipótesis de crecimiento la misma no sólo ha logrado solucionar un problema, sino que ha encontrado la forma de crecer sosteniblemente.

Esto marca el fin de la etapa de una startup según la definición dada en el marco teórico, la cual es:

*Compañía diseñada para buscar un modelo de negocios y crecer rápidamente, que crea nuevos productos y servicios y opera bajo condiciones de incertidumbre extrema.*

Con un modelo de negocios establecido, un crecimiento validado y muchas incertidumbres transformadas en certezas. Una organización que ha logrado validar un mercado está lista para pasar a una etapa de ejecución, para convertirse en una empresa que ofrezca esos nuevos productos o servicios que diseñó en su etapa de startup.

# Conclusión

- Dos observaciones
  - Una
  - Dos
- Antes
- Ahora
- Después
- Ingenieros pierden tiempo y recursos
- Planes de negocio raramente sobreviven contacto con clientes
- Aprender continuamente de clientes
- Rol de ingeniero en innovacion
- Ingenieros enfocados a métricas -> Aplicar esto a innovacion
- Ingeniero es mucho mas que codigo
- Startups requieren cambio de mentalidad respecto a management tradicional enseñado en universidades

# **Bibliografía**