



**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL**

**DESARROLLO DE MODELO DE NEGOCIO PARA UN GESTOR DE
LOGS PARA APLICACIONES DESARROLLADAS EN LA NUBE
(*CLOUD*).**

TESIS PARA OPTAR AL GRADO DE MAGISTER EN
GESTIÓN PARA LA GLOBALIZACIÓN

LEONARDO HUMBERTO SOTO MUÑOZ

PROFESOR GUÍA:
ANDREA NIETO EYZAGUIRRE

MIEMBROS DE LA COMISIÓN:
CHRISTIAN DIEZ FUENTES
HUMBERTO SALINAS TRONCOSO

SANTIAGO DE CHILE
2014

**RESUMEN DE LA MEMORIA PARA OPTAR AL
GRADO DE:** Magíster en Gestión para la Globalización
POR: Leonardo Humberto Soto Muñoz
FECHA: 23 de septiembre de 2014
PROFESOR GUÍA: Andrea Nieto Eyzaguirre.

DESARROLLO DE MODELO DE NEGOCIO PARA UN GESTOR DE LOGS PARA APLICACIONES DESARROLLADAS EN LA NUBE (CLOUD)

Este trabajo presenta el desarrollo de un modelo de negocios para “BeautifulLogs”, un gestor de logs de aplicaciones (móviles o web) desarrolladas para una infraestructura conocida como “la nube” (cloud computing).

El desarrollo del modelo de negocio sigue una metodología basada en los métodos Lean Startup y Customer Development, enfocada en validar con clientes reales los supuestos de un modelo de negocios y reflejando los avances y el estado de dicho modelo en un Canvas de Modelo de Negocios.

Cada validación se conforma de un experimento con un resultado esperado en el caso en que el supuesto sea correcto.

Durante el desarrollo de este proceso para el modelo de negocios de BeautifulLogs se encontró un espacio en el mercado (ya existente) de este tipo de herramientas, nicho basado en necesidades avanzadas que no está siendo cubierto por las soluciones actuales.

Entre estas necesidades se cuenta la necesidad de almacenar la información por más tiempo que las soluciones actuales (que promedian dos semanas), incrementar la potencia de las funciones de búsqueda y agregar elementos de métricas y analíticas inferidas a partir de los mismos logs.

En base a estas necesidades se establecieron propuestas de valor empaquetadas en planes con un precio validado por los propios usuarios encuestados. El modelo de negocio sigue el esquema SaaS: Software como servicio por el cual el cliente paga una mensualidad (o anualidad) a cambio de su uso.

Si bien los costos variables para poder ofrecer las soluciones ofrecidas en los planes resultaron ser relativamente elevados (para el mercado SaaS), existe un saludable margen entre el ingreso variable promedio por usuario (USD 143,33) y el costo variable por usuario proyectado (USD 68,13). Por otra parte, los costos de adquisición, la conversión de usuarios y el crecimiento neto de usuarios se presentan como las variables claves que determinarán la rentabilidad del negocio, que en un escenario moderado proyecta ventas por casi un millón de dólares en el octavo trimestre de vida del negocio. En su forma actual, el modelo requiere una inversión inicial de USD 350.000 a ocupar principalmente en desarrollo de la plataforma, así como también en capital de trabajo.

La recomendación a seguir es levantar una fracción de esa inversión para efectuar una validación de clientes mediante un producto mínimo viable, que permita bajar el riesgo presentado por la variación de las variables claves mencionadas anteriormente.

Tabla de Contenido

1. INTRODUCCIÓN	1
1.1. OBJETIVOS	1
2. PROCESO DE DESARROLLO DE TRABAJO DE TÍTULO DE EMPRENDIMIENTO	1
2.1. MARCO CONCEPTUAL	1
2.2. METODOLOGÍA Y TÉCNICAS	2
2.2.1. Concepto: Evidencia del Problema	3
2.2.2. Concepto: Canvas	5
2.2.3. Concepto: Supuestos Críticos	5
2.2.4. Concepto: Resumen Ejecutivo Tentativo	5
2.2.5. Experimento: Reporte del resultado	6
2.2.6. Pivote: Análisis del Problema Detectado	6
2.2.7. Pivote: Análisis de Caminos a Seguir	6
3. HISTORIA DEL DESARROLLO DEL MODELO DE NEGOCIO	7
3.1. SEMANA 1	7
3.1.1. Evidencia del Problema	7
3.1.2. Canvas Resultante	9
3.2. SEMANAS 2 Y 3	10
3.2.1. Canvas Resultante	10
3.3. SEMANA 4	15
3.3.1. Experimento 1: Interés Inicial en el Producto	15
3.3.2. Canvas Resultante	19
3.4. SEMANA 5	19
3.4.1. Experimento 2: Propuesta de Valor (parte 1)	19
3.4.2. Canvas Resultante	22
3.5. SEMANA 6	23
3.5.1. Experimento 3: Propuesta de Valor (parte 2)	23
3.5.2. Canvas Resultante	26
3.6. SEMANA 7	27
3.6.1. Análisis del Problema Detectado: Posibles Razones	27
3.6.2. Análisis de Caminos a Seguir	28
3.6.3. Canvas Resultante	29
3.7. SEMANA 8	29
3.7.1. Experimento 4: Pricing (parte 1)	29
3.7.2. Canvas Resultante	32
3.8. SEMANAS 9 Y 10	32
3.8.1. Experimento 5: Pricing (parte 2)	33
3.8.2. Canvas Resultante	35
3.9. SEMANA 11	36
3.9.1. Experimento 6: Definición de Planes	36
3.9.2. Canvas Resultante	39
3.10. SEMANA 12	40
3.10.1. Experimento 7: Definición de Costos	40

3.10.2.	Otros avances	43
3.10.3.	Canvas Resultante	43
3.11.	SEMANAS 13 Y 14.....	44
3.11.1.	Experimento 8: Validar disposición a pagar y a recomendar producto.....	44
3.11.2.	Canvas Resultante	48
4.	MODELO DE NEGOCIO RESULTANTE	49
4.1.	RESUMEN EJECUTIVO	49
4.1.1.	Concepto	49
4.1.2.	Oportunidad y Estrategia.....	50
4.1.3.	Ventajas Competitivas.....	50
4.1.4.	Mercado y Proyecciones	50
4.1.5.	Oferta.....	51
4.2.	MODELO DE NEGOCIO	51
4.2.1.	Customer Segments.....	52
4.2.2.	Value Proposition	53
4.2.3.	Customer Relationships	53
4.2.4.	Channels	53
4.2.5.	Revenue Streams.....	54
4.2.6.	Key Partners.....	54
4.2.7.	Key Activities	54
4.2.8.	Key Resources	54
4.2.9.	Cost Structure.....	54
4.3.	MÉTRICAS Y FLUJOS PROYECTADOS.....	55
4.4.	OPORTUNIDADES DE INTERNACIONALIZACIÓN.....	59
5.	CONCLUSIONES.....	61
5.1.	DESARROLLO DEL MODELO DE NEGOCIO	61
5.2.	FUTURO DEL NEGOCIO.....	64
6.	BIBLIOGRAFÍA.....	65
7.	ANEXOS	66
	ANEXO A. GLOSARIO DE TÉRMINOS.....	66
	ANEXO B. SUPUESTOS DEL MODEL DE NEGOCIO	68

Indice de Figuras

Figura 2.1:	Esquema del proceso de validación y construcción de modelos de negocio.....	4
Figura 3.1:	Ejemplo de un fragmento de un archivo de log	8
Figura 3.2:	Canvas inicial.....	10
Figura 3.3:	Canvas v1.2.....	11
Figura 3.4:	Mail enviado a los voluntarios para beta-tester	17
	18
Figura 3.5:	Reporte de reacción frente al email enviado a los voluntarios para beta-tester	18

Figura 3.6: Canvas v1.3.....	19
Figura 3.7: Maqueta de árbol navegable.....	20
Figura 3.8: Maqueta de estadísticas y métricas en tiempo real.....	21
Figura 3.9: Canvas v1.4.....	23
Figura 3.10: Maqueta de búsquedas y alertas manuales.....	24
Figura 3.11: Maqueta de insights automáticos.....	25
Figura 3.12: Canvas v1.5.....	27
Figura 3.13: Canvas v2.0.....	29
Figura 3.14: Canvas v2.1.....	32
Figura 3.15: Canvas v2.2.....	36
Figura 3.16: Canvas v2.3.....	39
Figura 3.17: Canvas v2.4.....	44
Figura 3.18: Canvas v2.5.....	48
Figura 4.1: Canvas Final.....	52
Figura 4.2: Métricas claves del modelo de negocio. Valores en USD.	56
Figura 4.3: Flujo de caja, escenario moderado. Valores en USD.	57
Figura 4.4: Flujo de caja, escenario pesimista. Valores en USD.	58
Figura 4.5: Flujo de caja, escenario optimista. Valores en USD.	58
Figura 5.1: Canvas inicial (arriba) y final (abajo).....	62
Figura 5.2: Cuatro pasos de la construcción de un negocio según modelo de Steve Blank.	63

1. INTRODUCCIÓN

El presente documento describe el proceso de descubrimiento de un modelo de negocios para el emprendimiento BeautifulLogs, una aplicación orientada al nicho de desarrolladores de aplicaciones web para la nube.

1.1. Objetivos

El objetivo de trabajo de título es validar los elementos clave de un modelo de negocios para BeautifulLogs.

De ser exitosa la búsqueda, se obtendrá las bases de un modelo de negocio escalable y repetible para este emprendimiento, que le permita establecer un plan para desarrollar la empresa, sea mediante financiación propia o con participación de inversionistas.

Asimismo, es posible que la búsqueda termine en un resultado negativo, en cuanto a que las hipótesis clave que forman parte de los supuestos para que el emprendimiento sea exitoso no se logren validar con usuarios y clientes reales. En tal caso, se obtendrá una comprensión mayor del mercado objetivo y las limitaciones que impiden que el modelo propuesto sea viable. Es posible que dicha comprensión abra nuevos caminos a explorar, lo cuales podrían ser abordados en futuros trabajos de títulos o de manera independiente por el autor.

2. PROCESO DE DESARROLLO DE TRABAJO DE TÍTULO DE EMPRENDIMIENTO

2.1. Marco Conceptual

En la literatura moderna sobre emprendimientos (startups) se ha establecido con claridad la máxima de que un emprendimiento no es una versión en miniatura de un negocio consolidado.

Más bien un emprendimiento es una organización temporal en búsqueda de un modelo de negocio que le permita, sólo una vez encontrado dicho modelo negocio, pasar a la etapa de ejecución en escala.

Se hace necesario por tanto utilizar conceptos y bibliografía diseñados para la fase de búsqueda de un modelo de negocio que caracteriza a los emprendimientos. La

siguiente lista refleja la bibliografía usada por el autor para lidiar con esta etapa, junto con una breve descripción justificando la relevancia de cada ítem:

- “Business Model Generation”, por Osterwalder y Pigneur [7]: Al nivel más fundamental, una startup requiere contar con un modelo de negocios. Es cierto que el modelo de negocios de una startup puede sufrir cambios continuamente durante el período de búsqueda de un modelo escalable y repetible. Pero igualmente cierto es que es necesario contar con un lenguaje compartido para que todos los stakeholders puedan comunicar el estado actual del modelo de negocio. Este libro ofrece un marco conceptual para entender un modelo de negocio a través de 9 bloques principales que lo construyen, y ofrece herramientas para visualizar y desarrollar dicho modelo.
- “New Venture Creation, Entrepreneurship for the 21st Century”, por Timmons y Spinell [13]: Una de las guías más completas y reconocidas para el desarrollo de nuevos emprendimientos.
- “The Lean Startup”, por Eric Ries [12]: Que el proceso de creación de una startup sea un proceso caótico y normalmente imprevisible no significa que no hayan formas de disminuir el riesgo de fracaso. La principal contribución de este libro es el desarrollo de metodologías más científicas (en el sentido de repetibles y verificables) para generar emprendimientos nuevos.

En base a lo sugerido por Ries [12], el desarrollo de un emprendimiento se puede ver como una iteración del ciclo “build-measure-learn” donde se inicia construyendo o variando una funcionalidad o elemento clave del modelo de negocios, con el objetivo de validar las hipótesis que sostienen el modelo lo más temprano posible.

La principal técnica para mantener en sincronía el trabajo de con la realidad descubierta en los experimentos es usar un business model canvas [7] que refleje de manera sucinta el estado del modelo en cada uno de los pasos.

2.2. Metodología y Técnicas

La metodología se basa en 3 procesos básicos basados en el ciclo “build-measure-learn”:

- Concepto: Definición del problema a resolver y de la posible solución al mismo.
- Experimento: Ejercicio práctico en el que se intenta probar o descartar algún supuesto del modelo de negocio.
- Pivote: Reformulación del modelo a realizar cada vez que un experimento arroje resultados que contradicen los supuestos del modelo de negocio.

Los últimos 2 procesos se repiten indefinidamente hasta cuando todos (o la mayoría de) los supuestos críticos del modelo cuenten con algún tipo de validación real.

Alternativamente, puede ocurrir que múltiples pivotes no produzcan cambios significativos en los resultados de experimentos, en cuyo caso se puede concluir que el modelo de negocio no es viable en base a los supuestos considerados. La figura 2.1 muestra de manera esquemática el proceso y los entregables de cada etapa. Estos entregables son posteriormente compilados en un sólo ente coherente que en este trabajo de título se reflejan en los capítulos 3 y 4.

El esquema propuesto es también consistente con el proceso de “Customer Discovery” propuesto por Blank y Dorf [1], incorporando elementos incipientes de “Customer Validation” aunque sin llegar a cubrirlo en su totalidad dada la limitación de tiempo de un proceso de titulación.

El proceso culmina cuando hay un potencial detectado para un “market fit”, es decir, donde la solución ofrecida coincida con demanda de usuarios o clientes reales. Alternativamente, el proceso puede concluir en un fallo si se determina que múltiples pivotes no generan progreso y el modelo de negocios propuesto inicialmente no tiene viabilidad aunque se varíen significativamente algunos elementos.

En cualquiera de los dos casos se procede a elaborar documentos finales (flujo de caja, modelo detallado y análisis de oportunidades de globalización en caso exitoso, reporte de inviabilidad en caso contrario) y a compilar los documentos en un todo coherente que conforma el trabajo de título.

A continuación se detallan los documentos que plasman la ejecución de cada una de las 3 fases principales (Concepto, Experimento y Técnica) y que conforman el grueso del trabajo de título una vez han sido compilados según lo dicho en el párrafo anterior.

2.2.1. Concepto: Evidencia del Problema

Resumen inicial que justifique la existencia de un problema a resolver (o de una necesidad insatisfecha). Da inicio al proceso y permite evaluar si el tema y mercado corresponden a una tesis de emprendimiento. Se recomienda la siguiente estructura:

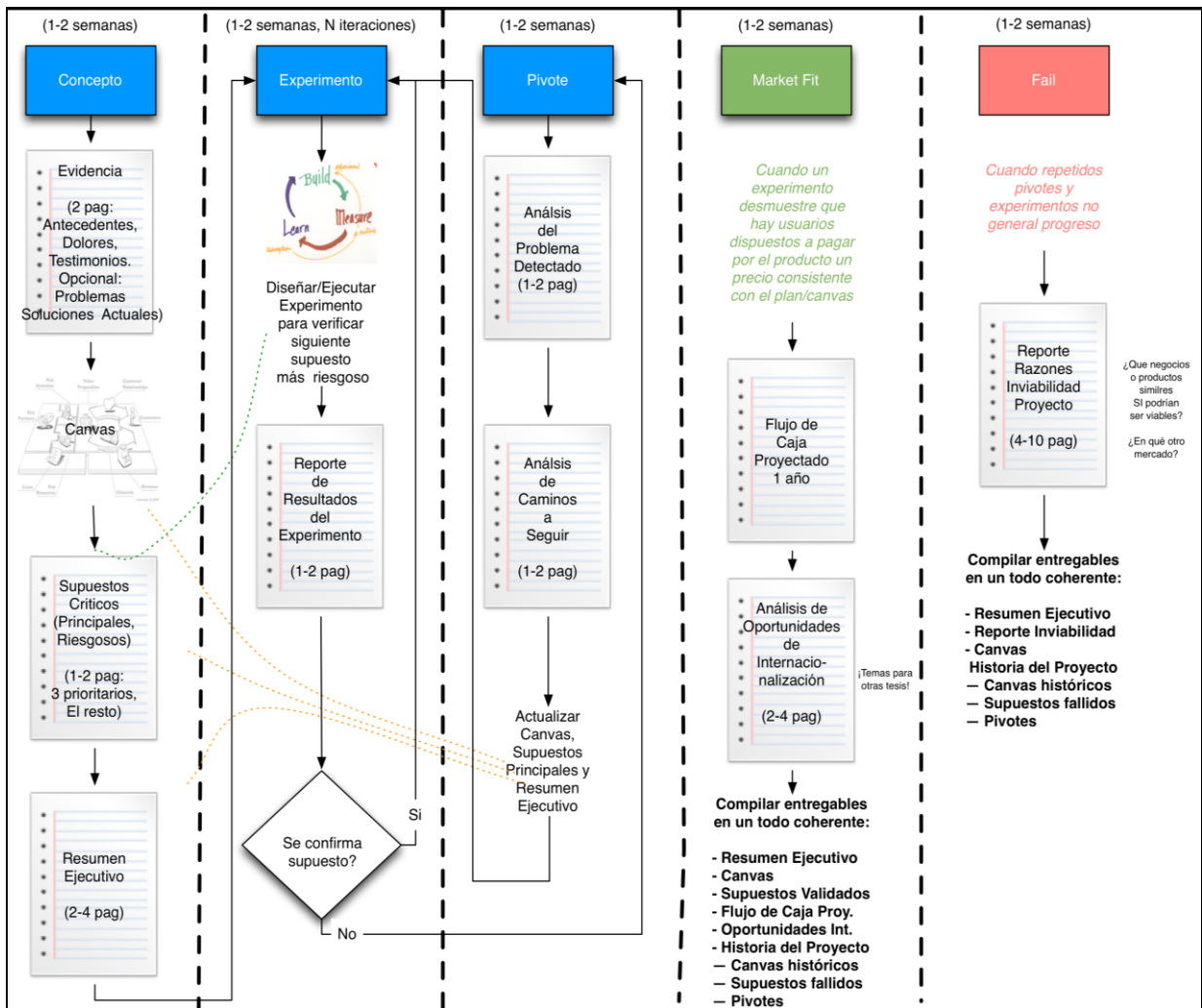


Figura 2.1: Esquema del proceso de validación y construcción de modelos de negocio

- Antecedentes: Breve introducción al mercado y/o problema.
- Dolores: Problemas concretos que alguien experimenta (o necesidades concretas que no se encuentran satisfechas).
- Testimonios: Citas de personas reales que demuestren que los problemas o necesidades insatisfechas existen.
- Problemas de las soluciones actuales: En caso que el mercado exista, indicar qué hace que las soluciones existentes no cubran completamente las necesidades de los potenciales clientes.

2.2.2. Concepto: Canvas

Business Model Canvas tentativo para un modelo de negocio que satisface la necesidad. Sirve para validar que además de una necesidad real, existe una posibilidad de establecer un negocio en torno a ella.

Este es un documento “vivo” en cuanto a que será modificado durante el desarrollo del modelo de negocio reflejando el aprendizaje, las validaciones y los nuevos supuestos que sea necesario incorporar al modelo en las fases de Experimentos y Pivotes.

2.2.3. Concepto: Supuestos Críticos

Descripción de qué supuestos del canvas (en esta fase prácticamente todos los elementos del canvas serán supuestos) son los más críticos. La criticidad de un supuesto está dada por el impacto en el modelo si es que el supuesto resulta ser falso. Al igual que el canvas, esta lista es un documento “vivo” que va evolucionando conforme los supuestos más críticos van siendo despejados y nuevos supuestos se van volviendo críticos.

2.2.4. Concepto: Resumen Ejecutivo Tentativo

Resumen ejecutivo que describa la idea bosquejada en el canvas inicial. Si bien está basado casi totalmente por supuestos, permite determinar cómo calzan las distintas piezas del modelo con el mercado actual en la visión del emprendedor. No se espera que el resumen ejecutivo final coincida con esta visión. Se recomienda la siguiente estructura, basada en [13].

- Concepto: Idea general del problema a resolver.
- Oportunidad y Estrategia: Descripción de la oportunidad presentada por el mercado y el problema, junto con la estrategia general para aprovecharla.
- Ventajas Competitivas: Elementos que harán sostenible y defendible la propuesta de valor frente a la competencia (existente y futura).
- Mercado y Proyecciones: Estimaciones que indiquen una comprensión general del tamaño del mercado y potencial del negocio.
- Oferta: En caso que se estime necesaria una inversión, indicar porcentajes de propiedad dispuestos a ofrecer a cambio de ésta, junto con una estimación del valor ofrecido al inversionista.

2.2.5. Experimento: Reporte del resultado

Este documento reporta la ejecución y resultado de un experimento. Se sugiere la siguiente estructura:

- Sigüientes supuestos a validar: Indicación de cuales supuestos son los determinados como críticos al momento de ejecutar el experimento.
- Definición: Propósito del experimento y pasos a seguir.
- Resultado Esperado: Métrica que indicará si el experimento es exitoso o no.
- Ejecución: Descripción sucinta de la ejecución del experimento. De particular interés es describir cualitativa o cuantitativamente las reacciones de usuarios, clientes u otros actores involucrados en el experimento.
- Resultado: Evaluación del experimento.

2.2.6. Pivote: Análisis del Problema Detectado

Es una continuación natural del resultado del experimento fallido que le precede, describiendo los posibles factores que pueden haber jugado un rol en la invalidación del supuesto respectivo.

Lo ideal es que dichos posibles factores (hipótesis en si mismos) sean respaldados por evidencias o indicios que los respalden.

2.2.7. Pivote: Análisis de Caminos a Seguir

A partir de los factores descritos en el análisis anterior, este documento lista los posibles caminos a seguir, dependiendo del peso que se le dé a cada factor, a las implicancias para el resto del modelo de negocio y/o a razones prácticas respecto a la capacidad de validar o invalidar futuros supuestos.

Debe concluir con una elección entre dichos caminos descritos.

3. HISTORIA DEL DESARROLLO DEL MODELO DE NEGOCIO

3.1. Semana 1

La primera semana de trabajo se documentó la evidencia del problema a resolver y se propuso un boceto de modelo de negocio basado en el canvas de la figura 3.2.

3.1.1. Evidencia del Problema

Antecedentes

Una de las maneras en las que los desarrolladores resuelven los errores de los programas computacionales consiste en ir registrando la actividad más importante del sistema en una bitácora (conocida como “log”[14]) de modo que cuando un error es detectado en el problema el desarrollador pueda ver los eventos que ocurrieron antes de dicho error. Esto es similar a la práctica en la que una caja negra de un avión registra un vuelo de modo de poder investigar a posteriori cualquier problema.

Normalmente estos logs se almacenan en un archivo. Además, cada componente de software suele tener asociados sus propios logs. Hoy en día, con la creciente tendencia a desplegar sistemas computacionales en la “nube” (cloud) especialmente en servicios web [3], una aplicación consta de distintos componentes, típicamente ejecutándose sobre múltiples computadores diferentes. Siguiendo la analogía del avión, los desarrolladores no cuentan con una sola caja negra, sino que con múltiples cajas negras ubicadas en múltiples partes del avión.

Dolores

El tener que leer múltiples logs desde múltiples archivos en múltiples computadores genera los siguientes dolores:

- Falta de acceso a un log por razones de seguridad. Sólo personal autorizado tiene acceso a los computadores de producción en ciertos entornos.
- Dificultad de “conciliar” logs para recrear la secuencia de eventos de un sistema distribuido.
Normalmente un desarrollador usará la fecha/hora de cada evento para intentar seguir la secuencia a través de varios archivos de logs, pero esto se vuelve más complejo mientras más logs existan por diversos factores:
 - En ocasiones los logs son generados bajo distintas zonas horarias (arquitecturas distribuidas o configuración errónea de servidores).

- Cada aplicación tiene libertad para generar logs en su propio formato, lo que dificulta el reconocimiento de patrones mientras mas formatos se involucran en la lectura de múltiples logs.
- Dificultad de buscar algo en particular (¿En cuál de todos los logs lo busco?)
- Para desarrolladores web: La inconveniencia de tener que salir del entorno de desarrollo (editor de textos más navegador web) para conseguir leer los logs.

```

2014-07-07 04:19:57+00:00 app web.1 Read fragment views/search_top_Chile (2.0ms)
2014-07-07 04:19:57+00:00 app web.1 Rendered jobs/_search_top.html.haml (2.1ms)
2014-07-07 04:19:57+00:00 app web.1 Rendered common/_logged.html.haml (0.9ms)
2014-07-07 04:19:57+00:00 app web.1 Rendered common/_footer.html.haml (1.3ms)
2014-07-07 04:19:57+00:00 app web.1 Completed 200 OK in 143ms (Views: 93.1ms | ActiveRecord: 48.0ms)
2014-07-07 04:19:56+00:00 app web.1 Processing by WebprosController#logout as HTML
2014-07-07 04:19:56+00:00 app web.1 Completed 302 Found in 6ms (ActiveRecord: 2.7ms)
2014-07-07 04:19:57+00:00 heroku router at=info method=GET path="/" host=www.getonbrd.cl request_id=c54662b9-f738-482c-8e45-1cd68b4cb39f fwd="190.20.241.216" dyno=web.1 connect=1ms service=158ms status=200 bytes=70655
2014-07-07 04:20:37+00:00 heroku web.1 Process running mem=659M(128.8%)
2014-07-07 04:20:37+00:00 heroku web.1 Error R14 (Memory quota exceeded)
2014-07-07 04:20:46+00:00 heroku router at=info method=GET path="/assets/application-82294bf66f070fb95eb1b9462f9ad8b9.js" host=getonbrd.com request_id=6b085e41-bd1e-4bc7-a576-34991192330b fwd="190.21.115.77" dyno=web.1 connect=1ms service=14ms status=200 bytes=180973
2014-07-07 04:20:46+00:00 app web.1 cache: [GET /assets/ajaxful_rating-aa13ba0467df7c2908e7c4efdd525a1d.css] fresh
2014-07-07 04:20:46+00:00 app web.1 cache: [GET /assets/continuum-eb5fbfe4332a8c90227c7b9909795129.png] fresh
2014-07-07 04:20:46+00:00 app web.1 cache: [GET /assets/flags/us_mini-8b4f2006ad0fc4b13b5d4bb4a23abff9.png] fresh
2014-07-07 04:20:46+00:00 heroku router at=info method=GET path="/assets/partners/dev_academy-88902e8249f0d2dbc95175475471ea7d.png" host=getonbrd.com request_id=56ecef0f-a5cb-4a93-b9c7-d16063a68d0b fwd="190.21.115.77" dyno=web.1 connect=1ms service=5ms status=200 bytes=4541
2014-07-07 04:20:45+00:00 heroku router at=info method=GET path="/" host=www.getonbrd.cl request_id=c240ffe0-c533-48b1-9e06-e53404955f91 fwd="190.21.115.77" dyno=web.1 connect=2ms service=167ms status=200 bytes=70655
2014-07-07 04:20:46+00:00 heroku router at=info method=GET path="/assets/continuum-eb5fbfe4332a8c90227c7b9909795129.png" host=getonbrd.com request_id=cef3683a-de86-4132-8d49-8a3486945bf5 fwd="190.21.115.77" dyno=web.1 connect=1ms service=4ms status=200 bytes=1934
2014-07-07 04:21:04+00:00 app web.1 Processing by CompaniesController#show as HTML
2014-07-07 04:21:05+00:00 heroku router at=info method=GET path="/assets/ajaxful_rating/star_small.png" host=getonbrd.com request_id=9925c2ae-8f8c-44b0-b984-85fcf66ba4ac fwd="186.107.215.118" dyno=web.1 connect=2ms service=5ms status=200 bytes=568
2014-07-07 04:21:04+00:00 app web.1 Read fragment views/right_container_Chile (2.3ms)
2014-07-07 04:21:04+00:00 app web.1 Rendered common/_network_social_container.html.haml (2.6ms)
2014-07-07 04:21:04+00:00 app web.1 Read fragment views/search_top_Chile (2.0ms)
2014-07-07 04:21:04+00:00 app web.1 Rendered jobs/_search_top.html.haml (2.2ms)
2014-07-07 04:21:04+00:00 app web.1 Rendered common/_logged.html.haml (1.0ms)
2014-07-07 04:21:04+00:00 app web.1 Completed 200 OK in 410ms (Views: 149.0ms | ActiveRecord: 253.5ms)
2014-07-07 04:21:05+00:00 app web.1 cache: [GET /assets/ajaxful_rating/star_small-13b19017fdfe4e4bb125dd663df17b4e.png] fresh

```

Figura 3.1: Ejemplo de un fragmento de un archivo de log

Testimonios

Para verificar si los dolores anteriores (experimentados personalmente por el autor y por sus compañeros de trabajo cercanos) eran experimentados por otras personas, el autor consultó a su red en Twitter cómo lidiaban los desarrolladores con esta clase de problemas y obtuvo las siguientes respuestas:

- “En desarrollo web echo de menos un lector de logs que se integre al browser como firebug o la consola de chrome” —Álvaro Aguirre, desarrollador web.
- “Tiro los logs a mongodb [una base de datos] para cosas mas importantes [para] poder hacer queries para encontrar lo que quiero [...]” — Mendi, desarrollador.
- “Busco los logs a mano, me fío de mi vista rápida. También me gustaría saber [de] alguna herramienta para mejorar su lectura” — Felipe Egas, deasrrollador.

Además, el autor encontró un email con una necesidad relacionada en la lista de correo de desarrolladores Ruby del cono sur:

- “Les escribo para consultar si es posible consultar a través de mi aplicación [web] rails los logs [...]” — Paolo Loran

Problemas de soluciones actuales

Los desarrolladores de Continuum (empresa donde trabaja el autor) han experimentado usando herramientas existentes para resolver estos dolores, especialmente Loggly [6], LogEntries [5] y Papertrail [9]

Loggly no resolvió apropiadamente los dolores mencionados debido a su aparente enfoque hacia administración de sistemas: Los logs no se combinan y sigue siendo necesario mirar cada archivo por separado.

Las debilidades de Logentries son iguales a las de Loggly, con el extra de que el área para leer logs es muy pequeña (menos de 1/3 del espacio disponible en una pantalla típica de 13 pulgadas).

Papertrail fue una mejor experiencia, pero hay fallos de usabilidad muy importantes. Por ejemplo, luego de realizar una búsqueda y encontrar un resultado, no es posible ver rápidamente qué otras cosas hay alrededor de ese resultado. Esto impide ver que ocurrió justo antes de algo importante, lo que es un patrón de uso extremadamente frecuente según la experiencia de los desarrolladores de Continuum.

3.1.2. Canvas Resultante

Los elementos del canvas están todos basados en supuestos. El color de cada elemento refleja que tan crítico es el supuesto para el modelo de negocios en general. La escala de colores va desde amarillo para lo menos crítico hasta rojo para elementos que deben ser validados cuanto an.

Los elementos que sean validados en el proceso serán pasados a color verde.

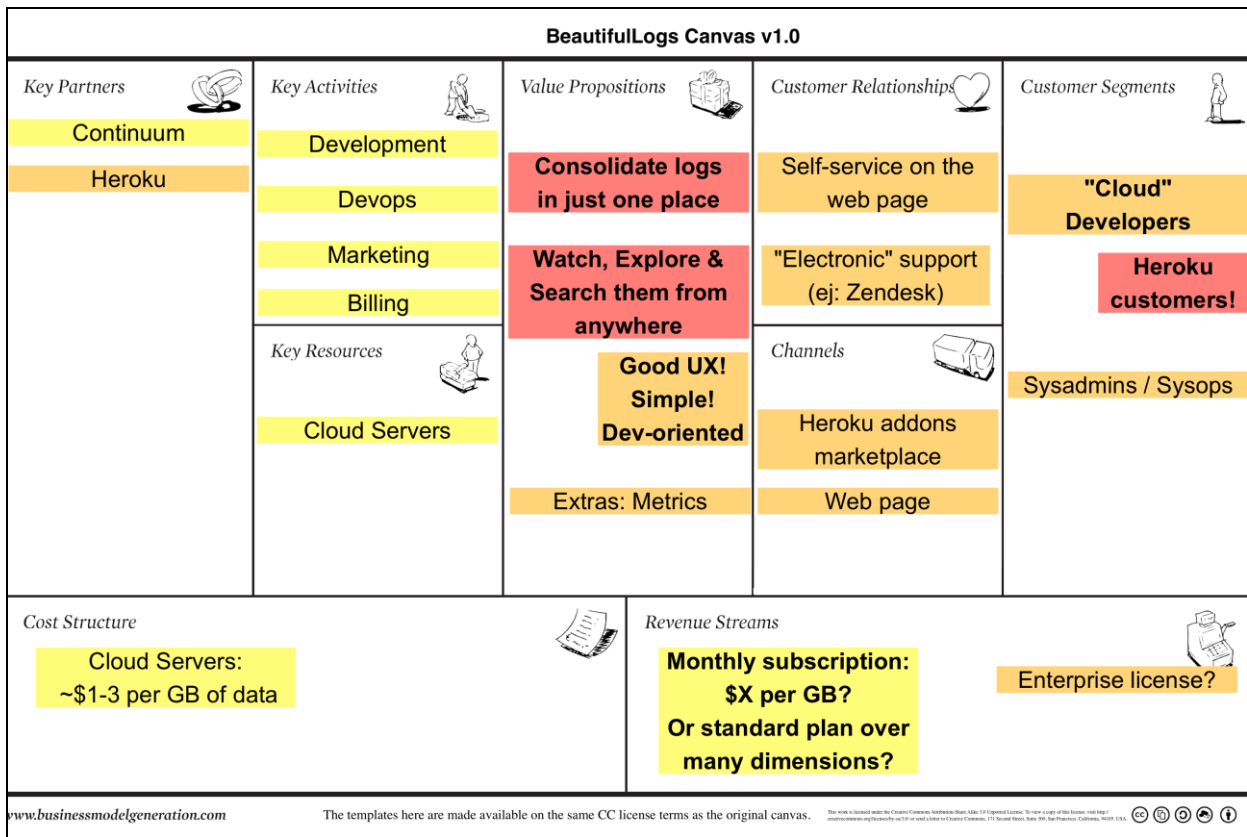


Figura 3.2: Canvas inicial

3.2. Semanas 2 y 3

En la segunda y tercera semana de trabajo se refinó el modelo inicial, mejorando la propuesta de valor de manera que permita diferenciar el producto con los existentes en el mercado, como lo refleja la figura 3.3.

3.2.1. Canvas Resultante

En base al feedback con la profesora guía se cambió completamente los elementos de la propuesta de valor para enfatizar la diferenciación del producto y conectar las características novedosas con necesidades reales de los usuarios. Se introdujo también la distinción entre el usuario (desarrollador) y el comprador económico (su organización o empleador). También se introdujeron hipótesis sobre los costos y sobre un flujo de ingresos basado en planes sin límites (consistente con la idea de tener planes sin límites en la propuesta de valor).

Todos estos cambios aparecen destacados con un borde azul en la figura siguiente.

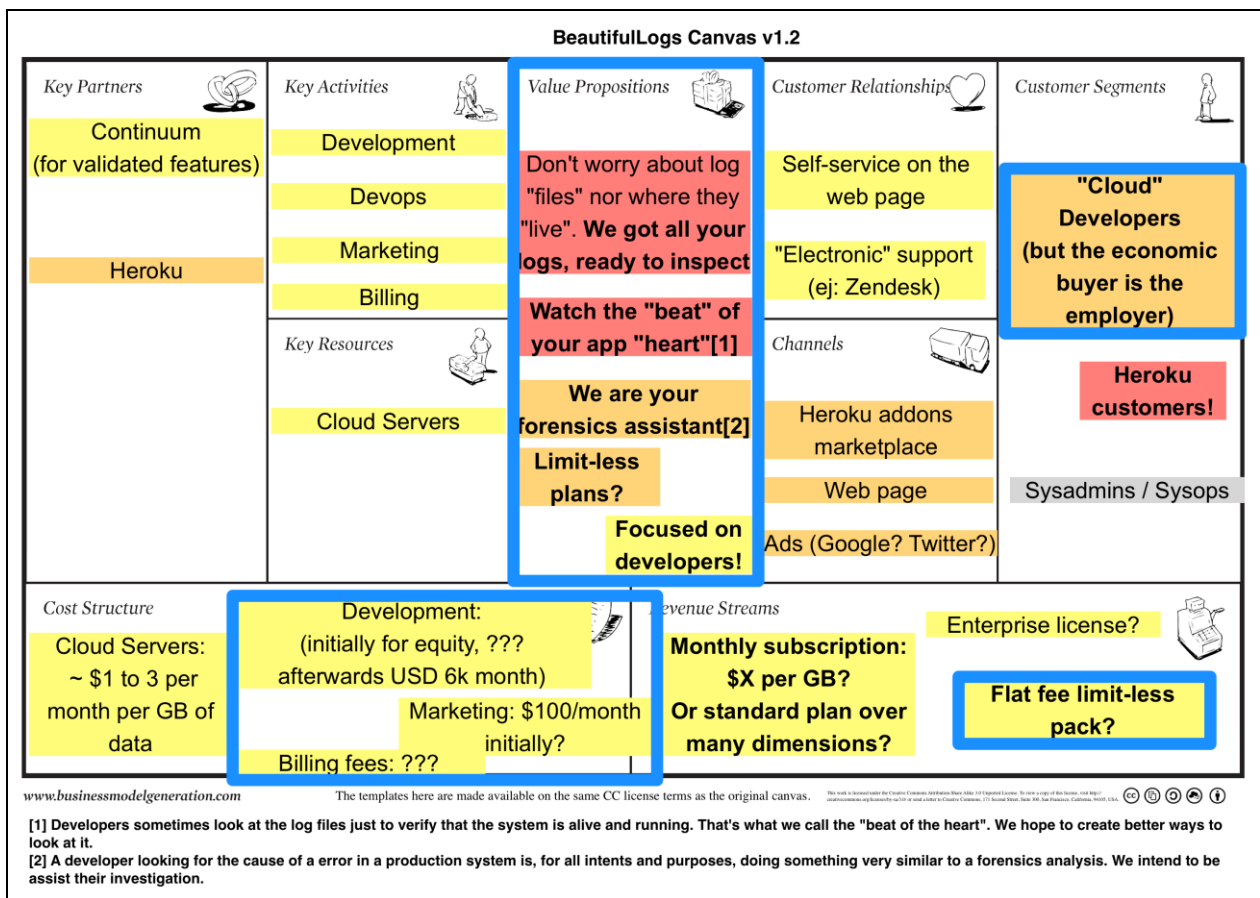


Figura 3.3: Canvas v1.2

En base a esto, se articuló por primera vez una visión muy pre-eliminar de la visión para el emprendimiento, reflejado en un resumen ejecutivo tentativo detallado en las secciones a continuación:

Concepto

BeautifulLogs sería una solución para desarrolladores de software desplegado en plataformas cloud a monitorear sus aplicaciones y encontrar más rápidamente la raíz de los inevitables bugs que ocurren en los sistemas en producción.

Las plataformas cloud le ofrecen a los desarrolladores una manera extremadamente económica de arrendar capacidad de cómputo bajo demanda, permitiéndole escalar los recursos utilizados por sus sistemas al mismo ritmo que escala el número de usuarios (o la frecuencia de uso). Para conseguir esto, las aplicaciones tienden a estar distribuidas en múltiples nodos, algunos de ellos idénticos entre sí.

Una desventaja de una plataforma distribuida es que investigar la causa de errores se hace más complejo, pues hay que mirar las "cajas negras" (logs) de cada uno de los subsistemas involucrados.

Muchas veces un desarrollador no sabrá de antemano en cuál de todos los nodos que cumplen la misma función (y que son idénticos entre sí) debe buscarse la evidencia del error.

Las aplicaciones web y aplicaciones móviles son casos típicos de estos escenarios. Imaginemos una aplicación:

- ejecutándose en el smartphone de un usuario,
- que se conecta a uno de los múltiples servidores usados para identificar al usuario de la aplicación,
- y donde dicho nodo encola un proceso que actualiza la información del usuario,
- el cual es ejecutado por otro nodo diferente encargado de procesar la cola de mensajes,
- el que recibe un extraño error de parte de la base de datos cuando intenta guardar información.

Si el desarrollador quisiera tener una imagen completa de lo que ocurrió para diagnosticar el error, tiene que juntar piezas del puzzle, las que residen en cada uno de los puntos enumerados. Y de hecho es muy probable que no consiga juntarlas todas.

Oportunidad y Estrategia

Una mirada superficial al problema llevaría a la solución (también superficial, aunque no por ello carente de valor) de consolidar los logs de los distintos componentes en un lugar central. Y si nos enfocáramos en desarrolladores de aplicaciones web y móviles, ser capaces de acceder a dichos logs mediante un navegador web es algo natural.

Justamente lo aquello es lo que tienen en común muchas de las soluciones existentes.

Yendo un poco más en profundidad, una vez que se cuenta con profusos logs el problema principal sería navegar y buscar información dentro de estos datos. Esto también lo implementan, de alguna u otra forma y con algunas deficiencias de usabilidad, las soluciones existentes. La primera oportunidad de BeautifulLogs para conseguir usuarios sería ofrecer una mejor usabilidad para explorar y buscar información en los logs.

Pero falta recordar para qué los desarrolladores necesitan navegar y buscar en sus logs, aunque ya se haya “resuelto” el inconveniente de tener esta información desperdigada. ¡Encontrar la razón de los errores! ¡Buscar patrones que expliquen un comportamiento extraño! ¡Revisar, entiendo real, que todo está sucediendo como se espera!

Esto último es algo que cubre el producto Librato. Pero resolver el problema pieza por pieza lleva nuevamente al dolor original: Un desarrollador que tiene que consultar

distintos sistemas (consolidador de logs, consolidador de metricas, etc) para resolver lo que en el fondo es un único problema: ¿Qué salió mal con mi aplicación?

La visión para BeautifulLogs sería cubrir todo el espectro de herramientas para ayudar a descubrir errores y comportamientos inesperados, tanto en tiempo real como en modo “forense” (analizando las pistas en un momento posterior a cuando ocurrió el error). Navegar y explorar los logs es el primer paso. Luego vendría BeautifulMetrics para llenar el espacio de métricas que se pueden resumir en gráficos e indicadores estadísticos. Y un último paso sería BeautifulInsights, donde análisis estadístico y “data mining” podría ayudar a detectar patrones de errores antes que el propio desarrollador o permitir correlacionar elementos de cada log y construir la historia real de un evento que llevó a un error. Ya no se trataría de juntar las piezas del puzzle en un mismo lugar, sino de asistir activamente a armar el puzzle.

Lo anterior es en cuanto a propuesta de valor. En cuanto a segmento de clientes el nicho sería el de los desarrolladores (y sus empresas, que son el comprador económico) que ya consumen productos cloud. De especial importancia serían los clientes actuales de Heroku (alrededor de 30 mil sitios web de acuerdo a BuiltWith.com), pues esta plataforma cuenta un mercado de addons que ya tiene acostumbrados a sus usuarios a pagar por servicios adicionales en modelo de suscripción mensual.

Dicho mercado sería el canal inicial de venta del producto y el modelo de cobro será probablemente en base a una estructura de 3 precios diferentes para discriminar por necesidades de los usuarios, además de ofrecer algún plan gratuito a modo de prueba por tiempo limitado.

El desarrollo del producto estaría a cargo del equipo de Continuum, empresa especializada en desarrollo web y móvil. Continuum obtendría un porcentaje de propiedad de BeautifulLogs a cambio del desarrollo. El producto estaría alojado en DigitalOcean, donde el costo por GB almacenado se estima entre USD 1 a USD 3 por mes.

Ventajas Competitivas

Servicios existentes como Loggly están orientados a usuarios con perfil de administrador de sistemas que deben monitorear múltiples sistemas, por lo que tienden a ofrecer más “amplitud” que “profundidad”. Otros como Papertrail son orientados al caso de uso de un desarrollador, pero presenta deficiencias de usabilidad que reflejan poca comprensión del problema de fondo descrito en la sección de Oportunidad y Estrategia.

Pero la ventaja competitiva de BeautifulLogs no sería encontrar primero las características adicionales que apoyarán a los desarrolladores a encontrar y resolver errores más rápido y más eficazmente. La ventaja sería contar con la perspectiva que permite empatizar con el desarrollador, la que nos lleva a pensar en estas características y nos permitiría en un futuro encontrar otras.

A mediano plazo se contemplaría ofrecer suficiente valor agregado en forma de métricas e insights generadas a partir de los mismos logs consolidados. La idea sería generar un lock-in de nuestros usuarios que impida que luego de un tiempo sea práctico migrar a un servicio de la competencia, aunque implemente características similares, ya que BeautifulLogs irá mucho más allá de simplemente consolidar logs y/o generar alertas básicas.

Mercado y Proyecciones

De acuerdo a BuiltWith.com, Heroku cuenta con 30.000 sitios web desarrollados sobre su plataforma. Asumiendo que uno de cada 3 clientes de Heroku es un cliente de pago, contamos con un universo de 10.000 potenciales clientes, que van desde individuos con sus proyectos personales, pasando por startups en distintas etapas de crecimiento e incluyendo también empresas Fortune 500.

Se asume un ingreso referencial por cliente de USD 50 por 10GB/mes (consistente con el precio promedio de las soluciones existentes actuales) se obtiene un margen inicial por cliente que oscila entre los 20 y 40 USD. De manera conservativa se considera que hasta la mitad de ese margen podría gastado en comisiones por cobros, procesamiento de tarjeta de crédito, y costos extras de infraestructura, dejando entre 10 y 20 USD mensual por cliente como margen bruto.

Con una penetración inicial en los primeros 6 meses en early adopters dentro del nicho de clientes de Heroku del 1 %, se tendrían ingresos mensuales de USD 5.000. De esto entre USD 1.000 a USD 2.000 corresponderían al margen bruto, lo que no alcanzaría a cubrir costos fijos de desarrollo ni de marketing/ventas. Para ejecutar el desarrollo del producto a una velocidad que permita competir en el mercado se espera requerir la contratación de un equipo de dos personas, con un costo estimado de USD 6.000/mes. Las pérdidas en este período se estiman en torno a USD 5.000/mes.

Una proyección objetivo a 18 meses dentro del nicho de clientes de Heroku sería llegar al 15% del universo total (1.500 clientes), lo que correspondería a ingresos de USD 75.000 al mes, con un margen bruto de USD 15.000 a USD 30.000, números con los cuales se espera que los costos fijos de desarrollo y marketing (que no se espera excedan los USD 15.000 por mes) sean cubiertos.

El universo global de potenciales clientes es por supuesto mucho mayor. De acuerdo a Built-With.com, Amazon cuenta con más de 300.000 sitios web alojados en su infraestructura cloud, lo que es órdenes de magnitud más grande que Heroku. Aún asumiendo que el cliente típico de Amazon Web Services tiene solo un quinto de posibilidades de ser un cliente nuestro con respecto al cliente típico de Heroku, estamos hablando de una proyección objetivo del 3% del universo, es decir, 9.000 clientes.

Esto sumado a los 1.500 de Heroku totalizan 10.500 clientes para un ingreso mensual de USD 525.000 y un margen bruto de entre USD 105.000 y USD 210.000, lo que anualizado se aproxima a entre 1 y 2 millones de dólares mensuales en margen bruto.

Considerando que la mitad de este margen puede ser usado para costos fijos en desarrollo y semi-fijos de marketing, el producto presentaría potencial de utilidades entre USD 500.000 y USD 1.000.000 al año.

Estas proyecciones asumen un escenario estático y no incluyen la tendencia creciente en el uso de soluciones cloud, por lo que de cumplirse los otros supuestos, hay margen para cifras aún mas auspiciosas de ingresos futuros.

Oferta

Una inversión de USD 200.000 permitiría financiar con holgura 18 meses de operación con pérdidas entre USD 5.000 y USD 10.000 mensuales. A cambio de esto el inversionista obtendría un 25% de participación en la empresa.

Si la empresa consigue, según las proyecciones anteriores, obtener ganancias por alrededor de USD 1.000.000 anuales, una valuación basada en un ratio P/E de 12 ubicaría el valor total de la empresa en USD 12.000.000, con lo que una participación del 25% equivaldría a un valor objetivo de USD 3.000.000 en un horizonte de 3 a 5 años.

3.3. Semana 4

La cuarta semana se abocó a validar uno de los supuestos críticos: La existencia del mercado objetivo.

3.3.1. Experimento 1: Interés Inicial en el Producto

Siguientes Supuestos a Validar

1. A. (CS): Los usuarios/clientes de proveedores de infraestructura y plataformas como servicio en la nube (ej: Heroku, AWS, Rackspace, etc) presentan los dolores diagnosticados en la evidencia del problema: Dificultad en acceso, conciliación y búsqueda en los logs, además de inconveniencia en el uso de los logs.
2. B. (VP): Nuestros usuarios/clientes no quieren preocuparse de archivos ni de en qué máquina “viven” los logs.
3. C. (VP): Nuestros usuarios/clientes quieren ver en tiempo real como se comportan sus aplicaciones (ver el “latido del corazón” de su aplicación).

Experimento

Definición. Invitar a desarrolladores de aplicaciones en la nube a participar como beta-testers del producto, explicando en términos generales sus objetivos y características. Los desarrolladores interesados deben seguir los siguientes pasos: (a) confirmar su interés de participar como beta testers de la aplicación sin saber demasiado sobre su

naturaleza mas allá de que se trataba de logs, (b) recibir un mail con mas detalles y en base a eso darse de alta en <http://beautifullogs.co/apps> usando su usuario y contraseña de GitHub, (c) registrar su aplicación en la aplicación y (d) conectarla manualmente para recibir los logs.

Si bien algunos pasos como (c) y (d) serían eliminados en versiones mas desarrolladas del producto, se consideró que tener una serie considerable de pasos era positivo para este experimento pues demuestra un nivel de interés de parte del desarrollador que llegue hasta el último de ellos.

Resultado Esperado. Que al menos el 10% de los interesados en hacer de beta testers (paso a) lleguen hasta el final del proceso (paso d).

Ejecución. Para enfocar el experimento en un nicho muy definido se decidió buscar beta testers que ya cuenten con aplicaciones en Heroku, uno de los proveedores PaaS usado por los desarrolladores de aplicaciones en la nube.

Para el paso (a) el autor contactó a varios grupos de posibles usuarios:

1. Conocidos personales: Desarrolladores amigos o conocidos que el autor consideró que tenían una alta posibilidad de convertirse en beta testers. De 11 contactos, 4 personas aceptaron sumarse al proceso.
2. Lista de correo "RubySur": Desarrolladores latinoamericanos especializados en lenguaje Ruby, uno de los más usados en Heroku. De aquí surgió 1 voluntario.
3. Lista de correo "Dynlang Chile": Desarrolladores chilenos de varios lenguajes (Python, Ruby, Javascript), todos con alta proporción de uso en Heroku. De aquí surgieron otros 10 voluntarios.

Para dar las instrucciones para los pasos (b), (c) y (d) se construyó una lista de correo en el servicio MailChimp desde donde se enviaron las instrucciones que se pueden ver en la figura 3.4.

¡Hola << Test First Name >>!

Sigue estas instrucciones para conectar tu app Heroku a BeautifulLogs:

1. Ingresa con tu **usuario y contraseña de GitHub** en <https://beautifullogs.co/apps/>
2. Haz click en "New App", ingresa el nombre de tu app heroku y haz click en "Create"
3. Tu app será creada y el status de tu app indicará la dirección en la que BeautifulLogs está recibiendo logs para tu app (activa el desplegar imágenes en este email para ver la imagen abajo):

test2000  - Listening on syslog://test2000.beautifullogs.co:24762 (TCP)

 Edit  View Logs  View Stats

4. Copia la dirección partiendo desde "syslog" y terminando el el puerto. En el ejemplo el texto a copiar sería "syslog://test2000.beautifullogs.co:24762". En la consola dentro de tu app heroku ejecuta el siguiente comando:

```
heroku drains:add [direccion syslog]
```

¡Listo! Ahora puedes hacer click sobre "View Logs" o simplemente dirigirte a [https://\[nombre-de-tu-app\].beautifullogs.co](https://[nombre-de-tu-app].beautifullogs.co) y podrás ver tus logs en tiempo real y hacer búsquedas sobre ellos.

Si algo no funciona (o no parece funcionar) no dudes en contactarnos respondiendo este mismo email. O contactándome directamente a leo.soto@continuum.cl.

¡Pronto agregaremos más features que te contaremos por esta misma vía!

Copyright © 2014 BeautifulLogs. All rights reserved.

Figura 3.4: Mail enviado a los voluntarios para beta-tester

De acuerdo a las estadísticas de MailChimp (ver figura 3.5) 10 de los 15 voluntarios abrieron el email dentro del día en que fue enviado, y 6 de los voluntarios hicieron click en el link que les permitía darse de alta.

Origen	Nro voluntarios	Nro voluntarios que llegó al paso (d)	%
Conocidos	4	1	25%
Ruby Sur	1	0	0%
Dynlang Chile	10	2	20%

Tabla 3.1: Resultado desglosado por origen de los voluntarios

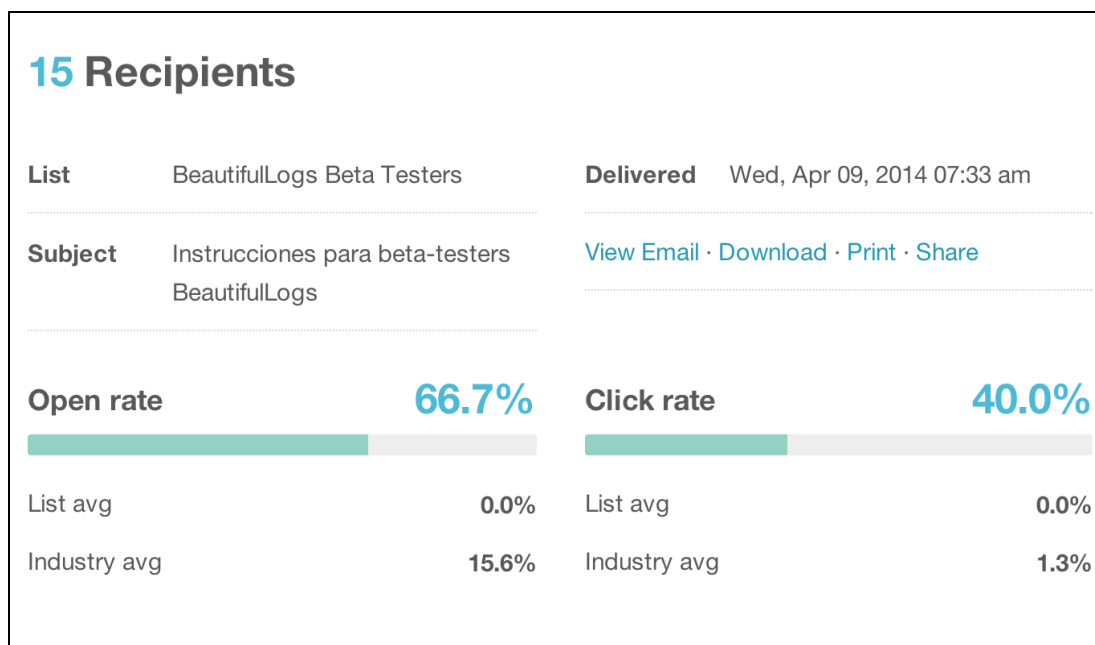


Figura 3.5: Reporte de reacción frente al email enviado a los voluntarios para beta-tester

De acuerdo a las estadísticas de la base de datos de beautifullogs.co, 5 de esos 6 voluntarios se registraron exitosamente (paso b), 4 inscribieron su aplicación (paso c) y 3 conectaron exitosamente los logs de su aplicación al sistema (paso d).

Resultado. El 20% de los voluntarios llegaron a conectar sus aplicaciones (o el 30% si se considera como universo quienes abrieron el email de instrucciones) El cuadro 3.1 muestra el desglose según el origen de los voluntarios.

El experimento se considera satisfactorio en cuanto a demostrar que hay un interés por este tipo de aplicaciones a pesar de la existencia de otras herramientas lo que es compatible con los supuestos A, B y C. En particular, se puede considerar que el supuesto A está razonablemente validado, pues un quinto de los voluntarios pasó por el trámite de gastar su tiempo conectando sus aplicaciones al sistema, lo que demuestra que esperan que resuelva problemas que no están cubiertos aún por las herramientas que están usando.

3.3.2. Canvas Resultante

Se reflejó en verde (como supuesto validado) el hecho de que los clientes actuales de heroku son un nicho de clientes potenciales para BeautifulLogs.

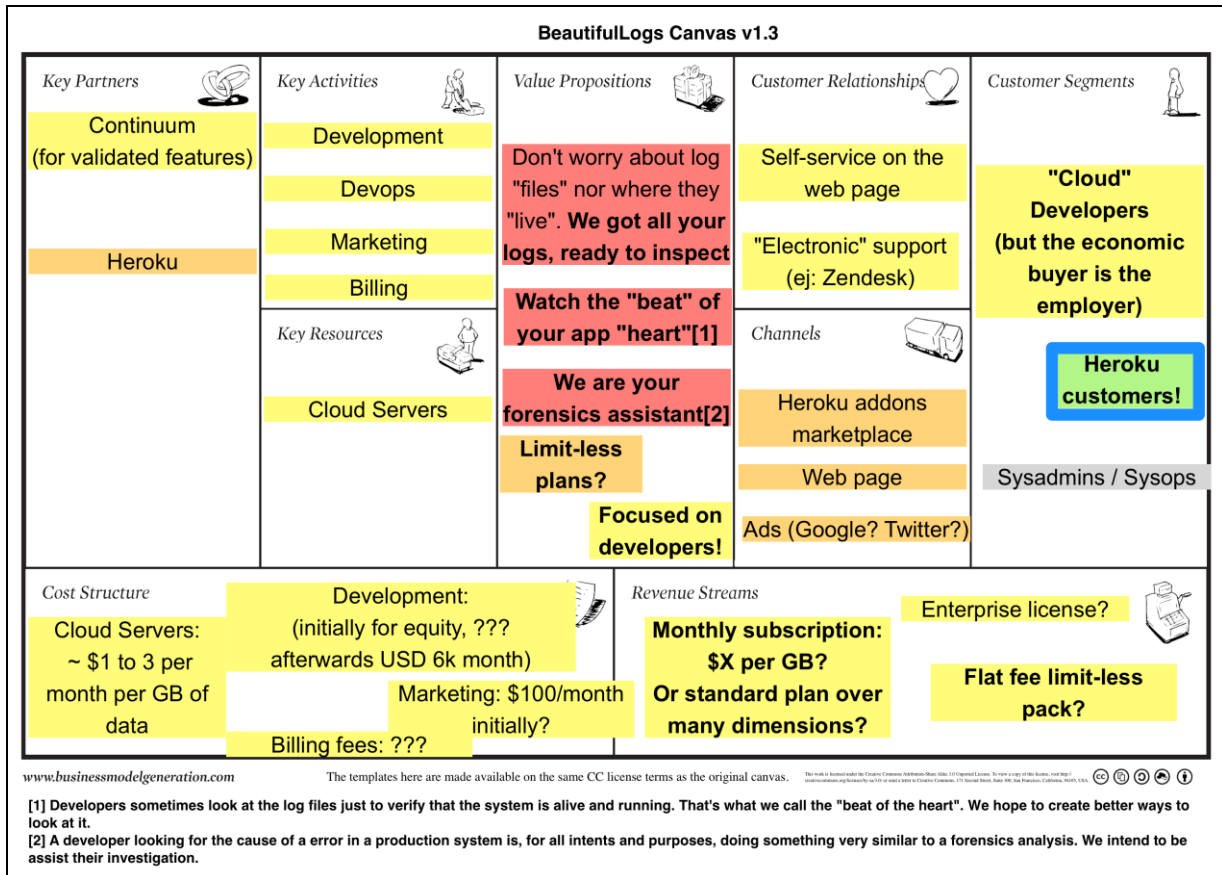


Figura 3.6: Canvas v1.3

3.4. Semana 5

En la quinta semana se procedió a validar la propuesta de valor, donde residían los supuestos más críticos del canvas en este momento.

3.4.1. Experimento 2: Propuesta de Valor (parte 1)

Siguientes Supuestos a Validar

1. B. (VP): Nuestros usuarios/clientes no quieren preocuparse de archivos ni de en qué máquina "viven" los logs.
2. C. (VP): Nuestros usuarios/clientes quieren ver en tiempo real como se comportan sus aplicaciones (ver el "latido del corazón" de su aplicación).

3. D. (VP): Nuestros usuarios/clientes valoran el “apoyo forense” a la hora de encontrar errores inspeccionando los logs.

Experimento

Definición. Encuestar a los 15 usuarios beta del producto reclutado en el experimento 1 con una simple pregunta: ¿Qué funcionalidad prefieres para la siguiente versión de BeautifulLogs? Se ofrecieron dos alternativas:

1. Árbol navegable de archivos de logs según donde se originan y que nombres de archivos tienen originalmente. Ver figura 3.7.
2. Estadísticas y métricas que transforman las líneas de logs en gráficos más fácilmente interpretable con métricas en tiempo real. Ver figura 3.8.

The screenshot displays the BeautifulLogs interface. On the left, a file tree shows a directory structure for 'heroku' containing 'app.log', 'router.log', 'influxdb.log', 'mongodb' (with sub-files 'mongo.log' and 'output.log'), 'nginx', 'rabbitmq', and 'redis.log'. The 'app.log' file is selected. The main area on the right shows a log stream with entries from 2014-04-07, including details like 'heroku router at=info method=GET path=/empleos/disenio-ux/web-interface-developer--2 host=www.getonbrd.com.ar request_id=2d265beb-b7f7-4a4b-8aeb-87fe4144067b fwd="66.249.66.123" dyno=web.1 connect=1ms service=96ms status=302 by 2014-04-07 20:14:32+00:00 app web.1 cache: [GET /assets/logo-getonboard-ee26f53c4d20e5c16b303d41f880bf23.png] fresh' and '2014-04-07 20:15:08+00:00 app web.1 Completed 200 OK in 174ms (Views: 109.0ms | ActiveRecord: 58.2ms)'. At the bottom, there is a search bar with the text 'Search in your logs', a magnifying glass icon, and a 'Q' button. On the right side of the bottom bar, there are checkboxes for 'tail -f', 'Apps', and 'Settings'.

Figura 3.7: Maqueta de árbol navegable

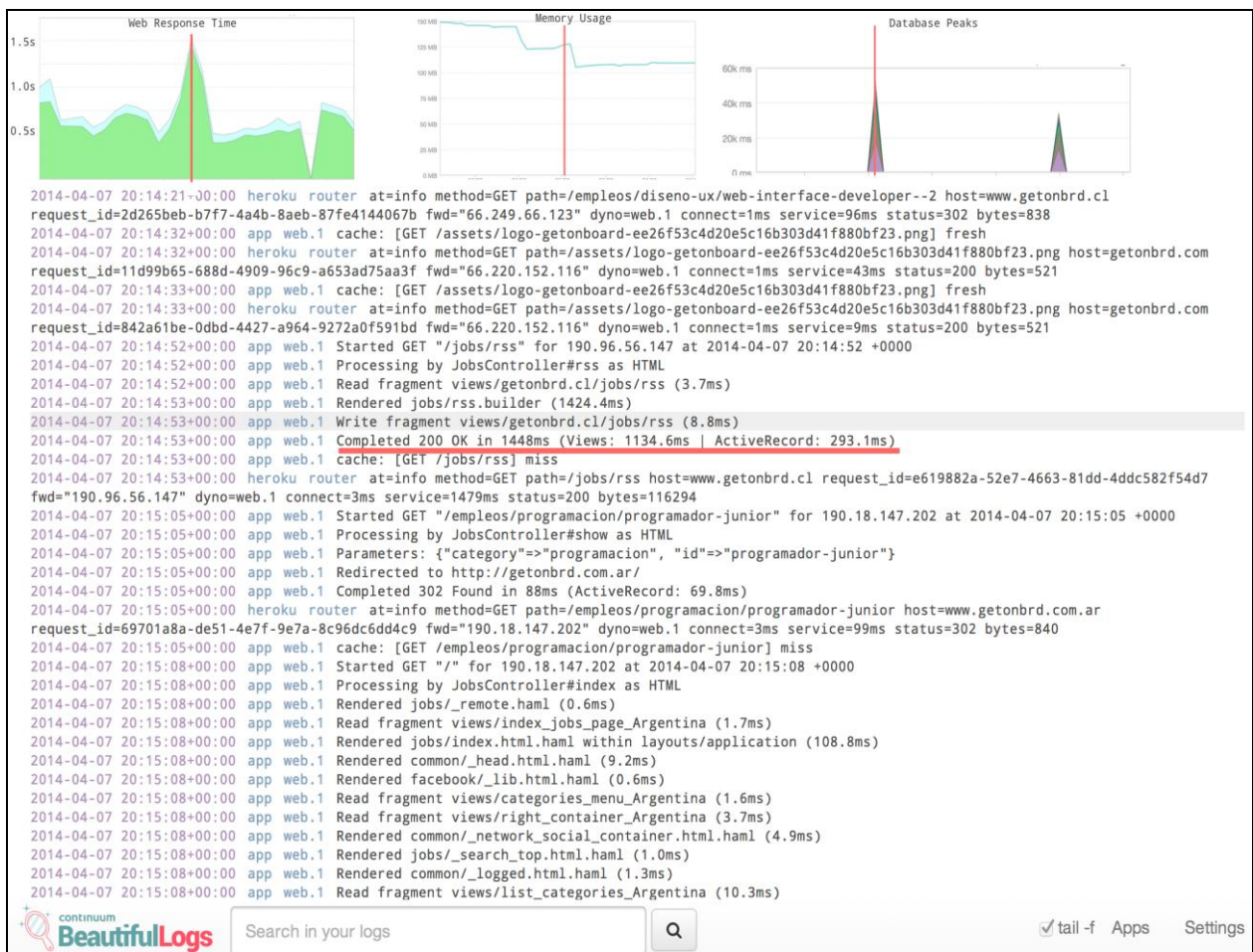


Figura 3.8: Maqueta de estadísticas y métricas en tiempo real

Resultado Esperado. Que más del 60% de los encuestados escojan la segunda alternativa, lo que validaría el diagnóstico de necesidades del cliente de los supuestos B y C.

Ejecución. En una primera instancia se envió la encuesta por email (vía MailChimp). Lamentablemente, menos del 30% de los destinatarios manifestaron una preferencia a una semana de enviada la encuesta, lo que llevó al autor a cambiar la forma de obtener feedback de parte de los usuarios.

En una segunda instancia, el autor contacto con emails personalizados a cada uno de los usuarios beta (y vía telefónica con los usuarios en que esto fue posible). Además, y en función de la respuesta de los usuarios, se profundizó en los motivos de cada usuario para comprender su decisión.

Asimismo, el autor decidió no aumentar el universo de encuestados (que era la alternativa en caso que los 15 usuarios originales no respondieran). Esto debido a que una rotación en los usuarios puede mermar la calidad, representatividad y por sobre todo la consistencia de los resultados obtenidos en los experimentos. Si bien es cierto

que 15 usuarios no son una muestra significativa en términos estadísticos, son un grupo manejable para una etapa donde (como se verá en la siguiente sección) no sólo importan las respuestas, sino que también comprender las motivaciones detrás de ellas. De esta forma, se podrá realizar un análisis etnográfico que, a pesar de ser relativamente superficial, permitirá comprender mucho mejor al cliente objetivo.

Resultado. Del universo de 15 personas, 14 respondieron a tiempo el mail personalizado. De ellas 11 (79 %) prefirió la alternativa de estadísticas en tiempo real por sobre la navegación jerárquica. De las 3 que prefirieron la alternativa de navegación, se consiguió mayor feedback de parte de dos de ellos:

- “Me parece que guardar búsquedas y tenerlas disponibles rápidamente a un click es muy útil para filtrar rápidamente la sábana de logs”
- “Normalmente sé donde empezar a buscar un error y sé el nombre del archivo donde ocurre...bueno, casi siempre. Igual es bueno que esté el filtro para esas veces en que uno quiere empezar enfocándose en algo específico”

El primero de estos usuarios claramente no entendió la idea del árbol de archivos presentado, creyendo que eran búsquedas guardadas en lugar de un árbol de archivos de origen de la información de los logs. Tras aclarar la situación, el usuario cambió su preferencia hacia las estadísticas en tiempo real.

Además el autor quiso descartar que alguno de los 11 usuarios que prefirieron la alternativa de estadísticas y métricas haya interpretado erróneamente el árbol de archivos. Tras enviar un mail aclaratorio y ofrecer la posibilidad de cambiar la opción, ninguno de esos 11 usuarios notificó al autor de un cambio de preferencias.

Por tanto, de las 14 respuestas se concluye con una preferencia de 12 personas por las estadísticas y métricas en tiempo real. En consecuencia, el experimento se considera satisfactorio en cuanto a demostrar que hay un interés por monitorear el comportamiento en tiempo real sus aplicaciones a partir de los logs y que, en términos relativos preocuparse por los nombres de archivos y las máquinas donde se originaron los logs no es lo más importante lo que es compatible con los supuestos B y C.

De esta manera, se puede considerar que los supuestos B y C están razonablemente validados.

3.4.2. Canvas Resultante

Como consecuencia del experimento, se reflejaron como validados dos elementos de la propuesta de valor del canvas. Además, se le dió mayor prioridad a la búsqueda de planes para suscripción mensual y se introdujeron nuevos elementos críticos como la capacidad de convencer al comprador económico y la posibilidad de cobrar un precio mayor a partir de la mejor propuesta de valor.

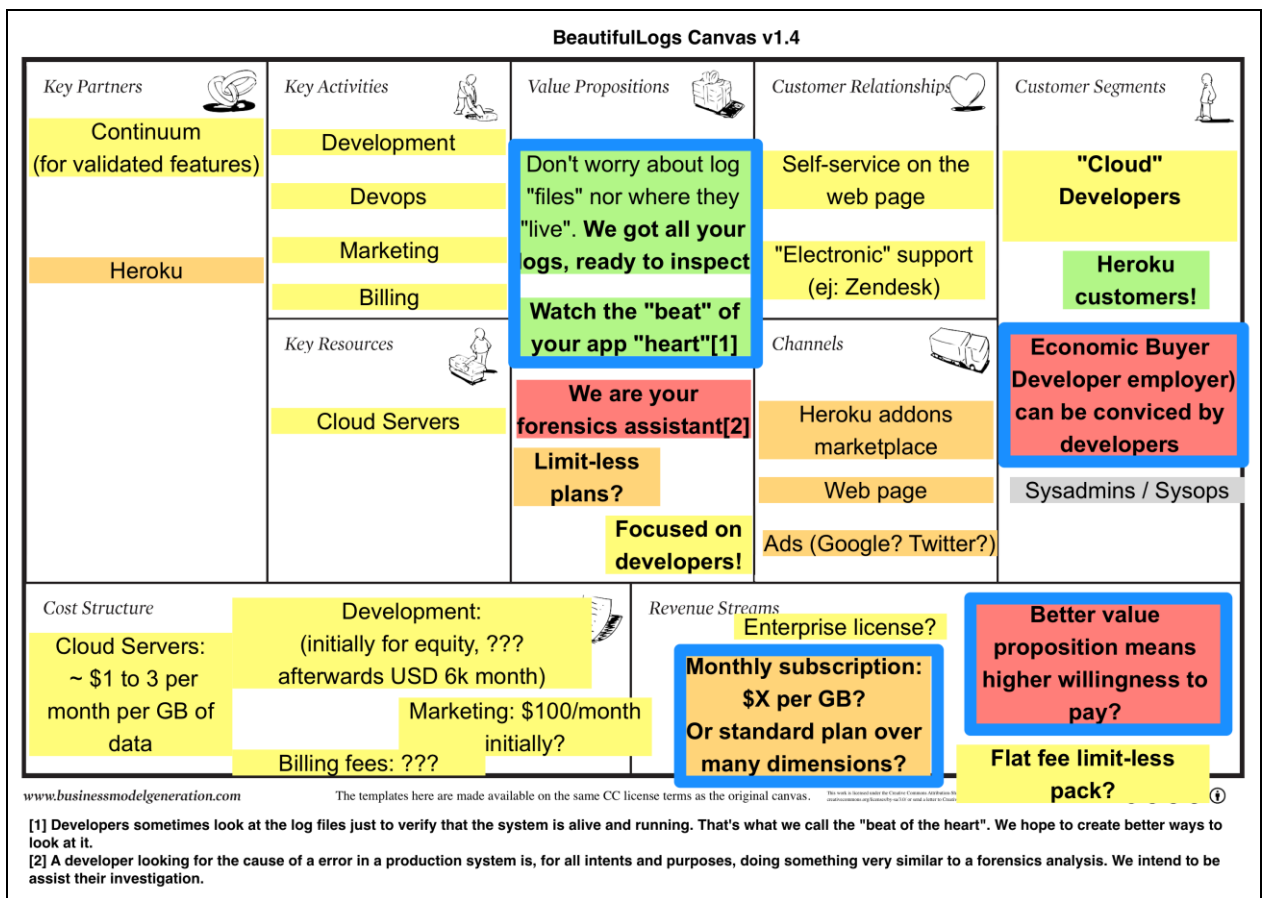


Figura 3.9: Canvas v1.4

3.5. Semana 6

En la sexta semana el experimento se enfocó la última parte crítica de la propuesta de valor que aún no estaba validada.

3.5.1. Experimento 3: Propuesta de Valor (parte 2)

Siguientes Supuestos a Validar

1. D. (VP): Nuestros usuarios/clientes valoran el "apoyo forense" a la hora de encontrar errores inspeccionando los logs.
2. E. (VP, R\$): La mejor propuesta de valor de BeautifulLogs le permite cobrar un precio mayor que la competencia (es decir, la diferenciación es valorada por los usuarios/clientes).
3. F. (CS, R\$): Nuestros usuarios/clientes (desarrolladores) tienen el poder de convencer al comprador económico (sus empleadores) para efectuar la compra.

Experimento

Definición. De igual manera que el experimento 2, encuestar a los 15 usuarios beta del producto reclutado en el experimento 1 con una simple pregunta: ¿Qué funcionalidad prefieres para la siguiente versión de BeautifulLogs? Nuevamente se ofrecieron dos alternativas:

1. Búsquedas guardadas y alertas manuales. Ver figura 3.10
2. Insights automáticos determinados por BeautifulLogs y que pueden indicar errores o problemas a partir del contenido de los logs. Ver figura 3.11.

```
2014-04-07 20:15:18+00:00 app web.1 Rendered jobs/_search_top.html.haml (1.7ms)
2014-04-07 20:15:18+00:00 app web.1 Rendered common/_logged.html.haml (1.2ms)
2014-04-07 20:15:18+00:00 app web.1 Read fragment views/list_categories_Argentina (0.7ms)
2014-04-07 20:15:18+00:00 app web.1 Rendered common/_sub_footer.html.haml (1.0ms)
2014-04-07 20:15:18+00:00 app web.1 Rendered common/_footer.html.haml (0.6ms)
2014-04-07 20:15:18+00:00 app web.1 Rendered common/_analytics.html.haml (0.1ms)
2014-04-07 20:15:18+00:00 app web.1 Completed 200 OK in 136ms (Views: 93.6ms | ActiveRecord: 40.9ms)
2014-04-07 20:15:18+00:00 app web.1 cache: [GET /] miss
2014-04-07 20:15:18+00:00 heroku router at=info method=GET path=/ host=www.getonbrd.com.ar request_id=80591cce-4dfd-4055-b1b8-3436c7e68c1b fwd="190.18.147.202" dyno=web.1 connect=1ms service=150ms status=200 bytes=38564
2014-04-07 20:15:18+00:00 app web.1 cache: [GET /assets/continuum-eb5fbfe4332a8c90227c7b9909795129.png] fresh
2014-04-07 20:15:18+00:00 heroku router at=info method=GET path=/assets/continuum-eb5fbfe4332a8c90227c7b9909795129.png host=getonbrd.com request_id=c98e2364-955a-4130-8968-2b13db8cb79b fwd="190.18.147.202" dyno=web.1 connect=1ms service=14ms status=200 bytes=1934
2014-04-07 20:15:24+00:00 app web.1 cache: [GET /assets/favicon-10f86194bb686ef534c50ab040c1961d.png] fresh
2014-04-07 20:15:24+00:00 heroku router at=info method=GET path=/assets/favicon-10f86194bb686ef534c50ab040c1961d.png host=getonbrd.com request_id=9e9200e4-91ac-4eae-ba8e-e0dc2ad0669c fwd="190.18.147.202" dyno=web.1 connect=1ms service=5ms status=200 bytes=783
2014-04-07 20:15:45+00:00 heroku router at=info method=GET path=/categorias/desarrollo-mobile/rss host=www.getonbrd.cl request_id=7fd5445-7c9f-4a19-88b1-a8229c4f35b2 fwd="50.143.116.46" dyno=web.1 connect=1ms service=14ms status=404 bytes=1011
2014-04-07 20:15:45+00:00 app web.1 Started GET "/categorias/desarrollo-mobile/rss" for 50.143.116.46 at 2014-04-07 20:15:45 +0000
2014-04-07 20:15:45+00:00 app web.1
2014-04-07 20:15:45+00:00 app web.1 ActionController::RoutingError (No route matches [GET] "/categorias/desarrollo-mobile/rss"):
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/actionpack-3.2.13/lib/action_dispatch/middleware/debug_exceptions.rb:21:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/actionpack-3.2.13/lib/action_dispatch/middleware/show_exceptions.rb:56:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/railties-3.2.13/lib/rails/rack/logger.rb:32:in `call_app'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/railties-3.2.13/lib/rails/rack/logger.rb:16:in `block in call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/activerecord-3.2.13/lib/active_record/connection_adapters/abstract_adapter.rb:22:in `tagged'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/railties-3.2.13/lib/rails/rack/logger.rb:16:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/actionpack-3.2.13/lib/action_dispatch/middleware/request_id.rb:22:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-1.4.5/lib/rack/methodoverride.rb:21:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-1.4.5/lib/rack/runtime.rb:17:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/activerecord-3.2.13/lib/active_record/connection_adapters/abstract_adapter.rb:22:in `tagged'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-1.4.5/lib/rack/lock.rb:15:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/actionpack-3.2.13/lib/action_dispatch/middleware/static.rb:63:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-cache-1.2/lib/rack/cache/context.rb:136:in `forward'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-cache-1.2/lib/rack/cache/context.rb:245:in `fetch'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-cache-1.2/lib/rack/cache/context.rb:185:in `lookup'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-cache-1.2/lib/rack/cache/context.rb:66:in `call!'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-cache-1.2/lib/rack/cache/context.rb:51:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/railties-3.2.13/lib/rails/engine.rb:479:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/railties-3.2.13/lib/rails/application.rb:223:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/railties-3.2.13/lib/rails/railtie/configurable.rb:30:in `method_missing'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/rack-cors-0.2.9/lib/rack/cors.rb:54:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/unicorn-4.6.3/lib/unicorn/http_server.rb:552:in `process_client'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/unicorn-4.6.3/lib/unicorn/http_server.rb:632:in `worker_loop'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/newrelic_rpm-3.6.6.147/lib/new_relic/agent/instrumentation/unicorn_instrumentation.rb:22:in `call'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/newrelic_rpm-3.6.6.147/lib/new_relic/agent/instrumentation/unicorn_instrumentation.rb:22:in `block (4 levels) in <stop (required)>'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/unicorn-4.6.3/lib/unicorn/http_server.rb:500:in `spawn_missing_workers'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/unicorn-4.6.3/lib/unicorn/http_server.rb:142:in `start'
2014-04-07 20:15:45+00:00 app web.1 vendor/bundle/ruby/1.9.1/gems/unicorn-4.6.3/lib/unicorn/http_server.rb:126:in `<start (required)>'
```

BeautifulLogs Search in your logs Q Favorite search & alerts: ★ Routing Error ★ /ad/new/payment ↓ Error R14 » tail -f Apps Settings

Figura 3.10: Maqueta de búsquedas y alertas manuales

```

2014-04-07 20:13:41+00:00 app web.1 Rendered jobs/_search_top.html.haml (1.0ms)
2014-04-07 20:13:41+00:00 app web.1 Rendered common/_logged.html.haml (0.9ms)
2014-04-07 20:13:41+00:00 app web.1 Read fragment views/list_categories_Chile (5.2ms)
2014-04-07 20:13:41+00:00 app web.1 Rendered common/_sub_footer.html.haml (5.4ms)
2014-04-07 20:13:41+00:00 app web.1 Rendered common/_footer.html.haml (0.7ms)
2014-04-07 20:13:41+00:00 app web.1 Rendered common/_analytics.html.haml (0.1ms)
2014-04-07 20:13:41+00:00 app web.1 Completed 200 OK in 547ms (Views: 394.1ms | ActiveRecord: 77.8ms)
🔴 Requests to /empleos/disen0-ux/front-end-developer--13 usually take 150ms-250ms to render. 547ms is too high.

2014-04-07 20:13:41+00:00 app web.1 cache: [GET /empleos/disen0-ux/front-end-developer--13] miss
2014-04-07 20:13:41+00:00 heroku router at=info method=GET path=/empleos/disen0-ux/front-end-developer--13 host=www.getonbrd.cl request_id=8775e4c6-fbcd-4ec4-ba82-348cd10cfc44
fwd="201.246.221.61" dyno=web.1 connect=3ms service=583ms status=200 bytes=25775
2014-04-07 20:13:50+00:00 app web.1 Started GET "/empleos/disen0-ux/productor-web-en-practica" for 190.45.120.225 at 2014-04-07 20:13:50 +0000
2014-04-07 20:13:50+00:00 app web.1 Processing by JobsController#show as HTML
2014-04-07 20:13:50+00:00 app web.1 Parameters: {"category"=>"disen0-ux", "id"=>"productor-web-en-practica"}
2014-04-07 20:13:50+00:00 app web.1 Rendered jobs/_company_logo.html.haml (0.7ms)
2014-04-07 20:13:50+00:00 app web.1 Rendered jobs/_job_header.haml (8.3ms)
2014-04-07 20:13:50+00:00 app web.1 Rendered jobs/_share_job.html.haml (2.2ms)
2014-04-07 20:13:50+00:00 app web.1 Rendered jobs/_company_logo.html.haml (0.6ms)
2014-04-07 20:13:50+00:00 heroku router at=info method=GET path=/assets/tooltipster-5b87808fde80cf25c06b68043b2eae48.css host=getonbrd.com request_id=144f3435-d415-4fd8-b9e8-fc39a3506f99
fwd="190.45.120.225" dyno=web.1 connect=3ms service=6ms status=200 bytes=7784
2014-04-07 20:13:50+00:00 app web.1 Completed 200 OK in 84ms (Views: 50.2ms | ActiveRecord: 21.0ms)
2014-04-07 20:13:50+00:00 app web.1 cache: [GET /empleos/disen0-ux/productor-web-en-practica] miss
2014-04-07 20:13:50+00:00 app web.1 cache: [GET /assets/application-6f1ac7a12a96a8277a518fa032a7f024.css] fresh
2014-04-07 20:13:50+00:00 app web.1 cache: [GET /assets/tooltipster-5b87808fde80cf25c06b68043b2eae48.css] fresh
2014-04-07 20:13:51+00:00 app web.1 cache: [GET /assets/ajaxful_rating-aal3ba0467df7c2908e7c4efd525a1d.css] fresh
2014-04-07 20:13:51+00:00 app web.1 cache: [GET /assets/application-1de7f5d76c87629ef26bfa1f3d215f4d.js] fresh
2014-04-07 20:13:51+00:00 heroku router at=info method=GET path=/assets/application-1de7f5d76c87629ef26bfa1f3d215f4d.js host=getonbrd.com request_id=5dfa9a34-f25d-489d-89d2-9db29b7ecda6
fwd="190.45.120.225" dyno=web.1 connect=2ms service=18ms status=200 bytes=179334
🔴 Serving static content from heroku is slowing down your dynos. Consider using a CDN, such as Cloudflare.

2014-04-07 20:13:50+00:00 heroku router at=info method=GET path=/empleos/disen0-ux/productor-web-en-practica host=www.getonbrd.cl request_id=f083ef5a-a487-4412-a881-762f928fad79
fwd="190.45.120.225" dyno=web.1 connect=4ms service=102ms status=200 bytes=20943
2014-04-07 20:13:50+00:00 heroku router at=info method=GET path=/assets/application-6f1ac7a12a96a8277a518fa032a7f024.css host=getonbrd.com request_id=acee70d8-72dc-46ac-baf6-608cefaf627e
fwd="190.45.120.225" dyno=web.1 connect=2ms service=19ms status=200 bytes=33421
2014-04-07 20:13:53+00:00 app web.1 cache: [GET /assets/flags/ar_mini-5b5144d4e09960c41bff99bad38fcc2d.png] fresh
2014-04-07 20:13:53+00:00 app web.1 cache: [GET /assets/flags/cl_mini-f0d6e3184fb04fc1d7f4b90755911e3a.png] fresh
2014-04-07 20:13:53+00:00 app web.1 cache: [GET /assets/continuum-eb5fbfe4332a8c90227c7b9909795129.png] fresh
2014-04-07 20:13:53+00:00 heroku router at=info method=GET path=/assets/continuum-eb5fbfe4332a8c90227c7b9909795129.png host=getonbrd.com request_id=39b90dff-2ee1-44cf-9f15-449fbc89438
fwd="190.45.120.225" dyno=web.1 connect=1ms service=20ms status=200 bytes=1934
2014-04-07 20:13:53+00:00 app web.1 cache: [GET /assets/flags/pe_mini-eb52a427c345bb4bcc65201759465186.png] fresh
2014-04-07 20:13:53+00:00 app web.1 cache: [GET /assets/flags/us_mini-837af2b52264354c9935e0a1cc117c80.png] fresh
2014-04-07 20:13:53+00:00 app web.1 cache: [GET /assets/logo-getonboard-f67417eddfda19cfca219ffdbcc2f78c0.png] fresh
2014-04-07 20:13:53+00:00 app web.1 cache: [GET /assets/continuum-eb5fbfe4332a8c90227c7b9909795129.png] fresh
2014-04-07 20:13:53+00:00 heroku router at=info method=GET path=/assets/continuum-eb5fbfe4332a8c90227c7b9909795129.png host=getonbrd.com request_id=c4bed4bd-f698-4229-9811-1a9e87f0f3ce
fwd="97.107.140.44" dyno=web.1 connect=11ms service=19ms status=200 bytes=1934
2014-04-07 20:13:53+00:00 heroku router at=info method=GET path=/assets/flags/ar_mini-5b5144d4e09960c41bff99bad38fcc2d.png host=getonbrd.com request_id=fa84ce19-fcca-4e81-a322-c6d949d2d321
fwd="190.45.120.225" dyno=web.1 connect=2ms service=5ms status=200 bytes=865
2014-04-07 20:13:53+00:00 heroku router at=info method=GET path=/assets/flags/pe_mini-eb52a427c345bb4bcc65201759465186.png host=getonbrd.com request_id=38647d92-f29b-4fba-9bea-51d772c99ab2

```

Figura 3.11: Maqueta de insights automáticos

Resultado Esperado. Que más del 60% de los encuestados escojan la segunda alternativa, lo que validaría el diagnóstico de necesidades del cliente del supuesto D en cuanto a tener un “asistente” que ayude a encontrar pistas para resolver los problemas a los que se enfrenta un desarrollador que está explorando logs.

Ejecución. Tal como en el experimento anterior, el autor contactó con emails personalizados y vía chat a cada uno de los usuarios beta para obtener una preferencia entre las alternativas propuestas. De igual forma, en función de la respuesta de los usuarios, se profundizó en los motivos de cada usuario para comprender su decisión.

Resultado. Del universo de 15 personas, 14 respondieron a tiempo el mail personalizado. De ellas 9 (64 %) prefirieron la alternativa de establecer manualmente búsquedas y alertas. Sólo 4 (29 %) prefirieron los insights automáticos y 1 se mostró disconforme con ambas alternativas, señalando que prefiere otras herramientas para diagnóstico de errores como AirBrake.io (que notifica a los desarrolladores vía email cuando ocurre una excepción en un programa).

De los 9 que prefirieron la alternativa manual el feedback sobre las motivaciones puede clasificarse en dos categorías:

1. “Quiero tener el control de las alertas yo”. Las razones esgrimidas para esto van desde no “ensuciar” los logs con advertencias que pueden no ser importantes hasta no “quedar mal” cuando alguien no especializado vea el log y encuentre cosas que en realidad no son tan terribles marcadas como errores hasta que no quieren recibir SPAM en su email por cosas que no pueden predecir desde antes.
2. “No confío en que algo automático funcione así de bien”. En el fondo, la característica parece demasiado buena para ser verdad, y los usuarios prefieren algo que tal vez prometa menos pero dé certidumbre. Muchas veces los reparos de la categoría de falta de control desembocaban (después de varias preguntas “pero ¿por qué?”) en respuestas en esta categoría.

También hubo reparos menores respecto al tono de las sugerencias de los insights. Dos usuarios consideraron algo “ofensivo” el tono autoritario de algo que, finalmente, es simplemente una máquina analizando cosas y que (en palabras de uno de ellos) “no sabe más que el programador que hizo el programa”.

En cualquier caso, el resultado de 29% de preferencias por los insights automáticos está muy por debajo del 60% esperado. En las conversaciones más al detalle con los usuarios el autor verificó que esto no se debió a una mala interpretación de las maquetas. Muchos usuarios manifestaron un explícito rechazo a la funcionalidad propuesta en base a argumentos de falta de control y poca confianza en que la funcionalidad pudiera ser implementada correctamente para todos los casos.

Más aún: Al intentar usar la analogía del “asistente o apoyo forense” un usuario manifestó abiertamente que consideraba al computador una “herramienta” pero nunca un “asistente”. Al verificar esta posición con otros 5 usuarios que fueron entrevistados después de quien declaró aquello, todos se mostraron de acuerdo (en distinto grado, pero fundamentalmente de acuerdo) con la posición de que buscan una “herramienta” y no un “asistente”.

En consecuencia, el experimento invalida el supuesto D. Los usuarios no buscan (y/o no creen que sea factible implementar) un “apoyo forense” en un sistema como BeautifulLogs.

3.5.2. Canvas Resultante

Se reflejó en el canvas el supuesto fallido.

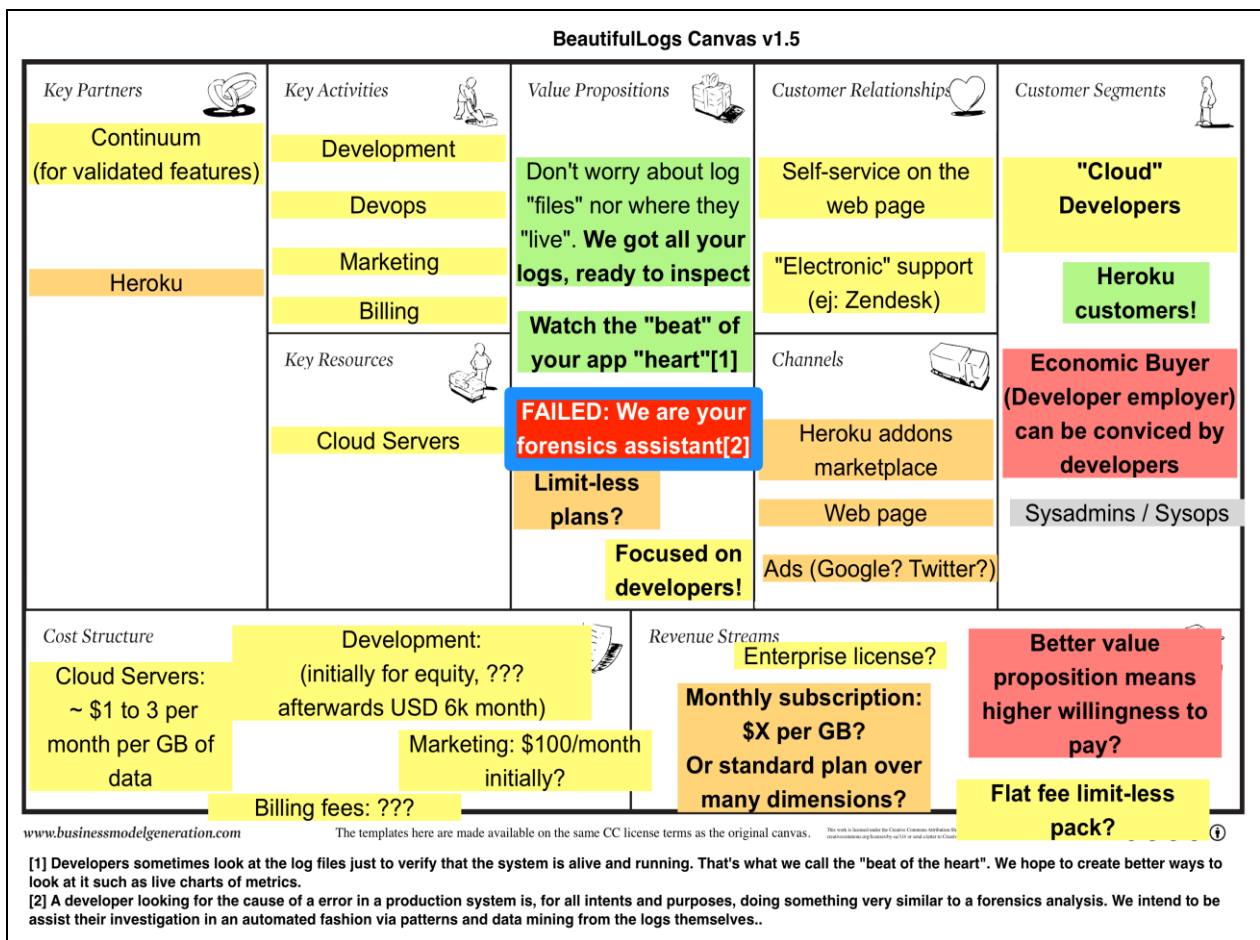


Figura 3.12: Canvas v1.5

3.6. Semana 7

En la séptima semana se pausó para evaluar los resultados adversos del experimento 3, corrigiendo el modelo para no depender de un supuesto que no ha podido ser validado.

3.6.1. Análisis del Problema Detectado: Posibles Razones

Inexistencia de la necesidad. Es posible que los usuarios no necesiten nunca (o muy rara vez) el apoyo automatizado en encontrar errores dentro de los logs. O puesto de otra forma, la ganancia marginal de esta funcionalidad puede ser muy menor al lado de otras técnicas ya utilizadas por los usuarios (como depuración en vivo).

Necesidad no descubierta. Una variante de lo anterior es que los usuarios no perciban hoy que necesitan este apoyo, debido a que los errores que podrían ser encontrados de esta manera no esten siendo detectados en absoluto hoy en día. En ese sentido, el experimento podría haber sido insuficiente en demostrar de una manera más palpable la necesidad.

Desconfianza a soluciones difíciles de comprender. También es posible que los usuarios no se sientan cómodos con la funcionalidad ofrecida debido a que es equivalente a presentar una “magia” que encontrará automáticamente algunos problemas, sin ofrecer predictibilidad sobre cuales serán los resultados exactos de estas alertas. Las dos grandes categorías de respuestas detalladas de los usuarios (“quiero tener el control de las alertas” y “no confío en que algo automático funcione así de bien”) apuntan en esta dirección. Los comentarios respecto al “tono autoritario” en que se presentarían las alertas automáticas podrían indicar que el problema principal es como se presenta la característica más que la característica en sí.

Síndrome NIH. El llamado síndrome NIH (de not invented here por sus singlas en inglés) se refiere a la propensión de evitar soluciones de origen “externo” y es especialmente relevante en la industria del software, como lo revela esta cita:

“In programming, it is also common to refer to the NIH ‘Syndrome’ as the tendency towards reinventing the wheel (reimplementing something that is already available) based on the belief that in-house developments are inherently better suited, more secure or more controlled than existing implementations.” [15].

Esta razón se compondría con la anterior para explicar el rechazo a las alertas o insights automáticos.

3.6.2. Análisis de Caminos a Seguir

Descartar propuesta de valor Si se acepta que la necesidad no existe, la manera más fácil de proseguir es simplemente eliminar la propuesta de valor. Después de todo, ya hay dos propuestas de valor validadas que ofrecen diferenciación respecto a los productos existentes actualmente.

Re-validar propuesta de valor modificada Si el problema es más bien la forma en que se presenta la funcionalidad y las expectativas que genera y no la funcionalidad en sí misma, es posible cambiar la propuesta de valor a algo como “Sugerencias o tips de alertas” que se complementa a la funcionalidad estándar de alertas manuales a la que los usuarios están acostumbrados y que les permita evaluar gradualmente el valor entregado por la funcionalidad mejorada.

Introducir propuesta de valor modificada y no revalidar inmediatamente Un compromiso entre las dos opciones anteriores: Introducir la propuesta de valor modificada en el modelo, pero no darle prioridad de validación. De este modo se podrá avanzar primero en otros supuestos críticos, relacionados con el precio y con los canales de compra de los usuarios. Esta es la opción que se decidió tomar, por no cerrar la puerta a futuros experimentos en un área tan clave como la propuesta de valor, pero establecer prioridades para validar otras áreas importantes del modelo.

3.6.3. Canvas Resultante

Se introdujo en el canvas el nuevo supuesto, sin darle una prioridad tan alta como otros elementos claves como el pricing.

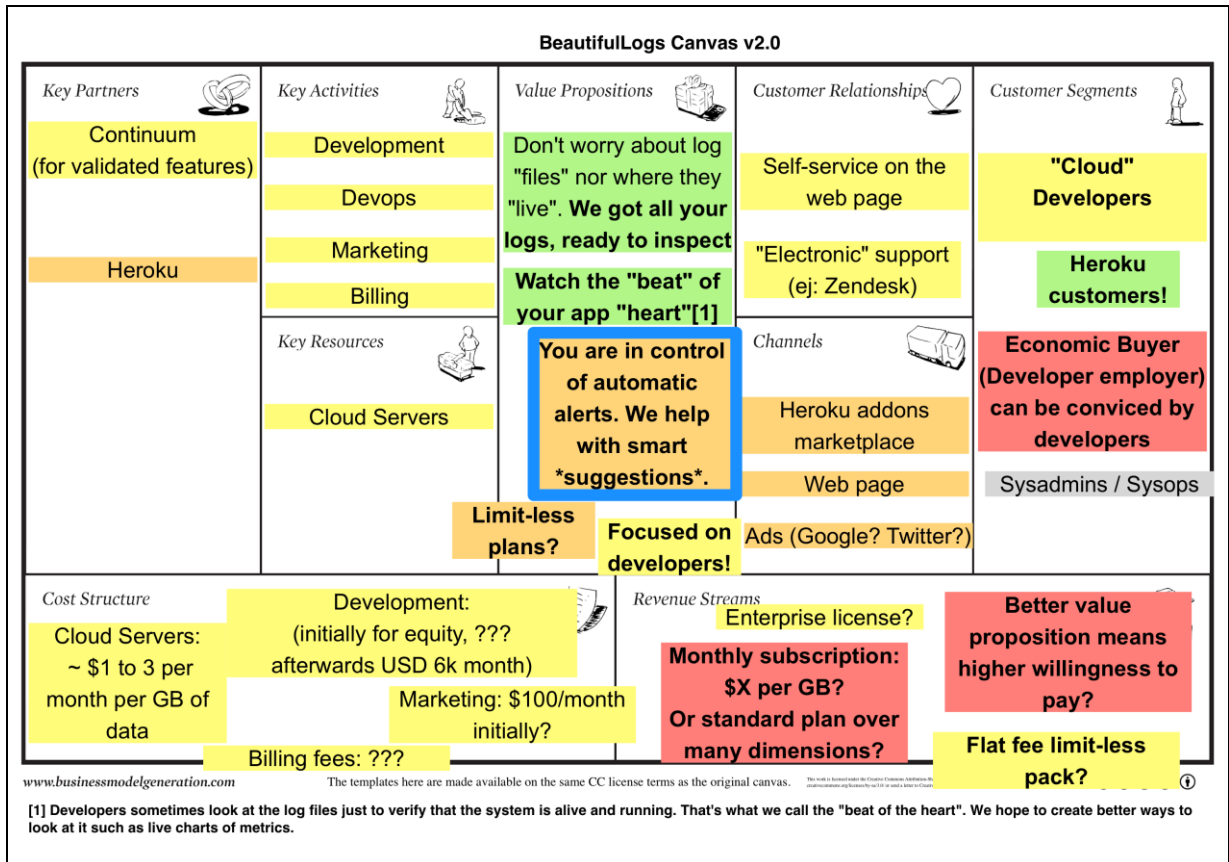


Figura 3.13: Canvas v2.0

3.7. Semana 8

En la semana 8 se retomaron los experimentos. Dado que la propuesta de valor se considera razonablemente validada, la presión pasa a validar los supuestos más críticos (marcados en rojo en el canvas v2.0), que tienen relación con el pricing y el comprador económico.

3.7.1. Experimento 4: Pricing (parte 1)

Siguientes Supuestos a Validar

1. E. (VP, R\$): La mejor propuesta de valor de BeautifulLogs le permite cobrar un precio mayor que la competencia (es decir, la diferenciación es valorada por los usuarios/clientes).

2. F. (CS, R\$): Nuestros usuarios/clientes (desarrolladores) tienen el poder de convencer al comprador económico (sus empleadores) para efectuar la compra.
3. G. (VP, R\$): Un dolor de nuestros usuarios/clientes cuando usan productos alternativos es tener límites en sus planes.

Experimento

Definición. Encuestar a los 14 usuarios¹, ofreciendo 5 características generales de sistemas de gestión de logs y pidiéndoles que elijan las 3 más importantes (en orden) a la hora de decidir cuánto pagar por un plan mensual.

El objetivo del experimento es obtener 3 dimensiones para hacer un análisis conjunto simple que permita estudiar esas características más importantes en más profundidad. (Un análisis conjunto se vuelve más complejo conforme más variables se introducen).

Las 5 dimensiones sugeridas son:

1. Más espacio de almacenamiento.
2. Búsqueda ilimitada (o más extendida que un par de semanas).
3. Generación de Métricas y/o Analíticas a partir de los logs.
4. Capacidades avanzadas de archivado y recarga posterior de logs.
5. Soporte telefónico (en adición al soporte online).

Resultado Esperado. Que las tres primeras opciones sean las escogidas por la mayoría de los usuarios, en el mismo orden presentado. Eso sería consistente con el supuesto de que los límites son un dolor para los usuarios y con que la diferenciación de BeautifulLogs (en cuanto a métricas) tenga potencial para generar una mayor disposición a pagar.

Para calcular las opciones “ganadoras” se le asignará 3 puntos a cada opción nombrada como primera prioridad por los usuarios, 2 puntos a cada opción nombrada como segunda prioridad y 1 punto a cada opción nombrada como tercera prioridad.

Ejecución. Como ya fue establecido por previos experimentos, el autor contactó personalmente via email a cada encuestado (y chat/teléfono cuando fue necesario para obtener respuestas a tiempo).

En los casos donde la respuesta de los usuarios fue sorpresiva o dubitativa se procedió a una conversación en mayor profundidad.

Resultado. La tabla 3.2 muestra los resultados individuales de la encuesta para cada uno de los 14 usuarios. Recordar que las características prioritarias son marcadas con 3, 2, y 1 punto, de mayor a menor prioridad (las dos menos prioritarias no llevan puntaje). El resultado total, sumando los puntajes asignados por cada usuario puede verse en la tabla 3.3.

¹ Uno de los 15 usuarios iniciales pidió dejar de ser encuestado en el proceso

ID	Más Almacenamiento	Búsqueda Ilimitada o Extendida	Métricas y/o Analíticas	Archivado y Recarga de Logs	Soporte Telefónico
U1	3	2	1		
U2	1	3	2		
U3	3	2	1		
U4	2	3			1
U5	1	3	2		
U6	3		2	1	
U7	3	2	1		
U8	3	2			
U9		3	2		1
U10	2		3	1	
U11	2	3	1		
U12	3	2	1		
U13		3	2		1
U14		2	3	1	

Tabla 3.2: Funcionalidades prioritarias para disponibilidad a pagar de cada usuario

Más Almacenamiento	Búsqueda Ilimitada o Extendida	Métricas y/o Analíticas	Archivado y Recarga de Logs	Soporte Telefónico
26	30	21	3	3

Tabla 3.3: Resultado final de prioridades para disponibilidad a pagar

En consecuencia, se valida que las tres funcionalidades prioritarias para los usuarios son las esperadas, aunque el orden de las prioridades varía sustantivamente de usuario a usuario. En el agregado, la Búsqueda ilimitada o extendida supera al almacenamiento, lo que es una señal en torno a que los usuarios valoran una mayor potencia en la búsquedas y/o el eliminar restricciones de cosas usualmente limitadas.

En cuanto a las conversaciones en mayor profundidad, se dió en muchos casos de que los usuarios tenían dificultad priorizando funcionalidades sin saber el precio que costaría cada cosa. El autor pidió a los encuestados hacer el mejor esfuerzo en determinar los 3 elementos que preferiría ver en una comparación mas aterrizada con precios incluidos (pues en efecto ese es el paso siguiente en el análisis conjunto).

Otro caso interesante fue un encuestado que se negó a elegir una tercera prioridad. Literalmente declaró “No me interesan las métricas, ni los archivos, ni el callcenter. Sólo quiero hartos espacio y buenas búsquedas”.

Como resultado del experimento no se pueden dar por validados aún los supuestos E ni G. Si bien existen indicios auspiciosos, se deberá esperar al análisis conjunto (siguiente experimento) para contar con una mejor información.

3.7.2. Canvas Resultante

El canvas no sufrió variaciones debido a que la parte 1 del experimento de pricing no ha llegado aún a resultados conclusivos.

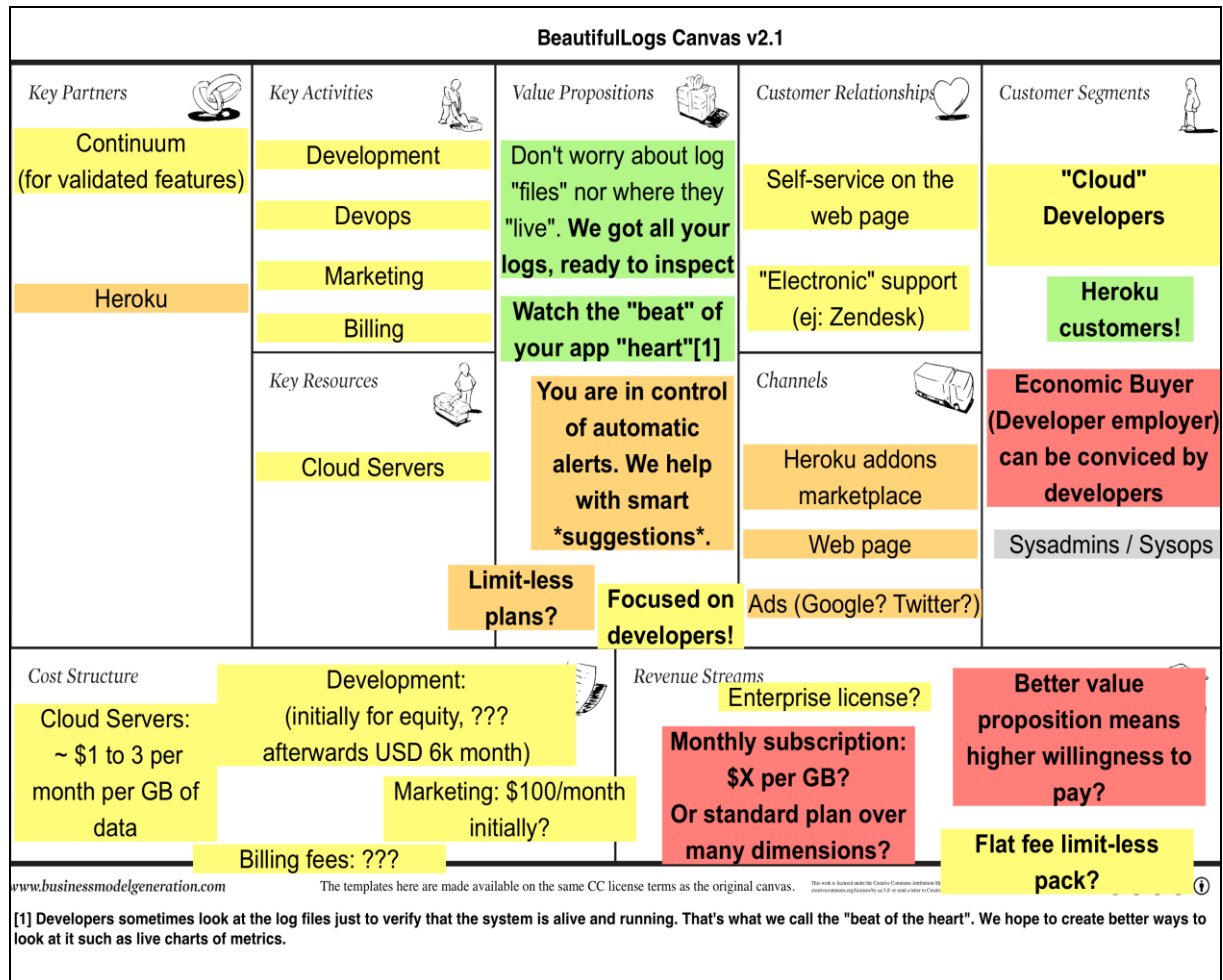


Figura 3.14: Canvas v2.1

3.8. Semanas 9 y 10

En las semanas 9 y 10 se completó el experimento de pricing, completando un análisis conjunto con los 14 usuarios, usando las 3 características declaradas como prioritarias por los mismos usuarios en el experimento anterior.

3.8.1. Experimento 5: Pricing (parte 2)

Siguientes Supuestos a Validar

1. E. (VP, R\$): La mejor propuesta de valor de BeautifulLogs le permite cobrar un precio mayor que la competencia (es decir, la diferenciación es valorada por los usuarios/clientes).
2. F. (CS, R\$): Nuestros usuarios/clientes (desarrolladores) tienen el poder de convencer al comprador económico (sus empleadores) para efectuar la compra.
3. G. (VP, R\$): Un dolor de nuestros usuarios/clientes cuando usan productos alternativos es tener límites en sus planes.

Experimento

Definición. Encuestar a los 14 usuarios, pidiéndoles ordenar de manera explícita 8 opciones de planes y precios para el producto, desde el más preferido al menos preferido.

Las opciones se basan en cuatro atributos, y cada atributo puede tomar uno de dos valores. A continuación se detallan dichos atributos y valores:

- **Búsqueda:** Antigüedad máxima de logs incluidos en una búsqueda. Los valores ofrecidos son 2 semanas (el promedio actual del mercado) o ilimitado.
- **Almacenamiento o Retención:** Cuánto espacio (en Gigabytes) habrá disponible. En este caso se obtuvo feedback temprano de usuarios quienes indicaron que preferirían no tener que calcular para cuánto les alcanza el espacio. El verdadero valor para ellos es por cuánto tiempo tendrán disponibles los logs en el sistema. Por lo tanto se ofrecieron valores que sugerían un lapso de tiempo de retención con un máximo de Gigabytes para evitar abusos. Los valores ofrecidos son 1 mes de 30GB (alrededor del promedio del mercado) y 1 año, o el equivalente de 300 GB.
- **Métricas:** Si se incluye el cálculo y presentación de métricas y/o analíticas a partir de los logs. Los valores ofrecidos son la ausencia o presencia de esta característica.
- **Precio:** El valor en dólares a pagar mensual por el servicio. Los valores ofrecidos son 50 dólares mensuales (aproximadamente el promedio del mercado) y 200 dólares mensuales (un precio objetivo si BeautifulLogs consigue establecerse como un proveedor premium en el nicho).

Las 8 opciones ofrecidas para rankear fueron las descritas por la tabla 3.4.

Opción	Búsqueda	Retención	Métricas	Precio
1	2 semanas	1 mes (30GB)	NO	200
2	Ilimitada	1 mes (30GB)	NO	50
3	2 semanas	1 año (300GB)	NO	50
4	Ilimitada	1 año (300GB)	NO	200
5	2 semanas	1 mes (30GB)	SI	50
6	Ilimitada	1 mes (30GB)	SI	200
7	2 semanas	1 año (300GB)	SI	200
8	Ilimitada	1 año (300GB)	SI	50

Tabla 3.4: Opciones ofrecidas en análisis conjunto

Nótese que la opción 1 es objetivamente la peor opción de todas, independiente de las preferencias relativas de cada atributo (menores características al más alto precio). De manera inversa, la opción 8 es objetivamente la mejor de todas (mayores características al menor precio). Esto permite identificar casos en que usuario ha contestado el ejercicio de manera inconsistente, sea por falta de una mayor explicación o de manera accidental.

Resultado Esperado. Que la suma de part-worths de los atributos probados supere al part-worth del precio en al menos 50% de los usuarios. En otras palabras, que el aumento de valor percibido por el usuario por las versiones aumentadas que podría ofrecer BeautifulLogs (Búsqueda ilimitada, 1 año de retención, Métricas/Analíticas) generen tanto valor que llegue a justificar un aumento de USD 150 en el precio a ofrecer.

Ejecución. A diferencia de otras encuestas más simples, este experimento requirió de mucho más trabajo caso a caso por parte del autor. Muchas respuestas fueron recibidas con inconsistencias (donde las opciones 1 y 8 no ocupaban el último y primer lugar respectivamente) y se debieron enviar nuevamente con mayor explicación.

Además 5 usuarios manifestaron de distinta forma que del feedback solicitado hasta ahora este es fue el más tedioso y difícil de visualizar correctamente. De todas formas, el experimento no se dió por concluido hasta obtener las 14 respuestas, lo que impactó en la duración del experimento que llegó a las 2 semanas.

Resultado. La tabla 3.5 muestra los resultados individuales, incluyendo el peso relativo de cada atributo en la decisión de qué plan elegir para cada usuario como un porcentaje (que corresponde al porcentaje del part-worth del atributo sobre la suma de los valores absolutos de part-worths totales). En rojo aparecen los casos donde la importancia del precio supera a la suma de todos los otros atributos.

ID	Post. Ranking ²	Búsqueda	Retención	Métricas	Precio
U1	8,5,2,3,4,6,7,1	21%	36%	0%	43%
U2	8,3,4,7,2,6,5,1	7%	7%	29%	57%
U3	8,6,7,4,3,2,5,1	36%	7%	50%	7%
U4	8,3,5,2,6,4,7,1	57%	21%	0%	21%
U5	8,2,3,5,4,6,7,1	29%	14%	0%	57%
U6	8,7,2,4,5,6,3,1	0%	57%	21%	21%
U7	8,5,2,4,6,7,3,1	7%	57%	7%	29%
U8	8,5,3,2,7,6,4,1	29%	57%	0%	14%
U9	8,3,2,5,4,7,6,1	14%	29%	0%	57%
U10	8,5,3,7,2,6,4,1	6%	19%	31%	44%
U11	8,3,2,6,4,7,5,1	6%	19%	31%	57%
U12	8,6,2,4,5,7,3,1	0%	57%	14%	29%
U13	8,2,5,6,3,4,7,1	31%	6%	19%	44%
U14	8,4,5,7,2,6,3,1	0%	14%	43%	43%

Tabla 3.5: Porcentaje de relevancia relativa de cada atributo según análisis conjunto

De los 14 encuestados, hay 10 que valoraron más la suma de características extra que una disminución del precio de 150 dólares. Además, hay 4 casos donde el peso relativo del atributo de retención o almacenamiento domina completamente su decisión y otro donde el horizonte de las búsquedas hace lo mismo.

Por tanto se puede considerar que los supuestos E y G, en cuanto a la mayor disposición a pagar por las características diferenciadoras de BeafullLogs y al dolor de tener límites en los planes ha sido razonablemente validada con los usuarios encuestados

3.8.2. Canvas Resultante

Se modificó la sección de flujos de ingresos para indicar la validación del precio a pagar por las características avanzadas, además de indicar la necesidad de dar con planes concretos que segmenten el mercado.

También se modificó en la propuesta de valor la idea de planes ilimitados por el concepto de planes generosos validados indirectamente por el experimento. Además, se le otorgó prioridad a determinar el costo de los servidores en la nube que ofrecerán el servicio para determinar si en una primera instancia el negocio es viable.

² Indica el ranking de la opción n-ésima. Por ejemplo la secuencia 8,7,2,3,4,5,6,1 indica que la opción 1 obtuvo el ranking 8vo, la opción 2 el ranking 7mo, la opción 3 obtuvo el 2do lugar de ranking, y así sucesivamente.

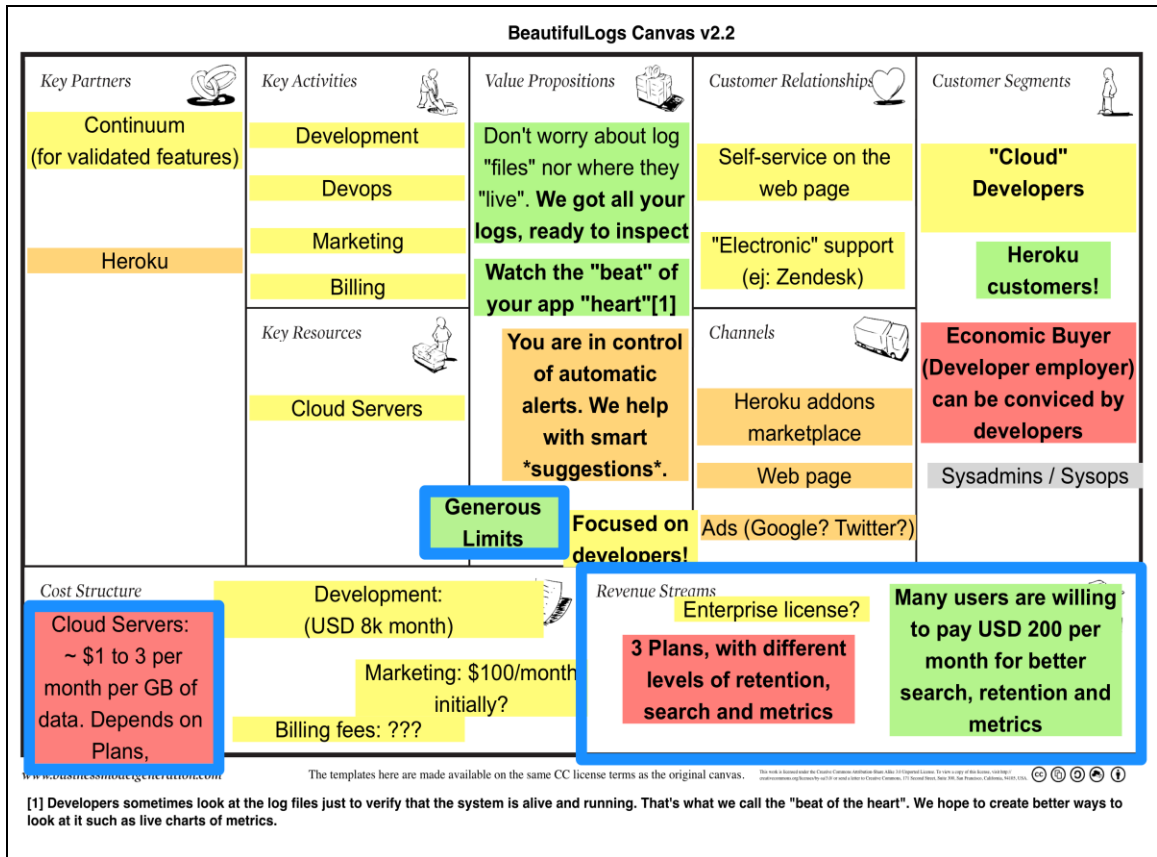


Figura 3.15: Canvas v2.2

3.9. Semana 11

El experimento de la semana 11 no requirió tanta interacción con los usuarios como los anteriores, pues a partir del análisis conjunto se pueden inferir planes con combinaciones que segmenten a los usuarios según su sensibilidad al precio. Es decir, definir un plan "económico" o "básico" que se ajuste a las necesidades de los usuarios que no están dispuestos a pagar más de 50 dólares mensuales; otro plan "recomendado" que se ajuste a los usuarios dispuestos a pagar un valor alrededor de 150 dólares y otro "premium" de 300 dólares.

3.9.1. Experimento 6: Definición de Planes

Siguientes Supuestos a Validar

1. F. (CS, R\$): Nuestros usuarios/clientes (desarrolladores) tienen el poder de convencer al comprador económico (sus empleadores) para efectuar la compra.

2. H. (CS, R\$, C\$): Podemos segmentar a los clientes en 3 distintos planes de precios.
3. I. (C\$\$, R\$, KR): Dentro del nicho de usuario de Heroku es posible llegar al breakeven en el costo de almacenamiento y procesamiento de las características extras ofrecidas por los planes definidos.

Experimento

Definición. A partir de la información obtenida en el análisis conjunto realizado en el experimento 5, definir 3 planes que segmenten a los usuarios según cuanto valúan los 3 atributos principales: GBs de almacenamiento (y el tiempo de retención asociado), búsqueda ilimitada y el cálculo de métricas. Usando la misma información se puede intentar predecir el plan que sería seleccionado por cada usuario.

Resultado Esperado. Que de acuerdo a la información obtenida en el análisis conjunto, al menos el 75% de los encuestados esté dispuesto a pagar uno de los planes definidos.

Ejecución. Los planes iniciales definidos por el autor (basados en la interacción con los usuarios y en comparación de precios con las soluciones existentes en el mercado) varían exclusivamente en el parámetro de retención/almacenamiento. Todos incluyen métricas y búsqueda ilimitada, de modo de posicionar el producto en un segmento avanzado dentro del nicho.

- Gold: 1 mes / 30 Gb de retención. USD 75 por mes.
- Platinum: 3 meses / 100 Gb de retención. USD 200 por mes.
- Diamond: 1 año / 500 Gb de retención. USD 600 por mes.

En teoría, esto permitiría segmentar a los clientes principalmente entre los planes Gold y Diamond, con el Platinum sirviendo como un plan de transición/comparación. Los clientes que valoran una retención de largo plazo (es decir, con una alta valoración por gigabytes extras) debieran tender hacia los planes mas altos.

En cuanto a los resultados del experimento 5, comparando los parts-worth de precio se puede aproximar un valor en dólares de cada uno de los 3 atributos de manera específica para cada encuestado.

Los parts-worth de precio corresponden a los utiles que entrega una disminución de 150 USD por mes en el precio. Por tanto, si por ejemplo un caso arrojó un parts-worth de 2 para el precio y de 1 para las métricas, se puede inferir que las métricas tienen un valor equivalente a 75 USD mensuales para ese usuario³.

³ Esto asume una función lineal en los utiles que los usuarios asignan a cada atributo. Si bien no es un supuesto con altas probabilidades de ser cierto en la práctica, provee una buena primera aproximación

ID	Búsqueda Ilimitada	Métricas	Incremento de 270 GBs	Incremento 1 GB	Plan Gold	Plan Platinum	Plan Diamond
U1	75,0	0,0	125,0	0,46	89	121	306
U2	18,8	75,0	18,8	0,07	96	101	129
U3	750,0	1050,0	150,0	0,56	1817	1856	2078
U4	400,0	0,0	150,0	0,56	417	456	678
U5	75,0	0,0	37,5	0,14	79	89	144
U6	0,0	150,0	400,0	1,48	194	298	891
U7	37,5	37,5	300,0	1,11	108	186	631
U8	300,0	0,0	600,0	2,22	367	522	1411
U9	37,5	0,0	75,0	0,28	46	65	176
U10	0,0	107,1	64,3	0,24	114	131	226
U11	18,8	18,8	75,0	0,28	46	65	176
U12	0,0	75,0	300,0	1,11	108	186	631
U13	107,1	64,3	0,0	0,0	171	171	171
U14	0,0	150,0	50,0	0,19	156	169	243

Tabla 3.6: Aproximación de disposición a pagar en USD de cada atributo del análisis conjunto y disposición a pagar resultante para cada plan. Cada fila corresponde a un encuestado

Usando esta relación, la tabla 3.6 muestra los valores en dólares para cada atributo y cada encuestado, junto con en valor equivalente en dólares que dicho encuestado le asignaría a cada plan.

Si el valor es mayor al precio del plan se presume que el encuestado podría decidirse por adquirir dicho plan (y estos casos se marcan en negrita en dicha tabla).

Resultado En base a los datos del análisis conjunto, es probable que 12 de los 14 encuestados considere que el valor de uno de los planes de BeautifulLogs provee un valor que justifica desembolsar desde 75 hasta 600 dólares mensuales. 6 de los 12 encuestados podrían incluso considerar el plan más alto (Platinum).

Por lo tanto se puede considerar provisionalmente validado el supuesto H relativo a la segmentación de usuarios en planes, aunque falta verificar que la disposición a pagar de los usuarios finales se traduzca en disposición a pagar del comprador económico que la mayoría de las veces será la organización que emplea al usuario final.

3.9.2. Canvas Resultante

Se reflejaron los planes y precios determinados por el experimento.

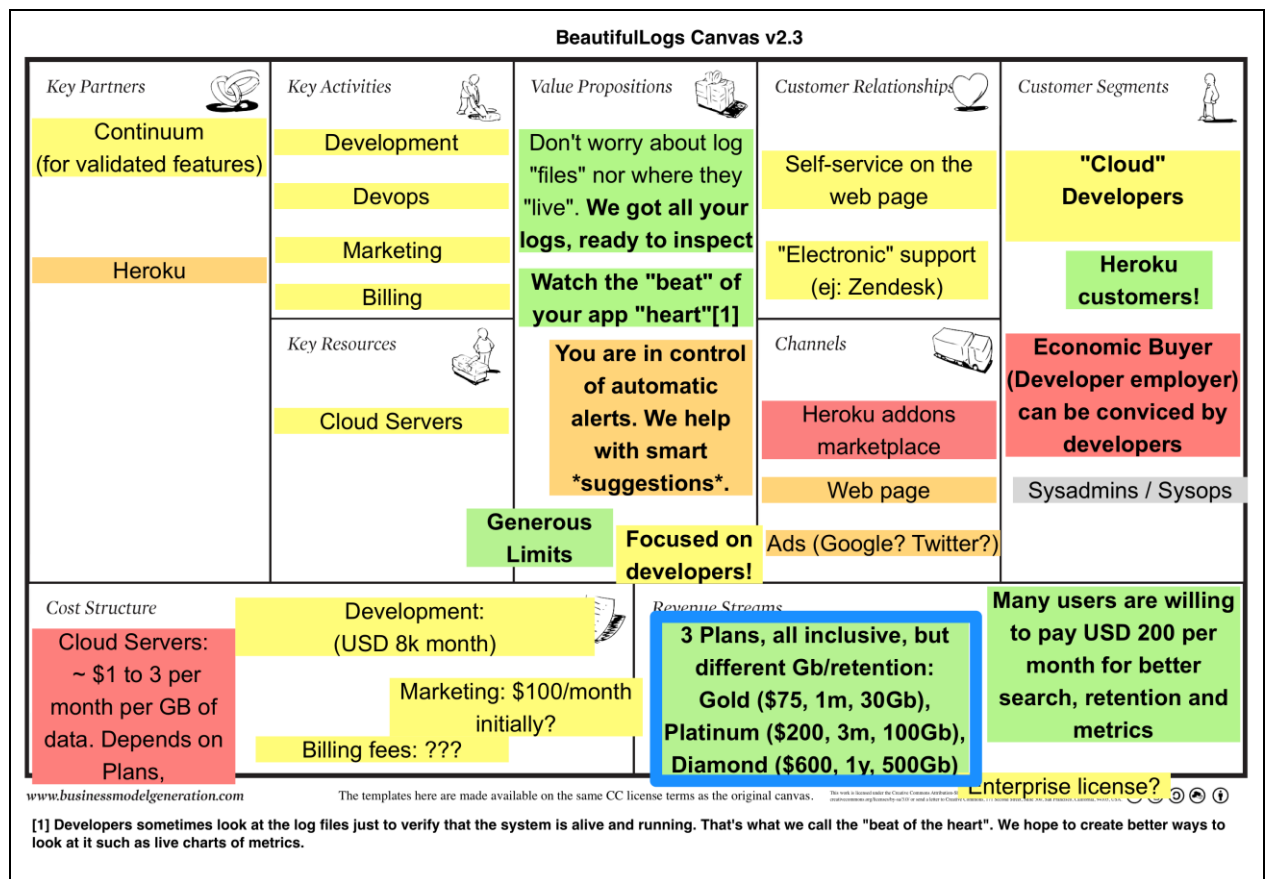


Figura 3.16: Canvas v2.3

3.10. Semana 12

La semana 12 se dedicó a definir los costos para implementar los planes definidos en el experimento 6 y determinar si el mercado de usuarios de plataformas PaaS como Heroku es suficiente para alcanzar el breakeven.

3.10.1. Experimento 7: Definición de Costos

Siguientes Supuestos a Validar

1. F. (CS, R\$): Nuestros usuarios/clientes (desarrolladores) tienen el poder de convencer al comprador económico (sus empleadores) para efectuar la compra.
2. I. (C\$, R\$, KR): Dentro del nicho de usuario de Heroku es posible llegar al breakeven en el costo de almacenamiento y procesamiento de las características extras ofrecidas por los planes definidos.
3. J. (KP, CH): Es factible para una empresa constituida en Chile ofrecer productos vía el marketplace de Heroku.

Experimento

Definición. A partir de los requerimientos impuestos por los planes Gold, Platinum y Diamond, definir una arquitectura general que permita calcular los costos aproximados de operación para cada cliente, más los costos de desarrollo de la plataforma.

Resultado. Esperado Que con una penetración del 1.5% del mercado de clientes de Heroku se obtenga el breakeven operacional para los planes definidos y los costos asociados a la plataforma.

Se presumirá que el 1% del mercado contrata el plan más económico, el 0.4% contrata el plan intermedio y el 0.1% contrata el plan más completo.

Ejecución Se realizó una prueba de concepto técnica en la que se simuló la inserción de líneas de logs en una base de datos PostgreSQL y métricas en una base de datos InfluxDB, de modo de calcular cuanto espacio físico real será necesario para almacenar 1Gb de datos crudos.

En las simulaciones del autor el resultado fué una sobrecarga de entre 15% y 20 %. Es decir, que en el peor de los casos, por cada 1Gb de datos ofrecidos, se ocuparán adicionalmente 200Mb en índices de búsqueda y métricas precalculadas.

Además, Continuum (boutique de desarrollo de software) efectuó una estimación del desarrollo inicial del producto en aproximadamente 6 meses. Considerando que un equipo de dos desarrolladores altamente capacitados puede costar un aproximado de

8000 USD al mes, se obtiene que la inversión inicial para el desarrollo del producto bordearía los 50.000 USD.

Asumiendo que con posterioridad a eso se mantiene el mismo equipo reaccionando a los cambios requeridos por el usuario, corrección de problemas, soporte y desarrollo de nuevas características, se tendrían costos fijos por 8000 USD mensual a partir del séptimo mes, los que pueden aproximarse a un presupuesto de 10000 USD mensual de costos fijos para considerar también otros gastos de oficina compartida, marketing digital, etc.

Los costos variables vienen dados por el uso de cada plan definido en el experimento anterior. La tabla 3.7 muestra los costos variables (y el margen de contribución) por cada cliente de cada plan.

Una conclusión inmediata que salta a la vista es que el almacenamiento en la nube presenta diseconomías de escala en la que el costo marginal del GB/mes aumenta conforme las necesidades de almacenamiento aumentan, lo que hace que el plan Platinum tenga un margen inferior al Gold y el plan Diamond presente pérdidas por 112 USD al mes por cliente.

Una forma del solventar este problema es reducir el espacio de retención del plan Diamond desde 500 GB/año a 400 GB/año, lo que permitiría bajar los costos de almacenamiento mensuales desde 640 USD/mes a 480 USD/mes, lo que reduce los costos totales a 552 USD/mes, permitiendo un margen de contribución de 48 USD/mes para el plan.

Esto obliga a reformular la tabla 3.6 obtenida en el experimento anterior, con la nueva disposición a pagar estimada. Esto lo refleja la tabla 3.8, donde en dos casos la disposición a pagar por el plan ha caído bajo su precio por lo que ahora 8 encuestados debieran preferir el plan Gold y 4 el nuevo plan Diamond. La nueva tabla de costos variables ajustados al nuevo plan Diamond definido en el párrafo anterior está reflejada en la tabla 3.9.

En base a los aproximadamente 30000 clientes que tendría Heroku de acuerdo a los datos de BuiltWith.com [2], se puede calcular los ingresos y costos totales mensuales con la penetración del 1.5% del mercado, lo que se refleja en la tabla 3.10.

Plan	Costo almacenamiento ⁴	Costo backups ⁵	Costo total	Precio	Margen Contribución
Gold	20	5	25	75	50
Platinum	160	15	175	200	25
Diamond	640	72	712	600	-112

Tabla 3.7: Costos variables mensuales y margen de contribución por cada plan

⁴ Calculado en base al plan más económico en DigitalOcean.com que permite almacenar el 120% del espacio ofrecido por el plan. Fuente: <https://www.digitalocean.com/pricing/>

⁵ Calculado en base al precio por GB de Amazon S3 (0,03 USD) multiplicado por el 120% del espacio ofrecido por el plan, multiplicado por 4 (para tener backups de dos días anteriores, semana anterior y mes anterior)

ID	Búsqueda ilimitada	Métricas	Incremento de 270 GBs	Incremento 1 GB	Plan Gold	Plan Platinum	Plan Diamond
U1	75,0	0,0	125,0	0,46	89	121	260
U2	18,8	75,0	18,8	0,07	96	101	122
U3	750,0	1050,0	150,0	0,56	1817	1856	2022
U4	400,0	0,0	150,0	0,56	417	456	622
U5	75,0	0,0	37,5	0,14	79	89	131
U6	0,0	150,0	400,0	1,48	194	298	743
U7	37,5	37,5	300,0	1,11	108	186	519
U8	300,0	0,0	600,0	2,22	367	522	1189
U9	37,5	0,0	75,0	0,28	46	65	149
U10	0,0	107,1	64,3	0,24	114	131	202
U11	18,8	18,8	75,0	0,28	46	65	149
U12	0,0	75,0	300,0	1,11	108	186	519
U13	107,1	64,3	0,0	0,0	171	171	171
U14	0,0	150,0	50,0	0,19	156	169	224

Tabla 3.8: Aproximación de disposición a pagar en USD de cada atributo del análisis conjunto y disposición a pagar resultante para cada plan, con plan Diamond nuevo. Cada fila corresponde a un encuestado

Plan	Costo almacenamiento	Costo backups	Costo total	Precio	Margen Contibución
Gold	20	5	25	75	50
Platinum	160	15	175	200	25
Diamond	480	72	552	600	48

Tabla 3.9: Costos variables mensuales alterando el plan Diamond para entregar márgenes positivos

.Plan	% Mercado	Nro usuarios	Ingresos totales	Costos totales	Margen Contibución
Gold	1%	300	22.500	7.500	15.000
Platinum	0,4%	120	22.000	21.000	3.000
Nuevo Diamond	0,1%	0	18.000	16.560	1.440
Total Planes	1,5%	450	64.500	45.060	19.440
- Costos Fijos					10.000
					9.440

Tabla 3.10: Ingresos y costos mensuales potenciales con 1.5% del mercado de usuarios de Heroku

Resultado. En cuanto a la obtención de un margen razonable y superior al breakeven operacional los resultados son positivos.

Sin embargo, se observa una falta de escalabilidad del negocio en cuanto a que los planes más altos contribuyen menos que el plan Gold en términos de margen, debido al mayor costo por GB/mes en almacenamiento en la nube conforme sube la demanda de espacio total requerido. Afortunadamente, la tendencia en el tiempo apunta a una disminución en el costo de almacenamiento conforme la tecnología y la capacidad productiva avanzan. Esto debiera proporcionar espacio para mayores márgenes en el futuro.

Por lo tanto se puede dar por validado el supuesto I ya que el nicho de usuarios de Heroku permitiría llegar al breakeven, lo que validaría un producto con posibilidades de extenderse al más amplio nicho PaaS (usuarios de Google App Engine, RedHat OpenShift, EngineYard, Amazon Elastic Beanstalk, Microsoft AzureWebsites) así como a los de servicios IaaS (usuarios de Amazon Web Services, Rackspace, DigitalOcean, Microsoft Azure, Google Compute Engine, etc).

3.10.2. Otros avances

Adicionalmente, el supuesto J, relativo a la posibilidad de que una empresa chilena pudiera ofrecer productos mediante el marketplace de Heroku se ha vuelto irrelevante debido a que se validó indirectamente que la apertura de una empresa en Delaware, Estados Unidos, es relativamente sencilla (Continuum, empresa de la cual el autor es socio, está abriendo empresa allí).

3.10.3. Canvas Resultante

Se modificó el plan Diamond para reflejar la nueva capacidad ofrecida, y se reflejaron los supuestos validados en la estructura de costos y en la viabilidad de servidores en la nube como recursos clave.

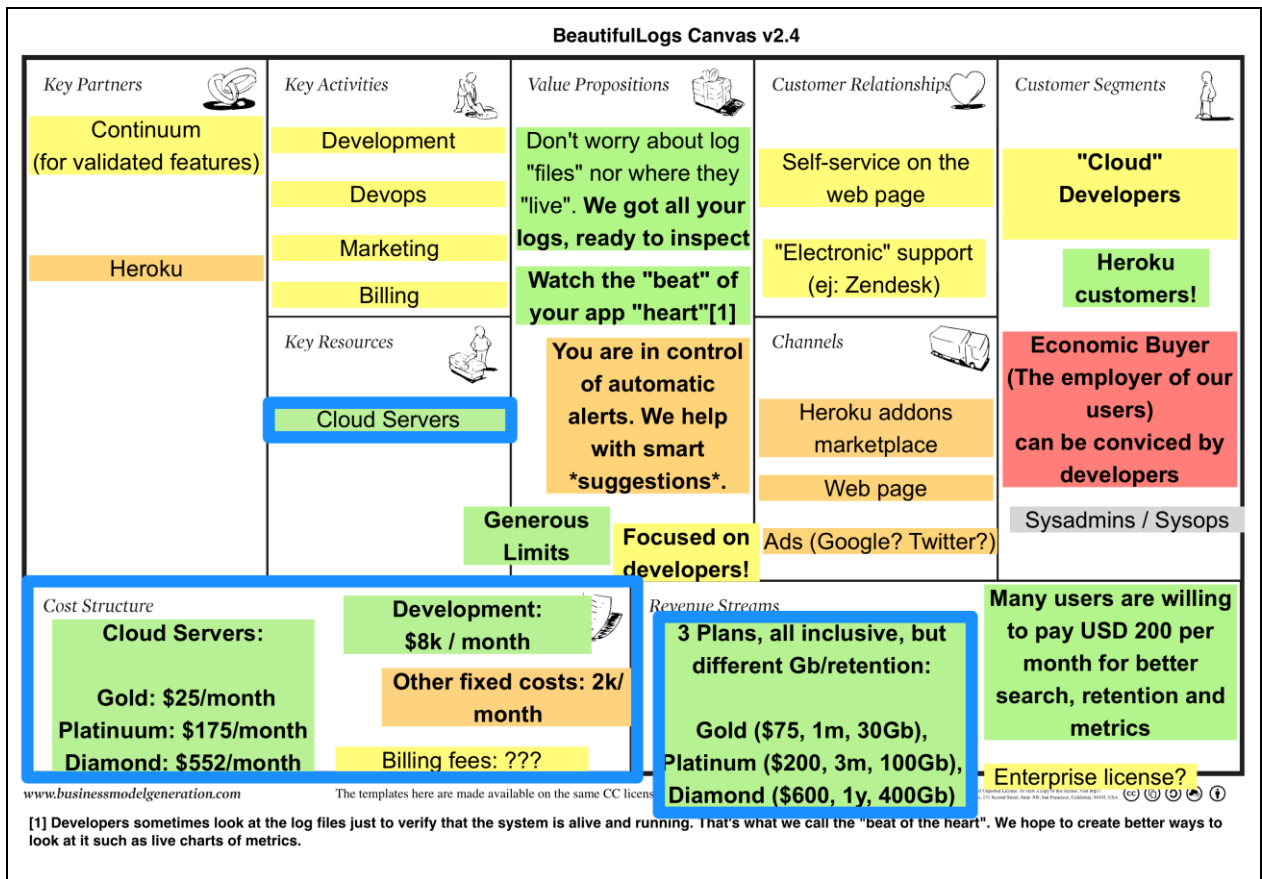


Figura 3.17: Canvas v2.4

3.11. Semanas 13 y 14

Las últimas dos semanas de desarrollo del modelo de negocio se dedicaron a validar las inferencias respecto a disponibilidad a pagar de los usuarios encuestados y la posibilidad de llegar al comprador económico (usualmente el empleador del usuario final).

3.11.1. Experimento 8: Validar disposición a pagar y a recomendar producto

Siguientes Supuestos a Validar

1. F. (CS, R\$): Nuestros usuarios/clientes (desarrolladores) tienen el poder de convencer al comprador económico (sus empleadores) para efectuar la compra.
2. K. (CS, R\$): Nuestros usuarios/clientes (desarrolladores) están dispuestos a convencer al comprador económico (sus empleadores).

Experimento

Definición. Verificar la validez de la inferencia hecha en la tabla 3.8 relativa a los planes que cada usuario encuestado debiera estar dispuesto a pagar, consultando a los usuarios que plan de los ofrecidos su organización debiera suscribir.

Resultado. Esperado Que al menos 4 de los 12 usuarios que estimamos estarían dispuestos a pagar un plan confirmen que considerarían uno de nuestros planes para usarlo en un proyecto/organización en el que participen. Y que de esos 4, al menos 1 se comprometa al nivel de ofrecer una reunión con un tomador de decisiones en su organización (de modo que el compromiso vaya más allá de una respuesta e involucre una acción real donde el usuario se juegue su reputación dentro de la organización apoyando el producto).

La tabla 3.11 refleja las inferencias (a partir del análisis conjunto y de la hipótesis de que el valor asignado a cada atributo sería cercano a una función lineal) respecto a los planes que cada uno de nuestros 14 encuestados debiera estar dispuesto a pagar.

ID	PLAN
U1	Gold
U2	Gold
U3	Diamond
U4	Diamond
U5	Gold
U6	Diamond
U7	Gold
U8	Diamond
U9	--
U10	Gold
U11	--
U12	Gold
U13	Gold
U14	Gold

Tabla 3.11: Planes que los encuestados debieran estar dispuestos a pagar

Ejecución. Se encuestó mediante email personalizado, chat y/o llamada telefónica a cada uno de los 14 usuarios, exponiendo los planes y su precio y solicitando indicar dos cosas:

- Sin considerar el precio, ¿Cuál de estos planes considerarías adecuado para la organización/proyecto en el que participas actualmente?
- Considerando el precio, ¿Cuál de estos planes recomendarías suscribir a la organización/proyecto en el que participas actualmente?

ID	Plan Inferido	Plan “adecuado”	Plan a suscribir	Comentarios seleccionados
U1	Gold	Gold	Gold	“Tengo dudas respecto a si pagarían ese precio, pero creo que lo vale”
U2	Gold	Diamond	Gold	“Creo que habría que comprar el plan más barato para evaluar bien la herramienta primero”
U3	Diamond	Gold	Gold	“No creo que necesitemos más que 1 mes de datos”
U4	Diamond	Gold	--	“Veo difícil pagar tan caro”
U5	Gold	--	--	“Creo que el target del producto es otro, no nosotros”
U6	Diamond	Platinum	Platinum	“Si funciona bien, lo pago” (Usuario es co-fundador de su propia startup)
U7	Gold	Gold	--	“Una vez que funcione a full y tengas casos de éxito podría recomendar algo”
U8	Diamond	Gold	--	“Creo que está un poco caro relativo a la competencia”
U9	--	Gold	--	“No veo algo que justifique el aumento de precio respecto a las otras soluciones”
U10	Gold	Gold	--	“Nuestros costos de infraestructura de tecnología son bajos, costaría justificar gastar tanto en logs” (Usuario es cofundador de su propia startup)
U11	---	Gold	Gold	“Estoy en varios proyectos, sólo en algunos de ellos prodría justificarse”
U12	Gold	Gold	--	“No me convence el precio”
U13	Gold	Gold	--	“Podríamos considerarlo una vez el producto sea sólido y ya sea usado por otros”
U14	Gold	Gold	--	“Lo encuentro caro”

Tabla 3.12: Respuestas de usuarios a planes para sus empresas/organizaciones

Las respuestas a estas preguntas se resumen en la tabla 3.12.

En base a esto se organizaron reuniones personales entre el autor y cada uno de los 5 usuarios que consideraban que su organización debiera suscribir uno de los planes con los precios indicados.

Allí, además de indicarles a dichos usuarios que el producto seguiría en fase beta por al menos 4 meses más y despejar cualquier duda respecto a las funcionalidades del producto, se le consultó a cada usuario si podía organizar una reunión con alguien con poder de tomar la decisión de compra dentro de su organización. Las alternativas ofrecidas para la respuesta fueron:

- “Me comprometo a organizar una reunión dentro de los próximos 2 meses, usando la beta (mejorada respecto a la situación actual) como demo”.
- “Me comprometo a organizar una reunión una vez el producto salga de la fase beta (aproximadamente en 4-5 meses)”
- “Por ahora no me puedo comprometer a organizar una reunión.”

El resultado es que los 5 usuarios se mostraron dispuestos a coordinar la reunión y apoyar la compra del producto. Como lo muestra la tabla 3.13, 3 de estos usuarios están dispuestos a hacer esto en la fase beta del producto (asumiendo que no pagarían el precio final hasta que el producto salga de la fase beta), y 2 usuarios lo harían una vez el producto salga de la fase beta.

ID	Se compromete a reunión una vez el producto esté en fase...
U1	Beta
U2	Beta
U3	Beta
U6	Final
U11	Final

Tabla 3.13: Compromiso de usuarios para coordinar una reunión con el tomador de decisiones para suscripción del producto

Resultado. Con 5 de los 14 usuarios ya dispuestos a recomendar el producto a su organización, se puede establecer el experimento como exitoso, superando con creces las expectativas establecidas en la sección de resultados esperados. El autor evaluó llevar el experimento más allá y coordinar las 3 reuniones para los usuarios dispuestos a promover el producto en sus organizaciones en la fase beta, decidiendo no hacerlo aún, pues se corre el riesgo de dañar la reputación de estos 3 usuarios en sus organizaciones en caso que el producto no sea desarrollado (sea por falta de inversión o por existir alternativas más atractivas de emprendimiento para el autor).

En consecuencia, se considera razonablemente validado el supuesto K, en cuanto los usuarios se comprometen a apoyar la adopción del producto dentro de sus organizaciones, dentro de las cuales la mayoría influye fuertemente en las decisiones de herramientas de desarrollo. Sin embargo, el supuesto F queda pendiente, pues sólo se puede validar consiguiendo una compra real, para lo que se requiere presentar una versión mejorada del producto.

3.11.2. Canvas Resultante

En lugar de dar como resuelto el último supuesto marcado en rojo, se introdujo un nuevo supuesto validado: Que los desarrolladores que valoran el producto se muestran dispuestos a ayudar a convencer a los compradores económicos, lo que es un supuesto más débil pero corresponde fielmente a lo realmente demostrado por el experimento.

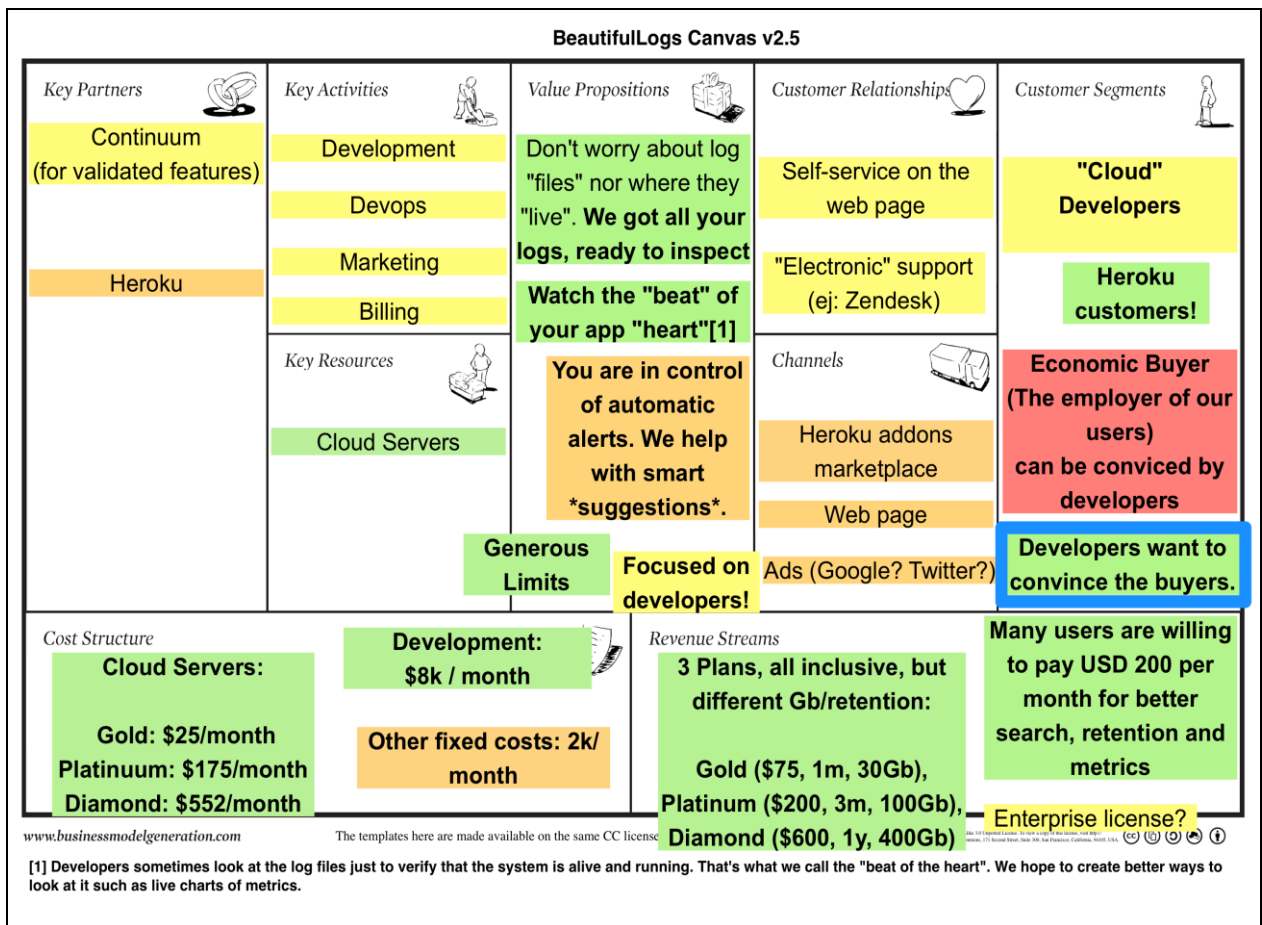


Figura 3.18: Canvas v2.5

4. MODELO DE NEGOCIO RESULTANTE

4.1. Resumen Ejecutivo

4.1.1. Concepto

Los desarrolladores de aplicaciones, históricamente, han tenido dos grandes maneras de encontrar errores en el software que programan. Una de ellas es depurar interactivamente un programa, lo que en términos simples es equivalente a ejecutar el programa defectuoso en cámara lenta y observar detenidamente todo lo que va ocurriendo hasta encontrar qué se comporta de manera extraña. Este proceso depende de contar con una serie de pasos conocida para causar el problema, probablemente reportada por un usuario o por un hábil miembro de un equipo de control de calidad.

Lamentablemente no siempre se cuenta con esa serie de pasos. Sin ella, es impráctico hacer la revisión en cámara lenta, tal como sería impráctico ver toda una película en cámara lenta para encontrar un pequeño detalle de continuidad mal logrado.

En esos casos los desarrolladores enfrentan los errores como misterios que deben ser investigados con todas las pistas posibles. Y una de las mayores fuentes de pistas son los logs, verdaderas cajas negras del software que registran (según lo indicado por el programador original) los pasos más importantes que lleva a cabo un programa. Estos archivos solían almacenarse en el mismo computador que el programa (o en un sistema de archivos cercano).

La aparición de la nube (cloud computing) como alternativa de infraestructura TI trajo múltiples beneficios, especialmente en costos y flexibilidad. Una aplicación que solía vivir en un datacenter monolítico, hoy está compuesta por distintos subsistemas, probablemente distribuidos en “la nube”. El problema es: ¿Donde se guardan las cajas negras (los logs) de cada uno de esos subsistemas? La respuesta varía desde “en muchas partes” hasta “en ningún lado”. Y por eso se hace necesario agregar a la nube otro componente que permita a los desarrolladores consolidar nuevamente los logs en un lugar único a consultar.

Sin embargo, el desafío va más allá de eso. El volumen de acceso, uso, y generación de datos de las aplicaciones en la nube no es menor. La posibilidad de que cualquier aplicación pueda escalar automáticamente a cientos de máquinas virtuales que le permitan soportar decenas de miles de usuarios simultáneos también provoca que el tamaño de estas cajas negras aumente considerablemente.

Encontrar una pista para resolver un error en medio de estos logs enormes se transforma en buscar una aguja en un pajar.

BeautifulLogs soluciona, entre otros, ese problema.

4.1.2. Oportunidad y Estrategia

BeautifulLogs es un servicio SaaS para el nicho de desarrolladores de aplicaciones alojadas en la nube, especialmente en plataformas PaaS como Heroku. BeautifulLogs participa del mercado de la gestión de logs de dichas aplicaciones, posicionándose como un servicio premium para atender a clientes con necesidades avanzadas que no son cubiertas a cabalidad por las soluciones existentes.

Dichas necesidades son principalmente la gestión de grandes volúmenes de datos sin las restricciones tradicionales que obligan a los desarrolladores a “archivar” la información por su propia cuenta y sólo tener disponible para operar la información de los últimos días o semanas.

Esta tradición responde más bien a limitaciones históricas que con las mayores capacidades de almacenamiento y procesamiento de los computadores actuales se han vuelto más manejables.

Otro aspecto clave de BeautifulLogs es su enfoque en repensar la interfaz de usuario para hacerla más simple y fácil de usar, a pesar de contar con mayores volúmenes de datos que procesar y mostrar. Más de la mitad de los usuarios que han visto prototipos de la interfaz de usuario se ha declarado satisfecho con el enfoque. El concepto es mezclar la potencia de manejar y buscar dentro de grandes volúmenes de datos junto con una manera de darle sentido a todos esos datos rutinarios que contienen los logs, como el uso de ciertos recursos, tiempos de respuesta de peticiones, etc; entregando visualizaciones gráficas que representan en una sola imagen el contenido de cientos o miles de líneas de logs.

4.1.3. Ventajas Competitivas

La principal ventaja competitiva de BeautifulLogs es el foco en manejar grandes volúmenes de datos sin comprometer la experiencia de usuario. Si bien esto encasilla al producto en un nicho dentro de un nicho, le permite tomar decisiones que son impensables en productos de la competencia que deben servir a una multitud de perfiles de usuarios (ej: operaciones, administradores de sistemas, desarrolladores, QA). BeautifulLogs se enfoca exclusivamente en desarrolladores y devops, y de éstos, solo en quienes cuentan con aplicaciones importantes en la nube.

4.1.4. Mercado y Proyecciones

El mercado PaaS contempla proveedores como Google (con su producto App Engine), Amazon (Beanstalk) y muchos otros. Sin embargo, destaca entre ellos Heroku, propiedad de Salesforce.com y uno de los más maduros en cuanto a extensibilidad mediante productos ofrecidos por terceros. Heroku cuenta con un marketplace que en el año 2012 contaba con 4 millones de descargas de addons ofrecidos por terceras partes. BeautifulLogs se posiciona como uno de estos addons.

Se estima que alrededor de 5 millones de aplicaciones se encuentran actualmente en la plataforma de Heroku. Esto significa que con alcanzar al 1% de ese mercado se cuenta con 50.000 clientes potenciales del servicio, sin considerar a posibles clientes de otros proveedores PaaS o incluso de servicios IaaS como Amazon Web Services.

Con un precio orientado al segmento premium y alcanzando a tan solo 5.500 clientes (alrededor de 0,1% de las aplicaciones instaladas en Heroku, sin considerar el crecimiento del mercado PaaS de un 24% durante el 2014 de acuerdo a RedHat [11]), BeautifulLogs llegaría a ingresos de más de 9,5 millones de dólares anuales y un flujo libre de caja superior a los 1,5 millones de dólares.

Una inversión de 350.000 dólares en el trimestre inicial con una participación del 15% podría así rentar más de un 75% anual y retornar 17 veces lo invertido (en términos nominales a 5 años).

4.1.5. Oferta

Una inversión de 350.000 USD permitiría financiar 2 años de operación con un burn rate que oscila alrededor de los 50000 USD por trimestre durante los primeros 6 trimestres. Con una participación en el rango del 10% al 30 %, los retornos potenciales superan el 1000% en un plazo de 5 años.

4.2. Modelo de negocio

La figura 4.1 representa el canvas del modelo de negocio propuesto para BeautifulLogs. En términos generales, se trata de un producto SaaS para gestión de logs para aplicaciones en la nube.

A continuación se detallan las secciones del modelo.

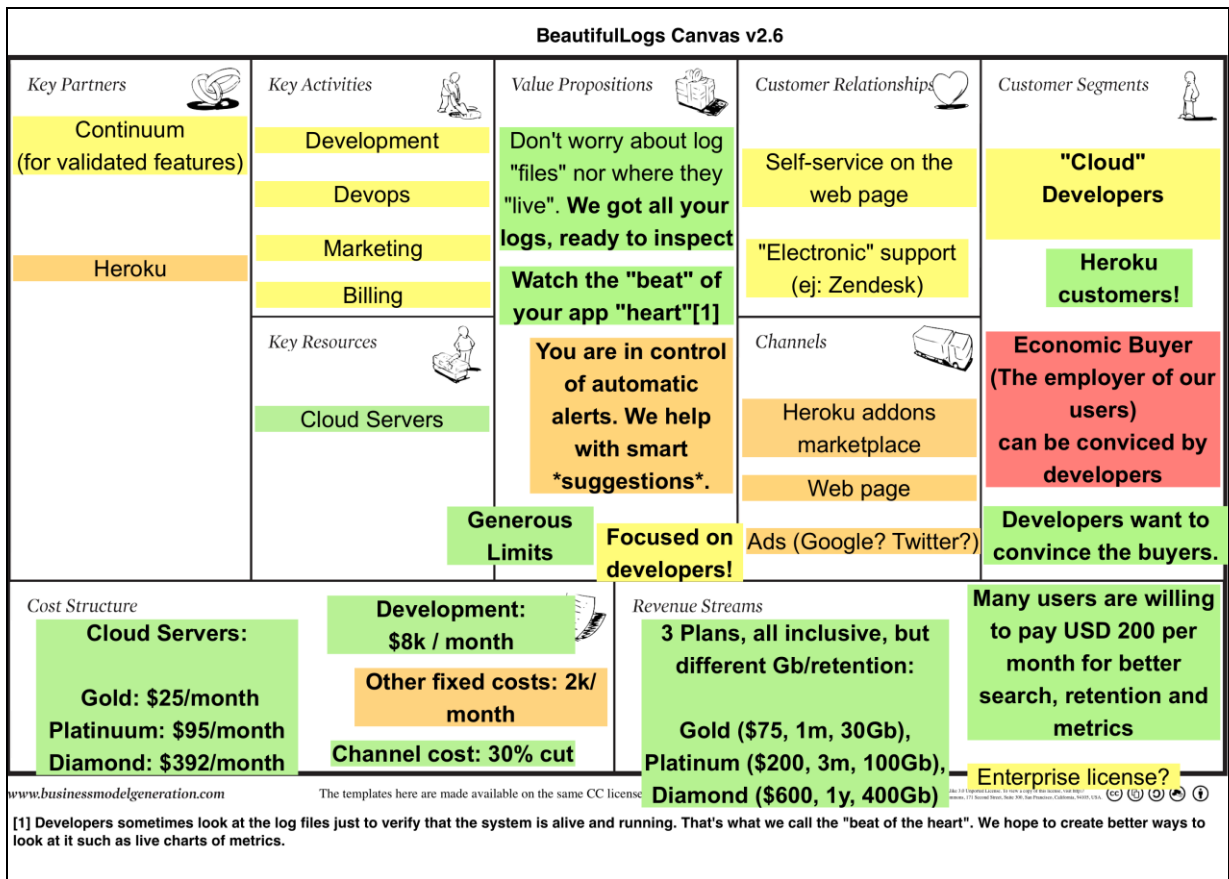


Figura 4.1: Canvas Final

4.2.1. Customer Segments

BeautifulLogs está apuntado a desarrolladores de aplicaciones en la nube, especialmente a los de un perfil que usa un producto PaaS como Heroku.

Esto debido a que los usuarios de Heroku están acostumbrados a pagar por addons que le ofrecen capacidades o funcionalidades superiores a las de heroku por un precio mensual adicional. Esto los hace receptivos a soluciones que les evitan tedio y esfuerzo en gestión de infraestructura a cambio de un precio fijo mensual en dólares.

Cabe destacar que normalmente no son los desarrolladores mismos quienes pagan por estos servicios, sino que la organización a la que pertenecen, por lo que el cliente final no es el comprador económico del producto.

El tamaño del mercado en este segmento es muchísimo más grande que lo originalmente proyectado. Basado en datos de BuiltWith.com, se consideró un universo de clientes de Heroku del orden de las decenas de miles. Sin embargo, información de prensa [10] indica que ya en 2013 existían más de 4 millones de aplicaciones en Heroku. Una estimación de crecimiento del 25% anual ubicaría el número de aplicaciones actuales en 5 millones, con lo que el objetivo de 1,5% de ese nicho de

mercado (basado en la conversión obtenida con el prototipo usado para los experimentos) asciende a más de 70.000 potenciales suscriptores para BeautifulLogs (y un objetivo más conservador de 0.75% alcanza más de 35.000 suscriptores).

4.2.2. Value Proposition

La propuesta de valor de BeautifulLogs se basa en simplificar la gestión de logs mucho más allá de lo que ofrecen otras soluciones actuales. En particular, los usuarios no tienen que preocuparse del origen de los logs ni de los sistemas que los generan. BeautifulLogs se enfoca más en encontrar la información que en obligar al usuario a organizarla.

Dado el volumen de información que se maneja en los logs de sistemas bajo alta carga, muchas soluciones proponen la organización de los logs como una forma de hacer que el flujo de información sea manejable. Debido a que BeautifulLogs no obligará al usuario a organizar nada, se ofrecen otras potentes herramientas para conseguir los objetivos que persiguen los usuarios:

- Búsquedas y filtros extensivos y sin límites para encontrar la información que se necesita.
- Métricas en tiempo real (e históricas) para analizar series de tiempo que se pueden extraer desde los logs.
- La posibilidad de sugerir alertas de manera automatizada (o asistida) ante comportamientos inusuales de las aplicaciones, pues muchos de estos comportamientos (ej: exceso de carga, aumento de la frecuencia de errores) se reflejan en los logs.

4.2.3. Customer Relationships

Dado que BeautifulLogs está pensado como un producto SaaS de auto-atención, la relación con los clientes se realizará mediante la propia página web del servicio y mediante sistemas de manejo de consultas, preguntas y/o feedback como Zendesk.

4.2.4. Channels

Los canales para difundir el producto serán tanto de naturaleza publicitaria (ej: Google Adwords) como también la presencia en marketplaces de los propios proveedores PaaS (ej: Heroku marketplace). La facturación y entrega del servicio correrá por cuenta de estos mismos marketplaces, lo que tiene un costo del 30% del precio.

4.2.5. Revenue Streams

Los ingresos provienen exclusivamente de la suscripción mensual de los usuarios al servicio ofrecido. Se ofrecen tres planes (Gold, Platinum, Diamond) con precios de USD 75, USD 200 y USD 600 mensual. Los planes más costosos entregan mayor valor mediante mayor capacidad y retención de la información almacenada.

Existe la posibilidad de explorar licencias de tipo enterprise para organizaciones con volúmenes de datos que superen lo ofrecido por los planes anteriores, o con necesidades diferentes en cuanto a seguridad y/o localización de la información.

4.2.6. Key Partners

Se identifican dos partners claves para la ejecución del modelo. Uno es la participación de plataformas PaaS (como Heroku) como canal de distribución del producto. El otro es Continuum, empresa de desarrollo de la que el autor es socio, como apoyo en el desarrollo del software y experticia técnica en los componentes más complejos a desarrollar.

4.2.7. Key Activities

Las actividades claves para la ejecución del negocio están encabezadas por el desarrollo y operación del software que garantice la escalabilidad del mismo.

Asimismo, marketing es clave para llegar con el producto al nicho de usuarios que valora las características del producto y para comunicar estas características de manera apropiada.

La facturación es la otra actividad clave identificada, debido al carácter de suscripción mensual en que se ofrecerá el producto a los usuarios finales.

4.2.8. Key Resources

Además del equipo, el recurso clave indiscutido serán los servidores en la nube desde donde operará el software de BeautifulLogs. El proveedor seleccionado es DigitalOcean debido a la excelente relación de precio con calidad (y discos duros rápidos de tipo SSD), pero en principio el software será posible desplegarlo en cualquier servidor apto en la nube.

4.2.9. Cost Structure

La estructura de costos en su parte fija está dominada por el equipo de desarrollo, que se estima en USD 8.000 al mes. Se presupuestan costos por USD 2.000 al mes para otros costos fijos como arriendo de oficina, presupuesto de marketing digital, y costos administrativos.

Se estima necesaria una inversión inicial equivalente a 6 meses de costos fijos para el desarrollo inicial del producto (USD 60.000).

El costo variable está dado directamente por el arriendo de servidores en la nube. Debido a la generosa oferta de espacio de BeautifulLogs se estima que deberá dejarse un servidor exclusivo para cada cliente. Además, existen costos por respaldos de información en la nube. La suma de todo esto da USD 25 en costos para un cliente Gold, USD 175 para un cliente Platinum y USD 552 para un cliente Diamond usando los servicios de DigitalOcean.com. Lamentablemente en este caso existen deseconomías de escala (el costo marginal de 1Gb de espacio aumenta conforme se solicita una capacidad contigua mayor), pero se prevee una posible baja en el costo en el tiempo.

De hecho, con posterioridad a la ejecución de los experimentos del capítulo anterior, Linode.com ofreció nuevos precios de sus servidores en la nube con precios menores a digital ocean, que ubican los costos variables en USD 25, USD 95 y USD 392 para los planes Gold, Platinum y Diamond respectivamente.

4.3. Métricas y Flujos Proyectados

Los flujos proyectados asumen un período de 2 trimestres (6 meses, llamados Q1 y Q2 en los flujos) en que se desarrolla el producto con calidad final y se pone en operación en modalidad de beta testing, en el cual los usuarios no pagan por el uso. Con posterioridad a eso se consideran 2 trimestres (Q3 y Q4) de crecimiento agresivo y luego otros 2 trimestres (Q5 y Q6) fuertes en crecimiento al final de los cuales se saturan los early adopters y el ratio de crecimiento empieza a decrecer (Q7 y Q8) hasta llegar a uno normal de la industria a partir del tercer año (Y3).

La figura 4.2 muestra las métricas claves del modelo de negocio a partir de la cual se proyectan los flujos potenciales.

		Pessimistic	Moderate	Optimistic
Price / month	Gold	\$75,00	\$75,00	\$75,00
	Diamond	\$200,00	\$200,00	\$200,00
	Platinum	\$600,00	\$600,00	\$600,00
Conversion	Gold	0,50%	1,00%	2%
	Diamond	0,20%	0,40%	0,80%
	Platinum	0,05%	0,10%	0,20%
Cost Per Click (or lead)		\$4,00	\$4,00	\$4,00
Customer Acquisition Cost		\$533,33	\$266,67	\$133,33
Average Monthly Revenue per Subscriber		\$143,33	\$143,33	\$143,33
Initial Subscribers on Q3, Q4		100	200	400
Quarterly Growth Rate (Including Churn) Q5-Q6		35,00%	70,00%	100,00%
Quarterly Growth Rate (Including Churn) Q7-Q8		20,00%	40,00%	80,00%
Annualized Growth Rate (Including Churn) Y3-Onwards		20,00%	35,00%	50,00%
Variable costs / month on Q3	Gold	\$25,00	\$25,00	\$25,00
	Diamond	\$95,00	\$95,00	\$95,00
	Platinum	\$392,00	\$392,00	\$392,00
Average Variable Cost per Subscriber		\$68,13	\$68,13	\$68,13

Figura 4.2: Métricas claves del modelo de negocio. Valores en USD.

Los datos de precios y costos salen directamente de la información recolectada anteriormente en este trabajo.

La información de conversión (porcentaje de usuarios que deciden suscribirse al producto) se basa en los resultados del capítulo anterior, en el que se estima que entre 1 y 3 usuarios conseguirán la aprobación para adquirir el producto, lo que si se considera el universo original de 50 usuarios (de entre quienes quedaron 15 usuarios como beta testers) representa entre el 2% y el 6%. Es probable que la muestra de usuarios esté sesgada hacia sujetos más proclives a contratar el servicio, por lo que las proyecciones usan una conversión de entre un 0,75% (caso pesimista) y un 3% (caso optimista).

El costo de adquisición se calcula como el costo de conseguir un lead (oportunidad de exponer a un usuario al material de marketing del producto) dividido por el porcentaje de conversión. El costo de un lead puede aproximarse en base a la información provista por Google en su planificador de AdWords, que fue usado por el autor para obtener un precio aproximado por click para búsquedas relacionadas con el producto (ej: “log

management”, “log viewer”). Por razones competitivas, es probable que leads de otras fuentes tengan un precio similar.

Finalmente, los suscriptores iniciales y los parámetros de la curva de crecimiento son basados en información de la industria SaaS y de emprendimientos en general. De acuerdo a [1], una aplicación web o mobile debe ser capaz de generar interés en al menos 100 usuarios para ser viable (en caso contrario debe detenerse el proyecto). Por tanto, el rango de usuarios suscritos durante los primeros 3 meses de operación va desde esos 100 usuarios mínimos hasta 400 usuarios en el caso más optimista. Los ratios de crecimiento se basan en promedios de la industria de acuerdo a la encuesta de Pacific Crest [8] (banco de inversión con foco en empresas SaaS) y al estudio de la industria SaaS de River Cities Capital Fund [4] y son consistentes con los ratios de crecimiento de emprendimientos que re-segmentan mercados existentes [1].

En función de estas métricas se calcula el flujo de caja simplificado proyectado para los tres escenarios previstos. El flujo de caja del escenario moderado puede observarse en la figura 4.3.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Y3	Y4	Y5
Net New Subscribers (Including Churn)	0	0	100	200	280	476	462	647	793	1070	1445
Total Subscribers	0	0	200	400	680	1.156	1.618	2.265	3.058	4.128	5.573
Beta Testers	50	100									
Subscription Revenue	\$0,00	\$0,00	\$86.000,00	\$172.000,00	\$292.400,00	\$497.080,00	\$695.740,00	\$973.950,00	\$5.259.760,00	\$7.100.160,00	\$9.585.560,00
Channel Costs (30% fee)	\$0,00	\$0,00	-\$25.800,00	-\$51.600,00	-\$87.720,00	-\$149.124,00	-\$208.722,00	-\$292.185,00	-\$1.577.928,00	-\$2.130.048,00	-\$2.875.668,00
Variable Operation Costs	-\$10.220,00	-\$20.440,00	-\$40.880,00	-\$81.760,00	-\$138.992,00	-\$236.286,40	-\$330.719,20	-\$462.966,00	-\$2.500.220,80	-\$3.375.052,80	-\$4.556.484,80
Acquisition Costs	-\$13.333,33	-\$13.333,33	-\$26.666,67	-\$53.333,33	-\$74.666,67	-\$126.933,33	-\$123.200,00	-\$172.533,33	-\$211.466,67	-\$285.333,33	-\$385.333,33
Fixed Ops/Dev/Adm. Costs	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$45.000,00	-\$45.000,00	-\$45.000,00	-\$45.000,00	-\$240.000,00	-\$240.000,00	-\$240.000,00
Investment round	\$350.000,00										
Net Cash Flow	\$296.496,67	-\$63.673,33	-\$37.046,67	-\$44.093,33	-\$53.018,67	-\$58.631,73	-\$9.821,20	\$4.177,67	\$733.995,53	\$1.074.923,87	\$1.535.091,87
Cash Balance	\$296.496,67	\$232.823,33	\$195.776,67	\$151.683,33	\$98.664,67	\$40.032,93	\$30.211,73	\$34.389,40	\$768.384,93	\$1.843.308,80	\$3.378.400,67

Figura 4.3: Flujo de caja, escenario moderado. Valores en USD.

Como se señaló anteriormente, el modelo presume que durante los 6 primeros meses se lleva a cabo el desarrollo de la aplicación en conjunto con un beta testing gratuito. Se presumen costos de 10.000 USD mensuales durante esta etapa en equipo de desarrollo (2 a 3 personas) y gastos administrativos.

Asimismo, los 50 usuarios beta a conseguir en esta fase tienen un costo de adquisición significativo, lo que se suma a los costos variables de pagar por la infraestructura para ejecutar la versión beta de la aplicación para dichos usuarios.

Dado el crecimiento supuesto en el escenario moderado, el punto de equilibrio se alcanza en el octavo trimestre (Q8) durante el cual se produce un flujo de caja positivo de un poco más de 4.000 USD. En este punto, la aplicación contaría con más de 2200 suscriptores e ingresos cercanos al millón de dólares por trimestre.

Un costo sustancial incorporado a partir del tercer trimestre (Q3) es el porcentaje de las ventas que se lleva Heroku o el canal usado para el cobro de la suscripción. El valor de 30% es el usado por Heroku y por otros proveedores PaaS y es un porcentaje usado en la industria del software en otros modelos similares, como la App Store de Apple.

Para sustentar los dos primeros años de flujo de caja negativo, se requiere una inversión inicial de 350.000 USD, a obtener mediante inversionistas ángeles y/o concursos de emprendimiento.

A partir del año 3 se generarían flujos de caja positivos que llegan en el año 5 a algo más de 1,5 millones de dólares. Con ingresos anuales de 9,5 millones de dólares y factores de valoración sobre ingresos de alrededor de 4 mostrados por [4] se podría llegar a una valoración de la empresa en torno a los 40 millones de dólares, con lo que una participación del 15% entregada a cambio de los 350 mil dólares de inversión podría rentar 17 veces, con una TIR del 76,5 %.

Flujos de caja de escenarios alternativos se muestran en las figuras 4.4 y 4.5 para el caso pesimista y optimista respectivamente.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Y3	Y4	Y5
Net New Subscribers (Including Churn)	0	0	50	100	70	95	73	88	105	126	151
Total Subscribers	0	0	100	200	270	365	438	526	631	757	908
Beta Testers	25	50									
Subscription Revenue	\$0,00	\$0,00	\$43.000,00	\$86.000,00	\$116.100,00	\$156.950,00	\$188.340,00	\$226.180,00	\$1.085.320,00	\$1.302.040,00	\$1.561.760,00
Channel Costs (30% fee)	\$0,00	\$0,00	-\$12.900,00	-\$25.800,00	-\$34.830,00	-\$47.085,00	-\$56.502,00	-\$67.854,00	-\$325.596,00	-\$390.612,00	-\$468.528,00
Variable Operation Costs	-\$5.110,00	-\$10.220,00	-\$20.440,00	-\$40.880,00	-\$55.188,00	-\$74.606,00	-\$89.527,20	-\$107.514,40	-\$515.905,60	-\$618.923,20	-\$742.380,80
Acquisition Costs	-\$13.333,33	-\$13.333,33	-\$26.666,67	-\$53.333,33	-\$37.333,33	-\$50.666,67	-\$38.933,33	-\$46.933,33	-\$56.000,00	-\$67.200,00	-\$80.533,33
Fixed Ops/Dev/Adm. Costs	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$120.000,00	-\$120.000,00	-\$120.000,00
Investment round	\$350.000,00										
Net Cash Flow	\$301.581,67	-\$53.503,33	-\$46.856,67	-\$63.713,33	-\$40.911,33	-\$44.947,67	-\$26.111,53	-\$25.507,73	\$68.554,40	\$106.187,80	\$151.376,87
Cash Balance	\$301.581,67	\$248.078,33	\$201.221,67	\$137.508,33	\$96.597,00	\$51.649,33	\$25.537,80	\$30,07	\$68.584,47	\$174.772,27	\$326.149,13

Figura 4.4: Flujo de caja, escenario pesimista. Valores en USD.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Y3	Y4	Y5
Net New Subscribers (Including Churn)	0	0	200	400	800	1600	2560	4608	5184	7776	11664
Total Subscribers	0	0	400	800	1.600	3.200	5.760	10.368	15.552	23.328	34.992
Beta Testers	100	200									
Subscription Revenue	\$0,00	\$0,00	\$172.000,00	\$344.000,00	\$688.000,00	\$1.376.000,00	\$2.476.800,00	\$4.458.240,00	\$26.749.440,00	\$40.124.160,00	\$60.186.240,00
Channel Costs (30% fee)	\$0,00	\$0,00	-\$51.600,00	-\$103.200,00	-\$206.400,00	-\$412.800,00	-\$743.040,00	-\$1.337.472,00	-\$8.024.832,00	-\$12.037.248,00	-\$18.055.872,00
Variable Operation Costs	-\$20.440,00	-\$40.880,00	-\$81.760,00	-\$163.520,00	-\$327.040,00	-\$654.080,00	-\$1.177.344,00	-\$2.119.219,20	-\$12.715.315,20	-\$19.072.972,80	-\$28.609.459,20
Acquisition Costs	-\$13.333,33	-\$13.333,33	-\$26.666,67	-\$53.333,33	-\$106.666,67	-\$213.333,33	-\$341.333,33	-\$614.400,00	-\$691.200,00	-\$1.036.800,00	-\$1.555.200,00
Fixed Ops/Dev/Adm. Costs	-\$30.000,00	-\$30.000,00	-\$30.000,00	-\$90.000,00	-\$90.000,00	-\$90.000,00	-\$90.000,00	-\$90.000,00	-\$480.000,00	-\$480.000,00	-\$480.000,00
Investment round	\$350.000,00										
Net Cash Flow	\$286.326,67	-\$84.013,33	-\$17.426,67	-\$64.853,33	-\$39.706,67	\$10.586,67	\$133.402,67	\$312.124,80	\$4.858.828,80	\$7.528.243,20	\$11.532.364,80
Cash Balance	\$286.326,67	\$202.313,33	\$184.886,67	\$120.033,33	\$80.326,67	\$90.913,33	\$224.316,00	\$536.440,80	\$5.395.269,60	\$12.923.512,80	\$24.455.877,60

Figura 4.5: Flujo de caja, escenario optimista. Valores en USD.

El escenario pesimista muestra el caso en el que la empresa sigue siendo viable pero sin obtener retornos atractivos. El punto de equilibrio se alcanza en el tercer año, y de darse este caso es muy probable que se requiera alguna inversión o puente adicional debido a que el balance de caja es prácticamente nulo al final del segundo año.

Sin embargo, de cumplirse el crecimiento lento pero sostenido en los años siguientes se obtiene un flujo de caja positivo de más de 150 mil dólares al final del quinto año, lo que podría permitir retornar al menos parte de la inversión original.

Por otro lado, el escenario optimista presenta el caso ideal en el que se llega a flujos de caja positivos por sobre los 10 millones de dólares en el quinto año. También se incluyen gastos fijos más altos que en los otros escenarios para considerar que la operación se escala a una mayor velocidad (ya en el cuarto trimestre se triplica el gasto en el equipo de desarrollo y operación para permitir aprovechar el mayor potencial, si es que efectivamente se consiguen 400 suscriptores al llegar al tercer trimestre). Aún así el punto de equilibrio se alcanza tempranamente, en el sexto trimestre.

4.4. Oportunidades de Internacionalización

El negocio de BeautifulLogs es inherentemente global, pues es un SaaS cuyo canal de venta permite pagar la suscripción a cualquier cliente mediante tarjeta de crédito internacional. Tanto el canal principal de venta (Heroku) como de marketing (Google Adwords y similares) son igualmente de alcance global.

Sin embargo, hay aspectos de la operación donde se debe hacer un esfuerzo adicional para alcanzar el potencial global del negocio. Aspectos como la localización de la aplicación a otros idiomas (aunque gran parte del mercado objetivo es capaz de entender el idioma inglés), así como esfuerzos de marketing presenciales (en eventos y/o conferencias, por ejemplo) requieren de un esfuerzo dedicado. La priorización de qué mercados pueden ser atacados con estos esfuerzos deberá ser determinada según el potencial de éstos, lo cual podrá deducirse según el patrón de uso

Una posible estrategia es apuntar hacia los hubs de emprendimiento global, donde se concentran la mayoría de las empresas que podrían convertirse en clientes. De acuerdo a la organización Startup Gnome, los 20 lugares identificados en orden de importancia son los siguientes:

1. Silicon Valley
2. Tel Aviv
3. Los Angeles
4. Seattle
5. New York City
6. Boston
7. London
8. Toronto

9. Vancouver
10. Chicago
11. Paris
12. Sydney
13. Sao Paulo
14. Moscow
15. Berlín
16. Waterloo
17. Singapur
18. Melbourne
19. Bangalore
20. Santiago

No es sorprendente la presencia dominante de ciudades estadounidenses en el ranking, lo que hace natural el plantear al menos una estrategia de networking hacia aquel país. O incluso contar con operaciones allí.

En cualquier caso, la constitución legal de la empresa será realizada en dicho país desde el trimestre inicial, no sólo por los atractivos para posibles inversionistas, sino también por los requerimientos legales para participar en cualquier marketplace o sistema de pago global que sea operado desde los Estados Unidos.

5. CONCLUSIONES

Las conclusiones del presente trabajo se separan en dos áreas principales descritas en las secciones a continuación: Conclusiones sobre el desarrollo del modelo de negocio y conclusiones sobre el potencial futuro del negocio.

5.1. Desarrollo del Modelo de Negocio

La figura 5.1 muestra la comparación entre el canvas con que se inició la serie de experimentos y análisis sobre ese modelo de negocios tentativo y el canvas resultante al final del proceso.

Como es apreciable a primera vista, el sector superior izquierdo del modelo no sufrió mayores cambios. Esto se explica tanto por no contener elementos determinados como críticos de validar, como por la existencia de material y experiencias propias que determinan los recursos y las actividades críticas de una plataforma SaaS.

De todas formas se observa un mucho mayor desarrollo de la estructura de costo que en la versión final está basada en los planes concretos a ofrecer. Además se incorporaron elementos de costos fijos al modelo.

Los mayores cambios ocurrieron (como es de esperar en un proceso de descubrimiento y validación de clientes) en el sector derecho del canvas. La propuesta de valor está mucho mejor definida, tanto en términos de diferenciación (que explican por qué es posible la introducción de una nueva oferta a un mercado ya establecido) como en estar enfocados desde la perspectiva del valor generado para el usuario (en lugar de ser un simple listado de funcionalidades).

De igual forma, los flujos de ingresos del modelo obtenido al final pueden ser descritos en base a planes definidos en base a la interacción con usuarios finales. Más aún, desde allí surge la comprensión de que hay espacio para un producto en el segmento alto del mercado, con un usuario dispuesto a pagar el valor adicional generado.



Figura 5.1: Canvas inicial (arriba) y final (abajo)

Si bien el modelo resultante tiene una carencia importante en la validación del comprador económico de la solución, se observa el desarrollo desde un mero esbozo de modelo negocio hacia un modelo con bases cualitativas y cuantitativas, al menos en

sus elementos más críticos (propuesta de valor, segmentos de clientes, flujo de ingresos, estructura de costos, recursos clave).

En términos cuantitativos se tienen 10 elementos del canvas convertidos en hechos validados y basados en interacción directa con usuarios objetivo, quienes han tenido que invertir tiempo significativo en el proceso (con un promedio conservador de 1 hora dedicada a cada experimento, cada voluntario invirtió 8 horas en el proceso). Y en términos cualitativos el avance realizado puede describirse en torno al modelo propuesto por Blank y Dorf[1] diagramado en la figura 5.2.

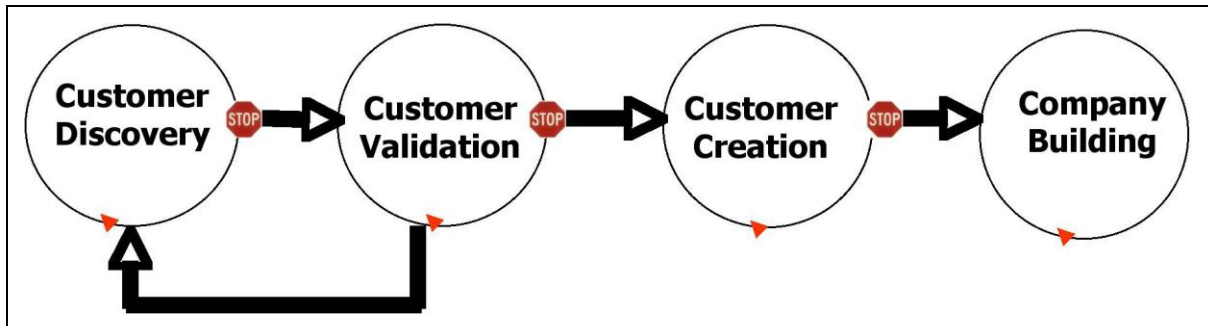


Figura 5.2: Cuatro pasos de la construcción de un negocio según modelo de Steve Blank.

El trabajo comenzó basado exclusivamente en supuestos. El primer paso, el “descubrimiento de clientes” fue realizado exhaustivamente, convirtiendo supuestos más críticos en hechos concretos. Y se dieron algunos pasos en la validación de clientes en cuanto a la validación inicial del prototipo y de los mockups presentados a los usuarios en los experimentos de las secciones 3.3.1, 3.4.1, y 3.5.1.

¿Puede concluirse que BeautifulLogs ha llegado al tercer paso del modelo, la “construcción de clientes”? La respuesta es un tajante no. Existe un potencial para que exista coincidencia entre la oferta del producto y el mercado, pero este potencial requiere más trabajo de verificación. En otras palabras: El negocio aún se encuentra en el paso 2, la “validación de clientes” (y podría volver al paso 1 si esta validación fracasara en encontrar un modelo escalable y repetible para el negocio).

Ahora bien, de acuerdo al modelo de Blank y Dorf[1], para superar el paso 2 es necesario construir un producto mínimo viable para desarrollar el proceso de ventas de manera iterativa. Es posible que la construcción de ese producto mínimo viable pueda ser parte del alcance de un trabajo de titulación para un caso donde el punto de partida sea más avanzado. Sin embargo, en casos donde los recursos se abocan al descubrimiento de clientes, es razonable considerar la validación de clientes como un trabajo futuro⁶

⁶ En retrospectiva, el autor es de la opinión de mejorar el modelo presentado en el capítulo 2 de modo de establecer de manera explícita el status inicial del modelo de negocios tomando como referencia el planteado por Blank y Dorf

Resumiendo: El desarrollo del modelo de negocio permitió evolucionar BeautifulLogs desde un concepto a un modelo de negocios tentativo basado en hechos (¡y clientes!) reales. Los cambios y aprendizajes de dicho proceso fueron claves y en más de una ocasión sorprendieron al autor. Esto permite concluir que el modelo propuesto en el capítulo 4 tiene un potencial de encajar con un mercado real.

5.2. Futuro del Negocio

Desde una perspectiva general se pueden obtener dos grandes conclusiones del mercado de BeautifulLogs:

1. La propuesta de valor, arraigada en un sub-nicho “premium” (dentro del nicho de gestión de logs para aplicaciones en la nube) conecta con necesidades reales de usuarios, quienes muestran un entusiasmo importante por el producto. Esto se traduce en que 3 de los 15 voluntarios están dispuestos a ayudar a convencer a sus organizaciones de usar el producto incluso antes de estar 100% operativo), y otros 2 están dispuestos a hacerlo una vez el producto esté en su etapa final. En otras palabras: La propuesta de valor conecta con la necesidad del segmento de clientes.
2. A pesar de la propuesta de sub-nicho, el mercado total es interesante, debido al crecimiento exponencial en el mercado, que determina fuertemente en tamaño del mercado alcanzable: Se estima que Heroku aloja alrededor de 5 millones de aplicaciones, y Heroku es uno de muchos actores en este mercado donde también participan entre otros AppEngine (Google), BeanStalk (Heroku), OpenShift (Red Hat).

Por lo tanto desde esa perspectiva general se puede afirmar que la oportunidad es atractiva.

Una mirada más al detalle del modelo propuesto en el capítulo 4 muestra números auspiciosos pero muy sensibles a las variables de crecimiento neto de clientes y de conversión efectiva de clientes, como lo muestra la millonaria variación en flujo de caja proyectado. Por ejemplo, si se toma el flujo proyectado a 3 años, se tiene un rango en el balance de caja que va desde USD 68.584 en el caso pesimista hasta USD 5.395.269 en el caso optimista. Lo que refleja el nivel de riesgo implícito en el modelo propuesto.

Si bien esta combinación de factores (oportunidad atractiva pero altamente riesgosa) es común en negocios de este tipo, eso no niega el valor de reducir el riesgo lo más posible para hacer más factible la atracción de capital de inversión. Una forma de bajar este riesgo es continuar el proceso de “validación de clientes” con la construcción de un producto mínimo viable tal como se mencionaba al final de la sección anterior.

Dicha construcción debe continuar la metodología establecida en este trabajo de título: plantear hipótesis sobre la que se puedan realizar experimentos y obtener conclusiones. Al momento de concluir este trabajo de título el autor se encuentra buscando opciones de financiamiento para dicha etapa incluyendo su ejecución como un proyecto dentro de Continuum, las llamadas 3F (“family, friends and fools”) y/o concursos públicos.

6. BIBLIOGRAFÍA

1. Steve Blank and Bob Dorf. The Startup Owner's Manual. K&S Ranch Press, first edition, 2012.
2. BuildWith.com — Cloud PaaS Usage [online]. URL: <http://trends.builtwith.com/hosting/cloud-paas> [cited 15 de Junio de 2014].
3. Cisco Global Cloud Networking Survey 2012 [online]. URL: http://www.cisco.com/c/dam/en/us/solutions/enterprise-networks/2012_Cisco_Global_Cloud_Networking_Survey_Results.pdf [cited 25 de Marzo de 2014].
4. River Cities Capital Fund. 2011 SaaS Operating Metrics & Valuation Benchmarking. 2011.
5. Logentriesl [online]. URL: <http://logentries.com/>.
6. Loggly [online]. URL: <http://loggly.com>.
7. Alexander Osterwalder and Yves Pigneur. Business Model Generation. Wiley, 2010.
8. 2013 Pacific Crest SaaS Survey [online]. URL: <http://trends.builtwith.com/hosting/cloud-paas> [cited 25 de Junio de 2014].
9. Papertrail [online]. URL: <http://papertrailapp.com>.
10. Cloud Star Heroku Preps for Second Act With New CEO — Wired Magazine [online]. URL: <http://www.wired.com/2013/06/new-heroku-ceo/> [cited 30 de Junio de 2014].
11. Platform as a Service (PaaS) is emerging [online]. URL: <http://www.redhat.com/enterprise-paas/> [cited 30 de Junio de 2014].
12. Eric Ries. The Lean Startup. Crown Publishing, third edition, 2011.
13. Jeffry Timmons and Stephen Spinelli. New Venture Creation: Entrepreneurship for the 21st Century. McGraw-Hill Irwin, eighth edition, 2009.
14. Logfile — Wikipedia [online]. URL: <http://en.wikipedia.org/wiki/Logfile> [cited 25 de Marzo de 2014].
15. Not invented here — Wikipedia [online]. URL: http://en.wikipedia.org/wiki/Not_invented_here [cited 21 de Mayo de 2014].

7. ANEXOS

ANEXO A. GLOSARIO DE TÉRMINOS

Airbrake.io Servicio para notificar vía email a desarrolladores cuando ocurre un error grave (técnicamente llamadas “excepciones”) en una aplicación.

Amazon Web Services Colección de servicios en la “nube” ofrecidos por Amazon. Considerado un servicio IaaS). Entre los principales servicios ofrecidos se encuentran EC2 (Elastic Computing Cloud) para arrendar servidores virtuales por hora y S3 (Simple Storage Service) para arrendar almacenamiento con un costo determinado por el uso efectivo de espacio mensual.

GitHub Plataforma para facilitar el desarrollo de software de manera colaborativa. Su función principal es servir de hosting del repositorio de código fuente de un proyecto, lo que complementa con herramientas de colaboración como una herramienta de seguimiento de bugs, wikis para documentación de un proyecto y sistemas de comentarios. Es extremadamente popular entre los desarrolladores, contando entre los proyectos alojados el kernel de Linux; herramientas de desarrollo como Rails, Node.js y JQuery; el código de la criptomoneda Bitcoin, entre muchos otros.

Heroku Colección de servicios en la “nube” enfocado en desarrolladores con el objetivo de ayudarlos a construir aplicaciones que puedan escalar a miles de usuarios simultáneos. Sus principales servicios son los “Dynos” (unidades de cómputo virtuales, capaces de procesar tareas o servir aplicaciones web) y Bases de Datos administradas por el personal de Heroku. Considerado un servicio PaaS. Subsidiaria de Salesforce.com (empresa pionera de servicios en la nube; Véase SaaS)

IaaS Infraestructura como Servicio (Infrastructure as a Service). Modelo de negocio en el que un proveedor arrienda el uso de infraestructura de cómputo o almacenamiento bajo un modelo de precio flexible y/o variable (por hora de cómputo, por gigabyte almacenado, etc).

LogEntries Servicio de manejo de logs en la nube, enfocado en “hacer fácil el manejo de logs y analíticas”.

Loggly Servicio de manejo de logs en la nube, enfocado en “resolver problemas operacionales rápido”.

Papertrail Servicio de manejo de logs en la nube, enfocado en “manejo de logs libre de frustración, sean de 2 o de 2 mil servidores”.

PaaS Plataforma como Servicio (Platform as a Service). Modelo de negocio en el que un proveedor arrienda el uso de una plataforma de cómputo o almacenamiento. Se diferencia de IaaS en el nivel de abstracción provisto: Mientras IaaS ofrece infraestructura de bajo nivel (ej: servidores virtuales, discos virtuales), PaaS ofrece una abstracción de más alto nivel. En lugar de servidores virtuales y discos virtuales se

suelen ofrecer unidades de cómputo o servicios específicos (servidor web, servidor de base de datos, servidor de correo, etc) que evitan al cliente tener que gestionar y mantener el software instalado sobre la infraestructura virtual.

SaaS Software como Servicio (Software as a Service). Modelo de negocio en el que un proveedor, en lugar de vender una licencia de uso perpetuo a sus clientes, arrienda el uso de software (generalmente accesible mediante un navegador web y alojado en “la nube”) típicamente en pagos mensuales.

ANEXO B. SUPUESTOS DEL MODEL DE NEGOCIO

Supuestos Validados

- A. (CS): Los usuarios/clientes de proveedores de infraestructura y plataformas como servicio en la nube (ej: Heroku, AWS, Rackspace, etc) presentan los dolores diagnosticados en la evidencia del problema: Dificultad en acceso, conciliación y búsqueda en los logs, además de inconveniencia en el uso de los logs. Supuesto validado en Experimento 1.
- B. (VP): Nuestros usuarios/clientes no quieren preocuparse de archivos ni de en qué máquina “viven” los logs. Supuesto validado en Experimento 2.
- C. (VP): Nuestros usuarios/clientes quieren ver en tiempo real como se comportan sus aplicaciones (ver el “latido del corazón” de su aplicación). Supuesto validado en Experimento 2
- E. (VP, R\$): La mejor propuesta de valor de BeautifulLogs le permite cobrar un precio mayor que la competencia (es decir, la diferenciación es valorada por los usuarios/clientes). Supuesto validado en Experimento 5.
- G. (VP, R\$): Un dolor de nuestros usuarios/clientes cuando usan productos alternativos es tener límites en sus planes. Supuesto validado en Experimento 5.
- H. (CS, R\$, C\$): Podemos segmentar a los clientes en 3 distintos planes de precios. Supuesto validado en Experimento 6.
- I. (C\$, R\$, KR): Dentro del nicho de usuario de Heroku es posible llegar al breakeven en el costo de almacenamiento y procesamiento de las características extras ofrecidas por los planes definidos. Supuesto validado en Experimento 7.

Supuestos Invalidados

- D. (VP): Nuestros usuarios/clientes valoran el “apoyo forense” a la hora de encontrar errores inspeccionando los logs. Supuesto invalidado en Experimento 3.

Supuestos Irrelevantes

- J. (KP, CH): Es factible para una empresa constituida en Chile ofrecer productos vía el marketplace de Heroku. Supuesto irrelevante pues constituir empresa en EEUU es simple.

Supuestos Críticos no Validados

- F. (CS, R\$): Nuestros usuarios/clientes (desarrolladores) tienen el poder de convencer al comprador económico (sus empleadores) para efectuar la compra.

Otros Supuestos Críticos

- (CS): Los desarrolladores de soluciones cloud fuera de Heroku tienen necesidades y dolores similares a los que trabajan sobre Heroku.