

Universidade de São Paulo  
Instituto de Matemática e Estatística  
Bachalerado em Ciência da Computação

Caio Teixeira da Quinta  
Eugênio Augusto Jimenes

**Plataforma Web para divulgação e centralização de  
eventos aplicando conceitos de Métodos Ágeis  
e Lean Startup**

São Paulo  
Dezembro de 2016

# Plataforma Web para divulgação e centralização de eventos aplicando conceitos de Métodos Ágeis e Lean Startup

Monografia final da disciplina  
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisor: Prof. Dr. Dr. Alfredo Goldman vel Lejbman  
Cosupervisor: Jorge Melegati

São Paulo  
Dezembro de 2016

# Agradecimientos



# Resumo

<em desenvolvimento>

É notável que a Cidade Universitária possui uma imensa diversidade acadêmica e cultural que se manifesta em uma variedade de eventos sendo promovidos e organizados pela Universidade de São Paulo além de iniciativas da própria comunidade para ocupar o espaço público.

Levando em consideração a quantidade de eventos acontecendo de forma simultânea, a extensão física do campus, a pulverização dos eventos espalhando-se por toda sua extensão e a quantidade de informações dispersas entre as várias redes de comunicação oficiais ou não muitas vezes não tomamos conhecimento a tempo de alguma oportunidade que poderia ser interessante.

Foi pensando na centralização e divulgação de eventos que foi proposto criar o USP Eventos, uma plataforma web colaborativa na qual qualquer membro frequentador no campus pode se informar do que acontece no mesmo além de organizar seu próprio evento.

Tomando como base uma metodologia de projeto baseada nos conceitos de Métodos Ágeis e Lean startup foi proposto desenvolver um sistema em Ruby on Rails utilizando 3 ciclos de Build Measure Learn sempre pautando o desenvolvimento o sistema segundo o aprendizado obtido através dos comentários e críticas dos usuários.

**Palavras-chave:** eventos, métodos ágeis, lean startup, desenvolvimento web.



# Abstract

Elemento obrigatório, elaborado com as mesmas características do resumo em língua portuguesa.

**Keywords:** keyword1, keyword2, keyword3.





# Contents

<b>Lista de Abreviaturas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação e Objetivos . . . . .	1
1.2 Capítulos . . . . .	1
<b>2 Lean Startup</b>	<b>3</b>
2.1 Lean Startup: O que é . . . . .	3
2.2 Produto Mínimo Viável (MVP) . . . . .	3
2.3 O ciclo de Build Measure Learn . . . . .	4
2.3.1 O modelo Cascata . . . . .	4
<b>3 Conclusões</b>	<b>7</b>
<b>A Título do apêndice</b>	<b>9</b>



# Lista de Abreviaturas

MVP   Produto Mínimo Viável ( *Minimum Viable Product* )



# Chapter 1

## Introdução

### 1.1 Motivação e Objetivos

É notável que a Cidade Universitária possui uma imensa diversidade acadêmica e cultural que se manifesta em uma variedade de eventos sendo promovidos e organizados pela Universidade de São Paulo além de iniciativas da própria comunidade para ocupar o espaço público.

Levando em consideração a quantidade de eventos acontecendo de forma simultânea, a extensão física do campus, a pulverização dos eventos espalhando-se por toda sua extensão e a quantidade de informações dispersas entre as várias redes de comunicação oficiais ou não muitas vezes não tomamos conhecimento a tempo de alguma oportunidade que poderia ser interessante.

Foi proposto criar um Sistema Web colaborativo para centralizar e divulgar os eventos que são organizados pela Universidade de São Paulo assim como pela comunidade que frequenta o campus e para isso adotamos uma metodologia de projeto seguindo os preceitos de Lean Startup e Métodos Ágeis.

A metodologia Lean Startup tem como foco evitar o desperdício de tempo e recursos focando em um rápido aprendizado dos interesses do público alvo para refinamento do produto portanto foi adotada no desenvolvimento do USP Eventos pois dada sua natureza colaborativa do projeto era vital podermos coletar informações sobre o nosso público, a comunidade usuária do campus da Cidade Universitária, a fim de que o projeto estivesse alinhado com seus interesses.

Para atingir os objetivos propostos pela metodologia Lean Startup aplicamos conceitos de Métodos Ágeis tais como testes, desenvolvimento iterativo, incremental e contínuo.

### 1.2 Capítulos



# Chapter 2

## Lean Startup

### 2.1 Lean Startup: O que é

TO DO

### 2.2 Produto Mínimo Viável (MVP)

Uma definição de Produto Mínimo Viável por Eric Ries: *"A Minimum Viable Product is that version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort."*

A idéia central do conceito de MVP é maximizar a validação de aprendizado sobre um produto utilizando o menor esforço possível. Como na maioria das startups os recursos financeiros e humanos são bastante escassos o tempo de validação do produto e determinação do interesse do público é um fator decisivo para o sucesso da mesma.

Um MVP deve possuir 3 características:

1. Ter valor suficiente para que uma pessoa queira utilizá-lo ou comprá-lo.
2. Possui suficientes benefícios para reter os chamados usuários pioneiros.
3. Ser capaz de prover um ciclo de feedback suficiente para guiar o desenvolvimento.

Durante a concepção do projeto são definidas algumas hipóteses sobre o produto e na etapa do MVP é definido então qual será o seu núcleo, ou seja, quais funcionalidades ou estratégias queremos testar de modo que possamos validar as nossas hipóteses iniciais e obter o máximo de aprendizado possível.

Além da validação de aprendizado outra vantagem significativa provida pelo MVP é evitar desperdícios de tempo e recursos como por exemplo através do aprendizado resultante do MVP é possível evitar que se desperdice mais tempo em recurso em uma funcionalidade que a princípio parecesse útil e necessária mas que foi constatado através de uma validação bem construída que a mesma não era interessante para os usuários podendo assim ser revista antes que fossem gastas mais horas de desenvolvimento.

Os termos de "mínimo" e "máximo" frequentemente se mostram vagos na documentação do que é um MVP, citando Eric Ries: *"[...]the definition's use of the words maximum and minimum means it is decidedly not formulaic. It requires judgment to figure out, for any given context, what MVP makes sense."*, levando em consideração é importante frisar que um MVP não é necessariamente um produto completo com as funcionalidades mínimas e sim um conjunto de características mínimas que configuram o serviço ou produto que está

sendo oferecido. Dessa forma um MVP pode ser apenas um protótipo, um produto completo ou mesmo apenas um *mock-up* do que será oferecido na versão completa.

Alguns tipos de MVP são:

- *Vídeo Explicativo*: Um vídeo curto contendo uma explicação clara do que o seu produto faz e porque as pessoas deveriam utilizá-lo. Esse é o caso do dropbox que fez um video com cerca de 5 minutos explicando o que era o seu serviço.
- *Landing Page*: Criar uma página inicial contendo uma explicação detalhada do que é o produto que você irá oferecer juntamente com um formulário de contato. Através de uma configuração simples pelo Google Analytics é possível manter um registro de conversões (no caso cadastros do formulario) a fim de medir o interesse das pessoas no seu produto.
- *MVP "Mago de OZ"*: A idéia é criar uma página visualmente completa que funcione como se fosse o produto final mas que por trás existe alguém manualmente executando as tarefas. Esse foi o caso da Zappos hoje a maior vendedora de sapatos dos Estados Unidos.
- *MVP com Consierge*: Ao invés de prover um produto você realiza manualmente o serviço realizando exatamente os mesmos passos que o usuário teria utilizando seu serviço. É um método não escalável e lento para executar pois requer que você esteja em contato direto com o cliente e realize as tarefas manualmente porém isso permite um rápido aprendizado sobre o produto e o cliente.

A empresa Food on the Table ajuda seus consumidores a criarem listas de compras, acharem receitas e conseguirem descontos nos ingredientes em seus supermercados favoritos, inicialmente seus fundadores encontraram uma senhora interessada no serviço e por 10 dólares/semana eles mantinham as listas de compra e procurava, por descontos nos supermercados em que ela fazia compras.

## 2.3 O ciclo de Build Measure Learn

Um ciclo de build measure learn é uma abordagem de desenvolvimento do produto que aprimora o tradicional modelo de desenvolvimento "Cascata" utilizado de forma abrangente durante o século XX.

### 2.3.1 O modelo Cascata

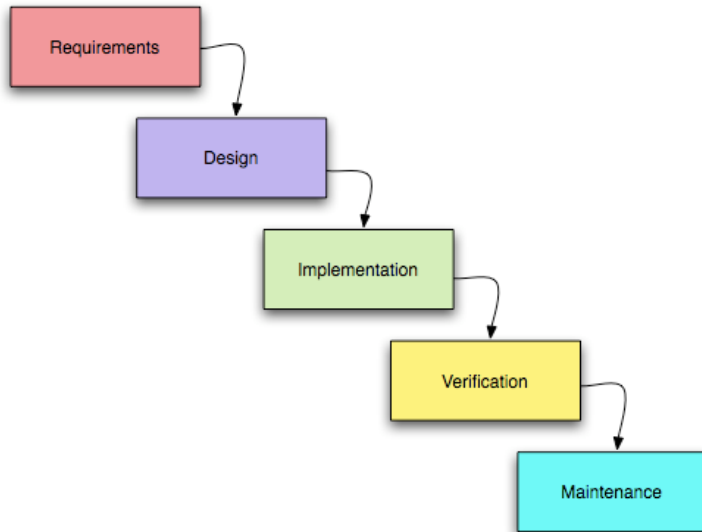
O nome "Cascata" (Waterfall) é bastante literal e tem origem na própria estrutura do método que seguia uma série de passos de forma sequencial de maneira bastante direta, como se fosse uma cascata. ??.

O método consiste em um desenvolvimento sequencial e não-interativo inicializando com a parte de análise de requisitos e de forma subsequente temos as etapas de design de projeto, implementação, verificação e manutenção.

Uma rápida explicação dos passos:

- *Requerimentos*: Realizar a análise de requisitos do projeto.
- *Design de Projeto*: Focando na estrutura de dados, arquitetura do software, detalhes procedais e caracterização das interfaces é formulado um documento de forma a apresentar os requerimentos de uma forma que possa ser interpretado pelos programadores.





**Figure 2.1:** *O Modelo Cascata.*

- *Implementação:* Etapa da codificação do projeto propriamente dita.
- *Verificação:* Etapa para teste do produtos visando eliminar qualquer bug que possa ter passado despercebido refinar a lógica interna do software caso necessário.
- *Manutenção:* Etapa para instalação do sistema no cliente, configuração de servidores, etc.

Algumas das críticas dessa abordagem é que dificilmente um desenvolvimento de software segue todas as etapas da forma como é proposto e nem sempre o cliente sabe definir bem os requisitos resultando em tempod e desenvolvimento desperdiçado em funcionalidades que no fim não atenderam as expectativas do cliente além de mudanças tardias no escopo do projeto que, além de encarecer o custo total, poderiam ter sido evitadas e contornadas de maneira mais satisfatória em um modelo com uma maior interação entre cliente e desenvolvedor.

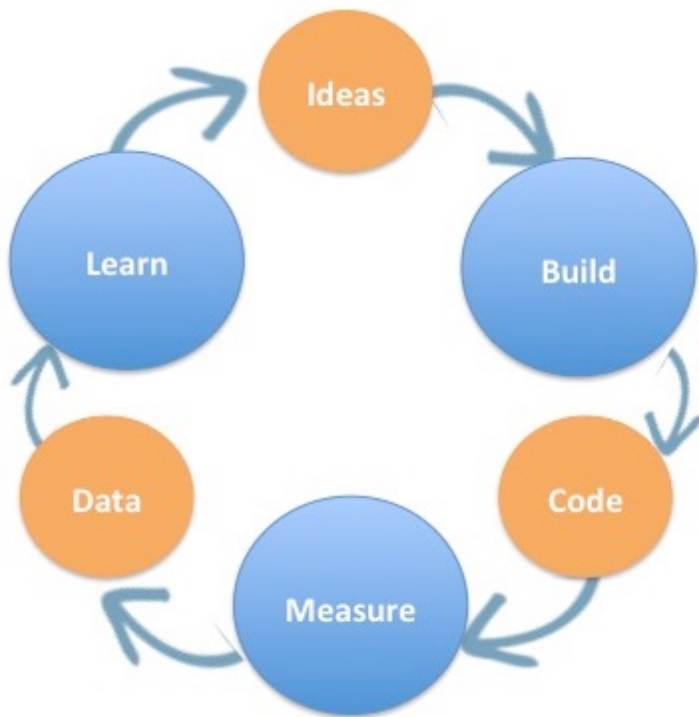
Ao contrário do modelo cascata no qual o contato do cliente com o software se dá apenas no fim do processo de desenvolvimento na etapa de testes o método de Build Measure Learn privilegia desde o início uma interação contínua com o cliente.

### 2.3.2 Build Measure Learn

O ciclo de Build Measure Learn é uma das idéias fundamentais do Lean Startup e consiste de um ciclo de feedback rápido cujo principal objetivo é eliminar as incertezas sobre as hipóteses do produto através de um aprendizado rápido sobre o mesmo assim minimizando os riscos e custos desnecessários que persistir em uma idéia equivocada poderiam causar.

Além das 3 idéias fases do ciclo vamos incluir no diagrama outras 3 menores (*ideas*, *data* e *code*) para ilustrar melhor o objetivo de cada uma das etapas.

- *Build:* Inicialmente temos algumas idéias que foram definidas a partir das hipóteses do produto que precisam ser implementadas (*code*) no MVP.
- *Measure:* Implementado o MVP coletamos dados(*data*) de uso e seguindo algumas métricas já definidas avaliamos seu desempenho. Todo o ciclo é baseado na idéia de aprendizado rápido para aprender o máximo possível sobre como os usuários reagem as idéias implementadas.



**Figure 2.2:** *O ciclo de Build Measure Learn.*

- *Learn:* A partir então da análise dos dados coletados podemos inferir sobre continuar sobre como continuar o desenvolvimento e o que funcionou ou não. Feito isso implementamos as novas idéias geradas e começamos novamente o ciclo para validá-las.

As fase do ciclo não precisam necessariamente ocorrer em ordem podendo se sobrepor ou mesmo serem unidas dependendo de como for o ciclo de desenvolvimento.

As alterações de software precisam ser feitas de maneira rápida de modo a testar o mais rápido possível novas idéias portanto é importante ressaltar que as funcionalidades devem se manter simples e diretas pois o foco é o aprendizado gerado e não desenvolver um software ou um protótipo completo pois dessa maneira é possível evitar que sejam desperdiçados recursos em uma funcionalidade que pode não ser bem recebida e não ter continuidade.

Apesar do desenvolvimento ser focado na simplicidade isso não quer dizer produzir um produto de baixa qualidade. Existem ferramentas que auxiliam na integração contínua do software e é comum realizar testes automatizados garantindo que seja possível manter um desenvolvimento consistente e confiável sem comprometer o tempo de execução pois um produto de baixa qualidade ou que possua bugs pode resultar em uma taxa de rejeição inesperada devido a problemas de implementação fazendo com que as conclusões do ciclo possam ser enviesadas.

Cada ciclo de build measure learn testa idéias específicas portanto é comum realizar diversos ciclos implementando e testando idéias diferentes

Dada sua natureza interativa e o desenvolvimento incremental que o o ciclo de Build Measure Learn propõe o o projeto final costuma estar mais refinado e alinhado com as expectativas dos clientes em contraponto aos problemas apresentados pelo modelo em Cascata sendo assim uma evolução bastante interessante do mesmo.

# Chapter 3

## Conclusões

[illegible]

<sup>1</sup>Exemplo de referência para página Web: [www.vision.ime.usp.br/~jmena/stuff/tese-exemplo](http://www.vision.ime.usp.br/~jmena/stuff/tese-exemplo)



## Appendix A

# Título do apêndice

[illegible]

