

01

## Waterfall

### Transcrição

Dentre as ideias que tivemos para ajudar o usuário a não viajar sozinho, temos:

- Chat
- Reviews
- Criar grupos
- Passeios próximos
- Alerta de passagens

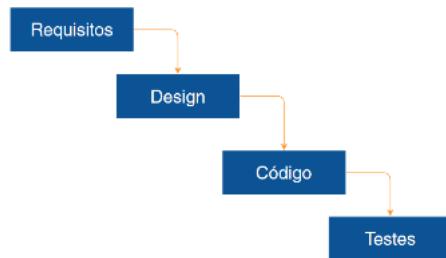
Qual dessas funcionalidades nós priorizaremos?

E se escolhermos lidar com todas as partes ao mesmo tempo? Como existem algumas dessas atividades que demoraremos para finalizar, ou será necessário contratarmos terceiros, imagine que chamemos um analista de requisitos para nos ajudar.

Suponhamos que essa pessoa demore 2 meses para entregar o produto. Podemos pensar que é necessário elaborar o design, as telas do aplicativo após o recebimento dos requisitos, e a criação disso tudo levará em média 4 meses. Também será necessário um front-end ou back-end para finalizar a parte de programação e desenvolvimento. A pessoa responsável por isso também acaba dando um prazo de 4 meses. Até agora temos a soma de 10 meses para a finalização do aplicativo.

Por fim, falta testar tudo isso que construímos e, pensando nisso, podemos terceirizar essa etapa, resultando em um mais um prazo de 2 meses. Assim, a duração do desenvolvimento do projeto será, ao todo, de 12 meses! Será que as necessidades e objetivos do cliente serão as mesmas quando chegarmos ao final desse projeto que levará todo este tempo de produção?

A ideia de começarmos uma atividade apenas quando outra termina implica no **Waterfall**, em que inicialmente trabalham-se os requisitos, passando para o design, e depois encaminha-se para a programação, ao final da qual fazem-se os testes. Ou seja, é uma lógica de cascata:



Nesse meio tempo o usuário pode ficar frustrado, seja esperando o aplicativo, ou o desenvolvimento de uma nova funcionalidade. Não que o modelo *Waterfall* não funcione, na verdade diversas empresas o utilizam. Porém, ele vai contra as ideias da engenharia da agilidade.

Agilidade consiste em lidarmos com pequenas partes de um projeto, em vez do todo. No lugar de focarmos no todo e fazer um aplicativo mega revolucionário, hiper completo, vamos pegar **uma ideia** e ver o que conseguimos desenvolver a partir dela.

O que apresentamos aqui é o **Produto Mínimo Viável** ou *Minimum Viable Product* (MVP), um conjunto mínimo de funcionalidades que podemos ter no produto para lançá-lo. No MVP precisaremos nos preocupar com questões sobre viabilidade.

Por ser um produto pequeno, uma fração de um todo, ele é lançado com mais agilidade. Assim, sempre que pensarmos em produtos e projetos, poderemos pensar também em como gerar valor para o usuário mais rapidamente, e em como ter um retorno rápido também.

Quem utilizará o MVP? O usuário! Então, é necessário pensarmos no MVP focando sempre no modelo mental do usuário, no usuário como sendo o centro das tomadas de decisão.



A imagem acima remete a "Como **não** construir o Produto Mínimo Viável". A imagem número 1, corresponde a uma roda. A imagem número dois, a uma estrutura além da roda. Quanto à terceira existe uma carroceria, e à quarta imagem, o carro completo.

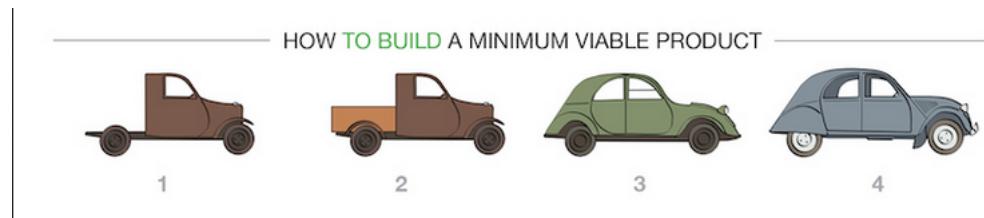
Será que conseguiremos andar pela cidade com apenas uma roda, como mostra a figura 1? E as imagens 2 e 3? Tampouco será possível nos locomovermos utilizando tais estruturas. Somente com a imagem 4 que conseguiremos efetivamente utilizar o carro para locomoção.

Outro exemplo para não construirmos um MVP é mostrado na imagem abaixo:



Embora o usuário já consiga se locomover a partir da imagem 1, a questão é que o usuário gostaria de, além disso, levar uma segunda pessoa, o que só poderia ser possível a partir da imagem 3 e, ainda, se a segurança fosse primordial, o produto só estaria disponível a partir da imagem 4.

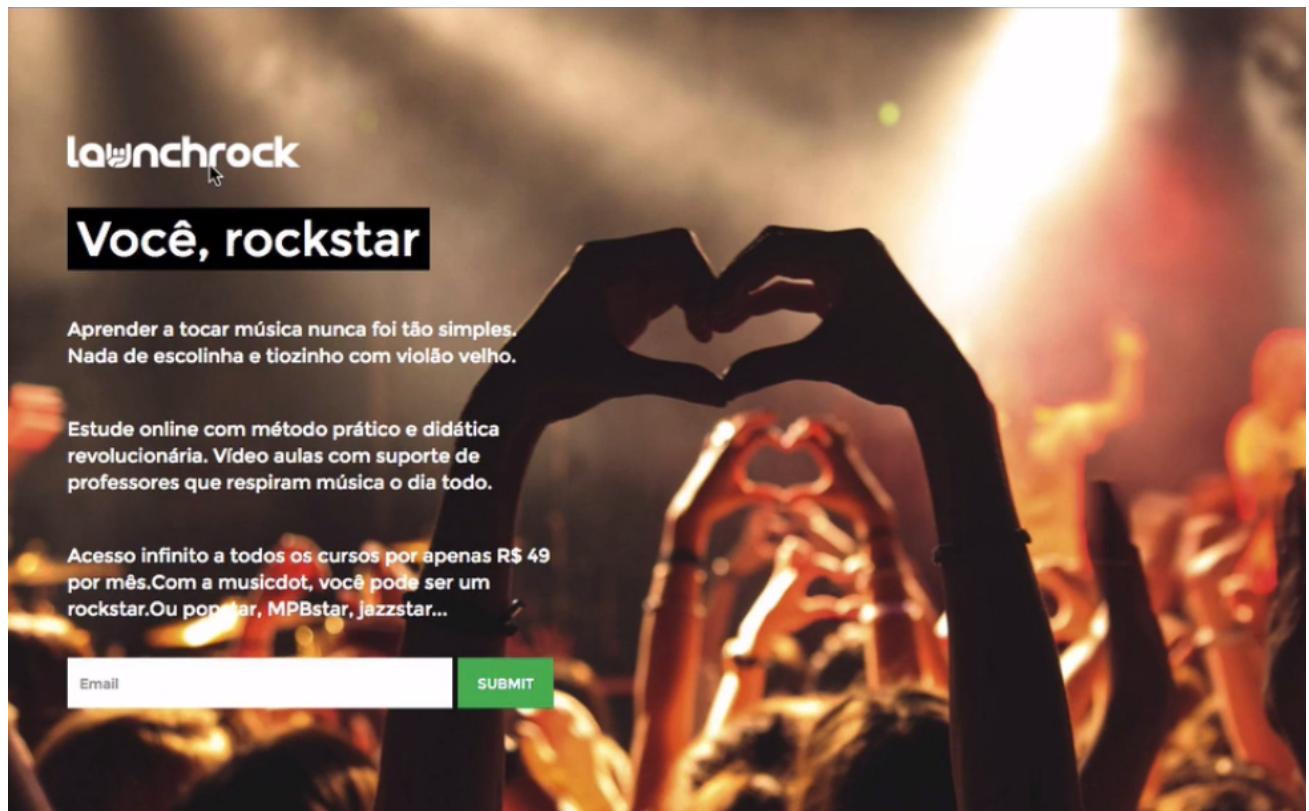
Finalmente, um exemplo de como construir um MVP.



No número 1 temos um caminhão simples, no 2 temos o mesmo caminhão com uma carroceria. No número 3 o "layout" é melhorado, e corresponde a um carro, e no 4 exibe-se um carro melhor ainda. Nesse exemplo é possível verificar que em todos os passos sempre foi gerado valor para o usuário!

É exatamente isso que desejamos que aconteça ao desenvolvemos o MVP, ou seja, que o valor seja gerado para o usuário. Vamos explorar um caso real que envolve o [musicdot \(https://www.musicdot.com.br/\)](https://www.musicdot.com.br/), uma empresa de ensino online focada em instrumentos musicais e aulas de música e canto.

Antes do site ser desenvolvido, foi medido o real interesse de diferentes indivíduos em relação ao produto, para verificar se de fato haveria interesse em pagar uma assinatura para aprender a tocar instrumentos dessa maneira. Além disto, quanto seria um valor interessante para tal produto?



A plataforma [Launchrock \(https://www.launchrock.com/\)](https://www.launchrock.com/) é composta de uma página, um texto, um *background* e um campo de busca, através dos quais é possível conseguir um levantamento de possíveis usuários de um site que muitas vezes nem chegamos a elaborar. A ferramenta serve justamente para medir este interesse.

Observe que o texto da página fala sobre acesso infinito, "chega de tiozinho com violão velho", valores mensais, e demais informações. Usar essa estratégia serve para averiguar a boa aceitação da ideia, e se as palavras empregadas atraem público.

Usando esta ferramenta, é possível testarmos distintos valores, por exemplo - em um primeiro momento o definimos com R\$49, e assim notamos que 50 pessoas manifestaram interesse. Quando colocamos o valor como R\$59, apenas 2 pessoas manifestaram vontade de fazer os cursos.