



# **Curso** **Qualidade de software**



# Qualidade de Software



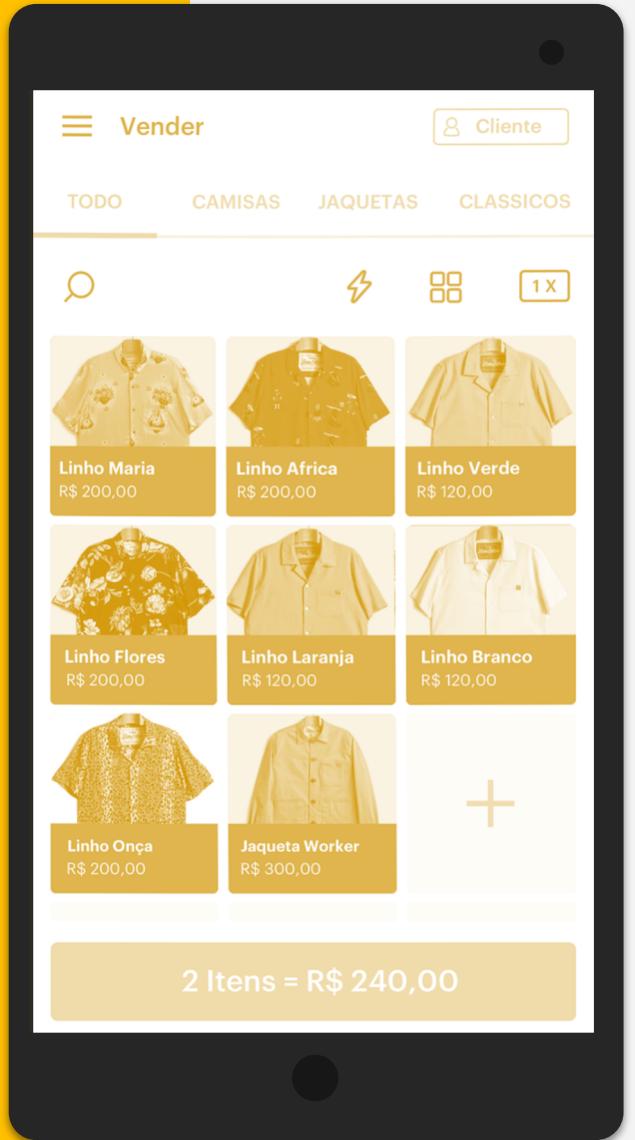
Módulo 6





# Metodologias Ágeis

Aula 1

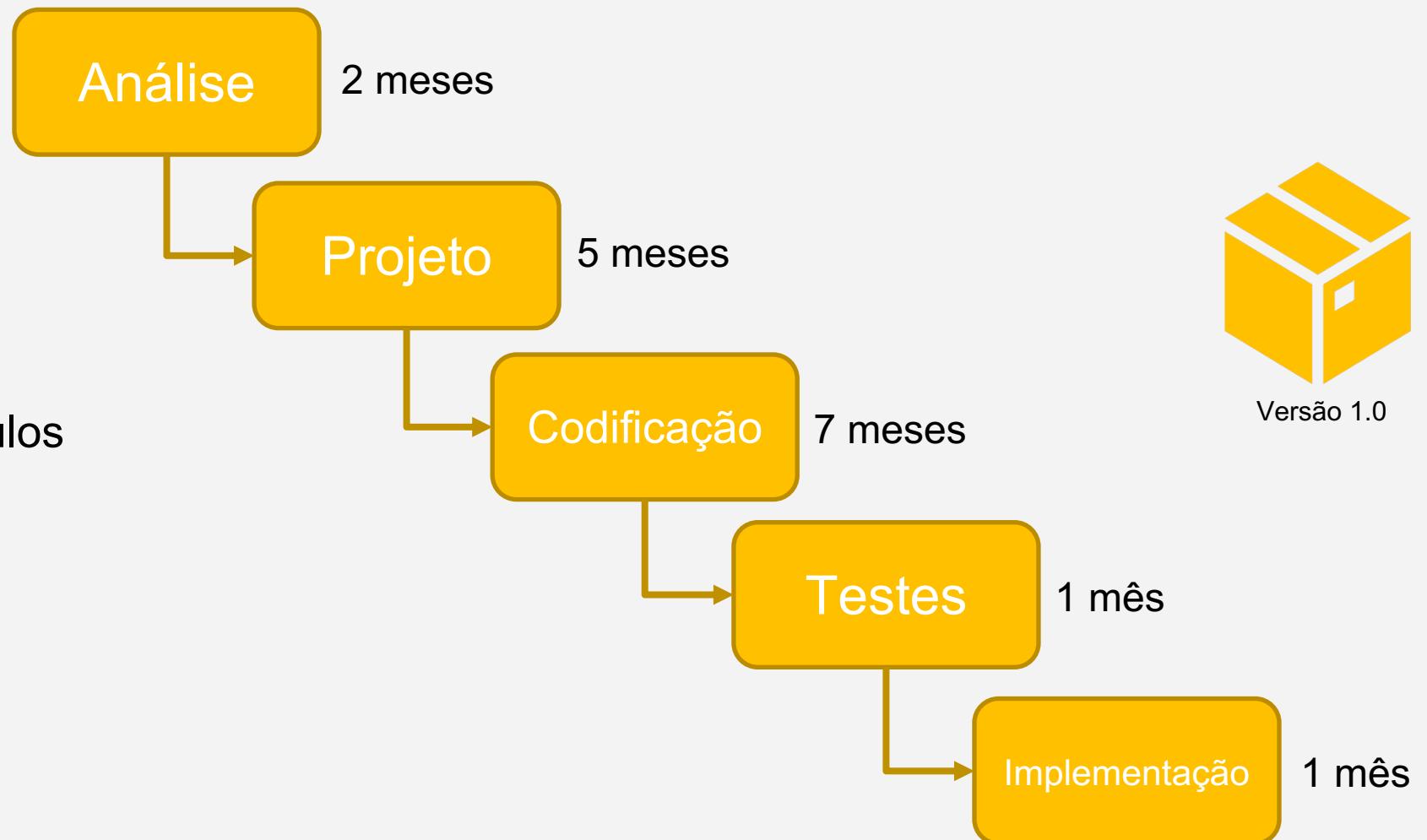


## App: Loja virtual de roupas

## Empresa A:

### Proposta:

- Entrega: Todos os módulos
- Estimativa: 16 meses
- Preço: R\$ 2.000.000,00



# Empresa B:

## Proposta:

- Entrega: Módulos que fazem sentido ao longo do tempo
- Estimativa: 2 meses para os primeiros módulos
- Preço: R\$ 2.000.000,00



Versão 1.0

Página de produtos



Versão 1.5

Cadastro



Versão 2.0

Login e Pagamento



Versão 2.1

Admin



Versão 2.2

Dashboard

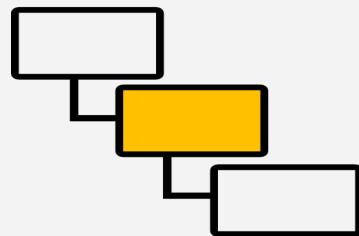


Etc.

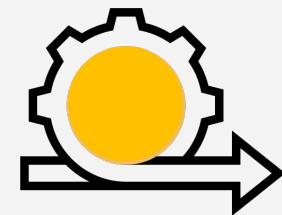
# Diferenças entre as propostas



- Proposta 1: Cascata



- Proposta 2: Ágil



# Desenvolvimento **Cascata**

- Base na Engenharia civil
  - Documentação
  - Analise
  - Design
  - Arquitetura
  - Construção (Desenvolvimento)
  - Validação (Testes)
  - Entrega (Implementação)
- Modelo mais controlado



# Modelos industriais



- Toyota - 1960
- Evitar desperdício
- Criaram métodos:
- **Lean:** Desenvolvimento Enxuto
- **Kanban:** Acompanhamento das atividades através de um quadro
- Gestão a vista



# Kanban





# Manifesto do desenvolvimento Ágil



- Em 2001, um grupo de programadores
- Já praticavam métodos ágeis: FDD, Scrum, XP, etc.
- Lançaram o o **Manifesto Ágil**
- Diretriz que tem como objetivo **satisfazer os clientes**;
- Entregas com **maior frequência**.

**ATENÇÃO:** Ágil não significa **entregar mais rápido**,  
mas sim **entregar valor mais rápido ao cliente**

# Valores



## INDIVÍDUOS E INTERAÇÕES

mais que processos e ferramentas



## COLABORAÇÃO COM O CLIENTE

mais que negociação de contratos



## SOFTWARE EM FUNCIONAMENTO

mais que documentação abrangente



## RESPONDER A MUDANÇAS

mais que seguir um plano

# Mundo **VUCA**

**V**olatility (Volátil)  
**U**ncertainty (Incerto)  
**C**omplexity (complexo)  
**A**mbiguity (Ambíguo)



# Princípios Ágeis

1

Satisfaça o consumidor



2

Aceite bem as mudanças



3

Entregas frequentes



4

Trabalhe em conjunto



5

Confie e apoie



6

Conversas face a face



7

Softwares funcionando



8

Desenvolvimento sustentável



9

Atenção contínua



10

Mantenha a simplicidade



11

Times auto-organizados



12

Refletir e ajustar



## Maiores obstáculos

- Mudança da cultura organizacional;
- Pessoas sem competências adequadas;
- Falta de apoio e participação da liderança;
- Resistência as mudanças;
- Não tem tempo e/ou orçamento disponível para transição.

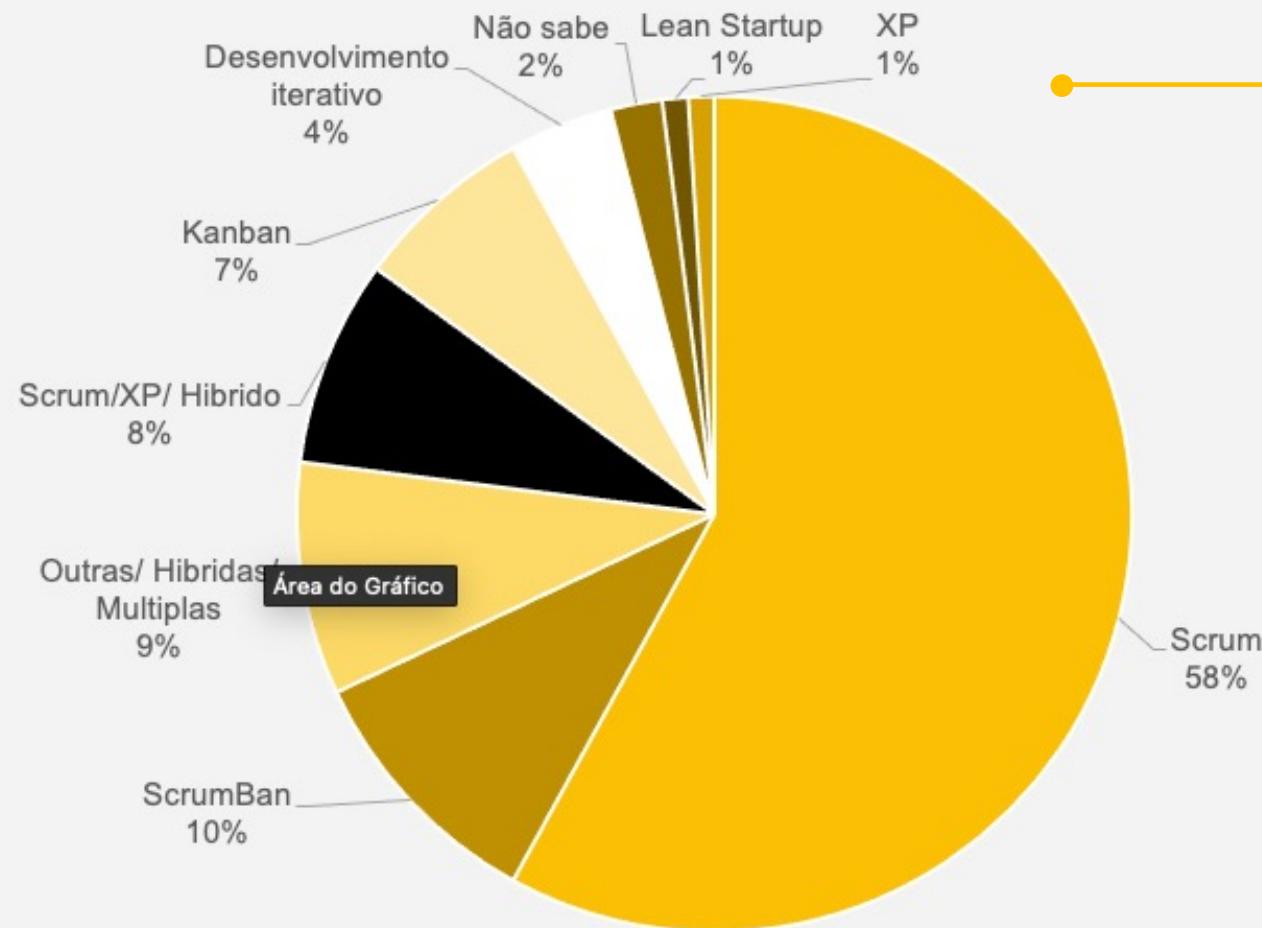


# Fatores de **sucesso**

- **Mentalidade** progressiva;
- Comportamento;
- Colaboração;
- Feedback mais rápido;
- Sem apego a ferramentas e processos;
- Melhoria contínua;
- Times **multidisciplinares**.



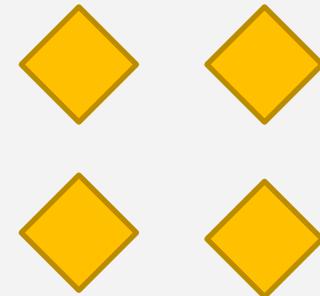
# Metodologias ágeis usadas no mundo



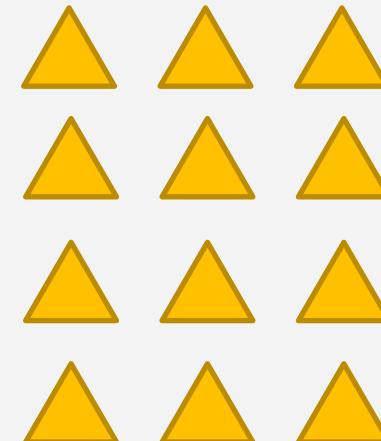
Ágil é um  
**MINDSET**



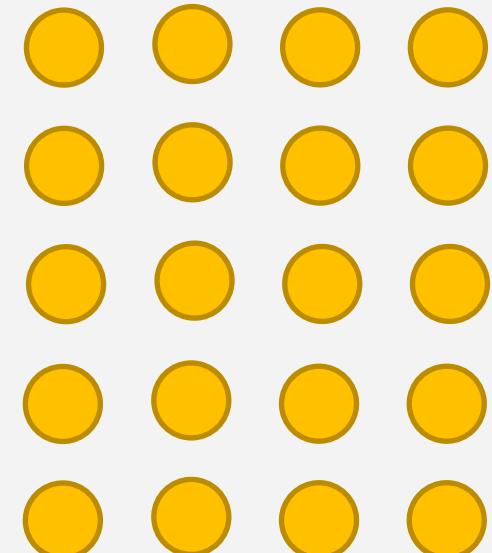
Estabelecido por  
**4 VALORES**



Fundamentados por  
**12 PRINCÍPIOS**



Manifestado por diversas  
**PRÁTICAS**



**Ser Ágil**

**Fazer Ágil**



# Scrum

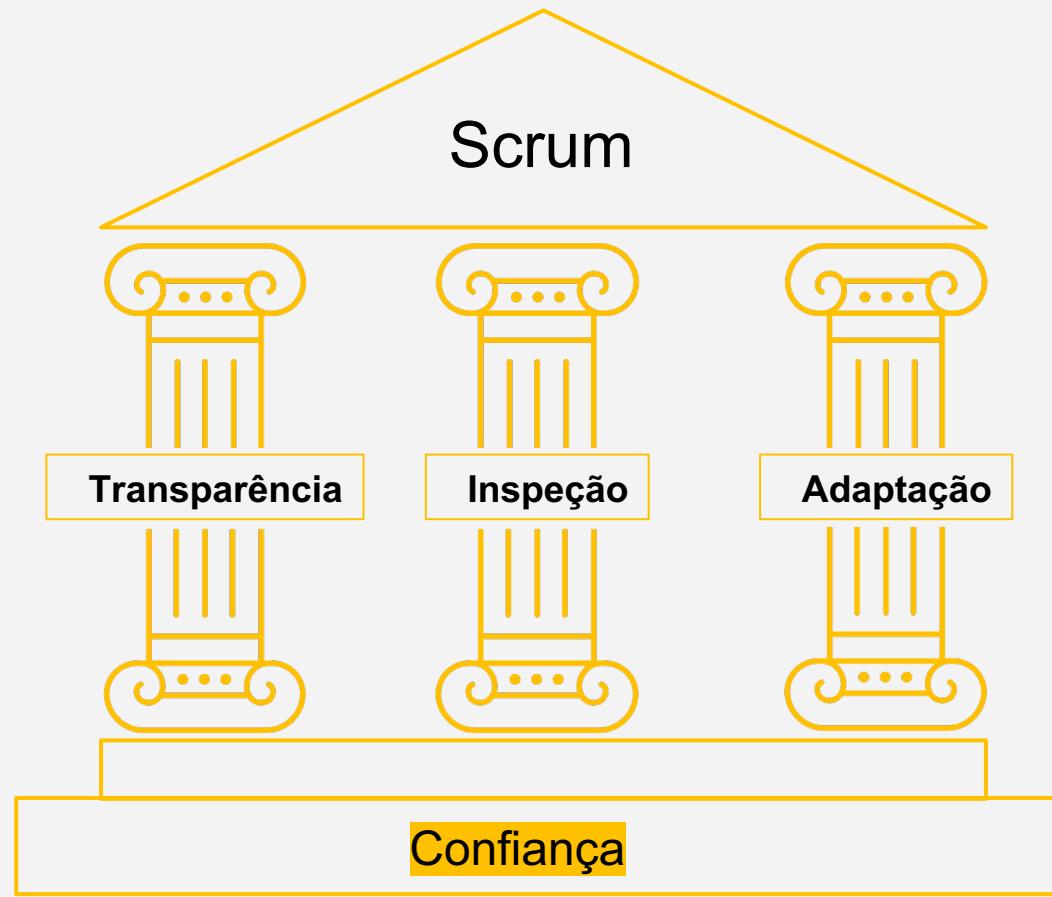


# Scrum

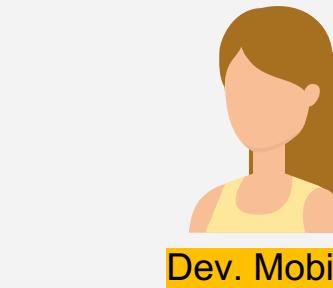
- O Scrum é uma estrutura que ajuda as equipes a trabalharem juntas;
- Uma abordagem ágil de gerenciamento de projetos;
- Estimula as equipes a aprenderem com as experiências, a se organizarem enquanto resolvem um problema;
- Estimula a reflexão sobre os êxitos e fracassos para melhorarem sempre;
- Scrum se resume em **trabalho em equipe**, desde que existe um grupo trabalhando juntos para um **objetivo**.



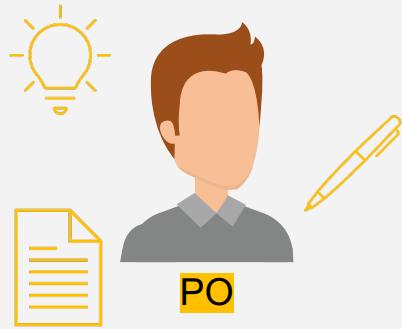
# Pilares do Scrum



# Scrum e seus papéis



# O papel do Product Owner (PO)



## **Product Owner (PO) – Foco em maximizar o valor do produto**

- Manter itens claros no backlog;
- Manter itens ordenados;
- Otimizar o trabalho do Time de Desenvolvimento;
- Garantir que os desejos estejam visíveis para todos;
- Garantir que o Time de Desenvolvimento entenda os itens no nível necessário.

# O papel do **Scrum Master (SM)**



## **Scrum Master (SM) – Pessoas e Processos (Facilitador)**

- Coach;
- Líder Servidor\*;
- Conhecedor do processo adotado (conhecer, aplicar, adaptar, etc);
- Escudo do time (interferências);
- Removedor de impedimentos ;
- Agente de mudanças;
- Apoio ao PO e ao Time de desenvolvimento.

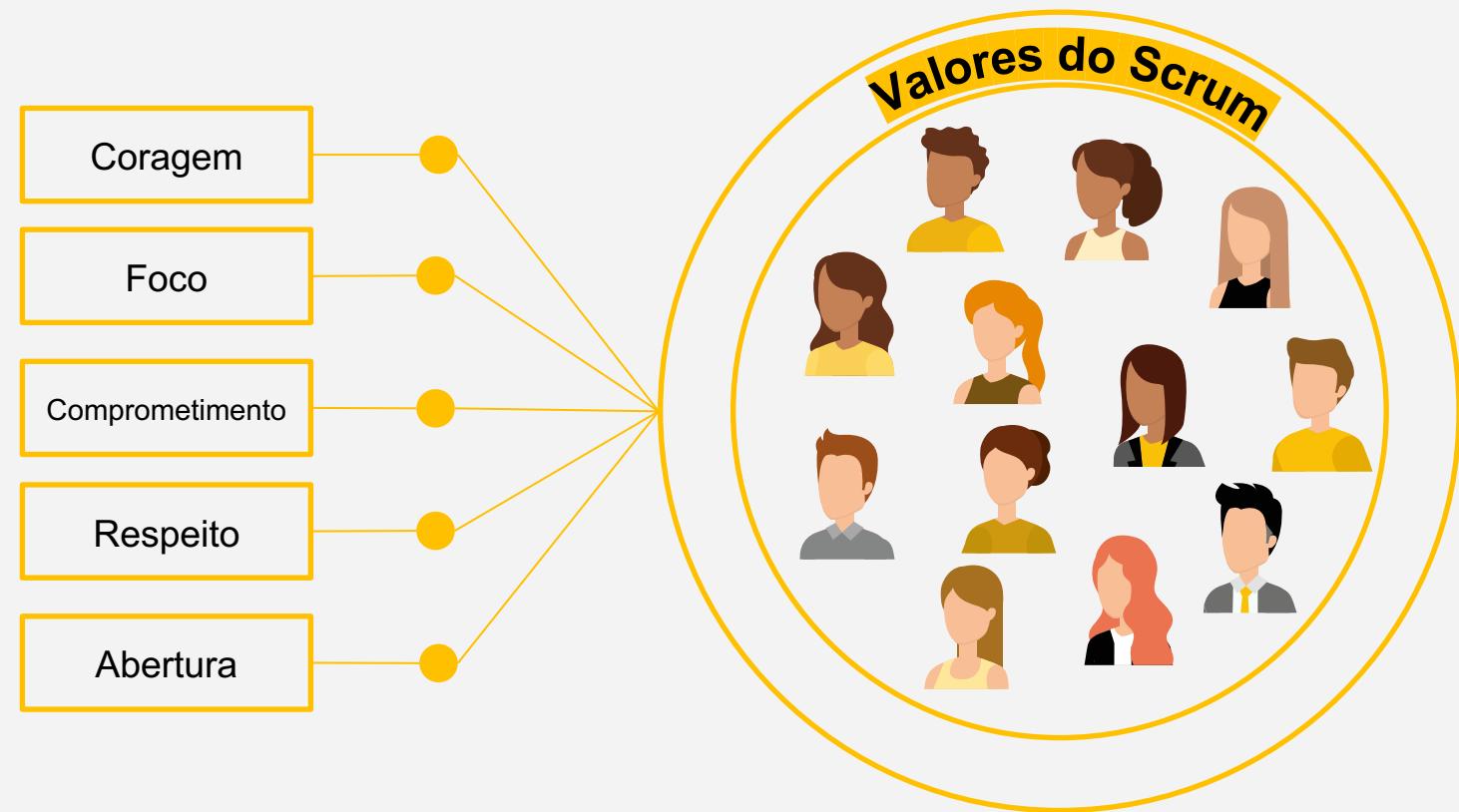
# Development Team (DevTeam)



## Responsáveis pela construção do projeto

- Auto organizáveis (Decide como será feito);
- Multidisciplinar;
- Os integrantes do DevTeam podem ter habilidades especializadas e área de especialização;
- Todos se responsabilizam pela entrega definidos nos critérios de entrega;
- Composição do time de acordo com o tipo de entrega.

# Valores do Scrum





# Product Backlog e Sprint

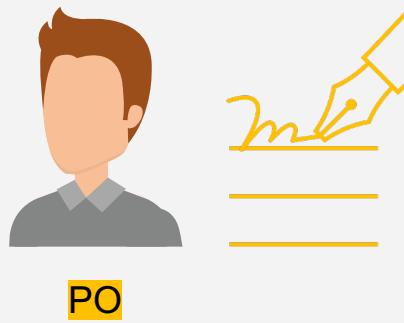
Aula 3

# Product Backlog (PB)

- É a lista de desejos do cliente
- O PB é uma lista ordenada de tudo que é conhecido ser necessário no produto;
- É incremental, ou seja, não precisa estar completo no início de um projeto;
- Lista dinâmica, pode mudar o tempo todo;
- Com o tempo, o Product Backlog cresce e muda à medida que se aprende mais sobre o produto e seus usuários.



# Funcionalidades desejadas



## Product Backlog

Cadastro

Login

Dashboard

Etc.

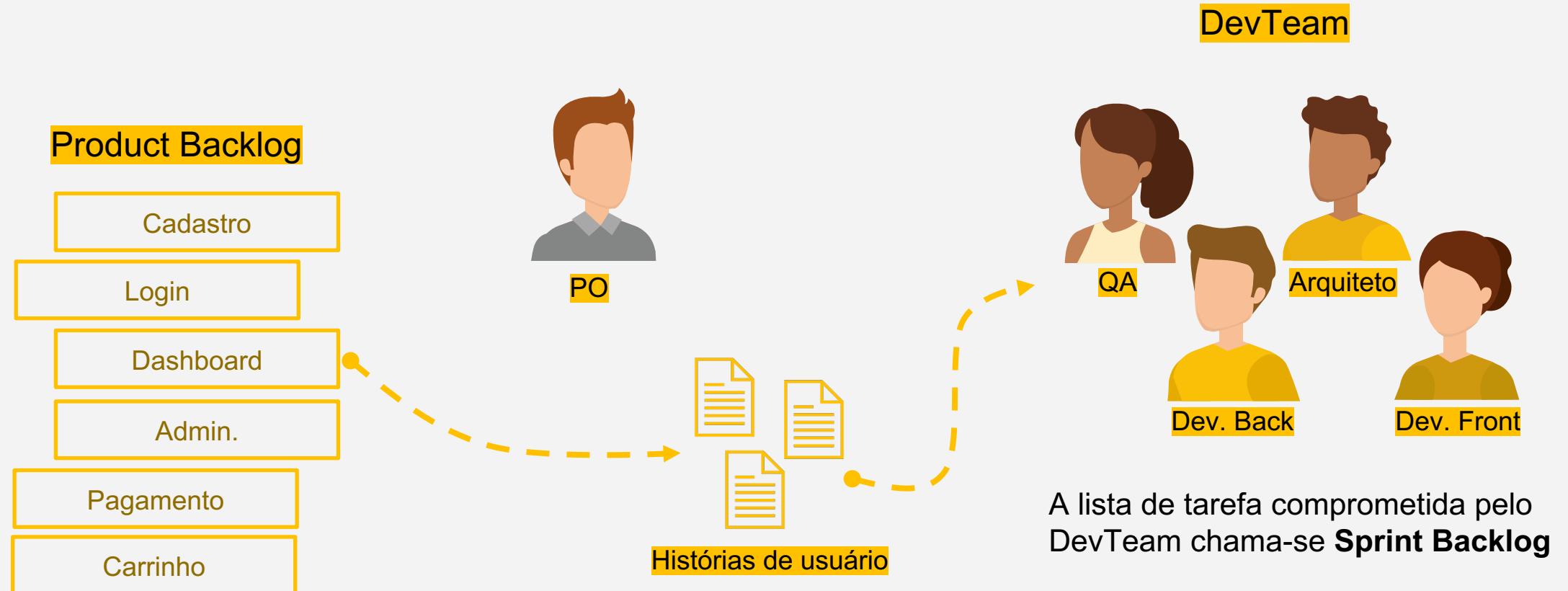
Exemplo: Dashboard

**Como** consumidor

**Quero** uma área no site (Dashboard)

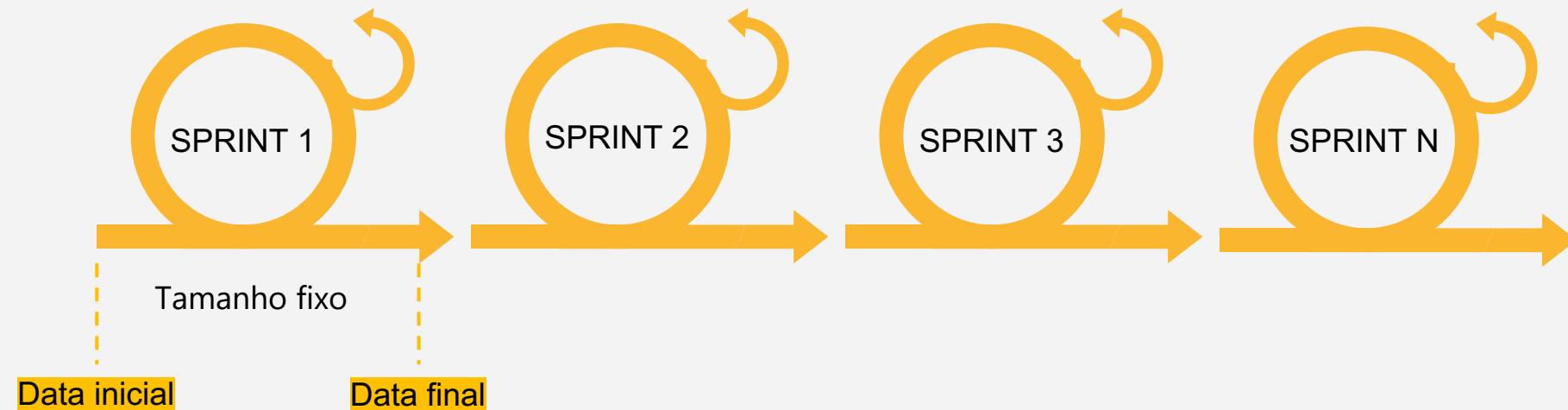
**Para** visualizar meus pedidos

# Priorização



# Sprint

- São etapas com tempo definido que podem variar de 1 até 4 semanas;
- Esse tempo é **time-box**, ou seja, tem data de início e data de término.



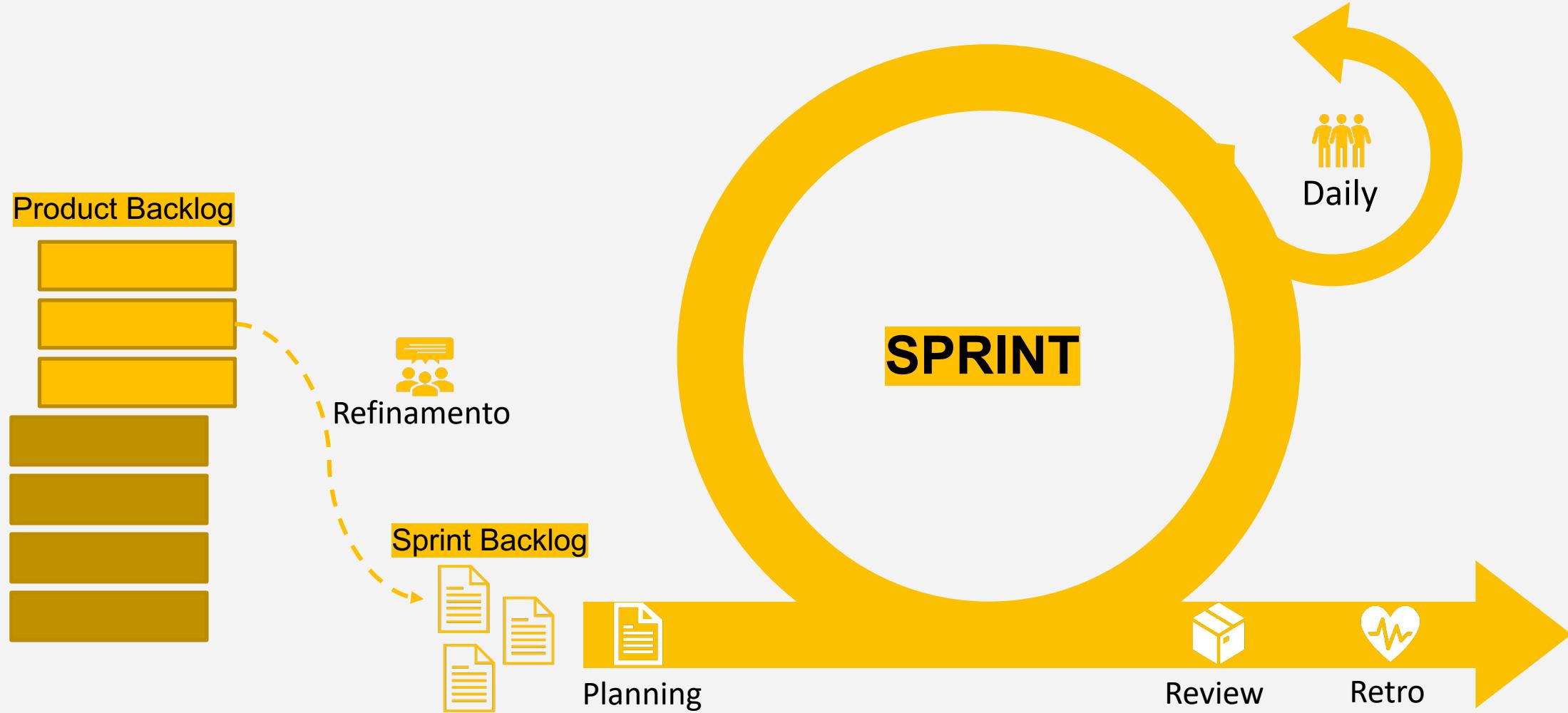
**CURIOSIDADE:** Esse conceito vem do termo **Sprint**, em inglês, que se refere arrancada em uma corrida. O Atleta deve percorrer uma distância curta o mais rápido possível.

# Fatores de **sucesso** da sprint

- Não podem ser feitas mudanças que possam por em perigo o objetivo da Sprint;
- As metas de qualidade não podem diminuir;
- Evitar interferências externas e mudança de escopo durante a sprint;
- O escopo pode ser clarificado e renegociado entre o Product Owner e o Time de Desenvolvimento conforme o aprendizado.



# Eventos do Scrum





# Kanban

Aula 4

# Kanban

- O Kanban começou, basicamente, como uma técnica de gestão de filas para controle de estoques, desenvolvida pela Toyota;
- Um método de Gestão a vista;
- Kanban significa **quadro**, onde de fato é feito um acompanhamento através de um quadro (board).



Toyota - 1960



# Quadro simples

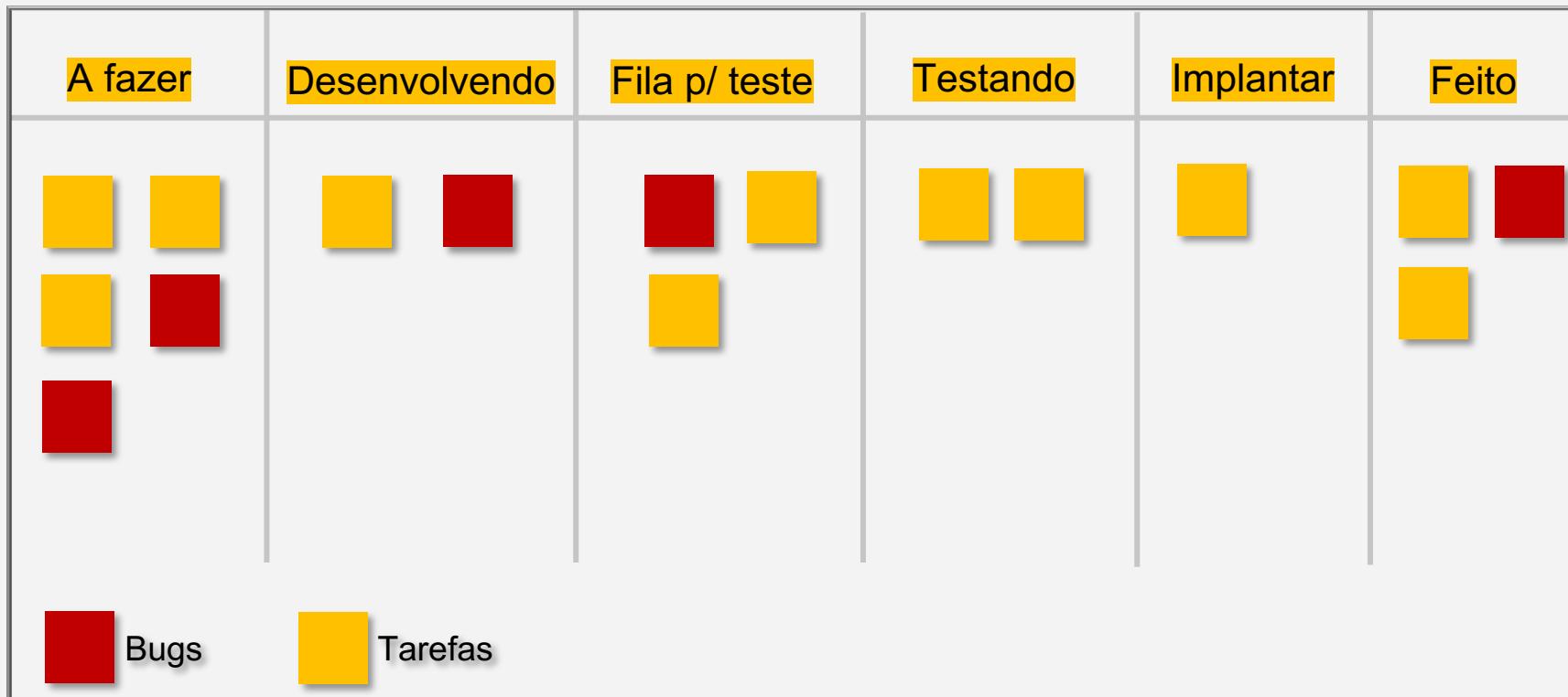
**Alguns quadros são mais simples, têm apenas 3 colunas:**

- Para fazer (To Do)
- Fazendo (In Progress)
- Feito (Done)

Para fazer	Fazendo	Feito
  	 	 

# Quadros customizável

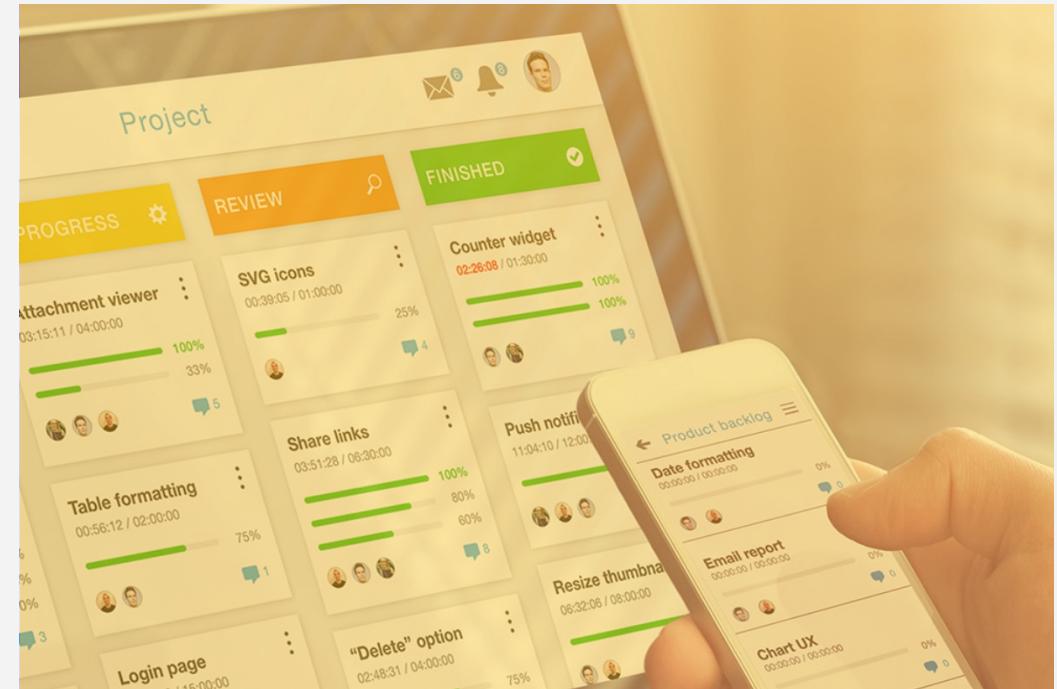
O quadro pode ser customizável em acordo com a necessidade do time:



# Quadros



Físico



Online

# Princípios fundamentais

- O **Kanban** é um dos métodos de desenvolvimento de software mais simples de implementar, tornando-se adaptável a quase todo tipo de cultura.
- O **Kanban** busca a evolução através de melhoria contínua.
- Veja alguns princípios fundamentais do Kankan:

1

Comece com o que você faz agora;

2

Concordar em buscar mudanças evolucionárias;

3

Respeite os papéis, responsabilidades e cargos atuais;

4

Incentive atos de liderança em todos os níveis.



# Práticas



Visualizar o fluxo de trabalho (workflow)



Limitar o trabalho em progresso (WIP)



Gerenciar e medir o fluxo



Tornar as políticas do processo explícitas



Implementar loops de feedback



Melhorar colaborativamente, evoluir experimentalmente

# WIP

- Work in progress (Trabalho em progresso);
- WIP é o número de tarefas que um time possui atualmente;
- Ele mostra a capacidade do fluxo de trabalho do seu time em qualquer momento.

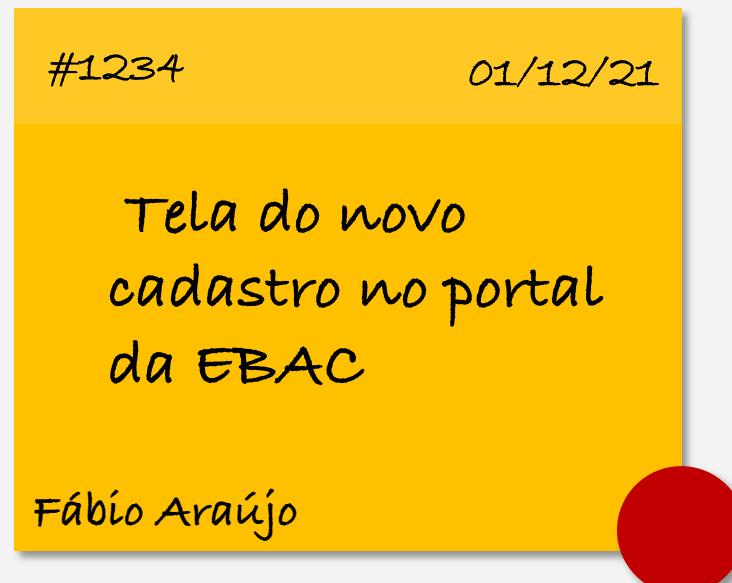


! ALERTA: Coluna cheia

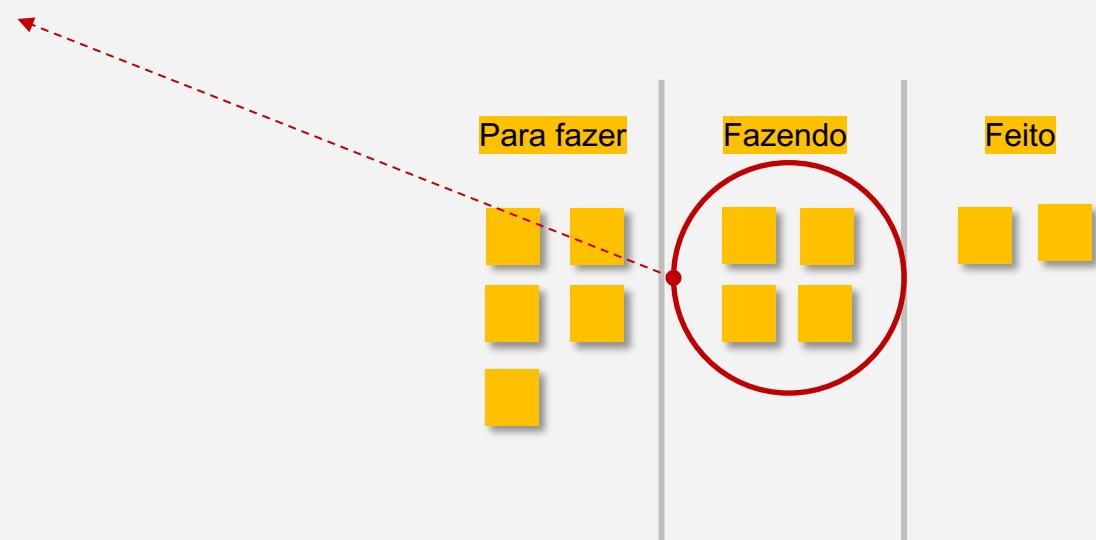


# O cartão

- Representa uma tarefa ou ação que necessita ser executada para que seja realizada a entrega.
- Os cartões podem ser coloridos indicando nível de urgência, quem é o responsável e o tipo de atividade.



# Priorização





# Vantagens

- Acelera e encurta o tempo de ciclo de atividades;
- Melhora a gestão nas mudanças de prioridade;
- Exige menos organização;
- Proporciona que o processo seja simplificado;
- Ajuda a ter mais visibilidade dos projetos;
- Facilita a redução de desperdício e custos;
- Elimina atividades que não agregam valor para a equipe;
- Estimula a autonomia;
- Aumenta a comunicação da equipe;
- Aumenta a produtividade;
- Melhora a motivação e desempenho da equipe.



# **Histórias de usuário e critérios de aceitação**

**Aula 5**

# História de usuário

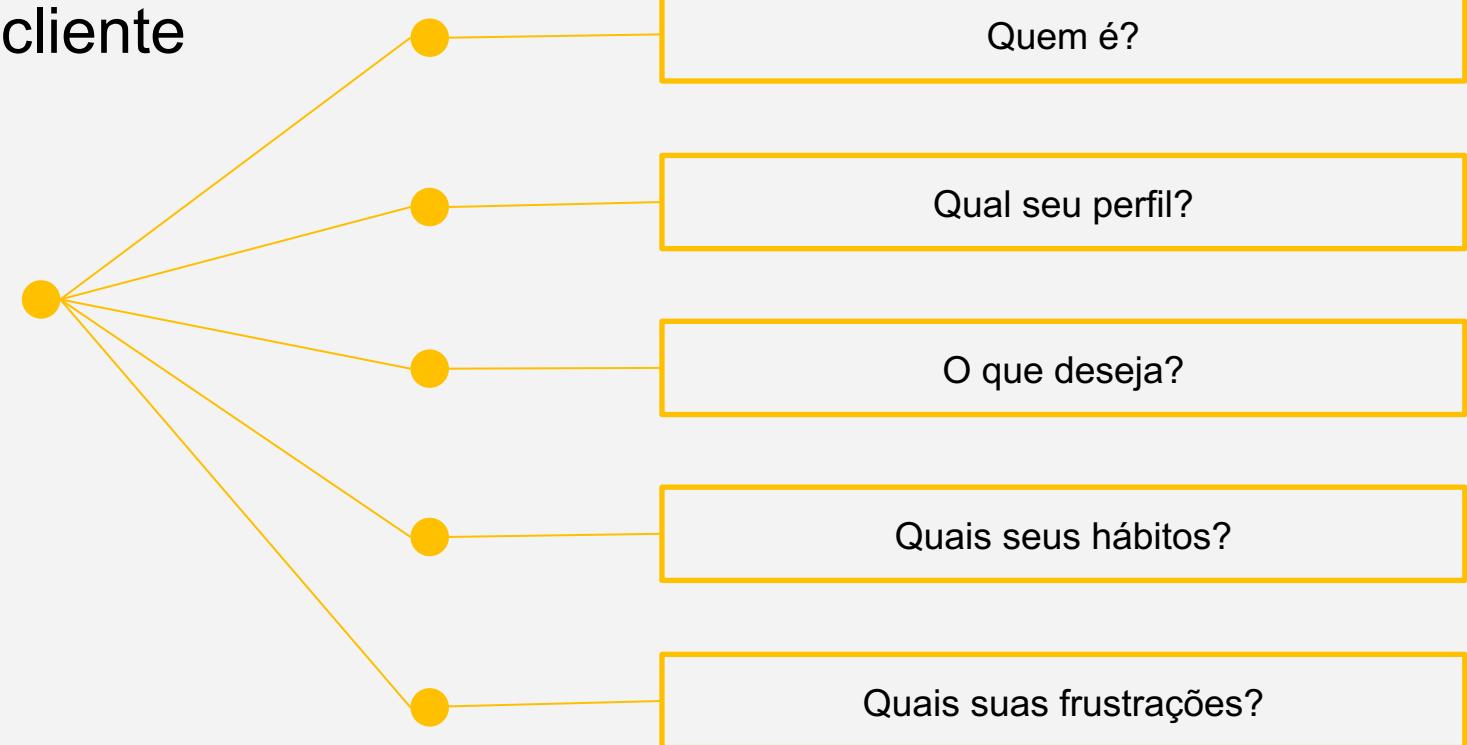
- Uma história do usuário (User Story) é uma explicação informal e geral sobre um recurso de software escrita a partir da perspectiva do usuário final;
- Seu objetivo é articular como um recurso de software pode gerar valor para o cliente;
- Responsável: **Product Owner**.

**História de usuário** não é Requisito funcional

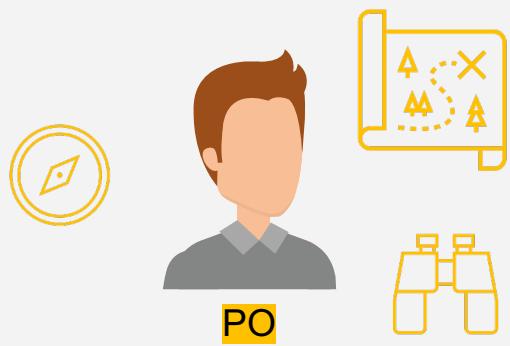


# Onde tudo **começa?**

## Conhecer o cliente

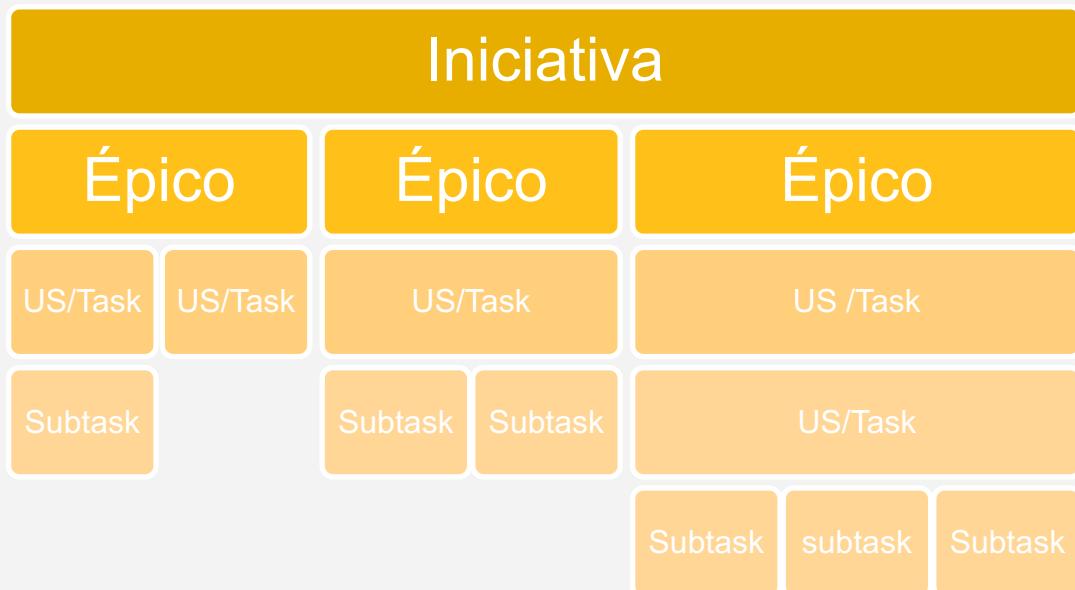


# Descoberta



- Entender o negócio;
- Entender o produto;
- Detalhar e validar o que será feito;
- Quebrar (dividir) as entregas;
- Definir prazo macro;
- Dar visibilidade do backlog;

# Estrutura





# Como é escrito?

Título da História de Usuário

Como cliente

Quero alguma coisa

Para atingir um **objetivo**

-Critérios de aceitação:

-Outras informações para ajudar o time:

## Importante:

- O foco é o cliente, portanto deve-se exercitar a empatia;
- Escrever sob o ponto de vista do cliente;
- Exprimir o negócio e não a solução;
- Deve gerar valor;
- **Critérios de aceitação não é opcional;**
- Sempre pensar no impacto (riscos) se a história não for cumprida.

# Qualidade nas Histórias

Para termos boas histórias usamos um conceito que dá suporte para nivelar a qualidade, vem do acrônimo inglês **INVEST.**



**Independente**  
**(Independent)**  
Deve ser autônoma, não dependem de outras



**Negociável**  
**(Negotiable)**  
Sempre podem ser alteradas e reescritas.



**Valiosa**  
**(Valuable)**  
Deve agregar valor ao produto desenvolvido.



**Estimável**  
**(Estimable)**  
Deve sempre poder estimar o tamanho.



**Pequena**  
**(Small)**  
Deve ser pequena. Facilita as estimativas



**Testável**  
**(Testable)**  
Deve fornecer informações necessárias para os testes.

# Critérios de aceitação

- É um conjunto de declarações, cada uma com um resultado de aprovação/falha, que especifica requisitos funcionais e não funcionais aplicáveis no projeto;
- Esses requisitos representam as "condições de satisfação" do solicitante;
- Não há aceitação parcial: Ou o critério é cumprido ou não é.



“

## Microsoft

*"Condições que um produto de software deve satisfazer para ser aceito por um usuário, cliente ou qualquer outro envolvido".*

## Google

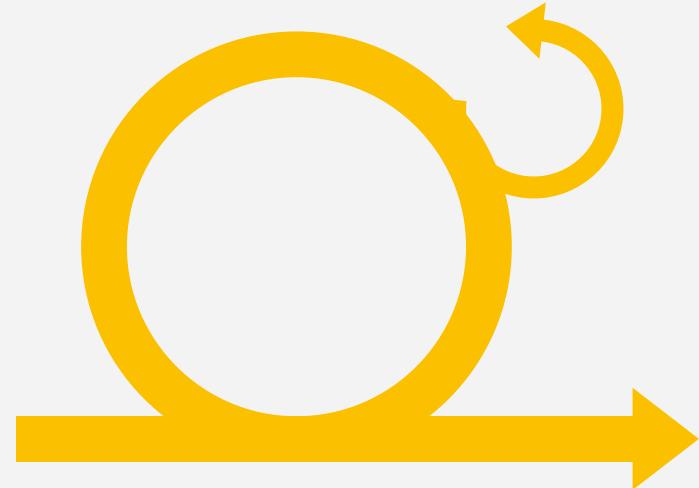
*"Padrões ou requisitos pré-estabelecidos que um produto ou projeto deve atender".*

”

### *Definition of Ready*



### *Desenvolvimento*



### *Definition of Done*

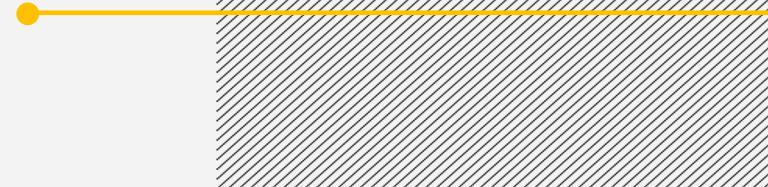


Todas as premissas necessárias para que o time comece o desenvolvimento (preparada).

Todas as premissas necessárias para a história ser considerada como entregue (pronta)



## Diretriz

- 
- Critérios devem ser sucintos;
  - Escritos de forma clara;
  - Um critério por tópico;
  - Não ser ambíguo;
  - Precisam ser demonstráveis;
  - Ser testável;
  - Podem identificar tarefas funcionais e não funcionais;
  - Os critérios de aceitação devem indicar a intenção, mas não uma solução;
  - Podem ser apoiado por um anexo (protótipos, arquitetura, segurança, banco de dados, etc);
  - Podem ser escritos com Gherkin (Dado, Quando, Então).

# Exemplo de História de usuário

ID + Título

**US001- Upload de exercícios**

Desejo do cliente

**Como** aluno da EBAC  
**Quero** quero submeter meus exercícios via upload  
**Para** correção dos tutores

Critérios de aceitação

**Critérios de Aceitação:**

- 1 - Deve permitir upload de arquivos com a extensão .ppt, .doc, .xls, .txt, .png, .jpg
- 2 – Não deve permitir arquivos de vídeo, áudio ou extensões desconhecidas
- 3 – O arquivo deve ter o tamanho máximo de 30mb
- 4- Não deve permitir arquivo com o mesmo nome
- 5 – Só deve permitir um envio por módulo.

# Critérios de aceitação com **Gherkin**

## US001- Upload de exercícios

**Como** aluno da EBAC

**Quero** quero submeter meus exercícios via upload

**Para** correção dos tutores

**Critérios de Aceitação:**

**Cenário:** Formatos de arquivos

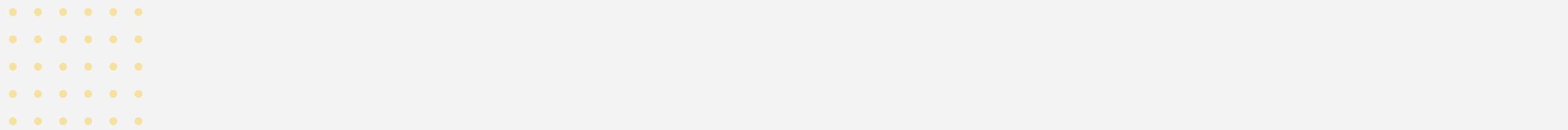
**Dado** que eu tenha os arquivos nos formatos especificados

Ex. .ppt, .doc, .xls, .txt, .png, .jpg

**Quando** eu fizer o upload

**Então** deve aparecer uma mensagem de sucesso

E o arquivo listado na minha área de exercício



## Referencias:

- <https://www.atlassian.com/br/agile/project-management/user-stories>
- <https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/>
- <http://www.desenvolvimentoagil.com.br/>
- <https://cloudscrum.com.br/sprint/>
- <https://www.zendesk.com.br/blog/o-que-e-metodologia-agil/>
- <https://www.youtube.com/watch?v=sGiWWgYyAwQ>
- [https://www.youtube.com/watch?v=ZuEF76Xs\\_Mw](https://www.youtube.com/watch?v=ZuEF76Xs_Mw)
- <https://agilemanifesto.org/iso/ptbr/manifesto.html>
- <https://hakanforss.wordpress.com/>