

# Requisitos de Software

Identificando Histórias de Usuário

# Abordagens de Requisitos

- ▶ IEEE 830
- ▶ Casos de Uso
- ▶ Cenários
- ▶ Features from Feature Drive Development (FDD)
- ▶ User Stories

# Abordagens de Requisitos

- A maior parte das abordagens acreditam que:
  - *“Se escrevermos os requisitos, o cliente receberá exatamente o que ele quer”*
- CUIDADO
  - Na melhor das hipóteses, ele vai obter exatamente o que foi escrito

*"Você construiu o que eu pedi, mas não é o que eu quero."*

Fonte: Mike Cohn

# Abordagens de Requisitos

- ▶ Problemas com a linguagem escrita
  - ▶ As palavras são imprecisas
  - ▶ As palavras têm múltiplos significados





O que é uma  
história de  
usuário !

# Uma história...

*“Uma história é uma descrição de uma pequena funcionalidade que o cliente pretende ver desenvolvida no sistema”*

*“É uma breve declaração de intenções que descreve algo que o sistema precisa fazer para o usuário”*

Fonte: Leffingwell, 2009

## Uma história...

*“A história de usuário captura uma breve declaração de uma função em um cartão, ou talvez em uma ferramenta online”*

### ► Exemplos:

- Entrar no meu portal web para o monitoramento de energia
- Ver o meu uso diário de energia
- Conferir a minha taxa de gasto atual de eletricidade

## Uma história...

*“Detalhes do comportamento do sistema não aparecem na breve declaração da história.*

*São deixados para ser desenvolvido mais tarde por meio de conversas e critérios de aceitação entre a equipe e o proprietário do produto.”*





# User Roles

# User Roles

- ▶ Alguns projetos assumem que existe apenas um tipo de usuário
- ▶ Esta simplificação é uma **falácia** e pode levar a equipe a perder histórias para os usuários que não se encaixam no molde geral do tipo de usuário principal do sistema
- ▶ Todas as histórias são escritas a partir da **perspectiva** de um **tipo de usuário**

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

- ▶ Cabe ressaltar que, cada usuário irá interagir com a solução de software, a partir de suas experiências, seus objetivos, etc
- ▶ Assim, antes de buscar identificar as histórias é importante **identificar os tipos de usuários**

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

- Função do Usuário (*User Role*):
  - *É uma coleção definida de atributos que caracterizam uma população de usuários e suas intenções de interação com o sistema*
  - Exemplo de Funções de Usuário:

Role	Who
Job Seeker	Scott
First Timer	Laura
Layoff Victim	Kindra
Geographic Searcher	Allan
Monitor	Ashish
Job Poster	Mario, Savannah
Resume Reader	Delaney, Savannah

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

- ▶ Naturalmente, no momento em que se estiver fazendo a identificação de funções de usuários, serão identificadas sobreposições
- ▶ Ou seja, funções de usuários que terão **comportamentos e objetivos comuns**

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

## ► Passos para identificar Funções de Usuários

1. *Brainstorm* inicial para identificar o conjunto de funções de usuários
2. Organizar o conjunto inicial
3. Consolidar as funções
4. Refinar as funções

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

## 1) *Brainstorm* inicial para identificar o conjunto de funções de usuários

- Nessa atividade, os participantes escrevem, em um cartão, as funções que eles acreditam que existam
- Assim, um participante de cada vez, escreve o nome da função no cartão e a coloca sobre a mesa, ou colada na parede
- Como estamos em um *brainstorm* não há discussão ou avaliação do que se está escrevendo

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

## 2) Organizando o conjunto inicial para identificar de funções de usuários

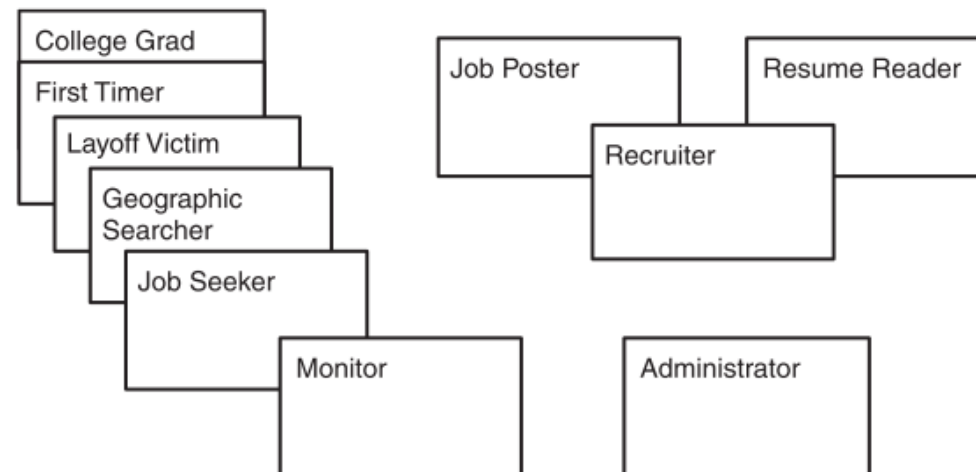
- Aqui, deve-se agrupar as funções indicando a relação entre eles
- Caso um ou mais funções identificadas se sobreponham totalmente, coloque-os um sobre o outro, totalmente
- Caso um ou mais funções identificadas se sobreponham parcialmente, coloque-os um sobre o outro, parcialmente

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn



# User Roles

## 2) Organizando o conjunto inicial para identificar de funções de usuários



Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

## 3) Consolidando as funções de usuários

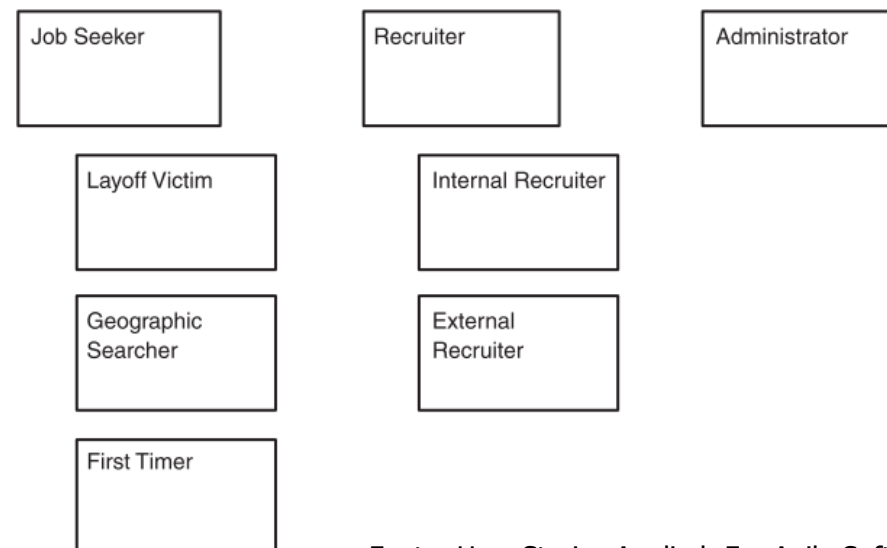
- ▶ Depois que as funções foram agrupadas, agora deve-se consolidá-las e condensá-las
- ▶ Comece com cartões que estão totalmente sobrepostas totalmente. Os autores dos cartões que se sobrepõem descrevem o que eles queriam dizer com esses nomes de funções. Depois de uma breve DISCUSSÃO o grupo decide se os papéis são equivalentes

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

## 3) Consolidando as funções de usuários

- Se equivalente, os papéis podem ser consolidados em um único papel, ou um dos cartões iniciais papel pode ser retirado



Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

## 4) Refinando Funções

- Uma vez que se tem as funções consolidadas e um entendimento básico de como referem-se umas as outras, é possível modelar as funções os papéis definindo atributos
- Um atributo de uma função é um fato ou informação útil sobre os usuários que preenchem a função

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

## 4) Refinando Funções

- Estrutura do Cartão da Função (descrição)
  - A frequência com que o usuário usa o software
  - O nível de conhecimento do usuário com o domínio
  - Nível geral do usuário de proficiência com computadores e software
  - Nível de proficiência com o software que está sendo desenvolvido para o usuário
  - Objetivo geral para o uso do software do usuário. Alguns usuários estão atrás de conveniência

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

# User Roles

## 4) Refinando Funções

- Cartão da Função de Usuário

Função do Usuário: Recrutador Interno

Descrição: Não usa o computador diariamente, mas muito habituada a usar a web. Vai usar o software raramente, mas intensamente. Vai ler anúncios de outras empresas para descobrir como melhorar seus anúncios. Facilidade de uso é importante, mas mais importante do que ela é a facilidade de se lembrar como utilizá-lo meses depois.

Fonte: User Stories Applied: For Agile Software Development, by Mike Cohn

The background features abstract geometric shapes in two shades of green. On the left, a light green triangle points upwards. On the right, a dark green triangle points downwards. A thin, light green line runs diagonally across the right side, intersecting the dark green triangle. The text is centered in the white space between these elements.

# Construindo (Formando) uma História de Usuário

# Forma de História de Usuário

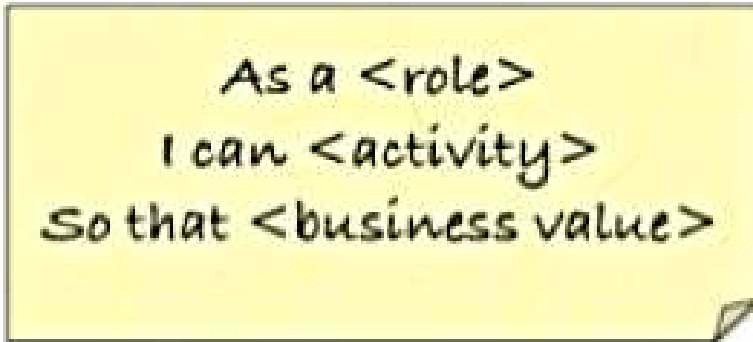
- Ron Jeffries, um dos criadores do XP, descreveu o que se tornou a forma mais usual de pensar em histórias de usuários
- A estrutura: **Cartão**, **Conversa**, e **Confirmação** para descrever os três elementos de uma história de usuário



# Forma de História de Usuário

## ► Cartões

- Representa 2-3 frases usadas para descrever a intenção da história
- O cartão serve como um símbolo que memoriza (registra) o resume de uma a intenção, a qual ainda não foi detalhada



As a <role>  
I can <activity>  
So that <business value>

# Forma de História de Usuário

- **Cartões** (*User Story Voice*)

*As a* <role> *I can* <activity> *so that* <business value>

- *Role*

- Representa quem está realizando a ação ou, talvez, aquele que está recebendo o valor da atividade.
- Pode até ser um outro sistema, se é isso que está iniciando a atividade.

# Forma de História de Usuário

- **Cartões** (*User Story Voice*)
  - *Activity*
    - Representa a ação que será executada pelo sistema
  - *Business value*
    - Representa o valor para o negócio

# Forma de História de Usuário

- **Cartões** (*User Story Voice*)

- Essa forma é chamada de “**voz do usuário**” e representa a expressão história, a qual é uma construção extremamente útil, pois abrange o espaço do **problema** (*<business value> entregue*) e o espaço de **solução** (*<activity> o usuário executa com o sistema*).
- Também fornece uma primeira perspectiva do usuário (*<role>*) para a equipe do projeto, o que os mantém focados em **valor de negócios** e **resolução de problemas reais para pessoas reais**

# Forma de História de Usuário

- **Cartões** (*User Story Voice*)

- Esta forma de história de usuário contribui com a compreensão dos “porquês” e “como”, a qual os desenvolvedores precisam para implementar um sistema que, realmente, atenda às necessidades dos usuários

*"Como consumidor, (<role>) eu quero ser capaz de ver o meu uso diário de energia (<que eu faço com o sistema>) para que eu possa começar a entender como reduzir meus custos ao longo do tempo (<valor comercial recebido>)"*

# Forma de História de Usuário

## ► Cartões

- Em XP e ágil, as histórias são muitas vezes escritas manualmente em fichas físicas
- O elemento "card" é capturado como texto e anexos em uma planilha ou ferramenta ágil de gerenciamento de projetos,
- Usualmente, as histórias são utilizadas pelas equipes, também, para o planejamento e realização de brainstorming

# Forma de História de Usuário

## ► Cartões

Como um [ator] eu quero/preciso de  
/devo/gostaria de [ação] para  
[funcionalidade].

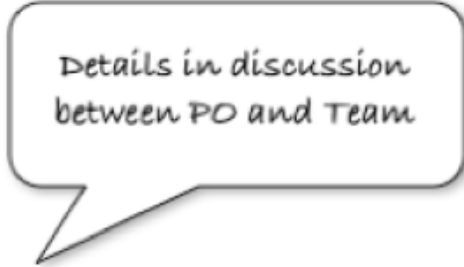
Como um cliente eu quero ver os filmes  
disponíveis para locação para que eu  
possa agendar uma reserva na data  
X.

Como um vendedor responsável pelo  
setor de livros eu quero procurar por  
livros filtrando por nome para que seja  
possível verificar se o livro X está  
disponível para pronta entrega.

# Forma de História de Usuário

## ► Conversação

- Representa (registra) uma **conversa** entre a equipe, cliente, proprietário do produto (*product owner*), e de outras partes interessadas, a qual é necessária para determinar o comportamento mais detalhado e necessário para implementar a história
- Em outras palavras, o cartão também representa uma “**conversa de promessa**” sobre a intenção a ser construída



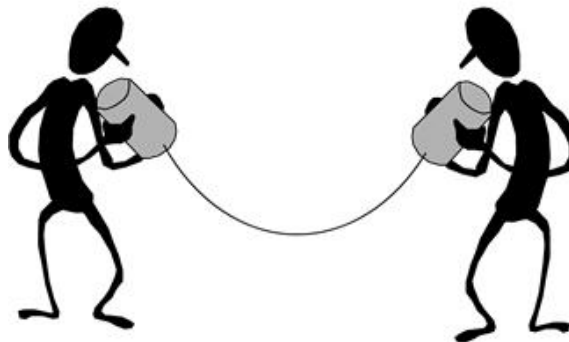
Details in discussion  
between PO and Team



# Forma de História de Usuário

## ► Conversação

- As conversas devem ser realizadas desde o início do projeto para que a equipe seja envolvida
- Caso sejam necessários mais detalhes sobre a história, isso pode ser fornecida na forma de um anexo (mockup, planilha, algoritmo, ou qualquer outro)



# Forma de História de Usuário

## ► Conversação

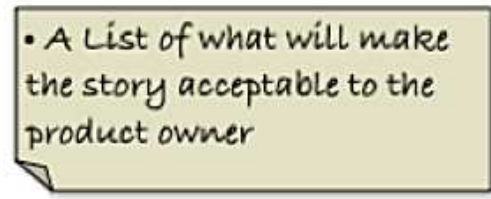
- Assim, a história serve como o "sinal", que também transporta o comportamento mais específico para a equipe
- Detalhes adicionais da história devem ser coletados ao longo do tempo (*just-in-time*) por meio de discussões e colaboração com a equipe e outras partes interessadas antes e durante o desenvolvimento



# Forma de História de Usuário

## ► Confirmação

- Representa o **teste de aceitação**, que é com o cliente ou proprietário do produto (*product owner*) irá confirmar se a história foi implementada satisfatoriamente
- Em outras palavras, a Confirmação representa as **condições de satisfação** que devem ser utilizadas para determinar se a história cumpre ou não, a intenção, bem como os requisitos mais detalhados



• A List of what will make the story acceptable to the product owner

# Forma de História de Usuário

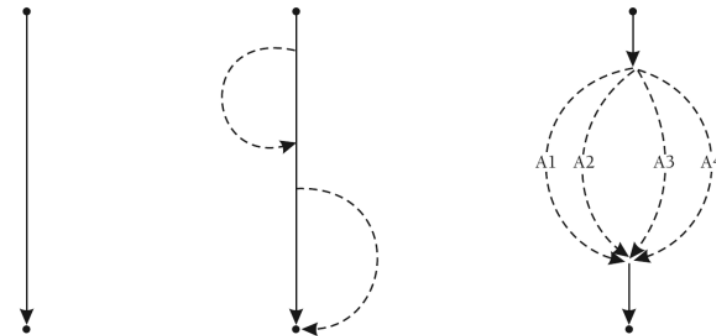
## ► Confirmação

- Notas adicionais, premissas e critérios de aceitação podem ser mantidas anexo a história de usuário
- Muitas discussões sobre uma história entre a equipe e os clientes provavelmente ocorrerá antes e durante o tempo da história está empenhada em código

# Forma de História de Usuário

- **Confirmação**

- Os fluxos alternativos da atividade, os limites de aceitação, e outros esclarecimentos devem ser capturados junto com a história
- Muitos deles podem ser transformados em casos de teste de aceitação, ou outros casos de testes funcionais, para a história



# Forma de História de Usuário

## ► **Confirmação** (Exemplo)

- Como consumidor, eu quero ser capaz de ver o meu uso diário de energia para que eu possa começar a entender como reduzir meus custos ao longo do tempo

### ► **Critérios de aceitação:**

- *Ler dados em decawatts a cada 10 segundos. Exibi-los no portal, em incrementos de 15 minutos e, exibir no display de casa, a cada leitura*
- *Leia quilowatt de novos dados como disponível e exibir no portal a cada hora e no display de casa depois de cada leitura*
- *Não apresenta uma tendência de consumo diário (outra história)*

# Forma de História de Usuário

## ► Confirmação

### ► IMPORTANTE !!!

- Critérios de aceitação não são testes funcionais ou de unidade
- Eles são as condições de satisfação a serem colocados no sistema
- Testes funcionais e de unidade vão muito mais além no teste de todos os fluxos funcionais, fluxos de exceção, condições de fronteira e relatórios associados com a história

# Forma de História de Usuário (Exemplo)

## Cartão

*“Como administrador eu quero cadastrar um jogo para que os apostadores possam fazer seus palpites de resultado”*

## Conversação

*O administrador pode cadastrar o jogo quando quiser? E se ele cadastrar muito em cima?*

- Ah, eu acho que ele tem que cadastrar com no mínimo 48h de antecedência*
- Legal... concordo*

## Confirmação

- Um administrador não poderá cadastrar um jogo com menos de 48h de antecedência*
- O jogo deve pertencer ao campeonato corrente*
- Um administrador não poderá cadastrar dois jogos envolvendo os mesmos times no mesmo horário*





# Casos de Uso x Histórias de Usuários

# Diferenças entre Histórias de Usuários e Casos de Uso

## ► Escopo

- Uso caso é quase sempre muito maior
- A história é semelhante a um cenário de (ou caminho através de) um caso de uso

## ► Nível de Completude

- Os testes de aceitação das histórias de usuário são basicamente a mesma coisa que os descritos em um caso de uso (James Grenning)

# Diferenças entre Histórias de Usuários e Casos de Uso

## ► Longevidade

- Os casos de uso são artefatos permanentes
- Cartões de história são rasgados

## ► Propósito

- Os casos de uso
  - Documenta o acordo entre o cliente e desenvolvedores
- Histórias
  - Escrito para facilitar a liberação e iteração planejamento
  - Espaços reservados para conversas futuras

# Diferenças entre Histórias de Usuários e Casos de Uso

- ▶ **Vantagens** na utilização de **Casos de Uso**
  - ▶ Apresenta claramente o comportamento do sistema
  - ▶ É utilizado para obter um bom controle do tamanho do sistema a ser desenvolvido e diminuir a o risco de condições inesperadas ocorrerem
  - ▶ Registra os resultados das discussões
  - ▶ Apresenta uma visão abrangente do comportamento em torno de uma solicitação de sistema, tornando mais fácil para as pessoas verem o que eles pedem

Fonte: Requirements Engineering in an Agile Environment, Yunyun Zhu, 2009

# Diferenças entre Histórias de Usuários e Casos de Uso

- ▶ **Limitações** na utilização de **Casos de Uso**
  - ▶ Difícil de implementar todo o caso de uso em uma única iteração
  - ▶ Não apropriado para ser utilizado em listas de erros e listas de melhoria, uma vez que um caso de uso não fornece um propósito real para isso

Fonte: Requirements Engineering in an Agile Environment, Yunyun Zhu, 2009

# Diferenças entre Histórias de Usuários e Casos de Uso

- ▶ **Vantagens** na utilização de Histórias de Usuário
  - ▶ São curtas
  - ▶ Pequenas e, portanto, mais fácil de serem estimadas, priorizados e gerenciados
  - ▶ Podem ser divididas o quanto se quer, o que significa que podem caber em qualquer tamanho de iteração

Fonte: Requirements Engineering in an Agile Environment, Yunyun Zhu, 2009

# Diferenças entre Histórias de Usuários e Casos de Uso

- ▶ **Limitações** na utilização de Histórias de Usuário
  - ▶ Pelo fato de serem descrições extremamente curtas de uma simples funcionalidade a ser implementada, uma das principais limitações refere-se a **dificuldade em cobrir grandes projetos só com histórias de usuários**

Fonte: Requirements Engineering in an Agile Environment, Yunyun Zhu, 2009

# Diferenças entre Histórias de Usuários e Casos de Uso

	Use Case	User Story
<b>Definição</b>	São descrições não técnicas da interação do usuário com o sistema, as ações que o usuário toma e o sistema responde até que o objetivo seja alcançado.	São descrições não técnicas, simples e curtas daquilo que o sistema terá que fazer.
<b>Qual a sua utilização</b>	Servem como um protótipo para que todos saibam o que esperar do sistema.	Servem para que a equipe possa estimar o que tem que fazer e quanto esforço será necessário.
<b>Quem cria o artefato</b>	Cliente ou Analista	Cliente
<b>Tempo de vida dos requisitos</b>	Os requisitos que dão origem ao use case são mantidos até o final do projeto.	Os requisitos utilizados numa user story são excluídos do projeto depois do teste de aceitação.
<b>Como tratam os requisitos</b>	Use Cases são exemplos funcionais dos requisitos e são tarefas “macro” do sistema.	User Stories são levantadas diretamente dos requisitos, são tarefas tão “micro” quanto necessário.

Fonte: Gerenciamento de Requisitos: Um Comparativo entre Metodologias Tradicionais e Ágeis sob a ótica dos Modelos de Qualidade, José André Dorigan, 2010



# Diferenças entre Histórias de Usuários e Casos de Uso

	Use Case	User Story
Pode ser usado como protótipo?	✓	✗
Pode ser estimado?	✓	✓
Pode ser utilizado para gerenciar o projeto?	✓	✓
E um documento técnico?	✗	✗
Direciona os testes?	✓	✓
Pode-se estimar o esforço necessário?	✓	✓
Gera documentação de requisitos?	✓	✗
O cliente é essencial na criação do artefato?	✗	✓
Utiliza formalidade em sua descrição?	✓	✗
Pode ser subdividido?	✗	✓

Fonte: Gerenciamento de Requisitos: Um Comparativo entre Metodologias Tradicionais e Ágeis sob a ótica dos Modelos de Qualidade, José André Dorigan, 2010

## Porque Histórias de Usuários ?



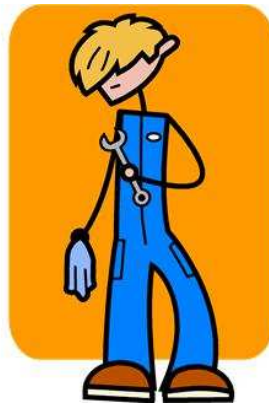
## Porque História de Usuário?

1. *Tiram o foco da escrita e vão para a fala*
2. *São facilmente entendidas por desenvolvedores e clientes*
3. *Apoiam e incentivam o desenvolvimento iterativo*
4. *São do tamanho certo para o planejamento*

## Porque História de Usuário?

- 5. *Apoiam a elaboração participativa*
- 6. *Enfatizam os objetivos do usuário, não os atributos do sistema*
- 7. *Pode ser entendida e priorizada pelo cliente*
- 8. *São basicamente, pequenos pedaços de funcionalidades*

## Exemplo Prático



# Exemplo

A empresa A Video Store (AVS) gerencia várias locadoras de vídeo bastante comuns. Antes que um vídeo possa ser colocado na prateleira, ele deve ser catalogado e inserido no bando de dados de vídeos. Todo cliente precisa ter um cartão de cliente válido da AVS para alugar um vídeo. Os clientes podem alugar vídeos por três dias a cada vez. Sempre que um cliente alugar um vídeo, o sistema terá de verificar se ele não tem nenhum vídeo pendente. Se isso ocorrer, os vídeos pendentes terão de ser devolvidos e uma taxa extra deverá ser paga antes que o cliente possa alugar mais vídeos. Da mesma forma, se o cliente tiver devolvido os vídeos pendentes sem ter pago a taxa extra, essa terá de ser quitada antes que novos vídeos possam ser alugados. Toda manhã, o gerente da locadora imprime um relatório que lista vídeos pendentes; se um vídeo estiver dois ou mais dias atrasado, o gerente entrará em contato com o cliente para lembrá-lo da devolução do vídeo. Se um vídeo for devolvido danificado, o gerente o removerá do banco de dados de vídeos e, em algumas situações, poderá cobrar do cliente.

Fonte: Kamila Holanda Crozara

# Exemplo

- US01 – Como gerente eu quero manter o cadastro de vídeos da locadora para que seja possível catalogar os vídeos existentes.
- US02 – Como gerente eu quero manter o cadastro de clientes da locadora para que as informações a respeito do cliente estejam disponíveis quando necessário.
- US03 – Como gerente eu quero imprimir um relatório dos vídeos pendentes para poder lembrar os clientes da devolução dos vídeos.
- US04 – Como cliente eu quero ter um cartão que me identifique para que eu possa alugar os vídeos.
- US05 – Como funcionário eu quero manter o controle dos vídeos alugados para que seja possível cobrar taxas em caso da devolução ocorrer com atraso.

Fonte: Kamila Holanda Crozara

# Exemplo

## US01

Como gerente eu quero manter o cadastro de vídeos da locadora para que seja possível catalogar os vídeos existentes

## Critérios para aceitação

- Cada vídeo cadastrado deve ter um ID associado
- Para cada vídeo deve ser possível cadastrar o nome, o ano, o gênero, a categoria e a quantidade existente

## US02

Como gerente eu quero manter o cadastro de clientes da locadora para que as informações a respeito do cliente estejam disponíveis quando necessário.

## Critérios para aceitação

- Cada cliente deve ter um cartão identificador associado
- Para cada cliente deve ser possível cadastrar nome, telefone, endereço, CPF e RG

Fonte: Kamila Holanda Crozara



# Exemplo

## US03

Como gerente eu quero imprimir um relatório dos vídeos pendentes para poder lembrar os clientes da data de devolução dos vídeos

## Critérios para aceitação

- Os vídeos com dois ou mais dias de atraso devem ser destacados

## US04

Como cliente eu quero ter um cartão que me identifique para que eu possa alugar os vídeos

## Critérios para aceitação

- Cada cartão deve ter um identificador único

Fonte: Kamila Holanda Crozara

# Exemplo

## US05

Como funcionário eu quero manter o controle dos vídeos alugados para que seja possível cobrar taxas em caso da devolução ocorrer com atraso

## Crítérios para aceitação

- Cada cliente pode alugar no máximo cinco vídeos por vez
- Os vídeos podem ser alugados por no máximo três dias
- O cliente só poderá alugar vídeos se não tiver vídeos pendentes para devolução
- Caso o cliente tenha vídeos pendentes ele deverá pagar uma extra antes de poder alugar mais vídeos
- O cliente só poderá alugar vídeos se não tiver débito pendente

Fonte: Kamila Holanda Crozara

# Requisitos de Software

Identificando Histórias de Usuário

