



## API – APRENDIZAGEM POR PROJETOS INTEGRADOS

Revisão: 2025-2

### GUIA DE ARTEFATOS DE REQUISITOS – MÉTODOS ÁGEIS

#### Sumário

1.	User Stories .....	2
1.1	O que constitui uma boa User Story (ou história de usuário)? .....	2
1.2	Critérios de Aceitação .....	4
2.	Backlog .....	5
2.1.	Product Backlog .....	5
2.2.	Sprint Backlog .....	6
3.	DoR – Definition of Ready .....	8
4.	Alocação de Tarefas, com base no DoR .....	10

## 1. User Stories

Uma *User Story* (ou história de usuário) é uma sentença curta e simples sobre uma funcionalidade desejada, escrita sob a perspectiva do cliente, cuja solução a ser concebida agrega valor ao seu negócio. Ela deve seguir o formato “**Como** <tipo de usuário>, **quero** <funcionalidade desejada> **para** <algum motivo que remeta valor para o negócio>”, onde:

- <tipo de usuário> representa um tipo de usuário real da aplicação (um *stakeholder*, uma persona mapeada ou um grupo de usuários). A *User Story* representa um desejo desse tipo de usuário. Para melhor entendimento das *User Stories*, é recomendado mapear, em seção a parte, cada um dos tipos de usuários envolvidos. **Importante:** Deve-se evitar a adoção de um tipo de usuário chamado “usuário”, visto que existem referências específicas que podem ser mais bem aplicadas. Exemplos de tipos de usuários: gerente, caixa, administrador de sistema, entre outros;
- <funcionalidade desejada> representa o desejo do tipo de usuário. Algo que ele precisa para solucionar um de seus problemas. O desejo descrito deve utilizar a linguagem cotidiana do usuário e não da Equipe de desenvolvimento. Exemplo: emitir notas fiscais; e
- <algum motivo que remeta valor para o negócio> representa o ganho (financeiro ou não) que a funcionalidade trará para o tipo de usuário. Exemplos de valor de negócio: reduzir o tempo necessário; eliminar redundâncias.

### 1.1 O que constitui uma boa User Story (ou história de usuário)?

As *User Stories* são compostas de três aspectos que Ron Jeffries nomeou em 2001 com a maravilhosa aliteração de cartão, conversa e confirmação (3 Cs):

1. **Cartão** - Descrição escrita da história, usada para planejamento e como lembrete
2. **Conversa** - Conversas sobre a história que servem para dar corpo aos detalhes da história
3. **Confirmação** – Testes que documentam detalhes que podem ser usados para determinar quando uma história está completa

Além disso, boas *User Stories* devem seguir os critérios INVEST (*Independent, Negotiable, Valuable, Estimable, Small, and Testable*):

- **Independent** – Uma *User Story* não pode depender de outra;
- **Negotiable** – O detalhamento de uma *User Story* deve ser realizado em conjunto pelo cliente e a Equipe de desenvolvimento;
- **Valuable** – O resultado de uma *User Story* deve apresentar valor real para o negócio do cliente;
- **Estimable** – Deve ser possível estimar, grosso modo, o tempo necessário para concluir uma *User Story*;
- **Small** – Uma *User Story* deve caber em um Sprint;

- **Testable** – Deve ser possível aferir se uma *User Story* foi finalizada por meio de testes.

Exemplos de boas *User Stories*, considerando o contexto de algumas funcionalidades desejadas em uma versão inicial do site de treinamentos da Scrum Agile Courses.

- **Como** membro do site, **quero** preencher um formulário **para** me tornar um Certified Scrum Trainer, poder ministrar cursos Certified Scrum Master (CSM) e Certified Scrum Product Owner (CSPO), e certificar outras pessoas
- **Como** treinador, **quero** que no meu perfil sejam listadas minhas próximas aulas e inclua um link para uma página detalhada sobre cada uma, **para** que os possíveis participantes possam encontrar meus cursos
- **Como** visitante do site, **quero** acessar notícias antigas que não estão mais na página inicial, **para** poder acessar coisas que me lembro do passado ou que outras pessoas mencionaram
- **Como** visitante do site, **quero** ver uma lista de todos os próximos “Cursos de Certificação”, **para** que poder escolher o melhor curso para mim

#### Referência:

- Wake, B. **INVEST in Good Stories, and SMART Tasks**. 2003. Acessado em: 20/09/2023. URL: <https://xp123.com/articles/invest-in-good-stories-and-smart-tasks/>
- Cohn, M. **User Stories**. Acessado em: 24/09/2023. URL: <https://www.mountangoatsoftware.com/agile/user-stories>

## 1.2 Critérios de Aceitação

Além da frase descrita anteriormente, uma *User Story* necessita de detalhamento para viabilizar seu desenvolvimento (Confirmação dos 3 Cs). Por esse motivo, é necessário enriquecê-la com uma lista de Critérios de Aceitação ou Condições de Satisfação que precisam ser alcançados para que ela seja considerada finalizada.

Cabe ao *Product Owner* (PO) a responsabilidade de garantir que haja um *backlog* de produto priorizado composto por *User Stories* detalhadas e aptas para serem desenvolvidas em uma *Sprint* antes que se inicie, contudo, qualquer membro da Equipe pode contribuir com a escrita das *User Stories*.

Uma *User Story* está preparada para o seu desenvolvimento, somente se a etapa de refinamento e entendimento coletivo da Equipe ocorrer e houver insumos suficientes no seu detalhamento para sua implementação e confirmação (*Definition of Ready ou DoR*).

Além disso, é extremamente importante que se encontrem detalhados todos os critérios assumidos pela Equipe, em comum comprometimento, que devem ser satisfeitos para que a *User Story* seja entregue (*Definition of Done ou DoD*), tal como: i) código devidamente versionado no Git; ii) abertura de PR antes da liberação para Code Review; iii) Remoção de fragmentos de código comentados; iv) implementação de testes de unidade com cobertura de código mínima de 70%; v) criação de *feature* ou *fix branch*; entre outros.

Bons critérios de aceitação auxiliam a Equipe a delimitar melhor o escopo de uma *User Story*, descobrir quando o desenvolvimento atenderá as regras de confirmação e a planejar os cenários de teste.

Exemplos de bons Critérios de Aceitação. Considerando a seguinte *User Story* “**Como** vice-presidente de marketing, **quero** selecionar um período de festas para ser usado na análise do desempenho de campanhas publicitárias anteriores, **para** poder identificar as mais lucrativas”, tem-se como critérios:

- Certifique-se de que funciona nos principais feriados do varejo: Natal, Páscoa, Dia das Mães, Dia dos Pais, Dia do Trabalho, Dia de Ano Novo;
- Certifique-se de suportar feriados que abrangem dois anos civis
- Certifique-se de validar/bloquear períodos que abrangem três anos civis;
- Os períodos de férias podem ser definidos de um feriado para o outro (como o Dia de Ação de Graças e o Natal); e
- As temporadas de feriados podem ser definidas para vários dias antes do feriado.

### Referência:

- Dias, K. **User Stories e Critérios de Aceitação**. 2018. Acessado em: 20/09/2023. URL: <https://medium.com/@karladiasn/user-stories-e-crit%C3%A9rios-de-aceita%C3%A7%C3%A3o-317c48403fcd>
- Cohn, M. **User Stories**. Acessado em: 24/09/2023. URL: <https://www.mountaingoatsoftware.com/agile/user-stories>

## 2. Backlog

### 2.1. Product Backlog

Um *Product Backlog* consiste numa lista priorizada de todas as *User Stories* do projeto. No topo devem estar todas as *User Stories* com maior valor agregado (*Return of Investment* – ROI) ao negócio. Não devem existir duas ou mais *User Stories* com mesmo *ranking*, visto que elas devem ser trabalhadas em ordem.

A Tabela 1 apresenta um exemplo de *Product Backlog*. Deve ser notado que as *User Stories* de maior importância ao ator principal devem fazer parte das primeiras *Sprints*. No exemplo, o fluxo principal de abertura de solicitações de vistoria e gerenciamento de solicitações estão no topo. Detalhes como gerenciamento de técnicos, configurações, entre outras funcionalidades, serão deixados para depois. Isso vai contra o pensamento geral dos desenvolvedores, mas faz sentido, visto que permite o uso do sistema imediatamente, mesmo que muitos procedimentos sigam ocorrendo de forma manual.

Por outro lado, o Item 18 representa uma *User Story* de Alta prioridade pelo valor que ela traria ao síndico do condomínio e aos seus moradores, considerando os frequentes defeitos e manutenções associados ao sistema de *tags* atual. Contudo é uma demanda que depende de infraestrutura, integração e outros fatores que inviabilizam o seu desenvolvimento nas *Sprints* iniciais, sendo, portanto, configurada com um *rank* alto na fila de execução.

**Obs. não deve ser aplicada estimativa de esforço nas turmas de 1º. Semestre dos cursos**

Tabela 1 – *Product Backlog*

Rank	Prioridade	User Story	Estimativa	Sprint
1	Alta	Como síndico, quero abrir solicitações de vistoria predial para atender à legislação de meu município	5	1
2	Alta	Como atendente, quero gerenciar as solicitações de vistoria para agendar atendimento	8	1
3	Alta	Como síndico, quero receber o laudo da vistoria predial para anexar ao processo de auditoria da prefeitura	5	1
...	Média	...	...	...
17	Baixa	Como atendente, quero um relatório de aniversários dos funcionários do condomínio que utilizam o sistema para enviar cartões acompanhados de brindes	2	6

18	Alta	Como síndico, quero realizar a abertura dos portões eletrônicos do condomínio através do e-Condos, para que eu possa reaproveitar a experiência dos condôminos com o aplicativo e-Condos e sua adesão a 100% dos moradores fornecendo uma alternativa a <i>tag</i> atual	40	?
----	------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----	---

## 2.2. Sprint Backlog

Um *Sprint Backlog* consiste em uma lista de *User Stories* a ser desenvolvida em uma determinada *Sprint*. *User Stories* de maior rank devem ser transferidas do topo do *Product Backlog* para o *Sprint Backlog* em quantidade que a Equipe consiga finalizar em uma *Sprint*, de acordo com as estimativas.

Equipes experientes tem uma boa noção da própria capacidade e conseguem indicar a quantidade máxima de pontos que conseguem trabalhar em uma *Sprint*. Nesses casos, eles podem se sentir tentados a "completar" a *Sprint* com *User Stories* de menor tamanho. Exemplo: Usando a Tabela 1, imaginemos que as *User Stories* serão trabalhadas por uma Equipe capaz de entregar, no máximo, 16 *Story Points* (unidade de estimativa) por *Sprint*. Sabendo disso, eles não teriam como incluir a *User Story* de rank 3 em seu *Sprint Backlog* (a soma seria de 18 *Story Points*). Eles poderiam incluir, por exemplo, a *User Story* com rank 17 para "completar" a *Sprint* (fechando 15 *Story Points*), contudo isso traria pouco valor ao cliente. O ideal seria que eles incluíssem a *User Story* de rank 3, sem se comprometer a entregá-la (ela não faria parte da Meta do *Sprint*), gerando assim um valor de negócio muito maior.

Uso da técnica de *Planning Poker* é bem vinda – [artigo](#).

Figura 1 – *Planning Poker*

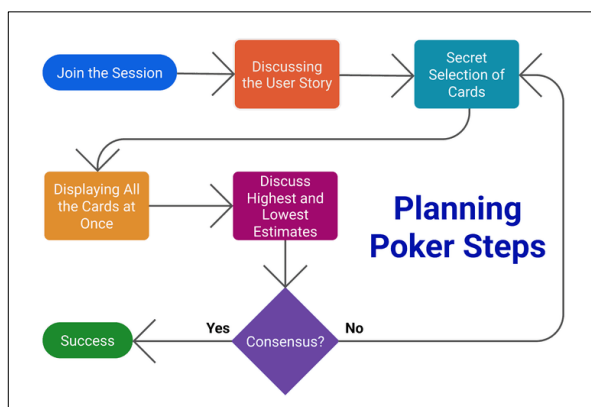


Tabela 2 – *Sprint Backlog*

Capacidade estimada da Equipe por Sprint:	16 Story Points
<b>Meta da Sprint:</b>	User Stories de rank 1 e rank 2 (total de 13 <i>Story Points</i> )
Previsão da Sprint (extras, sem compromisso de entrega):	User Story de rank 3 (5 <i>Story Points</i> )

Rank	Prioridade	User Story	Estimativa	Sprint
1	Alta	Como síndico, quero abrir solicitações de vistoria predial para atender à legislação de meu município	5	1
2	Alta	Como atendente, quero gerenciar as solicitações de vistoria para agendar atendimento	8	1
3	Alta	Como síndico, quero receber o laudo da vistoria predial para anexar ao processo de auditoria da prefeitura	5	1

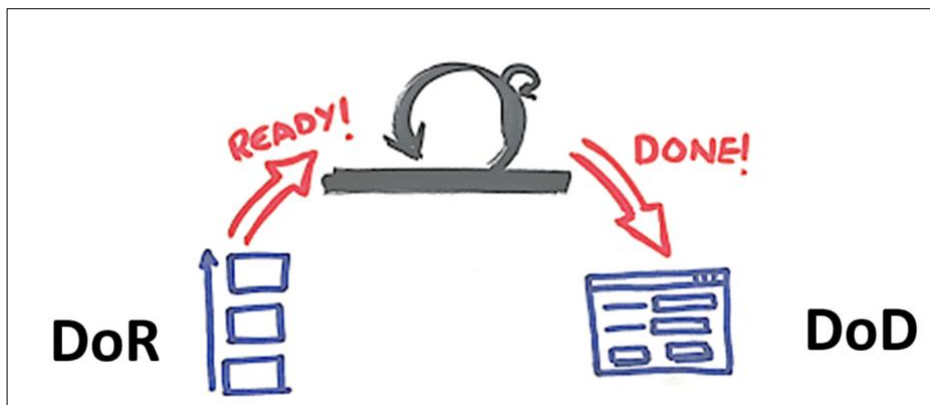


### 3. DoR – Definition of Ready

Antes do início de cada Sprint, os artefatos correlatos aos itens de Backlog de Produto que serão trabalhados em uma dada Sprint (no nosso caso, User Stories) devem estar prontos (em inglês, *ready*) para que a Equipe consiga realizar plenamente seu desenvolvimento.

Quando um item do Backlog de Produto é descrito como “Ready” ou “Pronto”, todos precisam entender o que “Ready” significa para o time.

Figura 2 – DoR e DoD



Comumente trata-se de um “acordo” entre a Dev. Team e o PO, tal como um checklist dos artefatos (forma, conteúdo e critérios de qualidade)

**Exemplos de artefatos de um DoR:** User Story detalhada com Critérios de Aceitação, Wireframes ou Mockups, Modelos de Dados, dentre outros.

Figura 2 – Da esquerda para direita: de Wireframe à Mockup

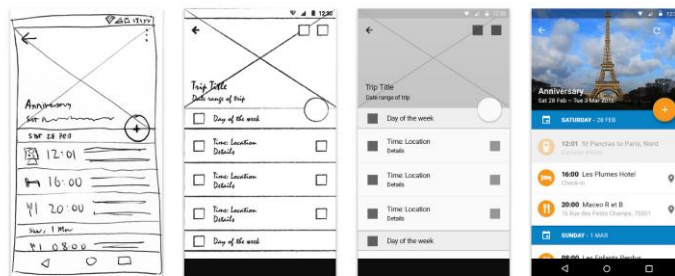
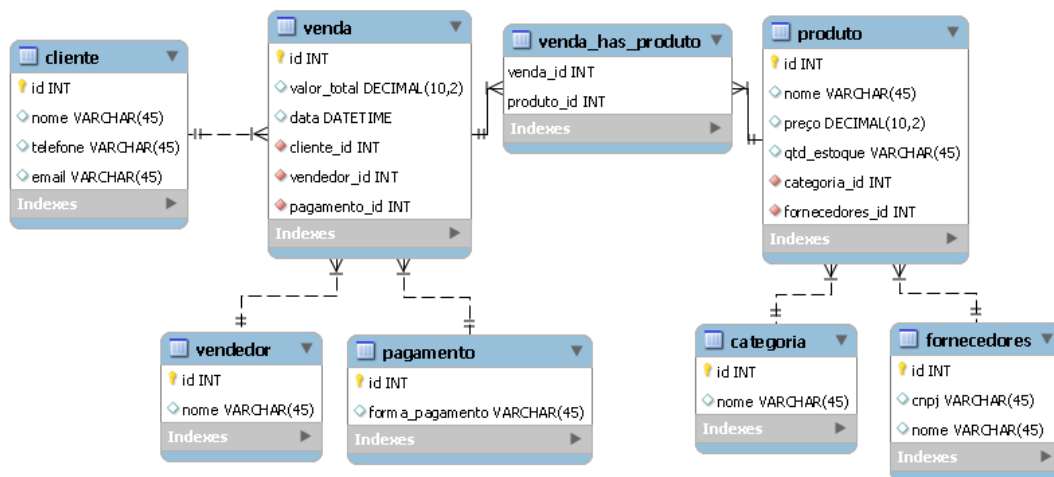




Figura 3 – Modelo de Dados usando notação de Engenharia da Informação (crow's foot)



Recomenda-se, fazer um checklist para o DoR. Segue um exemplo:

Sobre User Stories:

- ☐ Tem título claro, descrição bem definida e objetivo compreendido
- ☐ Tem critérios de aceitação escritos
- ☐ Tem regras de negócio claras
- ☐ Foi estimada pela equipe
- ☐ Sem dependências bloqueadoras
- ☐ Compreensão validada com o time

Sobre artefatos correlatos às User Stories:

- ☐ Design/documentação disponível
- ☐ Regras de negócio detalhadas (texto ou diagrama)
- ☐ Modelo de dados disponível
- ☐ Estratégia de testes definida

## 4. Alocação de Tarefas, com base no DoR

Para realizar uma boa alocação de Tarefas, o DoR deve ter sido garantido!

A alocação de tarefas deve ocorrer logo no início da Sprint, levando em consideração:

Habilidades técnicas (frontend, backend, QA etc.)

- Familiaridade com o domínio da tarefa;
- Carga atual de trabalho;
- Interesse ou objetivo de desenvolvimento profissional.

Classifique as tarefas, por complexidade, prioridade e dependência:

- Tarefas mais críticas ou complexas;
- Tarefas que dependem de outras.

**Importante:** NUNCA atribuir tarefas de forma que um membro da equipe fique com duas tarefas em paralelo.

Podem ser usadas ferramentas como o JIRA, por exemplo, para atribuir e acompanhar a entrega de tarefas, no entanto, existem outras ferramentas interessantes.