



# Certified Tech Developer

The Ultimate Degree

## Requisitos de software

Entender os requisitos e as histórias de usuários é necessário para a construção de um produto ou serviço que, no nosso caso, seria um software.

Depois de percorrer o conteúdo deste átomo, você será capaz de:

- Reconhecer os tipos de requisitos para o design de um produto / serviço.
- Detectar as informações necessárias em cada requisito para reduzir o risco de má interpretação.
- Escrever os requisitos no formato da história do usuário.
- Usar histórias de usuários para melhorar o resultado do seu produto.

### O que são requisitos de software

Na engenharia de software, um requisito ou requisito de um produto ou serviço é um documento com a especificação de uma condição ou capacidade de um software, que um usuário precisa para resolver um problema ou atingir um objetivo.

Os requisitos vêm em dois tipos: funcionais e não funcionais.

- Os **requisitos funcionais** apresentam o que o produto deve fazer. As regras de um jogo online são exemplos de requisitos funcionais.

- **Requisitos não funcionais** especificam com que qualidade um produto deve ter. Um possível requisito não funcional para um jogo online é que o jogo deve fornecer uma interface de fácil compreensão



ou que a resposta a uma ação do usuário deve ser gerenciada em menos do que um determinado tempo máximo.

Eles também podem ser restrições, onde mostram os limites dentro dos quais o sistema deve ser executado. Para um jogo online, uma possível restrição é que ele só pode ser jogado no Chrome ou Firefox.

### **Objetivo dos requisitos**

O objetivo dos requisitos é especificar as funcionalidades e características do software ou produto a ser desenvolvido, o que permite aos desenvolvedores entender o que o cliente quer ou espera dele, e especificar o método ou testes que serão necessários para verificar se o software ou produto atende às especificações exigidas pelo cliente.

### **Histórias do usuário**



User Story ou “história de usuário” é uma descrição concisa de uma necessidade do usuário do produto (ou seja, de um “requisito”) sob o ponto de vista deste usuário. A User Story busca descrever essa necessidade de uma forma simples e leve.

Um dos princípios por trás das User Stories é a de que o produto poderia ser integralmente representado por meio das necessidades de seus usuários (Jeffries et al., 2000). O produto desenvolvido com Scrum é descrito por meio de itens do Product Backlog e, assim, de acordo com esse princípio, cada um desses itens deveria ser representado no formato de User Stories. Ou seja, uma User Story representa um e apenas um item do Product Backlog.

### **Como escrever uma história de usuário:**

As histórias de usuários são compostas de duas partes principais:

> Como, eu quero, para.

Exemplo: "Como <persona> que se beneficiará com o desenvolvimento da história do usuário> eu quero <conteúdo da história> para <benefício que fornecerá a história>."

> Critérios de aceitação: são as condições que devem ser levadas em consideração no desenvolvimento e teste, que devem ser acordadas com o cliente.

### **Tipos e boas práticas:**

> Defina as funções do sistema e use corretamente a função para a qual cada história é direcionada.

> Critérios de aceitação completos: Embora você não deva se aprofundar em questões de design, é uma boa prática não deixar definições incompletas que levem à ambigüidade. Qualquer coisa que não seja especificada é candidata a ser um problema. Deixe claras as entradas e saídas esperadas.



> Use um vocabulário que todos os membros da equipe e o cliente tenham concordado.

> Para cada especificação, considere todo o impacto que ela pode ter em diferentes partes e componentes do sistema, seja dentro da mesma aplicação, com outros serviços internos ou com terceiros.

> Adicione todas as informações que agregam valor em uma seção separada: referências, anexos, maquetes, etc.

### **Dos requisitos às histórias do usuário:**

Os requisitos vêm de metodologias tradicionais de análise de software, relacionadas a um tipo de desenvolvimento em cascata - criado na era industrial - eficaz para projetos estruturados e previsíveis, onde a tarefa a ser realizada é padronizada e executada pelo responsável de acordo com as especificações. o resultado esperado é claro e as variáveis que o afetam podem ser previstas. Mas com o surgimento dos produtos digitais, este tipo de metodologias e requisitos

trazem muitas complicações em um contexto em constante mudança, onde devemos nos concentrar em ajudar o usuário ao invés de realizar a tarefa ou ficar preso nela.

Enquanto os requisitos tradicionais são mais focados do lado do sistema, o que o sistema deve fazer, nas metodologias ágeis o formato é alterado para expressar os requisitos e as histórias de usuário são implementadas para um produto. Estes estão focados no que o usuário deve alcançar, são escritos como se fosse o usuário na primeira pessoa, buscando empatia.



Os requisitos tradicionais foram detalhados e transmitidos de um para outro por meio de documentação complicada e extensa, gerando problemas de entendimento para o desenvolvedor. As histórias de usuários são curtas e transmitem a essência principal e as características mais importantes; Podem conter documentação de apoio ao entendimento mútuo, mas devem ser sempre transmitidos a todas as partes interessadas, oralmente, com suporte visual, para que todas as dúvidas sejam esclarecidas de forma prática e fluente.

Na agilidade, as funções da equipe de desenvolvimento são multifuncionais, ou seja, várias pessoas podem ser rotacionadas em diferentes funções e considera-se que não existe uma maneira única de fazer bem as coisas, o que, graças às histórias de usuários, está habilitado a desenvolvedor para tomar as decisões necessárias na hora de resolver a tarefa, desobstruindo os obstáculos ou complicações pelo caminho e cumprindo o verdadeiro espírito da entrega do software funcionando.