Módulo 1

**UML (unified modeling language)**

Considerando que o padrão atual da indústria de software é o paradigma orientado a objetos, a UML, ou seja, Linguagem de Modelagem Unificada, permite a geração de diversos documentos, de acordo com o referido paradigma, durante o processo de desenvolvimento de software, sendo esses documentos denominados **artefatos de software**, que podem ser textuais ou gráficos.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Levantamento de requisitos**

Envolve a identificação detalhada de requisitos funcionais, que descrevem as funcionalidades específicas que um sistema deve realizar. Além disso, o levantamento de requisitos abrange requisitos não funcionais, como desempenho e segurança, que definem qualidades do sistema. Igualmente importantes são os requisitos de domínio, que englobam as regras de negócio essenciais para o funcionamento do software em um contexto específico. Compreender esses conceitos é muito importante para projetar sistemas eficazes e eficientes e colaborar de forma significativa no desenvolvimento de soluções tecnológicas que atendam às necessidades do mercado.

Como podemos elicitar os requisitos?

O engenheiro de software poderá utilizar determinadas técnicas. Vejamos a seguir.

* Leitura de obras de referência e livros-texto
* Observação do ambiente do usuário
* Realização de entrevistas com os usuários
* Entrevistas com especialistas do domínio
* Reutilização de análises anteriores
* Comparação com sistemas preexistentes do mesmo domínio do negócio

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

**Minimundo do estudo de caso**

**O que é um Minimundo?**

O **minimundo** é uma **representação simplificada da realidade** que será tratada por um sistema de software. Ele define **o escopo** do que o sistema vai abranger, ignorando o que não é relevante para o software.

* Serve como base para **identificar e documentar os requisitos**.
* Ajuda a entender **como o sistema deve se comportar** dentro de um contexto específico.

**Requisitos Funcionais**

São as **funções ou comportamentos** que o sistema deve ter, de acordo com o minimundo.

Exemplos:

* O sistema deve permitir o cadastro de usuários.
* O sistema deve enviar notificações por e-mail.

**Requisitos Não Funcionais**

São os **atributos de qualidade** do sistema. Não dizem o que o sistema faz, mas **como ele faz**.

Exemplos:

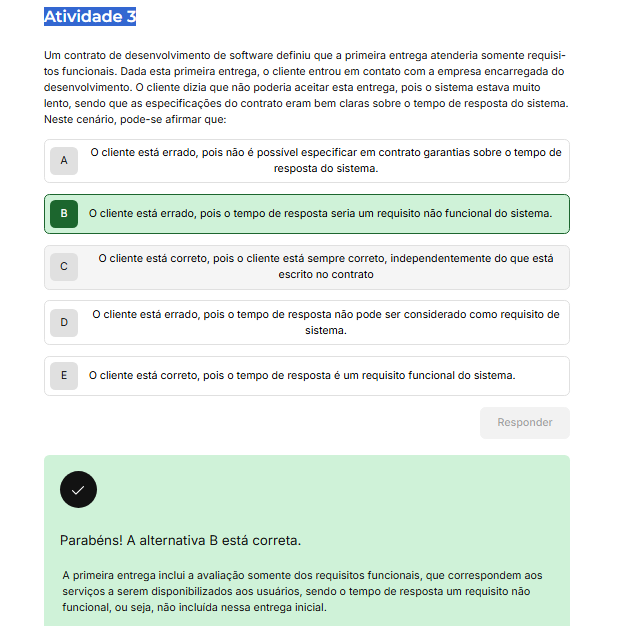
* O sistema deve responder em até 2 segundos.
* A interface deve estar em português.
* O sistema deve ser compatível com navegadores modernos.

**Requisitos de Domínio**

São regras ou restrições que vêm do **domínio de negócio** do minimundo.

Exemplos:

* Um cliente não pode fazer mais de 5 empréstimos simultâneos (em um sistema de biblioteca).
* Vendas só podem ocorrer durante o horário comercial.



**Documentação de requisitos na prática**

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

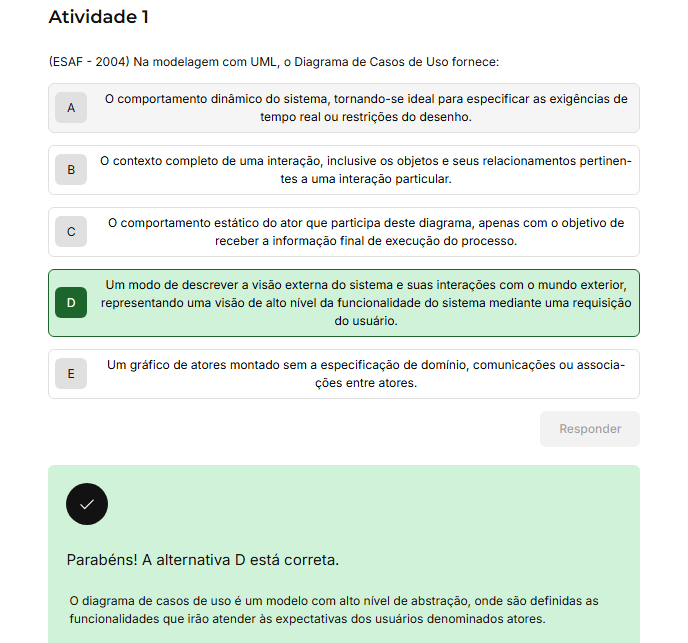
Módulo 2

**Modelo de casos de uso**

Modelos de casos de uso são essenciais na engenharia de software e fornecem uma representação clara e detalhada das interações entre os usuários (atores) e o sistema, facilitando a compreensão das necessidades e dos requisitos funcionais. Compreender esses modelos permite identificar e documentar corretamente os cenários de uso, essencial para desenvolver sistemas eficientes e alinhados às expectativas dos usuários, servindo de base para uma aplicação prática eficaz na modelagem de sistemas complexos.

**A referida abstração funcional é representada pelo modelo de casos de uso, que direciona diversas das atividades posteriores do ciclo de vida do sistema de software.**

O modelo de casos de uso é composto por um ou mais diagramas de casos de uso, artefatos gráficos, e pelas descrições de casos de usos, artefatos textuais. Os componentes do referido diagrama incluem casos de uso, atores, ou seja, elementos externos que interagem com o sistema, e relacionamentos entre os elementos anteriores.



**Estudo de caso – Diagrama de casos de uso**

**Importante destacar que um dos desafios iniciais do engenheiro de software é a modularização da solução do problema, sendo que esse empacotamento permite visualizar os principais módulos que poderão compor a referida modularização.**

**Também vale ressaltar que não existe uma solução única para o modelo de casos de uso e tampouco para o empacotamento, sendo muito provável que diferentes equipes produzam modelos diferentes para o mesmo minimundo.**

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.