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:

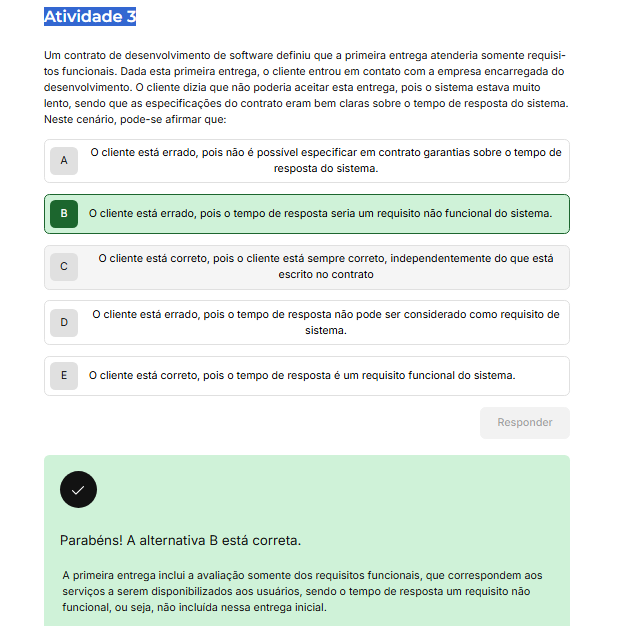
* O sistema deve responder em até 2 segundos.
* A interface deve estar em português.
* O sistema deve ser compatível com navegadores modernos.

**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.

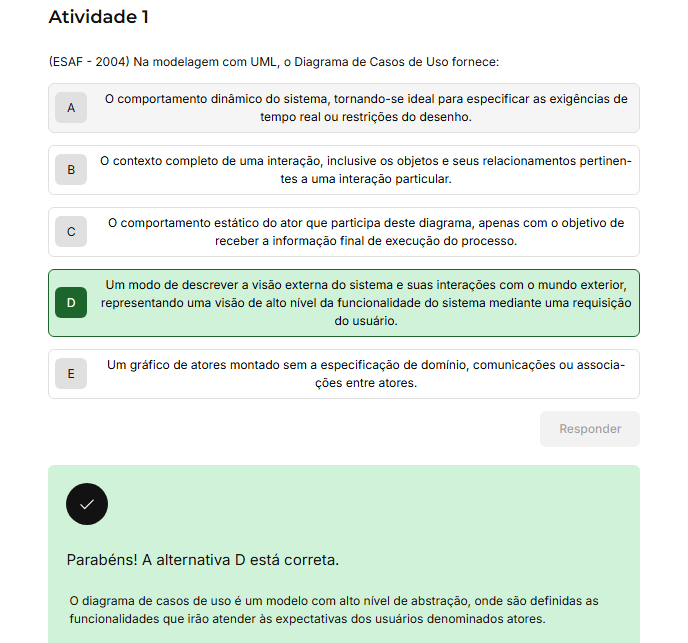
Módulo 2

**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.

**Módulo 3**

**Modelo de classes**

As referidas funcionalidades necessitam, internamente, que objetos colaborem mutuamente para produção dos resultados esperados pelos respectivos usuários; essa colaboração possui um aspecto estrutural estático que permite compreender como o sistema está estruturado internamente para que as citadas funcionalidades externamente visíveis sejam produzidas.

A abstração que permite identificar o aspecto estrutural e estático dos objetos que compõem a solução do problema é a **modelagem de classes**.

Texto

O conteúdo gerado por IA pode estar incorreto.

Classes se relacionam com outras classes ou mesmo com a própria classe através de associações, que podem ser associações simples ou específicas, como a agregação, a composição e a generalização/especialização. Esta última é também conhecida como herança, pelo fato de as classes especializadas (subclasses) herdarem as propriedades das classes genéricas (superclasses). Quando uma subclasse for uma especialização de diferentes superclasses, ocorre a herança múltipla, prevista na orientação a objetos e implementada pela UML.

**Estudo de caso – modelo de classes**

Daremos continuidade ao nosso estudo de caso a partir dos casos de uso identificados no módulo anterior, aplicando a técnica de identificação de classes denominada “**Análise de casos de uso**”, ou seja, a técnica de identificação dirigida por casos de uso. Nessa técnica, o engenheiro de software busca identificar as classes necessárias para produzir o comportamento que está documentado na descrição de caso de uso, de modo que quando todos os casos de uso tiverem sido analisados, todas as classes terão sido identificadas, ou a maioria delas.

\*O diagrama de classes de análise para o caso de uso Registrar Reserva, a seguir, propõe uma solução para a estrutura estática dos objetos identificados, cabendo ressaltar que a tipagem de dados no modelo de classes de análise não é obrigatória.\*

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

**Modelo de atividades**

O modelo de atividades é composto por diagramas de atividades que, entre outros, descrevem os aspectos dinâmicos de um sistema, ou seja, um diagrama de atividade exibe passos de uma computação. Os referidos diagramas possuem as seguintes aplicações:

Modelagem de processo de negócio

Cujo enfoque está em entender o comportamento do sistema no decorrer de diversos casos de uso (processos de negócio), ou seja, como determinados casos de uso do sistema se relacionam no decorrer do tempo.

Modelagem da lógica de um caso de uso

A realização de um caso de uso requer que alguma computação seja realizada e, nessas situações, é interessante complementar a descrição do caso de uso com um diagrama de atividade.

Modelagem da lógica de uma operação

Embora muitas das operações sejam simples, pode haver a necessidade de se descrever a lógica de uma operação mais complexa na forma gráfica.

**Estudo de caso ‒ modelo de atividades**

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

**Modelo de estados**

No contexto do paradigma orientado a objetos, um determinado objeto possui um estado particular, mudando de estado quando acontece algum evento interno ou externo ao sistema.

Nessa transição de um estado para outro, um objeto realiza determinadas ações dentro do sistema.

O modelo de estados é composto por diagramas de estados que descrevem os possíveis estados pelos quais objetos, instâncias de uma determinada classe, podem passar e as alterações dos estados como resultado de eventos que atingem esses objetos.

**Estudo de caso ‒ modelo de estados**

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Modelo de classes de análise**

Corresponde à estrutura estática dos objetos que irão colaborar mutuamente para a produção dos resultados esperados pelos respectivos usuários, ou seja, essa colaboração possui um aspecto estrutural estático que permite compreender como o sistema está estruturado internamente para que as citadas funcionalidades externamente visíveis sejam produzidas.

**Modelo de atividades**

Composto de diagramas de atividades, permite descrever os aspectos dinâmicos de um sistema tendo três possíveis aplicações: modelagem de processo de negócio, modelagem da lógica de um caso de uso e modelagem da lógica de uma operação ou método.

**Modelo de estados**

Composto de diagramas de estados, permite também identificar aspectos dinâmicos de um sistema, sendo que a transição entre estados ocorre a partir de um evento externo, incluindo uma ou mais condições de guarda e uma correspondente ação no sistema.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

**Módulo 4**

**Modelo de interação**

O modelo de interação é composto por diagramas de sequência e/ou diagramas de comunicação, podendo ser iniciado na fase de análise ou na fase de projeto para representar os aspectos dinâmicos do sistema. Consideremos que o início da sua construção ocorre na etapa de projeto do processo, tendo como finalidade principal detalhar os objetos e mensagens requeridas na realização de um caso de uso.

\*Embora os modelos representem visões distintas do sistema, esses modelos são interdependentes, estando a modelagem de interação diretamente relacionada com a modelagem de casos de uso e a modelagem de classes.\*

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

**Estudo de caso – modelo de interação**

Diagrama de sequência

Retornando à nossa descrição, analisemos o passo 1, “**O sistema busca as pousadas disponíveis**”, estando a respectiva modelagem ilustrada a seguir. Nos diagramas de interação, por convenção de boa prática, empregamos os sufixos Model, View e Controller na denominação dos objetos, para indicar as camadas respectivas na fase de implementação do modelo.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Diagrama de comunicação

Agora, vejamos a elaboração do diagrama de comunicação para o mesmo caso de uso, ou seja, “**Registrar Reserva**”. Nesse caso, podemos aplicar a mesma abstração aplicada na construção do diagrama de sequência, ou seja, a partir da descrição de caso de uso e do diagrama de classes identificar os objetos que fazem parte da solução do caso de uso e as respectivas trocas de mensagens, levando em consideração algum requisito não funcional relacionado com o padrão de arquitetura adotado (no caso MVC). A imagem a seguir ilustra o diagrama de comunicação proposto. Note que a sequência da execução é ordenada pelos números sequenciais das mensagens e o resultado das interações é o mesmo do diagrama de sequência.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

**Modelo de classes de projeto**

O modelo de classes de projeto é resultante de refinamentos no modelo de classes de análise, sendo incluídos detalhes úteis para a implementação das respectivas classes.

**Estudo de caso ‒ modelo de classes de projeto**

Aplicando os principais refinamentos no diagrama de classe de análise para o caso de uso Registrar Reserva ilustrado anteriormente no módulo 3, propomos o diagrama de classes de projeto apresentado a seguir. A sintaxe dos atributos foi definida no modelo de análise, sendo as visibilidades privativas (qualificador de visibilidade “–“).

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Podemos observar que as mensagens definidas no diagrama de sequência apresentado anteriormente foram incorporadas como métodos na respectivas classes, tal como a mensagem “10.1 Registrar()” implementada na classe destino “Reserva”, incluindo o tipo de visibilidade, no caso públicas (qualificador de visibilidade “+”), e tipo de retorno.

Vamos exemplificar refinamentos relacionados com a navegabilidade entre classes (repare nas multiplicidades de cada associação):

Texto

O conteúdo gerado por IA pode estar incorreto.

**Modelo de implementação**

O modelo de implementação é composto de um ou mais diagramas de componentes, que permite representar, de forma gráfica, os componentes do sistema e suas dependências.

**Estudo de caso ‒ modelo de implementação**

Diagrama

O conteúdo gerado por IA pode estar incorreto.

A partir do empacotamento de casos de uso, propomos uma solução (entre outras muitas!) onde cada pacote corresponde a uma aplicação. Cada aplicação possui suas próprias classes controladoras (pacote Controller) e views (pacote View); entretanto, as classes do tipo model (**stereotype**) são acessadas por todas as aplicações por uma interface IPadrao. Como especificado no requisito não funcional RNF 2 no módulo 1, o banco de dados é o MySql, sendo o framework de persistência o Hibernate (RNF 4). Lembramos que esse diagrama representa um alto nível de abstração, pois podemos detalhar cada aplicação com um respectivo diagrama de componentes.

**Modelo de implantação**

O modelo de implantação é composto de um ou mais [**diagramas de implantação**](javascript:void(0)) e define os nós de processamento disponíveis, ou seja, os componentes físicos do sistema e suas interdependências. A referida modelagem cresce de importância em função da complexidade do sistema.

**Estudo de caso ‒ modelo de implantação**

A imagem a seguir ilustra uma proposta de diagrama de implantação para o nosso estudo de caso, sendo necessários apenas três nós computacionais para sua implantação.

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Modelo de interação

Permite identificar os objetos e respectivas interações, por meio de mensagens, na realização de casos de uso, sendo esse modelo composto por diagramas de sequência e/ou diagramas de comunicação.

Modelo de classes de projeto

É gerado a partir do refinamento do modelo de classes de análise, ou seja, transformações aplicadas nos atributos, operações e associações. Esse modelo é construído em paralelo com o modelo de interações, contendo detalhes úteis para a implementação das classes nele contidas.

Modelo de implementação

Composto de diagramas de componentes, permite identificar os componentes e suas dependências.

Modelo de implantação

Composto por um ou mais diagramas de implantação, define os nós computacionais necessários à implantação do sistema.

Texto

O conteúdo gerado por IA pode estar incorreto.

Texto

O conteúdo gerado por IA pode estar incorreto.

Exercícios:

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Uma imagem contendo Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

Uma imagem contendo Diagrama

O conteúdo gerado por IA pode estar incorreto.