

# DESCRIÇÃO

A modelagem de um projeto de software de acordo com o paradigma orientado a objetos, usando a linguagem UML (Unified Modeling Language).

# PROPÓSITO

Aplicar os conceitos relacionados com a modelagem de um projeto de software, empregando os modelos da UML em um estudo de caso, é importante para a formação do profissional de Computação.

# PREPARAÇÃO

Antes de iniciar o conteúdo deste tema, instale em seu computador um programa que lhe permita elaborar modelos, sob a forma de diagramas da UML (Linguagem Unificada de Modelagem). Nossa sugestão inicial é o Astah Free Student License, e será necessário usar seu e-mail institucional para ativar a licença. Preencha os dados do formulário, envie e aguarde a liberação de sua licença em seu e-mail institucional. Ao receber a licença, siga as instruções do e-mail e instale o produto em seu computador. Após a instalação, utilize o Astah para acessar os arquivos utilizados neste tema: estudoCasoPousada\_analise e estudoCasoPousada\_projeto.

Sugestões de links adicionais de programas livres para modelagem de sistemas em UML (UML tools) podem ser encontradas em buscas na Internet.

# OBJETIVOS

# MÓDULO 1

Produzir o documento de requisitos para um estudo de caso de modelagem de sistemas

# MÓDULO 2

Construir o modelo de casos de uso para um estudo de caso de modelagem de sistemas em UML

# MÓDULO 3

Construir os modelos de análise para um estudo de caso de modelagem de sistemas em UML

# MÓDULO 4

Produzir o modelo de projeto para um estudo de caso de modelagem de sistemas em UML

# INTRODUÇÃO

O presente tema trata do desenvolvimento de um estudo de caso que inclui as principais atividades de modelagem exigidas durante um processo de desenvolvimento de software.

Na etapa de levantamento de requisitos, são identificados os serviços a serem fornecidos pelo sistema, bem como as restrições operacionais, ou seja, os requisitos funcionais e não funcionais, respectivamente.

A **etapa de análise** inclui a geração do **modelo de casos de uso** a partir dos **requisitos funcionais**, sendo esse modelo composto de diagramas de casos de uso, artefatos gráficos, complementados por descrições de casos de uso, artefatos textuais. Ainda na etapa de análise, são gerados: o **modelo de classes**, composto de diagramas de classes, que representam as

estruturas estáticas dos objetos; o **modelo de atividades**, composto de diagramas de atividades, que descrevem os aspectos dinâmicos do sistema exibindo passos de uma computação, sendo esses orientados a fluxos de controle; e o **modelo de estados**, composto de diagramas de estados, que também descrevem os aspectos dinâmicos do sistema, modelando o comportamento de objetos de uma classe na forma de uma máquina de estados, sendo esses orientados a eventos.

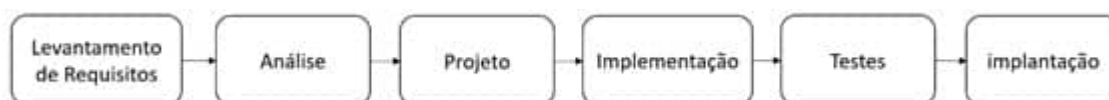
Na **etapa de projeto**, podemos destacar: a geração do **modelo de interação**, que inclui os diagramas de sequência e/ou diagramas de comunicação, usados para modelar a interação entre os objetos das diversas classes identificadas nas fases de análise e projeto; o **refinamento do modelo de classes**, com os aspectos estáticos e estruturais do sistema, ou seja, os diagramas de classes de análise; a montagem do **modelo de implementação**, composto de diagramas de componentes; e a elaboração do **modelo de implantação**, composto de diagramas de implantação.

## MÓDULO 1

---

### 🕒 Produzir o documento de requisitos para um estudo de caso de modelagem de sistemas

As atividades típicas que compõem o processo de desenvolvimento de software estão ilustradas a seguir, que mostra as atividades mais comuns dos processos de desenvolvimento de software, ou seja, qualquer processo deverá possuir as referidas atividades.



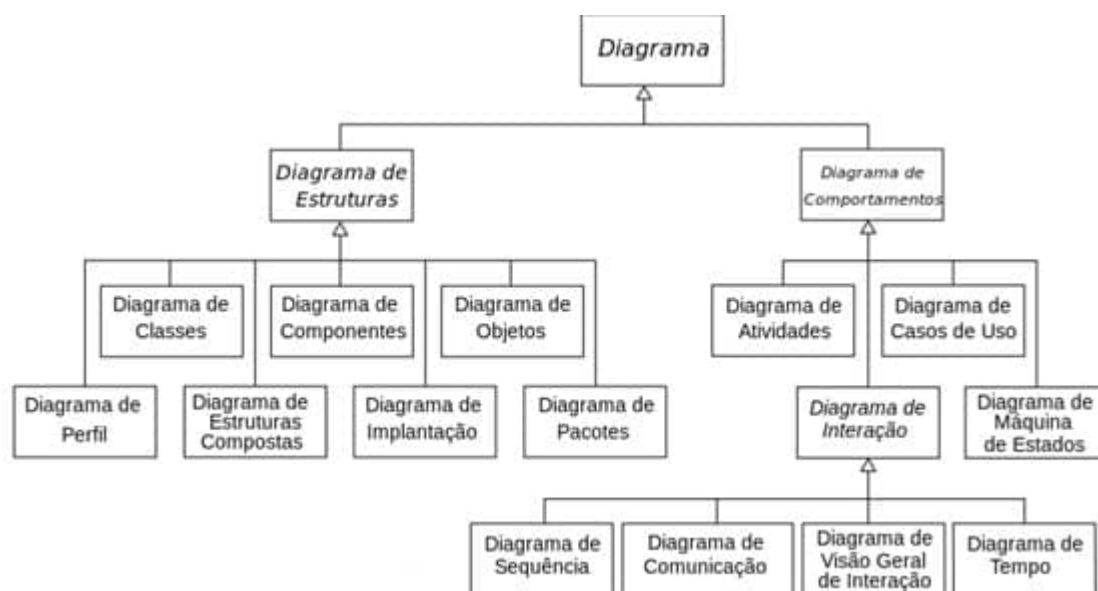
📷 Atividades típicas de um processo de desenvolvimento de software.

## 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.

Os artefatos gráficos produzidos em um sistema orientado a objetos são definidos através dos diagramas da UML. Como mostra a imagem a seguir, os diagramas da UML são classificados em diagramas de estruturas e de comportamentos, também conhecidos como diagramas estáticos e dinâmicos, respectivamente.



## 📷 Diagramas da UML.

Segue, no quadro abaixo, breve descrição dos diagramas:

Diagrama	Especificação
Diagrama de perfil	Permite a definição de tipos padronizados, como estereótipos, restrições e valores rotulados. O foco é a adequação aos modelos UML para diferentes plataformas, por exemplo.
Diagrama de classes	Descreve para cada classe suas propriedades (atributos e métodos) e seus relacionamentos com as demais classes. Classe é a base estrutural dos sistemas orientados a objetos.
Diagrama de estruturas	Possibilita a descrição de colaborações internas de classes, interfaces ou componentes visando a especificação de uma

compostas	funcionalidade.
Diagrama de componentes	Apresenta a estrutura e conexões entre os componentes de um sistema.
Diagrama de implantação	Especifica o ambiente computacional sobre o qual o sistema vai operar, além de distribuir os artefatos (pacotes, classes e componentes) nos nós (elementos computacionais).
Diagrama de objetos	É um diagrama de classes instanciado, ou seja, mostra exemplos de instâncias.
Diagrama de pacotes	Descreve estruturas hierárquicas que englobam outros elementos, como outros pacotes, casos de uso, classes. Usado para modularizar logicamente um sistema em suas partes relevantes (subsistemas).
Diagrama de atividades	Descreve o comportamento de procedimentos, processos de negócios e fluxos de trabalho, suportando processamento sequencial e paralelo.
Diagrama de casos de uso	Mostra como os usuários (atores) interagem com o sistema, do ponto de vista externo, evidenciando as funcionalidades com as quais interagem.
Diagrama de estados	Mostra como os eventos que afetam o sistema alteram o estado dos objetos ao longo de seus ciclos de vida.

Diagrama de sequência	Mostra como os objetos interagem, evidenciando o tempo (a sequência das interações).
Diagrama de comunicação	Mostra como os objetos interagem, evidenciando o tempo das conexões entre eles.
Diagrama de visão geral de interação	Um mix de diagrama de sequência e de atividades. Uma especialização do diagrama de atividades para interações.
Diagrama de tempo	Diagrama de interação, com foco nas restrições de temporização, por exemplo, evidenciar as mudanças no estado de um objeto ao longo do tempo.

Tabela: Alberto Tavares da Silva.

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

## DICA

Nem todos os 14 diagramas serão utilizados no estudo de caso. Na verdade, alguns raramente são usados na prática.

# LEVANTAMENTO DE REQUISITOS

Considerando as atividades genéricas que compõem o **processo de desenvolvimento de software**, a atividade de levantamento de requisitos, também denominada elicitación de requisitos, corresponde à etapa de compreensão do problema por parte do engenheiro de software ou equipe.

# PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

Levantamento de Requisitos, Análise, Projeto, Implementação, Testes e Implantação.

O principal objetivo do levantamento de requisitos é que usuários e equipe de desenvolvimento tenham a mesma visão do problema a ser resolvido.

Nessa etapa, os desenvolvedores, juntamente com os clientes, tentam levantar e definir as necessidades dos futuros usuários do sistema a ser desenvolvido, sendo essas necessidades denominadas requisitos. Um requisito corresponde a uma descrição de um serviço a ser fornecido pelo sistema ou uma restrição operacional.

## **Como podemos elicitar os requisitos?**

O engenheiro de software poderá utilizar determinadas técnicas:

Leitura de obras de referência e livros-texto

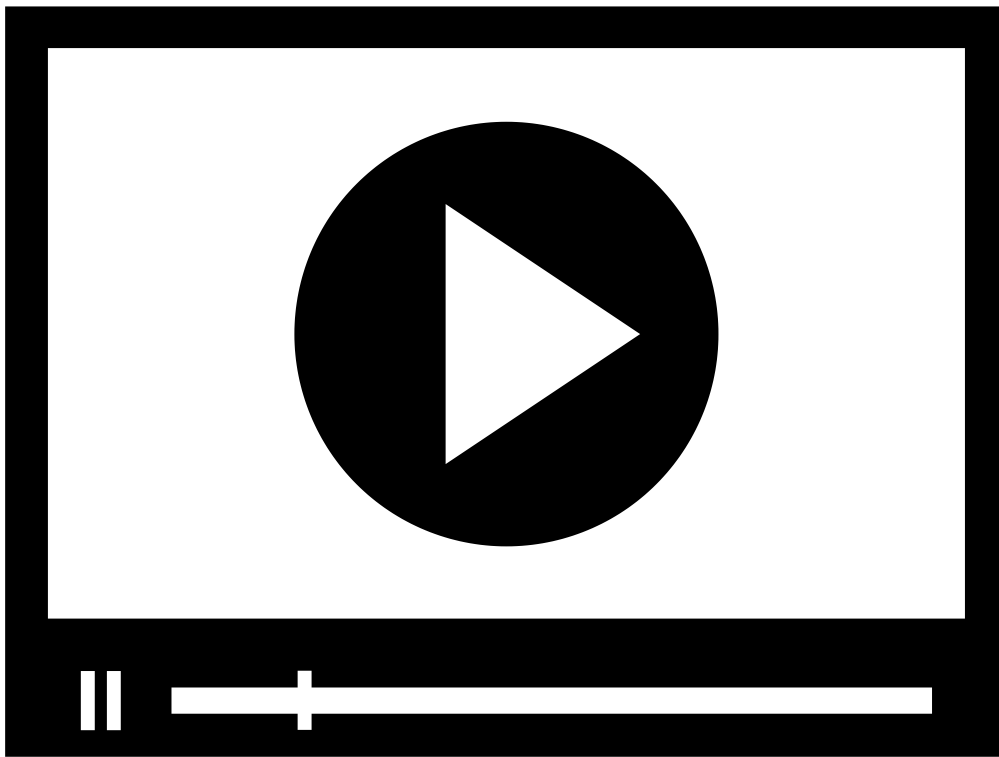
Observação do ambiente do usuário

Realização de entrevistas com os usuários

Entrevistas com especialistas do domínio

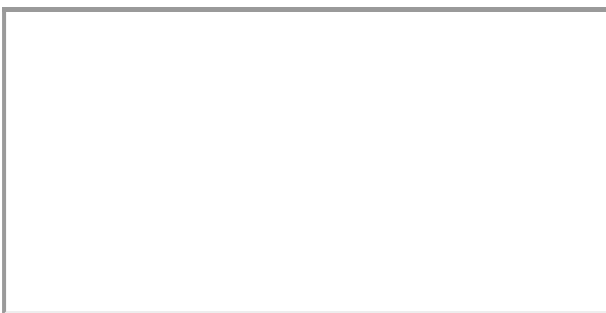
Reutilização de análises anteriores

Comparação com sistemas preexistentes do mesmo domínio do negócio



## ARTEFATOS DO LEVANTAMENTO DE REQUISITOS]

O especialista Alberto Tavares da Silva demonstra as atividades e os artefatos da fase de levantamento de requisitos do processo de desenvolvimento de software.



## MINIMUNDO DO ESTUDO DE CASO

Para fins didáticos, escolhemos um estudo de caso simples o suficiente para entendimento do problema, porém, necessariamente completo para ilustrar todas as atividades do processo de



análise e projeto do sistema, empregando os diagramas da UML mais frequentemente usados na prática.

**Aplicadas as técnicas de elicitação de requisitos, a equipe de engenheiros de software definiu a descrição do minimundo do estudo de caso a ser desenvolvido conforme a seguir.**

## DESCRIÇÃO DO MINIMUNDO

A empresa fictícia Pousadas Férias no Nordeste possui uma rede de pousadas e pretende contratar um sistema de software para a gestão de reservas de quartos online, incluindo a precificação das diárias e produtos disponíveis para consumo, bem como o pagamento dessas diárias e produtos consumidos pelos respectivos clientes.

Uma reserva pode ser realizada de forma online ou por meio de contato telefônico com um funcionário da recepção da pousada. O cadastro do cliente é obrigatório para confirmação da reserva. O cancelamento da reserva pode ser realizado online ou por telefone.

Os produtos disponíveis incluem bebidas, lanches e refeições diversas, sendo os consumos realizados pelos clientes registrados por funcionários atendentes de bar ou de cozinha da pousada.

Os funcionários do setor administrativo da empresa mantêm atualizados os registros necessários à realização de reservas, tais como funcionários, pousadas, quartos e outros, bem como a precificação de diárias e produtos.

A seguir, veremos os requisitos estabelecidos pela equipe de engenheiros de software.

## REQUISITOS FUNCIONAIS (RF)

A tabela a seguir lista os requisitos funcionais estabelecidos pela equipe de engenheiros de software.

RF 1	O sistema deve permitir que os clientes realizem reservas de quartos online ou por contato telefônico com a recepção da pousada.
---------	--

RF 2	O sistema deve permitir que um cliente realize uma reserva somente se estiver cadastrado, podendo o referido cadastro ocorrer de forma online ou por contato telefônico.
RF 3	Quando o cliente realiza uma reserva online, o sistema realiza a cobrança de uma diária no cartão de crédito para confirmação da reserva.
RF 4	Quando o cliente realiza uma reserva por meio telefônico, o sistema exige uma pré-reserva para confirmação da reserva. Nesse caso, a recepção realiza a validação (inclui o registro do valor depositado) ou cancelamento da pré-reserva.
RF 5	O cliente pode cancelar uma reserva.
RF 6	O sistema deve permitir a precificação das diárias dos quartos das pousadas pelo setor administrativo da empresa.
RF 7	O sistema deve permitir a precificação dos produtos disponíveis para consumo pelos clientes pelo setor administrativo da empresa.
RF 8	O sistema deve interagir com um sistema terceirizado (operadora de cartão crédito) para pagamentos com cartão de crédito.
RF 9	Os consumos de produtos por parte dos clientes devem ser registrados pelos atendentes de bares e cozinha das pousadas.

RF 10	Os funcionários do setor administrativo da empresa mantêm atualizados os registros necessários à realização de reservas, ou seja, funcionários, pousadas, quartos.
RF 11	A quitação da reserva, realizada pela recepção da pousada e somente com cartão de crédito, deve ocorrer no <i>check-out</i> .
RF 12	O setor administrativo gera relatórios periódicos das ocupações dos quartos das pousadas.
RF 13	O gerente da pousada gera relatórios periódicos de ocupação dos quartos da pousada.

Requisitos funcionais do estudo de caso. Tabela: Alberto Tavares da Silva.

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

## REQUISITOS NÃO FUNCIONAIS (RNF)

Os requisitos não funcionais estão associados às restrições impostas ao sistema que podem ser categorizados como de produto, organizacionais e externos. A tabela a seguir lista os requisitos não funcionais definidos pela equipe de engenheiros de software.

RNF 1	O sistema deve utilizar tecnologia WEB.
RNF 2	O SGBD adotado é o MySQL.

RNF 3	A linguagem de programação para desenvolvimento adotada é Java, sendo a <b>IDE</b> Eclipse..
RNF 4	O framework para o mapeamento objeto-relacional é o <i>Hibernate</i> .
RNF 5	O padrão de arquitetura adotado é o MVC (model-view-controller).
RNF 6	O processo de desenvolvimento de software adotado é o Scrum.
RNF 7	O prazo estimado do projeto de software é de 04 (quatro) meses.
RNF 8	Atributo de qualidade crítico a ser considerado na seleção de um provedor de serviço na nuvem e de internet: confiabilidade.

Requisitos não funcionais do estudo de caso. Tabela: Alberto Tavares da Silva.

## IDE

Do inglês Integrated Development Environment, ou ambiente de desenvolvimento integrado, é um software que combina ferramentas comuns de desenvolvimento em uma única interface gráfica, facilitando o desenvolvimento de soluções de softwares.

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

# REQUISITOS DE DOMÍNIO

Os requisitos de domínio, também conhecidos por regras de negócio (RN), originam-se do domínio de aplicação do sistema, em vez das necessidades específicas dos usuários do sistema.

A tabela a seguir lista as regras de negócio definidas pela equipe de engenheiros de software.

RN 1	Uma reserva é realizada por um único cliente.
RN 2	Uma reserva pode incluir no mínimo 01 (um) quarto e no máximo 03 (três) quartos disponíveis no momento da reserva.
RN 3	A pré-reserva exigida na reserva por meio telefônico impõe um depósito bancário, por parte do cliente, correspondente a 40% do total da reserva na conta da respectiva Pousada. Nesse caso, a validade ou cancelamento da reserva é realizada pela recepção. O cliente tem o prazo de 24h para envio do recibo de depósito via WhatsApp ou SMS.
RN 4	O cancelamento de uma reserva pode ser realizado até 24h antes da data de <i>check-in</i> estabelecida na reserva.
RN 5	No caso de cancelamento da reserva realizada por telefone, o valor da pré-reserva é estornado ao cliente.
RN 6	No caso de cancelamento por período menor que 24h, ou não comparecimento, será cobrada uma diária.

RN 7	Uma reserva realizada online somente é confirmada após a autorização da operadora de cartão (sistema terceirizado).
RN 8	Os consumos de produtos por parte dos clientes são pagos no momento do <i>check-out</i> , com o valor restante das diárias.
RN 9	O <i>check-in</i> é realizado a partir de 15h e o <i>check-out</i> , até 12h.
RN 10	O sistema deve permitir o pagamento da reserva online e no <i>check-out</i> somente com cartão de crédito.

Regras de negócio do estudo de caso. Tabela: Alberto Tavares da Silva.

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

Vimos neste módulo que a etapa de levantamento de requisitos, ou elicitação de requisitos, permite o entendimento do problema por parte da equipe de engenheiros de software.

Na etapa de levantamento de requisitos, é gerado o documento de requisitos categorizado em requisitos funcionais, não funcionais e de domínio, as denominadas **regras de negócio**.

A identificação dos requisitos é realizada por meio de técnicas, tais como leitura de obras de referência e livros-texto, observação do ambiente do usuário, realização de entrevistas com os usuários, entrevistas com especialistas do domínio e outras.

O documento de requisitos inclui os requisitos funcionais que correspondem aos serviços a serem fornecidos pelo sistema, os requisitos não funcionais, às restrições operacionais e os requisitos de domínio, que se originam do domínio de aplicação do sistema.

## VERIFICANDO O APRENDIZADO

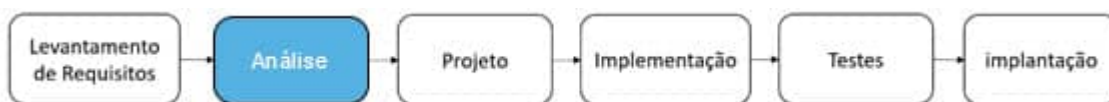
# MÓDULO 2

---

- ⦿ Construir o modelo de casos de uso para um estudo de caso de modelagem de sistemas em UML

## MODELO DE CASOS DE USO

Vejamos agora a etapa de análise do processo de desenvolvimento de software genérico.

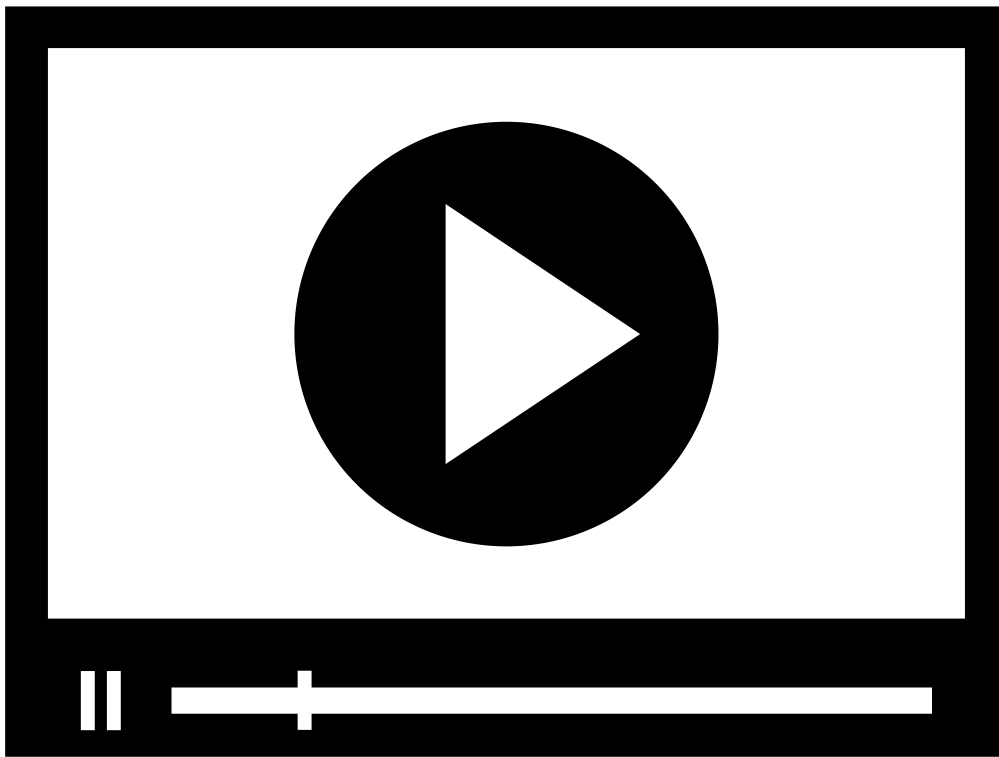


📷 Etapa: Análise.

Nessa etapa, os engenheiros de software realizam um estudo detalhado dos requisitos levantados na etapa anterior, construindo modelos para representar o sistema a ser desenvolvido. Usualmente, a primeira abstração é a funcional, pois permite aos engenheiros de software moldar o sistema de acordo com as necessidades dos usuários.

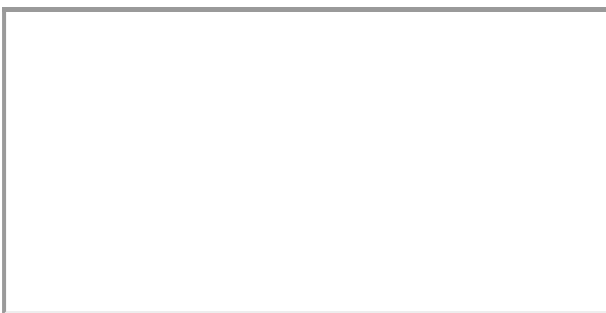
A referida abstração funcional é representada pelo modelo de casos de uso, que direciona diversas das atividades posteriores do ciclo de vida do sistema de software.

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.



## **ARTEFATOS DO MODELO FUNCIONAL – DIAGRAMAS DE CASOS DE USO**

O especialista Alberto Tavares da Silva apresenta o modelo de casos de uso e os diagramas do estudo de caso.



## **ESTUDO DE CASO – DIAGRAMA DE CASOS DE USO**



Vejamos agora a construção do diagrama de casos de uso com base nos requisitos funcionais listados no módulo anterior.

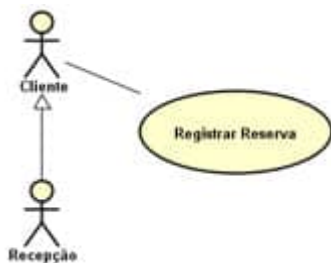
## RF 1

Inicialmente, vamos analisar o RF 1 a seguir:

RF 1	O sistema deve permitir que os clientes realizem reservas de quartos online ou por contato telefônico com a recepção da pousada.
---------	--

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

Podemos identificar no texto um caso de uso denominado “**Registrar Reserva**”, interagindo com dois atores: **Cliente** ou **Recepção**. A referida abstração é representada a seguir.



📷 Diagrama de caso de uso RF 1 gerado pela ferramenta Astah.

O relacionamento entre ambos os atores é denominado de generalização, ou seja, o ator Recepção pode realizar também o caso de uso do ator Cliente.

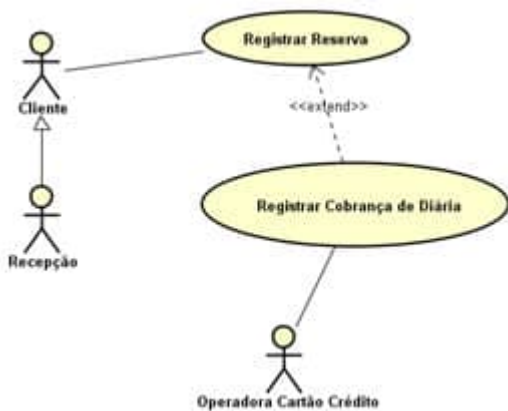
## RF 3

Vamos analisar o RF 3 a seguir:

RF 3	Quando o cliente realiza uma reserva online, o sistema realiza a cobrança de uma diária no cartão de crédito para confirmação da reserva.
---------	---

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

Podemos identificar uma funcionalidade denominada “**Registrar Cobrança de Diária**”, interagindo com o caso de uso “**Registrar Reserva**” e com o ator **Operadora Cartão Crédito**. A abstração proposta está representada na imagem a seguir. Nesse caso, fizemos uso da abstração denominada “**extend**” que representa um relacionamento entre casos de uso, ou seja, o caso de uso “Registrar Reserva” (**estendido**) pode, na forma de um desvio condicional, incluir o comportamento representado pelo caso de uso “Registrar Cobrança de Diária” (**extensor**).



📷 Diagrama de casos de uso RF 3 gerado pela ferramenta Astah.

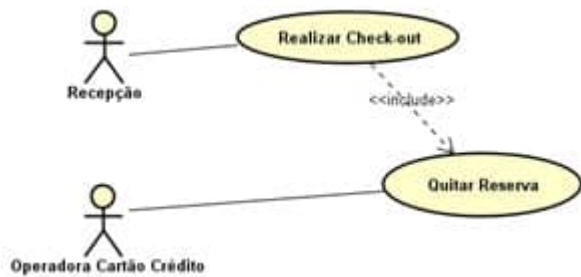
## RF 11

Vamos analisar o RF 11 a seguir:

RF 11	A quitação da reserva, realizada pela recepção da pousada e somente com cartão de crédito, deve ocorrer no <i>check-out</i> .
----------	---

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

Podemos identificar uma funcionalidade denominada “**Realizar Check-out**” e uma segunda, “**Quitar Reserva**”. A primeira interage com o ator **Recepção**. A abstração proposta está representada a seguir.

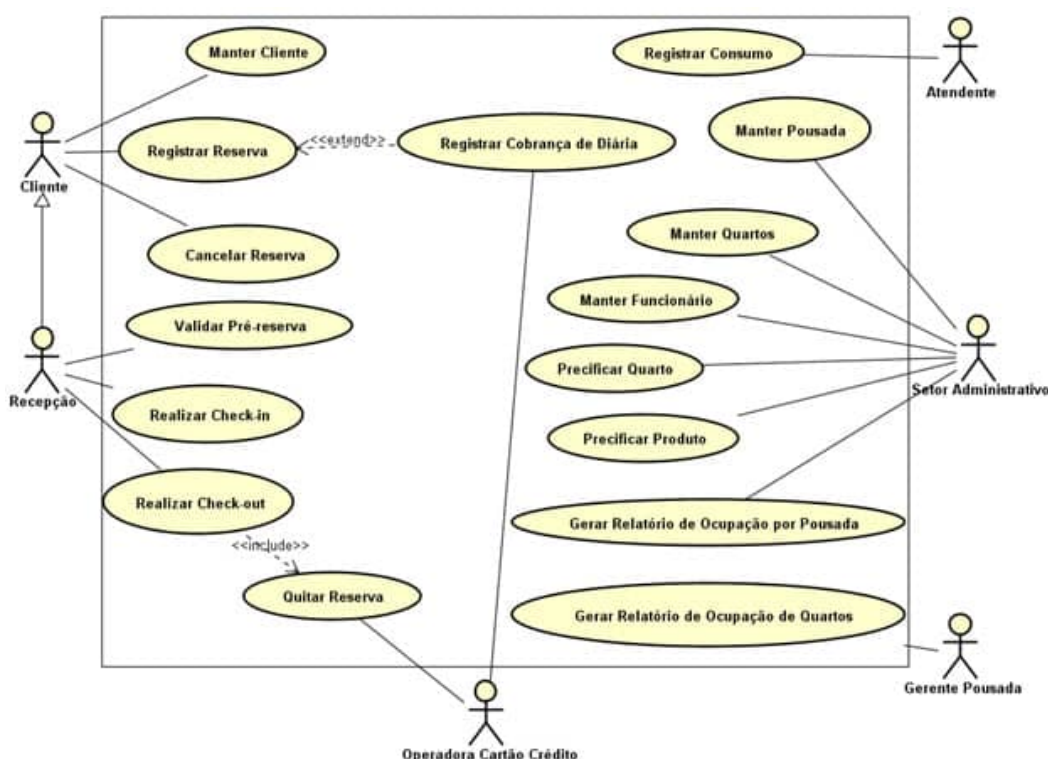


📷 Diagrama de casos de uso RF 11 gerado pela ferramenta Astah.

Nesse caso, fizemos uso da abstração denominada “**include**” que representa um relacionamento entre casos de uso, ou seja, o caso de uso “Realizar *Check-out*” (**inclusor**) utiliza, na forma de um desvio incondicional, o comportamento representado pelo caso de uso “Quitar Reserva” (**incluído**). Por sua vez, o caso de uso “Quitar Reserva” interage com o ator Operadora de Cartão de Crédito.

## DIAGRAMA DE CASOS DE USO APLICANDO TODOS RF

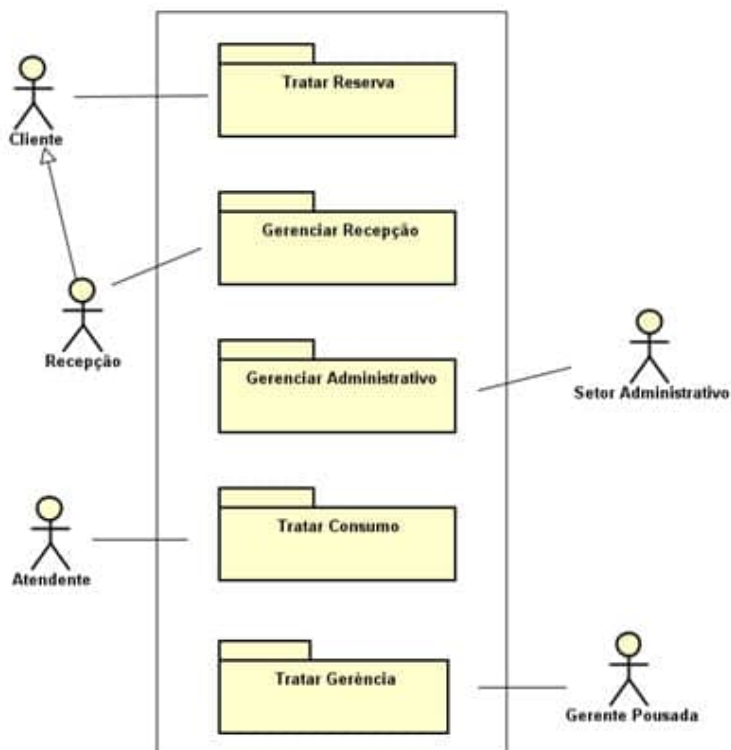
Aplicando as devidas abstrações para todos os requisitos funcionais listados na tabela de requisitos funcionais vista no módulo 1, propomos o diagrama de casos de uso, ilustrado a seguir, para nosso estudo de caso.



📷 Diagrama de casos de uso do estudo de caso gerado pela ferramenta Astah.

Em sistemas complexos, representar todos os casos de uso do sistema em um único diagrama de casos de uso talvez o torne um tanto ilegível. Uma possibilidade é realizar o agrupamento de casos de uso em pacotes, podendo esse agrupamento lógico ocorrer por ator, por ciclo de desenvolvimento ou por área específica da empresa.

Veja a seguir uma proposta de empacotamento por atores.



📷 Diagrama de pacotes de casos de uso do estudo de caso gerado pela ferramenta Astah.

## 📢 ATENÇÃO

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.

# ESTUDO DE CASO – DESCRIÇÕES DE CASOS DE USO

Lembramos que cada caso de uso é definido por meio de uma descrição textual das interações que ocorrem entre ator(es) e o sistema, não sendo a forma dessa descrição definida pela UML.

De acordo com Bezerra (2014), existem dois tipos de casos de uso:

## PRIMÁRIOS

Representam os objetivos dos atores e modelam os processos que estão sendo automatizados.

## SECUNDÁRIOS

Não agregam valor direto aos papéis dos atores, sendo, entretanto, necessários para o funcionamento do sistema.

Vejamos um exemplo a partir de um caso de uso que apresente um objetivo bem específico para os atores Cliente e Recepção: “**Registrar Reserva**”. Nesse caso, temos dois cenários: um principal, que considera a reserva online realizada pelo cliente, e um alternativo, que considera a reserva realizada por telefone e registrada por um funcionário da recepção.

Caso de uso: Registrar Reserva.

Atores: Cliente ou Recepção (funcionário)

Pré-condição: Cliente ou Recepção deve estar identificado pelo sistema.

Fluxo Principal:

O sistema busca as pousadas disponíveis.

O Cliente/Recepção informa a pousada destino e as datas de *check-in* e *check-out*.

O sistema busca quartos disponíveis no período informado.

O sistema lista os quartos disponíveis para o período informado por tipo, incluindo a quantidade máxima de pessoas a serem acomodadas.

O Cliente/Recepção seleciona os quartos e informa a quantidade de pessoas por quarto selecionado, sendo no mínimo 1 quarto e no máximo 3 (de acordo com RN 2).

O sistema exige os dados de login do cliente para efetivação da reserva online.

O cliente informa os dados para login no sistema.

O sistema valida o cliente.

O sistema calcula o valor total da reserva.

O sistema gera um resumo da reserva incluindo a pousada, período, quantidade de pessoas acomodadas por quarto selecionado e valor a pagar.

O Cliente/Recepção confirma a reserva.

Obter confirmação de pagamento parcial de reserva online por meio de cartão de crédito <<extend Registrar Cobrança de Diária>>.

O sistema registra a reserva e gera o código de reserva.

#### Fluxo Alternativo (6): reserva realizada por contato telefônico (recepção da pousada)

O sistema exige dados de login do funcionário da recepção para efetivação da reserva por contato telefônico.

O funcionário da recepção informa dados para login.

O funcionário da recepção informa o CPF do cliente.

O sistema valida o cliente.

Prosseguir no passo 9.

Fluxo Alternativo (10): reserva realizada por contato telefônico (recepção da pousada)

O sistema gera um resumo da reserva incluindo a pousada, período, quantidade de pessoas acomodadas por quarto selecionado, o valor total a pagar e o valor para depósito da pré-reserva pelo cliente (de acordo com RN 3).

Prosseguir passo 11.

Fluxo Alternativo (12): reserva realizada por contato telefônico (recepção da pousada)

Prosseguir no passo 13.

Fluxo de exceção (12): cartão de crédito não validado pela operadora.

Se o cliente tiver outro cartão, repetir o passo 12, senão encerrar caso de uso.

Pós-condição: o código da reserva deve ser informado ao cliente.

Regras de negócio: RN 2 e RN 3.

Tabela: Alberto Tavares da Silva.

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

Vejamos agora uma descrição de um caso de uso considerado secundário, cuja realização inclui as operações de **CRUD**.

# CRUD

CRUD (acrônimo do inglês *Create*, *Read*, *Update* and *Delete*) são as quatro operações básicas (criação, consulta, atualização e destruição de dados) utilizadas em bases de dados relacionais (RDBMS) fornecidas aos utilizadores do sistema.

## Caso de uso: Manter Pousada.

Atores: Setor administrativo (funcionário).

Pré-condição: O funcionário do setor administrativo deve estar identificado pelo sistema.

### Fluxo Principal:

O setor administrativo seleciona a operação:

Opção Inserir.

Opção Consultar.

Opção Alterar.

Opção Excluir.

### Fluxo Alternativo (1.1): Inserir

O sistema busca os funcionários na função de gerente.

O setor administrativo informa: CNPJ, nome, nome fantasia, endereço completo, telefone, e-mail de contato e seleciona o gerente da pousada.



O sistema valida dados da nova pousada.

O setor administrativo confirma o cadastro.

O sistema registra a pousada com status de “ativa”.

#### Fluxo Alternativo (1.2): Consultar

O sistema busca as pousadas cadastradas.

O setor administrativo seleciona a pousada.

O sistema busca os dados da pousada, incluindo o gerente dela.

#### Fluxo Alternativo (1.3): Alterar

O setor administrativo seleciona a pousada.

O sistema busca os dados da pousada, incluindo o gerente da pousada.

O sistema busca os funcionários na função de gerente.

O setor administrativo edita: CNPJ, nome, nome fantasia, endereço completo, telefone, e-mail de contato, seleciona novo gerente da pousada e/ou altera o status (ativa/inativa).

O setor administrativo confirma a atualização.

O sistema atualiza registro da pousada.

#### Fluxo Alternativo (1.4): Excluir

O sistema busca as pousadas cadastradas.

O setor administrativo seleciona a pousada.

O sistema busca os dados da pousada, incluindo o gerente da pousada.

O setor administrativo confirma exclusão.

O setor administrativo informa seus dados de login para confirmar a exclusão, pois serão excluídos todos os dados associados, tais como reservas, consumos, funcionários alocados.

O sistema realiza a exclusão da pousada.

Tabela: Alberto Tavares da Silva.

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

Vimos neste módulo a geração do modelo de casos de uso de um estudo de caso, atividade da etapa de **análise** de um **processo de desenvolvimento de software genérico**, a partir de um documento de requisitos gerado na etapa **levantamento de requisitos** do referido processo.

## VERIFICANDO O APRENDIZADO

### MÓDULO 3

---

⦿ Construir os modelos de análise para um estudo de caso de modelagem de sistemas em UML

## MODELO DE CLASSES

Após termos gerado o modelo de casos de uso com as funcionalidades a serem disponibilizadas aos usuários, e ainda na etapa de análise do processo de desenvolvimento de software genérico, vamos à construção do 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.

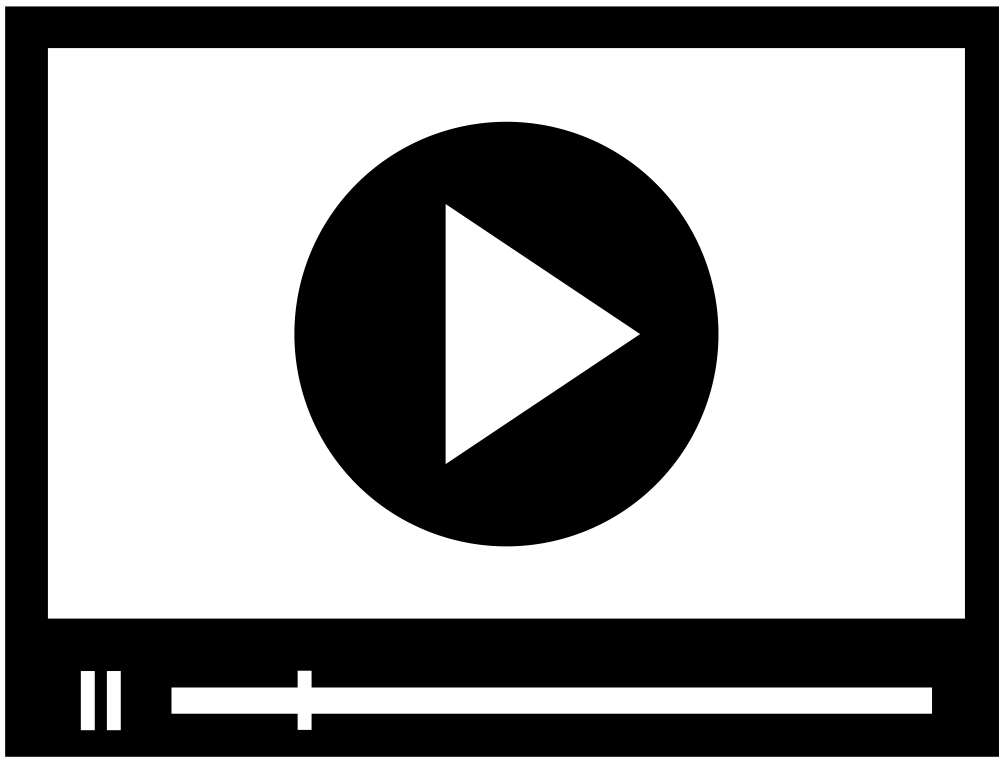
A abstração que permite identificar o aspecto estrutural e estático dos objetos que compõem a solução do problema é a **modelagem de classes**.

Um modelo de classes é composto por um ou mais diagramas de classes, mostrando a existência das classes e os seus relacionamentos numa visão lógica e estática do sistema.

## ATENÇÃO

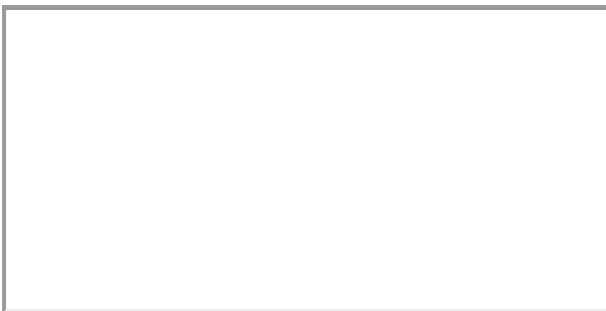
O modelo de classes evolui durante o processo de desenvolvimento. Portanto, o modelo de classes na etapa de análise não deve representar detalhes da solução do problema, limitando-se à identificação das classes, atributos e associações.

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.



## ARTEFATOS DO MODELO DE CLASSES

O especialista Alberto Tavares da Silva apresenta o modelo de classes e os diagramas do estudo de caso.



## 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.

Selecionemos o caso de uso “**Registrar Reserva**” para início da criação do nosso modelo de classes. Analisando a descrição textual desse caso de uso, mostrada anteriormente, podemos identificar os seguintes objetos:

Cliente

Funcionário (reserva realizada por telefone)

Reserva

Pousada

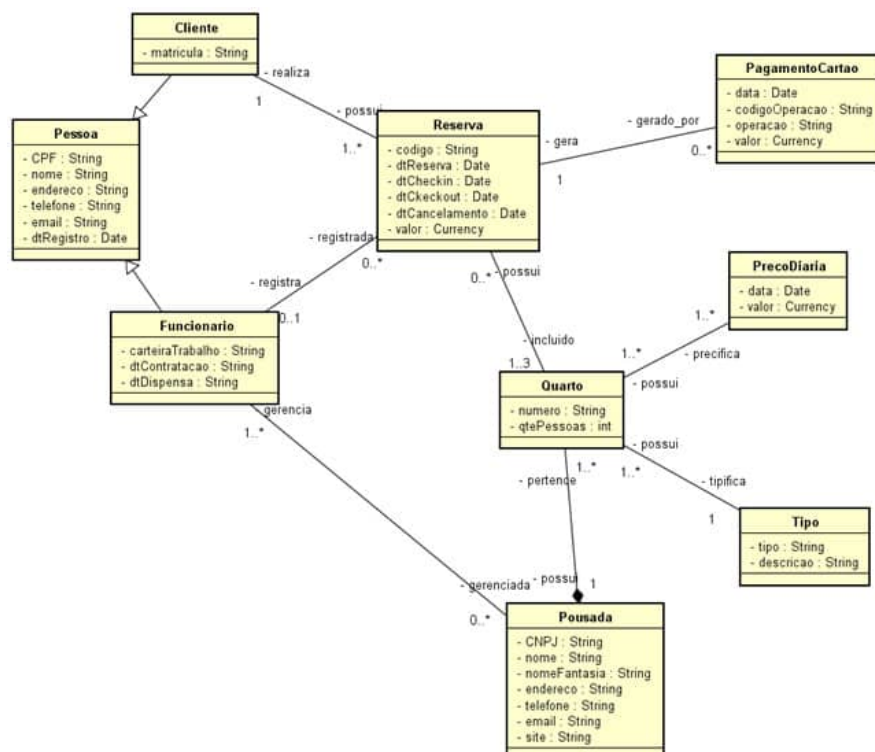
Quarto

Tipo de quarto

Preço da diária (para cálculo do valor total da reserva)

Pagamento (no caso de reserva online)

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 de classes de análise para o caso de uso Registrar Reserva gerado pela ferramenta Astah.

Vejam as principais abstrações aplicadas:

Pessoa é uma superclasse de Cliente e Funcionário.

Toda reserva está associada a 1 (um) cliente, que pode fazer 1 (uma) ou mais reservas.

Uma reserva é registrada por no mínimo 0 (zero) funcionário (reserva online) e no máximo 1 (um) funcionário (reserva por telefone).

Uma reserva pode possuir no mínimo 1 (um) e no máximo 3 (três) quartos (RN 2).

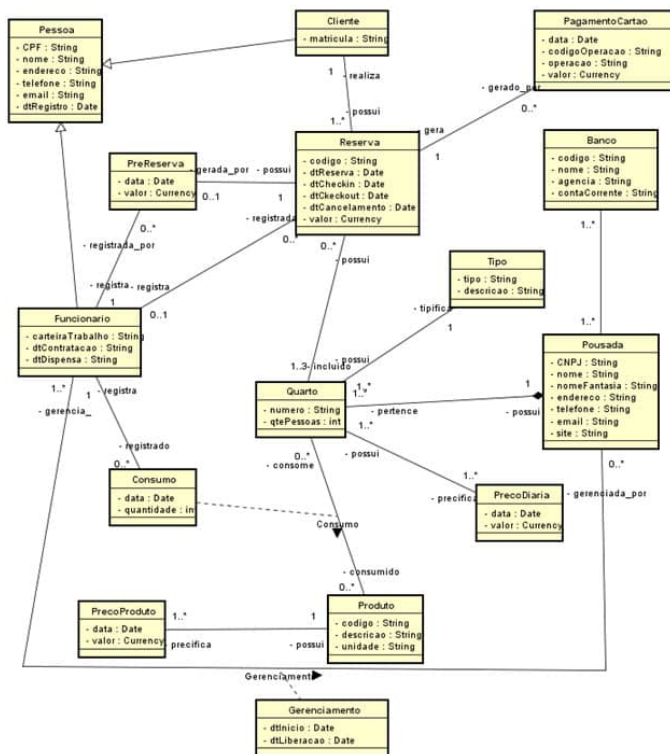
- Uma pousada possui no mínimo 1 (um) e no máximo vários quartos, sendo utilizada a associação de composição, ou seja, o objeto parte (Quarto) só tem existência se o objeto todo (Pousada) existir.

Um quarto se associa com várias instâncias da classe PrecoDiaria, sendo o valor cobrado dependente da data da reserva.

Uma reserva pode estar associada a no mínimo 0 (zero), caso de reserva por telefone, ou no máximo vários pagamentos por cartão, caso de reserva online.

Historicamente, um funcionário pode ter gerenciado 0 (zero) ou mais pousadas e uma pousada pode ter sido gerenciada por 1 (um) ou mais gerentes.

Aplicando a mesma abstração, com foco nos objetos, para os demais casos de uso, temos como resultado o diagrama de classes de análise proposto para o nosso estudo de caso ilustrado a seguir.



📷 Diagrama de classes do estudo de caso gerado pela ferramenta Astah.

## 📣 ATENÇÃO

Assim como no modelo de casos de uso, é importante ressaltar que não há uma solução única para o modelo de classes, em face das diversas possibilidades de modelagem de classes, atributos e associações resultantes do modelo de casos de uso.

Destacamos a abstração aplicada à classe Consumo, sendo essa uma classe associativa resultante da associação entre Quarto e Produto, e que depois se associa à classe Funcionário, responsável pelo registro do consumo. Normalmente, uma classe associativa é necessária quando duas ou mais classes estão associadas, sendo necessário manter informações sobre essa associação, ou seja, toda vez que ocorre um consumo é registrada a data e a quantidade do produto consumido por um funcionário. Essa solução foi aplicada à classe associativa Gerenciamento que permite identificar a data em que um funcionário assume a função de gerente em uma pousada, bem como a respectiva data de liberação da função.

# **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**

## **MODELAGEM DA LÓGICA DE UM CASO DE USO**

## **MODELAGEM DA LÓGICA DE UMA OPERAÇÃO**

## **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**

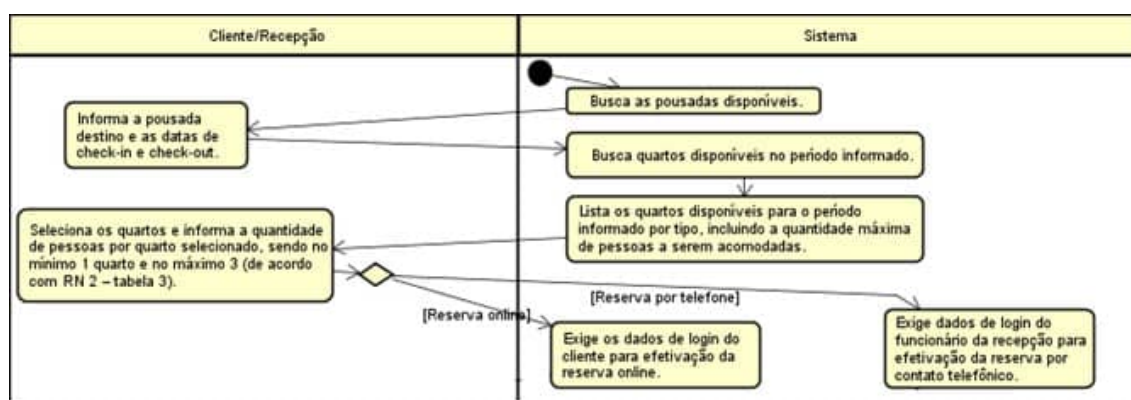
Embora muitas das operações sejam simples, pode haver a necessidade de se descrever a lógica de uma operação mais complexa na forma gráfica.



# ESTUDO DE CASO – MODELO DE ATIVIDADES

Prossigamos no nosso estudo de caso construindo o diagrama de atividades que represente graficamente a modelagem da descrição de caso de uso “**Registrar Reserva**”, estabelecida anteriormente de forma textual em “**Estudo de caso – descrições de casos de uso**”.

Analisando a referida descrição, podemos observar que a interação ocorre entre os atores Cliente/Recepção (que possuem um relacionamento de generalização) e o sistema, ou seja, o diagrama de atividades deve conter duas raias de acordo com a imagem a seguir, de modo que algumas atividades são de responsabilidade dos atores e outras, do sistema.



📷 Diagrama de atividades parcial do caso de uso Registrar Reserva gerado pela ferramenta Astah.

No diagrama de atividades parcial do caso de uso Registrar Reserva, podemos ver as primeiras etapas de montagem do diagrama considerando os seguintes passos computacionais:

## Fluxo Principal:

O sistema busca as pousadas disponíveis.

O Cliente/Recepção informa a pousada destino e as datas de *check-in* e *check-out*.

O sistema busca quartos disponíveis no período informado.

O sistema lista os quartos disponíveis para o período informado por tipo, incluindo a quantidade máxima de pessoas a serem acomodadas.

O Cliente/Recepção seleciona os quartos e informa a quantidade de pessoas por quarto selecionado, sendo no mínimo 1 quarto e no máximo 3 (de acordo com RN 2 – Tabela 3).

O sistema exige os dados de login do cliente para efetivação da reserva online.

...

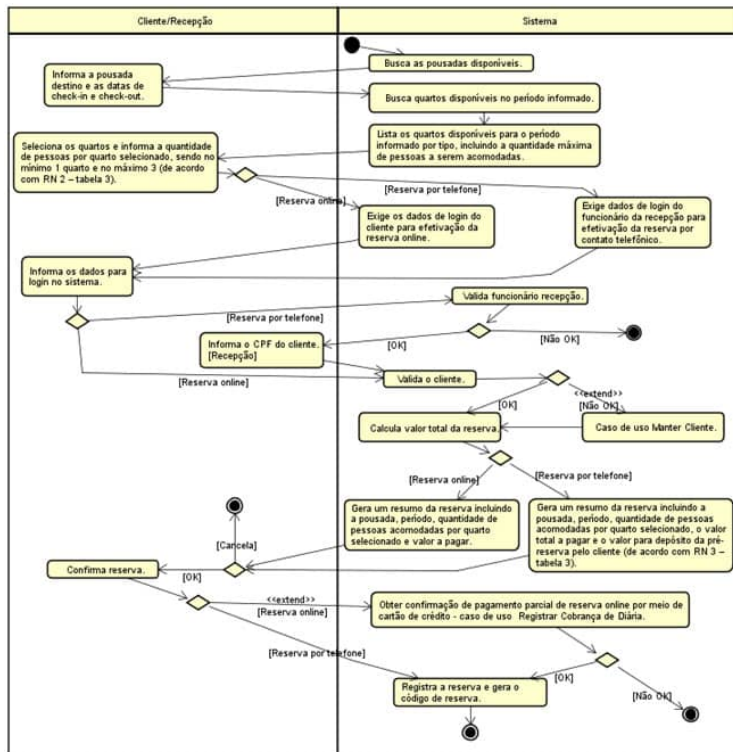
Fluxo Alternativo (6): reserva realizada por contato telefônico (recepção da pousada)

O sistema exige dados de login do funcionário da recepção para efetivação da reserva por contato telefônico.

Tabela: Alberto Tavares da Silva.

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

Após a definição das raias, podemos inserir o elemento que representa o início do processo (círculo escuro). O primeiro passo a considerar é “**Busca as pousadas disponíveis**” sob responsabilidade do Sistema, sendo a próxima atividade “**Informa a pousada destino e as datas de *check-in* e *check-out***”, do ator Cliente/Recepção. Interessante observar que no passo 6 “**O sistema exige os dados de login do cliente para efetivação da reserva online**” temos um fluxo alternativo (losango), ou seja, caso a reserva seja online, a atividade a ser realizada pelo sistema é “**Exige os dados de login do cliente para efetivação da reserva online**” e, no caso de reserva por telefone, a atividade será “**Exige dados de login do funcionário da recepção para efetivação da reserva por contato telefônico**”. Aplicando a mesma abstração nos demais fluxos da referida descrição, teremos o diagrama de atividades do caso de uso “Registrar Reserva” ilustrado na imagem a seguir, onde podemos observar que o fluxo de atividades tem quatro possíveis encerramentos, representados pelos círculos vazados.



📷 Diagrama de atividades do caso de uso Registrar Reserva gerado pela ferramenta Astah.

## MODELO DE ESTADOS

Ainda na etapa de análise do processo, vejamos o 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

Dando continuidade ao nosso estudo de caso, podemos identificar, no diagrama de classes do estudo de caso, uma classe cujos objetos possuem mudanças significativas de estados, a

classe Reserva. Uma reserva pode assumir estados durante a execução do software, tal como no diagrama de atividades do caso de uso Registrar Reserva. Uma reserva pode ter dois estados iniciais:

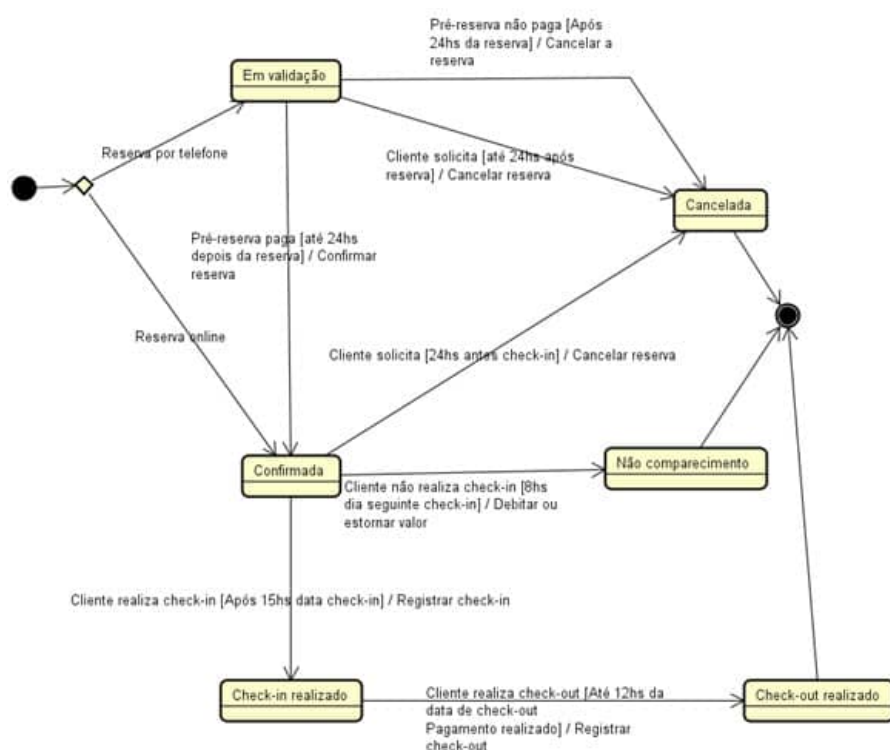
## EM VALIDAÇÃO

Quando a reserva é realizada por telefone

## CONFIRMADA

Quando a reserva é realizada de forma online

Analisemos a mudança de estado de “Em validação” para “Confirmada”, cuja expressão para transição é “**Cliente deposita pré-reserva [até 24h depois da reserva] / Confirmar reserva**”, ou seja, temos a expressão “**evento [condição de guarda] / ação**”. O evento corresponde a “Cliente deposita pré-reserva”, a condição para a mudança de estado inclui “[até 24h depois da reserva]” e, finalmente, a ação correspondente no sistema é denominada de “**Confirmar reserva**”. Aplicando a mesma expressão às demais mudanças de estados, temos o diagrama de estados ilustrado a seguir.



📷 Diagrama de estados para a classe Reserva gerado pela ferramenta Astah.

Vimos, neste módulo, a construção dos artefatos da fase de análise, a saber:

## 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.

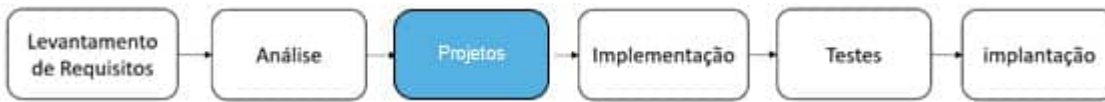
# VERIFICANDO O APRENDIZADO

## MÓDULO 4

---

### 🕒 Produzir o modelo de projeto para um estudo de caso de modelagem de sistemas em UML

Daremos continuidade ao nosso estudo de caso, que trata do gerenciamento de uma rede de pousadas, desenvolvendo os principais modelos da etapa de **projeto** do processo de desenvolvimento de software genérico.



📷 Etapa: Projeto.

## 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.

### DIAGRAMA DE SEQUÊNCIA

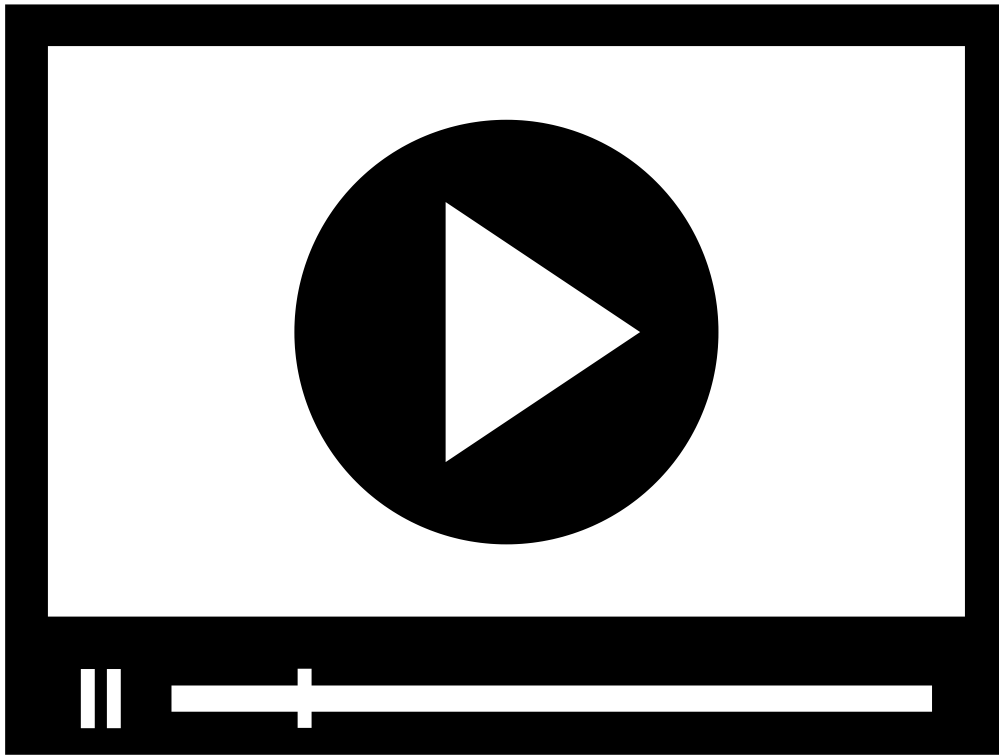
Este diagrama organiza os objetos participantes da interação na horizontal, sendo que cada objeto possui uma linha de vida e as mensagens entre esses objetos são representadas com linhas horizontais rotuladas partindo da linha de vida do objeto remetente e chegando à linha de vida do objeto receptor. A posição vertical das mensagens permite deduzir a ordem na qual elas são enviadas.

### DIAGRAMA DE COMUNICAÇÃO

Este diagrama organiza os objetos em forma de nós, sendo adicionados setas e rótulos de mensagens nas ligações entre esses objetos; a ordem das mensagens é explicitada pela sua numeração sequencial no diagrama de comunicação.

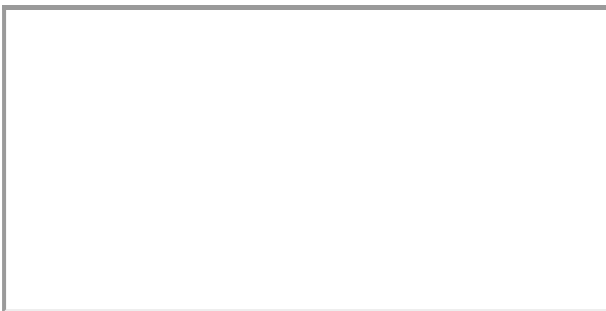
### 📢 ATENÇÃO

Os diagramas de comunicação e sequência representam as mesmas interações e devem estar consistentes com o diagrama de classes e com a respectiva descrição de caso de uso.



## ARTEFATOS DO MODELO DE INTERAÇÃO

O especialista Alberto Tavares da Silva apresenta o modelo de interação e seus diagramas do estudo de caso.



## ESTUDO DE CASO – MODELO DE INTERAÇÃO

Daremos continuidade ao nosso estudo de caso, gerando o diagrama de sequência do caso de uso “**Registrar Reserva**”. Como afirmado anteriormente, esse diagrama possui uma interdependência com os modelos de casos de uso e modelo de classes, portanto vamos

iniciar nossa abstração “dinâmica” com a descrição do referido caso de uso desenvolvida no tópico **Estudo de caso – modelo de casos de uso**, sendo parcialmente replicada a seguir:

Caso de uso: Registrar Reserva.

Atores: Cliente e Recepção (funcionário)

Pré-condição: Cliente ou Recepção deve estar identificado pelo sistema.

Fluxo Principal:

O sistema busca as pousadas disponíveis.

O Cliente/Recepção informa a pousada destino e as datas de *check-in* e *check-out*.

O sistema busca quartos disponíveis no período informado.

O sistema lista os quartos disponíveis para o período informado por tipo, incluindo a quantidade máxima de pessoas a serem acomodadas.

O Cliente/Recepção seleciona os quartos e informa a quantidade de pessoas por quarto selecionado, sendo no mínimo 1 (um) quarto e no máximo 3 (três) (de acordo com RN 2).

O sistema exige os dados de login do cliente para efetivação da reserva online.

...

Fluxo Alternativo (6): reserva realizada por contato telefônico (recepção da pousada)

O sistema exige dados de login do funcionário da recepção para efetivação da



reserva por contato telefônico.

O funcionário da recepção informa dados para login.

O funcionário da recepção informa o CPF do cliente.

O sistema valida o cliente.

Prosseguir no passo 9.

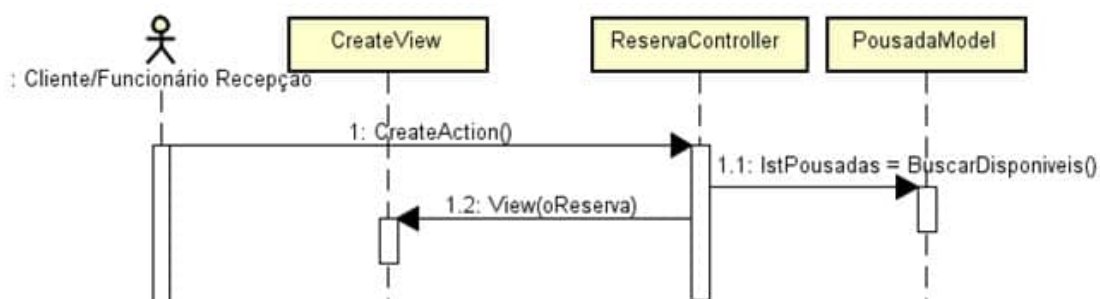
Tabela: Alberto Tavares da Silva.

**Atenção!** Para visualização completa da tabela utilize a rolagem horizontal

Temos que considerar na construção do modelo de interação o requisito não funcional “RNF 5 – O padrão de arquitetura adotado é o MVC (model-view-controller)”, pois novos objetos serão incorporados, a fim de permitir que o projeto do sistema tenha um nível de detalhamento suficiente para sua implementação.

## 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.

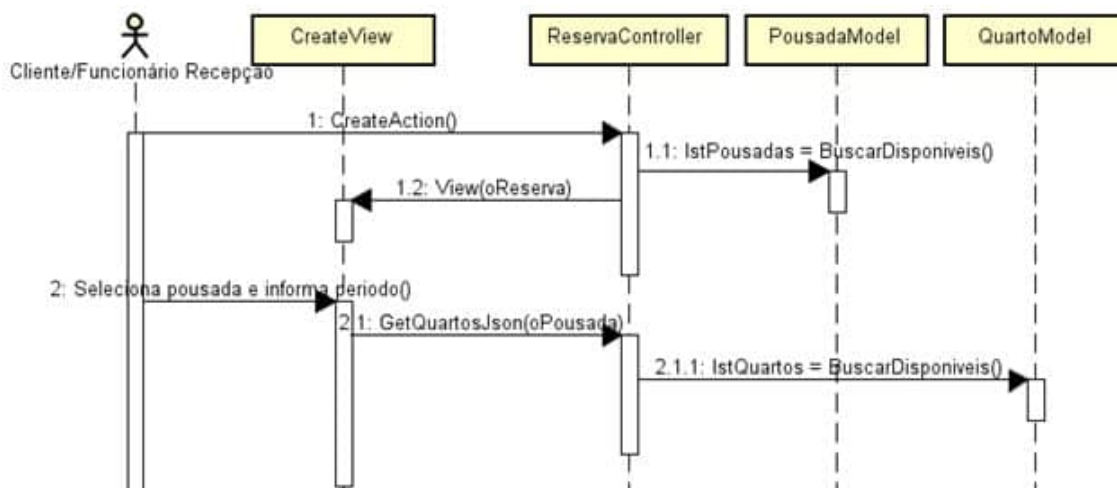


📷 Diagrama de sequência parcial do caso de uso “Registrar Reserva” gerado pela ferramenta Astah.

A interação do ator, cliente ou funcionário da recepção definido no diagrama de casos de uso do estudo de caso, é iniciada pelo método **CreateAction** da classe controladora

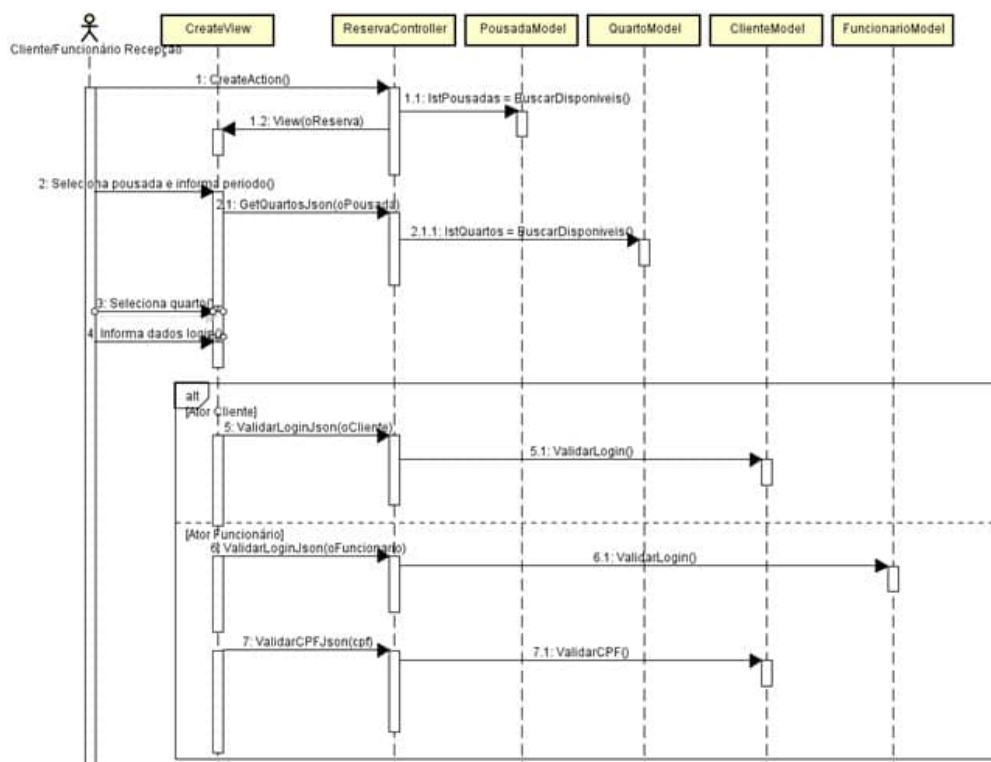
**ReservaController** que, por sua vez, instancia um objeto **PousadaModel**, que é um objeto entidade identificado a partir do diagrama de classes de análise, a fim de que esse busque as pousadas disponíveis.

Em seguida, analisemos os passos 2, “**O Cliente/Recepção informa a pousada destino e as datas de *check-in* e *check-out***” e 3, “**O sistema busca quartos disponíveis no período informado**”, cujas modelagens estão representadas a seguir.



📷 Diagrama de sequência parcial do caso de uso “Registrar Reserva” passo 2 gerado pela ferramenta Astah.

Após uma segunda interação do ator, ou seja, seleção da pousada, uma função JavaScript na camada View aciona um método na classe controladora a fim de buscar os quartos disponíveis para a selecionada pousada no período desejado, conforme observado na imagem a seguir.



📷 Diagrama de sequência parcial do caso de uso “Registrar Reserva” passo 3 gerado pela ferramenta Astah.

Finalmente, vamos analisar o passo 6, **“O sistema exige os dados de login do cliente para efetivação da reserva online”**.

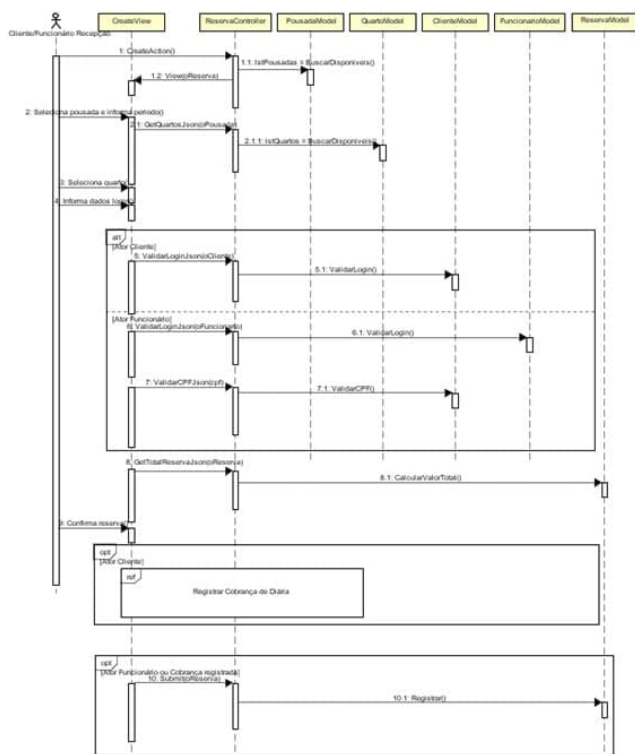
Após uma quarta interação do ator, ou seja, informação de dados do login, ocorre a possibilidade de dois cenários estabelecidos pelos fluxos principal e alternativo, ou seja, reserva realizada pelo cliente (principal) ou pelo funcionário da recepção (alternativo). O operador “alt” permite representar essa abstração, pois caso o ator seja o cliente, será acionado o respectivo método de validação, e caso seja o funcionário, ocorrerá a respectiva validação de login, bem como a validação do CPF do cliente.

Na quinta interação, a confirmação da reserva, temos duas opções de acordo com o cenário; no cenário que a reserva é realizada pelo ator cliente, ocorre a extensão do caso de uso Registrar Cobrança de Diária, reveja o diagrama de casos de uso do estudo de caso:



📷 Extend em diagrama de casos de uso do estudo de caso gerado pela ferramenta Astah.

Caso essa cobrança seja confirmada, ou a reserva seja realizada pelo ator funcionário da recepção (cenário alternativo), ocorre o registro da reserva. Após a aplicação das abstrações descritas, propomos o diagrama de sequência ilustrado a seguir.

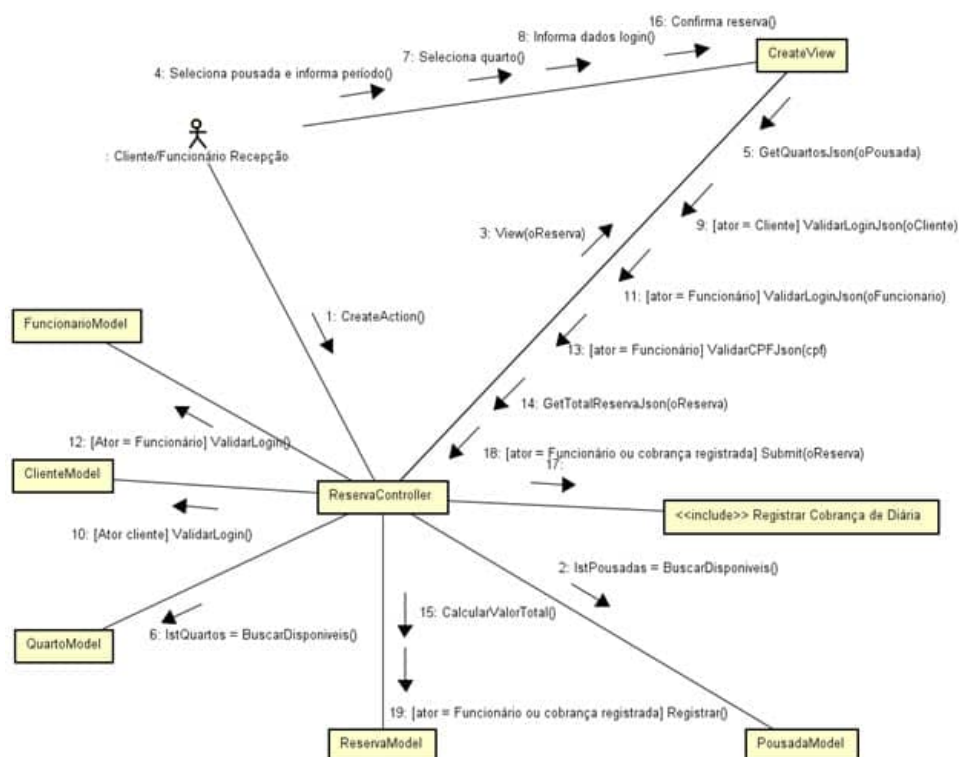


📷 Diagrama de sequência do caso de uso “Registrar Reserva” gerado pela ferramenta Astah.

## 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 de comunicação do caso de uso “Registrar Reserva” gerado pela ferramenta Astah.

## MODELO DE CLASSES DE PROJETO

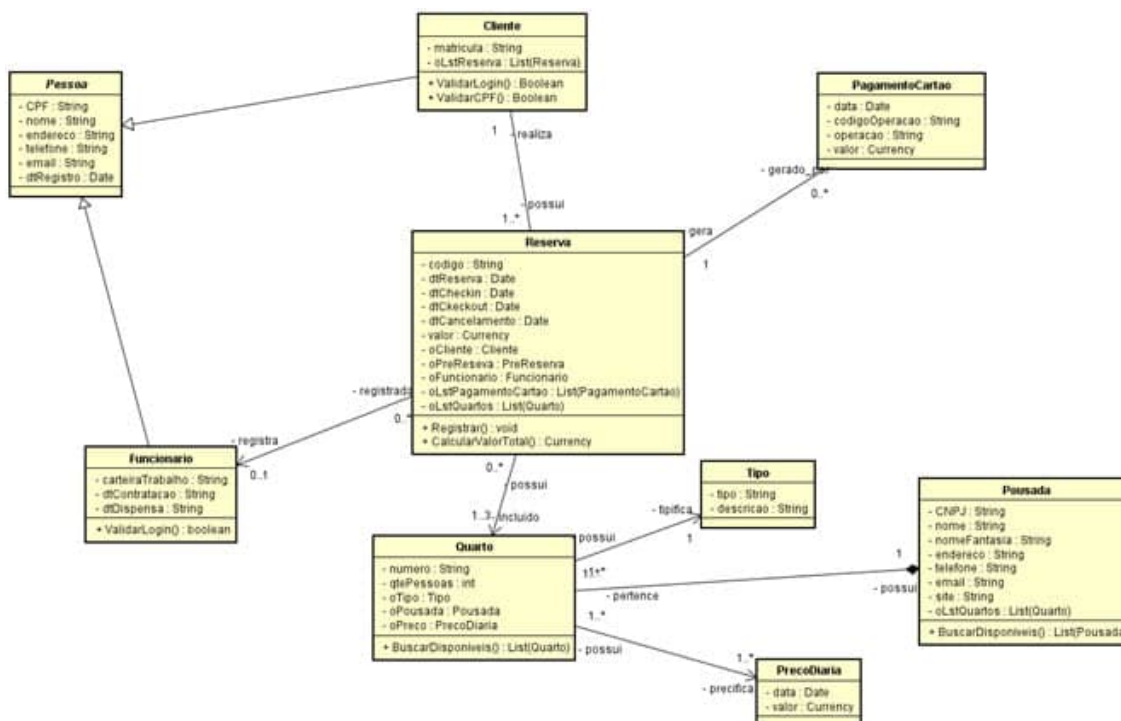
O modelo de classes de projeto é resultante de refinamentos no modelo de classes de análise, sendo incluídos detalhes úteis para a implementação das respectivas classes.

### 💡 DICA

Como boa prática, sugerimos que esse modelo seja construído em paralelo com o modelo de interações.

# 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 de classes de projeto para o caso de uso Registrar Reserva gerado pela ferramenta Astah.

Podemos observar que as mensagens definidas no diagrama de sequência apresentado anteriormente foram incorporadas como métodos nas 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):

## (A)

ocorrência de uma associação bidirecional entre Cliente/Reserva, pois a navegabilidade não está representada por flechas, ou seja, o envio de mensagens deverá ocorrer em ambos os sentidos, sendo que um objeto Cliente poderá instanciar uma coleção de objetos Reserva (veja atributo oLstReserva) e um objeto Reserva poderá instanciar um objeto Cliente (veja o atributo oCliente).

## (B)

ocorrência de uma associação unidirecional entre Reserva/Quarto, pois a navegabilidade está representada por uma flecha, ou seja, o envio de mensagens deverá ocorrer em um único sentido, sendo que um objeto Reserva poderá instanciar uma coleção de objetos Quarto (veja atributo oLstQuarto).

Nosso diagrama de classes de projeto para o caso de uso Registrar Reserva permite ilustrar, também, o refinamento relacionado com o conceito de classe abstrata, que permite organizar hierarquias gen/spec e a implementação do princípio do polimorfismo. A classe Pessoa é uma classe abstrata estando a mesma em *itálico*.

Aplicando a abstração descrita para o caso de uso “Registrar Reserva” para os demais casos de uso, teremos o diagrama de projeto do sistema.

## DICA

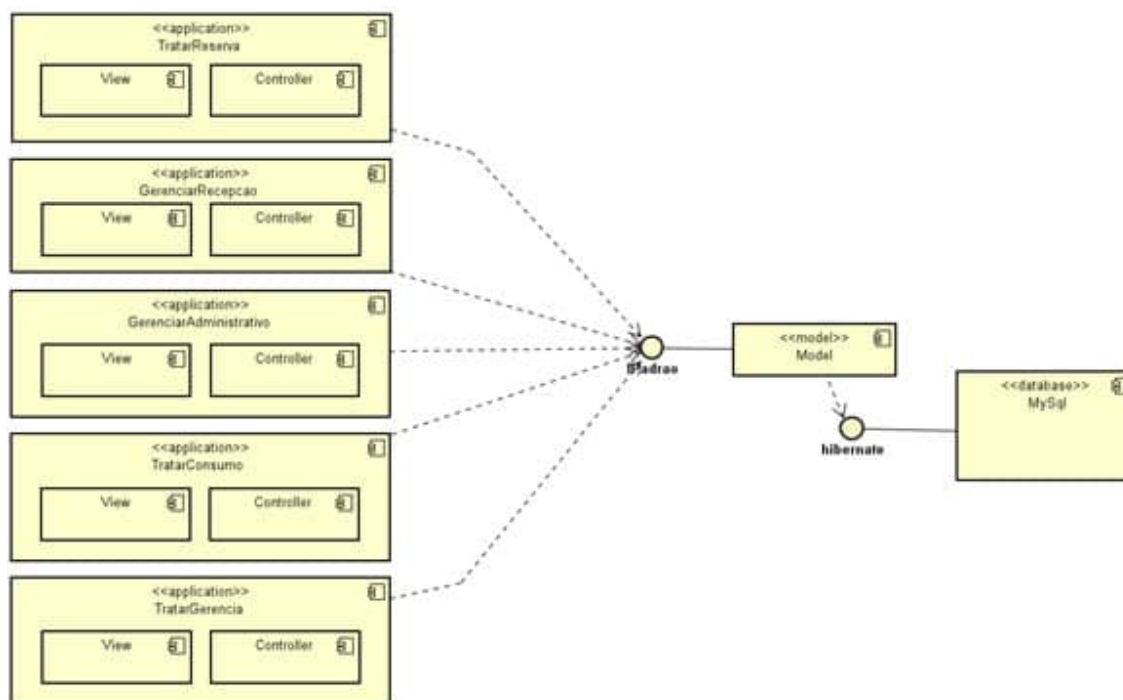
Os métodos de acesso get/set, ou as propriedades, não estão representados em função da geração automática pelos compiladores modernos a partir dos atributos.

# 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

Retornemos ao nosso estudo de caso, ainda na fase de projeto do processo. A imagem a seguir ilustra uma proposta para o diagrama de componentes, considerando os modelos de casos de uso (módulo 2), de classes (módulo 3) e de interação (módulo 4).



📷 Diagrama de componentes para o estudo de caso gerado pela ferramenta Astah.

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

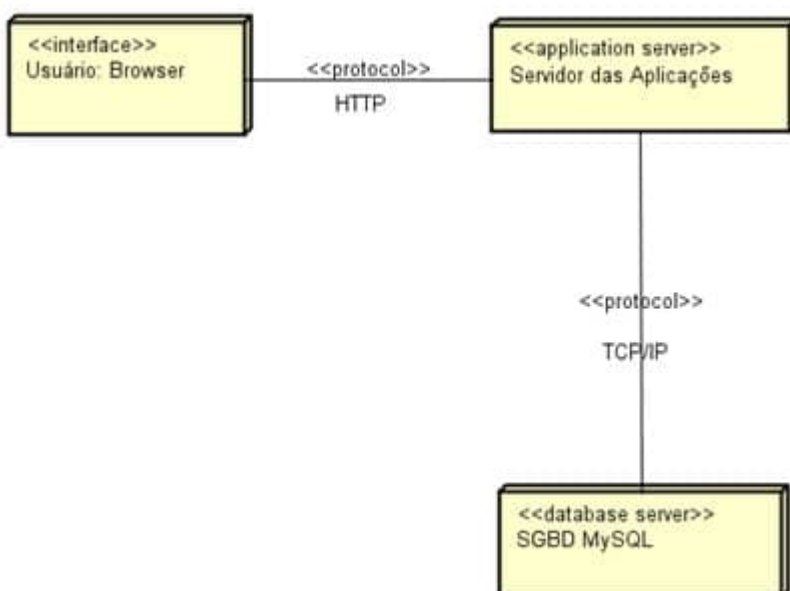
O modelo de implantação é composto de um ou mais **diagramas de implantação** e define os nós de processamento disponíveis, ou seja, os componentes físicos do sistema e suas interdependências. A referida modelagem cresce de importância em função da complexidade do sistema.

## DIAGRAMAS DE IMPLANTAÇÃO

Deployment

## 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 de implantação para o estudo de caso gerado pela ferramenta Astah.

Neste módulo, demos continuidade ao estudo de caso que trata do gerenciamento de uma rede de pousadas desenvolvendo os principais modelos da etapa de projeto do processo de desenvolvimento de software genérico.

Neste módulo vimos os seguintes modelos:

## **MODELO DE INTERAÇÃO**

## **MODELO DE CLASSES DE PROJETO**

## **MODELO DE IMPLEMENTAÇÃO**

## **MODELO DE IMPLANTAÇÃO**

## **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.

## VERIFICANDO O APRENDIZADO

## CONCLUSÃO

## CONSIDERAÇÕES FINAIS

Apresentamos neste tema um estudo de caso que permitiu a aplicação dos conceitos relacionados com a modelagem de um projeto de software, incluindo as etapas de levantamento de requisitos, análise e projeto do processo de desenvolvimento de software genérico.

Na etapa de levantamento de requisitos, identificamos os serviços a serem fornecidos pelo sistema, bem como as restrições operacionais, ou seja, os requisitos funcionais, não funcionais e as regras de negócio.

Na etapa de análise foram gerados, a partir dos requisitos funcionais, o modelo de casos de uso, composto de diagramas de casos de uso e de descrições de casos de uso, o modelo de classes, composto de diagramas de classes, o modelo de atividades, composto de diagramas de atividades, e o modelo de estados, composto de diagramas de estados.

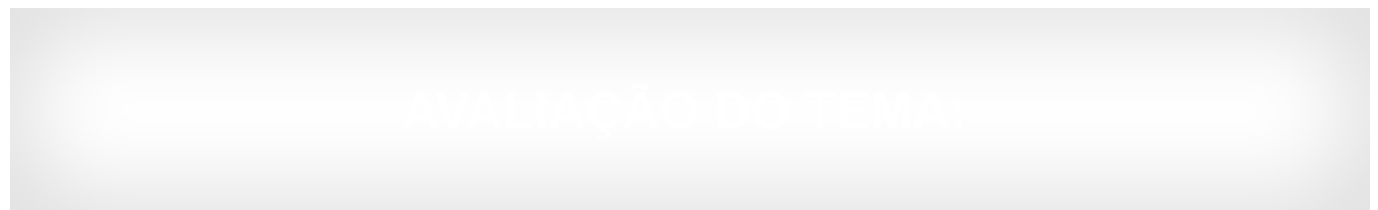
Na etapa de projeto, foi gerado o modelo de interação, que inclui os diagramas de sequência e/ou diagramas de comunicação, o modelo de classes de projeto, composto de diagramas de classes devidamente refinados a partir dos diagramas de classes de análise, o modelo de

implementação, composto de diagramas de componentes, e o modelo de implantação, composto de diagramas de implantação.



## ! PODCAST

Ouçá um resumo do tema com o especialista Alberto Tavares da Silva.



## REFERÊNCIAS

BEZERRA, E. **Princípios de Análise e Projeto de Sistemas com UML**. 3. ed. Rio de Janeiro: Elsevier, 2014.

WAZLAWICK, R. S. **Análise e Projeto de Sistemas de Informação Orientados a Objetos**. 2. ed. Rio de Janeiro: Elsevier, 2011.

---

## EXPLORE+

Para saber mais sobre os assuntos tratados neste tema, leia:

Capítulos 4 a 11 – *Princípios de Análise e Projeto de Sistemas*.

# CONTEUDISTA

Alberto Tavares da Silva

 **CURRÍCULO LATTES**