

Contextualização: Você foi contratado como Arquiteto de Software júnior pela startup "StayWell Soluções Hoteleiras". Sua primeira grande tarefa é projetar e construir o protótipo do núcleo de um novo Sistema de Gestão de Hóspedes (SGH). O objetivo é criar um modelo de objetos coeso e bem estruturado que servirá de base para todo o sistema futuro.

Após reuniões com o cliente, a equipe de análise levantou os seguintes requisitos e regras de negócio:

O sistema deve ser capaz de gerenciar um **Hotel**. Um hotel é definido por seu nome e endereço. A característica principal de um hotel é que ele é **composto por um conjunto de Quartos**. É fundamental entender que **estes quartos são partes intrínsecas e inseparáveis do hotel**; um quarto específico do "Grand Hotel Maringá" não existe sem o hotel. Cada **Quarto** é identificado unicamente por um número, possui um tipo (ex: "Standard", "Deluxe", "Suíte") e um preçoPorNoite. Para simplificar nosso protótipo, cada quarto também terá um status para sabermos se ele está ocupado ou não.

O sistema também precisa registrar os **Hóspedes**. Um hóspede é uma entidade independente, definida por seu nome e cpf, e ele existe no sistema mesmo que não esteja hospedado em nenhum hotel no momento.

A operação central do sistema é a **Hospedagem**. Uma hospedagem representa a estadia de um Hóspede em um Quarto específico do hotel. Ao realizar a hospedagem, o quarto que antes estava livre passa a ficar ocupado. O hotel precisa de uma funcionalidade que permita **hospedar** um cliente em um quarto disponível, bem como uma funcionalidade para fazer o **check-out**, que libera o quarto, tornando-o ocupado = false. Além disso, o hotel precisa ser capaz de **exibir um relatório** mostrando o status de todos os seus quartos (número, tipo e se está ocupado ou não, incluindo os dados do hóspede, caso esteja).

O Desafio

Sua missão é traduzir a especificação textual acima em uma solução de software completa, seguindo os princípios da Programação Orientada a Objetos.

Parte 1: O Design

1. Analise cuidadosamente a contextualização e as regras de negócio descritas.
2. Identifique todas as classes necessárias, seus atributos e seus métodos.
3. Determine os tipos de relacionamento entre as classes (Associação, Agregação, Composição) e suas cardinalidades.
4. Crie um **Diagrama de Classes UML completo** no Astah que represente fielmente a arquitetura da solução.

Parte 2: A Construção

1. **Implemente em Java todas as classes que você projetou no seu diagrama UML.**
2. Siga rigorosamente as boas práticas que aprendemos:
 - **Encapsulamento:** Todos os atributos devem ser private.
 - **Construtores:** Crie construtores para garantir que os objetos sejam criados em um estado consistente e com seus dados essenciais.
 - **Getters e Setters:** Crie os métodos de acesso necessários. Lembre-se que um setter é o local ideal para regras de negócio (ex: a lógica de ocupar/desocupar um quarto).
 - **Coleções:** Utilize ArrayList para gerenciar os relacionamentos "um-para-muitos".

Parte 3: A Demonstração

1. Crie uma classe Main.java com um método main para demonstrar que seu protótipo funciona.
2. Siga este roteiro de demonstração:

- a. Crie um objeto Hotel.
- b. Crie e adicione vários objetos Quarto a este hotel.
- c. Crie dois objetos Hospede diferentes.
- d. Exiba o status inicial de todos os quartos do hotel (todos devem estar livres).
- e. Realize a hospedagem do primeiro hóspede em um dos quartos.
- f. Tente hospedar o segundo hóspede no mesmo quarto (o sistema deve impedir, pois já está ocupado).
- g. Hospede o segundo hóspede em outro quarto vago.
- h. Exiba novamente o status de todos os quartos (agora dois devem estar ocupados).
- i. Realize o check-out do primeiro hóspede.
- j. Exiba o status final de todos os quartos.

