



Kata Reservas corporativas de hotel

Construa um motor para hotéis empresariais. Sua implementação deverá atender a três atores:

- **Gerente de hotel:** Configurar os quartos do seu hotel
- **Gestor da empresa:** Gerencia funcionários e cria políticas para a empresa e para funcionários
- **Funcionário:** Fazer reservas

Para isso, é necessário que o motor tenha 4 serviços que trabalham em colaboração uns com os outros, sendo descritos abaixo. Onde houver `<?>`, use o tipo (classe) que você preferir.

Serviço de Hotel

O **gerente** o utiliza para definir os tipos e números de cada quarto. Também recupera informações de um hotel a partir do ID.

```
public class HotelService {  
  
    // Dependências(?)  
  
    void criarHotel(<?> idHotel, <?> nomeHotel);  
  
    void definirQuarto(<?> idHotel, <?> numeroQuarto, <?> tipoQuarto);  
  
    <?> buscarHotelPorId(<?> idHotel);  
  
}
```

Regras

O método `criarHotel` deve criar um hotel, ou lançar uma exception se o ID do hotel já existe.

O método `definirQuarto` deve **criar ou atualizar** um quarto, ou lançar uma exceção se o hotel não existir.

O método `buscarHotelPorId` deve retornar todas as informações sobre o hotel solicitado.

Serviço de Empresa

Usado pelos gestores para gerenciar seus funcionários.

```
public class EmpresaService {  
  
    // Dependências(?)  
  
    void criarFuncionario(<?> idEmpresa, <?> idFuncionario);  
  
    void excluirFuncionario(<?> idFuncionario);  
  
}
```

Regras

- Não devem haver funcionários repetidos
- Deletar um funcionário deleta todas as informações associadas a ele (políticas, reservas, etc.)

Serviço de política de reserva

Permite que gestores criem políticas de reserva **para a empresa** ou **por funcionário**.

Essas políticas determinam se o funcionário tem permissão da empresa para reservar um quarto. Temos dois tipos de política:

- Política de empresa: Indica os tipos de quartos que podem ser reservados (por exemplo, uma empresa pode permitir apenas a reserva de quartos simples ou de quartos duplos, mas não de quartos Deluxe)
- Política de funcionário: No mesmo modelo da política de empresa, mas se aplica apenas a um funcionário específico

```
public class PoliticaReservaService {  
  
    // Dependências(?)  
  
    void definirPoliticaDeEmpresa(<?> idEmpresa, <?> tiposDeQuarto);  
  
    void definirPoliticaDeFuncionario(<?> idFuncionario, <?>  
tiposDeQuarto);  
  
    boolean podeReservar(<?> idFuncionario, <?> tipoDeQuarto);  
  
}
```

Regras

- A política de funcionário tem precedência, sobrescrevendo a política da empresa
- Se não há uma política de funcionário, a política de empresa deve ser utilizada
- Se não houver nenhuma política por funcionário ou por empresa, o funcionário pode reservar qualquer quarto

Regras técnicas

- Os métodos `definirPoliticaDeEmpresa(...)` e `definirPoliticaDeFuncionario(...)` devem criar ou atualizar uma política. Não são permitidas políticas duplicadas
- O método `podeReservar(...)` deve levar em conta as políticas de empresa e de funcionário

Serviço de Reserva

Permite que funcionários reservem quartos em hotéis.

```
public class ReservaService {  
  
    // Dependência (?)  
  
    Reserva reservar(<?> idFuncionario, <?> idHotel, <?> tipoQuarto, Date  
checkIn, Date checkOut);  
  
}
```

Regras

- Respeitar a assinatura do método `reservar(...)`
- A data de check-out deve ser pelo menos um dia depois da data de check-in
- O hotel precisa existir, e precisa haver um quarto do tipo solicitado disponível durante todo o período da reserva
- As políticas associadas ao funcionário (empresarial e específica) devem ser respeitadas
- Mantenha registro de todas as reservas. Se o hotel tiver 5 quartos padrão, não podemos ter mais de 5 reservas desse tipo de quarto no mesmo dia
- Os quartos de hotel podem ser reservados sempre que não houverem conflitos de data
- Em caso de sucesso, retorne uma confirmação de reserva ou um erro

Regras do desafio

- 100% de cobertura de testes