

AT3/N1 - Atividade prática coletiva - Bimestre N1

Crie um projeto (**em Java 17**) que simule um sistema de reserva e controle de quartos em um hotel, utilizando *threads* em Java.

O sistema deve representar as seguintes entidades, no mínimo:

- Quarto
 - No mínimo, devem existir 10 quartos;
- Hóspede
 - Cada hóspede deve ser representado por uma thread;
 - No mínimo, devem existir 50 hóspedes;
- Camareira;
 - Cada camareira deve ser representada por uma thread;
 - No mínimo, devem existir 10 camareiras;
- Recepcionista:
 - Cada recepcionista deve ser representado por uma thread;
 - No mínimo, devem existir 5 recepcionistas;

E deve se basear na seguintes regras:

- O hotel deve contar com vários recepcionistas, que trabalham juntos e que **devem alocar hóspedes apenas em quartos vagos**;
- O hotel deve contar com várias camareiras;
- Cada quarto possui capacidade para até 4 hóspedes e uma única chave;
- Caso um grupo ou uma família possua mais do que 4 membros, eles devem ser divididos em vários quartos;
- Quando os hóspedes de um quarto saem do hotel para passear, devem deixar a chave na recepção;
- Uma camareira só pode entrar em um quarto caso ele esteja **vago** ou os hóspedes não estejam nele, ou seja, a chave esteja na recepção;
- A limpeza dos quartos é feita sempre após a passagem dos hóspedes pelo quarto. Isso significa que toda vez que os hóspedes saem do quarto (para passear ou terminando sua estadia), deve haver a entrada de uma camareira para limpeza do quarto e **os hóspedes só podem retornar após o fim da limpeza**;
- Um quarto vago que passa por limpeza não pode ser alocado para um hóspede novo;
- Caso uma pessoa chegue e não tenha quartos vagos, **ele deve esperar em uma fila** até algum quarto ficar vago. Caso a espera demore muito, ele passeia pela cidade e retorna após um tempo para tentar alugar um quarto novamente;
- Caso a pessoa tente duas vezes alugar um quarto e não consiga, ele deixa uma reclamação e vai embora.

Observações:

- Não há a possibilidade de, para um mesmo quarto, somente parte dos hóspedes saírem para passear. Ou saem todos ou nenhum;
- **A implementação deve ser abrangente e simular várias situações:** número diferentes de hóspedes chegando, grupos com mais de 4 pessoas, todos os quartos lotados, etc.
- **Atentem-se para a descrição de cada regra!! Deve haver sincronia e coordenação entre as entidades.**

Critérios de avaliação

O projeto vale no máximo 4 pontos na nota do bimestre (N1), sendo que a avaliação será baseada em dos tipos de critérios:

Coletivos (3 pontos):

- **Criação das entidades propostas:** 0.4 (0.1 por entidade);
 - Correta implementação de cada classe e suas operações;
- **Implementação das funcionalidades/regras :** 2 (0.2 por regra);
 - **Serão considerados a lógica utilizada e a corretude das operações que cada entidade deve conter.**
 - Para as regras dependentes de outras, a nota integral só será atribuída caso o conjunto de regras funcione corretamente.
- **Estruturação e organização do código:** 0.5;
 - Divisão do projeto em classes;
 - Modularização do código, utilizando métodos sempre que possível;
 - Organização/clareza do código (nomes significativos de variáveis, indentação, etc.);
- **Organização do GitHub (README, comentários e tamanho dos commits, etc.)** (0.1);

Individuais (1 ponto):

- Participação no desenvolvimento do projeto (quantidade e qualidade dos commits e proporção de commits feitos por cada membro) (0.8);
- Participação na apresentação final do projeto (0.2).

Observações

- **O trabalho deve ser feito em grupo, mas qualquer tipo de plágio/cola será penalizado (o projeto receberá nota 0).**
- **Projetos com erro de sintaxe ou que não possam ser executados irão receber nota 0 também**
- A entrega será dia 06/05, até 23:59, pelo AVA.
- O envio deve conter apenas um link para o repositório (que deve estar público) com o código do projeto.
 - **NÃO ENVIEM UM ZIP OU COLOQUEM O PROJETO ZIPADO NO GITHUB!**