

## 2º. Trabalho de POO

**Data de Entrega:** 03/06/2025

**Natureza do Trabalho:** em grupo de 2 alunos

**Forma de Entrega:** Pasta do projeto compactada. A pasta do projeto deve conter os arquivos gerados pela ferramenta Netbeans 8.2. A pasta que vai conter os arquivos do trabalho deve ser nomeada da seguinte forma:

**prjPOO****N1N2**

onde **N1N2** representa os nomes dos integrantes

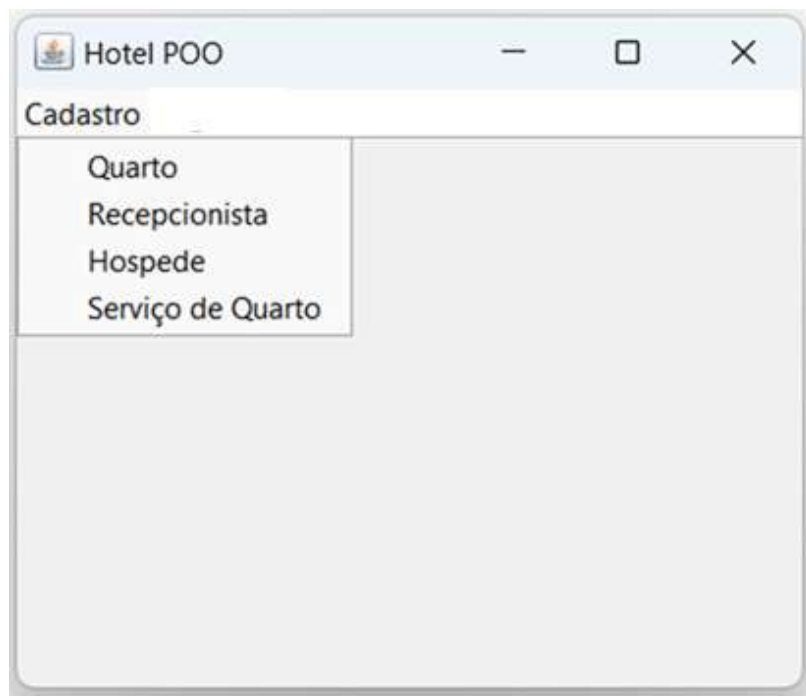
Exemplo: **prjPOOChristinaVinicius**

**Enunciado:**

- A partir da modelagem do diagrama de classe do 1º. Trabalho faça a implementação das classes view, a implementação das classes control e a implementação das operações de consulta, inclusão, alteração e exclusão de cada cadastro. Faça as implementações solicitadas de acordo com as especificações abaixo.

## 1) Implementação das classes View

- Menu de Opções



- Cadastro Quarto

The screenshot shows a window titled 'Cadastro Quarto' with standard Windows window controls. The form contains the following elements:

- Nº Quarto:** A text input field.
- Valor Diária:** A text input field.
- Tipo:** A group box containing two radio buttons: 'Solteiro' (selected) and 'Casal'.
- Action Buttons:** A row of five buttons at the bottom: 'Consultar' (with a magnifying glass icon), 'Inserir' (with a plus icon), 'Alterar' (with a pencil icon), 'Excluir' (with a minus icon), and 'Sair' (with a door exit icon).

- Cadastro Recepcionista

The screenshot shows a software window titled "Cadastro Recepcionista". It contains several input fields: "Registro Funcional", "Nome", "Endereço", and "Telefone". Below these is a "Turno" section with three radio buttons labeled "Manhã", "Tarde", and "Noite". At the bottom, there is a row of five buttons: "Consultar" (with a magnifying glass icon), "Inserir", "Alterar" (with a pencil icon), "Excluir", and "Sair" (with a yellow exit icon).

- Serviço de Quarto

The screenshot shows a software window titled "Cadastro Serviço de Quarto". It contains three input fields: "Código", "Descrição" (which is a dropdown menu currently showing "Água"), and "Valor". At the bottom, there is a row of five buttons: "Consultar" (with a magnifying glass icon), "Inserir", "Alterar" (with a pencil icon), "Excluir", and "Sair" (with a yellow exit icon).

- Cadastro Hóspede

The screenshot shows a software window titled "Cadastro Hóspede". It contains five input fields: "CPF" (with a mask of dots and dashes), "Nome", "Endereço", "Telefone", and "Taxa de desconto (%)". At the bottom, there is a row of five buttons: "Consultar" (with a magnifying glass icon), "Inserir", "Alterar" (with a pencil icon), "Excluir", and "Sair" (with a yellow exit icon).

## Considerações:

- Inserir as Gui's no pacote **fatec.poo.view**
- Copiar a pasta **icon** para dentro da pasta view
- A entrada de dados para o cpf do hóspede deve ser feita utilizando a classe JFormattedTextField com a respectiva formatação **###.###.###-##**
- O combobox do serviço de quarto deve ser preenchido com os seguintes itens: Água, Doce, Lanche, Refrigerante, Salgado
- Fazer a programação para os eventos ActionPerformed para as opções de cada **Item de Menu**.
- Fazer a programação para os eventos ActionPerformed para o botão **Sair** de cada Gui.

## 2) Implementação das classes Control

- Insira a classe PreparaConexao do projeto **prjExemplo\_OO\_GUI\_BD** no pacote fatec.poo.control
- Faça a implementação das classe DaoHospede, DaoRecepcionista, DaoQuarto, DaoServicoQuarto. Insira as classes no pacote **fatec.poo.control**

## 3) Funcionalidades que devem ser implementadas na aplicação

### 3.1) Implementação das tabelas numa base de dados Oracle

- Faça a implementação das tabelas tblHospede, tblRecepcionista, tblQuarto e tblServicoQuarto no SGBD Oracle de acordo com o script TrabPOO.sql

### 3.2) Implementação das Operações de Cadastro

- Copie o arquivo **ojdbc6\_g.jar** do projeto **prjExemplo\_OO\_GUI\_BD** para a pasta projeto do trabalho e em seguida, utilizando o NetBeans, faça a inclusão deste arquivo no projeto (**Definir Configuração** → **Personalizar** → **Bibliotecas** → **Adicionar Jar** )
- Faça a implementação das operações Consulta, Inclusão, Alteração e Exclusão dos cadastros de Hospede, de Recepcionista, de Quarto e de ServicoQuarto.

#### 3.2.1) Cadastro de Quarto

- Na operação Consultar caso seja informado um Número de Quarto inválido (qualquer valor diferente de um dado numérico inteiro) exiba uma mensagem de advertência e reposicione o foco na entrada de dados para o Número de Quarto. Caso o Número de Quarto seja válido, o Número de Quarto deve ser pesquisado na tabela de quarto. Caso seja encontrado o Quarto na base de dados, o objeto da classe Quarto deve ser instanciado e o conteúdo dos seus atributos devem ser exibidos na GUI, o botão Consultar deve ser desabilitado e os botões Alterar e Excluir devem ser habilitados, a caixa de edição do Número de Quarto deve ser desabilitada e os demais recursos de interface devem ser habilitados. O foco deve ser posicionado na caixa de edição valor diária. Caso não exista na base de dados um Quarto com o código informado, o botão Consultar deve ser desabilitado e o botão Incluir deve ser habilitado. A caixa de edição do Número de Quarto deve ser desabilitada e os demais recursos de interface devem ser habilitados. O foco deve ser posicionado na caixa de edição valor diária.
- Faça a implementação das operações de Consulta, Inclusão, Alteração e Exclusão. Utilize como referência o projeto **prjExemplo\_OO\_GUI\_BD**.

### 3.2.2) Cadastro de Recepcionista

- Na operação Consultar caso seja informado um Registro Funcional inválido (qualquer valor diferente de um dado numérico inteiro) exiba uma mensagem de advertência e reposicione o foco na entrada de dados para o Registro Funcional. Caso o Registro Funcional seja válido, o Registro Funcional deve ser pesquisado na tabela de recepcionista. Caso seja encontrado o Recepcionista na base de dados, o objeto da classe Recepcionista deve ser instanciado e o conteúdo dos seus atributos devem ser exibidos na GUI, o botão Consultar deve ser desabilitado e os botões Alterar e Excluir devem ser habilitados, a caixa de edição do Registro Funcional deve ser desabilitada e os demais recursos de interface devem ser habilitados. O foco deve ser posicionado na caixa de edição nome. Caso não exista na base de dados um Recepcionista com o Registro Funcional informado, o botão Consultar deve ser desabilitado e o botão Incluir deve ser habilitado. A caixa de edição do Registro Funcional deve ser desabilitada e os demais recursos de interface devem ser habilitados. O foco deve ser posicionado na caixa de edição nome.
- Faça a implementação das operações de Consulta, Inclusão, Alteração e Exclusão. Utilize como referência o projeto **prjExemplo\_OO\_GUI\_BD**.

### 3.2.3) Cadastro de Serviço de Quarto

- Na operação Consultar caso seja informado um Código inválido (qualquer valor diferente de um dado numérico inteiro) exiba uma mensagem de advertência e reposicione o foco na entrada de dados para o Código. Caso o Código seja válido, o Código deve ser pesquisado na tabela de ServicoQuarto. Caso seja encontrado o ServicoQuarto na base de dados, o objeto da classe ServicoQuarto deve ser instanciado e o conteúdo dos seus atributos devem ser exibidos na GUI, o botão Consultar deve ser desabilitado e os botões Alterar e Excluir devem ser habilitados, a caixa de edição do Código deve ser desabilitada e os demais recursos de interface devem ser habilitados. O foco deve ser posicionado no combobox Descricao. Caso não exista na base de dados um ServicoQuarto com o Código informado, o botão Consultar deve ser desabilitado e o botão Incluir deve ser habilitado. A caixa de edição do Código deve ser desabilitada os demais recursos de interface devem ser habilitados. O foco deve ser posicionado no combobox Descricao.
- Faça a implementação das operações de Consulta, Inclusão, Alteração e Exclusão. Utilize como referência o projeto **prjExemplo\_OO\_GUI\_BD**.

### 3.2.3) Cadastro de Hóspede

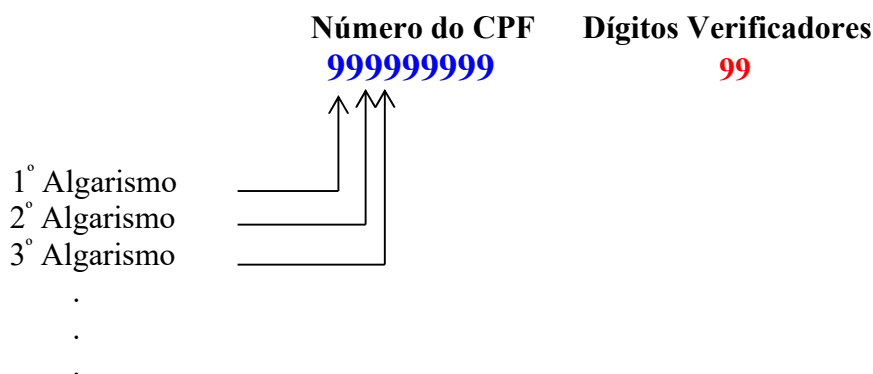
- Na operação Consultar caso seja informado um CPF inválido exiba uma mensagem de advertência e reposicione o foco na entrada de dados para o CPF. Caso o CPF seja válido, o cpf do hóspede deve ser pesquisado na tabela de hospede. Caso seja encontrado o hóspede na base de dados, o objeto da classe hospede deve ser instanciado e o conteúdo dos seus atributos devem ser exibidos na GUI, o botão Consultar deve ser desabilitado e os botões Alterar e Excluir devem ser habilitados, a caixa de edição do CPF deve ser desabilitada e os demais recursos de interface devem ser habilitados. O foco deve ser posicionado na caixa de edição nome. Caso não exista na base de dados um hospede com o cpf informado, o botão Consultar deve ser desabilitado e o botão Incluir deve ser habilitado. A caixa de edição do CPF deve ser desabilitada e os demais recursos de interface devem ser habilitados. O foco deve ser posicionado na caixa de edição nome. **O cpf deve ser inserido na base de dados sem a formatação da entrada de dados.**
- O CPF deve ser validado, para tal, na classe Hospede defina um **método de classe** que tenha como parâmetro de entrada o CPF (sem a formatação da entrada de dado) e tem como retorno um valor booleano (true – cpf válido. False-cpf inválido).

```
public static boolean validarCPF(<string>){..... return(<boolean>)};
```

O CPF deve ser validado conforme os critérios definidos abaixo:

### Forma do cálculo dos Dígitos Verificadores:

O CPF é um número composto por 9 algarismos. Os dígitos verificadores são calculados com base nos valores e na posição sequencial destes algarismos.



Inicialmente calcula-se o valor do primeiro dígito, que será utilizado posteriormente para o cálculo do segundo dígito.

### Cálculo do 1º Dígito Verificador

1. Multiplicar cada um dos 9 dígitos iniciais pelo número correspondente a sua posição no CPF, e somar os resultados de cada multiplicação:

$$(\text{Algarismo1} * 1) + (\text{Algarismo 2} * 2) + (\text{Algarismo 3} * 3) + \dots + (\text{Algarismo 9} * 9)$$

2. Calcular o resto da divisão do valor obtido na soma por 11. Esse resto é o primeiro dígito verificador e, caso seja igual a 10, considerá-lo como sendo o número zero.

### Cálculo do 2º Dígito Verificador

- 3) Multiplicar cada um dos 10 algarismos disponíveis (os 9 do CPF mais o primeiro dígito verificador já obtido) de acordo com a seguinte regra: O primeiro (o mais à esquerda) por 11, o segundo por 10, o terceiro por 9, sucessivamente até o último (o mais à direita, que é o primeiro dígito verificador) que será multiplicado por 2, e somar os resultados de cada multiplicação:

$$(\text{Algarismo1} * 11) + (\text{Algarismo2} * 10) + (\text{Algarismo3} * 9) + \dots + (1^\circ \text{ Dígito Verificador} * 2)$$

1. Multiplicar o valor obtido com a soma por 10 e, em seguida, calcular o resto da divisão desse resultado por 11. Esse resto é o segundo dígito verificador e, caso seja igual a 10, considerá-lo como sendo o número zero.

- CPFs com todos os números iguais (111.111.111-11, 222.222.222-22, ....) deve ser considerados inválidos.
- Faça a implementação das operações de Consulta, Inclusão, Alteração e Exclusão. Utilize como referência o projeto **prjExemplo\_OO\_GUI\_BD**.