



| | | | | |
|---|-----|---|-------------------|---|
|  | | Código: EC-P10305 Disciplina: Programação Orientada a objetos N2 1º bimestre Curso: <u>EC</u> Turma: 3 18/09/2023 - 19:15 Prof.(ª): Leide Vieira Coord.: Rodrigo Fontes | |  |
| Aluno (a): | Nº: | RA: | Nota: | |
| | | | | |
| Orientações: Duração: 90 minutos Individual: <input checked="" type="checkbox"/> Sim <input type="checkbox"/> Não | | | | |
| Data da vista da prova: 25 / 09 / 2023 - horário: 19 h 15 . Declaro estar de acordo com a nota e com a revisão e correção realizada pelo professor. O não comparecimento do aluno na vista dos instrumentos avaliativos, na data prevista, significa a concordância tácita com as notas atribuídas, desobrigando a instituição de deferir eventuais pedidos de revisão de notas (após a divulgação de notas no portal do aluno, dentro do semestre letivo ou em situações futuras). | | | Rubrica do aluno: | |

Desenvolvimento de uma aplicação para Salão de Beleza / Barbearia

Objetivo: Criar um programa de salão de beleza ou barbearia para aplicar conceitos de programação orientada a objetos, diagrama de classes, bem como tratamento de erros.

Etapas:

Diagrama: Elaborar um diagrama de classes, considerando, ao menos, três classes: Cliente, Funcionário e Agendamento. Cada classe deve ter atributos adequados para seu propósito e métodos que auxiliem nas operações do salão de beleza e/ou barbearia.

Implementação das Classes: criar as classes no código-fonte do projeto. Definir os atributos apropriados e os métodos necessários para cada classe com base no diagrama de classes desenvolvido.

Menu Interativo: elaborar um menu interativo que permita a interação com o programa, incluindo a capacidade de cadastrar clientes, funcionários e agendar serviços. Tratar possíveis exceções para garantir que o programa não falhe.

Entregáveis:

Diagrama de classes em formato de imagem ou PDF;
 Projeto completo compactado em ZIP ou RAR.

Critérios de avaliação:

Aplicação de Conceitos Aprendidos: Verifique se o código aplica os conceitos abordados na disciplina.

Funcionalidade: Verificar se o código cumpre o propósito especificado e se resolve o problema proposto de maneira correta;

Eficiência: Avalie a eficiência do código em termos de uso de recursos computacionais, como tempo de execução e uso de memória.

Comentários e Documentação: Verifique se o código inclui comentários claros e explicativos, bem como documentação que descreva a finalidade e o funcionamento do programa.

Legibilidade e Manutenção: Analise a clareza e a legibilidade do código. É bem estruturado e documentado? Outros programadores podem entender e manter o código com facilidade?

Boas Práticas de Programação: Verifique se o código segue as boas práticas de programação, como nomes de variáveis significativos, formatação consistente e uso adequado de estruturas de controle.

Reutilização de Código: Avalie se o código aproveita bibliotecas e módulos existentes quando apropriado, em vez de reinventar a roda.

Inovação e Criatividade:** Valorize abordagens inovadoras e soluções criativas para problemas complexos.

Estilo Pessoal:** Considere a autenticidade do estilo de programação, desde que esteja alinhado com as boas práticas.