

 MAUÁ	P2 – 2ª Prova Anual		
	1º semestre - 2025		
Disc.: LINGUAGEM DE PROGRAMAÇÃO I	Cód. Disciplina: ECM251		
Curso: ENGENHARIA DA COMPUTAÇÃO			

Aluno: Pedro Wilian Palumbo Bevilacqua		Test Code: ZLLCJ
Curso: ECM	Série 3ª Período: Matutino	
São Caetano do Sul, 17 de junho de 2025.	RA: 23.01307-9	
Assinatura: Pedro Wilian Palumbo Bevilacqua	Nota:	

INSTRUÇÕES GERAIS

- Esta prova é individual e prática, devendo ser realizada nos computadores do IMT, sendo permitido ao aluno, se assim desejar, utilizar seu próprio computador, sob sua inteira responsabilidade;
- Não poderá haver acesso à Internet, sob nenhuma circunstância, exceto ao CanvasLMS do próprio aluno e, mesmo assim, somente em duas ocasiões: para receber (“baixar”) as questões de sua prova na máquina em uso; e para entregar (“subir”) as resoluções das questões de sua prova, no devido local de entrega, na mesma plataforma.
IMPORTANTE: O aluno deverá manter sua máquina em “Modo Avião” o tempo todo ativado, exceto quando informar ao professor que fará os dois referidos acessos permitidos ao CanvasLMS pela Internet;
- Poderá haver consulta a qualquer material do próprio aluno, seja ele físico (livros, artigos, material de aula etc.) ou virtual (livros, artigos, material de aula, exercícios, resoluções etc.), desde que esse material esteja previamente armazenado em seu computador ou qualquer dispositivo de armazenamento externo (*pendrive*, *drive* externo etc.).
IMPORTANTE: Não será permitido o acesso pela Internet a pastas compartilhadas (*Google Drive*, *OneDrive* etc.), nem a repositórios virtuais (*GitHub*, *GitLab*, *BitBucket* etc.), mesmo sendo de posse e administração do próprio aluno;
- O aluno deverá responder às questões da prova gerando os códigos necessários na I.D.E. de sua preferência, na linguagem de programação JAVA e utilizando sempre e somente o paradigma da Programação Orientada à Objetos visto nas aulas;
- Para realizar a entrega da prova na plataforma CanvasLMS, o aluno deverá gerar e entregar um único arquivo compactado (*.rar* ou *.zip*), tendo seu RA e seu NOME completo como nome desse arquivo. Nesse arquivo compactado o aluno deverá fornecer, obrigatoriamente, os seus dados acadêmicos preenchidos na folha de prova (*.docx*) e os arquivos e códigos gerados em suas resoluções, uma pasta para cada questão, contendo as suas classes e demais arquivos que possibilitem sua posterior execução pelo professor, durante a resolução.
IMPORTANTE: Na correção, para executar o código gerado pelo aluno em sua prova, o professor seguirá exatamente as instruções fornecidas pelos alunos e contidas nas resoluções das provas! Caso não obtenha sucesso, a questão será considerada errada.
- Não poderá haver troca de informações, nem de materiais, sejam físicos ou virtuais, entre os alunos durante a prova;
- Não é permitido ao estudante se ausentar da sala antes da entrega da prova;
- Celulares, *Smartphones* e outros equipamentos eletrônicos devem permanecer desligados enquanto o estudante estiver na sala;
- O tempo limite para realização da prova é de **90** minutos;
- Mantenha sobre a carteira um documento com foto, caneta, lápis e borracha;
- O entendimento das questões faz parte da avaliação;
- O tempo mínimo de permanência na sala é de **30** minutos;
- O estudante que chegar atrasado em até **30** minutos do início da prova poderá fazê-la.
- O aluno que não preencher, ou não preencher corretamente o campo “Test Code” do cabeçalho com o código correto, fornecido a ele durante a prova, terá sua resolução de prova invalidada, sendo sua nota automaticamente 0 (ZERO).

Questão única: (1+1+2+2+2+2 = 10 pontos)

Fornecer os códigos fontes, em *Java*, das classes necessárias, com sua respectiva classe de teste, implementadas sob o paradigma da Programação Orientada a Objetos, além do programa *.jar* que as executa, incluindo sua interface gráfica do usuário – GUI, orientada a menu, geradas manualmente por qualquer IDE, para resolver o seguinte problema:

Requisitos: Implementar a aplicação “*CEM – Consumo de Energia Mensal*” para um usuário de uma residência, conforme a figura e as informações abaixo, contendo e executando as funcionalidades solicitadas pela GUI.

Na GUI, os retângulos preenchidos na cor branca representam os locais onde o usuário irá digitar as informações, dos campos que serão usados, para os devidos cálculos. Vide fórmula abaixo.

Na GUI, os retângulos preenchidos na cor cinza representam os locais onde serão apresentados ao usuário os cálculos espontâneos e instantâneos, realizados pela aplicação, utilizando os valores lidos dos campos na cor branca. Vide fórmula abaixo.

Quando o usuário clicar sobre o item “*Opção*”, ou através da digitação do seu mnemônico, a aplicação deverá abrir um menu com os itens “*Salvar no B.D.*”, “*Ler do B.D.*” e “*Sair*”.

Se o usuário clicar no item “*Salvar no B.D.*”, ou através da digitação do seu mnemônico, a aplicação lerá os valores dos campos textos da GUI: “*Nome do Cliente*”; “*Valor do kWh (R\$)*”; “*Cômodo*”; e “*Consumo*” dos 3 (três) cômodos nela especificados, armazenando-os no B.D., usando como campo chave um *ID* atrelado ao “*Nome do Cliente*” lido.

Se o usuário clicar no item “*Ler do B.D.*”, ou através da digitação do seu mnemônico, a aplicação lerá os valores armazenados no B.D. para o “*Nome do Cliente*” especificado no campo da GUI e os colocará nos respectivos campos textos da GUI: “*Nome do Cliente*”; “*Valor do kWh (R\$)*”; “*Cômodo*”; e “*Consumo*” dos 3 (três) cômodos nela especificados.

Se o usuário clicar no item “*Sair*”, ou através da digitação do seu mnemônico, ou no botão “*X*” da janela da GUI, a aplicação deverá fechar sua GUI e encerrar o programa.

O campo “*Test Code*” da GUI deverá apresentar o código da prova recebido pelo aluno, no ato da assinatura da lista de presença da prova, armazenando-o no próprio código da aplicação *Java* (em *hard code*) e não no B.D..

O Banco de Dados – B.D. utilizado e conectado ao aplicativo *Java* deverá ser o *MySQL*.

Dados: - “*Nome do Cliente*”; “*Valor do kWh (R\$)*”; “*Cômodo*” (campos 1, 2 e 3); e “*Consumo*” (campos 1, 2 e 3), devem ser do tipo *Campo Texto*, digitados pelo usuário;

- “*Custo (R\$)*” (campos 1, 2 e 3) devem ser do tipo *Campo Texto*, calculados e apresentados pela aplicação;

- “*Test Code*” deve ser do tipo *Campo Texto*, lido diretamente no código *Java* (*hard code*) e apresentado na GUI pela aplicação *Java*;

- Cálculo: $\text{Custo (R\$)} = \text{Valor do kWh (R\$)} * \text{Consumo (kWh)}$.

Cômodo	Consumo (kWh)	Custo (R\$)
Sala	100.00	50.00
Cozinha	50.00	25.00
Quarto	10.00	5.00

ATENÇÃO:

- I. O índice de acerto da questão será diretamente proporcional à aparência e às funcionalidades por ela solicitadas;
- II. Lembrar que o professor tentará utilizar, primeiramente, o arquivo *.jar*, os *scripts* (*.sql*) do B.D. e as instruções fornecidas pelo aluno (usuário, senha, base de dados, tabelas e campos) para a conexão com o B.D), para executar este programa pelo *Prompt de Comando* do *Windows 10/11*, durante a fase de correção da questão;
- III. Caso o professor não obtenha sucesso na execução do item II, o professor tentará, então, executar as classes fornecidas pelo aluno em *Java*, através de sua IDE *VSCode*, no lugar de usar o arquivo *.jar*;
- IV. Critérios de correção – Para o caso de:
 - a. O arquivo *.jar* da aplicação ser fornecido e estar funcional: até 1 ponto;
 - b. O arquivo *.sql* do B.D. ser fornecido, estar funcional e aderente à aplicação solicitada: até 1 ponto;
 - c. A *GUI* implementada estar aderente à *GUI* solicitada: até 2 pontos; se o campo “*Test Code*” da *GUI* não apresentar o respectivo código fornecido ao aluno este item será invalidado (0 ponto);
 - d. As funcionalidades da *GUI* implementada estarem aderentes às da *GUI* solicitada: até 2 pontos;
 - e. As tabelas do B.D. implementadas estarem aderentes às tabelas solicitadas: até 2 pontos;
 - f. A conexão ao B.D. do aplicativo implementado estar funcional e aderente à solicitada: até 2 pontos;