

Módulo 2 - Projeto Avaliativo 1

Peso: 45% da nota do módulo 2

DEVinHouse

Sumário

1. Introdução	1
2. Requisitos da Aplicação	1
3. Entrega	2
4. Critérios de Avaliação	2
5. Roteiro da aplicação	4
6. Regras de Negócio	5
7. Plano de Projeto	6

1. Introdução

Durante as últimas semanas aprendemos muito sobre a **linguagem C#**, e em muitas ocasiões nossas práticas e exercícios tinham características do que a indústria de software se refere como back-end, aprendendo sobre tipos de dados, variáveis, Arrays, Listas, e Conceitos aplicados à Programação Orientada a Objetos.

Que tal mostrar para o pessoal da **Softplan** que com o que aprendemos é possível sim construir um sisteminha 100% funcional, sem depender de nenhuma outra tecnologia para exibição !? Vamos construir uma pequena **fintech**.

2. Requisitos da Aplicação

A aplicação deverá ser realizada **individualmente**, necessitando contemplar os seguintes requisitos:

1. O sistema deverá ser desenvolvido em C#;
2. O sistema deve seguir o Roteiro da Aplicação;
3. O desenvolvimento das mensagens de saída da fintech, esperando pela ação do usuário;
4. Captura da interação do usuário via entrada padrão;
5. O sistema deverá ser apresentado diretamente na linha de comando ou API;

Observe que os atributos não possuem os modificadores explícitos (private, public, package-private, static e final), ficando a critério do aluno determinar qual nível de acesso ou modificador deve ser aplicado à classe, atributo ou método.

3. Entrega

O código desenvolvido deverá ser submetido no [GitHub](#), e o link deverá ser disponibilizado na tarefa **Módulo 2 - Projeto Avaliativo 1**, presente na semana 6 do AVA até o dia **20/03/2022** às **23h55**.

O repositório deverá ser **privado**, com as seguintes pessoas adicionadas:

- Pablo Monteiro - [phmonte](#)
- Cesar Abascal - [cesar-abascal](#)
- Fernando Puntel - [fepuntel](#)

Entregas realizadas após a data limite sofrerão **decréscimo na nota** de avaliação, sendo considerado **80%** da nota para tarefas submetidas até o dia **27/03/2022** às **23h55**. Não serão avaliados projetos submetidos após a data de término da atividade.

Importante: Será considerado como data de entrega a **última atualização** no repositório do projeto no GitHub. Lembre-se de não modificar o código até receber sua nota.

4. Critérios de Avaliação

A tabela abaixo apresenta os critérios que serão avaliados durante a correção do projeto. O mesmo possui variação de nota de 0 (zero) a 10 (dez) como nota mínima e máxima, e possui peso de **45% sobre a avaliação do módulo 2**.

Serão **desconsiderados e atribuída a nota 0 (zero)** os projetos que apresentarem plágio de soluções encontradas na internet ou de outros colegas. Lembre-se: Você está livre para utilizar outras soluções como base, mas **não é permitida** a cópia.

Nº	Critério de Avaliação	0	0,25 a 0,5	0,75 a 1
1	O aluno seguiu as convenções de código da linguagem?	O aluno não deixou seu algoritmo organizado.	O aluno deixou seu algoritmo parcialmente organizado.	O aluno deixou seu algoritmo organizado.

2	O aluno apresentou um menu de fácil interação ao usuário?	O aluno não apresentou o menu.	O aluno apresentou o menu, contudo, de forma confusa.	O aluno apresentou o menu de forma clara para o usuário.
3	O aluno utilizou o conceito de heranças?	O aluno não utilizou o conceito de herança.	O aluno utilizou, parcialmente, os conceitos de heranças.	O aluno utilizou de forma correta o conceito de herança.
4	O aluno criou tratamentos de erros seguindo o descrito no roteiro da aplicação?	O aluno não realizou tratamento de erros	O aluno realizou o tratamento de erros, porém não de forma aderente aos princípios da linguagem e da orientação à objetos	O aluno realizou o tratamento de erros corretamente por meio de estruturas específicas e reutilizáveis
Nº	Critério de Avaliação	0	0,25 a 1	1,25 a 2
5	O aluno implementou todas as classes com seus respectivos atributos e métodos?	O aluno não implementou as classes.	O aluno implementou de forma parcial as classes, atributos e métodos	O aluno implementou todas as classes, juntamente com seus atributos e métodos.
6	O aluno implementou e apresentou os relatórios solicitados?	O aluno não implementou os relatórios.	O aluno implementou e apresentou, de forma parcial, os relatórios.	O aluno implementou e apresentou todos os relatórios.
7	O aluno implementou todas as operações bancárias e a classe de transações?	O aluno não implementou as operações bancárias e nem a classe de transações.	O aluno implementou, de forma parcial, as operações bancárias e nem a classe de transações.	O aluno implementou todas as operações bancárias e nem a classe de transações.

5. Roteiro da aplicação

A fintech **DEVinBank** deseja automatizar todo o seu sistema de armazenamento de informações referentes aos seus clientes. O sistema deve conter os seguintes tipos de contas, cada uma com suas características:

- **Conta corrente**

- Na conta corrente o cliente tem direito ao cheque especial, ou seja, poderá ficar negativo durante um período de tempo. O sistema deve definir o total do cheque especial, conforme a renda mensal do correntista (10% da renda mensal).
- Extrato das transações

- **Conta poupança**

- Na conta poupança o cliente não tem direito ao cheque especial.
- Na conta poupança, o cliente poderá simular quanto o seu valor renderá em um determinado tempo, para isso, o cliente deve informar a quantidade de tempo (em meses) e a rentabilidade anual da poupança.
 - Por exemplo, se o cliente deseja saber a rentabilidade em 6 meses é preciso calcular o rendimento proporcional ao valor mensal.
- Extrato das transações

- **Conta investimento**

- Neste tipo de conta, o cliente poderá escolher um tipo de investimento e o sistema deve apresentar o rendimento anual do investimento solicitado, conforme os valores abaixo:
 - LCI: 8% ao ano.
 - Tempo mínimo de aplicação: 6 meses
 - LCA: 9% ao ano.
 - Tempo mínimo de aplicação: 12 meses
 - CDB: 10% ao ano.
 - Tempo mínimo de aplicação: 36 meses
- O cliente pode realizar uma simulação do valor aplicado, para isso, ele deve indicar:
 - Valor que será aplicado.
 - Tempo que ficará aplicado o valor (em meses).
 - Valor da CDI atual, caso escolha LCI ou LCA.
 - Ao final da simulação deve-se apresentar uma mensagem para o cliente perguntando se ele deseja efetuar o investimento.
- Extrato das transações. Neste caso, sempre que o cliente efetuar uma transação deve-se armazenar:
 - Valor aplicado.
 - Tipo da aplicação.
 - Data da aplicação.
 - Data da retirada do valor.
- Obs. o rendimento do valor aplicado sempre será diário. Por exemplo, se escolher uma aplicação que renda 10% ao ano, é preciso verificar o rendimento diário sobre o valor aplicado.

6. Regras de Negócio

Todas as contas devem ser derivadas da classe Conta, que possui os seguintes atributos e métodos:

- **Conta**
 - Atributos
 - Nome
 - CPF (é necessário validar o CPF)
 - Endereço
 - Renda mensal
 - Conta (o sistema deverá gerar um número da conta de forma sequencial)
 - Agência
 - Atualmente o banco possui duas agências:
 - 001 - Florianópolis
 - 002 - São José
 - 003 - Biguaçu
 - Obs. quando o cliente criar uma nova conta o sistema deve apresentar em qual das agências sua conta estará vinculada.
 - Saldo
- **Métodos**
 - Saque
 - Depósito
 - Saldo
 - Extrato
 - Transferência
 - Alterar dados cadastrais (Exceto CPF)

A fintech também deseja manter um histórico das transferências, que deverá armazenar (utilizar conceitos de composição):

- Dados Conta Origem
- Dados Conta Destino
- Valor
- Data (pegar a data e hora do sistema)

O sistema também deverá apresentar os seguintes relatórios:

- Listar todas as contas
 - Correntes
 - Poupanças
 - ou Investimento
- Contas com saldo negativo
- Total do valor investido
- Todas as transações de um determinado cliente.

É importante que algumas transações não possam ser executadas em caso de problemas percebidos em suas operações:

- Transferência entre contas cujo montante supera o saldo acrescido do limite do cheque especial da conta de origem
- Operações em momentos anteriores ao dia/hora da transação
- Transferências durante o final de semana (sábado ou domingo)
- Não é possível fazer transferências para si próprio

Observação: O sistema deve iniciar no dia do seu sistema operacional e você deve pegar essa informação de forma automática. Para simular os valores de investimentos, faça uma função que adiante o tempo no seu algoritmo. Por exemplo: Adiantar em 1 ano o sistema bancário.

7. Plano de Projeto

Ao construir o projeto, o aluno estará colocando em prática os aprendizados em:

- **POO:** Conceitos fundamentais da programação orientada a objetos, tais como classes, objetos, métodos, atributos, construtores, modificadores, encapsulamento, sobrecarga, herança e polimorfismo.
- **C#:** Sintaxe da linguagem, representação de entidades, entrada/saída de informações via terminal, tipos de dados, arrays, lista e tratamento de exceções.