

Parte 01

Professor: Edwar Saliba Júnior

Valor: 20 pontos

Contextualização:

Você e sua equipe desenvolveram na disciplina de APOO (Análise e Projetos Orientados a Objetos), um diagrama de classes UML que é o esboço de um sistema que será desenvolvido. Este diagrama já deve ter sido mostrado e avaliado em sala de aula pelo Professor e demais alunos presentes. Agora chegou a hora de desenvolver o sistema proposto. Assim sendo, de acordo com o diagrama proposto pela equipe, deverá ser desenvolvido o sistema, em linguagem Java com componentes Swing, usando a **JDK 11** e tudo que vocês aprenderam na disciplina até então.

Observação: na descrição a seguir foi usada a palavra vetor para se referir a uma estrutura que possibilita o cadastro diversos objetos na memória do computador. Contudo, você não é obrigado a usar vetor em seu software, poderá usar qualquer uma das *Collections* que foram apresentadas na disciplina.

- 1) sua equipe deverá seguir fielmente os nomes das classes, os nomes dos atributos, tipos e métodos que foram propostos no diagrama apresentado em APOO para este trabalho. Ou seja, o diagrama de classes de APOO deve ser um espelho do núcleo do sistema que será desenvolvido. Eventualmente pode ter acontecido algum tipo de inconsistência na análise feita na disciplina de APOO e, conseqüentemente, esta inconsistência pode ter se refletido no diagrama de classes, por exemplo: por meio de um atributo com o tipo errado ou mesmo faltando ou sobrando atributos, falta ou excesso de métodos e etc. Caso isto tenha acontecido com sua equipe, não tem problema! Basta que vocês mantenham o diagrama de classes sempre atualizados e, ao entregarem o sistema na disciplina de PV também entreguem a nova versão do diagrama de classes na disciplina de APOO. Caso não aconteça, haverá dura penalização para a equipe;
- 2) o software deverá ser todo visual e orientado a objetos;
- 3) para facilitar o desenvolvimento do sistema, sua equipe deverá implementar o *software*, em princípio, utilizando *ArrayList* para armazenamento, em memória principal, dos objetos criados pelos usuários. E qualquer tipo de pesquisa e manipulação do *ArrayList* deverá ser feita utilizando-se um objeto "Iterator";
- 4) para facilitar manutenções futuras, exige-se o padrão MVC (*Model, View, Controller*) para implementação do *software*. O assunto MVC não será lecionado neste curso, sendo parte do trabalho a pesquisa e utilização do conceito por parte do aluno;
- 5) o *software* deverá também, possuir uma tela de login com usuário e senha e deverá fazer um controle de usuários. Sendo que a senha deverá ficar criptografada (criptação de mão única) e arquivada em arquivo binário ou texto. O assunto "criptografia" e o assunto "arquivos" tampouco serão apresentados nesta disciplina, sendo parte do trabalho a pesquisa e utilização dos conceitos por parte dos alunos;

- 6) uma versão prévia e funcional deste *software* deverá ser apresentada até o dia 18/05/2023. Podendo ser apresentada em qualquer data antes desta. Esta versão não precisa ter o controle de usuário implementado ainda.
- 7) ATENÇÃO: o seu *software* só deverá possuir um JFrame, que será a tela principal, as outras telas deverão ser de um ou mais tipos a seguir: JDialog, JInternalFrame, Application Sample, MDI Application Sample ou Ok / Cancel Dialog Sample. O JFrame da tela Principal deverá possuir um componente JMenu, que possibilitará o acesso as outras telas dos *software*;
- 8) todos os cadastros deverão possibilitar o uso das 5 operações básicas: inclusão, exclusão, alteração, consulta e relatório;
- 9) Regras para a entrega do trabalho:
- deverá ser apresentado e entregue, o projeto (compactado) do código-fonte;
 - o código-fonte que será entregue, apresentado e não deverá possuir nenhum tipo de comentário;
 - antes de compactar o seu projeto de *software*, lembre-se de apagar a pasta "dist", caso ela exista na estrutura de pastas do projeto e também os arquivos com extensão ".class";
 - tudo deverá ser enviado para o e-mail: eddiesaliba3@gmail.com (de acordo com as regras que serão apresentadas abaixo);
 - o e-mail deverá ser enviado até, no máximo, um dia antes da data de apresentação e
 - os trabalhos enviados após a data limite serão avaliados com 5 pontos a menos para cada dia de atraso.
- 10) Regras para envio do e-mail:
- trabalhos enviados após a data marcada para entrega, serão avaliados em 5 pontos a menos para cada dia de atraso;
 - a data para entrega via e-mail é um dia antes da data de apresentação do trabalho no laboratório ou sala de aula;
 - a apresentação deverá ser realizada no laboratório onde houver aula de PV;
 - caso alguma equipe possua alguma restrição ou dificuldade no cumprimento desta regra, então, deverá avisar ao professor com antecedência mínima de 24 horas da data de apresentação;
 - no campo assunto do e-mail deve constar apenas o título: **IFTM - PV - 2023/02 - Trabalho 01**
 - no corpo do e-mail deverá conter, única e exclusivamente, o nome completo de todos os membros da equipe (um em cada linha) e
 - só será aceito UM e-mail por equipe. Portanto, verifique se está tudo certo com seu e-mail e trabalho antes de enviá-lo.

Observação: o desrespeito a qualquer das regras ou tarefas acima implicará na perda de créditos.

Critérios de avaliação do software:

- conformidade do software em relação ao solicitado;
- legibilidade do código (organização, indentação e etc.);
- usabilidade das interfaces do software e
- entendimento individual a respeito do código-fonte apresentado.