



GBC035 - Programação Orientada a Objetos 1

Trabalho Final de Implementação

1. Objetivo

Construir um sistema computacional que incorpore os conceitos de Orientação a Objetos.

2. Temas

Segue uma lista com sugestões de temas para os sistemas. Os alunos devem propor requisitos adicionais às especificações apresentadas:

- a) **Jogo clássico:** A aplicação deve implementar algum jogo clássico. Exemplos de jogo incluem Tetris (https://youtu.be/eu_Gqi6ysVk), Space Invaders (<https://youtu.be/MU4psw3ccUI>), Arkanoid (<https://youtu.be/Th-Z6QQ5AOQ>), dentre outros. Qualquer outro jogo clássico poderá ser implementado, mas deve-se ter em mente as dificuldades associadas a cada um deles, e que podem ser impeditivos para o tempo de realização do projeto. Além disso, os jogos podem apresentar modificações sugeridas pelo grupo, de forma a “personalizar” o jogo;
- b) **App de gerenciamento de agenda e compatibilização de Reuniões** (ex: <https://www.when2meet.com/>): A aplicação deve fazer a gestão de eventos agendados de diversos usuários. Os usuários devem ser cadastrados no sistema. Qualquer usuário pode criar um evento, adicionando todas as informações necessárias (descrição, material associado, dentre outros), informando também possíveis datas, e compartilha esse evento com outros usuários interessados, que então precisam encontrar um dia (dentre os sugeridos) nos quais há disponibilidade para o evento. Todos os usuários podem consultar, a qualquer momento, os melhores horários, e então decidir quando será o evento. O sistema possui uma agenda na qual os eventos são cadastrados, e deve permitir aos usuários consultar sua agenda diária/semanal/mensal. O sistema pode avisar sobre os eventos já agendados e que estão próximos de acontecer.
- c) **App de Gestão de ensino remoto** (ex: Teams): A aplicação deve fazer a gestão dos professores, suas turmas e os alunos de cada turma. Além disso, o professor deve cadastrar uma aula, e para essa aula cadastrar todo o material necessário (teoria + exercícios) e uma área de mensagens trocadas durante essa aula. Mensagens podem também ser trocadas entre os integrantes da turma. Devem ser cadastradas tarefas avaliativas ou não, e computadas as faltas nas aulas. A qualquer momento, o sistema deve gerar um relatório de faltas nas aulas já ministradas, bem como a nota dos alunos nas atividades avaliativas já computadas.
- d) **App de Gestão de clínica médica:** A aplicação deve fazer a gestão dos médicos, pacientes e medicamentos utilizados, que devem todos ser cadastrados previamente. Cada médico tem uma especialidade e pode ou não ser afiliado a um ou mais convênios. Além disso, obedece a uma agenda que contém os dias e horários disponíveis para atendimento. Deve ser possível ao médico consultar sua agenda diária/semanal. Os pacientes também são cadastrados no sistema, sendo necessário armazenar seus dados pessoais. Esse paciente, ao procurar a clínica, agenda um horário com o especialista escolhido, e durante a consulta, o médico deve sempre registrar os batimentos cardíacos, pressão e temperatura, o diagnóstico, qual a medicação utilizada, e demais observações pertinentes. A aplicação deve fornecer relatórios com o histórico de determinado paciente (considerando um período determinado pelo usuário), de forma que um médico possa acompanhar a evolução do mesmo, bem como saber o histórico de doenças passadas. Além disso, a aplicação deve fornecer um relatório ao próprio médico, que permita visualizar as consultas realizadas em determinado período escolhido por ele. Finalmente, a aplicação deve fornecer um relatório com o histórico de utilização dos medicamentos disponíveis.

- e) **App para um jogo de Perguntas e Respostas - quiz:** o sistema controla o gerenciamento e aplicação de competições de perguntas e respostas, de diferentes tópicos e assuntos. Usuários devem ser cadastrados, tanto os administradores, que criarão os jogos, quanto os jogadores. Os administradores devem ser capazes de cadastrar perguntas, juntamente com as respostas possíveis e as respostas corretas. As perguntas serão de determinado assunto, e deverão seguir uma classificação de acordo com sua dificuldade. Os jogos serão compostos por um conjunto de perguntas de determinado(s) assunto(s). Deve haver uma pontuação associada às perguntas, e outras regras poderão ser adicionadas também ao jogo como um todo. O número de participantes do jogo também deverá ser definido, além do número de rodadas. Os participantes poderão desistir de responder determinada pergunta, e uma penalidade será aplicada nesse caso. O número de vitórias, bem como a pontuação em cada jogo, deverão ser associadas aos jogadores, de forma a ser possível construir um ranking.
- f) **App de Gestão de Agência de Viagens** (ex: <https://www.decolar.com/>): A aplicação deve fazer a gestão dos usuários, trechos, hospedagens e atrações. Os usuários cadastrados enviam uma requisição de viagem, envolvendo data, destino e número de pessoas, e o sistema monta um roteiro sugerido incluindo transporte até o local, hospedagens e passeios incluídos. É possível no entanto ao usuário alterar o roteiro sugerido. O sistema deve emitir as passagens, montar um relatório e enviar ao cliente, de acordo com a viagem escolhida. O sistema deve gerar relatórios com dados estatísticos básicos organizados por roteiro, por data, por idade, dentre outros.
- g) **Os alunos poderão sugerir um outro tema diferente dos propostos, que passará por uma avaliação de viabilidade pelo professor.**

3. Regras do trabalho

- Os alunos deverão fazer o trabalho em grupos de 4 pessoas. Dado o alto número de alunos matriculados, não serão aceitos grupos com menos de 4 pessoas, a não ser que o total de alunos não permita a divisão exata. As listas com os nomes dos integrantes dos grupos, juntamente com o tema escolhido, deverão ser enviados via plataforma Microsoft Teams, na data especificada na **Seção 6**. Após essa data, não será permitido o envio de listas com os nomes dos integrantes, e os alunos que não estiverem em nenhum grupo ficarão com zero na nota final, sem chance para entrega do mesmo;
- O software deverá ser construído em Java, utilizando os conceitos de Orientação a Objetos vistos em sala de aula. **Não basta que o software funcione, ele deve ser construído UTILIZANDO OS CONCEITOS DE ORIENTAÇÃO A OBJETOS. Não é necessário que todos os conceitos sejam utilizados, mas será avaliado a BOA UTILIZAÇÃO dos recursos empregados;**
- O trabalho será desenvolvido em duas fases:
Fase 1: especificação detalhada do sistema, contendo descrição dos requisitos, funcionalidades previstas e regras de negócio, bem como um diagrama de classes. Com exceção do diagrama de classes, não há um padrão para o documento, o grupo deve inserir todas as informações que achar pertinente;
Fase 2: sistema completo.
- Não é obrigatório o uso de bancos de dados no sistema, mas se o grupo optar por utilizar um, deverá ser o PostgreSQL (<https://www.postgresql.org/>).

4. Entrega

- As entregas dos trabalhos referentes às duas fases serão feitas via plataforma Microsoft Teams, na data especificada na **Seção 6**;
- Caso o grupo não entregue nenhum trabalho na Fase 01, ficará com ZERO no trabalho inteiro, e não terá a chance de entregar o restante do mesmo na Fase 02;
- O professor não se responsabiliza por falhas na submissão, falhas no MICROSOFT TEAMS, e QUALQUER OUTRO PROBLEMA RELACIONADO. É de responsabilidade do aluno não deixar a entrega para última hora, e com isso se prevenir de tais problemas. Não serão aceitas,**

posteriormente, nenhuma justificativa, e qualquer e-mail de tentativa de justificativa por parte dos alunos será descartado.

- d. O material referente à Fase 01 deverá ser entregue em formato .pdf;
- e. O sistema pronto, referente à Fase 02, deverá ser transmitido em um arquivo compactado (.zip ou .rar) contendo TRÊS PASTAS:
 - i. PASTA 01: Códigos-fonte que compõem o software, ou projeto COMPLETO da IDE utilizada;
 - ii. PASTA 02: O código-fonte compilado (APENAS arquivos .jar), de forma que seja possível executar o software sem necessidade de compilação;
 - iii. PASTA 03: Documentação, contendo:
 - 1. Qualquer biblioteca ou pacote utilizado, **com documentação de como instalá-la, configurá-la e utilizá-la**. Além disso, ao utilizar uma biblioteca, o grupo deverá explicar quais elementos dela são utilizados no sistema, e como são utilizados para o funcionamento do mesmo;
 - 2. Script para geração do banco de dados, caso utilizado no projeto;
 - 3. Arquivo texto contendo os nomes dos integrantes do grupo.

5. Avaliação

- a. O trabalho vale 50 pontos, e será avaliado da seguinte maneira:
 - **NOTA-SOFTWARE → 50,0**
 - i. Estrutura do software em relação à Orientação a objetos → 30,0;
 - ii. Funcionamento do sistema e corretude das funcionalidades presentes → 15,0 (**caso o programa não execute, a nota referente a esse critério é automaticamente zerada**);
 - iii. Documentação → 5,0.
 - **NOTA-APRESENTAÇÃO → [0 - 1]**
 - i. Os grupos deverão fazer uma apresentação a respeito do software entregue, mostrando seu funcionamento e respondendo a perguntas sobre sua estrutura;
 - ii. A apresentação será feita em duas fases. Na primeira, os alunos mostrarão o sistema funcionando, sem mostrar nenhuma linha de código. Na segunda, o professor questionará cada integrante do grupo separadamente sobre o código. Caso a apresentação seja feita por video conferência, as câmeras deverão permanecer ligadas;
 - iii. **Os grupos que não entregarem o software NÃO TERÃO DIREITO À APRESENTAÇÃO;**
 - iv. Apenas os integrantes cujos nomes estiverem no arquivo entregue junto ao software terão direito à apresentação, e serão avaliados individualmente;
 - v. **Alunos que não apresentarem o trabalho ficarão com nota total do trabalho igual a 0, e a entrega do software não será considerada nesse caso.**

NOTA FINAL = NOTA-SOFTWARE x NOTA-APRESENTAÇÃO

6. Datas Importantes

- a. Entrega da lista de integrantes e tema do trabalho: **31/03/2023, ATÉ AS 23:59**
- b. Entrega da Fase 01: **12/04/2023, ATÉ AS 23:00;**
- c. Entrega da Fase 02: **07/06/2023, ATÉ AS 18:00;**
- d. Apresentação: **12/06/2023 à 19/06/2023.**