



## **Trabalho Final (40 pontos)**

### **Objetivo**

Construir, em grupos de até três estudantes, um sistema OO sobre um determinado tema.

### **Tema**

O tema da aplicação deverá ser escolhido entre os integrantes de cada grupo, e postado no MS Teams, num tópico criado no fórum exclusivamente para definição dos temas, até a data limite de 24/02/2025 às 23:59. Devido ao alto número de discentes na disciplina, só serão aceitos grupos com dois ou 3 integrantes. Após essa data, não será permitido o envio de listas com os nomes dos integrantes, e discentes que não estiverem em nenhum grupo ficarão com zero na nota final, sem chance para entrega do trabalho.

Atenção: Os grupos que não postarem até a data limite de 24/02/2025 o seu tema, não terão direito à entrega e à avaliação do mesmo, obtendo conceito zero no trabalho. Com isso, cada grupo deverá incluir uma postagem no tópico contendo uma descrição do projeto. Os temas poderão se repetir apenas uma vez, então cada grupo deve observar no tópico quais já foram escolhidos. Sendo assim, recomenda-se agilidade na definição do mesmo. Se necessário, poderá ser solicitada a alteração do tema. Basta que um(a) aluno(a) de cada grupo submeta o tema, incluindo, além dos itens anteriores, a relação dos integrantes.

Algumas sugestões de temas para os sistemas, devendo propor requisitos adicionais às especificações apresentadas:

**1. 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.

**2. 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.

**3. 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.

**4. 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.

5. Os grupos poderão sugerir um outro tema diferente dos propostos, que passará por uma avaliação de viabilidade pelo professor.

### **Regras**

- A turma deverá se organizar em grupos de no máximo três pessoas. Não será permitida alteração (ex.: saída de aluno, substituição etc.) do grupo, após divulgado seu respectivo tema no MS Teams.
- O trabalho deve ser realizado sobre os conceitos de Orientação a Objetos. 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;

*Observação: não será permitida a utilização de bibliotecas e ferramentas de terceiros. O mesmo deve ser composto somente de código próprio e/ou classes fornecidas pelo Java(ex.: collections).*

- Não é permitido o uso de frameworks, padrões de projeto e bancos de dados. Arquivos-texto podem ser utilizados, desde que sejam apenas para carregar dados previamente cadastrados no sistema. Em outras palavras, o sistema não deve realizar operações diretamente no arquivo, ou depender totalmente dele para sua execução;
- O trabalho será desenvolvido em duas fases:
  - Fase 1 (07/03): 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 (11/04): implementação de 50% do que foi descrito na Fase 1;
- Fase 3 (02/05, 05/05 e 06/05): Sistema completo.

### **Entrega**

- As entregas dos trabalhos referentes às duas primeiras fases serão feitas via plataforma Microsoft Teams, na data especificada acima;
- Caso o grupo não entregue nenhum trabalho na Fase 01 e na Fase 02, ficará com ZERO no trabalho inteiro, e não terá a chance de entregar o restante do mesmo na Fase 03;
- O professor não se responsabiliza por falhas na submissão, falhas no MICROSOFT TEAMS, e QUALQUER OUTRO PROBLEMA RELACIONADO. É de responsabilidade do(a) discente 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.
- O material referente à Fase 01 deverá ser entregue em formato .pdf;
- O sistema parcialmente pronto e totalmente pronto, referente as Fases 02 e 03, deverão ser transmitido em um arquivo compactado (.zip), cujo nome deve ser GrupoXX.zip, onde XX deverá ser substituído pelo número do grupo, de acordo com a tabela a ser disponibilizada após definição dos grupos. O arquivo deverá conter TRÊS PASTAS:
  - PASTA 01: Códigos-fonte que compõem o software, ou projeto COMPLETO da IDE utilizada. No caso do NetBeans, deverá incluir toda a pasta do projeto (incluindo arquivos build e manifest, bem como as pastas build, nbproject, src e test; ou equivalentes se usar outra IDE, como eclipse). Execute um “clean” (limpeza) no projeto, antes de compactá-lo para envio.
  - PASTA 02: O código-fonte compilado (APENAS arquivos .jar), de forma que seja possível executar o software sem necessidade de compilação. Esse arquivo deve estar SEPARADO da pasta citada no item anterior;
  - PASTA 03: Documentação, contendo: 1. Qualquer biblioteca ou pacote externo utilizado (CASO SEJA 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. Arquivos adicionais contendo os dados (ex.: dados de entrada) para funcionamento correto do programa deverão ser também incluídos, quando for o caso; 3. Arquivo texto contendo os nomes dos integrantes do grupo.