



### INFORMAÇÕES SOBRE A AVALIAÇÃO

LAB05	Laboratório 05 - 20 pontos
-------	----------------------------

INFORMAÇÕES DOCENTE						
CURSO: ENGENHARIA DE SOFTWARE	DISCIPLINA: LABORATÓRIO DE DESENVOLVIMENTO DE SOFTWARE	TURNO	MANHÃ	TARDE	NOITE	PERÍODO/SALA: 4º
			x		x	
<b>PROFESSOR (A):</b> João Paulo Carneiro Aramuni						

### Sistema de Moeda Estudantil (Release 3) - Versão final

Pretende-se desenvolver um sistema para estimular o reconhecimento do mérito estudantil através de uma moeda virtual. Essa moeda pode ser distribuída por professores aos seus alunos e trocada pelos alunos por produtos e descontos em empresas parceiras. Após a análise inicial de requisitos do sistema, foram levantadas as seguintes informações:

#### Descrição do Sistema:

Os alunos que desejam ingressar no sistema de mérito devem realizar um cadastro, indicando nome, email, CPF, RG, Endereço, Instituição de Ensino e curso. As instituições participantes já estão pré-cadastradas no sistema, para que o aluno selecione.

Os professores já estarão pré-cadastrados no sistema (a instituição envia a lista no momento da parceria). Cada professor terá armazenado o seu nome, CPF e departamento que está vinculado. É necessário deixar explícito que ele faz parte de uma instituição.

A cada semestre, os professores receberão um total de mil moedas, que podem ser distribuídas aos seus alunos como forma de reconhecimento por bom comportamento, participação em aula, etcs. Esse total é acumulável no semestre (isto é, se o professor não distribuir todas as moedas num semestre, o total de 1.000 novas moedas será adicionado ao seu saldo corrente).

Para enviar moedas, o professor deve possuir saldo suficiente, indicando qual aluno deverá receber o montante, bem como o motivo pelo qual ele está sendo reconhecido (uma mensagem aberta, obrigatória).



## PUC Minas

Ao receber uma moeda, o aluno deve ser notificado por email.

Professores e alunos devem ser capazes de consultar o extrato de sua conta, visualizando o total de moedas que ainda possui, bem como as transações que realizou (para o professor, o envio de moedas; para o aluno, recebimento ou troca de moedas).

Para trocar moedas, o aluno deve selecionar uma das vantagens cadastradas no sistema. Elas incluem, por exemplo: desconto em restaurantes da universidade, desconto de mensalidade, ou compra de materiais específicos.

Empresas que sejam realizar parceria também devem se cadastrar no sistema, incluindo as vantagens que deseja oferecer e o custo de cada uma dela (em moedas).

Para cadastrar uma vantagem, a empresa parceira deve adicionar também uma descrição e foto do produto.

Ao resgatar uma vantagem, o aluno deve ter o valor descontado do seu saldo. Um email de cupom deve ser enviado para que ele utilize na troca presencial. Um email também deve ser enviado ao parceiro, para que ele possa conferir a troca. Ambos os emails devem incluir um código gerado pelo sistema, a fim de facilitar o processo de conferência.

Por fim, alunos, professores e empresas parceiras precisam ter um login e uma senha cadastrados para acessar o sistema. Em todos os casos, um processo de autenticação é necessário para realização dos requisitos.

### Apresentação Final:

Ao final da última sprint (Sprint 03), os alunos deverão apresentar o protótipo produzido para a **Release 3**, comparando-os com os modelos descritos inicialmente, bem como apresentando as modificações inseridas para o funcionamento adequado do software (conforme a especificação anterior).

O sistema deverá ser desenvolvido utilizando arquitetura MVC, e atendendo aos requisitos apresentados e cumprindo a modelagem produzida nas sprints iniciais do projeto. O repositório GitHub deve estar atualizado, com todas as versões produzidas dos modelos UML e o código final desenvolvido. A avaliação final do projeto levará em consideração os seguintes aspectos:



## PUC Minas

- Qualidade do sistema produzido (adequação aos requisitos apresentados);
- Alinhamento entre modelo (de classes e de arquitetura) e código;
- Atualizações dos modelos conforme necessidade do projeto.

### Processo de Desenvolvimento:

#### Release 01:

**Lab03S01:** Modelagem do sistema: Diagrama de Casos de Uso, Histórias do Usuário, Diagrama de Classes e Diagrama de Componentes.

**Lab03S02:** Modelo ER, definição e implementação da estratégia de acesso ao banco de dados (ORM, Padrão DAO, etc), CRUDs de aluno e empresa parceira (versão inicial: front-end e comunicação com back-end).

**Lab03S03:** CRUDs de aluno e empresa parceira (versão final), apresentação da arquitetura do sistema e camada de persistência (obs.: preparar slides).

#### Release 02:

**Lab04S01:** Implementação dos casos de uso de envio de moedas e consultas de extrato (para professores e alunos). Enviar email com a confirmação do envio e recebimento de moedas (template para professor e template para aluno). Demonstração via vídeo se não for possível reproduzir o ambiente no laboratório.

**Lab04S02:** Diagrama de Sequências e implementação dos casos de uso de cadastro de vantagens (pela empresa parceira) e listagem de vantagens (para o aluno).

**Lab04S03:** Diagrama de Sequências e implementação dos casos de uso de troca de vantagens (pelo aluno).

#### Release 03:

**Lab05S01:** Diagramas de comunicação e de implantação. Implementação dos casos uso de envio de cupons por e-mail. O cupom deve incluir um QR Code único gerado automaticamente pelo sistema. Além disso, a aplicação deve ser implantada na nuvem, utilizando preferencialmente Vercel e Render.

**Lab05S02:** Análise crítica dos projetos e refatorações:

Cada grupo deve realizar a análise crítica de um projeto da turma.



**PUC Minas**

Sugestão de sorteio de grupos:

- G1 → Analisar e refatorar projeto G2
- G2 → Analisar e refatorar projeto G3
- G3 → Analisar e refatorar projeto G4
- G4 → Analisar e refatorar projeto G5
- G5 → Analisar e refatorar projeto G1

A análise consiste na entrega de um **relatório de análise crítica do projeto**, discutindo:

- (i) arquitetura e tecnologias utilizadas
- (ii) organização do GitHub
- (iii) dificuldade para configuração do ambiente
- (iv) sugestões de melhorias

Além disso, o grupo deve identificar três partes do código que podem ser refatoradas, visando melhorar a compreensão e manutenção do sistema. A **entrega das refatorações deve ser realizada via sistema de pull requests do GitHub**, e documentadas no relatório (mostrando e descrevendo o código antes e depois das mudanças, refatorações aplicadas, e justificativa).

**Os proprietários do projeto não precisam aceitar as contribuições.**

Utilize o template de **relatório de análise crítica do projeto** disponível aqui:  
[https://github.com/joaopauloaramuni/laboratorio-de-desenvolvimento-de-software/blob/main/TEMPLATES/template\\_report.md](https://github.com/joaopauloaramuni/laboratorio-de-desenvolvimento-de-software/blob/main/TEMPLATES/template_report.md)

Para este projeto, utilize o template de **README.md** disponível aqui:  
[https://github.com/joaopauloaramuni/laboratorio-de-desenvolvimento-de-software/blob/main/TEMPLATES/template\\_README.md](https://github.com/joaopauloaramuni/laboratorio-de-desenvolvimento-de-software/blob/main/TEMPLATES/template_README.md)

Prazo final (Acesse o cronograma): <https://github.com/joaopauloaramuni/laboratorio-de-desenvolvimento-de-software/tree/main/CRONOGRAMA>

Valor total: 20 pontos | Desconto de 1.0 ponto por dia de atraso | **Entregas em atraso não serão aceitas, assim como, implementações parciais (ex.: apenas front-end ou back-end).**

Atenção: semanalmente, todos os grupos deverão apresentar o andamento das entregas durante a aula. A não participação do grupo implicará na perda automática de 50% dos pontos da sprint.