

## **Projeto – Comércio Eletrônico**

**Disciplina:** Tópicos Avançados

**Curso:** Tecnologia em Análise e Desenvolvimento de Sistemas

**Professore:** Sirlon Thiago Diniz Lacerda

### **RESUMO E OBJETIVOS**

O projeto tem como objetivo expor o aluno aos desafios de uma aplicação web utilizando Ruby on Rails e MongoDB a fim de prepará-lo para diversas situações que possam surgir durante sua experiência profissional no mercado de trabalho.

Deve ser feito, desta forma, o desenvolvimento de uma aplicação de Comércio Eletrônico.

### **REQUISITOS PARA APROVAÇÃO E INSTRUÇÕES GERAIS**

1. O projeto deve ser desenvolvido de forma **individual**.
2. O aluno deverá desenvolver o projeto conforme os requisitos apresentados a seguir.
3. O código-fonte deverá estar em um repositório público no GitHub.
4. 20% da nota (2 pontos) serão dados para o código-fonte, onde serão avaliados requisitos como (mas não limitado a): legibilidade, utilização de conceitos vistos em sala de aula, utilização de boas práticas, utilização de padrões de projeto etc.
5. No dia 13 de fevereiro de 2024, o aluno deverá estar presente em sala de aula/laboratório para que seja feita a arguição oral. O professor irá clonar o repositório e rodar a aplicação localmente. Nesse momento, o aluno será arguido sobre o código escrito, práticas escolhidas, dentre outros assuntos pertinentes à disciplina. A atividade de arguição oral valerá 80% da nota (8 pontos).
6. Será verificado se os requisitos e regras de negócio foram implementadas na aplicação.
7. **NÃO** será avaliada a estética do produto, mas sim funcionalidade.



8. A seguir serão detalhados os itens que são obrigatórios e opcionais.
9. Códigos-fonte que não puderem ser testados serão desconsiderados e a atividade de entrega do código-fonte será ZERADA.
10. Não estar presente no dia da arguição oral automaticamente indicará desistência e a atividade de arguição oral será ZERADA.
11. A não entrega dos códigos-fonte implica na desistência da disciplina, pois, não será feita arguição de códigos-fonte não disponíveis no GitHub.

## SISTEMA DE COMÉRCIO ELETRÔNICO

O documento a seguir é dividido em:

1. **Requisitos do sistema.** São obrigatórios para composição da nota da disciplina e serão verificados individualmente se foram implementados na aplicação.
2. **Regras de Negócio.** São obrigatórias para composição da nota da disciplina e serão verificadas individualmente se foram implementadas na aplicação.
3. **Extras.** Requisitos que valerão nota extra aos alunos que conseguirem executá-los. NÃO são, portanto, obrigatórios para composição da nota da disciplina.
4. **Desafios.** Funcionalidades/requisitos para os alunos que decidirem se aventurar e expandir seus conhecimentos em Java e Banco de Dados. Implicará na aplicação de conhecimentos que podem não ter sido abordados em sala de aula. NÃO são, portanto, obrigatórios para composição da nota da disciplina.

## REQUISITOS

- O sistema deve permitir que, em sua tela inicial, o usuário possa visualizar uma lista de produtos.
- O sistema deve permitir que o usuário possa acessar uma tela que lhe permita escolher se quer acessar o sistema ou cadastrar-se (recomenda-se utilizar a gem devise para autenticação).



- O sistema deve permitir que o usuário possa colocar um produto no carrinho.
- O sistema deve permitir que, após feito o login, o usuário possa comprar os produtos do carrinho. Na finalização da compra, deve ser gerado um pedido que deve ser gravado no banco de dados.
- O sistema deve tratar as mensagens de erro e devolvê-las de forma amigável ao usuário.
- O cadastro de produtos deve ser feito via seeder.
- As fotos do produto podem ser fixas (placeholders), não sendo necessário fazer nenhum upload.

### REGRAS DE NEGÓCIO

- E-mails devem ser únicos. O acesso ao sistema deve ser feito por meio do endereço de e-mail.
- Para finalização do pedido, o sistema deverá utilizar o MercadoPago como meio de pagamento.

### EXTRAS

- Acrescentar uma funcionalidade que permita ao usuário pesquisar pelo nome e descrição de um produto.
- Acrescentar uma funcionalidade que permita que o usuário possa alterar seus dados.
- Acrescentar uma funcionalidade de lista de pedidos onde o usuário pode, ao abrir tal lista, verificar os itens e valores de cada pedido que ele realizou.
- Estilizar a aplicação com Bootstrap e/ou outros frameworks afins.

### DESAFIO

- Acrescentar paginação na lista de produtos de forma que o usuário possa navegar de forma mais fácil.