

Prof. Dr. Jean M rio Moreira de Lima
jean.lima@imd.ufrn.br

ATIVIDADE AVALIATIVA: PROJETO DA UNIDADE I

I. Informes

1. O projeto da unidade ser  em grupo de at  4 pessoas. Recomendo que fa am em grupo para dividir as diversas tarefas!
2. O grupo deve criar um reposit rio em alguma plataforma git aberta (GitHub, GitLab) para trabalhar no projeto de maneira **colaborativa**.
3. **APRESENTA O OBRIGAT RIA: 09/04/2022**. O grupo dever  preparar uma breve apresenta o (.pptx / .pdf) e apresentar o objetivo do projeto, bem como as funcionalidades, destacando os recursos implementados relativos aos crit rios de avalia o descritos no item III. O tempo de apresenta o   de no m ximo 8 min. **O grupo que n o apresentar, ter  a nota zerada.**
4. **PARTICIPA O OBRIGAT RIA**: TODOS os componentes do grupo devem estar presentes e cada um vai explicar a parte que ficou respons vel. **O componente do grupo que estiver ausente ter  a nota zerada.**
5. **FORMA DE ENTREGA**: Exclusivamente **via SIGAA**, **arquivo com**:
 - a. Nomes e Matr culas dos componentes do grupo.
 - b. Link do reposit rio do projeto (Github, GitLab) (certificar-se que h  permiss o de acesso)
6. **NOTA**: 0,00 a 10,00.
 - a. Obs: Peso de 80% na nota da unidade.

II. Proposta de Projeto

Desenvolver um sistema web *backend* com p ginas renderizadas no lado do servidor por algum *template engine*. Utilize o framework de sua prefer ncia - **recomendo: SPRING** e o *template engine* de sua prefer ncia - **recomendo: Thymeleaf, Handlebars ou Mustache**. O sistema deve ter funcionalidade expl cita. O grupo precisa validar a ideia do projeto, durante aula, via email ou discord com o professor. Obs: N o ser o aceitas propostas iguais  s que foram desenvolvidas em aula ou nos mini-projetos!

III. Crit rios de Avalia o

1. O sistema deve estar funcional e com funcionalidade explícita. Deve dividir, de maneira adequada, componentes importantes da aplicação, criando seus próprios pacotes e subdiretórios. Apresentação e propostas adequadas **(1,00 ponto)**.
2. O sistema deve fazer extenso uso dos recursos de persistência e mapeamento objeto-relacional. *Recomendo a utilização do Spring Data Jpa com JpaRepository*. Deve fazer uso de, no mínimo, **5 entidades que serão mapeadas em tabelas**. Deve existir relacionamentos entre as tabelas, pelo menos um relacionamento de cada tipo: um-para-um, um-para-muitos (muitos-para-um), e muitos-para-muitos. Pelo menos um dos relacionamentos feitos deve ser bidirecional. Sua aplicação deve habilitar todas as operações de CRUD para pelo menos 4 entidades. **(4,00 pontos)**.
3. Sua aplicação deve seguir a arquitetura MVC. Para cada modelo operacional, implemente o respectivo controller, que deve ser o responsável pelas requisições e direcionamento das respostas (páginas carregadas) e os *services* para desacoplar sua aplicação das classes repositórios ou de regras de negócio. **Pelo menos duas classes de repositórios devem conter queries de busca personalizadas** **(2,00 pontos)**.
4. Personalização e Usabilidade: Utilize recursos de javascript e css para personalizar sua aplicação. Você pode utilizar *frameworks de layout* como Bootstrap e Materialize para personalizar suas páginas. **(3,00 pontos)**.