



# Técnico em Informática - Integrado

## DESENVOLVIMENTO WEB II

### Trabalho Avaliativo 01: Projeto com Laravel

#### *Código-Fonte PHP de Referência utilizado nas aulas*

#### 1. Descrição

O trabalho proposto tem por objetivo aplicar de forma prática o conteúdo lecionado durante a disciplina utilizando o [Laravel Framework](#) para criação de sistemas web. Utilizando seus recursos como: *Rota, Controller, View, Blade, Model, Migration, Seeder, Eloquent, Session*) e *Bootstrap* ou *Tail Wind CSS* (Layout). O aluno ou dupla deve apresentar o tema (sistema) que pretende desenvolver, respondendo através da postagem feita no fórum do *Moodle*. Não será aceito o mesmo tema de um aluno ou dupla, **não** é permitido a construção de trios, ou mais. Neste sistema, o aluno ou dupla deverá seguir os critérios avaliativos descritos no item 4.

#### 2. Definição do Conceito

Seguir as funcionalidades globais do Laravel Framework, que estão condicionados ao uso obrigatório dos seguintes conceitos:

- *View + Controller*;
- *Model + Migration*;
- *Seeder* (Inserir dados de testes usados durante o desenvolvimento da aplicação);
- Associar chave estrangeiras com tabelas em formulários e exibir nos grids (Tabela HTML);
- Um **bom Layout\*** com [Bootstrap](#) ou [Tail Wind CSS](#);

#### 3. Método de Avaliação

Os conceitos apresentados devem ser seguidos obrigatoriamente, durante o desenvolvimento da aplicação, e serão avaliados pelo professor no sentido de verificar se o aluno foi capaz de empregá-los adequadamente, ou seja, ele os utilizou quando realmente eram necessários e funcionaram como esperado dentro da aplicação.

Os mesmos serão avaliados considerando se o aluno foi capaz de contemplar totalmente a funcionalidade especificada durante a proposta. Por exemplo, o item 4, para funcionalidade (f) do documento, o sistema é realmente capaz de **vincular as relações com as chaves estrangeiras**. Caso o sistema não contemple a funcionalidade, serão desconsiderados por não seguir o critério avaliativo.

Além disso, será avaliado o perfeito funcionamento da aplicação, ou seja, tudo que foi proposto se está funcionando corretamente, tanto através da apresentação ou gravação do vídeo demonstrando o sistema, como na conferência do código-fonte e banco de dados enviado. As entradas de dados da aplicação devem ser filtradas não permitindo que o usuário gere “bugs”, como campos de entradas vazios ou com valores incorretos. Além do mais, todas as possíveis ações danosas por parte do usuário devem ser previstas.

#### 4. Critérios Avaliativos

**Trabalho Avaliativo (8 pontos - 80% da nota): Desenvolvimento do projeto em etapas**

**Etapla 1 (2 pontos): Estruturação, Layout e Banco de Dados**

- Configuração do ambiente Laravel e estruturação das rotas iniciais.
- Implementação de um layout adequado utilizando Bootstrap ou Tailwind CSS (**1 ponto**).
- Banco de Dados – (**1 Ponto**)
  - Modelagem do Banco de Dados;
  - **Migração:** Deve ser criado pelo menos 3 arquivos de migração;
  - **Seeders e Factory:** As tabelas deve ter o seu arquivo *Seeders* e *Factory* para inserir os registros dos dados;
  - Ao realizar o comando `php artisan migrate --seed`. Deve ser criado as tabelas do banco de dados e realizado a inserção dos registros base do sistema.

**Etapla 2 (3 pontos): CRUDs**

- Desenvolvimento de no mínimo três CRUDs, cada um contendo as operações de *Salvar, Listar, Buscar, Atualizar e Deletar* (**2,5 pontos**).
  - Cada CRUD deve conter no mínimo três campos;
  - Cada listagem deve ter um formulário com pelo menos um campo de busca;
- Validação de todos os campos obrigatórios do sistema (**0,5 ponto**).

**Etapla 3 (3 pontos): Refinamento e finalização dos três CRUDs**

- Finalização do CRUDs e Layout (**1 ponto**);
- Relacionamentos usando chaves estrangeiras em pelo menos dois CRUDs – (**2 pontos**);
  - Em formulários trazer em dados através das tags *Selects*;
  - Na listagem, invés mostrar o id, mostrar o nome do atributo da chave estrangeira;
- Testes para garantir o funcionamento correto e ajustes visuais.

**Entrega Final:**

- Apresentação de 5 a 10 minutos a aplicação e o código-fonte defesa do projeto, incluindo desafios enfrentados e soluções aplicadas.

**Atividades Alternativas (2 pontos - 20% da nota):**

- **Demonstração Prática:** Alunos apresentam o código escrito e explicam o funcionamento do projeto durante as aulas;
- **Organização dos arquivos e código-fonte:** garantindo que a aplicação não apresente erros durante o uso.

#### 5. Observações do trabalho e sobre o seu envio:

- Trabalhos com os campos dos CRUDs bem parecidos, mudando apenas o layout do projeto com de outra dupla, será zerado os 2 trabalhos.**
- O códigos-fonte e o script do banco de dados do projeto devem ser compactado em um único arquivo com a extensão **.zip** com o nome\_sobrenome\_aluno01\_nome\_sobrenome\_aluno02.zip. E o nome do aluno ou dupla dentro de um arquivo **.txt com o link do repositório do projeto no Github**.
- Compartilhar o projeto em uma pasta no Google Drive e dar permissão de **acesso externo** seguindo a opção. (**Qualquer pessoa com o link pode ver. Na opção Gerar link, ir em compartilhado e em seguida - qualquer pessoa com o link pode ver**);
- A pasta deve conter o projeto compactado no formato **.zip** e o **vídeo\*** com explicação do projeto;
- Se o trabalho for individual colocar o nome\_sobrenome\_aluno, se for em duplo nome\_sobrenome\_aluno01 e nome\_sobrenome\_aluno02 em um arquivo **.txt** dentro da pasta junto com o projeto compactado e o **vídeo**;
- Caso seja enviado o link da pasta compactada sem a permissão informada no **item b**, será atribuído a **nota zero** para o aluno ou dupla.
- Colocar o **script** do banco de dados com a extensão **.sql na pasta database do projeto**;
- Caso não seja enviado o script do banco, será atribuído a **nota zero**.

### \* Observações sobre a gravação do Vídeo

Itens a serem apresentados no vídeo, deve seguir o seguinte roteiro

- Breve introdução explicando a ideia do projeto, com apresentação do(s) aluno(s) e uma visão geral do sistema;
- **Apresentar cada parte do projeto de acordo com os critérios avaliativos, de forma resumida;**
- Mostrar em até 1 minuto a Modelagem e o Banco de Dados;
- Mostrar o sistema funcionando simulando o seu uso;
- Se for dupla, todos os integrantes devem se apresentar falando o seu nome;
- Cada integrante deve falar uma parte sobre o código-fonte, referente ao que apresentou no projeto.

### \* O que é um bom layout

- Aplicativo com Personalidade da empresa, da marca, ou do profissional. É possível transmitir um pouco (ou muito) da personalidade da empresa através do layout.
- Diagramação e hierarquia. Um aplicativo que possa ser clean.
- Fotos de qualidade. A qualidade de imagem faz diferença em um layout.
- Legibilidade.
- Navegabilidade.

### Links Úteis

- Laravel [Documentação Oficial](#);
- Layout [Bootstrap](#) ou [Tail Wind CSS](#)
- Ícones Gratuitos [Flaticon](#)
- Gravação de Tela com o [ScreenRec](#) gratuita
- [Vídeo explicando a ferramenta Screen Rec](#)
- Editor de Vídeo [Schotcut Gratuito](#)
- [Vídeo explicando como usar o Schotcut](#)