

PROJETO 02 - FRONT-END COM FRAMEWORKS
PROF. GUILHERME E VICTOR

EDITAL DO ENTREGÁVEL FINAL

1) FRONT-END

- **Construa uma SPA (front-end) com Angular 19.**
 - Construa o front-end para o back-end do Projeto Integrador, que será a entrega do mensal.
 - Utilize sempre como referência as implementações contidas nos repositórios de exemplo do professor (carro-front e carro-back).
 - Utilize os enunciados das atividades semanais, bem como os vídeos e slides da Black como forma de revisão de conteúdo para realizar as implementações.
 - Os repositórios de back e de front deverão ser separados.
- **Back-end: Todo o regramento contido no Edital de entrega do Back-end se mantém.**
 - O Front deve refletir as implementações das entidades, endpoints e regras de negócio já implementadas no back-end.
 - É obrigatório o refatoramento para remover os try catch das controllers e incluir a classe GlobalExceptionHandler que está no repositório de exemplo do professor. Isso é necessário para que os erros de validations e regras de negócio sejam recebidos pelo front.
 - [Link para o repositório de Exemplo do Exception Handler](#)
- **O front deve estar 100% integrado com o back (CRUD básico + filtros personalizados).**
 - TODOS os endpoints devem ser consumidos pelo front. Ou seja, TODOS os filtros que foram criados para o back, precisam existir no front.
 - Lembre-se que as requisições devem ser definidas nos Services do front e chamadas por meio de Subscribe nos componentes com a estrutura next e error.
- **Estrutura do projeto do front:**
 - Os serviços deverão sempre ser criados em caminhos como, por exemplo: /services/entidade.
 - Os componentes deverão sempre ser criados em caminhos como, por exemplo: /components/entidade/entidade-list ou /components/entidade/entidade-form.
 - Os models deverão sempre ser criados em caminhos como, por exemplo: /models/entidade.
 - Em regra, para cada model deverá existir um componente list e outro form.
 - Deverá ter, no mínimo, a estrutura básica de layout apresentado em sala: componente de login, componente de menu superior e componente principal com o menu superior fixo.
- **Estilização:**
 - Deverá ser utilizado algum framework de CSS para auxiliar nos elementos gráficos que comporão o layout. Sugiro fortemente que utilizem o MDB.
 - Caso não utilizem o MDB, outro framework de CSS só poderá ser utilizado se compatível com a versão 19 do Angular esse instalado da forma correta por meio do npm. Na dúvida, utilizem o MDB.
 - Mesmo que seja utilizado o MDB, será avaliada também a personalização do sistema: cores, imagens, etc. Seja criativo.

- Deverá ser instalado e utilizado o SweetAlert para gerar mensagens aos usuários. Não utilizar o alert() padrão do JS.
- **Roteamento:**
 - Todas as rotas, exceto o login, deverá conter o menu superior fixado.
- **Data-binding e Janelas Modais:**
 - Os CRUDs em si podem ser acessados por rotas ou modais, à sua escolha.
 - Já os RELACIONAMENTOS, TODOS deverão ser feitos com janelas modais + Property Binding e Event Binding.
 - Somente será permitido o uso de relacionamentos N para 1 utilizando <select> para casos de ENUMS ou de, no máximo, 2 relacionamentos.
- **Tratamento de erros:**
 - Todos os erros do back deverão ser exibidos no front acessando erro.error. Não deverá ser colocado o erro genérico "Deu erro".
 - Colocando o erro.error dentro do alerta do Sweet, os erros de validation ou de regras de negócio do back serão mostrados automaticamente ao usuário.

2) REQUISITOS QUE SERÃO AVALIADOS PELO PROFESSOR:

FICHA DE AVALIAÇÃO DO PROFESSOR							GRUPO XX
PROJETO MENSAL 01 - FRONT-END - ENTREGA FINAL (entrega 11/04 - apres. 14 e 15/04)							CORREÇÕES A SEREM FEITAS PARA A ENTREGA NO PI:
CRITÉRIO	PESO	ALUNO	ALUNO	ALUNO	ALUNO	OBSERVAÇÕES	
Entrega Semanal 01	10%						
Entrega Semanal 02	10%						
Back-end com controllers refatoradas e existência do GlobalExceptionHandler?	5%						
Definição interna correta de models e services?	5%						
Integração 100% de todos os CRUDs básicos?	10%						
Integração 100% de todos os endpoints personalizados?	10%						
Estrutura externa correta (pastas) dos componentes, services e modais?	5%						
Estilização personalizada? Variou cores? Colocou imagens, etc.? MDB e SweetAlert?	5%						
Roteamento correto? Variáveis de rota no caso de forms?	5%						
Data binding: Modais + Property e Event Binding em todos os relacionamentos?	15%						
Tratamento de erros corretamente? Exibição dos erros do back nos alerts?	5%						
Validações no html equivalentes às validations do back? Inativou botão de salvar?	5%						
O sistema está 100% completo? Faltando somente autenticação e autorização?	10%						
TOTAL	100%	0	0	0	0		

* lembrando que essa nota será ponderada nos 60% da entrega final

0

3) FORMA DE ENTREGA E PRAZOS:

ENTREGA:

- **Data:** **21/09/2025.**
- **Repositório Git:** Apenas um do grupo envia a URL do repositório por meio da BlackBoard, desde que este liste os nomes de todos os membros do grupo.
- Mesmo grupo do PI.

APRESENTAÇÕES:

- **Data:** **24/09** e **25/09/2025** (a ordem será de acordo com os grupos).
- Mostrar e explicar a solução e o código de acordo com os requisitos do Item 1.

- Todos os membros devem falar/explicar algo (a avaliação é individual).
- Tempo médio de apresentação: entre 10 e 15 minutos.

4) ORIENTAÇÕES:

- **Dias de apresentações não são dias para tirar dúvidas ou corrigir bugs!** Portanto, iniciem desde já o desenvolvimento e tirem suas dúvidas até quinta-feira (18/09).
- **Não utilizem ngFor e ngIf.** Essas estruturas são das versões antigas do Angular. Utilizar sempre @for e @if para controle de fluxo.