**Lista 1 – Desenvolvimento Web**

**Regras:**

* Entrega no portal
* Individual (github link) é obrigatório a entrega no GIT

**Teóricos para fixação dos fundamentos:**

**Questão 1:**

Diferencie front end e backend. Cite duas tecnologias em cada uma delas.

**Frontend**: cuida das partes visuais e da interação entre usuários e tela. Gerencia a usabilidade, o design e a experiência do usuário. **Ex**: React, Angula.

**Backend**: cuida do funcionamento estrutural de uma página. **Ex**: Csharp, C++.

**Questão 2:**

Nos conceitos estudados em aula, falamos sobre controller e models. Responda:

1. Defina o papel de cada um e responda por que ao criarmos nosso controller fizemos uma herança de controllerbase?

**R:** Modelo é a ponte entre as camadas Visão (View) e Controle (Controller), consiste na parte lógica da aplicação, que gerencia o comportamento dos dados através de regras de negócios, lógica e funções. Criamos para reutilizar métodos e atributos que são utilizados várias vezes (evitar repetição de código e aplicar a lei do reuso).

1. Em sala, mostramos que ao abrir um navegador conseguimos chamar uma rota GET e POST não, explique por que só conseguimos executar a Rota GET no browser.

**R:** Porque o GET busca a informação e o POST atualiza.E não tem como chamar para atualizar no browser.

1. Qual o papel do Swagger, por que utilizamos ele?

**R:** Trata-se de uma aplicação open source que auxilia desenvolvedores nos processos de definir, criar, documentar e consumir APIs REST.

**Questão 3:**

Dadas ações cite o verbo http correto para utilizarmos, considere ações no AVA do uniaraxá: a) Criar a aula - **POST**

1. Remover um material - **DELETE**
2. Listar as disciplinas disponíveis para o professor no semestre - **GET**
3. Lista os alunos da disciplina - **GET**
4. Atualizar um material - **PUT**
5. Lançar uma nota - **POST**

**Questão 4:**

Em sala mostramos que ao fazer um return da action utilizamos um Status Http , por exemplo, Ok(), BadRequest(), NotFound() , entre outros. Especifique a diferença de cada um.

**R:** Ok(): sucesso da pesquisa

BadRequest(): requisição mal sucessida

NotFound(): é quando não é encontrado a pesquisa

**Questão 5:**

A seguir , temos um exemplo de endpoint de API:

<https://minhaapi.com.br/api/vendas>explique cada parte:

* https://: **protocolo**
* minhaapi.com.br: **domínio**
* /api/vendas: **endpoint**

**Questão prática:**

No seu visual studio, crie um controller chamado de AlunoController que faça o CRUD para o aluno com os dados: (Nome, Ra, Email, Cpf, Ativo)

OBS: Para persistir, faça como fizemos em aula, ou seja, utilize um arquivo json para persistirmos os dados.