**Super.br**

**Lab 01 - Backend - Criando controllers**

## Instalação do PHP

Abra o terminal e execute o comando:

|  |
| --- |
| **sudo add-apt-repository ppa:ondrej/php** |

Atualize os repositórios:

|  |
| --- |
| **sudo apt-get update** |

Rode o comando abaixo para instalar o php e os pacotes necessários:

|  |
| --- |
| **sudo apt-get install php8.2 php8.2-cli php8.2-common php8.2-xml php8.2-dev php8.2-sqlite3 php8.2-intl php8.2-mbstring** |

Verifique a versão do php e veja se a instalação ocorreu corretamente:

|  |
| --- |
| **php -v** |

## 2. Instalação do Composer

Rode o comando abaixo para instalar o composer:

|  |
| --- |
| **sudo apt-get install composer** |

3. Instalação do Insomnia e VisualCode

Dentro da pasta do lab1 execute:

|  |
| --- |
| **sudo dpkg -i Insomnia.Core-2023.1.0.deb** |

Para instalação do visualcode seguir os passos do endereço: https://code.visualstudio.com/docs/setup/linux

4. Instalação do Symfony

Rode os comando abaixo para criar uma nova aplicação symfony:

|  |
| --- |
| **composer create-project symfony/skeleton:"6.2.\*" app\_teste** |

Acesse o diretório com a aplicação criada e inicie um servidor php:

|  |
| --- |
| **php -S localhost:8000 -t public/** |

Para testar, abra o navegador e digite o endereço: [http://localhost:80](http://localhost:4200/)00

5. Criando um controller

No diretório da aplicação, acesse src/Controller crie um arquivo chamado TesteController.php

Adicione o código:

<?php

namespace App\Controller;

use Symfony\Component\HttpFoundation\Response;

use Symfony\Component\Routing\Annotation\Route;

class TesteController  
{  
 #[Route('/teste')]  
public function index(): Response  
 {  
 $n = rand();  
  
 return new Response(  
 '<html><body>Número aleatório: '.$n.'</body></html>'  
 );  
 }  
}

Abra o navegador e digite o endereço: <http://localhost:8000/teste>

Exercício 1

Avançando nos conceitos aprendidos, crie um novo controlador chamado Exercicio1Controller.php que deverá responder atráves da rota **/exercicio1**.

Esse controlador deve receber um número por parâmetro da URL, verificar se ele é impar ou par e apresentar na tela.

Exercício 2

Para o exercício 2 instale a biblioteca do twig dentro da aplicação: **composer require symfony/twig-bundle**

Crie um novo controlador chamado Exercicio2Controller.php que deverá responder atráves da rota **/exercicio2**.

Esse controlador deve receber um número por parâmetro da URL, aplicar os operadores de adição, subtração, multiplicação e divisão, e ser renderizado na tela através de um template html a ser criado dentro da pasta **templates/exercicio2/exercicio2.html.twig**