

SW-II

SISTEMAS WEB II

Prof. Anderson Vanin

Competências

1. Criar componentes de negócio distribuídos e reutilizáveis.
2. Identificar e implementar componentes de acesso.
3. Desenvolver sistemas para Internet com tecnologia de página única.

Habilidades

- 1.1 Identificar e classificar APIs.
- 1.2 Implementar APIs, utilizando os métodos HTTP.
- 2.1 Definir acesso e permissões para aplicações.
- 2.2 Definir acesso e permissões para APIs REST.
- 3.1. Codificar software em linguagem para Web.
- 3.2. Utilizar framework adequado para aplicações em página única.

Bases Tecnológicas

Serviços e microserviços

- Microsserviços x serviços;
- Conceitos de arquitetura de microserviços;
- Modelagem de APIs REST;
- Gerar documentação de APIs REST;
- Gerar mock de APIs REST.

Bases Tecnológicas

Manipulação de dados via APIs REST

- Implementação de APIs a partir de modelagem;
- Testes unitários com mocks;
- Testes integrados;
- Distribuição e publicação de APIs REST.

Bases Tecnológicas

Autenticação e autorização

- Estratégias para aplicações Web;
- Estratégias para APIs REST.

Bases Tecnológicas

Aplicações Web de página única (SPA)

- Renderização de HTML;
- Templates e estilização;
- Componentes;
- Estados;
- Roteamento e navegação;
- Eventos;
- Formulários;
- Validação.

Critérios de Avaliação

- Exercícios práticos
- Pesquisa Técnica
- Avaliação Prática
- Projeto Prático
- Assiduidade e Participação em sala de aula

Critérios de Avaliação

I : menos de 50%

R : de 50% até 70%

B : de 70% até 90%

MB : de 90% a 100%

Material das aulas

- Github:

https://github.com/ProfAndersonVanin/SW-II_2026

AULA 01 - Introdução ao Desenvolvimento Web

1. Objetivos da Aula

- Entender o conceito de **aplicações web e sistemas distribuídos**.
- Diferenciar **arquiteturas monolíticas e microsserviços**.
- Conhecer os conceitos básicos do PHP e preparar o ambiente de desenvolvimento.

2. Conceitos de Aplicações Web e Sistemas Distribuídos

O que são Aplicações Web?

Uma aplicação web é um **sistema acessado por meio de um navegador**, sem necessidade de instalação no dispositivo do usuário. Elas são hospedadas em servidores e podem ter funcionalidades dinâmicas, como login, interação com banco de dados e processamento de informações.

💡 Exemplos de Aplicações Web:

- Redes sociais (Facebook, Twitter)
- E-commerce (Amazon, Mercado Livre)
- Aplicativos bancários
- Sistemas internos de empresas (ERP, CRM)

2. Conceitos de Aplicações Web e Sistemas Distribuídos

O que são Sistemas Distribuídos?

Um sistema distribuído é um conjunto de computadores interconectados que compartilham recursos e processam dados de forma cooperativa. Esse modelo permite escalabilidade e maior disponibilidade de serviços.

❖ Exemplos de Sistemas Distribuídos:

- Netflix (servidores em várias regiões do mundo)
- Jogos online multiplayer
- Aplicações que utilizam serviços de nuvem (Google Drive, OneDrive)

3. Arquiteturas: Monolítica vs. Microsserviços

Arquitetura Monolítica

- 📌 **Definição:** O código da aplicação está todo em um único projeto. Backend, frontend e banco de dados são fortemente acoplados.
- 📌 **Vantagens:** ✓ Fácil de desenvolver no início. ✓ Simples para pequenas aplicações.
- 📌 **Desvantagens:** ✗ Dificuldade para escalar. ✗ Se uma parte falhar, toda a aplicação pode ser comprometida.
- 📌 **Exemplo de Arquitetura Monolítica:** Um e-commerce onde o sistema de login, o carrinho de compras e a emissão de notas fiscais estão no mesmo servidor.

3. Arquiteturas: Monolítica vs. Microsserviços

Arquitetura de Microsserviços

- 📌 **Definição:** A aplicação é dividida em pequenos serviços independentes que se comunicam entre si. Cada serviço pode ser desenvolvido e escalado separadamente.
- 📌 **Vantagens:** ✓ Maior escalabilidade. ✓ Manutenção facilitada.
- 📌 **Desvantagens:** ✗ Maior complexidade na comunicação entre serviços. ✗ Necessidade de maior gerenciamento.
- 📌 **Exemplo de Arquitetura de Microsserviços:** Um e-commerce onde o carrinho de compras, o catálogo de produtos e o sistema de pagamentos são **APIs independentes**.

3. Arquiteturas: Monolítica vs. Microsserviços

Comparação Resumida:

Característica	Arquitetura Monolítica	Arquitetura de Microsserviços
Escalabilidade	Baixa	Alta
Manutenção	Difícil	Mais fácil
Desempenho	Pode ser comprometido	Melhor, pois serviços podem ser separados
Complexidade	Menor	Maior, exige mais configurações

Atividade 01

Elaborar uma pesquisa sobre os seguintes assuntos:

- O que são Aplicações Web?
- O que são Sistemas Distribuídos?
- Arquitetura Monolítica;
- Arquitetura de Microsserviços.

Utilizar formatação ABNT (Capa, Sumário, Objetivos, Textual, Conclusão, Referências Bibliográficas).

Data de Entrega: Próxima aula

4. Introdução ao PHP e Revisão de Conceitos Básicos

💡 O que é PHP?

PHP (Hypertext Preprocessor) é uma linguagem de programação de código aberto usada para desenvolvimento web, especialmente para a criação de aplicações dinâmicas.

💡 **Principais Características do PHP:** Fácil de aprender. Executado no servidor antes de enviar o conteúdo ao navegador. Integração nativa com MySQL. Suporte a programação orientada a objetos.

4. Introdução ao PHP e Revisão de Conceitos Básicos

Exemplo 1: Primeiro código em PHP

Crie um arquivo chamado *index.php* e adicione o seguinte código:

```
<?php  
    echo "Olá, mundo!";  
?>
```

4. Introdução ao PHP e Revisão de Conceitos Básicos

Exemplo 2: Variáveis e Tipos de Dados

```
<?php
    $nome = "Anderson";
    $idade = 25;
    $preco = 10.50;
    $ativo = true;
    echo "Nome: " . $nome . "<br>";
    echo "Idade: " . $idade . "<br>";
    echo "Preço: R$ " . $preco . "<br>";
    echo "Status: " . ($ativo ? "Ativo" : "Inativo");
?>
```

4. Introdução ao PHP e Revisão de Conceitos Básicos

💡 Explicação:

\$nome, \$idade, \$preco, \$ativo → Declaração de variáveis em PHP.

". \$variavel . " → Concatenação de strings.

(\$ativo ? "Ativo" : "Inativo") → **Operador ternário** para exibir um status.

4. Introdução ao PHP e Revisão de Conceitos Básicos

Exemplo 3: Estruturas de Controle (if, loop)

```
<?php
    $nota = 7;
    if ($nota >= 6) {
        echo "Aprovado!";
    } else {
        echo "Reprovado!";
    }
    echo "<br>";
    for ($i = 1; $i <= 5; $i++) {
        echo "Número: " . $i . "<br>";
    }
?>
```

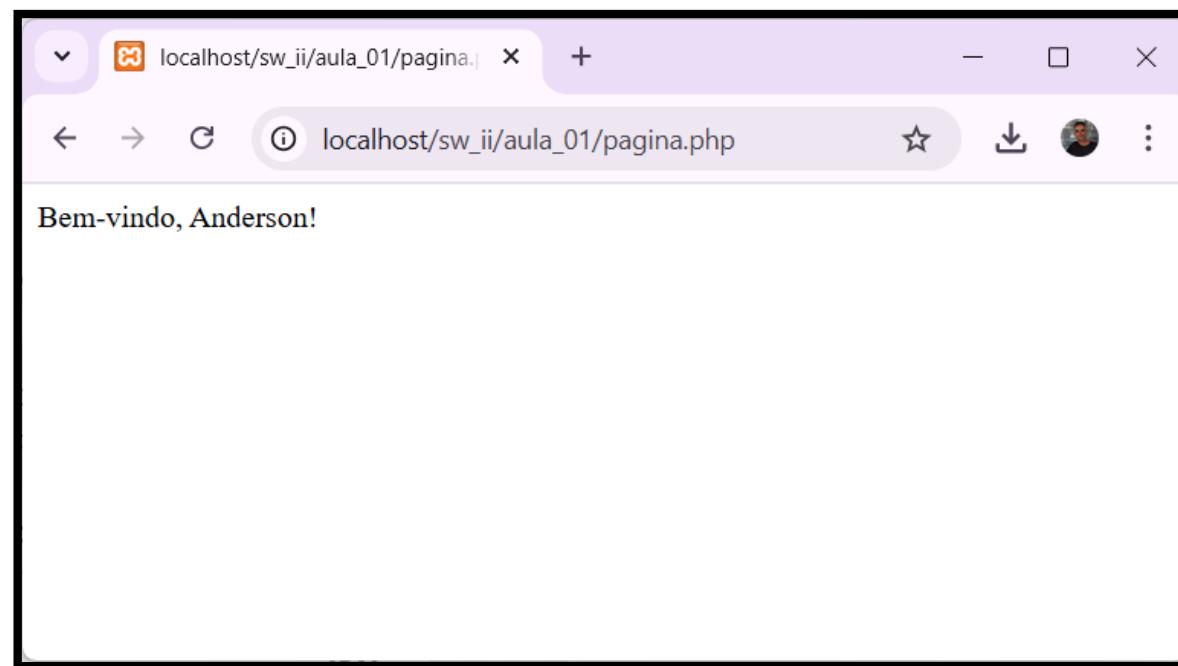
Importante!

- Crie um repositório em seu ambiente Github chamado: **SW-II_2025**
- Separe os conteúdos por aulas (pastas). Exemplo: *aula_01, aula_02, ...*
- Mantenha, sempre seu repositório atualizado!
- **Não copie as soluções de outros colegas. Tente fazer! Se não conseguir peça ajuda ao seu professor!**

5. Exercícios Práticos

Exercício 1: Criando uma Página Web Dinâmica

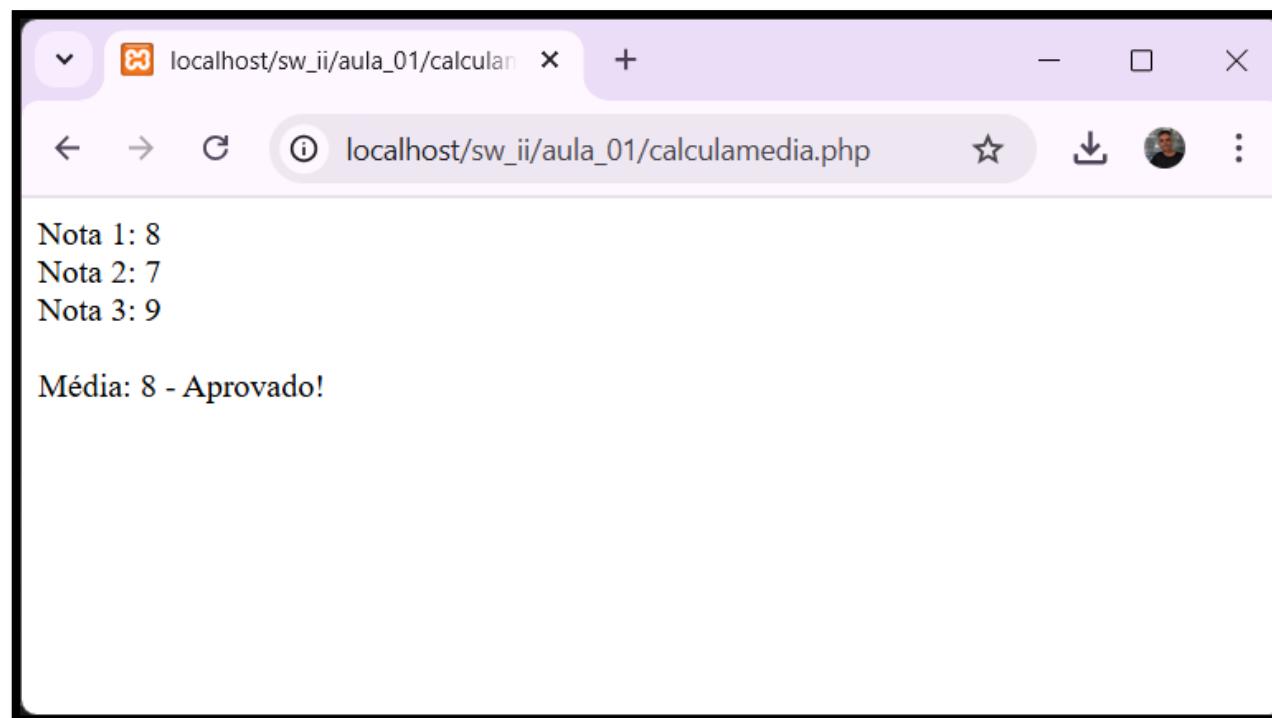
Crie um arquivo ***pagina.php*** que exiba uma mensagem personalizada para o usuário.



5. Exercícios Práticos

Exercício 2: Cálculo de Média

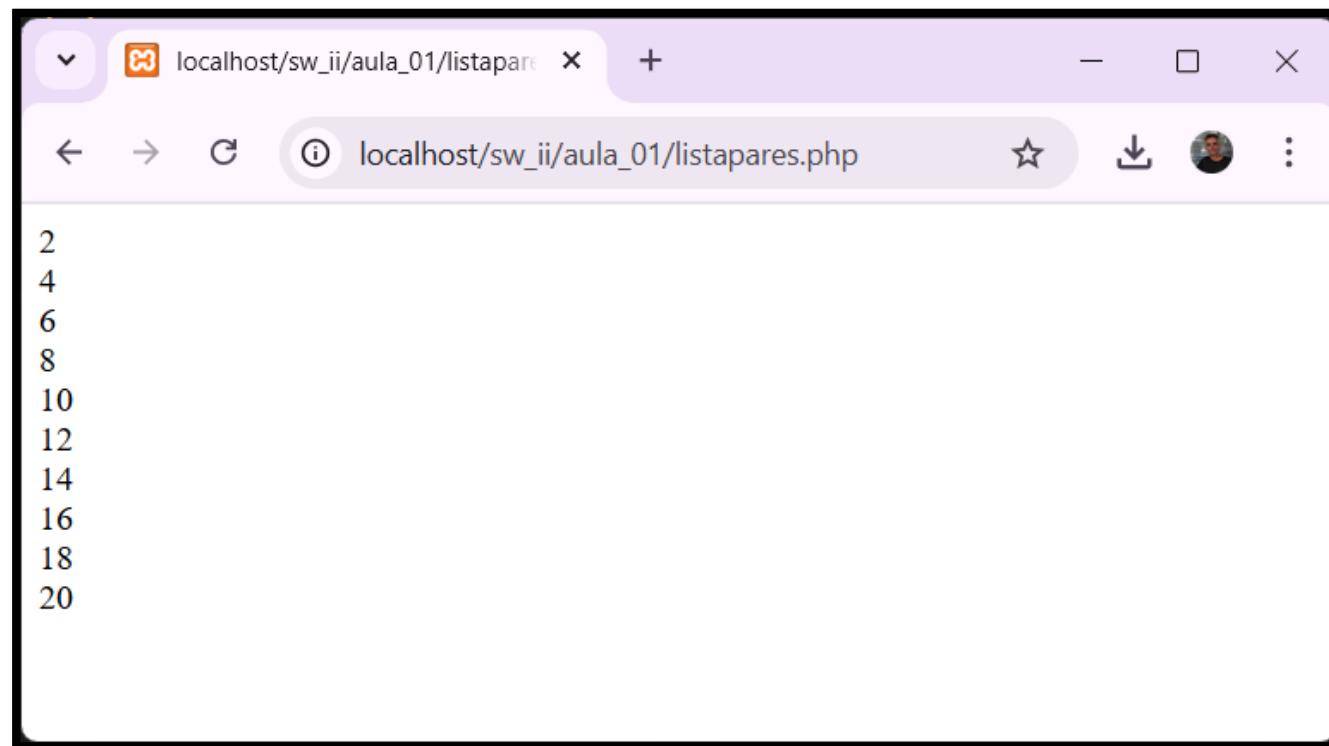
Escreva um código PHP, *calculamedia.php*, que recebe três notas e calcula a média.



5. Exercícios Práticos

Exercício 3: Listando Números Pares

Crie um script, *listapares.php*, que liste os números pares de 1 a 20.



6. Conclusão

Nesta aula, aprendemos:

- ✓ O que são aplicações web e sistemas distribuídos.
 - ✓ Diferenças entre arquitetura monolítica e microsserviços.
 - ✓ Introdução ao PHP, com exemplos práticos.
- ◊ **Próxima Aula:** Configuração do Ambiente e Primeira API REST em PHP. 