

# Desenvolvimento Web II

## Aula 01 - Introdução

Prof. Fabricio Bizotto

Instituto Federal Catarinense

*[fabricio.bizotto@ifc.edu.br](mailto:fabricio.bizotto@ifc.edu.br)*

Ciência da Computação  
19 de dezembro de 2023

## 1 Introdução a Arquitetura de Sistemas Web

- Conceitos
- Modelos Arquiteturais
  - Monolito
  - Arquitetura em N camadas (N-tier)
  - Microserviços
- Material Complementar
- Columns

## 2 Table and Figure Examples

- Table
- Figure

## 3 Mathematics

## 4 Referencing

A **arquitetura de aplicações web** descreve a estrutura interna e interações entre seus componentes. A arquitetura de uma aplicação web é composta por:

- **Componentes:** partes que compõem a aplicação web. Exemplos: cliente, servidor, banco de dados, etc.
- **Conectores:** mecanismos que permitem a comunicação entre os componentes. Exemplos: protocolos de comunicação, APIs, etc.
- **Restrições:** regras que definem como os componentes e conectores podem interagir. Exemplos: autenticação, autorização, etc.

# Principais Arquiteturas de Aplicações Web

## Monolito - Definição

Abordagem tradicional no desenvolvimento de software na qual todos os componentes de uma aplicação são combinados em uma única unidade totalmente integrada. A aplicação é implantada como uma **única base de código** que contém todas as funcionalidades.

# Principais Arquiteturas de Aplicações Web

## Monolito - Vantagens

### Vantagens

- **Simplicidade da Arquitetura:** não existem muitas camadas e componentes para gerenciar. É mais fácil para começar.
- **Tecnologias:** usar uma única linguagem de programação e tecnologias para desenvolver a aplicação pode facilitar o entendimento da equipe.
- **Fluxo de implantação:** o 'deploy' é simples de fazer e gerenciar. Não há necessidade de implantar vários componentes separadamente.

### Desvantagens

- **Acoplamento forte:** a aplicação é uma única unidade totalmente integrada, o que significa que qualquer alteração em um componente pode afetar outros componentes da aplicação.
- **Escalabilidade limitada:** a aplicação é implantada como uma única unidade, o que significa que todos os componentes da aplicação devem ser escalados juntos horizontalmente.
- **Implantação única:** a aplicação é implantada como uma única unidade, o que significa que todos os componentes da aplicação devem ser implantados juntos. Qualquer alteração, por menor que seja, requer a implantação de toda a aplicação.
- **Tecnologias limitadas:** todos os componentes da aplicação devem ser desenvolvidos usando a mesma linguagem de programação e tecnologias.

# Principais Arquiteturas de Aplicações Web

## Monolito - Representação

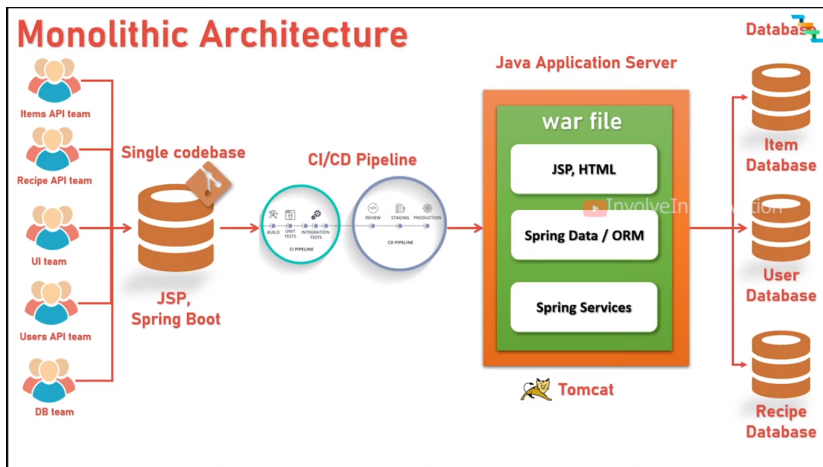


Figura: Arquitetura Monolítica.

# Principais Arquiteturas de Aplicações Web

## Monolito - Escalando Horizontalmente

- **Escalabilidade horizontal:** adicionar mais instâncias de um componente.
- **Escalabilidade vertical:** adicionar mais recursos (CPU, memória, etc) a um componente.

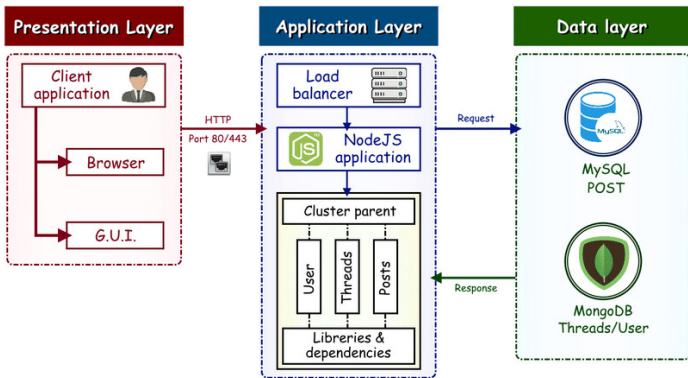


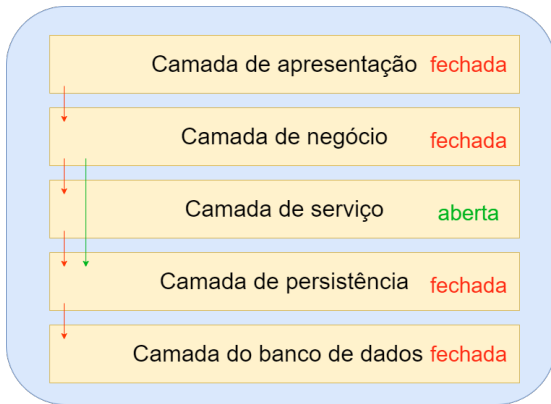
Figura: Arquitetura Monolítica com Load Balancer.



# Principais Arquiteturas de Aplicações Web

## Arquitetura em N camadas (N-tier) - Definição

A arquitetura em N camadas é um padrão de arquitetura de software no qual a aplicação é dividida em camadas lógicas ou físicas.



**Figura:** Arquitetura em camadas - Fluxo.

# Principais Arquiteturas de Aplicações Web

## Arquitetura em N camadas (N-tier) - Vantagens

### Vantagens

- **Separação de Responsabilidades:** A separação clara das responsabilidades em diferentes camadas (como apresentação, lógica de negócios e acesso a dados) facilita a manutenção e a evolução do sistema.
- **Escalabilidade:** A escalabilidade é facilitada, pois cada camada pode ser dimensionada independentemente das outras, permitindo a otimização de recursos.
- **Facilidade de Testes:** Cada camada pode ser testada separadamente, o que simplifica os testes unitários e facilita a identificação e correção de falhas.
- **Manutenção:** Alterações em uma camada específica não devem afetar as outras, tornando a manutenção mais simples e menos propensa a efeitos colaterais indesejados.

# Principais Arquiteturas de Aplicações Web

## Arquitetura em N camadas (N-tier) - Desvantagens

### Desvantagens

- **Complexidade Inicial:** A implementação de uma arquitetura em camadas pode ser mais complexa inicialmente, especialmente para projetos pequenos ou simples.
- **Comunicação entre camadas:** A comunicação entre camadas pode resultar em algum overhead, especialmente em sistemas distribuídos, o que pode impactar o desempenho.
- **Duplicação de Lógica:** Pode ocorrer uma duplicação de lógica entre as camadas, o que pode levar a inconsistências se não for gerenciado adequadamente.
- **Aumento da Latência:** O acesso a dados através de várias camadas pode aumentar a latência.

# Principais Arquiteturas de Aplicações Web

## Arquitetura em N camadas (N-tier) - Representação

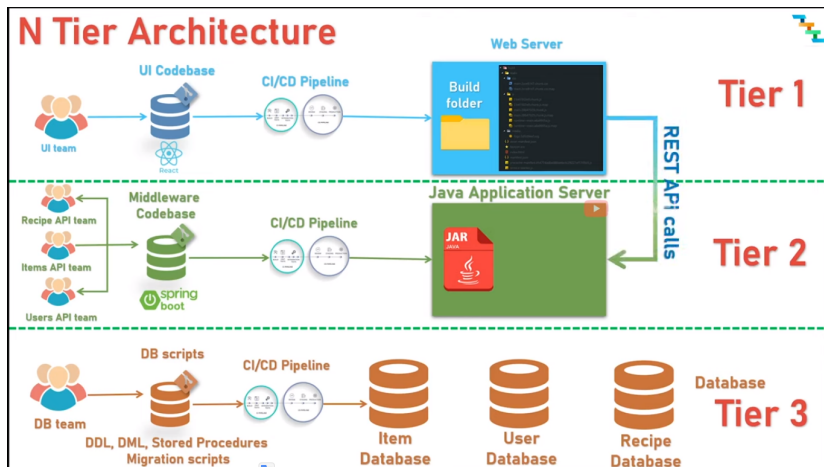


Figura: Arquitetura em N camadas (N-tier).

# Principais Arquiteturas de Aplicações Web

## Microserviços - Definição

Microserviços são uma abordagem arquitetônica e organizacional do desenvolvimento de software na qual o software consiste em pequenos serviços independentes que se comunicam usando APIs bem definidas. Esses serviços pertencem a pequenas equipes autossuficientes. É similar à **arquitetura orientada a serviços (SOA)**, mas com algumas diferenças importantes, tais como, por exemplo, o tamanho dos serviços e a forma como eles se comunicam.

### Vantagens

- **Separação de Responsabilidades:** Tudo é desenvolvido através de pequenas unidades de código e publicado em processos de deploy automatizados e independentes.
- **Tecnologias:** Essa abordagem permite que cada serviço seja desenvolvido usando a *stack* mais adequadas para o problema que está sendo resolvido.
- **Aplicabilidade:** em aplicações de grande porte e complexas que precisam ser escaladas rapidamente ou em organizações com equipes distribuídas.

### Desvantagens

- **Complexidade:** A complexidade de uma arquitetura de microserviços é maior do que a de uma arquitetura monolítica, pois existem mais componentes para gerenciar.
- **Comunicação:** A comunicação entre os serviços pode resultar em algum overhead, especialmente em sistemas distribuídos, o que pode impactar o desempenho.
- **Testes:** Os testes de integração são mais complexos, pois envolvem a comunicação entre os serviços.
- **Gerenciamento:** O gerenciamento de uma arquitetura de microserviços (governança) é mais complexo, pois existem mais componentes para gerenciar.

# Principais Arquiteturas de Aplicações Web

## Microserviços - Representação

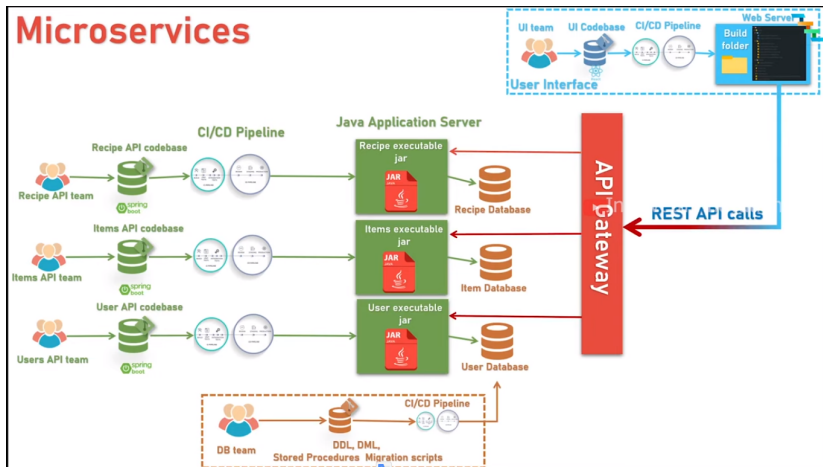


Figura: Arquitetura de Microserviços.



# Principais Arquiteturas de Aplicações Web

## Monolito vs. Microsserviços

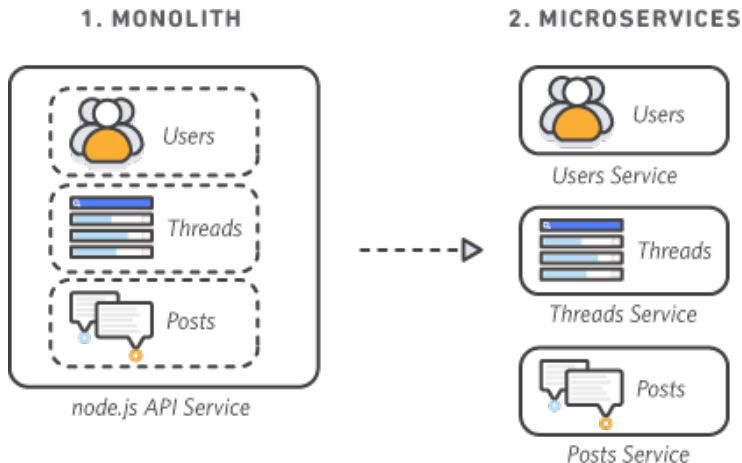


Figura: Monolito vs. Microsserviços.

- **Aplicação Monolítica // Dicionário do Programador.** Canal **Código Fonte TV**. Disponível em: <https://youtu.be/CsrHHHPHKwE>.
- **Arquitetura de Software: Monolítica x SOA x Microserviços.** Canal **Marcos Dósea**. Disponível em: <https://youtu.be/suZfVAk7hco>.
- **Microservices // Dicionário do Programador.** Canal **Código Fonte TV**. Disponível em:  
[https://www.youtube.com/watch?v=\\_2bDOCTnbKc](https://www.youtube.com/watch?v=_2bDOCTnbKc).
- **Microservices na prática.** Canal **Full Cycle**. Disponível em:  
[https://youtu.be/gtv9szE\\_P1U](https://youtu.be/gtv9szE_P1U).

# Multiple Columns

Subtitle

## Heading

- 1 Statement
- 2 Explanation
- 3 Example

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer lectus nisl, ultricies in feugiat rutrum, porttitor sit amet augue. Aliquam ut tortor mauris. Sed volutpat ante purus, quis accumsan dolor.

# Table

Subtitle

Treatments	Response 1	Response 2
Treatment 1	0.0003262	0.562
Treatment 2	0.0015681	0.910
Treatment 3	0.0009271	0.296

**Tabela:** Table caption



**INSTITUTO FEDERAL**  
Catarinense  
Campus Videira

Figura: IFC Videira.

# Definitions & Examples

## Definition

A **prime number** is a number that has exactly two divisors.

## Example

- 2 is prime (two divisors: 1 and 2).
- 3 is prime (two divisors: 1 and 3).
- 4 is not prime (**three** divisors: 1, 2, and 4).

You can also use the `theorem`, `lemma`, `proof` and `corollary` environments.

# Theorem, Corollary & Proof

## Theorem (Mass–energy equivalence)

$$E = mc^2$$

## Corollary

$$x + y = y + x$$

## Demonstração.

$$\omega + \phi = \epsilon$$



# Equation

$$\cos^3 \theta = \frac{1}{4} \cos \theta + \frac{3}{4} \cos 3\theta \quad (1)$$



## Example (Theorem Slide Code)

```
\begin{frame}  
\frametitle{Theorem}  
\begin{theorem}[Mass--energy equivalence]  
$E = mc^2$  
\end{theorem}  
\end{frame}
```

Slide without title.

# Citing References

An example of the `\cite` command to cite within the presentation:

This statement requires citation.

# References

# Acknowledgements

## Smith Lab

- Alice Smith
- Devon Brown

## Cook Lab

- Margaret
- Jennifer
- Yuan

## Funding

- British Royal Navy
- Norwegian Government

# The End

Questions? Comments?