



# Aula 01

Prof: Henrique Augusto Maltauro

# Codificar Back-end de Aplicações Web

# Arquitetura de Aplicações Web

# Arquitetura de Aplicações Web

- Front-End

Podemos definir o **front-end** como a parte visível e interativa de uma página web.

Envolve toda a questão gráfica construída com **HTML**, **CSS** e **JavaScript**.

# Arquitetura de Aplicações Web

- Back-End

Podemos definir o **back-end** como a parte mais robusta e complexa da lógica de programação de um site.

Podemos dizer que são os bastidores de uma aplicação web, é a infraestrutura por trás do que está visível para os usuários.

# Arquitetura de Aplicações Web

Existem diversas arquiteturas diferentes para trabalharmos com aplicações web, mas vamos nesse começo darmos atenção as duas mais básicas e que são de extrema importância a todo desenvolvedor web conhecer e diferenciar:

- Monolito
- Micro Serviço

# Arquitetura de Aplicações Web

- Monolito

Esta é a arquitetura mais comum e mais utilizada para desenvolvimento de aplicações web devido a sua simplicidade e também por ser a mais antiga utilizada.

A ideia aqui é basicamente agrupar todo o código e todas as funcionalidades em uma única aplicação.

# Arquitetura de Aplicações Web

- Monolito (Vantagens)
- Maior facilidade de desenvolvimento, principalmente pela uniformidade de tecnologias entre todas as camadas da aplicação.
- Mais simples para testar já que toda a aplicação é entregue em um único pacote, o que facilita todo o processo de testes.

# Arquitetura de Aplicações Web

- Monolito (Vantagens)
- Mais simples e rápido para efetuar a entrega, já que temos um único pacote para se preocupar.
- Mais simples de se escalabilizar a aplicação.
- Necessita de um time de desenvolvimento menor para manter a aplicação.



# Arquitetura de Aplicações Web

- Monolito (Vantagens)
- Os desenvolvedores compartilham de um conhecimento equalizado, já que trata-se de um único código base.
- É mais simples e fácil de aprender.
- O desenvolvimento inicial é mais rápido.
- Precisa de uma infraestrutura simples.

# Arquitetura de Aplicações Web

- Monolito (Desvantagens)
  - Os componentes têm alto acoplamento, o que resulta em efeitos colaterais como regressão de problemas devido a atualização.
  - Tente a se tornar muito complexo e gigante depois de certo tempo, aumentando o tempo de desenvolvimento.

# Arquitetura de Aplicações Web

- Monolito (Desvantagens)
- Novas funcionalidades vão demorar mais para serem desenvolvidas e a refatoração será cada vez mais complexa devido ao alto acoplamento.
- A aplicação toda precisa de um processo de entrega para a adição de qualquer funcionalidade nova.

# Arquitetura de Aplicações Web

- Monolito (Desvantagens)
  - É menos confiável devido ao acoplamento, um pequeno problema em algum processo não muito importante pode quebrar a aplicação toda.
  - A utilização de novas tecnologias é difícil e geralmente toda a aplicação precisa ser migrada, o que geralmente resulta em um código defasado a longo prazo.

# Arquitetura de Aplicações Web

- Monolito (Desvantagens)
  - Serviços críticos não podem ser escalados individualmente, o que aumenta o uso de recursos.
  - Tempo longo de inicialização e grande utilização de CPU e memória.
  - As equipes de desenvolvimento são muito dependentes e é desafiador escalar as equipes.

# Arquitetura de Aplicações Web

- Monolito (Quando usar?)
  - Quando o escopo do aplicativo é pequeno e bem definido, e se tem certeza de que o aplicativo não crescerá tremendamente em termos de recursos.
  - Quando o tamanho da equipe é pequeno, por exemplo, menos de oito pessoas.

# Arquitetura de Aplicações Web

- Monolito (Quando usar?)
  - Quando o conjunto de habilidades médio da equipe é de desenvolvedores Júnior ou Pleno.
  - Quando o tempo de entrega ao mercado é crítico.

# Arquitetura de Aplicações Web

- Monolito (Quando usar?)
  - Quando não se quer gastar muito em infra-estrutura, monitoramento e assim por diante.
  - Quando a base de usuários é bastante pequena e não se espera que eles cresçam.



# Arquitetura de Aplicações Web

- Micro Serviço

Este tipo de arquitetura pode oferecer quase todas as funcionalidades de uma arquitetura monolítica e adicionalmente, oferece muitas outras funcionalidades e maior flexibilidade e consequentemente é geralmente considerada uma escolha superior para aplicações complexas.

# Arquitetura de Aplicações Web

- Micro Serviço

Diferentemente da arquitetura monolítica é difícil generalizar esta opção pois ela pode variar muito conforme o caso de uso e aplicações.

A ideia aqui é basicamente agrupar as funcionalidades em grupos, e assim, em diversas aplicações diferentes.

# Arquitetura de Aplicações Web

- Micro Serviço (Vantagens)
  - Baixo acoplamento e melhor isolamento, é mais rápido para inicializar a aplicação.
  - Novos recursos podem ser desenvolvidos mais rapidamente e os existentes facilmente refatorados.

# Arquitetura de Aplicações Web

- Micro Serviço (Vantagens)
- Problemas nos processos estão isolados e não vão derrubar a aplicação toda.
- A utilização de novas tecnologias é mais simples, os componentes podem ser atualizados de forma independente e de forma incremental, possibilitando um código com diferentes tecnologias e linguagens.

# Arquitetura de Aplicações Web

- Micro Serviço (Vantagens)
- Os serviços serão inicializados de forma mais rápida e possivelmente de forma paralela.
- As equipes de desenvolvimento são independentes, o que se encaixa perfeitamente nas metodologias de desenvolvimento ágil.

# Arquitetura de Aplicações Web

- Micro Serviço (Desvantagens)
  - Em termos gerais a aplicação é mais complexa.
  - Os processos de testes e integração são complexos e custosos, pois existem mais partes “móveis” dentro da aplicação.
  - A aplicação toda é mais complexa de efetuar a entrega.

# Arquitetura de Aplicações Web

- Micro Serviço (Desvantagens)
  - Escalar é mais fácil, porém definir as regras de escalamento são mais complexas e necessitam de funcionalidades avançadas.
  - Geralmente precisa de um time de desenvolvimento maior para manter o software.

# Arquitetura de Aplicações Web

- Micro Serviço (Desvantagens)
  - As habilidades da equipe se tornam mais variadas, dificultando o compartilhamento de conhecimento e deixando a reposição de desenvolvedores mais complicada.
  - O agrupamento de tecnologias é mais complexo e é mais difícil de aprender.



# Arquitetura de Aplicações Web

- Micro Serviço (Desvantagens)
  - O desenvolvimento inicial é mais lento.
  - Precisa de uma infraestrutura complexa.

# Arquitetura de Aplicações Web

- Micro Serviço (Quando usar?)
- Quando o escopo do aplicativo é grande e bem definido e se tem certeza que o aplicativo crescerá tremendamente em termos de recursos.
- Quando o tamanho da equipe é grande, possuindo membros suficientes para desenvolver os componentes individuais de maneira independente.

# Arquitetura de Aplicações Web

- Micro Serviço (Quando usar?)
- Quando o conjunto médio de habilidades da equipe é bom e os membros da equipe estão confiantes sobre padrões avançados de micro serviço.
- Quando o tempo de comercialização não é crítico. A arquitetura de micro serviço levará mais tempo para lançar a primeira versão.

# Arquitetura de Aplicações Web

- Micro Serviço (Quando usar?)
- Quando se estiver de acordo com gastar mais em infraestrutura, monitoramento e assim por diante, a fim de melhorar a qualidade do produto.
- Quando a base de usuários é enorme e se espera que ela cresça.

# Arquitetura de Aplicações Web

- API

Uma das maneiras mais comuns de se construir o **back-end** de aplicações é através de **APIs**.

A **API**, do inglês **Application Programming Interface** (**Interface de Programação de Aplicação**) tem o objetivo de permitir que dois softwares, totalmente distintos, possam se comunicar através de um conjunto de definições e protocolos.

# Arquitetura de Aplicações Web

- API

Nesse contexto, nós conseguimos fazer uma relação do objetivo com a sigla API.

Onde, a aplicação vai ser qualquer software com funcionalidades distintas, e a interface vai ser imaginada como um contrato entre duas aplicações.

# Arquitetura de Aplicações Web

- API (Interação front-end e back-end)

Sendo uma API também uma aplicação back-end, podemos facilmente acessar os processos da API através de chamadas executadas no front-end.