

ARQUITETURA DE SOFTWARE

João Choma Neto

joao.choma@unicesumar.edu.br

<https://github.com/JoaoChoma/arquitetura-software>

Unicesumar – Maringá



2 BIMESTRE

- Arquitetura monolítica
- Arquitetura em camadas
- Arquitetura cliente-servidor
- Arquitetura baseada em serviços
- Arquitetura de microserviços

ARQUITETURA BASEADA EM SERVIÇOS

REST (Representational State Transfer)

- REST não é um protocolo, mas sim um conjunto de princípios arquiteturais. Utiliza os métodos HTTP padrões (GET, POST, PUT, DELETE, etc.) de maneira mais direta e eficiente.
- REST é um estilo arquitetural sem estado, significando que cada pedido HTTP deve conter todas as informações necessárias para compreendê-lo, sem exigir que o servidor memorize qualquer estado de sessão.

REST (Representational State Transfer)

- Embora o JSON seja o formato de dados mais popular usado em APIs REST devido à sua facilidade de uso com JavaScript e sua leveza em comparação ao XML, REST permite o uso de diferentes formatos como XML, HTML, YAML, etc.

FIREBASE

- Firebase, uma plataforma de desenvolvimento de aplicativos da Google, também oferece várias APIs REST que permitem aos desenvolvedores interagir com seus serviços de backend

GOV BR

CATALOGO DE APIs DO GOV BR

- <https://www.gov.br/conecta/catalogo/>

CATALOGO DE APIs DO IBGE

- <https://servicodados.ibge.gov.br/api/docs/>

ARQUITETURA DE MICROSERVIÇOS

MICROSERVIÇOS

- Microserviços são uma abordagem arquitetural no desenvolvimento de software onde uma aplicação é estruturada como um conjunto de pequenos serviços
- Cada serviço executa uma única função

MICROSERVIÇOS

- Cada serviços comunica-se com outros serviços através de interfaces bem definidas, geralmente APIs.
- Cada microserviço é desenvolvido, implantado e escalado independentemente, permitindo uma maior flexibilidade e modularidade na gestão do sistema.

Características dos Microserviços

- **Descentralização:**
 - Cada microserviço é responsável por uma parte específica da funcionalidade da aplicação, e pode ser desenvolvido e mantido por diferentes equipes.

Características dos Microserviços

- **Independência de Implantação:**
 - Os microserviços podem ser implantados, atualizados, escalados e reiniciados de forma independente, sem afetar os demais serviços.

Características dos Microserviços

- **Escalabilidade:**
 - Permite escalar apenas os serviços que necessitam de mais recursos, ao invés de escalar toda a aplicação.

Características dos Microserviços

- **Tolerância a Falhas:**
 - A falha em um microserviço não necessariamente compromete toda a aplicação, pois os outros serviços continuam funcionando.

Características dos Microserviços

- **Tecnologia Heterogênea:**
 - Cada microserviço pode ser desenvolvido usando diferentes tecnologias e linguagens de programação, desde que respeitem as interfaces de comunicação.

Benefícios dos microserviços

- Manutenção Facilitada: Código mais simples e específico facilita a correção de bugs e a implementação de novas funcionalidades.
- Implantação Contínua: Permite implantar novas versões de serviços individualmente, facilitando a integração contínua e entrega contínua (CI/CD).

Cuidados com a utilização

- Gerenciamento de Dados: Cada serviço pode ter seu próprio banco de dados, necessitando de uma abordagem consistente para transações de dados distribuídas.
- Segurança: Aumenta a superfície de ataque, requerendo uma atenção especial à segurança de cada serviço e suas comunicações.

e-COMMERCE

EM MICROSERVIÇOS

SERVIÇO DE AUTENTICAÇÃO

- Responsável por gerenciar o login, logout e a criação de usuários.
- Verifica as credenciais e emite tokens de autenticação.

SERVIÇO DE CATÁLOGO DE PRODUTOS

- Gerencia os dados dos produtos disponíveis para venda.
- Permite adicionar, atualizar, remover e listar produtos.

SERVIÇO DE CARRINHO DE COMPRAS

- Gerencia o carrinho de compras do usuário.
- Adiciona, remove e lista itens no carrinho.

SERVIÇO DE PEDIDO

- Processa pedidos de compra.
- Gerencia o status do pedido e histórico de compras.

SERVIÇO DE PAGAMENTO

- Processa pagamentos e transações financeiras.
- Integra com gateways de pagamento.

SERVIÇO DE NOTIFICAÇÃO

- Envia notificações por email ou SMS sobre o status do pedido, promoções, etc.

PRODUTOS

Firebase

- **Firebase Authentication:** Serviço de autenticação para gerenciar usuários.
- **Firestore:** Banco de dados NoSQL para armazenamento de dados.
- **Firebase Cloud Messaging (FCM):** Serviço de envio de notificações push.

AWS (Amazon Web Services)

- **AWS Lambda:** Serviço de computação sem servidor para executar código em resposta a eventos.
- **Amazon S3:** Armazenamento de objetos para arquivos e dados.
- **Amazon RDS:** Banco de dados relacional gerenciado.
- **Amazon SNS:** Serviço de notificações simples para envio de mensagens.

Azure (Microsoft)

- **Azure Functions:** Serviço de computação sem servidor para executar código sob demanda.
- **Azure Cosmos DB:** Banco de dados NoSQL globalmente distribuído.
- **Azure Logic Apps:** Serviço de automação e orquestração de fluxos de trabalho.

Google Cloud Platform (GCP)

- **Google Cloud Functions:** Serviço de computação sem servidor.
- **Google Cloud Pub/Sub:** Serviço de mensageria para troca de mensagens assíncronas.
- **Google Cloud Firestore:** Banco de dados NoSQL em tempo real.

Heroku

Heroku Postgres: Banco de dados PostgreSQL gerenciado.

Heroku Redis: Armazenamento em cache e banco de dados de chave-valor.

Heroku Connect: Sincronização de dados entre Salesforce e Heroku Postgres.

Modelagem de uma arquitetura

- **Autenticação e Autorização:** Firebase Authentication, Auth0
- **Armazenamento de Dados:** Firestore, Amazon RDS, Azure Cosmos DB
- **Computação Sem Servidor:** AWS Lambda, Azure Functions, Google Cloud Functions
- **Mensageria e Notificações:** Firebase Cloud Messaging, Amazon SNS

Ideia

1. **Autenticação:** Usar Firebase Authentication para gerenciar os logins e a criação de contas.
2. **Banco de Dados:** Usar Firestore para armazenar dados de produtos, usuários, e pedidos.
3. **Funções de Backend:** Usar AWS Lambda para processar eventos como a criação de pedidos e a verificação de pagamentos.
4. **Mensageria:** Usar Google Cloud Pub/Sub para enviar mensagens entre serviços, como notificações de novos pedidos.
5. **Notificações:** Usar Firebase Cloud Messaging para enviar notificações push aos usuários sobre o status do pedido.

ARQUITETURA DE SOFTWARE

João Choma Neto

joao.choma@unicesumar.edu.br

<https://github.com/JoaoChoma/arquitetura-software>

Unicesumar – Maringá

