



Sistemas Distribuídos
Dr. Joseffe Barroso de Oliveira

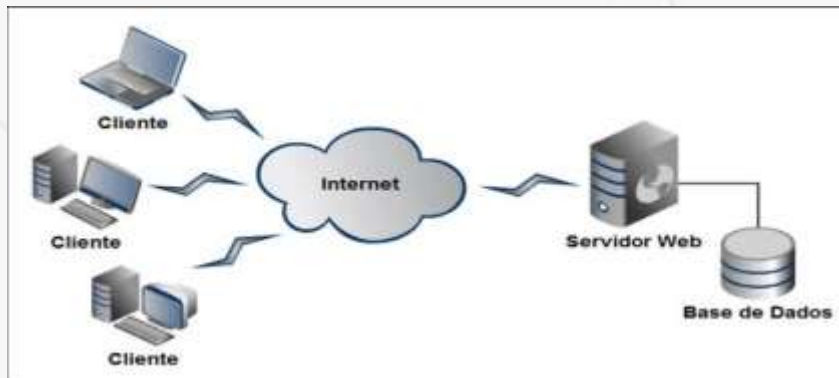


Introdução

A arquitetura de um sistema é **sua estrutura em termos de componentes especificados separadamente e suas inter-relações**. O **objetivo global é garantir que a estrutura atenda às demandas atuais e, provavelmente, às futuras demandas** impostas sobre ela. As maiores preocupações são tornar o sistema confiável, gerenciável, adaptável e rentável. O projeto arquitetônico de um prédio tem aspectos similares – ele determina não apenas sua aparência, mas também sua estrutura geral e seu estilo arquitetônico, fornecendo um padrão de referência coerente para seu projeto.

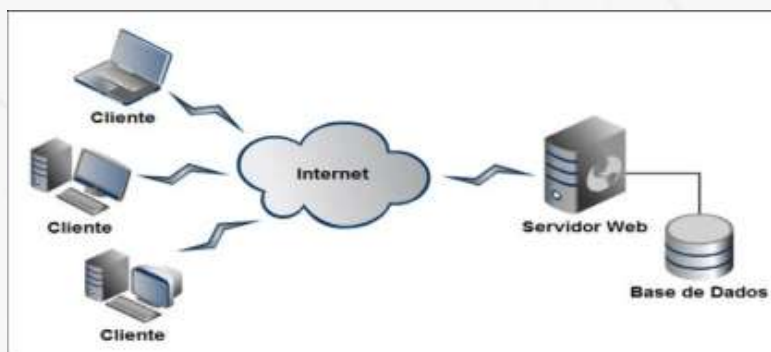
Arquitetura Cliente-Servidor

Uma arquitetura cliente-servidor é dividida em **duas responsabilidades principais**. O **cliente** é responsável pela apresentação da interface do usuário, que então se conecta pela rede ao servidor. O **servidor** é responsável por lidar com a lógica empresarial e o gerenciamento do estado.



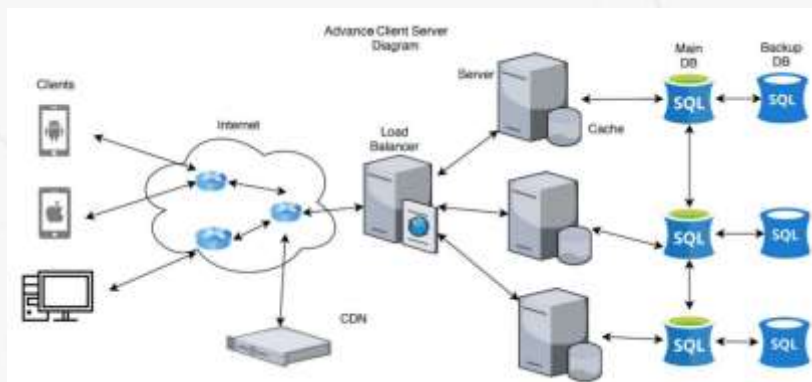
Arquitetura Cliente-Servidor

Uma arquitetura cliente-servidor pode ser **degradada com facilidade para uma arquitetura centralizada se o servidor não for redundante**. Uma configuração cliente-servidor de fato distribuída vai ter vários pontos centrais de servidor para distribuir as conexões do cliente. **A maioria das arquiteturas cliente-servidor modernas são clientes que se conectam a um sistema distribuído encapsulado no servidor.**



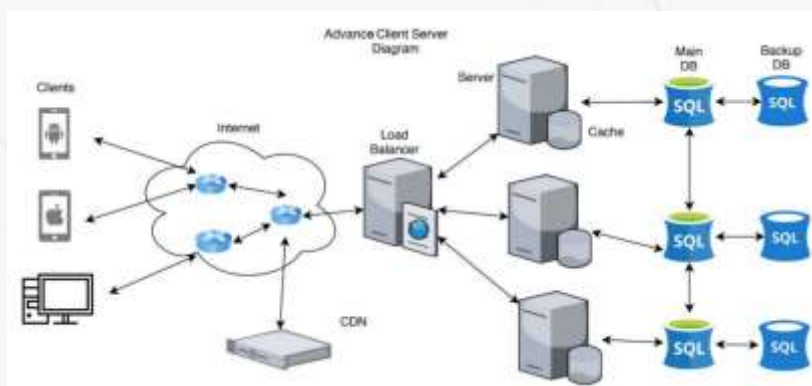
Arquitetura Vários Níveis

Uma arquitetura de vários níveis **expande a arquitetura cliente-servidor**. O servidor em uma arquitetura de vários níveis é decomposto em pontos centrais granulares adicionais, que dissociam responsabilidades adicionais do servidor de back-end, como processamento e gerenciamento de dados.



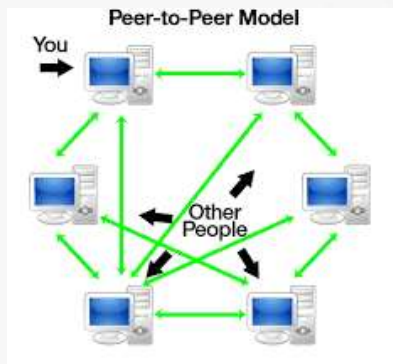
Arquitetura Vários Níveis

Esses pontos centrais adicionais são usados para **processar trabalhos de longa duração sem sincronia** e liberar os pontos centrais de back-end restantes para se concentrarem na resposta às solicitações do cliente e na interface com o armazenamento de dados.



Arquitetura Peer-to-Peer (Par-a-Par)

Em um sistema distribuído par-a-par, **cada ponto central contém a instância completa de um aplicativo**. Não há separação de pontos centrais da apresentação e do processamento de dados. Um ponto central contém a camada de apresentação e as camadas de manipulação de dados. Os pontos centrais de mesmo nível podem conter todos os dados de estado de todo o sistema.



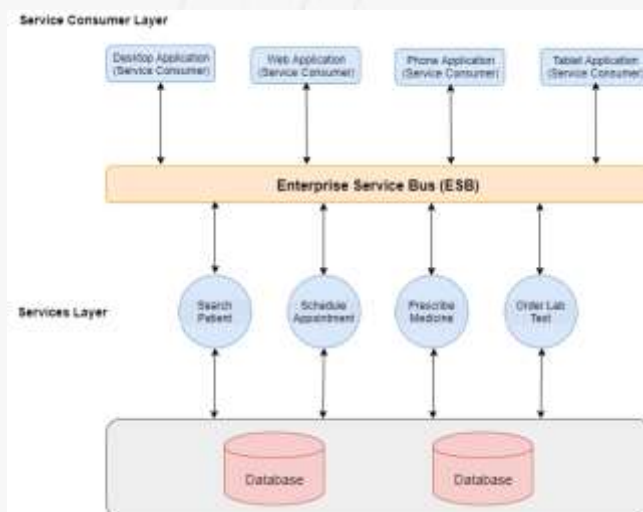
Arquitetura Peer-to-Peer (Par-a-Par)

Os sistemas par-a-par têm o benefício de extrema redundância. Quando um ponto central par-a-par é inicializado e colocado on-line, ele descobre e se conecta a outros pares e sincroniza seu estado local com o estado do sistema maior. Com esse recurso, a falha de um ponto central em um sistema par-a-par não vai interromper nenhum dos outros pontos centrais, e o sistema par-a-par vai persistir.



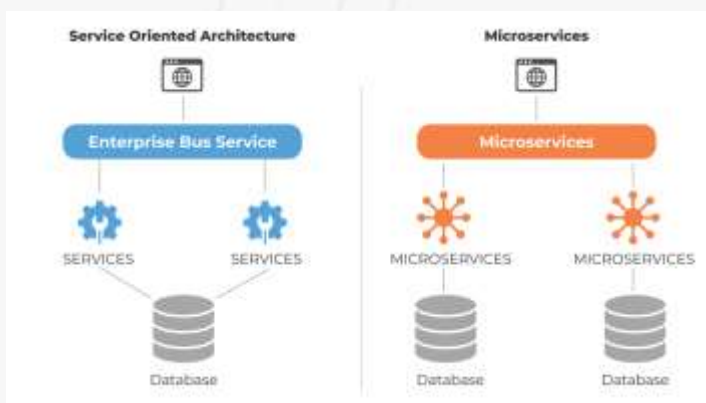
Arquitetura Orientada a Serviço (SOA)

A arquitetura orientada a Serviço (SOA) é uma antecessora dos microsserviços. A principal diferença entre a SOA e os microsserviços é o escopo do ponto central — o escopo dos pontos centrais de microsserviço existe no nível do recurso.



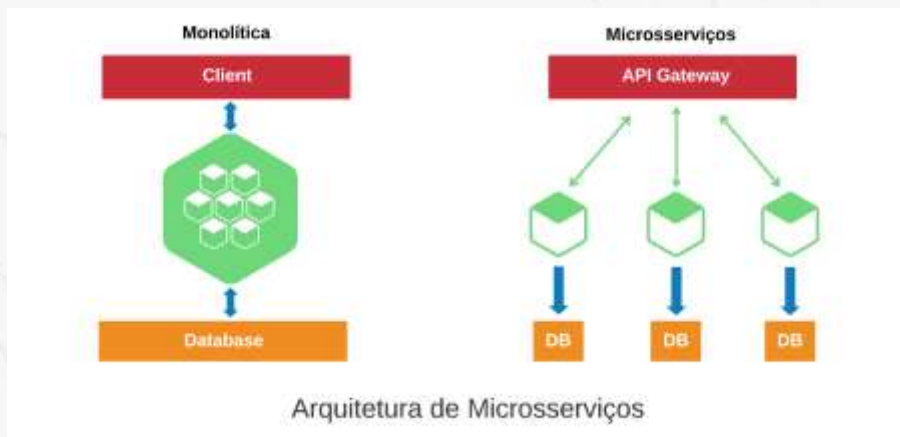
Arquitetura Orientada a Serviço (SOA)

Os microsserviços contêm **vários pontos centrais** de lógica empresarial diferentes que interagem com pontos centrais de banco de dados independentes. Em comparação, os pontos centrais da SOA **encapsulam um aplicativo inteiro ou uma divisão corporativa**. O limite de serviço para pontos centrais da SOA, em geral, inclui um sistema de banco de dados inteiro dentro do ponto central.



Arquitetura de Microserviços

Os microserviços surgiram como uma alternativa mais popular a SOA devido aos seus benefícios. Os microserviços são mais robustos e **permitem um escalonamento vertical e horizontal mais dinâmico**.



Arquitetura de Microserviços

Além disso, a arquitetura de micro serviços é utilizada para desenvolver uma aplicação como um **conjunto de pequenos serviços**, que funcionam com seu próprio processo. Cada serviço é desenvolvido em torno de um conjunto de **regras de negócio específicas**, e é **implementado de forma independente**.

