



Anhanguera

# COMPUTAÇÃO EM NUVEM

Modelos de arquitetura em nuvem

Prof. Me Fernando S. Claro  
fernando.claro@anhanguera.com

## ÍNDICE

**01.** Introdução

**02.** Arquitetura centralizada

**03.** Arquitetura descentralizada

**04.** Arquitetura híbrida

## ÍNDICE

- 05. Aplicações multicamadas
- 06. Modelo de microserviços
- 07. Momento da prática
- 08. Exercícios

- As aplicações em nuvem são sistemas de software complexos, na medida em que implementam diversas funcionalidades e fazem uso de variados serviços que são acessados por meio da Internet.
- Para lidar com essa complexidade, as aplicações podem ser divididas em módulos (ou componentes) funcionais.

- Por exemplo, uma aplicação web de comércio eletrônico pode incluir vários módulos, como um responsável pela autenticação de usuários ou outro responsável pela geração de relatórios de vendas.
- Cada um deles pode ser um software independente, capaz de ser executado, por exemplo, em uma máquina virtual ou em um contêiner de um provedor de nuvem pública.

- A definição da arquitetura de uma aplicação consiste em definir quais seriam os seus módulos funcionais e como eles devem interagir entre si (COULOURIS *et al.*, 2013).
- O projeto da arquitetura de aplicações em nuvem é um grande desafio e as decisões nele envolvidas podem influenciar vários aspectos da aplicação, como desempenho, escalabilidade e segurança.

- A fase de projeto é importante para tentar antecipar a identificação de possíveis problemas como: gargalos de desempenho, ameaças de segurança e impacto da falha de componentes específicos na disponibilidade da aplicação.
- Componentes críticos podem ser replicados para viabilizar mecanismos de tolerância a falhas e balanceamento de carga.

- Existem três modelos básicos para a arquitetura de aplicações distribuídas: arquitetura centralizada, descentralizada e híbrida (TANENBAUM; STEEN, 2008).
- **A arquitetura centralizada** é o modelo tradicional cliente-servidor. Quando um componente requisita um serviço de outro, o que faz a requisição é o cliente e o que responde é o servidor. As aplicações web representam um exemplo típico do modelo de arquitetura centralizada: o servidor web responde as requisições enviadas por navegadores web.

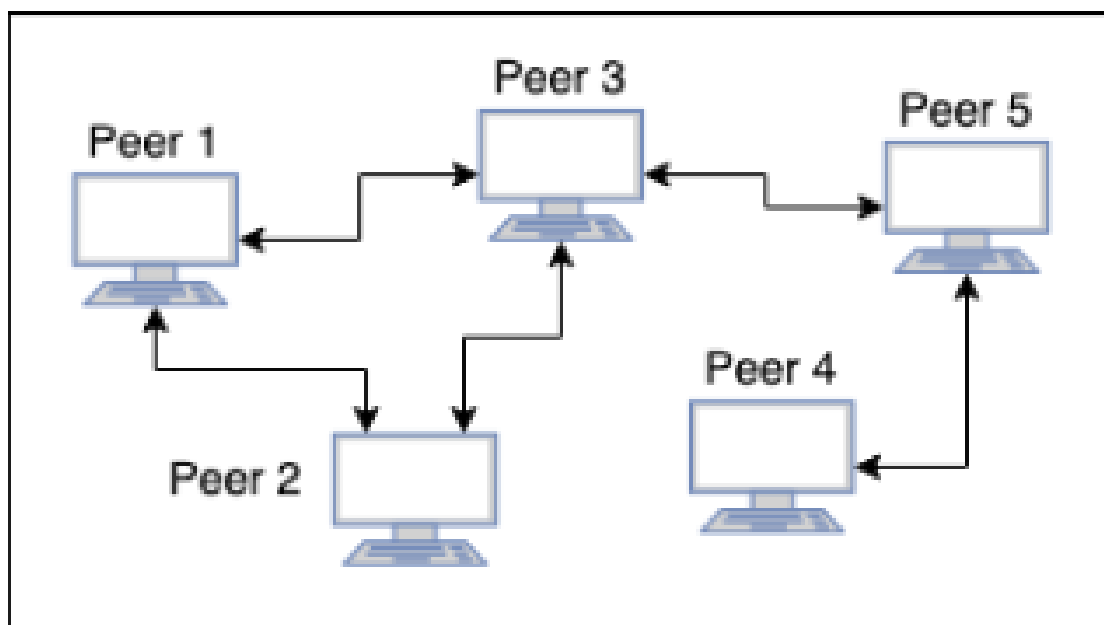




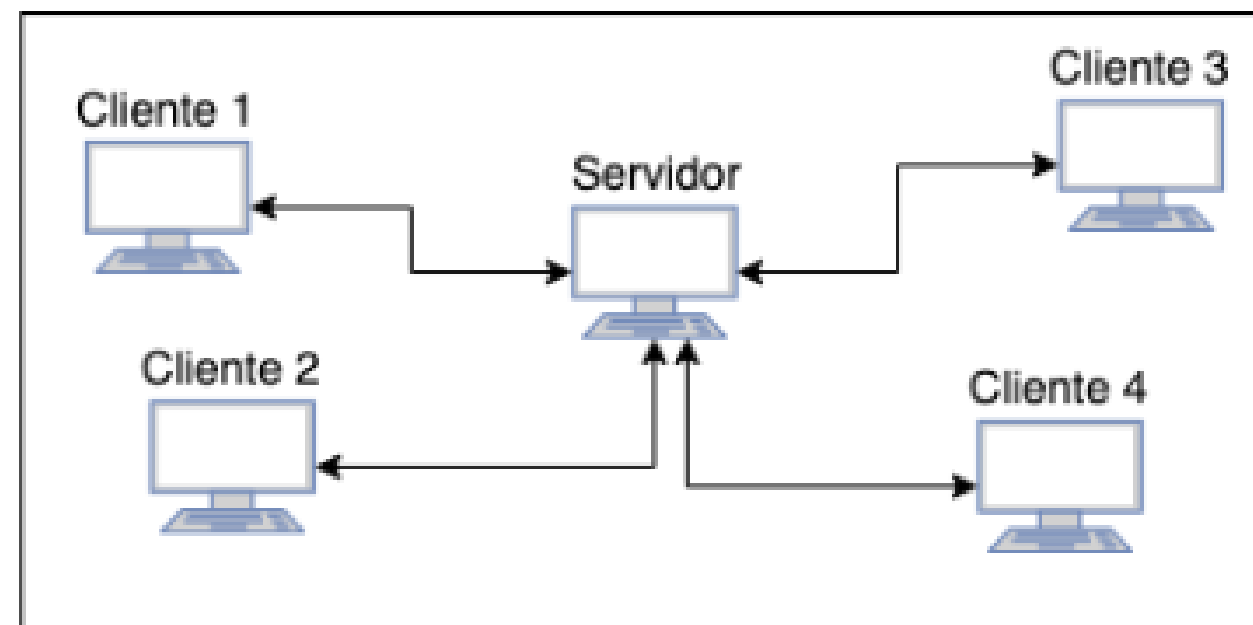
- **A arquitetura descentralizada** é caracterizada pelo modelo Peer-to-Peer (P2P). Nesse caso, não há distinção entre clientes e servidores, e os componentes de software, por sua vez, podem fazer requisições entre si de forma arbitrária.

A Figura a seguir ilustra a lógica dos modelos cliente-servidor e P2P.

# Arquitetura descentralizada



P2P



Cliente-Servidor

- Uma alternativa interessante a escolher um modelo centralizado ou descentralizado é a **arquitetura híbrida**, na qual uma mesma aplicação utiliza os dois modelos, ou seja, algumas funcionalidades da aplicação são implementadas na forma cliente-servidor e outras são implementadas na forma P2P.

Assim, pode-se aproveitar o melhor de cada arquitetura.

- **Exemplificando**

- Você conhece aplicações com arquitetura híbrida? Esse é o caso de algumas aplicações de compartilhamento de arquivos, como aquelas baseadas no **protocolo BitTorrent**.

Nessas aplicações, a transferência dos arquivos é implementada conforme o modelo P2P, mas a busca pelos arquivos é no modelo cliente-servidor.

- Outro exemplo são aplicações de comunicação por voz, como o Skype.

Nesse caso, a verificação dos contatos on-line é implementada na forma cliente-servidor, mas as chamadas podem ser P2P.



- A complexidade existente no modelo P2P, fez com que surgissem outros modelos como o cliente-servidor com uma **arquitetura em múltiplas camadas**.
- Nesse modelo de arquitetura, a aplicação é dividida em várias camadas, sendo cada uma delas responsável por um conjunto específico de funcionalidades.



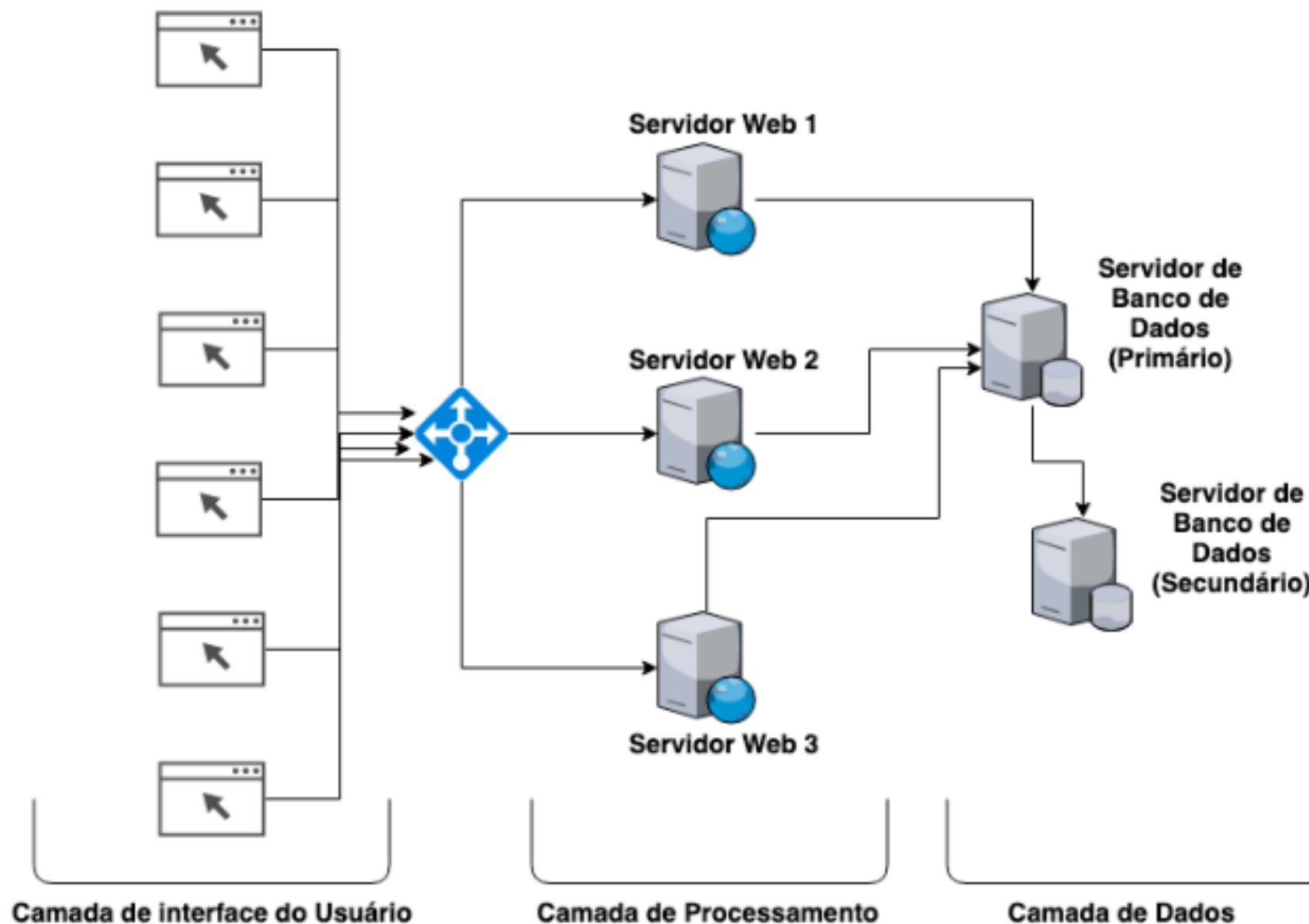
- Os componentes de uma camada podem interagir com os componentes das camadas vizinhas, além de serem executados em servidores diferentes para melhorar o desempenho.
- Nesse caso, temos uma separação física entre as camadas.



- Em geral, as aplicações multicamadas são divididas em três:
1. Camada de interface do usuário (controle da interação com o usuário).
  2. Camada de processamento (implementação da lógica das principais funcionalidades da aplicação).
  3. Camada de dados (armazenamento persistente de dados).



# Arquitetura em múltiplas camadas

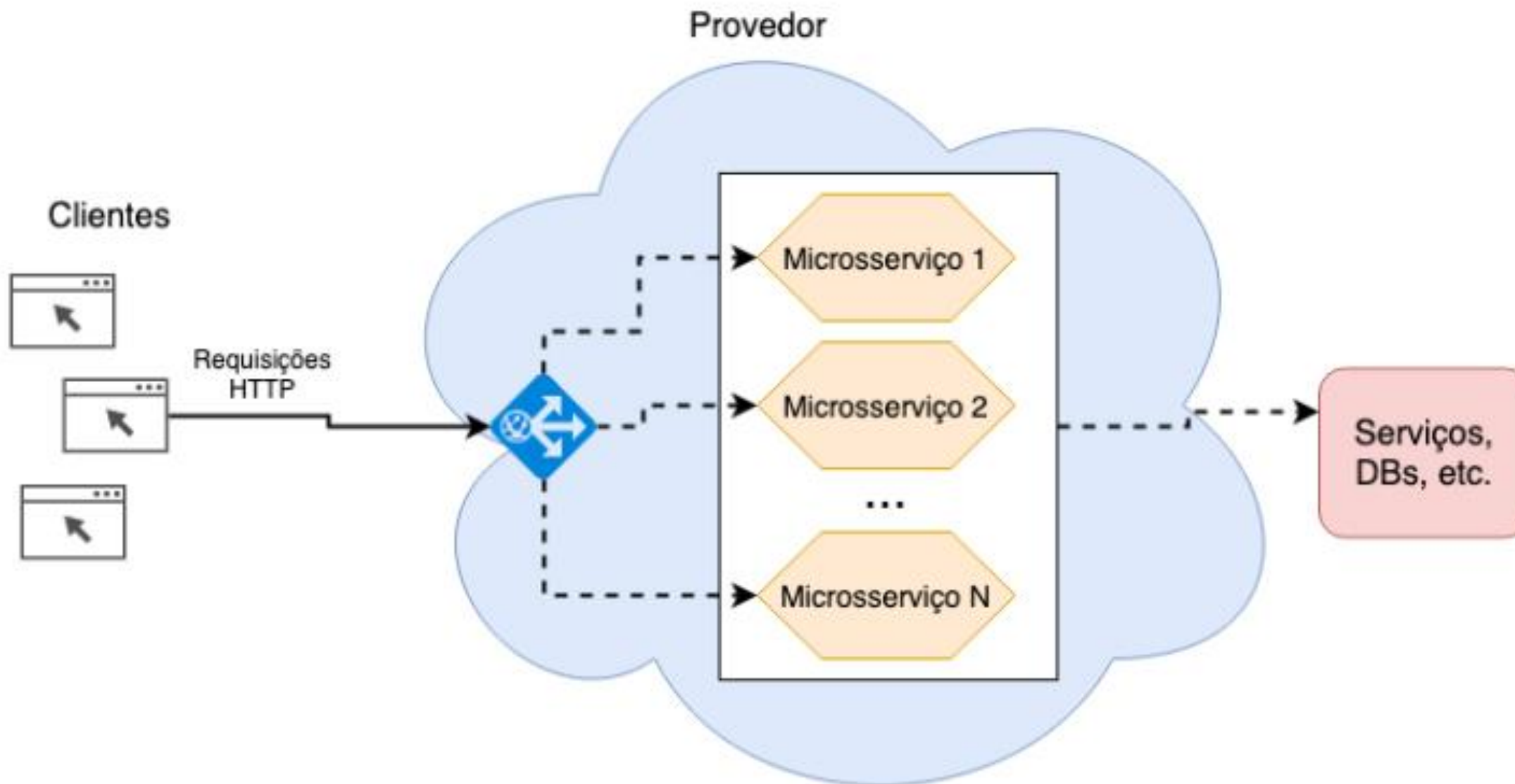




- No **modelo de microsserviços**, cada um dos módulos funcionais de uma aplicação monolítica vira um serviço menor e especializado, denominado microsserviço.
- Cada microsserviço é um componente de software independente.
- Assim, ele pode ser executado como um serviço web em um contêiner ou máquina virtual.

- A figura a seguir exemplifica a arquitetura de microsserviços.
- Nesse caso, uma aplicação é modularizada em N microsserviços independentes, que podem se comunicar com serviços remotos, como outra aplicação, com sistemas legados ou com um banco de dados (DB-*Database*).

# Modelo de microsserviços





- A figura também ilustra o papel de um ***API Gateway***, que é um componente responsável por redirecionar as requisições dos clientes para o microsserviço apropriado de acordo com a funcionalidade requisitada.
- A arquitetura de microsserviços é adequada para aplicações complexas, por exemplo, uma aplicação para consolidação de transações financeiras com cartão de crédito, pois envolve várias etapas e entidades, como lojas, bancos, adquirentes, etc.

- Nesse caso, a aplicação pode ser modularizada em vários microsserviços específicos que lidam com cada parte do problema.
- Duas características principais definem um microsserviço: o **alto grau de coesão** das suas funcionalidades e **sua capacidade de responder requisições**.



- A coesão significa que um microsserviço implementa um conjunto de funcionalidades relacionadas, que dependem umas das outras.
- A capacidade de responder requisições implica que os microsserviços, de fato, se comportam como um servidor, ou seja, provêm algum serviço.
- Eles podem responder requisições de clientes ou requisições de outros microsserviços.

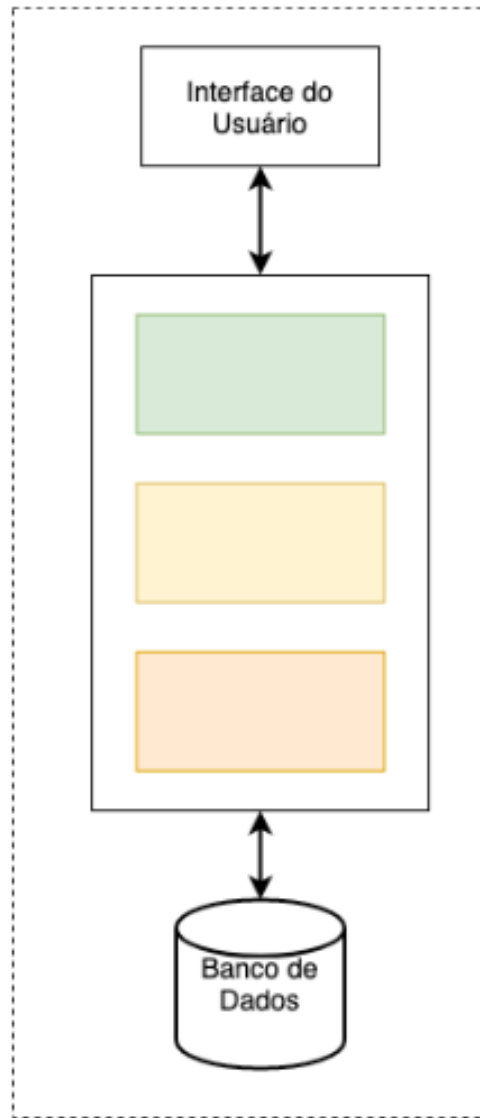
- A arquitetura de microsserviços representa aplicações distribuídas que são formadas pela composição de microsserviços independentes.
- Esse conceito é ilustrado na figura a seguir.
- Nesse caso, temos um serviço monolítico formado por três módulos funcionais.



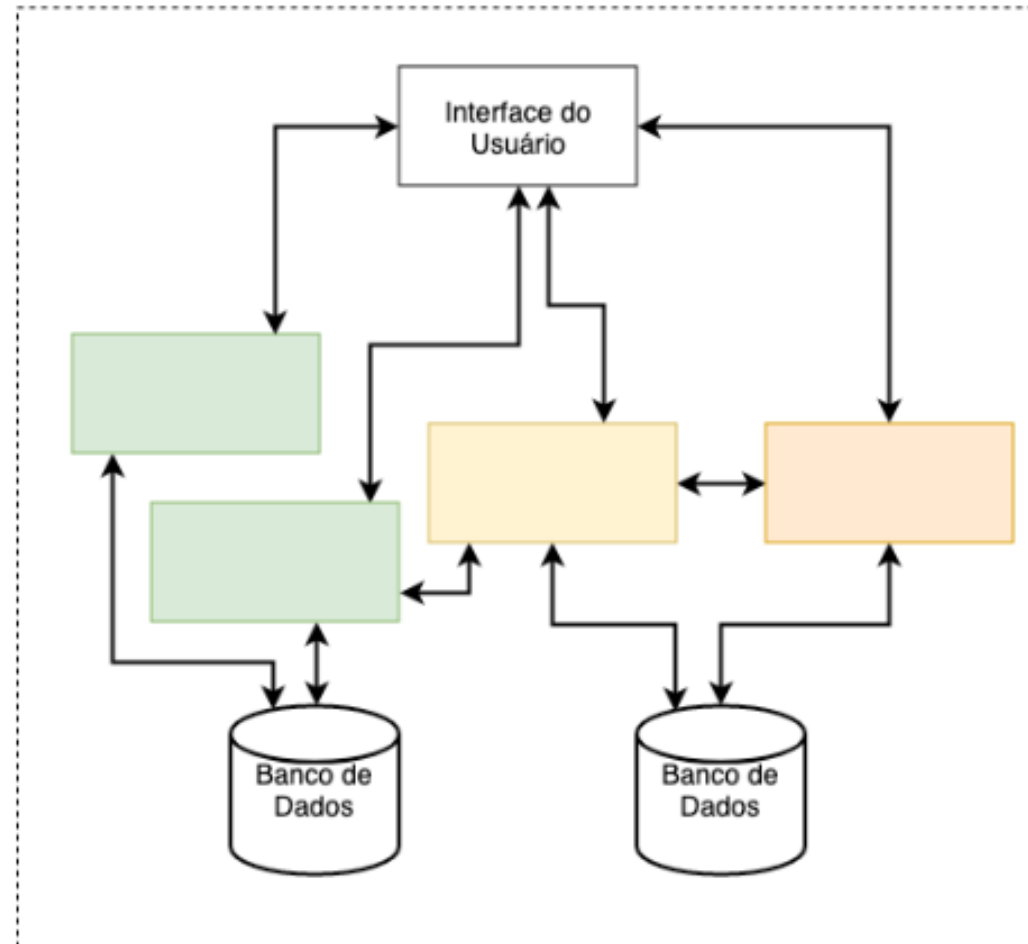


- Em uma arquitetura de microsserviços, cada um desses módulos poderia ser implementado como um (micro) serviço independente.
- Como mostra a figura, apenas alguns dos microsserviços podem ser replicados e cada um deles pode ter seu próprio banco de dados.

# Modelo de microsserviços



Monolítico



Microsserviços



- Em uma arquitetura de microsserviços, cada um desses módulos poderia ser implementado como um (micro) serviço independente.
- Como mostra a figura, apenas alguns dos microsserviços podem ser replicados e cada um deles pode ter seu próprio banco de dados.

- **Software de Controle para Veículos autônomos**
- Você é analista de TI em uma empresa do setor automotivo que decidiu iniciar a fabricação de veículos autônomos, que não precisam de motoristas, pois eles possuem um sistema de controle sofisticado capaz de conduzir o veículo com segurança.

- Seu papel é liderar a equipe que vai implementar o software de controle para condução automática dos veículos.
- Você precisa, inicialmente, escolher um modelo de arquitetura para a solução desses veículos.

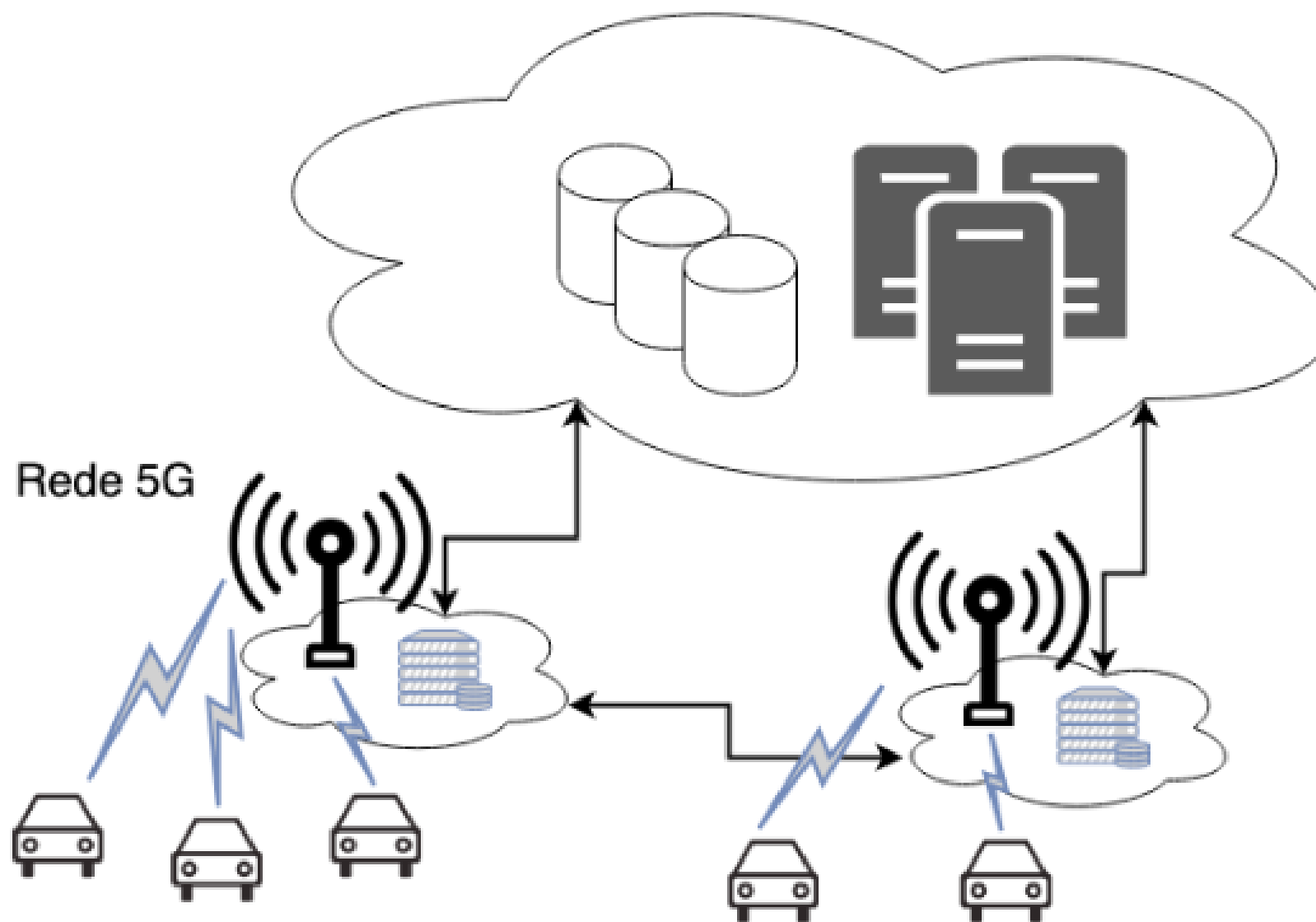
- **Resolução da situação-problema**
- Existem muitos aplicativos de navegação para veículos baseados em soluções em nuvem.
- No entanto, um software de controle para condução de veículo precisa tomar decisões em tempo real.

- Além do uso de serviços de inteligência artificial, esse tipo de aplicação requer baixa latência de comunicação e baixo tempo de resposta no processamento de dados.
- Se os veículos tivessem que se comunicar com servidores na nuvem, esses requisitos poderiam não ser atendidos.
- O modelo mais adequado nesse caso seria uma abordagem de **Edge Computing**, como ilustrado na figura a seguir.



- Os carros poderiam se comunicar com altas taxas de transmissão por meio de uma rede sem fio 5G e aproveitar a capacidade de processamento e armazenamento de dados das estações de transmissão de dados para executar funcionalidades em tempo real.
- Mesmo assim, serviços em nuvem poderiam ser utilizados para agregar informações, armazenar dados históricos para análise de estatísticas e para cálculos de rotas longas que exigem dados do trânsito em várias regiões.





1. O modelo básico de arquitetura centralizada (cliente-servidor) evoluiu para um modelo com múltiplas camadas.

Nesse caso, o sistema é dividido em camadas que têm responsabilidades específicas. Tradicionalmente, as aplicações Web são divididas em três camadas: camada de interface do usuário, camada de processamento e camada de dados.

Considerando a arquitetura multicamadas, é correto o que se afirma em:

- a. Na camada de dados, ficam os componentes responsáveis pela entrada de dados da aplicação.
- b. A lógica da aplicação deve ser implementada na camada de interface de usuários.
- c. Um componente da camada interface do usuário não deveria interagir diretamente com um componente da camada de dados.
- d. A separação das camadas é somente lógica, pois todos os componentes de uma aplicação precisam ser executados em um mesmo servidor.
- e. Por questões de consistência dos dados, não pode haver replicação de componentes da camada de processamento.

2. A arquitetura de microsserviços apresenta uma série de vantagens sobre aplicações monolíticas. Nesse contexto, avalie as afirmativas a seguir:
  - I. Caso seja necessário, microsserviços que compõem uma mesma aplicação podem ser implementados em linguagens diferentes em função de requisitos específicos.
  - II. Como o nível de modularização é maior nas aplicações baseadas em microsserviços, essa arquitetura permite agilizar as atualizações necessárias.

III. Como os microsserviços são altamente coesos, testes de integração são desnecessários.

Considerando as afirmativas apresentadas, é correto o que se afirma em:

- a. I, apenas.
- b. II, apenas.
- c. III, apenas.
- d. I e II, apenas.
- e. II e III, apenas.

3. As arquiteturas mais modernas para aplicações em nuvem são representadas pelos modelos FaaS (Serverless) e Edge Computing. Sobre esses modelos, avalie as afirmativas a seguir:

- I. Uma desvantagem do modelo FaaS é a possibilidade de maior latência (maior tempo de resposta), pois os recursos precisam ainda ser alocados quando a execução de uma função é requisitada.



- II. O modelo *Edge Computing* seria apropriado para uma aplicação de mineração de dados aplicada à Inteligência de Negócios.
- III. Em uma aplicação implementada com FaaS, uma função pode efetuar requisições para um banco de dados ou outra aplicação, mas não para outra função na mesma aplicação.

Considerando as afirmativas apresentadas, é correto o que se afirma em:

- a. I, apenas.
- b. II, apenas.
- c. III, apenas.
- d. I e II, apenas.
- e. II e III, apenas.