

ALUNO: João Gabriel Santos Andrade Almeida

A arquitetura de microsserviços surgiu como um novo modelo no desenvolvimento de software, ganhando espaço nas empresas que criam aplicativos. Essa forma de projetar sistemas propõe uma maneira bem diferente da tradicional, a chamada arquitetura monolítica, trazendo consigo novas possibilidades e também desafios para as empresas.

O modelo de microsserviços se baseia em construir um aplicativo como um conjunto de pequenos serviços que funcionam sozinhos. Cada um desses serviços roda de forma independente e se comunica com os outros usando métodos simples, como as APIs HTTP. Eles são organizados de acordo com as necessidades específicas de cada área da empresa e podem ser colocados em funcionamento de forma separada, usando sistemas de implantação totalmente automáticos.

Um ponto importante dos microsserviços é que eles não precisam de um controle central forte. Cada serviço pode ser feito com diferentes linguagens de programação e usar tecnologias de armazenamento de dados diferentes, dando uma liberdade nunca vista antes na escolha das ferramentas.

A arquitetura monolítica tradicional organiza os aplicativos em três partes principais: a interface que o usuário vê (HTML/JavaScript), um banco de dados centralizado e o aplicativo que roda no servidor. Esse aplicativo no servidor é como uma peça única que processa os pedidos HTTP, faz o que precisa ser feito, cuida dos dados e envia as respostas.

Esse modelo é simples no começo: toda a lógica fica em um único lugar, o que facilita o desenvolvimento, os testes e a implantação. Os desenvolvedores conseguem trabalhar e testar tudo no próprio computador, usando os métodos de implantação que já conhecem. E é possível aumentar a capacidade do sistema colocando várias cópias dele atrás de balanceadores de carga.

Mas os aplicativos monolíticos começam a dar dor de cabeça, principalmente quando estão na nuvem. As mudanças ficam ligadas umas às outras: qualquer pequena alteração exige refazer e colocar todo o aplicativo no ar de novo. Com o tempo, fica difícil separar as partes, o que complica a tarefa de isolar as mudanças. E para aumentar a capacidade, é preciso aumentar tudo, mesmo que só algumas partes precisem de mais recursos.

Os microsserviços usam componentes independentes, os serviços, em vez de bibliotecas. Por serem processos separados que se comunicam por meio de chamadas remotas (HTTP/RPC), eles oferecem interfaces mais claras e menos dependência entre as partes. Cada serviço se torna um componente que pode ser trocado e atualizado de forma independente.

Ao contrário da separação por áreas técnicas (UI, banco de dados, backend), os microsserviços são organizados de acordo com as funções da empresa. Isso faz com que as equipes sejam responsáveis por serviços completos, desde a criação

até o funcionamento. Cada equipe domina todos os aspectos técnicos necessários para a sua área.

Os microsserviços adotam a ideia de "produto, não projeto". Em vez de entregar um software "pronto" para equipes de manutenção, as equipes continuam responsáveis por todo o ciclo de vida do serviço. O lema "você constrói, você opera" da Amazon mostra bem essa forma de pensar, onde os desenvolvedores são responsáveis pelo funcionamento contínuo do que criam.

Ao contrário das arquiteturas SOA que se apoiam no ESB (Barramento de Serviço Empresarial), os microsserviços priorizam instâncias com lógica autônoma e uma comunicação direta. Eles usam protocolos mais simples, como REST via HTTP, ou sistemas de mensagens básicos, evitando a complexidade de um middleware robusto.

Cada serviço tem a liberdade de escolher tecnologias diferentes, incluindo linguagens de programação e sistemas de bancos de dados variados (Persistência Poliglota). As equipes criam ferramentas que podem ser reaproveitadas e compartilham práticas usando modelos semelhantes aos de código aberto interno, incentivando a inovação de forma descentralizada.

Cada microsserviço administra seu próprio banco de dados, diminuindo expressivamente a dependência entre os serviços. Essa tática troca a consistência imediata por uma consistência que se concretiza com o tempo, usando mecanismos de compensação para lidar com imprevistos distribuídos.

Os microsserviços se tornam praticáveis por meio de uma automação ampla, que abrange Integração Contínua, Entrega Contínua, provisionamento automático e monitoramento de ponta. Empresas como a Netflix mostram como ferramentas de código aberto podem dar suporte a ecossistemas complexos de microsserviços.

Os microsserviços devem ser capazes de resistir a falhas de outros serviços por meio de padrões como Circuit Breaker e Bulkhead. O monitoramento em tempo real se torna crucial para identificar comportamentos emergentes problemáticos em sistemas distribuídos intrincados.

Os microsserviços agilizam modificações rápidas, pequenas e bem supervisionadas. Os serviços podem ser completamente substituídos em vez de evoluírem indefinidamente, permitindo uma renovação tecnológica contínua sem afetar todo o sistema.

Os microsserviços proporcionam uma escalabilidade precisa, permitindo aumentar a capacidade apenas de componentes específicos quando necessário. A implementação independente acelera os ciclos de desenvolvimento e minimiza os riscos de implantação. A organização clara simplifica a compreensão e a manutenção. A maior sintonia com as capacidades de negócio otimiza a eficiência organizacional.

Entretanto, os microsserviços adicionam uma complexidade considerável na comunicação entre os serviços, exigindo um gerenciamento sofisticado de rede, latência e falhas parciais. O monitoramento se torna exponencialmente mais complexo, exigindo rastreamento distribuído e observabilidade avançada. A coordenação entre vários serviços para funcionalidades transversais apresenta desafios arquiteturais importantes.

A refatoração das fronteiras entre os serviços pode ser mais difícil do que a refatoração interna de monólitos. Os microsserviços exigem uma maior experiência da equipe em áreas como sistemas distribuídos, DevOps e monitoramento operacional.

A transição para microsserviços não deve ser feita de forma precoce. A recomendação comum sugere começar com um monólito modular bem estruturado, migrando para microsserviços apenas quando problemas de escala, evolução ou organização realmente surgirem.

Essa abordagem gradual permite que as empresas desenvolvam as competências necessárias aos poucos, evitando a complexidade desnecessária nas fases iniciais do projeto. Os microsserviços são uma solução poderosa para empresas maduras com desafios específicos de escala e evolução, mas não são uma solução universal para todos os cenários de desenvolvimento de software.

A arquitetura de microsserviços surgiu como um avanço importante no desenvolvimento de software, apresentando alternativas inteligentes para lidar com a expansão, a rapidez e a estruturação de sistemas corporativos intrincados. No entanto, é crucial avaliar com atenção a implementação dessa abordagem, comparando as vantagens com a dificuldade extra que acompanha os sistemas descentralizados. O êxito está diretamente ligado à capacidade da empresa e à pertinência no cenário particular do projeto.