

A arquitetura de **microservices** evolui a partir de um design evolutivo, onde a decomposição de serviços possibilita o controle de mudanças frequentes e rápidas sem comprometer a qualidade do software. Uma questão fundamental ao dividir sistemas é identificar pontos onde componentes podem ser substituídos ou atualizados de forma independente. Esse princípio é explorado com a expectativa de que muitos serviços sejam descartados e recriados ao invés de serem continuamente evoluídos.

O exemplo do site do **The Guardian** ilustra essa abordagem. Embora o núcleo do site seja um monólito, novas funcionalidades são adicionadas como microservices, utilizando a API do monólito. Essa abordagem permite flexibilidade ao desenvolver funcionalidades temporárias, como páginas para eventos esportivos, e removê-las posteriormente. A modularidade é guiada pelo padrão de mudanças: componentes que mudam frequentemente são mantidos juntos, enquanto aqueles com menos alterações devem ser separados.

Uma das grandes vantagens dos microservices é o planejamento granular de lançamentos, onde apenas o serviço modificado é implantado, acelerando o processo de release. Porém, isso traz o desafio de garantir que mudanças em um serviço não quebrem seus consumidores. A versão de serviços é vista como um último recurso, preferindo-se a criação de serviços tolerantes a mudanças.

Embora a abordagem de microservices tenha sido favoravelmente recebida por muitas equipes, outras encontram dificuldades, como a complexidade deslocada para as conexões entre serviços. O sucesso dos microservices depende fortemente da habilidade de identificar corretamente as fronteiras dos componentes e de refatorar-las conforme necessário. Além disso, a competência da equipe desempenha um papel crucial na implementação eficaz de uma arquitetura de microservices.

Uma recomendação comum é começar com um monólito modular e migrar para microservices conforme a necessidade. Isso porque interfaces in-process geralmente não são adequadas para interfaces de serviços distribuídos. A experiência até o momento traz um otimismo cauteloso, mas é preciso mais tempo para avaliar a maturidade da arquitetura de microservices em longo prazo.