

Architettura a Microservizi

Prof. Fedeli Massimo

1 Perché studiare i microservizi

Oggi molte delle applicazioni che utilizziamo quotidianamente (streaming, e-commerce, social, servizi online) sono costruite utilizzando l'architettura a microservizi.

Comprendere questo modello significa capire come vengono progettati i sistemi software moderni e scalabili.

2 Dal monolite ai microservizi

Immaginiamo una applicazione per la gestione di una palestra.

Un sistema monolitico contiene:

- gestione iscrizioni
- gestione pagamenti
- gestione corsi
- gestione certificati medici

Tutto è contenuto in un unico programma.

Problemi:

- difficile da modificare
- difficile da scalare
- un errore blocca tutto il sistema

3 La soluzione: microservizi

Si suddivide il sistema in servizi separati:

- Servizio Iscrizioni
- Servizio Pagamenti
- Servizio Corsi
- Servizio Certificati

Ogni servizio è indipendente.

4 Cosa è un microservizio

Un microservizio è un programma autonomo che:

- svolge una sola funzione
- comunica con altri servizi
- può essere aggiornato senza fermare tutto il sistema

Esempio: Il servizio pagamenti gestisce solo i pagamenti.

5 Architettura generale

Una applicazione a microservizi include:

- servizi indipendenti
- API Gateway
- database separati
- comunicazione via rete

6 API Gateway

È il punto di accesso unico.

Esempio: Uno studente accede all'app della palestra.

La richiesta passa dall'API Gateway che decide:

- quale servizio coinvolgere
- come gestire la risposta

7 Esempio reale

Sistema di e-commerce:

- Servizio Ordini
- Servizio Catalogo
- Servizio Pagamenti
- Servizio Spedizioni

Se aumenta il traffico sugli ordini, si scala solo quel servizio.

8 Autonomia dei servizi

Ogni servizio può:

- usare tecnologie diverse
- avere il proprio database
- essere sviluppato da team diversi

9 Comunicazione tra servizi

Due modalità principali:

9.1 Sincrona

Un servizio chiama direttamente un altro.

Esempio: Ordini chiede al servizio pagamenti di verificare il pagamento.

9.2 Asincrona

Un servizio invia un evento.

Esempio: Pagamento completato → evento inviato → spedizioni si attiva.

10 Gestione dei dati

Ogni microservizio ha il proprio database.

Vantaggi:

- maggiore indipendenza
- migliore scalabilità

11 Problema delle transazioni

Se più servizi collaborano, non si può usare una transazione unica.

Soluzione: Saga.

Esempio: Ordine → pagamento → spedizione.

Se fallisce la spedizione, si annulla il pagamento.

12 Scalabilità

Si può aumentare solo il servizio necessario.

Esempio: Durante i saldi aumenta il servizio ordini.

13 Disponibilità

Se un servizio si blocca, gli altri continuano a funzionare.

14 Migrazione dal monolite

Si usa il Pattern Strangler.

Passaggi:

- si inserisce un API Gateway
- si sostituiscono gradualmente le funzioni

15 DDD e microservizi

Il Domain Driven Design aiuta a dividere il sistema.

Esempio palestra:

- dominio clienti
- dominio pagamenti
- dominio corsi

16 Sfide

- maggiore complessità
- gestione rete
- gestione eventi

17 Quando usarli

Non sempre sono la soluzione migliore.

Sono utili quando:

- il sistema cresce
- servono aggiornamenti frequenti
- serve alta disponibilità

18 Conclusione

I microservizi rappresentano un modello moderno per costruire software scalabile, modulare e aggiornabile.

19 Esercizio per studenti

Progetta un sistema a microservizi per una scuola.

Individua:

- i servizi
- i dati
- le comunicazioni