

# Architettura a un livello Tier 1, architettura a due livelli Tier 2, architettura a tre livelli Tier 3. Descrivere vantaggi e svantaggi, fornire degli esempi.

Le architetture software **tier 1 (monolitica)**, **tier 2** e **tier 3** (multilivello) rappresentano diversi approcci per organizzare le applicazioni in base alla loro struttura e ai componenti coinvolti. Ogni modello ha vantaggi e svantaggi, a seconda delle esigenze dell'applicazione.

---

## 1. Architettura a un livello (Tier 1)

programma che si occupa di interfaccia con utente, logica e memorizzazione  
L'architettura monolitica (single-tier) concentra tutte le funzionalità in un unico livello.

### Esempi:

- Applicazioni desktop standalone (ad esempio, software di calcolo locale).
- Semplici gestionali o applicazioni offline.

### Vantaggi:

- **Semplicità:** Facile da sviluppare, implementare e gestire.
- **Performance:** Nessuna latenza dovuta alla comunicazione tra livelli.
- **Costo:** Economica per applicazioni piccole.

### Svantaggi:

- **Mancanza di scalabilità:** Difficile da adattare ad ambienti più complessi o con molti utenti.
  - **Difficile manutenzione:** Modifiche al codice possono influenzare l'intera applicazione.
  - **Non flessibile:** Poco adatta per ambienti distribuiti.
- 

## 2. Architettura a due livelli (Tier 2)

L'architettura a due livelli separa il front-end (presentazione) dal back-end (database).  
La gestione dei dati dell'applicazione è gestita da un database e la separo con la applicazione.  
parte logica(applicazione) separata dalla parte della gestione dei dati. Modifichiamo i dati senza modificare il programma(indipendenza delle 2 parti).

### Esempi:

- Applicazioni client-server classiche (come un'applicazione desktop che si collega a un database remoto).
- Software aziendali di gestione dati.

### Vantaggi:

- **Chiarezza:** Separazione tra logica di presentazione e gestione dati.
- **Scalabilità limitata:** È possibile supportare più utenti rispetto al modello monolitico.
- **Facile da implementare:** Richiede meno risorse rispetto all'architettura a tre livelli.

### Svantaggi:

- **Dipendenza tra livelli:** Cambiamenti al database possono richiedere modifiche al client.
- **Problemi di scalabilità:** Può non supportare grandi quantità di utenti o traffico.
- **Carico del client:** Il client gestisce parte della logica applicativa.

---

### 3. Architettura a tre livelli (Tier 3)

L'architettura a tre livelli separa l'applicazione in tre componenti distinti:

1. **Presentazione:** L'interfaccia utente (browser o applicazione client).
2. **Logica applicativa:** Un server che elabora i dati.
3. **Gestione dati:** Database.

#### Esempi:

- Applicazioni web moderne (e-commerce, gestionali online).
- Sistemi ERP o CRM.

#### Vantaggi:

- **Flessibilità:** Ogni livello può essere sviluppato, aggiornato e scalato separatamente.
- **Scalabilità elevata:** Ideale per applicazioni con molti utenti.
- **Manutenzione più semplice:** Cambiamenti a un livello non influenzano gli altri.
- **Compatibilità:** Supporta ambienti distribuiti e cloud computing.

#### Svantaggi:

- **Complessità:** Richiede maggiore pianificazione e competenze tecniche.
- **Performance:** La comunicazione tra livelli può aggiungere latenza.
- **Costo:** Richiede più risorse e infrastruttura rispetto ai modelli precedenti.