

# Alcune idee sui sistemi software e la loro architettura

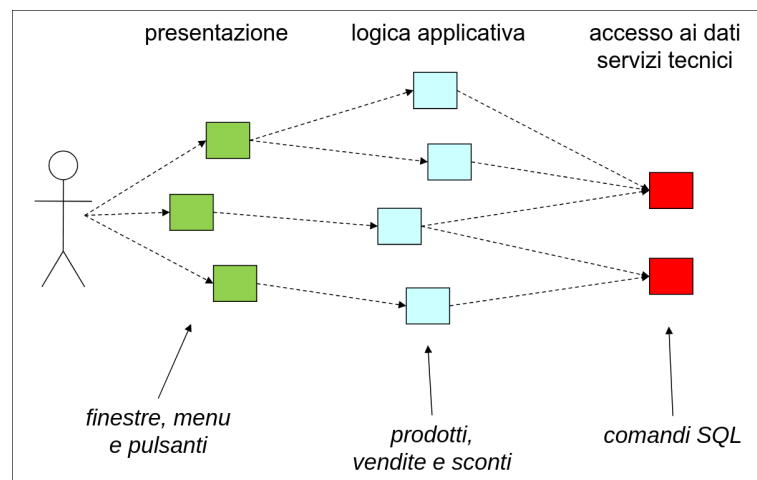
Analisi e progettazione del software

Flavio Colacicchi

13/03/2025

La scelta dell'architettura è una scelta cruciale da fare all'inizio dello sviluppo e che sarà difficile da cambiare in seguito, un'architettura comunque è quella a strati che interagiscono al seguente modo: gli strati alti possono fare chiamate a quelli più bassi ma non è vero il contrario. Una possibile scelta, minimale ma comune è la seguente

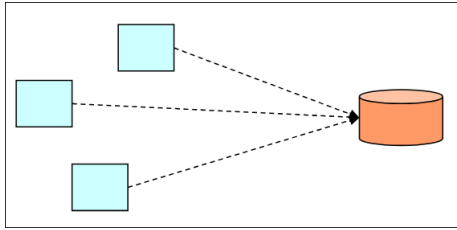
- **Presentazione**  
UI e ha lo scopo di capire cosa vuole fare l'utente per poi fare una chiamata allo strato sottostante
- **Logica applicativa**  
Delega la richiesta alla base di dati da cui otterrà dei dati che elabora e restituisce allo strato superiore per la presentazione all'utente
- **Accesso alla base di dati e altri servizi tecnici**



La scelta del modo in cui è organizzato lo strato della logica operativa influisce pesantemente sul metodo per l'analisi del software, ci sono diversi approcci:

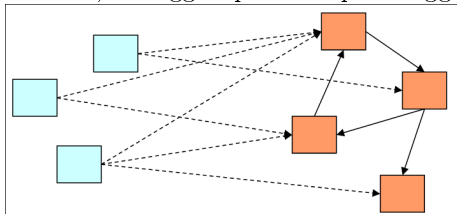
- Approccio “tradizionale”
  - I dati sono gestiti da una base di dati

- Le operazioni sono transazioni sulla base di dati, ciascun oggetto/classe della logica applicativa corrisponde a una procedura/transazione che l'utente può richiedere al sistema



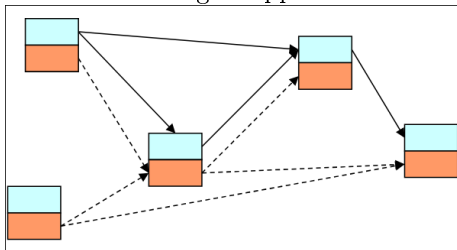
- Approccio “procedurale”

- I dati sono gestiti in memoria principale
- Alcuni oggetti sono usati per rappresentarare dati/informazioni
- Altri oggetti (separati) definiscono le operazioni che possono essere applicate alle informazioni, a maggior parte di questi oggetti di dominio incapsulano sia dati che operazioni



- Domani model

- lo strato della logica applicativa è realizzato a oggetti che si ripartiscono le responsabilità del sistema
- lo strato della logica applicativa viene anche chiamato strato di dominio

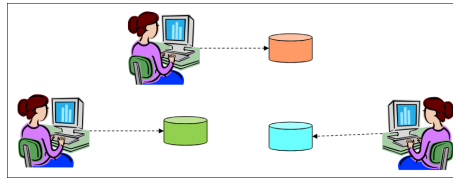


Per quanto riguarda l'organizzazione della logica applicativa, questo corso fa riferimento alla strategia Domain Model

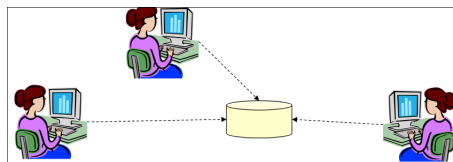
In generale un'applicazione software offre ai suoi utilizzatori un certo numero di funzionalità che sono relative alla gestione di alcuni tipi di informazioni (dati).

È utile distinguere tra due tipologie principali:

- Applicazioni stand-alone (monopoly)  
Applicazioni mono-utente i cui dati non sono condivisi con utenti diversi della stessa applicazione



- Applicazioni client-server  
Applicazioni che possono essere accedute in modo concorrente da più utenti tramite accesso in rete o sul web  
Queste applicazioni gestiscono in genere anche i dati che devono essere condivisi dall'utente



Per la progettazione

- Il caso di un'applicazione stand-alone è relativamente semplice  
È infatti sufficiente pensare a una singola istanza/esecuzione dell'applicazione, dal punto di vista di un singolo utente
- Il caso di un'applicazione client server è molto più complesso  
Da una parte, l'applicazione deve gestire alcuni dati condivisi tra tutti i suoi utenti/client  
Inoltre, per ciascun utente/client, l'applicazione deve gestire alcuni dati specifici per la conversazione/sessione con quel particolare client

Dunque un'applicazione client/server per quanto riguarda lo stato delle sessioni in questo corso ipotizziamo di utilizzare una tecnologia che consenta di ragionare (e scrivere programmi) in termini di singola conversazione/sessione. Una soluzione comune è l'introduzione di un ulteriore strato ("application") tra quello di presentazione e quello della logica

