

Alcune idee sui sistemi software e la loro architettura

drew

March 18, 2021

1 Architettura a strati

I sistemi software sono comunemente composti da diversi elementi (classi, package, ...), che spesso vengono organizzati tramite un’architettura a strati.

Un’applicazione è composta da una pila verticale di strati, che a sua volta comprende diversi package, che comprendono diverse classi.

L’utente, di solito, interagisce con lo strato più alto, ovvero l’interfaccia utente. Le richieste dell’utente propagano da questo strato verso gli strati più bassi. Le risposte, a loro volta, vengono fatte risalire dagli strati più bassi a quelli più alti.

Una possibile scelta (minimale ma comune) per gli strati è:

- presentazione (interfaccia utente)
- logica applicativa (vendita, pagamento...)
- accesso base di dati e altri servizi tecnici (accesso alla base di dati...)

Il codice è quindi organizzato a strati: le classi sono dedicate per ogni strato.

2 Sullo strato della logica applicativa

Alcune strategie principali per l’organizzazione dello strato della logica applicativa

2.1 Approccio transazionale

I dati sono gestiti in una base di dati, quindi 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.

2.2 Approccio procedurale

I dati sono gestiti in memoria principale, quindi alcuni oggetti sono usati per rappresentare dati/informazioni e altri oggetti (separati) definiscono le operazioni che possono essere applicate alle informazioni.

2.3 Domain Model

Questo strato della logica applicativa è realizzato a oggetti, che si ripartiscono la responsabilità del sistema - la maggior parte degli oggetti di dominio encapsulano sia dati che operazioni

2.4 Applicazioni stand-alone e client-server

In generale, un'applicazione software si occupa di offrire ai suoi utenti l'esecuzione di un certo numero di *funzionalità* relative alla gestione di alcune tipologie di *informazioni (dati)*.

Si possono distinguere due tipologie di applicazioni:

- **Applicazioni stand-alone**

Applicazioni mono-utente, i cui dati non sono condivisi tra utenti diversi della stessa applicazione. I dati vengono memorizzati localmente nel computer dell'utente. Un esempio è un'applicazione che permette di gestire la rubrica memorizzata su un file del computer.

- **Applicazioni client-server**

Applicazioni che possono essere usate contemporaneamente da più utenti, in rete o sul web. Queste applicazioni gestiscono in genere anche dati che devono essere condivisi da più utenti, che vengono memorizzati quindi su una base di dati dell'applicazione e non sul computer dell'utente che vi accede.

3 Progettazione per applicazioni client-server

Nella progettazione di applicazioni client-server viene introdotto un ulteriore strato, chiamato *application*, tra quello di presentazione e quello della logica applicativa.

L'*application layer* si occupa della gestione dello stato delle sessioni, ovvero gli utenti che in un momento stanno accedendo a dati nel *domain layer*.

In pratica, nello sviluppo delle applicazioni client-server nell'analisi e progettazione, si devono fare ragionamenti opportuni (e separati) per quanto riguarda la gestione dei dati condivisi dell'applicazione e la gestione dello stato della sessione di un client. Inoltre, nella programmazione per queste applicazioni si dovranno usare delle opportune tecnologie, e si dovranno prendere in considerazione diversamente quelle parti del progetto che riguardano la gestione dei dati condivisi da quelle che riguardano la gestione dello stato della sessione.