

Paradigma MVC (Model, View, Controller)

La chiave per il successo delle applicazioni consiste nel pianificare correttamente il layout del sistema prima di iniziare a programmare e a sviluppare.

Costruiamo un'auto

La creazione di un'applicazione non differisce molto dall'assemblaggio di un'auto. I singoli componenti dell'auto (carrozzeria, ruote, elettronica, motore) vengono prima assemblati singolarmente.

Ogni componente completato viene poi collocato all'interno del telaio dell'auto.

Una volta assemblata l'auto, tutti i componenti lavorano assieme.

Se un componente si rompe può essere sostituito senza dover sostituire o modificare altri componenti dell'auto.

Programmazione modulare

L'idea di programmazione modulare si basa sulla metodologia dei blocchi di codice che possono essere creati singolarmente per essere assemblati con altri moduli per produrre un'applicazione funzionante (la nostra auto). I moduli possono essere modificati o sostituiti senza dover modificare altri moduli.

Abbiamo fondamentalmente tre moduli (livelli): dati, interfaccia, regole operative.

Dati (Model)

Gestisce l'accesso e la logica legata al **database**.

- Si occupa di interagire con il database
- Contiene le regole per manipolare i dati (INSERT, DELETE, UPDATE, SELECT)
- Non sa nulla dell'interfaccia utente

Interfaccia (View)

È la parte che l'utente vede, l'interfaccia grafica:

- Mostra i dati all'utente in modo comprensibile
- Raccoglie input dall'utente
- Non elabora direttamente i dati, li riceve già pronti

Regole operative (Controller)

È il "direttore d'orchestra" che coordina tutto:

- Riceve le richieste dell'utente

1

- Chiede al Model i dati necessari
- Passa i dati alla View giusta per mostrarli

Coordina il flusso di dati tra Model e View

Facciamo un esempio teorico: un'applicazione per visualizzare il profilo di un utente.

Model: Una classe PHP che si connette al database e recupera le informazioni dell'utente Controller: Riceve la richiesta "mostra profilo dell'utente X", chiede i dati al Model e li passa alla View View: Un template HTML con segnaposti che verranno riempiti con i dati dell'utente

Questo approccio rende il codice:

- Più organizzato e facile da gestire
- Più facile da modificare (potete cambiare l'aspetto senza toccare la logica)
- Più adatto al lavoro in team (un designer può lavorare sulle View mentre un programmatore sui Model)

IN SINTESI

Il livello dei dati utilizza istruzioni SQL (INSERT, DELETE, UPDATE, SELECT) per recuperare ed aggiornare informazioni in un DB (MODEL)

Tuttavia, la richiesta di modificare le informazioni proviene sempre dal livello delle regole operative. (CONTROLLER)

Per esempio, un utente che richiede una modifica del suo indirizzo in un'applicazione web immetterà tale richiesta nel livello dell'interfaccia (VIEW); la richiesta di aggiornamento verrà poi gestita nel livello delle regole operative (CONTROLLER) e poi le informazioni nel database verranno modificate tramite il livello dei dati (MODEL).

Vantaggi dell'uso di MVC

- Separazione chiara tra logica, dati e interfaccia.
- Codice riutilizzabile e organizzato.
- Facilità di manutenzione e test.

