

GABRIELE ROMANATO

Storia del design pattern MVC

Il pattern Model-View-Controller (MVC) è uno dei design pattern più celebri e utilizzati nella progettazione software. La sua importanza risiede nella capacità di separare la logica dell'applicazione dalla presentazione e dal controllo dell'input, favorendo così un'architettura modulare, riutilizzabile e facilmente manutenibile. Esploriamo la storia di questo pattern e alcune delle sue implementazioni più note.

Origini del Pattern MVC

Il concetto di MVC è stato introdotto negli anni '70 da Trygve Reenskaug, un informatico norvegese, durante il suo lavoro sui sistemi grafici Smalltalk-76 presso il Xerox Palo Alto Research Center (PARC). Reenskaug intendeva creare un modello che permettesse agli utenti di interagire con il sistema in modo più intuitivo e che separasse chiaramente i componenti del software per facilitare lo sviluppo e la manutenzione.

Il pattern MVC è strutturato in tre componenti principali:

1. **Model (Modello):** Rappresenta la logica di business e lo stato dell'applicazione. Gestisce i dati e le regole di business, comunicando con il database e altri servizi di persistenza.
2. **View (Vista):** Gestisce la presentazione dei dati all'utente. È responsabile del rendering dell'interfaccia utente e dell'aggiornamento della visualizzazione quando i dati del modello cambiano.
3. **Controller (Controllore):** Interpreta l'input dell'utente e invia comandi al modello e/o alla vista. Funziona come un intermediario tra il modello e la vista.

Implementazioni Celebri del Pattern MVC

Nel corso degli anni, il pattern MVC è stato adottato e implementato in numerosi framework e linguaggi di programmazione. Alcune delle implementazioni più celebri includono:

1. **Smalltalk-80**: La prima implementazione completa di MVC. Smalltalk-80 ha dimostrato l'efficacia del pattern nella progettazione di interfacce utente complesse, diventando un punto di riferimento per gli sviluppatori.
2. **Ruby on Rails**: Un framework web scritto in Ruby, Ruby on Rails ha reso il pattern MVC popolare tra gli sviluppatori web. Rails utilizza MVC per separare chiaramente la logica di business (model), la logica di presentazione (view) e il controllo dell'input (controller), facilitando lo sviluppo di applicazioni web complesse.
3. **ASP.NET MVC**: Microsoft ha adottato il pattern MVC per il framework ASP.NET, offrendo agli sviluppatori un modo strutturato di creare applicazioni web. ASP.NET MVC permette di creare applicazioni testabili e manutenibili, separando le responsabilità delle diverse componenti.
4. **AngularJS**: Sebbene non sia una implementazione MVC classica, AngularJS di Google ha introdotto un pattern simile chiamato MVVM (Model-View-ViewModel), che incorpora molte delle stesse idee di separazione delle preoccupazioni presenti in MVC. AngularJS è stato ampiamente utilizzato per lo sviluppo di applicazioni web dinamiche.
5. **Django**: Un framework web Python che implementa una variante di MVC chiamata MTV (Model-Template-View). Django è noto per la sua "filosofia di batteria inclusa", che fornisce strumenti e convenzioni per lo sviluppo rapido di applicazioni web.

Conclusioni

Il pattern MVC ha avuto un impatto duraturo sulla progettazione software, favorendo la modularità e la manutenibilità del codice. Le sue

implementazioni in diversi framework hanno dimostrato la sua versatilità e utilità in vari contesti di sviluppo. Oggi, MVC continua a essere una scelta popolare tra gli sviluppatori, evolvendosi e adattandosi alle nuove tecnologie e paradigmi, ma mantenendo sempre la sua essenza di separazione delle preoccupazioni che lo rende così efficace.