



DelphiDay

italian conference

MVC in Delphi

Sviluppare facilmente in Delphi con il modello MVC



Cleber M Grings

Centro Software - Sviluppatore

@ grings@grings.it

github.com/grings

in linkedin.com/in/grings



11-12 Giugno 2024
Piacenza





AGENDA

1. Cos'è MVC
2. Possibilità
3. Ma anche
4. Delphi



Cos'è MVC

1



MVC è un pattern architetturale

- È un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito della programmazione orientata agli oggetti e in applicazioni web, in grado di separare la logica di presentazione dei dati dalla logica di business.



Storia di MVC

- MVC architettura prima discusso nel 1979 da Trygve Reenskaug
- MVC modello è stato introdotto nel 1987 nel linguaggio di programmazione Smalltalk.
- MVC è stato accettato per la prima volta come concetto generale, in un articolo del 1988
- Negli ultimi tempi, il pattern MVC è ampiamente utilizzato nelle moderne applicazioni



Caratteristiche di MVC

- Testabilità facile e senza attrito.
- Chiara separazione della logica: Modello, Vista, Controller.
Separazione delle attività applicative vale a dire. logica aziendale, logica UI e logica di input.
- Tre importanti MVC i componenti sono:
 - Controller: Un'interfaccia tra Model e View componenti
 - Modello: Include tutti i dati e le relative logica
 - Visualizzazione: Presentare i dati dell'utente o gestisce l'interazione dell'utente



View

- Un punto di Vista è che la parte dell'applicazione che rappresenta la presentazione dei dati.,
- Le viste vengono create dai dati raccolti dai dati del modello. Una vista richiede al modello di fornire informazioni in modo che rinvii la presentazione di output all'utente.
- La vista rappresenta anche i dati di chat, diagrammi e tabelle. Ad esempio, qualsiasi vista cliente includerà tutti i componenti dell'interfaccia utente come caselle di testo, menu a discesa, ecc.



Controller

- Il Controller è quella parte dell'applicazione che gestisce l'interazione dell'utente. Il controller interpreta gli input del mouse e della tastiera da parte dell'utente, informando modello e la vista di cambiare a seconda dei casi.,
- Un Controller invia i comandi al modello per aggiornarne lo stato(ad esempio, salvando un documento specifico). Il controller invia anche comandi alla vista associata per modificare la presentazione della vista (ad esempio scorrendo un particolare documento).



Model

- Il componente del modello memorizza i dati e la relativa logica. Rappresenta i dati che vengono trasferiti tra i componenti del controller o qualsiasi altra logica di business correlata. Ad esempio, un oggetto Controller recupererà le informazioni del cliente dal database., Manipola i dati e li rimanda al database o li usa per renderizzare gli stessi dati.
- Risponde alla richiesta delle viste e risponde anche alle istruzioni del controller per aggiornarsi. È anche il livello più basso del modello che è responsabile del mantenimento dei dati.



Possibile usi

2



Quache possibilità

- Possiamo utilizzare più engine per i database (FireDAC, UniDAC, dbExpress, ...)
- Multiple View (VCL, FireMonkey, UniGUI, Web)
- Sfruttare i controlli e Modelli sia della VCL o API



Ma anche

3



S.O.L.I.D/Clean Code

- S - Il principio di singola responsabilità (SRP)
- O - Il principio aperto/chiuso (OCP)
- L - Il principio di sostituzione di Liskov (LSP)
- I - Il principio di segregazione delle interfacce (ISP)
- D - Il principio di inversione delle dipendenze (DIP)



Pattern e altri

- Interface
- Fluente
- Pattern Factory
- Overload
- Injection



Delphi

4



Esempi

- <https://github.com/grings/DelphiDay>



Grazie mille