

tesi di laurea

Il Pattern MVC nei Framework di sviluppo per applicazioni Web. Analisi e comparazione di SPRING MVC Framework e ASP.NET MVC Framework.

Anno Accademico 2008/2009

relatore

Ch.mo prof. Porfirio Tramontana

correlatore

Ing. Luca Anniciello

candidato

Francesco Maisto

Matr. 534/1568



- **Obiettivi del progetto:**
 - Analizzare due framework di sviluppo Open Source basati sulle piattaforme .NET e Java che implementano il pattern MVC
 - Sviluppare una applicazione Web MVC modello/demo nei due framework che interagisce con una semplice base dati relazionale scelta in base alla comodità di applicazione
- **Necessità dell'azienda Aera:**
 - Valutare l'adozione del paradigma MVC e delle tecnologie che lo implementano per applicazioni Web che interagisco con database relazionali

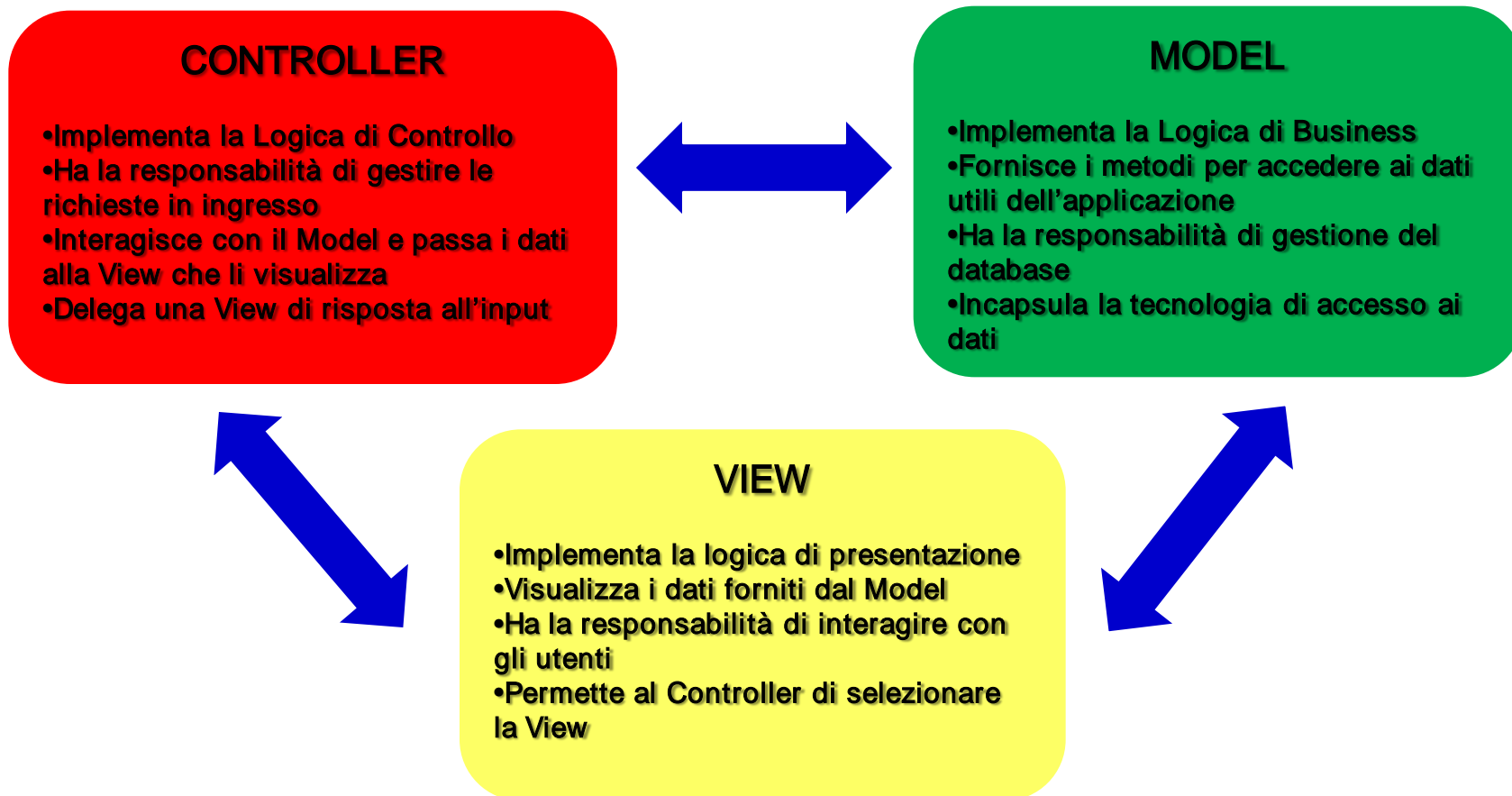
Framework

- **Mondo Web veloce e in continuo sviluppo**
 - Sempre più applicazioni Web di medie e grandi dimensioni con accesso a basi di dati
 - Necessità di produrre software in modo efficace ed efficiente, di qualità e con costi contenuti
- **Progettisti e sviluppatori si rivolgono ai Framework di sviluppo che adottano soluzioni già testate: design pattern**
- **Framework: strutture di supporto su cui un software può essere progettato e sviluppato**



Pattern Model-View-Controller

- Architectural Pattern, tilizzato per dividere l'applicazione in 3 componenti con ruoli e funzionalità ben distinte



Framework MVC

- Due tra i framework che hanno riscosso maggior successo:
SPRING MVC Framework ASP.NET MVC Framework



Spring

Java Application Framework



- SPRING MVC: tecnologia Web basata su piattaforma Java
- ASP.NET MVC: tecnologia Web basata su piattaforma .NET

- Entrambi Open Source

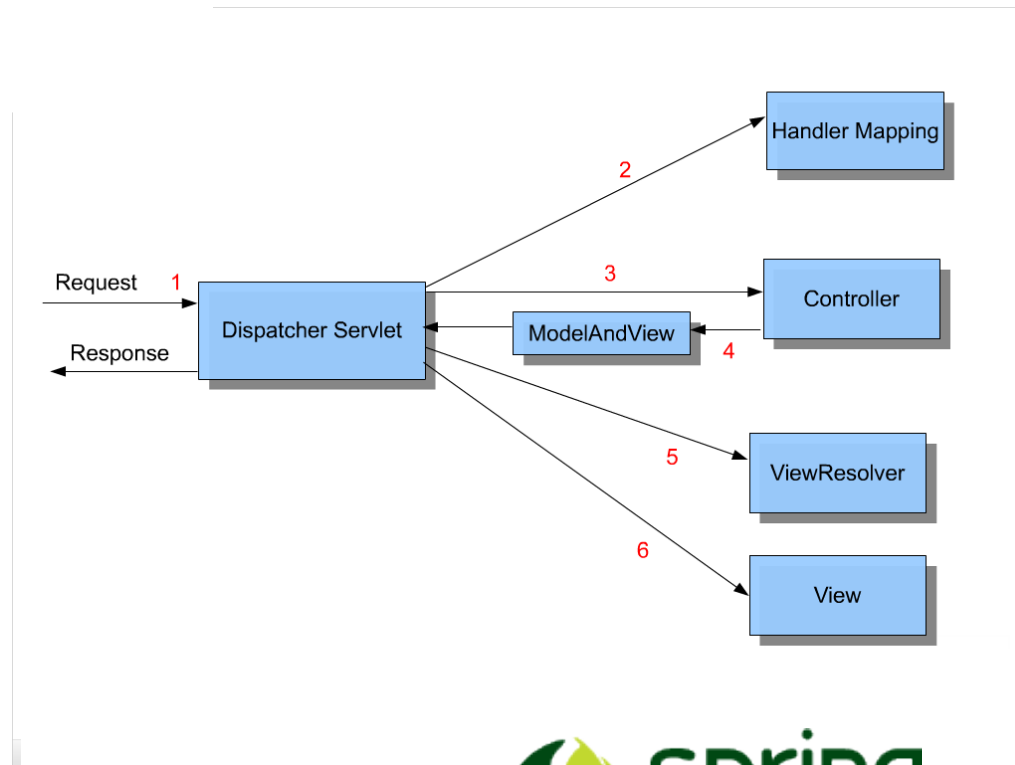
Rappresenta una delle soluzioni MVC più adottate nel mondo Java

È realizzato con il pattern Front Controller, implementato dalla DispatcherServlet: tutte le richieste sono filtrate dalla Servlet che ne gestisce il flusso esecutivo delegando i componenti Controller e View .

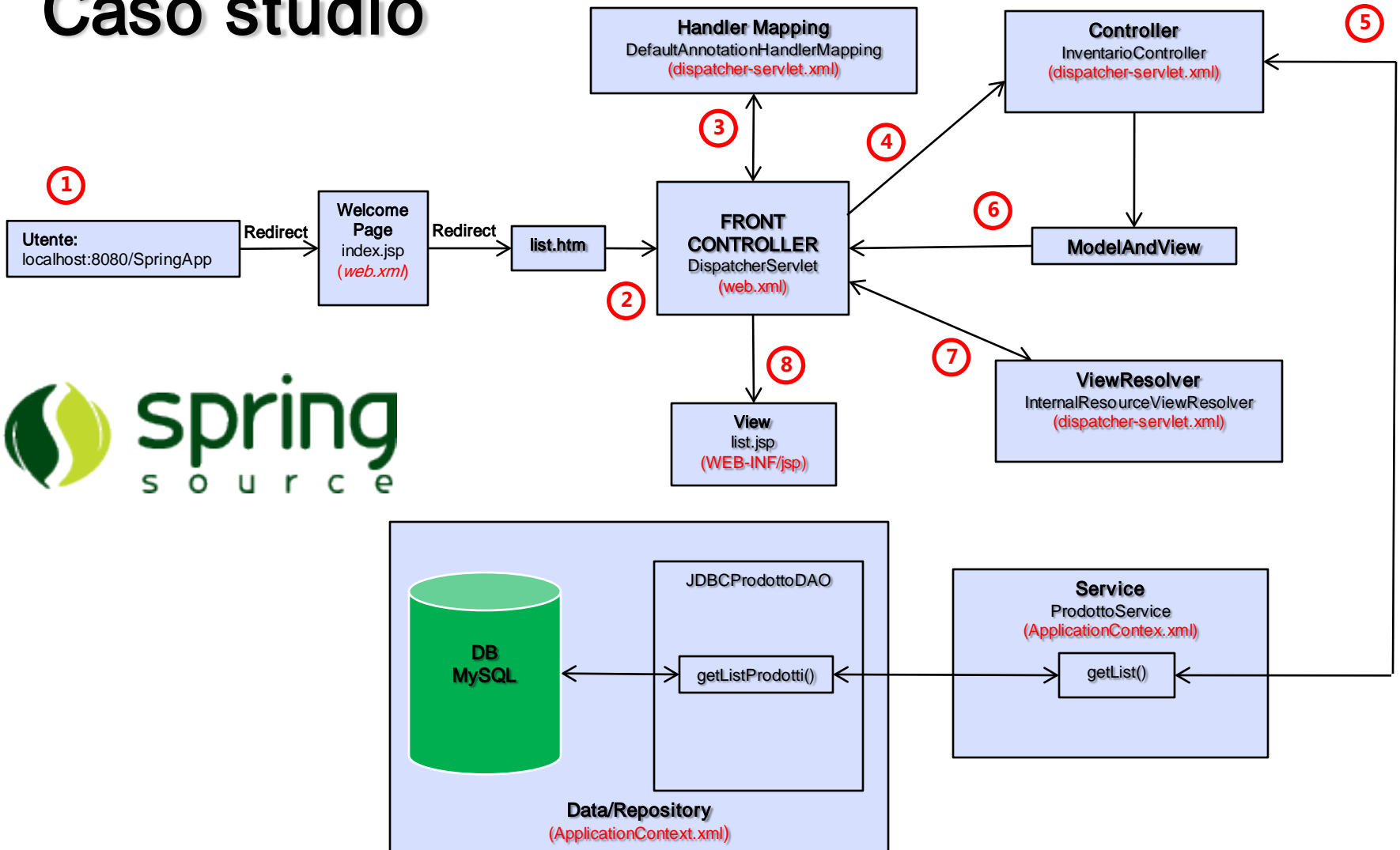
Fa un uso estensivo della Dependency Injection (DI), tecnica di Inversion Of Control introdotta da Martin Fowler: implementata con metodi Setter e dipendenze delle classi definite in un file di configurazione XML.

Il Pattern MVC nei Framework di sviluppo per applicazioni Web. Analisi e comparazione di SPRING MVC Framework e ASP.NET MVC Framework.

DispatcherServlet



Caso studio



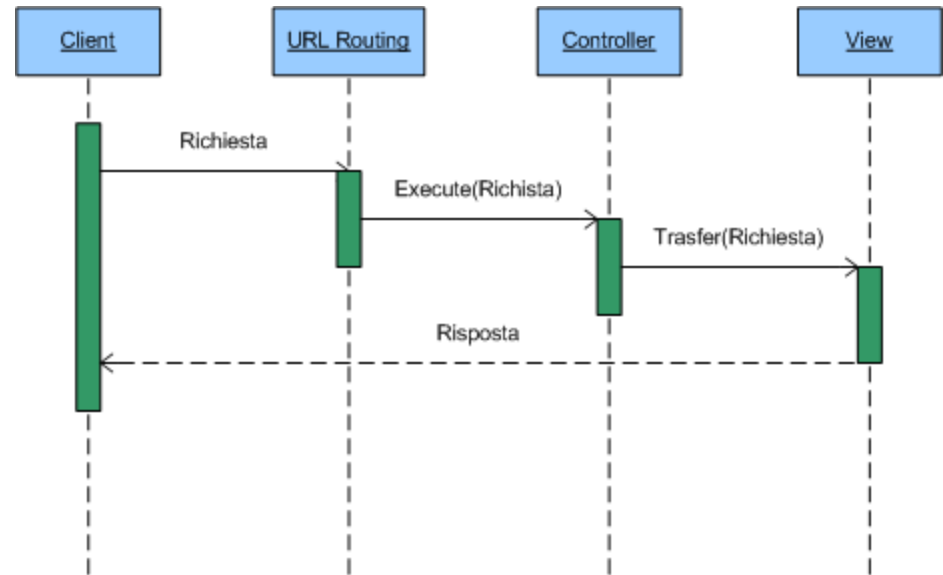
Tecnologia di casa .NET, è una tecnologia emergente che sta andando affermandosi sempre di più nel mondo Microsoft

MVC è realizzato con il Pattern Front Controller, implementato da un meccanismo di routing, chiamato ASP.NET Routing: tutte le richieste sono filtrate da un modulo HTTP. Delega per ogni richiesta l'azione del Controller che ha il compito di gestirla.

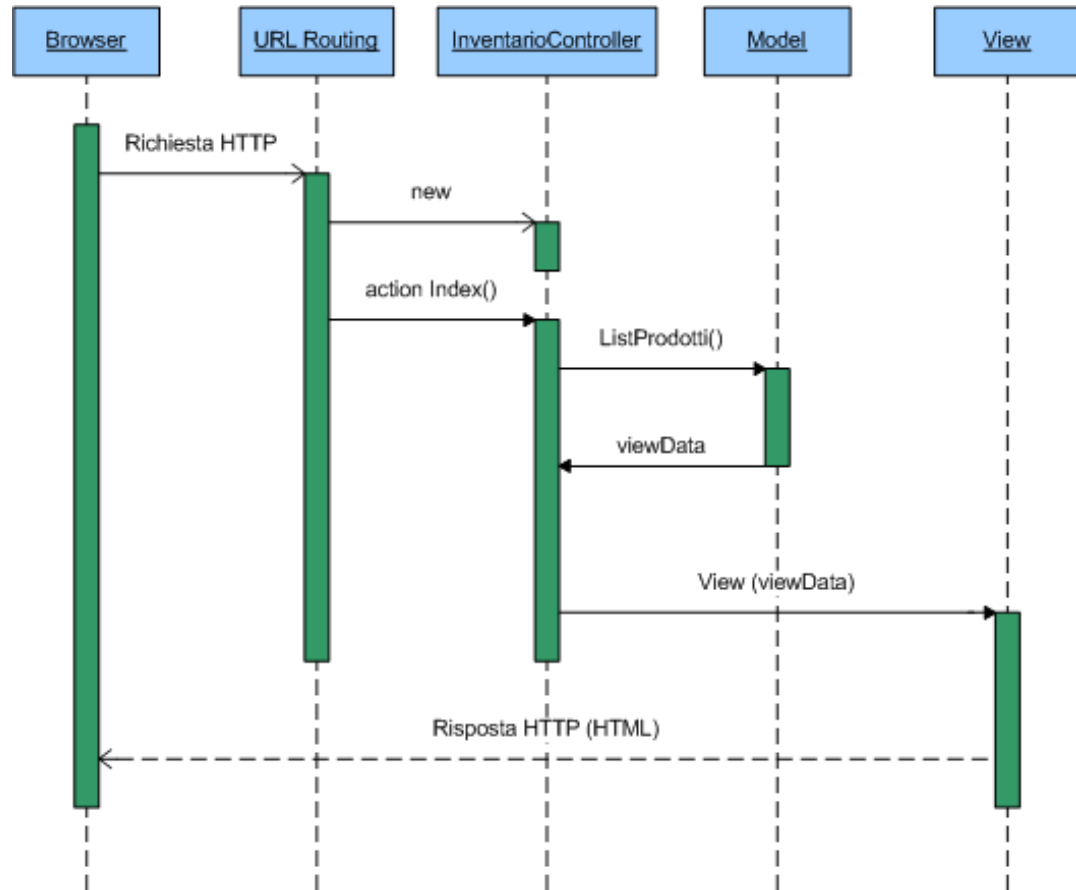
Implementa il concetto di Convention Over Configuration

Utilizza la Dependency Injection (DI): è implementata con un metodo Constructor che prevede due costruttori in una classe. La dipendenza è definita nel primo costruttore.

ASP.NET Routing



Caso studio



Confronto

- **SPRING MVC**
 - Utilizza Front Controller
 - Controller: interagisce con Model e crea oggetto ModelAndView che restituisce alla DispatcherServlet. È la DS che delega la View.
 - Utilizza la DI per rendere applicazioni Loosely Coupled
 - Model: Composto da Repository Layer e Service Layer. Il Repository Layer è implementato con pattern DAO.
- **ASP.NET MVC**
 - Utilizza Front Controller
 - Controller: interagisce con Model e delega direttamente la View a cui passare il controllo.
 - Utilizza la DI per rendere applicazioni Loosely Coupled
 - Model: Composto da Repository Layer e Service Layer. Il Repository Layer è implementato con Repository Pattern.

Conclusioni

- **SPRING MVC e ASP.NET MVC risultano essere due valide alternative per lo sviluppo di applicazioni Web, con maggiore interesse verso quelle di medie e grandi dimensioni**
- **Vantaggi:**
 - **Adatti allo sviluppo parallelo**
 - **Alta manutenibilità: entrambi i framework utilizzano la DI per rendere le applicazioni Web Loosely Coupled. Il basso accoppiamento permette di modificare un componente o addirittura sostituirne la tecnologia senza dover apportare modifiche agli altri componenti.**
 - **Gestione della complessità: all'aumentare della dimensione dell'applicazione la complessità di gestione resta pressoché costante dato che le modifiche sono distribuite nei tre componenti Model, View e Controller.**
- **Svantaggi**
 - **Non sono soluzioni adatte per applicazioni Web con alto grado di interazione con l'utente.**