

Scelte di Progetto: Lido Codici Sballati

Caso A - Ristrutturazione

Francesca Corrente (1087460)

Davide Dell'Anno (1085788)

Università degli Studi di Bergamo

1 Architettura e Pattern MVC

Per la ristrutturazione del primo progetto (Caso A) si è optato per l'utilizzo di **Java**, **Tomcat** e **Thymeleaf**. L'applicativo è stato re-ingegnerizzato seguendo rigorosamente il pattern architetturale **Model-View-Controller (MVC)**.

- **Model:** Implementato tramite classi Java (es. `Cliente.java`) e gestito tramite pattern DAO (`ClienteDAO.java`) per incapsulare la logica di accesso ai dati.
- **Controller:** Implementato tramite classi `HttpServlet` (es. `PrenotaServlet.java`). I controller gestiscono le richieste HTTP, elaborano la business logic comunicando con il database e passano i dati al motore di templating.
- **View:** Sviluppata utilizzando **Thymeleaf**. Questo ha permesso di rimuovere completamente la logica Java dalle pagine HTML, garantendo una netta separazione delle responsabilità rispetto allo sviluppo in JSP o PHP nativo.

2 Gestione del Database

La connessione al database MySQL è centralizzata tramite la classe `DatabaseManager.java`. Per garantire l'efficienza e prevenire l'apertura di connessioni multiple ridondanti, la classe implementa il pattern creazionale **Singleton**. Tutte le interrogazioni SQL sfruttano i `PreparedStatement` per garantire la sicurezza contro attacchi di tipo SQL Injection. Per operazioni critiche, come la registrazione di un contratto e il simultaneo aggiornamento della disponibilità dei posti, è stata implementata la gestione manuale delle transazioni (`conn.setAutoCommit(false)` con successivo `commit()` o `rollback()`), garantendo le proprietà ACID in caso di concorrenza.

3 Automazione del Deploy (Zero-Config)

In conformità con il vincolo di verifica rapida in ambiente locale, è stato sviluppato un layer di automazione multiplatforma (`avvia.bat` per Windows e `avvia.sh` per sistemi Unix-like). Considerando l'assenza garantita di software pre-installato come Maven o MySQL nell'ambiente di test, gli script sono progettati per agire come bootstrap completi. Utilizzando le interfacce a riga di comando (CLI) del sistema operativo, il progetto risolve le dipendenze mancanti, inietta il demone del database, auto-rileva le credenziali locali (effettuando l'override dinamico nel file `DatabaseManager.java`), scarica le dipendenze Maven e avvia il container Tomcat integrato, azzerando i tempi di configurazione manuale.