Università degli Studi di Bergamo

Corso di Programmazione Web

SCELTE DI PROGETTO:
Lido codici sballati

Caso A - Ristrutturazione

Progetto sviluppato da:

Francesca Corrente (matricola: 1087460)
Davide Dell'Anno (matricola: 1085788)

Indice

1 Architettura e Pattern MVC **1**

2 Gestione del Database **1**

3 Automazione del Deploy (Zero-Config) **1**

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 scelta di adottare **MySQL** come DBMS è stata dettata da ragioni di semplicità e continuità: avendo già impiegato con successo questa tecnologia in combinazione con PHP durante lo sviluppo del primo progetto.

La connessione al database 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);
// Esecuzione query...
conn.commit();
```

Questo garantisce 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 multipiattaforma. Considerando l'assenza garantita di software pre-installato, gli script agiscono come bootstrap completi:

- **Windows:** `avvia.bat`
- **Unix-like:** `avvia.sh`

Utilizzando le interfacce a riga di comando (CLI), il progetto risolve le dipendenze, inizializza il database, auto-rileva le credenziali locali effettuando l'override dinamico nel file `DatabaseManager.java` e avvia il container Tomcat integrato, azzerando i tempi di configurazione manuale.