

Progetto informatica/TPSIT: 'Memorizzazione e gestione ristoranti della provincia di Treviso'

1.Descrizione del progetto

Il progetto consiste nel creare un'applicazione java, che gestisce le informazioni relative ai ristoranti presenti nella provincia di Treviso. L'utente ha la possibilità da questa applicazione di visualizzare i ristoranti della provincia, di aggiungere nuovi ristoranti e di eliminare quelli già esistenti

2.Architettura del sistema

Il sistema è formato complessivamente da un database, due pagine web con linguaggio PHP e un'applicazione in linguaggio Java.

Il database è composto da tre tabelle, nominate Comuni, Ristoranti e Servizi. La tabella Ristoranti ha due chiavi esterne, di cui una rappresenta il legame con la tabella Comuni, mentre l'altra rappresenta il legame con la tabella Servizi. La tabella Comuni ha i seguenti attributi: CAP, che è la chiave primaria, e nome. La tabella Ristoranti, invece, ha i seguenti attributi: cod_rist (chiave primaria), tipologia, telefono, valutazione_media, descrizione, CAP (chiave esterna), id_serv (chiave esterna). Infine, gli attributi della tabella Servizi sono: ID_Serv (chiave primaria), asporto, cons_posto, ser_domicilio, rist_pranzo, rist_cena, Wi-Fi, toilette.

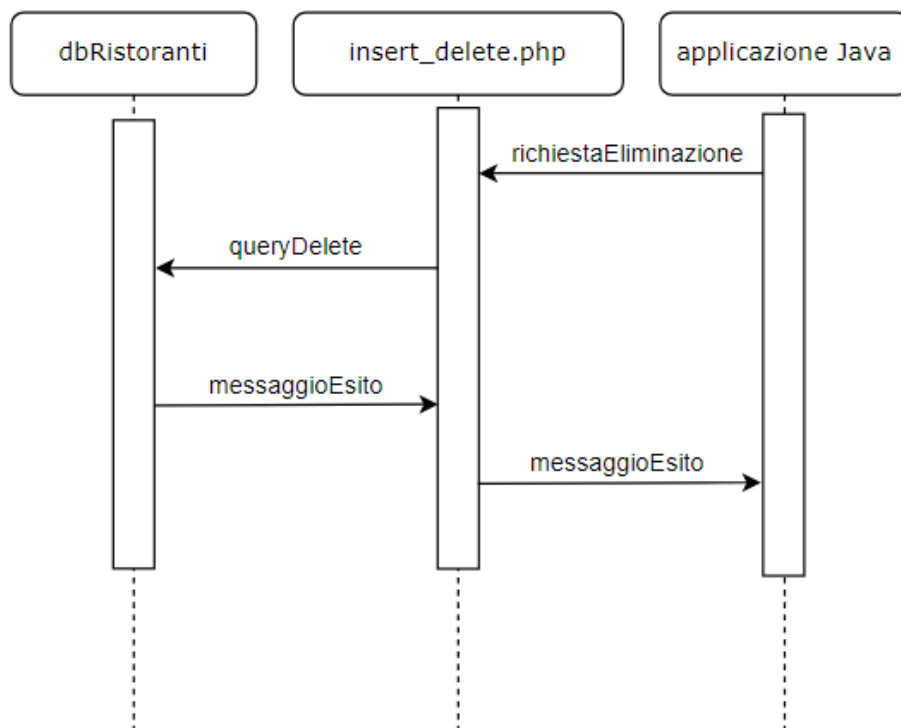
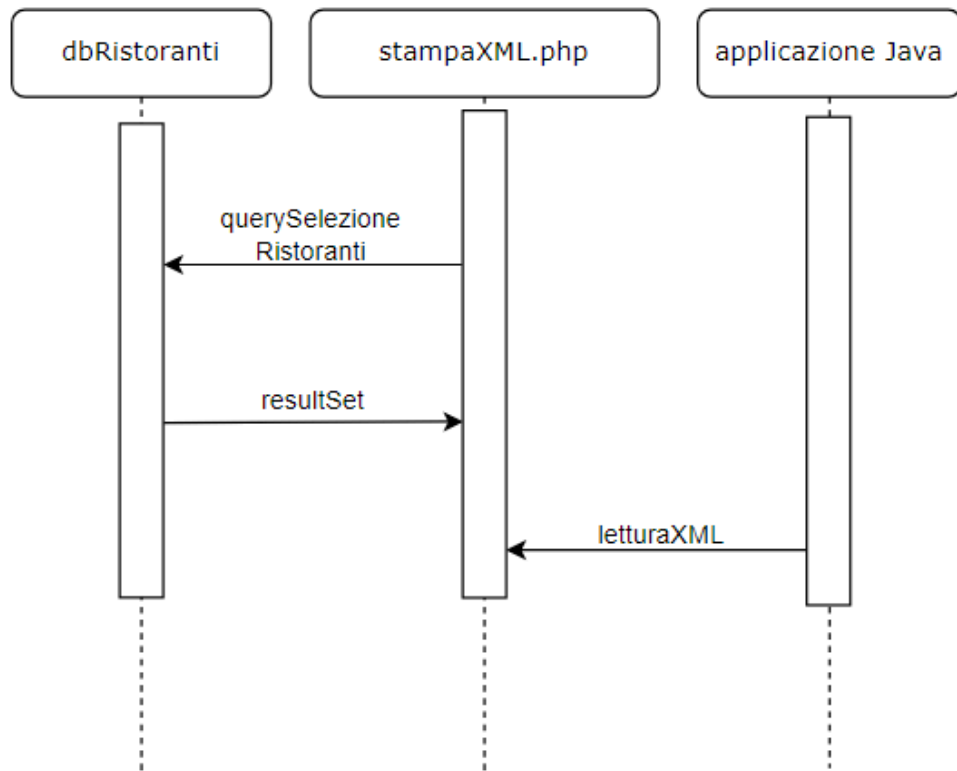
In una delle due pagine web, invece, si va a stampare a schermo i dati prelevati dal database, attraverso la costruzione di un documento XML definito da un file XSD.

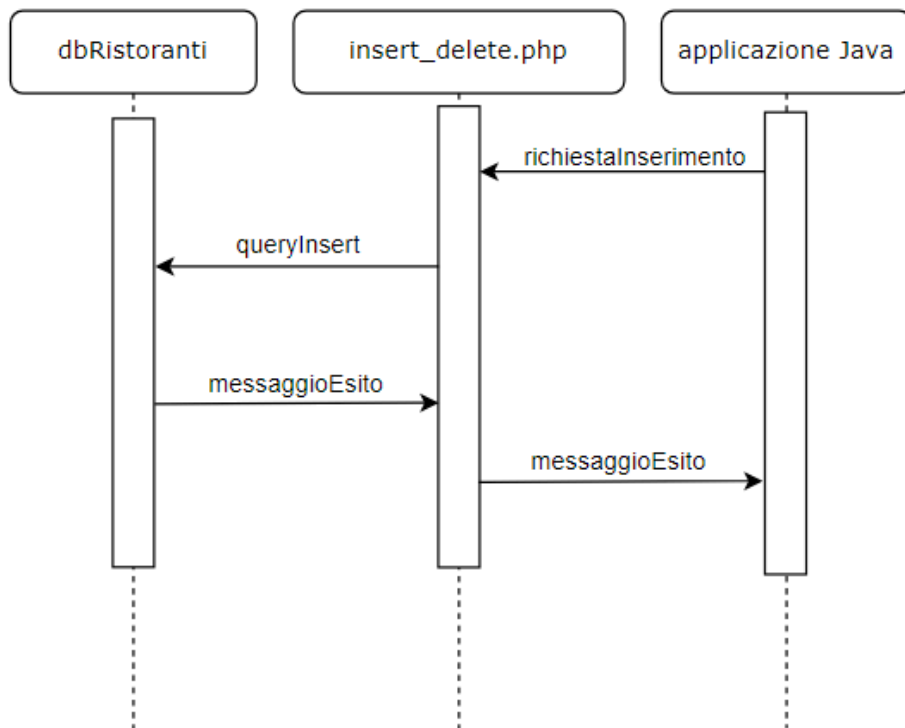
Dopo di che, l'applicazione in Java va a leggere i dati del documento XML, convertendoli in oggetti, per renderli visualizzabili su interfaccia grafica all'utente. Inoltre, l'applicazione Java è in grado di inviare dei determinati dati di un ristorante ad una seconda pagina PHP. Dopo averli inviati, questa pagina dovrà occuparsi dell'inserimento di un nuovo ristorante o dell'eliminazione di un ristorante già esistente, a seconda dei dati che riceve e della richiesta fatta dall'utente dall'applicazione.

Architettura del sistema

Gli schemi di architettura del database sono riportati alla fine della documentazione, come allegati. Qui sono riportati i diagrammi di sequenza.

Diagrammi di sequenza:





3.Descrizione del Web-service:

Il Web-service ha tre scopi principali, che corrispondono alle funzionalità delle due pagine PHP descritte nel punto precedente. In sintesi, questi scopi sono:

- Fornire un documento XML, in grado di rappresentare tutte le informazioni relative ai ristoranti;
- Aggiungere un nuovo ristorante all'interno del database;
- Eliminare un ristorante già esistente nel database.

Per il primo punto, la pagina PHP che si occupa di fornire il documento XML ha soltanto bisogno di ricevere in input i dati dei ristoranti del database.

Per la pagina PHP che si occupa, invece, di fornire l'inserimento e l'eliminazione di un ristorante, c'è bisogno di utilizzare innanzitutto un parametro in input per entrambi i casi: una stringa che rappresenta la parola 'aggiungere' o la parola 'eliminare', per fare in modo che la pagina possa capire se deve fare un inserimento o un'eliminazione. Dopo questo parametro, ne avrà bisogno di altri, a seconda della richiesta.

Parametri di input per l'inserimento di un ristorante:

- codice del ristorante da aggiungere;
- sito web/social del ristorante;
- tipologia di ristorante;
- locazione;
- numero di telefono;
- valutazione_media;

-descrizione;

-nome del comune in cui è situato il ristorante;

-tutti i servizi offerti dal ristorante;

Nota: la descrizione e il sito web/social non sono parametri obbligatori.

Per l'eliminazione, è necessario solo un altro parametro, ovvero il codice del ristorante da eliminare.

Esempio di produzione di un documento XML di un ristorante:

Si ipotizza che nel database ci sia un solo ristorante, e che nella tabella Comuni e Servizi ci siano le tuple che ci servono. Le caratteristiche del ristorante sono:

- Codice: 0001;
- Nome: 'San Michele';
- Sito web/social: www.sanmichele.it;
- Tipologia: 'pizzeria';
- Locazione: 'Via Galilei 14/H';
- Telefono: 550 231 6345;
- Valutazione media: 4.5;
- Descrizione: null;
- CAP: 31059;
- ID servizi: 0001.

La pagina richiede le informazioni relative ai ristoranti. Successivamente, fornirà il seguente risultato, stampato su pagina:

```
<?xml version="1.0" ?>
```

```
<provinciaTreviso>
```

```
  <ristorante cod_rist=0001>
```

```
    <nome>San Michele</nome>
```

```
    <sitoWeb_social>www.sanmichele.it</sitoWeb_social>
```

```
    <tipologia>pizzeria</tipologia>
```

```
    <telefono>550 231 6345</telefono>
```

```
    <val_media>4.5</val_media>
```

```
    <descrizione></descrizione>
```

```
    <comune CAP=31059>Zero Branco</comune>
```

```
    <servizi id_serv=0001>
```

```
      <asporto>true</asporto>
```

```
      <cons_posto>true</cons_posto>
```

```
      <ser_domicilio>false</ser_domicilio>
```

```
      <pranzo>true</pranzo>
```

```
      <cena>true</cena>
```

```
      <Wi-fi>false</Wi-Fi>
```

```
      <toilette>true</toilette>
```

```
    </servizi>
```

```
  </ristorante>
```

```
</provinciaTreviso>
```

Esempio di inserimento di un ristorante:

Si ipotizza l'aggiunta di un ristorante con i seguenti parametri:

- Codice: 0002;
- Nome: 'Don Mario';

- Sito web/social: null;
- Tipologia: 'italiana meridionale';
- Locazione: 'Via Foscolo 15/F';
- Telefono: 540 256 2355;
- Valutazione media: 3.5;
- Descrizione: null;
- CAP: 31058;
- Servizio asporto: no;
- Consumazione sul posto: sì;
- Servizio a domicilio: sì;
- Servizio pranzo: no;
- Servizio cena: sì;
- Wi-Fi: sì;
- Servizio toilette: sì.

La pagina PHP cerca di inserire il nuovo ristorante nel database. Come risultato, egli fornisce un messaggio di conferma, se l'operazione è avvenuta correttamente, altrimenti fornisce un messaggio di errore.

Esempio di eliminazione di un ristorante:

Si ipotizza l'eliminazione di un ristorante con il seguente parametro:

- Codice del ristorante:0002;

La pagina PHP cerca di eliminare il ristorante che ha questo codice, inviando la query di eliminazione. Come risultato, fornisce un messaggio di conferma per indicare che l'operazione è avvenuta.

4.Requisiti del progetto:

Requisiti funzionali del sistema:

- Il Web service deve scrivere il documento XML nel modo corretto;
- L'applicazione deve convertire i dati in XML in oggetti Java nella maniera corretta;
- L'applicazione deve inviare correttamente i parametri per l'inserimento di un nuovo ristorante;
- L'applicazione deve inviare nella maniera corretta i parametri per l'eliminazione di un nuovo ristorante;
- Il Web Service deve essere in grado di fornire i messaggi sugli esiti degli inserimenti e delle eliminazioni dei ristoranti;

Requisiti non funzionali:

- Le funzionalità deve essere svolte il più velocemente possibile;
- L'aggiunta o l'eliminazione di un ristorante non deve influire negativamente sulle funzionalità dell'intero sistema;

Requisiti di integrazione:

- Servizio Apache;
- Servizio MySQL;
- Java Runtime Environment(JRE): software che serve ad eseguire o sviluppare applicazioni Java.

5.Implementazione:

Linguaggi di programmazione utilizzati:

- SQL;
- PHP;
- Java;

Framework e librerie utilizzati:

- JAXB-RJ: framework utilizzato per la deserializzazione dei dati XML in oggetti Java;
- Java Swing: framework utilizzato per realizzazione dell'interfaccia grafica in Java;
- Libreria standard di Java.

Descrizione delle principali fasi di sviluppo:

Inizialmente, lo sviluppo è iniziato partendo dalla creazione del database *dbRistoranti*, dove sono state aggiunte tutte le tabelle necessarie per il sistema.

Successivamente, si è passati alla prima fase della creazione del Web-service, dove è stata realizzata la pagina PHP *stampaXML.php* per la stampa in XML dei dati del database.

Contemporaneamente, sono state anche realizzate le interfacce grafiche dell'applicazione, e il file *XSDRistoranti.xsd* per fare la generazione automatica della classe Java

ProvinciaTreviso.java, tramite l'applicazione *xjc.bat*.

Dopo queste operazioni, si è passati alla parte di back-end dell'applicazione, dove sono stati implementati in modo consecutivo le seguenti funzionalità: visualizzazione totale dei ristoranti, inserimento di un nuovo ristorante, eliminazione di un ristorante e ricerca dei ristoranti per nome o per tipologia. Durante la fase d'inserimento e di eliminazione, è stata effettuata in contemporanea anche la seconda fase di realizzazione del Web-service, in cui è stata fatta la pagina *inser_elim_ristoranti.php*.

Struttura del codice sorgente:

Dal punto di vista applicativo vi è la seguente struttura:

- Web-service: *stampaXML.php* e *inser_elim_ristoranti.php*
- Applicativo Java:
 - package model:
 - ProvinciaTreviso.java;
 - LetturaJAXB.java;
 - InviaRichiesteHTTP.java.
 - package view:
 - InterfacciaPrinc.java;
 - Inserisci.java;
 - Elimina.java.
 - package main:
 - Main.java
 - package control:
 - Controller.java.

Descrizione delle scelte architettureali o di design

Il Web-service è stato realizzato per far sì che facesse da intermediario tra il database e l'applicazione. Infatti, l'applicazione Java non effettua nessuna connessione diretta con il database, ma accede a quest'ultima tramite le pagine php.

Per il design grafico dell'applicazione, i componenti sono stati inseriti nel modo più ordinato possibile. Tuttavia, non sono state realizzate grafiche particolari.

6.Test e validazione:

Strategia di test adottata

I test sono stati attuati ogni qual volta si finiva di programmare la struttura principale di una determinata funzionalità. Dopo di che, venivano effettuati dei miglioramenti, e anche questi ultimi venivano testati per verificare la loro correttezza.

Test fatti:

- Test di inserimento dei comuni, servizi e ristoranti nel database:
Le prove sono state fatte dalla shell di xampp e sono state fatte per verificare l'inserimento dei dati nel database. Dopo aver fatto alcune modifiche per risolvere alcune problematiche dovute alle definizioni degli attributi delle tabelle, si è riusciti a inserire i dati nella giusta maniera.
- Test di verifica della stampa dei dati in XML:
Sono state effettuate delle prove per verificare la correttezza della stampa dei dati del database in XML. Dopo alcuni tentativi, si è riusciti a far funzionare la stampa nella maniera corretta, poiché inizialmente i dati in XML non venivano stampati sulla base della struttura definita nel file XSD.
- Test di visualizzazione dei ristoranti nell'applicazione:
Esse sono state fatte per vedere se l'applicazione leggeva i dati XML nella maniera corretta. In questi test, sono stati riscontrati inizialmente delle problematiche, poiché l'applicazione non leggeva i dati correttamente. Questo era dovuto al fatto che i nomi dei tag XML non corrispondevano ai riferimenti XML inseriti nella classe *ProvinciaTreviso.java*.
- Test di inserimento dei ristoranti:
I test sono stati fatti per verificare la correttezza della logica principale della funzione. Dopo averla verificata correttamente con successo, sono stati inseriti i dovuti controlli di input dell'interfaccia utente, e anche questi sono stati verificati. Tralasciando qualche errore di scrittura sintattica di codice, non sono state riscontrate grosse problematiche.
- Test di cancellazione dei ristoranti:
E' stata verificata la correttezza del processo di cancellazione e il controllo dell'input. Tralasciando qualche errore di scrittura sintattica di codice, non sono state riscontrate grosse problematiche.
- Test ricerca per nome o per tipologia:
I test sono stati effettuati per verificarne la correttezza. I risultati sono stati giusti fin da subito, anche dopo l'aggiunta dei controlli di input.

7.Deployment e gestione del progetto:

Descrizione del processo di deployment del progetto

Quando si apre l'applicazione, la prima interfaccia ad apparire è quella principale, dove vi è possibile visualizzare tutti ristoranti, tramite il pulsante "VISUALIZZA", oppure fare la ricerca dei ristoranti per nome o per tipologia. Dall'area di testo non editabile, si possono vedere i dovuti ristoranti.

Visualizza

Buttons: VISUALIZZA, INSERISCI, ELIMINA

Search filters:

- Text input field:
- Button: Cerca nome
- Dropdown menu: Caffè
- Button: Cerca tipologia

Restaurant details (Ristorante San Michele):

- Codice ristorante: 1
- Sito web/social: www.sanmichele.it
- Tipologia: pizzeria
- Locazione: Via Galilei 14/H
- Recensione media: 4.5
- Telefono: 2147483647
- Descrizione: non disponibile
- Comune: Zero Branco (CAP=31059)
- Servizi:
 - 1)Asporto=Si
 - 2)Consumazione sul posto=Si
 - 3)Servizio a domicilio=No
 - 4)Servizio pranzo=Si
 - 5)Servizio cena=Si
 - 6)WiFi=No
 - 7)toilette=Si

Restaurant details (Ristorante ciao):

- Codice ristorante: 5
- Sito web/social: www.ciao.it
- Tipologia: fast food
- Locazione: Via Piero 54/E
- Recensione media: 2.0
- Telefono: 0867316507

Se si clicca sul pulsante "INSERISCI", si accede all'interfaccia di inserimento. Da cui si inserisce il nuovo ristorante, attraverso le caselle di testo, i Radiobutton e il pulsante "INSERISCI NEL DATABASE".

ATTENZIONE! Quando si inserisce il ristorante, bisogna rispettare i seguenti punti:

- tutte le caselle devono obbligatoriamente essere piene, eccetto quella del sito e della descrizione;
- la casella telefono deve contenere 10 cifre tutte attaccate;
- la casella CAP deve contenere 5 cifre e devono corrispondere ai seguenti comuni: Villorba, Preganziol, Vittorio Veneto, Casale sul Sile, Quinto di Treviso e Zero Branco.
- la casella descrizione non può superare i 255 caratteri.
- la casella valutazione deve avere un valore compreso tra 0 e 5 (0 escluso).

Con il pulsante "ANNULLA", si ritorna all'interfaccia principale.

Dall'interfaccia principale, se si clicca il pulsante "ELIMINA", si accede all'interfaccia per l'eliminazione del ristorante, dove si può effettuare l'eliminazione attraverso il codice del ristorante scritto e il click sul pulsante "CANCELLA DAL DATABASE".

Il pulsante "ANNULLA" permette il ritorno all'interfaccia principale.

Pianificazione e gestione del progetto

Lo studente Matteo Vignola ha realizzato i seguenti punti:

- Creazione di tutte le pagine del Web-service;
- Realizzazione della struttura MVC dell'applicazione Java;
- Creazione della funzionalità visualizza e ricerca ristoranti nell'applicazione ;
- Aggiunta della funzionalità di inserimento del nuovo ristorante;
- Creazione delle classi *LetturaJAXB.java* e *InvioRichiesteHTTP.java*

Lo studente Tommaso Guzzo, invece, ha realizzato i seguenti:

- Creazione del database;
- Creazione del file XSD;
- Generazione della classe *ProvinciaTreviso.java*;
- Aggiunta della funzionalità di cancellazione di un ristorante nell'applicazione;
- Creazione delle interfacce grafiche;

8.Conclusioni:

Riepilogo dei risultati raggiunti:

Il progetto ha raggiunto gli obiettivi prefissati, fornendo un sistema completo e funzionale per la gestione delle informazioni sui ristoranti della provincia.

- Abbiamo sviluppato un'applicazione Java che consente agli utenti di visualizzare, aggiungere e eliminare ristoranti nella provincia di Treviso
- L'architettura del sistema è stata progettata in modo efficiente, utilizzando un database relazionale per la memorizzazione dei dati, pagine web PHP per gestione dell'XML e l'applicazione Java per la manipolazione e la visualizzazione dei dati.
- Integrato correttamente l'utilizzo di Apache per la gestione di Mysql
- Sono stati utilizzati correttamente linguaggi di programmazione come SQL, PHP e Java, insieme a framework e librerie come JAXB-Ri e Java Swing

Possibili sviluppi futuri o miglioramenti:

Alcuni miglioramenti possibili possono essere:

- l'implementazione della ricerca dei ristoranti in base ad altri parametri
- grafica migliorata
- più opzioni dei servizi offerti dai ristoranti
- ampliare il database ad altre province e/o stati
- risolvere problematica del CAP associato a più comuni: il database ha come pk il CAP. Di conseguenza, se si vogliono aggiungere comuni con lo stesso CAP, non è possibile per incongruenza.

9.Appendici:

Glossario tecnico:

- JDBC (Java Database Connectivity): API per la connessione a database da applicazioni Java.
- Framework: Struttura di supporto per lo sviluppo di software.
- XML (eXtensible Markup Language): Linguaggio di markup per la rappresentazione strutturata dei dati.
- XSD (XML Schema Definition): Definizione dello schema XML che specifica la struttura e i vincoli dei documenti XML.
- Java Swing: Libreria grafica per la creazione di interfacce utente in Java.
- JAXB (Java Architecture for XML Binding): Framework per la conversione di documenti XML in oggetti Java e viceversa.
- Apache: Software open-source per il server web.

- MySQL: Sistema di gestione di database relazionali open-source.
- JRE (Java Runtime Environment): Ambiente di esecuzione necessario per l'avvio di applicazioni Java.
- SQL (Structured Query Language): Linguaggio di interrogazione per la gestione dei database relazionali.
- URI (Uniform Resource Identifier): Identificatore univoco per risorse su Internet.
- DOM (Document Object Model): Modello standard per la rappresentazione e la manipolazione di documenti XML e HTML come oggetti.
- JSON (JavaScript Object Notation): Formato di scambio dati leggero basato su testo.
- Back-end: Parte dell'applicazione responsabile dell'elaborazione dei dati e della logica del programma.
- Shell: interfaccia utente a righe di comando.
- MVC (Model-View-Controller): pattern architetturale utilizzato per lo sviluppo di un software

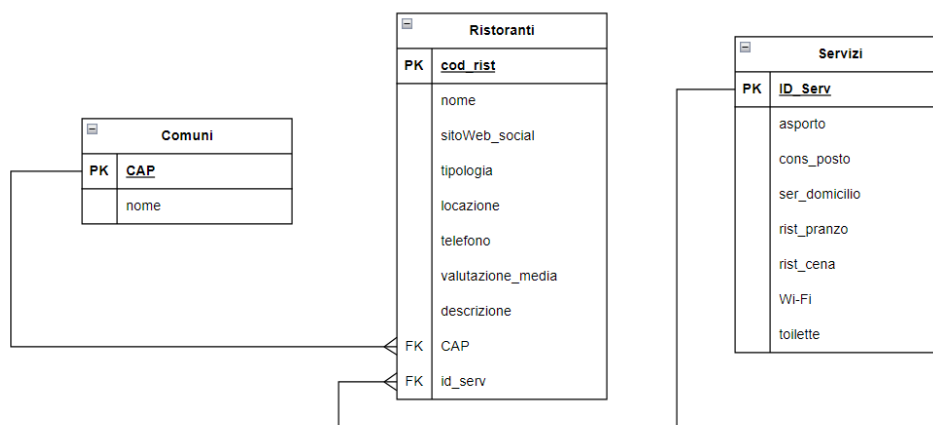
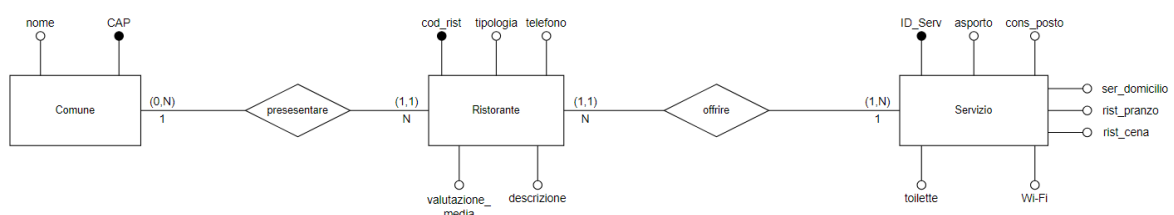
Riferimenti o link utili:

- Documentazione ufficiale di Java: <https://docs.oracle.com/en/java/>
- Documentazione PHP: <https://www.php.net/docs.php>
- Documentazione MySQL: <https://dev.mysql.com/doc/>
- Apache HTTP Server: <https://httpd.apache.org/docs/>
- JAXB - Java Architecture for XML Binding: <https://jaxb.java.net/>
- Swing - Java Swing Documentation: <https://docs.oracle.com/javase/tutorial/uiswing/>
- JDBC - Java Database Connectivity: <https://docs.oracle.com/javase/tutorial/jdbc/>

Altri dettagli e informazioni aggiuntive:

E' stata utilizzata la chatbot ChatGPT per risolvere il problema riguardante l'invio di informazioni da php a java.

Schema ER e Logico:



Comuni(CAP,nome)

Ristoranti(cod_rist, nome, sitoWeb_social, tipologia, locazione, telefono, valutazione_media, descrizione, cap, id_serv)

Servizi(ID-Serv, asporto, cons_posto, ser_domicilio, rist_pranzo, rist_cena, Wi-Fi-toilette)

Create Table:

```
CREATE TABLE `comuni` (  
  `CAP` int(5) NOT NULL,  
  `nome` varchar(255) NOT NULL,  
  PRIMARY KEY (`CAP`)  
)
```

```
CREATE TABLE `ristoranti` (  
  `cod_rist` int(10) unsigned zerofill NOT NULL AUTO_INCREMENT,  
  `nome` varchar(255) NOT NULL,  
  `sitoWeb_social` varchar(255) DEFAULT NULL,  
  `tipologia` enum('misto','caffè','cinese','fast food','indiana','giapponese','mexicana','italiana settentrionale','pizzeria','pesce','italiana meridionale','vegetariana','enoteca') DEFAULT 'misto',  
  `locazione` varchar(255) NOT NULL,  
  `recensione_media` decimal(2,1) NOT NULL,  
  `telefono` bigint(12) DEFAULT NULL,  
  `descrizione` varchar(255) DEFAULT NULL,  
  `cap` int(5) NOT NULL,  
  `id_serv` int(10) unsigned zerofill NOT NULL,  
  PRIMARY KEY (`cod_rist`),  
  KEY `cap` (`cap`),  
  KEY `id_serv` (`id_serv`),  
  CONSTRAINT `rist_serv` FOREIGN KEY (`id_serv`) REFERENCES `servizi` (`ID_Serv`) ON UPDATE CASCADE,  
  CONSTRAINT `ristoranti_ibfk_1` FOREIGN KEY (`cap`) REFERENCES `comuni` (`CAP`)  
)
```

```
CREATE TABLE `servizi` (  
  `ID_Serv` int(10) unsigned zerofill NOT NULL AUTO_INCREMENT,  
  `asporto` tinyint(1) NOT NULL,  
  `cons_posto` tinyint(1) NOT NULL,  
  `ser_domicilio` tinyint(1) NOT NULL,  
  `rist_pranzo` tinyint(1) NOT NULL,  
  `rist_cena` tinyint(1) NOT NULL,  
  `WiFi` tinyint(1) NOT NULL,  
  `toilette` tinyint(1) NOT NULL,  
  PRIMARY KEY (`ID_Serv`)  
)
```