

## Documentazione di CupidoServlet

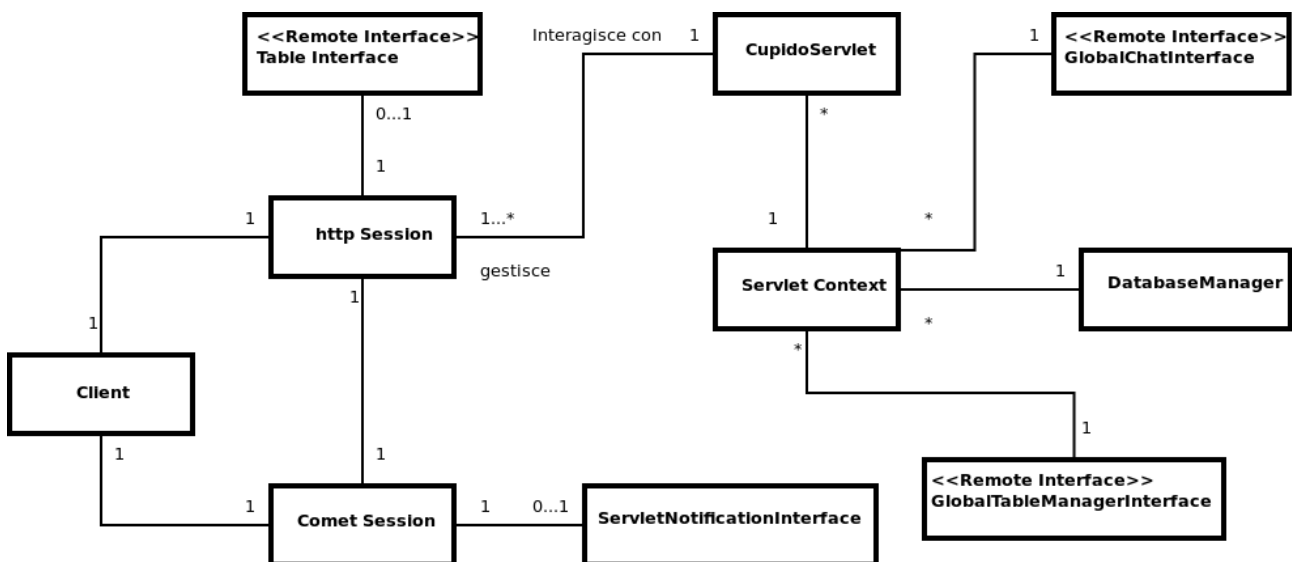
Questo documento spiega le principali caratteristiche sull'implementazione e funzionamento del componente Servlet del software cupido. Si presuppone che il lettore conosca l'architettura ed il design dell'applicazione, descritti nell'allegato Architettura\_e\_design.pdf .

Il presente elaborato è stato prodotto da Lorenzo Belli durante la prima iterazione di sviluppo del componente Servlet.

Il compito principale del componente CupidoServlet è gestire ed inoltrare correttamente le richieste del client ai componenti che le soddisferanno. Oltre a ciò deve predisporre i meccanismi per ricevere notifiche dal componente Table ed inoltrarle al client; queste notifiche riguardano i cambiamenti di stato delle partite e della chat locale

La servlet deve inoltre effettuare le operazioni di Registrazione e Login degli utenti.

È inoltre richiesto che la servlet non inoltri richieste di utenti non positivamente registrati e loggati.



Il Web Server utilizzato è apache/tomcat 6.0 [<http://tomcat.apache.org/>]. La servlet (CupidoServlet) gestisce un qualsiasi numero di Client e dunque di HttpSession. Ogni client instaura una singola HttpSession ed una CometSession con il web-server.

Ad ogni HttpSession è associata l'interfaccia remota TableInterface del tavolo in cui si trova l'utente. Similmente, ad ogni CometSession è associata una ServletNotificationInterface con cui il client riceve notifiche dal componente Table tramite java-rmi. Queste notifiche vengono poi inoltrate tramite al client tramite la CometSession.

Tutte le servlet di un server appartengono al medesimo ServletContext. Il ServletContext memorizza le interfacce remote GlobalTableManagerInterface e GlobalChatInterface che sono implementate da altri componenti del software cupido. ServletContext inoltre memorizza l'oggetto DatabaseManager che permette di accedere al database degli utenti.

CupidoServlet riceve invocazioni di metodi da Client, attraverso l'interfaccia CupidoInterface (non mostrata nel diagramma). La Servlet inoltra queste richieste ai componenti TableInterface, GlobalChatInterface, GlobalTableManagerInterface e DatabaseManager.

## ITERAZIONI DI SVILUPPO

1. Nella prima iterazione è stata scelta la struttura delle classi e degli oggetti definiti dalla servlet. Sono poi state create le classi che implementano le opportune interfacce, lasciando i metodi non implementati.
2. Nella seconda iterazione sono stati implementati tutti i metodi relativi all'instaurazione ed alla chiusura delle connessioni, al login, ed alla registrazione. In particolare è stata implementata la classe *CupidoSessionListener* che attende l'evento di chiusura della connessione http. Questa fase è stata testata creando un apposito client che simulasse l'instaurazione delle connessioni.
3. Nella terza iterazione sono stati implementati tutti i metodi riguardanti la logica di gioco, la chat globale, e la comunicazione con il database. Il testing è stato effettuato utilizzando l'applicativo client già sviluppato solo dopo che il backend era ad uno stadio di sviluppo già avanzato.
4. Nella quarta iterazione è stata implementata *ServletNotificationInterface* che gestisce la ricezione e l'invio delle notifiche. Il testing è stato effettuato utilizzando il client ed il backend già disponibili.

Durante tutte le iterazioni è stato necessario porre modifiche alle interfacce definite nella fase di Design; tutte queste modifiche sono state discusse ed approvate da tutto il gruppo.