UNIVERSITÀ
DI PARMA

Facoltà di Ingegneria - Ingegneria informatica, elettronica e delle telecomunicazioni.

Tecnologie internet 2022-2023

Indovina chi? P2P

Relazione progetto Tecnologie internet

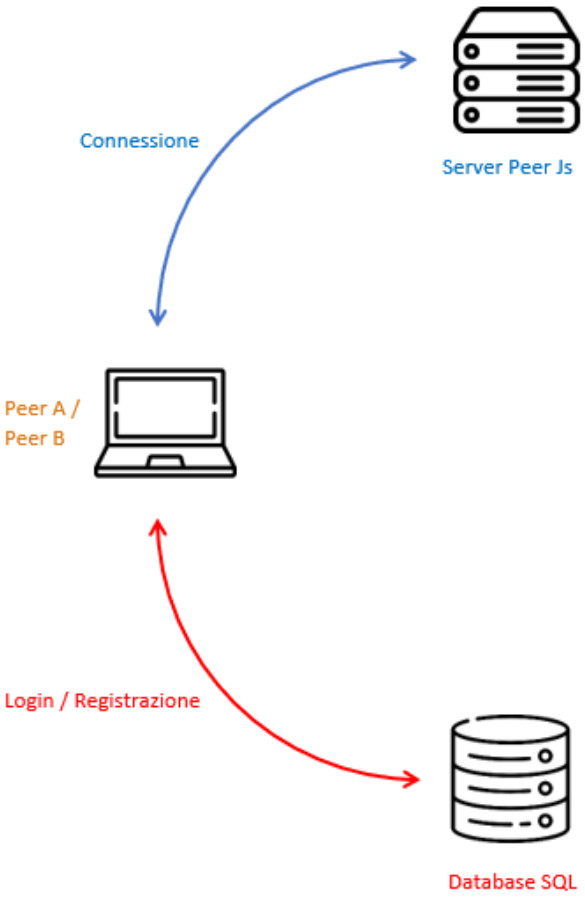
Alessandro Zoppi 318814 – Federico Zilioli 318817

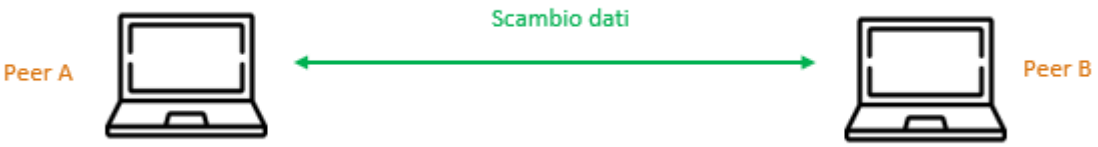
Introduzione

L'obiettivo di questo progetto è quello di realizzare un'applicazione basata su un sistema peer-to-peer utilizzando un framework JavaScript (React) e un framework per la realizzazione di applicazioni web (Node.js) che permetta a due utenti di giocare al gioco Indovina chi?

Per poter iniziare a giocare è necessario che entrambi gli utenti siano registrati e abbiano eseguito l'accesso con le proprie credenziali (nome utente e password). Dopo l'accesso un utente potrà invitare altri giocatori registrati nel sistema (indicandone il nome utente) e attendere che tali inviti vengano accettati o rifiutati. Una volta accettato l'invito, i due utenti potranno giocare insieme.

Struttura del progetto





Questa è una visione stilizzata del sistema, abbiamo i due peer che inizialmente eseguono la registrazione o il log-in mediante credenziali memorizzate nel database SQL. Eseguito l'accesso i peer comunicano con il Server PeerJs per stabilire la connessione, fatto ciò, potranno iniziare a comunicare direttamente tra di loro. Terminata la partita i due peer eseguiranno un aggiornamento del database sul risultato delle proprie partite.

Database

Il database SQL del sistema è composto da un'unica tabella contenente Username, Password, Partite giocate, Partite vinte, Status utili per creare la classifica dei tre migliori giocatori mentre Status viene usato per la gestione delle richieste di sfida. La password per maggiore sicurezza viene crittografata immediatamente mediante la funzione crittografica SHA-3 512 dopo l'inserimento nell'applicazione.

Comunicazione tra peer

Per la comunicazione sono stati utilizzati dei messaggi particolari composti da un carattere speciale seguito dai dati, in questo modo alla ricezione di un messaggio viene eseguito uno slice del primo carattere e in base a questo i dati vengono gestiti. I caratteri utilizzati per riconoscere il tipo di messaggio sono:

- **?**: utilizzato per riconoscere le domande scambiate tra i due giocatori
- **#**: utilizzato per riconoscere le risposte alle domande dei giocatori
- **!**: utilizzato per riconoscere quando un giocatore ha inviato il nome del personaggio che pensa abbia selezionato il suo avversario.

Dopo la ricezione del messaggio per selezionare il personaggio dell'avversario viene eseguito un controllo se effettivamente corrisponde al personaggio scelto. Se è così, chi riceve la soluzione invia all'altro peer il messaggio "**Hai vinto**" altrimenti gli invia il messaggio "**Hai perso**". Oltre a questi messaggi sono ne sono utilizzati anche altri come "**ACCETTATO**" , "**RIFIUTATO**" , "**SELEZIONATO**" utili per la corretta gestione delle richieste.

Comunicazione Server SQL

Per la gestione delle richieste al server sono state utilizzati tipi di richieste come:

- /peer/:Username (richiesta GET): utilizzato per ottenere le credenziali da verificare dell'utente selezionato.
- /newPeer (richiesta POST): utilizzato per la creazione di un nuovo utente con le relative credenziali.

- /points/:Username (richiesta GET): utilizzato per ottenere i valori delle partite giocate e delle partite vinte dell'utente selezionato.
- /updatePeer/:Username (richiesta POST): utilizzato per aggiornare il campo delle partite giocate e partite vinte dell'utente selezionato.
- /peersPoints (richiesta GET): utilizzato per ottenere i punti dei primi tre giocatori ordinati per maggior numero di partite vinte.
- /updateStatusPeer (richiesta POST): utilizzato per aggiornare lo stato dell'utente selezionato.
- /playersStatus (richiesta GET): utilizzato per ottenere lo stato di tutti gli utenti.
- /playerStatus/:Username (richiesta GET): utilizzato per ottenere lo stato dell'utente selezionato.

Istallazione

Per poter utilizzare l'applicazione è necessario installare Node.js e npm. Per installare Node.js e npm è possibile seguire la guida presente al seguente link: https://nodejs.org/it/download/package-manager/, una volta installati Node.js e npm è possibile installare le dipendenze necessarie per l'esecuzione dell'applicazione tramite il comando npm install. Per eseguire l'applicazione è necessario prima modificare il parametro IP che varia in base alla rete sotto cui si è, e successivamente eseguire il comando npm run start. L'applicazione sarà disponibile all'indirizzo http://IP:3000/. È necessario inoltre lanciare il server PeerJs tramite il comando node server.js e il comando node Database.js per avviare il database SQL. Terminate le operazioni di installazione e avvio dell'applicazione, è possibile accedere all'applicazione tramite il browser all'indirizzo http://IP:3000/.