**Prefazione**

Per un'astrazione sufficientemente adeguata è necessario introdurre varie strutture dati.

**Dati da ottenere**

* Record di punteggio raggiunto per sessione e per partita;
* Numero di sessioni giocate;
* Media più alta di punti per sessione e per partita;
* Maggior numero di parole duplicate

è necessario memorizzare tutte le parole citate nella sessione di gioco

* Maggior numero di parole non valide
* Le parole più gettonate

Ovvero le parole che solitamente vengono rilevate dai giocatori

* Il numero medio di turno per partita

valore che andrà a variare in base ai turni di gioco

* L’occorrenza media delle letture uscite nelle griglie generare per la partita.
* L’occorrenza delle parole cui è stato richiesto il significato
* L’identificativo delle partite cui è stato richiesto di dare la definizione di una parola

**Servizi**

**GetSessioniGiocate( id\_partita )**

Esegue una query al DB e restituisce il numero (intero) di sessioni giocate in una specifica partita.

**getNicknameMaxPuntiPerSessione()**

Ritorna il nickname del giocatore che ha ottenuto il maggior numero per sessione.

**getNicknameMaxPuntiPerPartita()**

Ritorna il nickname del giocatore che ha ottenuto il maggior numero di punti per partita.

**getNicknameMaxSessioniGiocate()**

Ritorna il nickname del giocatore che ha giocato il maggior numero di sessioni.

**getNicknameMaxMediaPerPartita()**

Ritorna il nickname del giocatore che ha ottenuto in media il maggior numero di punti per partita.

**getNicknameMaxMediaPerSessione()**

Ritorna il nickname del giocatore che ha ottenuto in media il maggior numero di punti per sessione.

**getNicknameMaxParoleDuplicate()**

Ritorna il nickname del giocatore che ha giocato il numero più alto di parole duplicate.

**getNicknameMaxParoleNonValide()**

Ritorna il nickname del giocatore che ha giocato il numero più alto di parole non valide.

**getParolePiuComuni()**

Restituisce un elenco delle parole più comunemente trovate dai giocatori.

**getParolePiuCospicue()**

Restituisce un elenco delle parole che hanno causato il più grande incremento di punti.

Nell’elenco non vengono restituite le parole annullate.

(da usare il LENGTH in SQL per ottenere la lunghezza della parola)

**getNumMedioTurniPerPartita( num\_partecipanti )**

Ottiene un valore numerico reale (con la virgola) che rappresenta il numero medio di turni in base al numero di partecipanti.

Verrà poi arrotondato in base alle esigenze.

**getMaxTurniPerPartita( num\_partecipanti )**

Il numero massimo di turni che una partita ha svolto.

**getMinTurniPerPartita( num\_partecipanti )**

Il numero minimo di turni che una partita ha svolto.

**getOccorrenzaMediaLettereNellaGriglia()**

Ottiene un elenco delle lettere apparse nella griglia ordinate per occorrenza; adotteranno la seguente struttura:

|  |  |
| --- | --- |
| **Lettera** | **Occorreza** |

**GetParolePiuDiscutibili()**

Ottiene un elenco delle parole cui è stato richiesto di dare la definizione ordinate per occorrenza. Adotteranno la seguente struttura:

|  |  |
| --- | --- |
| **Parola** | **Occorrenza** |

**getPartiteConRichiestaDefinizione( parola )**

Restituisce un elenco di partite in cui è stato richiesto di dare la definizione di una specifica parola.

**Package**

I metodi precedentemente citati saranno presenti all’interno di un package chiamato “statistiche”.

I metodi saranno tutti statici (e forse protetti (?)).

**Strutture**

Le strutture utilizzate per l’interazione con DB sono varie.

Tipi primitivi:

* int
* double
* string (non è primitivo ma è implementato come tale)

Tipi derivati:

* Partita
* OccorrenzaParole:
  + parola: string;
  + numero di occorrenze: int.
* OccorrenzaLettere:
  + lettera: char;
  + numero di occorrenze: int.
* PunteggioPerPartita:
  + numero di punti: int;
  + partita: Partita.
* ParolePiuCospicue:
  + parole: string;
  + punteggio: int.

Per gli elenchi sarà forse necessario l'adozione dei tipi generici (quali Collection)