
DATABASE GAMEINSIGHT

Basi di Dati 2



Author

Ferrara Francesco (0522501959)

UNISA

2024-2025

Indice

1	Introduzione	3
1.1	Contesto applicativo	3
1.2	Obiettivi	3
1.3	Vincoli	4
1.4	Tecnologie utilizzate	4
2	Metodologia	5
3	Descrizione del miniworld	5
3.1	Analisi esplorativa del dataset	5
3.2	Gestione del dataset	6
3.2.1	Definizione delle Collection	6
4	Soluzione proposta	7
4.1	Suddivisione delle componenti	7
4.2	Operazioni CRUD	8
4.2.1	CREATE	8
4.2.2	READ	10
4.2.3	UPDATE	13
4.2.4	DELETE	15
4.3	Analisi dati (JOIN)	17
5	Conclusioni e Sviluppi Futuri	18

1 Introduzione

1.1 Contesto applicativo

Il mondo videoludico ha subito un'enorme crescita negli ultimi anni: questo è dovuto all'avanzamento tecnologico, all'aumento delle piattaforme e del numero di giocatori. Nel corso del tempo si è passati da giochi utilizzabili solo su apparecchi specifici, come i cabinati, i quali, tra l'altro, erano disponibili solo in alcuni luoghi, rendendo quindi più complicata la possibilità di giocare, a console portatili o fisse, fino ad arrivare ai giorni nostri, in cui è possibile giocare a diversi titoli attraverso l'utilizzo di uno smartphone e la disponibilità di una connessione Internet stabile. Tutto ciò è stato ulteriormente amplificato dal crescente successo del **cloud gaming**: con questo termine si intende la possibilità di giocare a un determinato videogioco senza utilizzare le risorse del proprio dispositivo, ma quelle di un server, spesso molto più potente, pagando non il gioco in sé, ma il servizio. L'idea alla base è la stessa adottata dai servizi di streaming come **Netflix** o **Prime Video**: l'utente finale non possiede fisicamente il film, ma accede a una piattaforma che mette a disposizione il suo catalogo. Allo stesso modo, un gamer può utilizzare **GeForce Now**, servizio dell'azienda NVIDIA, per usufruire di un ampio catalogo di giochi attraverso il cloud. In questo contesto si inserisce l'**online gaming**: molti titoli vengono utilizzati da milioni di giocatori in contemporanea e da remoto, rendendo quindi interessante l'analisi del contesto videoludico, sia per proporre nuove esperienze di gioco, sia per comprendere meglio cosa appassiona maggiormente le persone.

1.2 Obiettivi

Il progetto ha come obiettivo principale lo sviluppo di un sistema interattivo, sotto forma di **web application**, per la consultazione e l'analisi dei dati relativi al comportamento dei giocatori online. L'applicazione mira a fornire una piattaforma che consenta di esplorare, filtrare e correlare informazioni sui profili dei giocatori, le loro abitudini di gioco e le statistiche aggregate, in modo da rendere l'esperienza di analisi intuitiva e accessibile anche a utenti non esperti.

1.3 Vincoli

Per riuscire a perseguire gli obiettivi poc'anzi menzionati, verranno considerati i seguenti vincoli progettuali:

1. Utilizzare un dataset pubblico come **miniworld** di riferimento per il progetto;
2. Adottare un approccio **NoSQL** per la progettazione e gestione del database;
3. Includere almeno due entità (**collection**) logicamente collegate attraverso un attributo condiviso (l'equivalente di una **foreign key**);
4. Implementare le operazioni fondamentali **CRUD** (Create, Read, Update, Delete) in modo efficiente e coerente, rispettando le proprietà **BASE** (Basically Available, Soft State, Eventual Consistency);
5. Integrare almeno un'operazione di **JOIN** tra più collection, rispettando almeno due (a scelta) dei tre vincoli del teorema **CAP** (Consistency, Availability, Partition Tolerance);
6. Progettare e sviluppare un'interfaccia utente, anche minimale, che permetta di eseguire in modo interattivo le operazioni CRUD e le interrogazioni basate su JOIN.

1.4 Tecnologie utilizzate

Per la realizzazione della web application verranno impiegate le seguenti tecnologie:

- **MongoDB**: sistema di gestione di database NoSQL basato su documenti, scelto per la sua flessibilità e capacità di gestire dati non strutturati e relazioni dinamiche tra collezioni;
- **React**: libreria open-source basata su JavaScript, utilizzata per la creazione dell'interfaccia utente (*front-end*) dell'applicazione, garantendo modularità e aggiornamento dinamico dei componenti;
- **Node.js**: ambiente di runtime open-source per JavaScript, impiegato per la gestione della logica di *back-end* e la comunicazione tra client e database;
- **Python**: linguaggio di programmazione usato per l'analisi preliminare dei dati, la pulizia del dataset e l'eventuale generazione di statistiche descrittive e visualizzazioni.
- **Vite.js**: strumento di build e sviluppo rapido per progetti frontend basati su JavaScript, usato per ottimizzare e avviare il client React.

L'integrazione tra queste tecnologie consentirà di sviluppare un sistema completo e scalabile, in grado di supportare operazioni di analisi dati e di interazione utente in tempo reale.

2 Metodologia

Lo sviluppo della webapp è iniziato con la **ricerca e selezione del dataset** sulla piattaforma *Kaggle*, scelta per l'ampia disponibilità di dati pubblici di qualità.

Dopo aver individuato un dataset adatto al dominio del progetto, è stata condotta un'**analisi preliminare** per verificare la coerenza e la completezza dei dati, individuando eventuali anomalie o valori mancanti.

Successivamente, si è proceduto al **preprocessing** dei dati all'interno di un notebook dedicato, con l'obiettivo di pulire, trasformare e strutturare le informazioni in modo coerente con le esigenze del modello dati. I file CSV risultanti da questa fase sono stati impiegati per la **creazione delle collection MongoDB**, che costituiscono la base dati del sistema.

Completata la fase di preparazione dei dati, si è avviato lo **sviluppo software** della webapp, seguendo un approccio modulare che separa la logica backend dalla parte frontend. Sono state impiegate le tecnologie descritte nelle sezioni precedenti per realizzare un'applicazione completa, capace di gestire operazioni CRUD, query aggregate e visualizzazioni interattive dei dati.

Infine è stata eseguita una fase di **testing** per verificare che la webapp soddisfasse i vincoli e i requisiti progettuali, oltre che una fase di **debugging** per risolvere eventuali errori riscontrati.

3 Descrizione del miniworld

Il dataset scelto è legato al mondo del gaming ed è disponibile cliccando sul link Kaggle: Dataset Online Gaming Behaviour.

3.1 Analisi esplorativa del dataset

Il dataset possiede 40034 righe, 13 feature, nessun dato mancante e nessun duplicato. Questo consente di fare analisi accurate senza dover effettuare operazioni di preprocessing. Le feature e la loro descrizione viene riportata di seguito:

- **PlayerID**: è un numero intero progressivo che indica l'ID univoco di un gamer. Varia tra 9000 e 49033;
- **Age**: è l'età del giocatore. I valori variano tra 15 e 49 anni;
- **Gender**: indica se la persona è Maschio o Femmina;
- **Location**: indica la regione geografica da cui si gioca. I valori che ha sono USA, Europe, Asia e Other;
- **GameGenre**: indica il genere del videogioco, i cui valori sono Strategy, Sports, Action, RPG e Simulation;
- **PlayTimeHours**: indica il numero totale di ore giocate;
- **InGamePurchases**: è un valore binario che attesta se il giocatore ha fatto acquisti nel gioco;

- **GameDifficulty**: indica la difficoltà del gioco giocato. Ha 3 valori che sono Easy, Medium e Hard;
- **SessionsPerWeek**: indica il numero di sessioni di gioco settimanali effettuate;
- **AvgSessionDurationMinutes**: indica la durata media della sessione di gioco, espressa in minuti;
- **PlayerLevel**: è il livello del gamer;
- **AchievementsUnlocked**: indica il numero di obiettivi sbloccati;
- **EngagementLevel**: indica il livello di coinvolgimento, ossia quanto il giocatore si è sentito coinvolto all'interno del gioco. E' una variabile categorica con valori Low, Medium e High.

3.2 Gestione del dataset

Per poter rispettare vincolo 3 , il dataset verrà diviso in 2 Collection, ognuna delle quali avrà i suoi attributi.

In particolare, il file **online-gaming-insights.csv** sarà diviso in 2 sotto-dataset: **Player.csv** e **Session.csv**.

3.2.1 Definizione delle Collection

Di seguito viene mostrata la suddivisione delle collection, con la descrizione per ogni feature.

Feature	Descrizione
PlayerID	Identificatore di ogni giocatore
Age	Età del giocatore
Gender	Sesso della persona
Location	Area geografica di gioco
PlayerLevel	Livello del giocatore
AchievementsUnlocked	Numero di trofei sbloccati
EngagementLevel	Livello di coinvolgimento del giocatore

Table 1: Collection riferita a Player.csv

Feature	Descrizione
SessionID	Identificativo della sessione di gioco
PlayerID	Identificativo del giocatore
PlayTimeHours	Numero di ore giocate
InGamePurchases	Acquisti in gioco (Si/No)
SessionsPerWeek	Numero di sessioni di gioco a settimana
AvgSessionDurationMinutes	Durata media di una sessione, espressa in minuti
GameGenre	Genere del gioco
GameDifficulty	Difficoltà del gioco

Table 2: Collection riferita a Session.csv

La feature **SessionID** è artificiale, ossia non è presente nel dataset originale, ma creata ad hoc in modo da poter avere due collezioni e poter quindi rispettare il vincolo 3. Questa feature viene calcolata tenendo conto del fatto che ogni record del dataset è unico, ossia anche se un giocatore gioca allo stesso gioco, le sessioni saranno diverse, e questo ci permette di mapparle tramite un ID univoco.

4 Soluzione proposta

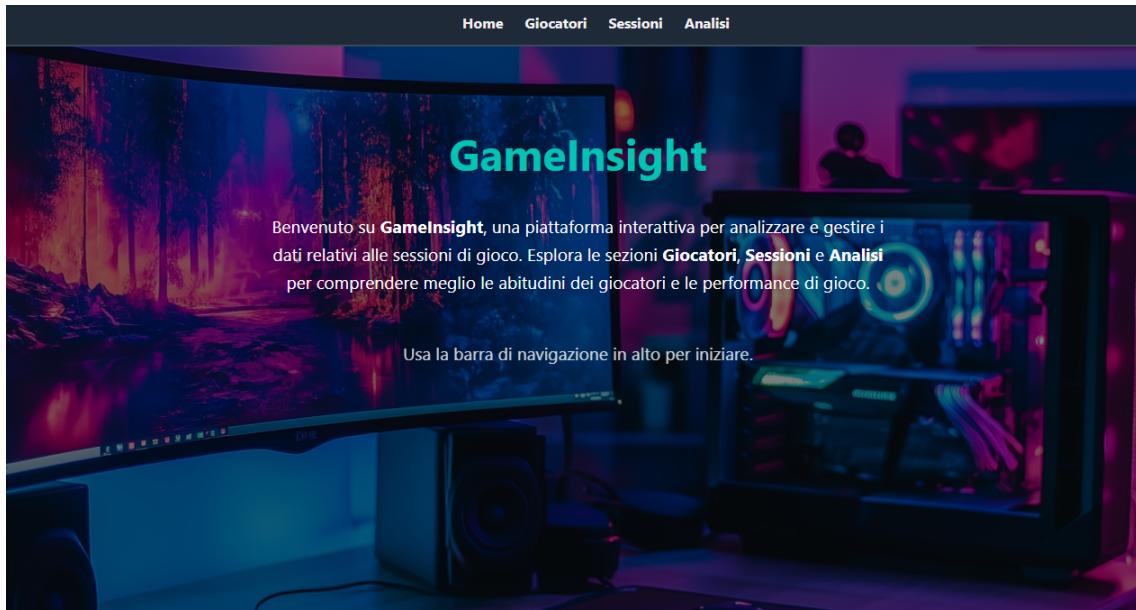


Figure 1: HomePage di GameInsight

In questa sezione viene esplorata la pipeline progettuale, andando ad analizzare nel dettaglio la metodologia e l'implementazione.

4.1 Suddivisione delle componenti

La web application è organizzata secondo una struttura modulare che separa logicamente le diverse componenti del sistema. All'interno della cartella principale sono presenti quattro sottocartelle: client, server, database e data_preprocessing.

- La cartella client contiene gli script riguardanti la parte frontend del progetto, ossia ciò che l'utente finale osserva. Tra questi file figurano la gestione delle diverse pagine (come l'HomePage), i moduli Api per la comunicazione con il backend, le componenti dedicate ai Form e alle liste, oltre ai fogli di stile CSS adottati per la definizione dell'aspetto grafico.

- La cartella server raccoglie la parte backend del progetto, che gestisce la logica applicativa, le operazioni CRUD e le interazioni con il database. Tra i vari file figurano i Controller che implementano la logica delle rotte e coordinano le richieste del client, i Model, che definiscono gli schemi delle collection MongoDB, e le Routes che mappano le richieste HTTP alle relative funzioni.
- La cartella database contiene file di supporto alla configurazione e connessione con MongoDB, come eventuali script di inizializzazione o definizioni di connessione.
- All'interno della cartella data_preprocessing sono stati inseriti il notebook .ipynb, che si occupa dell'analisi iniziale del dataset, e i vari csv utilizzati, comprendendo sia quello iniziale che le due collection derivate.

4.2 Operazioni CRUD

Le operazioni che richiedono la creazione, la lettura, l'aggiornamento e la rimozione di un elemento sono state implementate nelle pagine Giocatori e Sessioni.

4.2.1 CREATE

Di seguito vengono mostrate le schermate di creazione di un Giocatore e di una Sessione.

The screenshot shows a dark-themed web application for managing players. At the top, there's a navigation bar with buttons for 'Crea', 'Visualizza', 'Aggiorna', and 'Rimuovi'. Below the header, a modal window is open with the title 'Crea un nuovo Giocatore'. Inside the modal, there are several input fields: 'Età' (Age), 'Sesso' (Gender), 'Località' (Location), 'Livello del Giocatore' (Player Level), 'Numero di traguardi sbloccati' (Number of unlocked milestones), and 'Coinvolgimento' (Involvement). A large green button at the bottom right of the modal is labeled 'Salva giocatore' (Save player).

Figure 2: Form di creazione di un giocatore

Tramite il form mostrato nella figura di sopra, è possibile creare un nuovo Player. Il PlayerID viene assegnato di default dal sistema, controllando l'ultimo inserito ed aggiungendo un 1.



Figure 3: Giocatore creato con successo

Se la creazione va a buon fine, verrà mostrato a schermo il box riportato sopra. Lo stesso discorso si applica alla Sessione, le cui schermate di creazione e successo della stessa sono mostrate successivamente.

Figure 4: Form di creazione della Sessione

La creazione di una sessione prevede che ci sia prima un giocatore, ed è per questo che nel suo form viene inserito anche il campo PlayerID: se esiste un giocatore, e quindi un PlayerID, si potrà procedere con la creazione di una sessione, altrimenti bisognerà cambiare scheda e andare nella sezione "Crea" del giocatore.

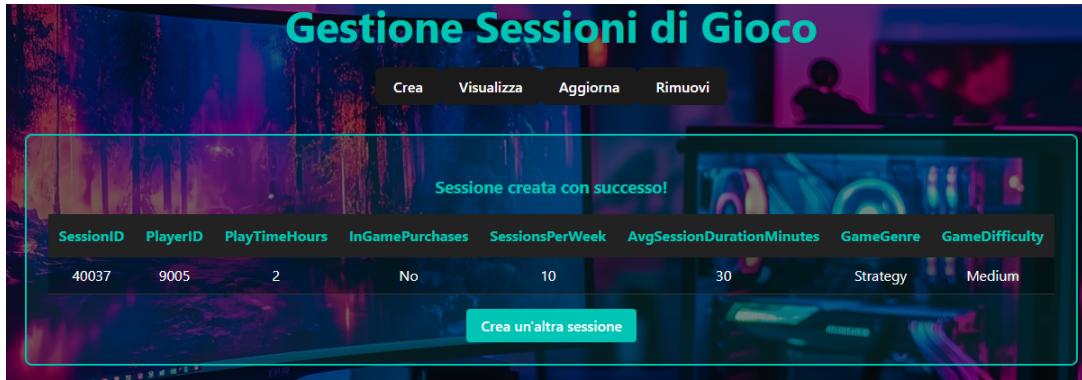


Figure 5: Sessione creata con successo

4.2.2 READ

Nella sezione READ, ossia quella che riguarda la visualizzazione delle informazioni contenute nel db, vengono mostrate le schermate che si hanno una volta che si clicca su "Visualizza".

PlayerID	Age	Gender	Location	Level	Achievements	Engagement
9001	29	Female	USA	11	10	Medium
9002	22	Female	USA	35	41	High
9003	35	Male	USA	57	47	Medium
9004	33	Male	Europe	95	37	Medium
9005	37	Male	Europe	74	22	Low
9006	25	Male	USA	13	2	Low
9007	25	Female	Asia	27	23	Medium
9008	38	Female	Europe	23	41	Medium
9009	38	Female	Other	99	36	High
9010	17	Male	USA	14	12	High

Figure 6: Visualizzazione della lista Giocatori

Inizialmente viene mostrata l'intera lista, divisa per pagine, ognuna delle quali mostra 50 entry. Tale pagina consente anche la ricerca diretta tramite il PlayerID.

Lista Giocatori						
Cerca		Mostra tutti				
PlayerID	Age	Gender	Location	Level	Achievements	Engagement
15000	31	Male	USA	28	36	Low

Figure 7: Ricerca giocatore (Successo)

Lista Giocatori						
Cerca		Mostra tutti				
PlayerID	Age	Gender	Location	Level	Achievements	Engagement
Giocatore con ID "0" non trovato.						
Nessun giocatore trovato.						

Figure 8: Ricerca giocatore (Fallimento)

Se la ricerca ottiene il successo sperato verrà visualizzato il record ricercato, altrimenti la lista da mostrare sarà vuota.

Gestione Sessioni di Gioco											
Lista Sessioni											
SessionID	PlayerID	PlayTimeHours	InGamePurchases	SessionsPerWeek	AvgSessionDurationMinutes	GameGenre	GameDifficulty	Crea	Visualizza	Aggiorna	Rimuovi
1	9000	16.27	No	6	108	Strategy	Medium				
2	9001	5.53	No	5	144	Strategy	Medium				
3	9002	8.22	No	16	142	Sports	Easy				
4	9003	5.27	Si	9	85	Action	Easy				
5	9004	15.53	No	2	131	Action	Medium				
6	9005	20.56	No	2	81	RPG	Easy				
7	9006	9.75	No	1	50	Action	Hard				
8	9007	4.4	No	10	48	RPG	Medium				
9	9008	18.15	No	5	101	Simulation	Easy				

Figure 9: Visualizzazione della lista Sessioni

Lista Sessioni								
SessionID	PlayerID	PlayTimeHours	InGamePurchases	SessionsPerWeek	AvgSessionDurationMinutes	GameGenre	GameDifficulty	
15000	23999	15.46	No	6	15	Strategy	Easy	

Figure 10: Ricerca sessione (Successo)

Lista Sessioni								
SessionID	PlayerID	PlayTimeHours	InGamePurchases	SessionsPerWeek	AvgSessionDurationMinutes	GameGenre	GameDifficulty	
Sessione con ID "0" non trovata.								

Figure 11: Ricerca sessione (Fallimento)

4.2.3 UPDATE

Aggiorna Dati Giocatore

Cerca il giocatore da modificare inserendo il suo PlayerID

40000

Giocatore trovato: 40000
Modifica i campi che desideri aggiornare

Età
27

Sesso
Female

Località
Asia

Livello Giocatore
59

Obiettivi Sbloccati
30

Coinvolgimento
High

Aggiorna Giocatore

Figure 12: Form di aggiornamento del giocatore

L'update di un giocatore o di una sessione è strutturato in maniera simile a quanto succede con una create, nel senso che l'utente si troverà di fronte un form con le varie componenti da modificare. Può decidere di modificare tutto o solo una parte.



Figure 13: Successo dell’aggiornamento del Player

Una volta completato il processo, verrà mostrato il record aggiornato.

The screenshot shows a dark-themed form titled 'Aggiorna Dati Sessione'. It starts with a search bar containing '9005' and a 'Cerca' button. Below the search results, a message says 'Sessione trovata: 9005' and 'Modifica i campi che desideri aggiornare'. The form fields include: 'Player ID (non modificabile)' with value '18004'; 'Ore di gioco' with value '4,5'; 'Acquisti in-game' with dropdown 'Selezione'; 'Sessioni a settimana' with value '5'; 'Durata media (minuti)' with value '148'; 'Genere' with dropdown 'Strategy'; and 'Difficoltà' with dropdown 'Medium'. At the bottom is a large blue button labeled 'Aggiorna Sessione'.

Figure 14: Form di aggiornamento della sessione

Lo stesso discorso vale per la sessione. Nel suo form però viene anche mostrato il PlayerID ad essa associato: questo valore però non sarà modificabile.



Figure 15: Successo dell'aggiornamento della Sessione

4.2.4 DELETE

PlayerID	Age	Gender	Location	Level	Achievements	Engagement	Azioni
9001	29	Female	USA	11	10	Medium	<button>Elimina</button>
9002	22	Female	USA	35	41	High	<button>Elimina</button>
9003	35	Male	USA	57	47	Medium	<button>Elimina</button>
9004	33	Male	Europe	95	37	Medium	<button>Elimina</button>
9005	37	Male	Europe	74	22	Low	<button>Elimina</button>
9006	25	Male	USA	13	2	Low	<button>Elimina</button>

Figure 16: Schermata di rimozione di un Player

All'interno della sezione "Rimuovi" viene mostrata una lista dei vari giocatori presenti e la barra di ricerca, ma a differenza di quella contenuta in "Visualizza" mostra anche un pulsante di rimozione. Se si procederà con la rimozione dell'entità, allora verrà visualizzato il messaggio "Giocatore eliminato con successo".

Elimina una Sessione

Cerca la sessione e clicca sul pulsante "Elimina".

Lista Sessioni

Cerca per SessionID... Cerca

SessionID	PlayerID	PlayTimeHours	InGamePurchases	SessionsPerWeek	AvgSessionDurationMinutes	GameGenre	GameDifficulty	Azioni
1	9000	16.27	No	6	108	Strategy	Medium	<button>Elimina</button>
2	9001	5.53	No	5	144	Strategy	Medium	<button>Elimina</button>
3	9002	8.22	No	16	142	Sports	Easy	<button>Elimina</button>
4	9003	5.27	Si	9	85	Action	Easy	<button>Elimina</button>
5	9004	15.53	No	2	131	Action	Medium	<button>Elimina</button>

Figure 17: Schermata di rimozione di una Sessione

4.3 Analisi dati (JOIN)

All'interno della sezione Analisi della webapp sono stati riportati due grafici che mostrano la situazione del gaming.

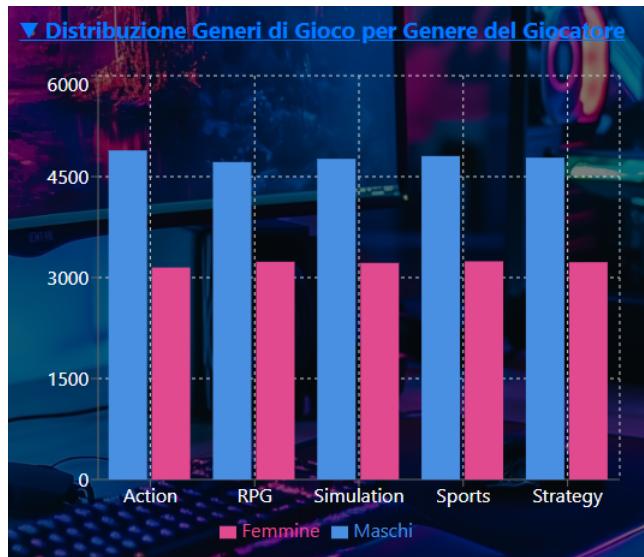


Figure 18: Distribuzione GameGenre per Genere del Player

La figura 9 mostra come sono distribuite le categorie maschio e femmina nel contesto del gaming, facendo un raffronto con il genere di gioco. Si nota una situazione identica per ogni genere di gioco, ossia abbiamo che si ha la medesima distribuzione di maschi e femmine, a prescindere da quella che è la categoria. Questo grafico però non corrisponde ad una situazione reale, ossia una distribuzione così perfetta è frutto di quelle che sono le caratteristiche del dataset preso in esame.

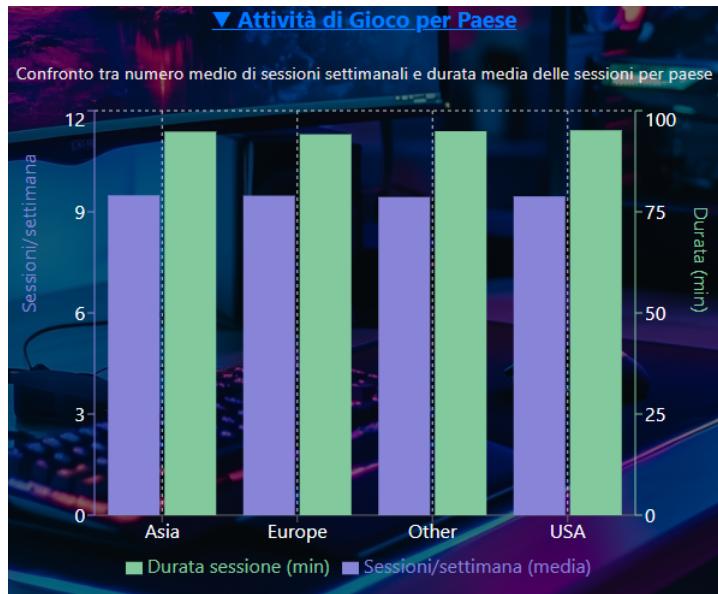


Figure 19: Attività di gioco per Paese

La figura 10 mostra il confronto tra numero medio di sessioni settimanali e durata media delle sessioni per paese. Anche qui la situazione ricalca quanto visto in precedenza, ossia a prescindere dalla zona geografica, la distribuzione della durata della sessione e il numero medio di sessioni settimanali, risultano identici tra i vari paesi. Sebbene questo sia più verosimile rispetto alla situazione precedente, è comunque poco probabile che nella realtà si abbia una distribuzione del genere. La conclusione che se ne trae è che la composizione del dataset tende ad avere un alto bilanciamento di tutte le feature in esso presenti.

5 Conclusioni e Sviluppi Futuri

Giunto alla fine di questo progetto posso ritenermi soddisfatto del lavoro svolto: ho potuto utilizzare nuove tecnologie e cimentarmi per la prima volta nella creazione di una webapp. Nonostante ciò, credo che GameInsight possa essere migliorato con, tra le altre cose, la ricerca di un nuovo dataset che abbia dati reali, il perfezionamento della grafica tramite componenti dinamiche, l'aggiunta di ulteriori grafici analitici e l'inserimento dell'https.