



Laurea in Informatica - Università degli Studi di Salerno  
Corso di Ingegneria del Software - Prof. A. De Lucia



**INFINITY GAMES**

# System Design Document

## Membri del Team

Falciano Ciro	0512103596
Schiroso Alessandro	0512104514
Ruggiero Simone	0512106074
De Lucia Antonino	0512105366

## REVISION HISTORY

DATA	VERSIONE	DESCRIZIONE	AUTORE
30/11/20	1.0	Prima versione del documento.	C. Falciano, S. Ruggiero, A. Schirosa, A. De Lucia
03/11/20	1.1	Aggiunta dei punti 3,4,5	C. Falciano, S. Ruggiero, A. Schirosa, A. De Lucia

## SOMMARIO

1. Introduzione
  - 1.1. Obiettivi del sistema
  - 1.2. Design goals
  - 1.3. Definizioni, Acronimi e Abbreviazioni
  - 1.4. Riferimenti
  - 1.5. Panoramica
2. Architettura del Sistema corrente
3. Architettura del Sistema proposto
  - 3.1. Panoramica
  - 3.2. Decomposizione sottosistemi
  - 3.3. Mapping hardware/software
    - 3.3.1. Deployment
  - 3.4. Gestione dati persistenti
    - 3.4.1. Schema logico
    - 3.4.2. Struttura delle Tabelle
  - 3.5. Controllo degli accessi e sicurezza
  - 3.6. Controllo globale del software
  - 3.7. Condizione limite
4. Servizi dei Sottosistemi
5. Glossario

# 1. INTRODUZIONE

## 1.1. Obiettivi del sistema

L'obiettivo del sistema è quello di consentire agli utenti di poter acquistare videogiochi e visualizzarli in una libreria personale, oltre che a ricercare prodotti nello store e aggiungerli alla lista dei desideri.

L'interfaccia del sistema dovrà essere estremamente intuitiva, in modo da invogliare gli utenti ad effettuare gli acquisti.

## 1.2. Design Goals

ID	DESCRIZIONE	CATEGORIA	ORIGINE
<b>DG_1 Tempo di risposta</b>	Il sistema deve garantire un tempo di risposta inferiore ad un secondo.	Performance	NFR_10
<b>DG_2 Portabilità</b>	Il sistema sarà responsive, in quanto il sito sarà sviluppato in modo da permettere la visualizzazione anche da tablet e smartphone.	Usabilità Interfaccia	NFR_1 NFR_2 NFR_21
<b>DG_3 Usabilità</b>	L'usabilità del sistema sarà data dall'interfaccia intuitiva, contenente pulsanti, titoli e messaggi di errori semplici da comprendere.	Usabilità	NFR_4

ID	DESCRIZIONE	CATEGORIA	ORIGINE
<b>DG_4 Sicurezza</b>	Il sistema consente l'accesso tramite autenticazione ed ogni utente può usufruire esclusivamente delle funzionalità associate alla propria categoria. Inoltre, il sistema deve tenere al sicuro i dati da chi non è autorizzato ad accedervi.	Affidabilità	NFR_5 NFR_8 NFR_9
<b>DG_5 Memoria</b>	Si utilizzerà il database relazionale MySQL per gestire la grande quantità di dati, pertanto il sistema non dovrà subire rallentamenti.	Performance	NFR_11 NFR_13
<b>DG_6 Disponibilità</b>	Il sistema dovrà essere perennemente usufruibile dagli utenti, fatta eccezione per le ore di manutenzione.	Affidabilità Implementazione	NFR_6 NFR_7 NFR_18
<b>DG_7 Robustezza</b>	Eventuali inserimenti non validi o mancanti da parte dell'utente devono essere segnalati tramite messaggi di errore.	Affidabilità	NFR_8
<b>DG_8 Affidabilità</b>	Il sistema garantirà una corretta gestione delle funzionalità.	Affidabilità	NFR_8 NFR_5 NFR_6

ID	DESCRIZIONE	CATEGORIA	ORIGINE
<b>DG_9</b> <b>Adattabilità</b>	Il sistema dovrà adattarsi facilmente alle nuove necessità del cliente.	Interfaccia	NFR_21

### 1.3. Definizione, Acronimi e Abbreviazioni

**RAD:** Requirements Analysis Document.

**SDD:** System Design Document.

**ODD:** Object Design Document.

**NFR:** Requisiti non funzionali.

**UC:** Use case.

**FR:** Requisiti funzionali.

**DG:** Design Goal

**MVC:** Model View Controller

**GUI:** Graphic User Interface

**DBMS:** Database Management System

**JSP:** Java Server Page

**HTTP:** HyperText Transfer Protocol

**Amministratore:** Utente che si occupa di gestire la piattaforma e gli utenti.

**Utente:** Un qualsiasi utilizzatore della piattaforma.

### 1.4. Riferimenti

-Object-Oriented Software Engineering Using UML, Patterns, and Java, Bernd Bruegge & Allen H. Dutoit.

-Steam

-Infinity Games - RAD

## 1.5. Panoramica

Il documento si compone di una prima parte in cui vengono introdotti gli obiettivi di design.

Al secondo punto del documento viene presentata l'architettura del sistema corrente.

Al terzo punto viene presentata l'architettura del sistema proposto e nel dettaglio:

- La decomposizione del sistema in sottosistemi;
- Il mapping Hardware/Software;
- La gestione dei dati persistenti;
- Il controllo degli accessi e della sicurezza;
- Il controllo del flusso globale del sistema;
- Le condizioni limite.

Al quarto punto vengono presentati i servizi di ogni sottosistema.

Al quinto punto viene fornito il glossario dei termini utilizzati nel documento con le relative definizioni.

## **2. ARCHITETTURA DEL SISTEMA CORRENTE**

L'architettura del sistema corrente si rifà a Steam, una delle piattaforme più importanti ed utilizzate nell'ambito della vendita di prodotti di stampo videoludico.

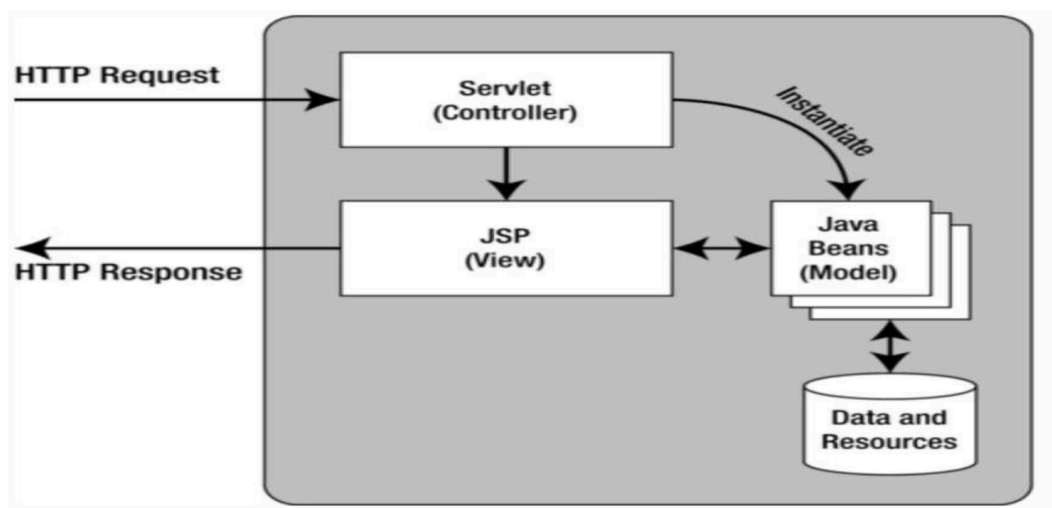
## **3. ARCHITETTURA DEL SISTEMA PROPOSTO**

### **3.1. Panoramica**

L'architettura del sistema proposto si rifà a quella del sistema corrente, offrendo delle funzioni fondamentali:

- Consentire la visualizzazione dei videogiochi più rilevanti sul mercato, facilitando la ricerca del prodotto.
- Rendere possibile acquistare prodotti e visualizzare tutti i prodotti acquistati in una libreria virtuale.
- Consentire la gestione degli ordini effettuati da parte del singolo utente.

Per la progettazione e per lo sviluppo di Infinity Games è stata scelta un'architettura MVC (Model-View-Controller) che separa la logica di business da quella di presentazione.



**Model:** Si occupa di gestire i dati persistenti, contiene metodi di lettura e scrittura sul Database.

**View:** Si occupa di far visualizzare i dati all'utente e si occupa dell'interazione tra quest'ultimo ed il Sistema.

**Controller:** Si occupa della logica di controllo dell'applicazione interagendo con gli altri due componenti.

### 3.2. Decomposizione sottosistemi

Presentiamo una decomposizione del nostro sistema in 3 layer (o livelli: Presentation Layer, Application Logic Layer e Data Access Layer che si occupano di gestire funzionalità ed aspetti differenti del nostro sistema:

**Presentation Layer(View):** Livello che gestisce la comunicazione con le entità esterne al sistema tramite delle componenti che si occupano di rappresentare l'informazione verso i client, consentendo essi di interagire con il sistema.

**Utente non registrato:** GUI che gestisce l'interfaccia degli utenti che non hanno effettuato l'accesso e/o non hanno effettuato la registrazione alla piattaforma, i quali sono limitati alla visualizzazione dei prodotti dello store.

**Utente registrato:** GUI che gestisce l'interfaccia degli utenti che sono registrati e che hanno effettuato l'accesso alla piattaforma, e che sono abilitati ad acquistare prodotti, visualizzare la propria area personale, contattare l'assistenza ed inserire prodotti nella propria lista dei desideri.

**Amministratore:** GUI che gestisce l'interfaccia dedicata ai responsabili della piattaforma, e che sono abilitati all'aggiunta e alla rimozione di prodotti e categorie, promuovere utenti ad amministratori, declassare utenti dallo stato di amministratore, visualizzare gli ordini di ogni singolo utente e rimuovere utenti dalla piattaforma.

**Application Layer(Control):** E' il livello del sistema che si occupa del processamento dei dati necessario per produrre i risultati da inoltrare al Presentation Layer.



**Gestione Utente:** Sottosistema che permette di registrarsi, effettuare il login e il logout, visualizzazione dell'area personale e modifica dei dati relativi a quest'ultima.

**Gestione Amministratore:** Sottosistema che permette di visualizzare gli utenti della piattaforma, promuovere un utente ad amministratore, declassare un amministratore e rimuovere un utente dalla piattaforma.

**Gestione Prodotto:** Sottosistema che permette di aggiungere un prodotto al carrello, aggiungere un prodotto alla lista dei desideri, ricercare un prodotto nello store, aggiungere un prodotto allo store da amministratore, rimuovere un prodotto dallo store da amministratore e modificare un prodotto da amministratore.

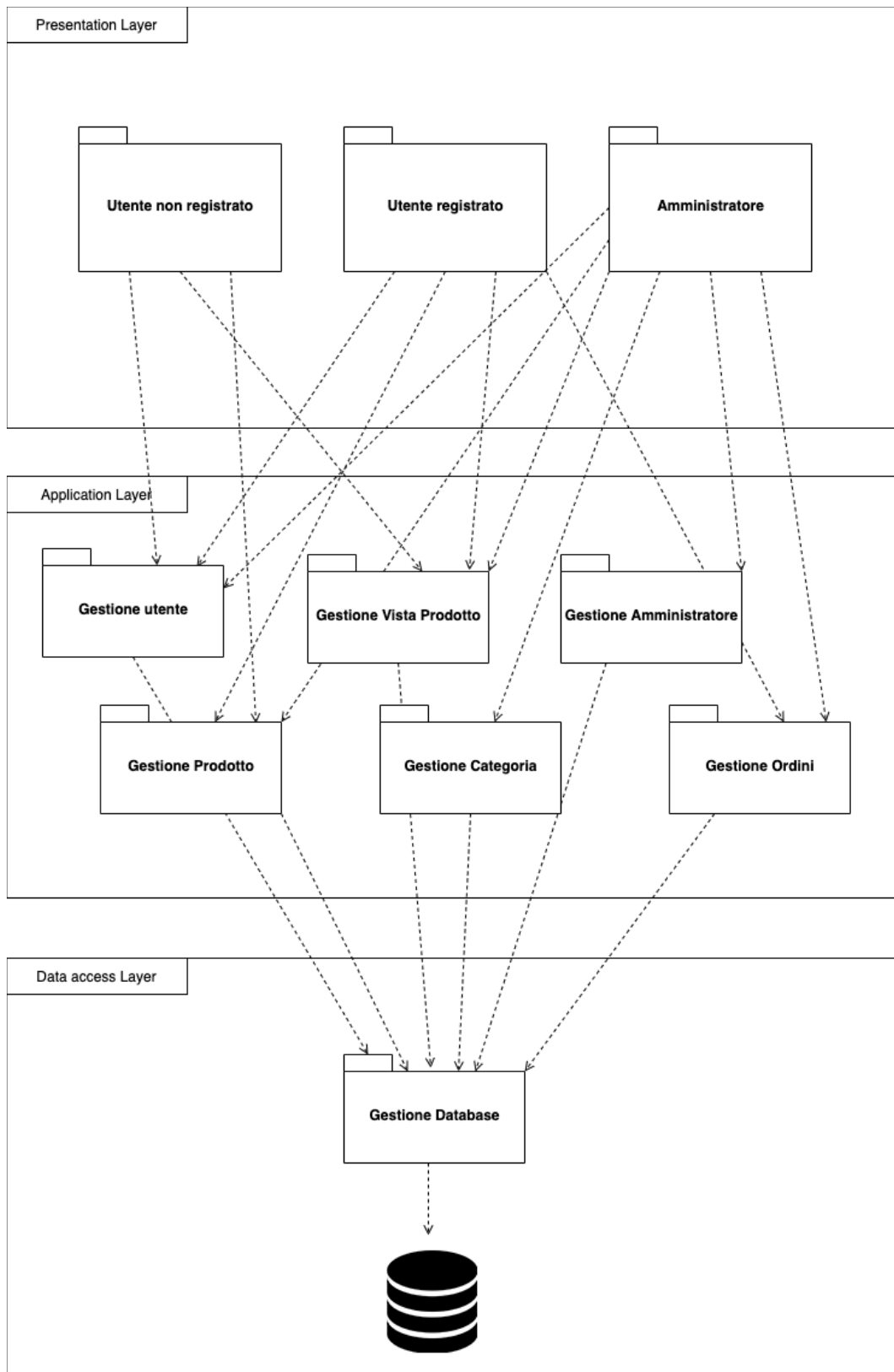
**Gestione Vista Prodotto:** Sottosistema che permette di visualizzare la pagina del prodotto, visualizzare una lista di prodotti filtrati per categoria, popolari, più venduti, per scontati, per il prezzo, visualizzare la libreria personale di videogiochi acquistati e visualizzare la lista dei desideri, visualizzare il carrello.

**Gestione Categoria:** Sottosistema che permette di aggiungere una categoria allo store da amministratore, modificare una categoria da amministratore e rimuovere una categoria dallo store da amministratore.

**Gestione Ordini:** Sottosistema che permette di effettuare un acquisto, visualizzare lo storico degli ordini, visualizzare gli ordini degli utenti da amministratore.

**Data Access Layer(Model) :** Livello che gestisce i dati necessari al funzionamento dell'intero sistema ovvero i dati persistenti.

**Gestione DataBase(DB):** sottosistema che si occupa di immagazzinare e di prelevare i dati persistenti dal nostro database.



### 3.3. Mapping hardware/software

Per il sistema che si vuole realizzare ci interessa un framework o librerie per applicazioni web.

Il framework che andremo ad utilizzare è jQuery, Javascript, AJAX, JSON, JSTL.

jQuery è una libreria JavaScript per accelerare lo sviluppo della business logic delle interfacce web.

Javascript è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client.

AJAX, è una tecnica di sviluppo software per la realizzazione di applicazioni web interattive (Rich Internet Application), basandosi su uno scambio di dati in background fra web browser e server.

JavaScript Object Notation, è un formato adatto all'interscambio di dati fra applicazioni client/server. Viene usato in AJAX come alternativa a XML/XSLT.

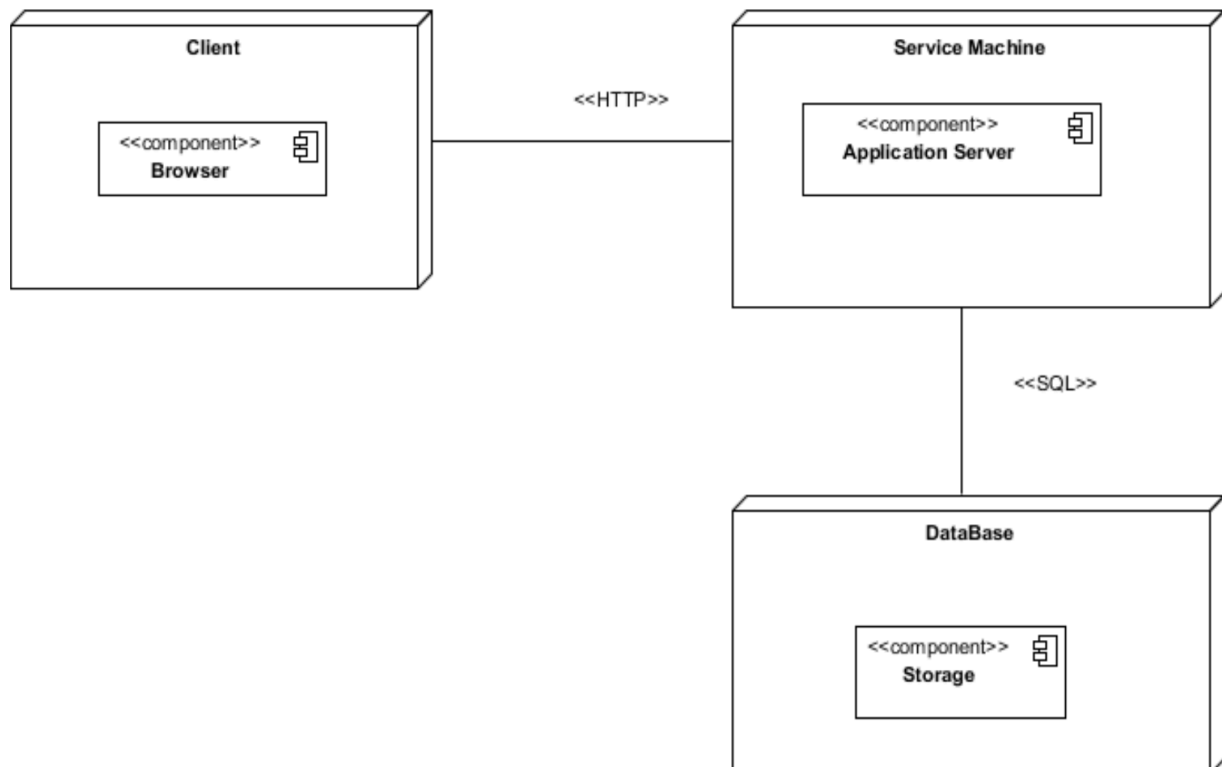
JSTL è un'estensione di JSP ed incorpora un insieme di tag HTML definiti tramite file XML e programmati in linguaggio Java.

Per il sistema, basato su un'architettura MVC, sono state scelti questi tipi di configurazioni:

- per la parte di Interface (View) usiamo le JSP, HTML, CSS, JavaScript.
- per la parte di business logic (Control) usiamo Apache Tomcat e Java.
- per la parte di Data access (Model) usiamo MySQL con JDBC.
- Come Protocollo di comunicazione usiamo HTTP, TCP/IP.

### 3.3.1. Deployment

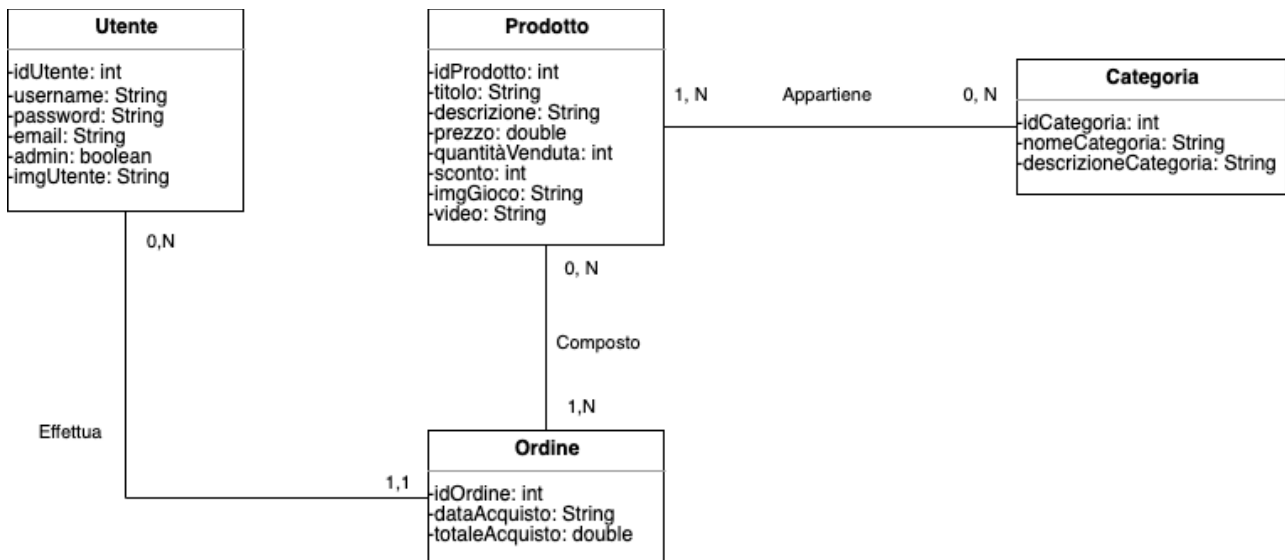
Gli utenti potranno interagire con Infinity Games tramite un WebBrowser collegandosi all'indirizzo del nostro sito. Il Control si occupa di: interpretare gli eventi generati dall'utente, richiedere e prelevare le opportune risorse dalla Base di Dati ed inviare le risposte tramite protocollo HTTP.



## 3.4. Gestione Dati Persistenti

Il sistema che si vuole sviluppare sarà installato su un solo computer e utilizzerà un DBMS MySQL installato sullo stesso. Il sistema sarà diviso in client e server che inizialmente saranno lo stesso pc, ma in caso di futura necessità potranno essere facilmente divisi in quanto i servizi saranno progettati separatamente. Sul server ci sarà un DBMS per la gestione dei dati persistenti di nostro interesse.

### 3.4.1. Schema Logico



### 3.4.2. Struttura delle Tabelle

#### Tabella Utente

Attributo	Tipo	Vincoli
idUtente	int	primary key
username	string	not null unique
password	string	not null
email	string	not null unique
admin	boolean	not null
imgUtente	string	

#### Tabella Ordine

Attributo	Tipo	Vincoli
idOrdine	int	primary key
idUtente	int	foreign key Utente(idUtente)
dataAcquisto	string	not null
totaleAcquisto	double	not null

#### Tabella Prodotto

Attributo	Tipo	Vincoli
idProdotto	int	primary key
titolo	string	not null
descrizione	string	not null
prezzo	double	not null
quantitàVenduta	int	not null
sconto	int	
imgGioco	string	
video	string	

## Tabella Categoria

Attributo	Tipo	Vincoli
idCategoria	int	primary key
nomeCategoria	string	not null
descrizione	string	not null

### 3.5. Controllo degli accessi

Il sistema può essere utilizzato da qualsiasi utente possidente una connessione internet per il compimento di un insieme ristretto di funzionalità, che va ad allargarsi quando l'utente si registra alla piattaforma. La piattaforma predispone dei controlli per l'iscrizione al sistema, dando ad ogni utente la possibilità di registrarsi utilizzando tre credenziali (e-mail, password e username, dove la mail deve necessariamente seguire dei pattern di nomenclatura cosicché il sistema possa verificare l'appartenenza dell'utente all'ateneo). L'accesso alla piattaforma per gli utenti è garantito dall'utilizzo di username e password, e verrà chiesto ad ogni scadenza di sessione.

La sicurezza dei dati è garantita tramite l'utilizzo di metodi POST e le password sono memorizzate nel database utilizzando la crittografia SHA-1.

Tutte le funzionalità del sistema sono descritte nella matrice degli accessi sottostante.

Attore	Gestione Utente	Gestione Amministratore	Gestione Prodotto	Gestione Vista Prodotto	Gestione Categoria	Gestione Ordine
Utente non registrato	<<Registrazione>>		<<Aggiungi prodotto al Carrello>>  <<Rimuovi prodotto dal carrello>>  <<Ricerca Prodotto Store>>	<<Visualizza prodotto nello store>>  <<Visualizza Carrello>>		

Attore	Gestione Utente	Gestione Amministratore	Gestione Prodotto	Gestione Vista Prodotto	Gestione Categoria	Gestione Ordine
Utente registrato	<<Login>> <<Logout>> <<Visualizza Area Personale>> <<Modifica Profilo>>		<<Aggiungi prodotto al Carrello>> <<Rimuovi prodotto dal carrello>> <<Ricerca Prodotto Store>> <<Aggiungi Prodotto alla Lista dei Desideri>> <<Rimuovi dalla Lista dei Desideri>>	<<Visualizza prodotto nello store>> <<Visualizza Carrello>> <<Visualizza Libreria>> <<Visualizza Lista dei Desideri>>		<<Acquisto Prodotto>> <<Visualizza Storico Ordini>>



Attore	Gestione Utente	Gestione Amministratore	Gestione Prodotto	Gestione Vista Prodotto	Gestione Categoria	Gestione Ordine
Amministratore	<<Login>> <<Logout>> <<Visualizza Area Personale>> <<Modifica Profilo>>	<<Visualizza Utenti della Piattaforma>> <<Promozioni e utente ad Amministratore>> <<Declassa utente Amministratore>> <<Rimuovi Utente>>	<<Aggiungi prodotto al Carrello>> <<Rimuovi prodotto dal carrello>> <<Ricerca Prodotto Store>> <<Aggiungi Prodotto alla Lista dei Desideri>> <<Rimuovi dalla Lista dei Desideri>> <<Aggiungi Prodotto allo store>> <<Modifica Prodotto dello store>> <<Rimuovi Prodotto dallo store>>	<<Visualizza prodotto nello store>> <<Visualizza Carrello>> <<Visualizza Libreria>> <<Visualizza Lista dei Desideri>>	<<Aggiungi Categoria allo store>> <<Modifica Categoria dello store>> <<Rimuovi Categoria dallo store>>	<<Acquisto Prodotto>> <<Visualizza Storico Ordini>> <<Visualizza Ordini Utenti da Amministratore>>

## 3.6. Controllo globale del Software

Il controllo del flusso del software è regolato da classi Java che fungono da ricevitori di eventi e che rispondono alle attivazioni di client. Le richieste vengono generate da un client e la classe preposta a gestire quel determinato evento associato alla richiesta, prendendo talvolta degli input si preoccupa di inizializzare le richieste e di inoltrarle alle classi per lo svolgimento dell'operazione. Una volta ottenuto il risultato dell'operazione, la classe gestore si preoccupa di inoltrarlo al client che aveva generato la richiesta.

Il sistema software è gestito con l'uso di Servlet e JSP.

Il server centrale attende le richieste di un client (web browser) e una volta ricevuta una richiesta, la processa e la smista alla Servlet deputata.

## 3.7. Condizioni limite

### Server

**-Fase di Startup:** La fase di avvio del server Apache Tomcat dura circa trenta secondi, dopodiché viene avviato il server dove è contenuto il DBMS MySQL. La connessione al DBMS verrà effettuata tramite il driver JDBC.

Successivamente viene avviata la JVM sul server e vengono letti i dati di configurazione del database. Infine, vengono caricati tutti i dati in memoria volatile e il sistema è pronto all'uso.

**-Fase di Terminazione:** Il server può essere spento in caso di guasto o in caso di necessità di manutenzione. Terminata l'attività di manutenzione o di riparazione del guasto il processo server viene avviato nuovamente.

**-Fallimento:** Il server può fallire in caso di interruzione inaspettata dell'alimentazione e/o di guasti hardware. Il server può andare in crash quando viene sollevata un'eccezione che non è stata gestita.

## Client

-**Scenario “Startup”:** Ugo è un tecnico che intende avviare la piattaforma InfinityGames.

Una volta avviato il web server Apache Tomcat, configura il DBMS MySQL per essere raggiunto dal medesimo web server. A questo punto il web server carica su Tomcat l'eseguibile della piattaforma InfinityGames. Dopo trenta secondi dall'avvio la piattaforma è operativa e raggiungibile dagli utenti.

-**Fase di Terminazione:** All'utente basterà chiudere la pagina del sistema per interrompere la comunicazione, interrompendo così la sessione TCP.

-**Scenario “Fallimento”:** Il server mostra al client una schermata di errore. Per evitare la perdita di dati, viene comunicato al DBMS di effettuare una forzata scrittura su memoria fisica delle informazioni ancora presenti in memoria volatile. Il DBMS mantiene dei registri di log dove sono annotate tutte le operazioni sul DB.

## **4. SERVIZI DEI SOTTOSISTEMI**

### **Gestione Utente**

<b>Sottosistema</b>	<b>Gestione Utente</b>
<b>Descrizione</b>	Questo sottosistema permette di effettuare tutte le operazioni di accesso e le operazioni relative al profilo degli utenti registrati alla piattaforma.
<b>Servizi Offerti</b>	
<b>Servizio</b>	<b>Descrizione</b>
Registrazione	Questo servizio consente la registrazione di qualsiasi utente che non sia già registrato.
Login	Questo servizio consente l'autenticazione di qualsiasi utente che sia già registrato.
Logout	Questo servizio consente il logout di qualsiasi utente che sia già registrato.
Visualizzazione Area Personale	Questo servizio consente la visualizzazione dell'area personale di utente che sia già registrato.
Modifica Profilo	Questo servizio consente la modifica della password, dell'username e dell'immagine di profilo dell'utente registrato.

## Gestione Prodotto

Sottosistema	Gestione Prodotto
<b>Descrizione</b>	Questo sottosistema permette le interazioni tra utente (registrato, visitatore ed amministratore) e i prodotti.
Servizi Offerti	
Servizio	Descrizione
Aggiungi Prodotto al Carrello	Questo servizio permette di aggiungere un prodotto al carrello.
Rimuovi Prodotto dal Carrello	Questo servizio permette di rimuovere un prodotto dal carrello aggiunto in precedenza.
Aggiungi Prodotto alla Lista dei Desideri	Questo servizio permette di aggiungere un prodotto alla lista dei desideri.
Rimuovi Prodotto dalla Lista dei Desideri	Questo servizio permette di rimuovere un prodotto dalla lista dei desideri.
Ricerca Prodotto	Questo servizio permette di ricercare un prodotto nello store tramite la digitazione del titolo.
Aggiungi Prodotto allo store da Amministratore	Questo servizio permette di aggiungere un prodotto allo store da amministratore.
Rimuovi Prodotto dallo store da Amministratore	Questo servizio permette di rimuovere un prodotto dallo store da amministratore.
Modifica Prodotto nello store da Amministratore	Questo servizio permette di modificare un prodotto nello store da amministratore.

## Gestione Vista Prodotto

Sottosistema	Gestione Vista Prodotto
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni di visualizzazione dei prodotti.
Servizi Offerti	
Servizio	Descrizione
Visualizza lista prodotti nello store	Questo servizio consente la visualizzazione dei prodotti divisi per categoria, più popolari, più venduti, più recenti e in sconto.
Visualizza Pagina Prodotto	Questo servizio consente di visualizzare un singolo prodotto nello store e le relative informazioni.
Visualizza Libreria	Questo servizio consente di visualizzare tutti i prodotti acquistati in una lista.
Visualizza Carrello	Questo servizio consente di visualizzare la pagina del carrello.
Visualizza Lista dei Desideri	Questo servizio consente di visualizzare i prodotti aggiunti alla lista dei desideri tramite una lista.

## Gestione Categoria

Sottosistema	Gestione Categoria
Descrizione	Questo sottosistema permette di gestire le informazioni relative alle categorie di prodotti presenti sulla piattaforma.
Servizi Offerti	
Servizio	Descrizione
Aggiungi Categoria allo Store	Questo servizio consente di aggiungere una categoria di videogiochi allo store.
Rimuovi Categoria dallo Store	Questo servizio consente di rimuovere una categoria di videogiochi già presente dallo store.
Modifica Categoria nello Store	Questo servizio consente di modificare una categoria di videogiochi già presente dallo store.

## Gestione Amministratore

Sottosistema	Gestione Amministratore
Descrizione	Questo sottosistema permette la gestione degli utenti della piattaforma da parte dell'amministratore.
Servizi Offerti	
Servizio	Descrizione
Visualizza Lista utenti Registrati	Questo servizio consente di visualizzare la lista di utenti registrati alla piattaforma, con le relative informazioni riguardanti il ruolo (amministratore o utente normale)
Promuovi Utente ad Amministratore	Questo servizio consente di promuovere un utente ad amministratore aggiuntivo della piattaforma.
Declassa Amministratore	Questo servizio consente di declassare un amministratore ad utente normale.
Rimuovi utente registrato	Questo servizio consente di rimuovere dalla piattaforma un utente registrato.



## Gestione Ordini

Sottosistema	Gestione Ordini
Descrizione	Questo sottosistema permette la gestione degli ordini e degli acquisti.
Servizi Offerti	
Servizio	Descrizione
Acquisto Prodotto	Questo servizio consente di acquistare un prodotto.
Visualizza Storico Ordini	Questo servizio consente di visualizzare la lista di ordini effettuati dall'utente.
Visualizza Ordini utente da Amministratore	Questo servizio consente di visualizzare la lista di ordini effettuati dall'utente da parte dell'amministratore.

## **5. GLOSSARIO**

**Amministratore:** Utilizza il sistema per modificare contenuti sulla piattaforma e gestire gli utenti.

**Application Logic Layer:** si occupa della gestione della logica del Sistema.

**Client:** componente che accede a servizi e risorse di un altro componente detto server.

**Data access layer:** si occupa della gestione dei dati persistenti e dello scambio di dati tra i sottosistemi.

**DBMS:** sistema software per la creazione, manipolazione e interrogazione efficiente di database.

**HTTP:** protocollo di trasferimento di ipertesti che consente a due macchine, client e server, di interagire attraverso un meccanismo di richiesta/risposta. Il client inoltra una richiesta al server, che verrà soddisfatta con la risposta di quest'ultimo.

**JSP:** tecnologia di programmazione web utilizzata per fornire contenuti dinamici.

**Layer:** suddivisione del sistema in livelli (o strati).

**Piattaforma:** Definisce l'insieme delle funzionalità fornite dal sistema attraverso l'applicazione web.

**Presentation layer:** si occupa della visualizzazione della piattaforma.

**RAD (Requirement Analysis Document):** documento contenente informazioni inerenti al sistema da realizzare raccolte durante la fase di Requirement Analysis e Requirement Elicitation.

**SDD (System Design Document):** Documento formalizzato alla definizione di obiettivi di progettazione del sistema, decomposizione del sistema in sottosistemi più piccoli e scelta di architettura software più adatta al sistema.

**Server:** componente che gestisce traffico di informazioni e fornisce servizi e risorse attraverso la rete.

**Servlet:** oggetti Java all'interno del server web che permettono di creare web applications in combinazione con JSP.

**Sistema:** Definisce la totalità delle componenti dal punto di vista strutturale/implementativo.

**Utente:** Un utilizzatore del sistema che sia esso un amministratore, un utente normale.

**Visitatore:** Utente non registrato visualizza i prodotti nello store e aggiunge prodotti al carrello senza poter proseguire con l'acquisto.

**Web Browser:** applicazione software installata sul client che permette di visualizzare e navigare le risorse del web.