



UNIVERSITÀ DEGLI STUDI DI SALERNO  
FACOLTÀ DI SCIENZE MM.FF.NN.  
CORSO DI LAUREA TRIENNALE IN INFORMATICA  
CORSO DI INGEGNERIA DEL SOFTWARE

# Gamerland

## **System Design Document (S.D.D)**

## **Specifica Gestione dati persistenti**

**VERSIONE 1.1**

**Anno Accademico 2017/2018**

## Top Manager:

Professore
Prof. De Lucia Andrea
Prof. Francese Rita

## Partecipanti:

Nome	Matricola
Capone Natale	0512103586
Stanzione Roberto	0512103763
Iuliano Gianpaolo	0512103514

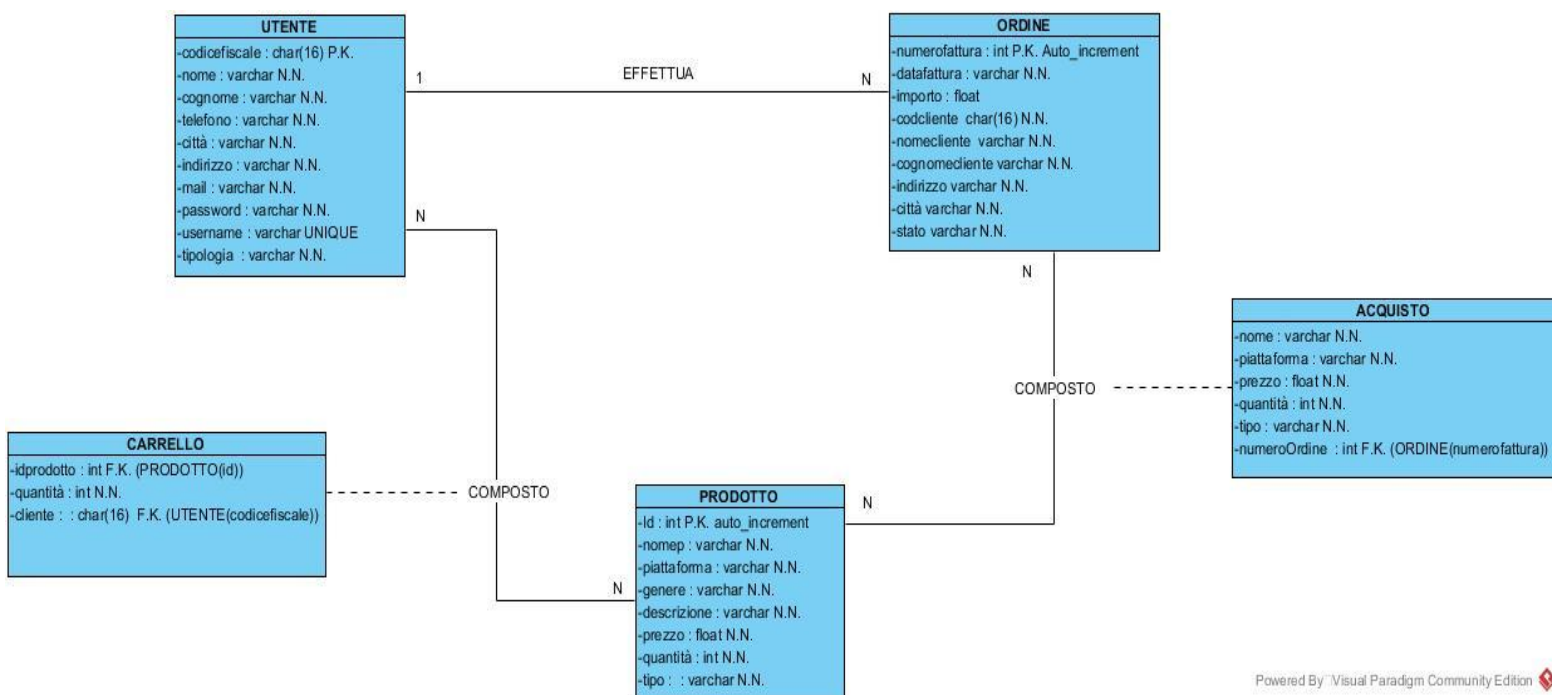
## Revision History:

Data	Versione	Descrizione	Autore
14/12/2017	1.0	Descrizione gestione dati persistenti	Tutto il team
22/12/2017	1.1	Inserimento schema e definizione entità	Tutto il team

# GESTIONE DEI DATI PERSISTENTI

Il sistema utilizza un database relazionale per semplificare la gestione dei dati persistenti e le operazioni da eseguire su di essi. Il DBMS utilizzato per memorizzare questi dati sarà Mysql e comunicherà con il sistema utilizzando JDBC. La scelta di utilizzare questo DBMS è data dal fatto che verrà utilizzato il linguaggio SQL per le query, e quindi per interrogare, aggiornare e gestire i dati memorizzati.

Di seguito, viene mostrato lo schema E-R del database del sistema:



Inizialmente, si era pensato di avere tre entità separate per rappresentare i diversi tipi di prodotti che il sistema permette di acquistare: videogioco, accessorio e console. Dopo una fase di analisi, si è notato che i tre tipi di prodotti avevano gli stessi attributi tranne che per videogioco, in quanto quest'ultimo ha in aggiunta l'attributo genere. Nonostante questo, si è deciso di non specificare tutte e tre le entità in modo separato, ma di rappresentarle in un'unica entità PRODOTTO e differenziarle attraverso la presenza di un attributo, "tipo", che verrà settato in modo appropriato a seconda della tipologia del prodotto. Nel caso in cui la tipologia non sia videogioco, l'attributo genere sarà lasciato vuoto. Nell'entità ORDINE sono presenti alcuni attributi riferiti all'utente che effettua l'ordine. Questi attributi creano ridondanza dato che li possiamo trovare anche nell'entità UTENTE, ma è stato deciso di inserirli ugualmente a causa del fatto che, dato che il sistema prevede la possibilità di un cliente di cancellare il proprio account, queste informazioni dell'ordine sono importanti e devono essere conservate anche nel caso in cui un cliente decida di cancellare il proprio account dal sistema

▪ DESCRIZIONE DELLE IDENTITA' DEL DATABASE

TABELLA	ATTRIBUTI
UTENTE	Codicefiscale (char (16)) Primary key Nome (varchar (20)) not null Cognome (varchar (40)) not null Telefono (varchar (10)) not null Città (varchar (30)) not null Indirizzo (varchar (50)) not null Mail (varchar (40)) not null Password (varchar (10)) not null Id (varchar (10)) not null unique Tipologia (varchar (50)) not null
ORDINE	Numerofattura (int, auto_increment) primary key Datafattura (varchar (30)) not null codcliente (char (16)) not null nomecliente (varchar (30)) not null cognomecliente (varchar (40)) not null indirizzo (varchar (50)) not null città (varchar (30)) not null Importo (float) not null Stato (varchar(20)) not null
PRODOTTO	Id (int, auto_increment) primary key Nomep (varchar (50)) not null Prezzo (float) not null Genere (varchar (50)) not null Piattaforma (varchar (50)) not null Descrizione (varchar (200)) not null Quantità (int) not null Tipo (varchar (50)) not null
ACQUISTO	Nome (varchar (50)) not null Piattaforma (varchar (50)) not null Prezzo (float) not null Quantità (int) not null Tipo (varchar (50)) not null NumeroOrdine (int) foreign key (ORDINE)
CARRELLO	Idprodotto (int) foreign key (PRODOTTO) Quantità (int) not null Cliente (char (16)) foreign key (UTENTE)