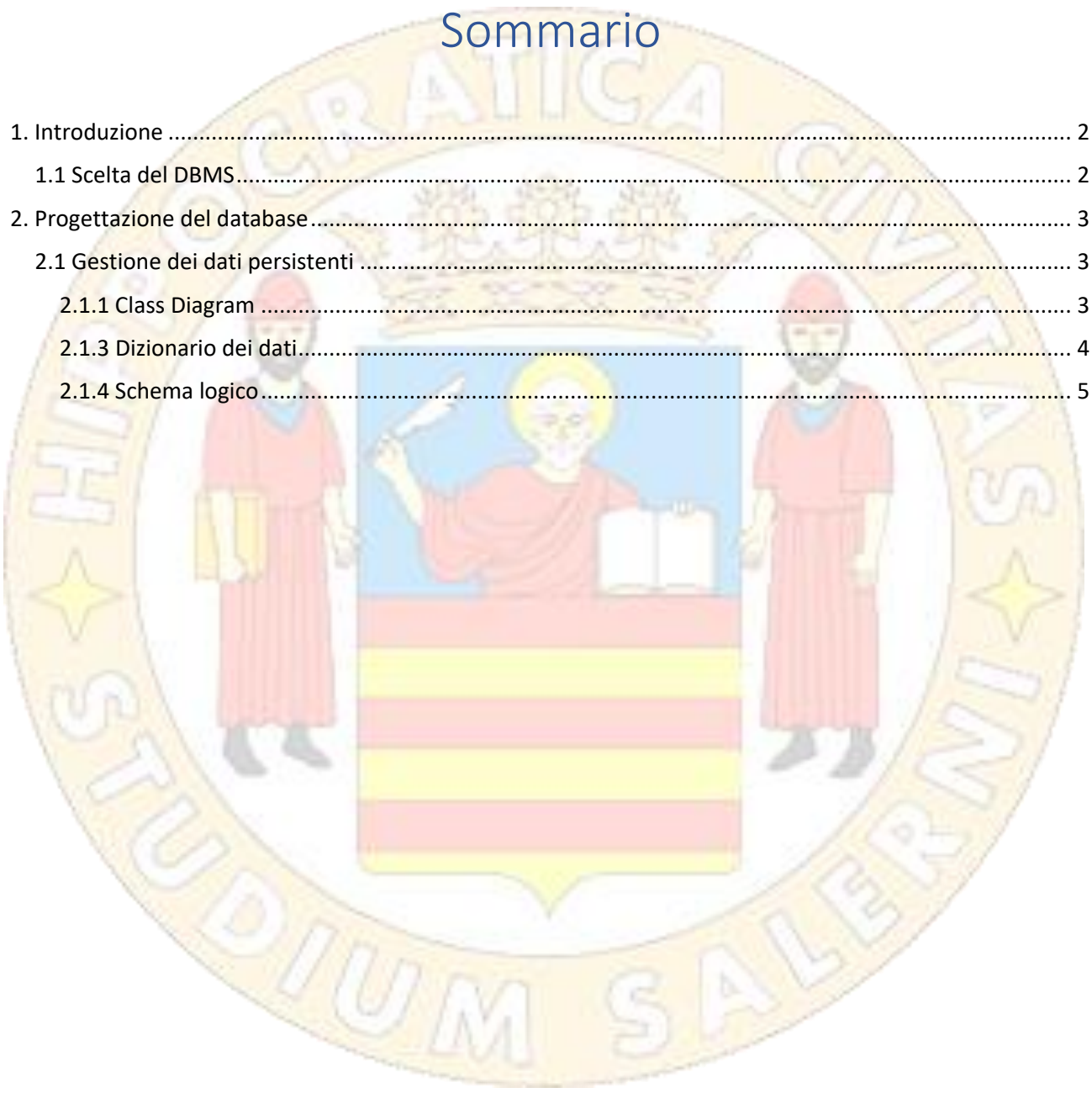
Università Degli Studi Di Salerno

Progetto di Ingegneria del software 2018/2019

Gestione dati persistenti

Sommario



1. Introduzione

Questo documento descrive nel dettaglio in DBMS che si è scelti di utilizzare e tutta la logica riguardante la gestione dei dati persistenti.

1.1 Scelta del DBMS

Il problema della persistenza dei dati nasce dalla necessità di rendere permanenti alcune informazioni all'interno di un sistema anche quando questo è spento. Molte volte capita di avere dei programmi con un gran numero di dati da gestire, dati che non possono essere reinseriti dopo un blackout o dopo lo spegnimento della macchina; si pensi ad un archivio di mille utenti contenente tutti i dati anagrafici. Sarebbe impensabile il reinserimento degli stessi ogni volta che la macchina sul quale risiedono i dati si spegne. Bisogna, quindi, trovare il modo per poter tener traccia dei dati utili anche quando l'applicazione che li ha utilizzati e/o creati ha smesso di funzionare. I dati persistenti sono sempre lì e vengono cancellati solo se lo si richiede espressamente. Nel caso specifico, in previsione del gran numero di accessi al database e alla necessità di gestire anche più query contemporaneamente, si è optati per un DBMS relazionale, MySQL. La sigla SQL presente all'interno del nome sta a indicare che si tratta di un DBMS in grado di interpretare le istruzioni del linguaggio SQL (utile per le query).

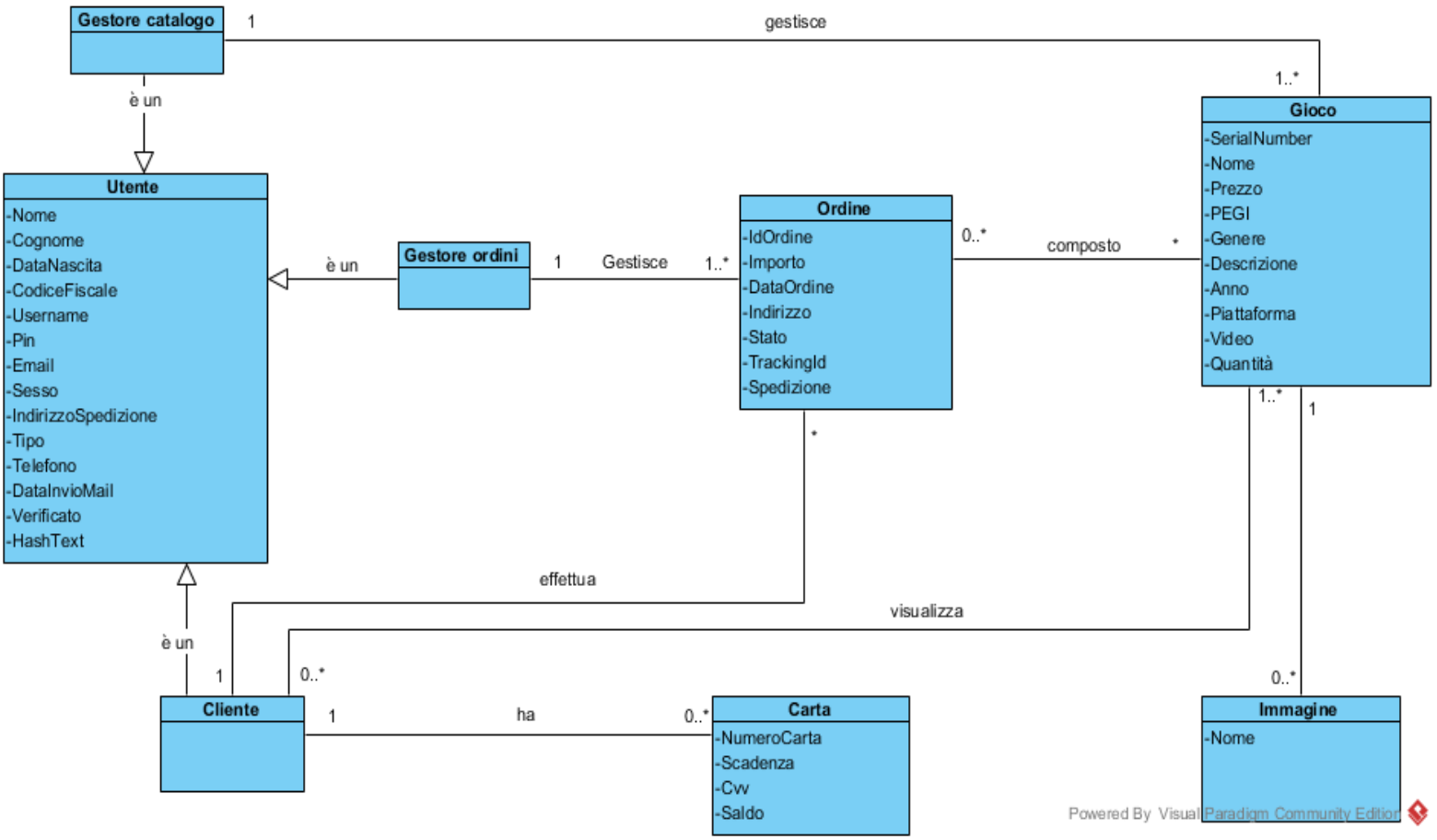
MySQL permette la creazione di "database relazionali" ossia consente la conservazione dei dati in tabelle separate anziché in un'unica grande entità. Questa sua particolare caratteristica consente di raggiungere una buona flessibilità e velocità di accesso ai dati ed una maggior modellazione delle basi dati. Semplicità d'uso, robustezza e velocità di esecuzione sono le caratteristiche principali di questo DBMS.

La sua struttura multi-thread consta di un thread fisso che controlla le connessioni in ingresso e un thread attivo per ogni connessione. I vari client non devono aspettare che le queries di altri siano soddisfatte, possono lavorare simultaneamente. Il thread che controlla le connessioni impedisce che due thread scrivano sulla stessa tabella nello stesso momento. In ogni tabella all'interno del nostro database sarà possibile accedere ai dati per compiere operazioni quali inserimento, modifica, cancellazione o semplicemente consultazione.

2. Progettazione del database

2.1 Gestione dei dati persistenti

2.1.1 Class Diagram

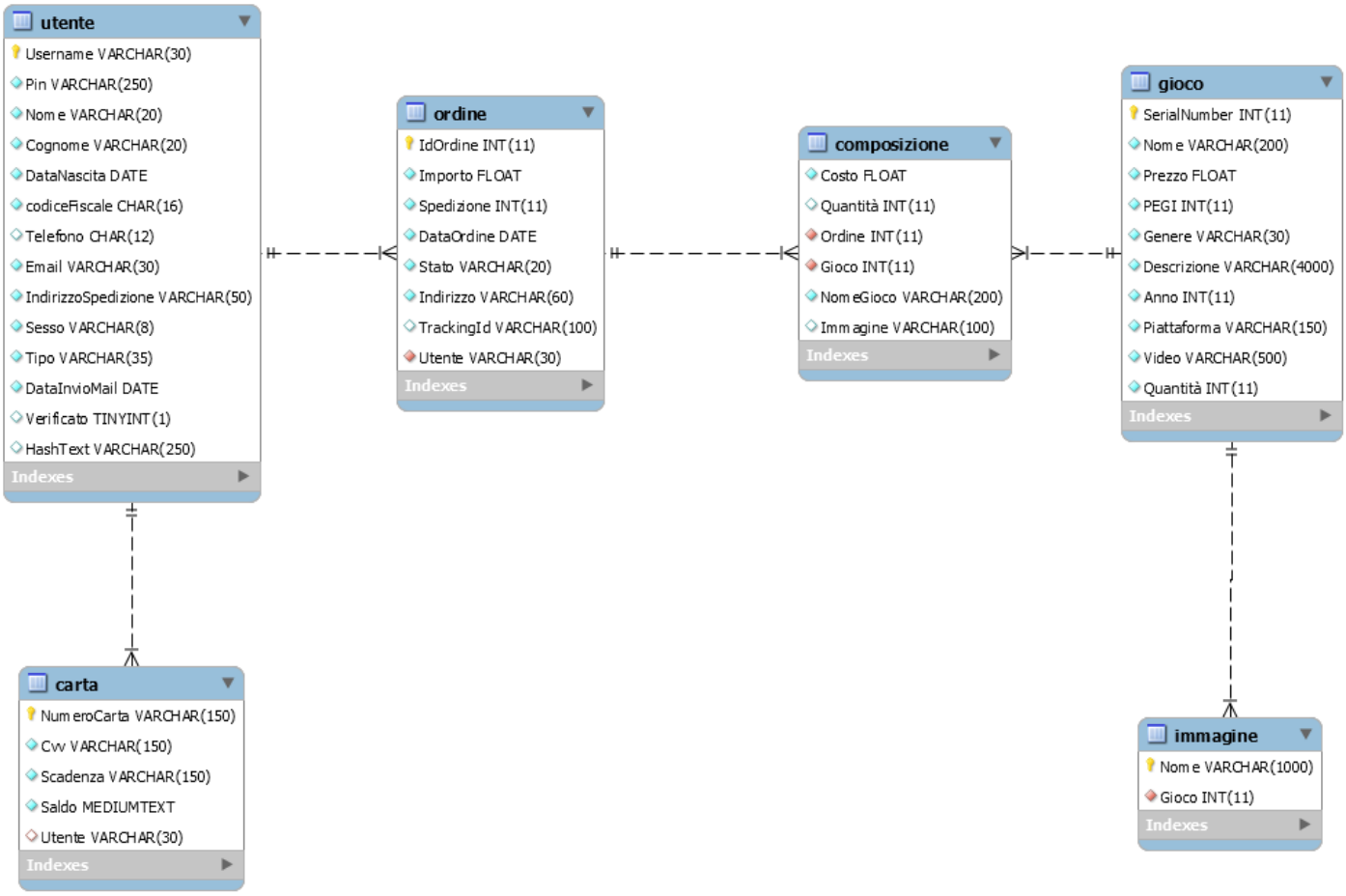


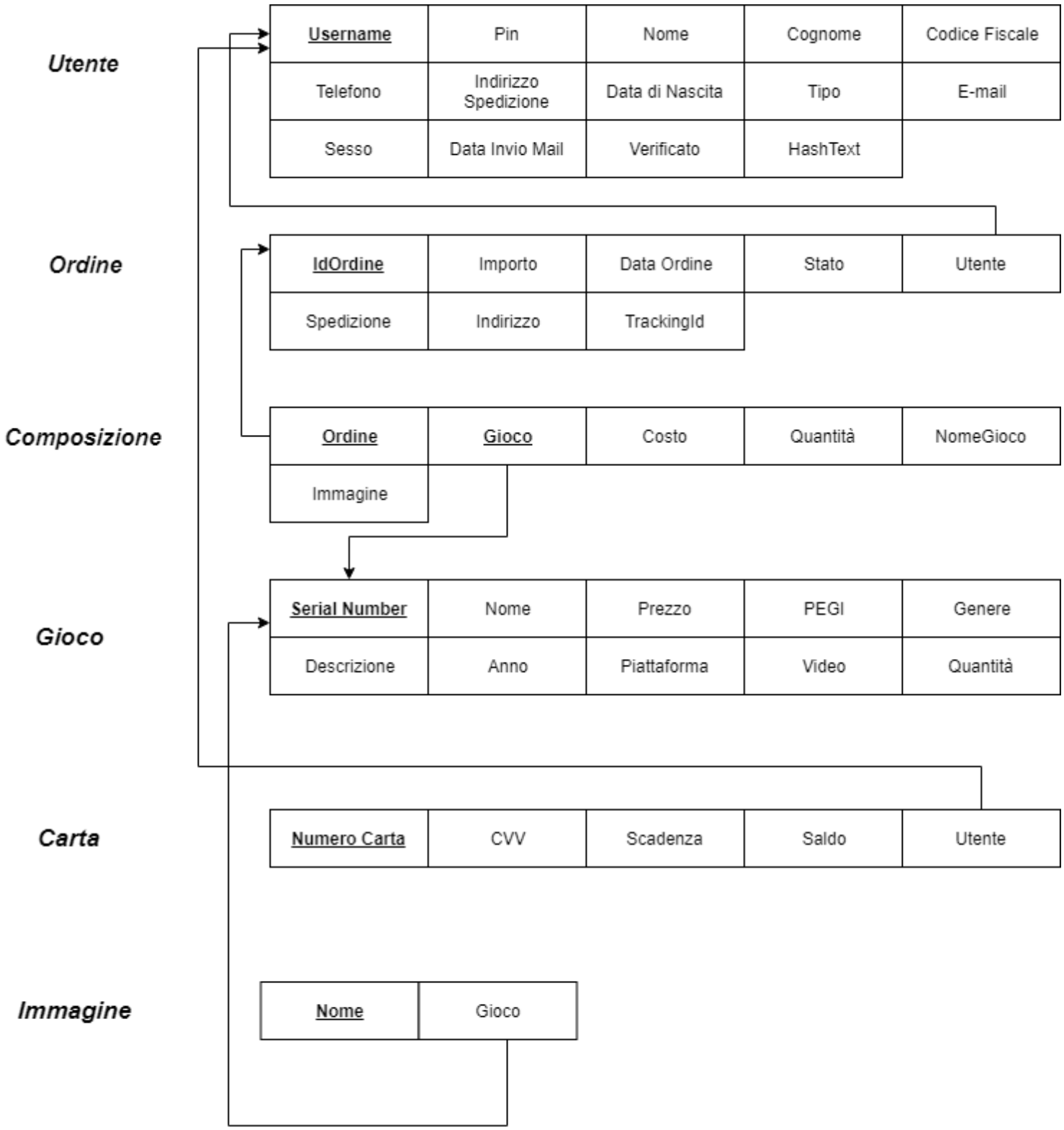
2.1.3 Dizionario dei dati

2.1.3.1 Entità

Entità	Descrizione	Attributi	Identificatore
Utente	Contiene le informazioni di un utente del sistema	Nome, cognome, username, password, e-mail, sesso, indirizzo, codiceFiscale, dataDiNascita, tipo, telefono, DataInvioEmail, HashText, Verificato	Username
Carta	Contiene le informazioni relative ad una carta di pagamento.	numeroCarta, scadenzacarta, cvv, saldo.	numeroCarta
Gioco	Contiene le informazioni relative ad un gioco presente nel sistema.	SerialNumber, Nome, prezzo, PEGI, genere, descrizione, piattaforma, video, quantità, anno.	SerialNumber
Ordine	Contiene le informazioni relative ad un ordine effettuato da un utente.	IdOrdine, importo, dataOrdine, stato, indirizzoConsegna, trackingID, spedizione.	IdOrdine.
Immagine	Contiene le informazioni relative ad un immagine di un gioco.	nome	nome.

2.1.4 Schema logico




Utente
Username
Pin
Nome
Cognome
Codice Fiscale
Telefono
Indirizzo Spedizione
Data di Nascita
Tipo
E-mail
Sesso
Data Invio Mail
Verificato
HashText
Ordine
IdOrdine
Importo
Data Ordine
Stato
Utente
Spedizione
Indirizzo
TrackingId
Composizione
Ordine
Gioco
Costo
Quantità
NomeGioco
Immagine
Gioco
Serial Number
Nome
Prezzo
PEGI
Genere
Descrizione
Anno
Piattaforma
Video
Quantità
Carta
Numero Carta
CVV
Scadenza
Saldo
Utente
Immagine
Nome
Gioco