

Nopaynogame

Progetto di Basi di Dati Modulo 2

Anno Accademico  
2017/1018

**Gruppo**  
*Elia Ferrotti*, 847059  
*Fabrizio Ortotti*, 848856  
*Alessia Michela di Campi*, 861844  
*Andrea Corazza*, 843173

Indice dei contenuti

[1. Introduzione 2](#_Toc511857477)

[2 Descrizione del Progetto 2](#_Toc511857478)

[2.1 Scelta del Progetto 3](#_Toc511857479)

[2.2 Modifica e/o aggiunte alla consegna 3](#_Toc511857480)

[3 Database 3](#_Toc511857481)

[3.1 Schema Concettuale 3](#_Toc511857482)

[3.2 Schema Relazionale 3](#_Toc511857483)

[3.3 Creazione del Database 4](#_Toc511857484)

[3.4 Popolamento del Database 7](#_Toc511857485)

[4 Diagramma Applicazione Web 7](#_Toc511857486)

[5 Interfacce 7](#_Toc511857487)

[6 Struttura del Progetto 7](#_Toc511857488)

[6.1 File 1 7](#_Toc511857489)

[6.2 File 2 7](#_Toc511857490)

[6.3 File 3 7](#_Toc511857491)

[6.4 File 4 7](#_Toc511857492)

[7 Open Points 8](#_Toc511857493)

[7.1 Struttura 8](#_Toc511857494)

[7.2 Dubbi 8](#_Toc511857495)

# Introduzione

Il presente documento vuole documentare e presentare il progetto del sito NoPayNoGame svolto dal gruppo presentato in copertina.

Il documento presenterà una descrizione della scelta di progetto con eventuali modifiche inserite, la presentazione del modello dei dati dall’analisi concettuale all’analisi relazione alla realizzazione, la struttura e la descrizione dell’applicazione web, l’analisi delle query effettuate per gestire le diverse funzionalità.

# Descrizione del Progetto

Lo scopo del progetto è la creazione di un negozio elettronico di videogiochi che offre diverse funzionalità in base al tipo di utilizzatori. I possibili utilizzatori previsti sono: consumatori, magazzinieri, amministratori e superutenti.

Nel caso l’utilizzatore sia un Utente/Consumatore non registrato, le feature offerte sono:

* La visualizzazione del catalogo di giochi;
* Le ricerche sul catalogo stesso;
* La visualizzazione del singolo gioco;
* La creazione un nuovo profilo;
* L’accesso alla piattaforma.

Nel caso l’utilizzatore che ha effettuato l’accesso sia un Consumatore normale, le feature offerte sono:

* L’accesso ai cataloghi e le ricerche;
* La visualizzazione del singolo gioco;
* La creazione di una recensione del gioco se già acquistato;
* L’aggiunta al carrello degli articoli disponibili;
* La visualizzazione del carrello;
* La modifica/cancellazione del carrello;
* La creazione di un ordine;
* La modifica/cancellazione di un ordine;
* La visualizzazione della pagina del suo profilo;
* La modifica dei dati del proprio profilo;
* La visualizzazione di tutte le recensioni scritte;
* La modifica/cancellazione delle recensioni scritte.

Nel caso l’utilizzatore che ha effettuato l’accesso sia un Magazziniere, le feature offerte sono:

* L’accesso alla pagina di amministrazione;
* La modifica/cancellazione dei giochi correnti;
* L’aggiunta di nuovi giochi.

Nel caso l’utilizzatore che ha effettuato l’accesso sia un Amministratore, le feature offerte sono quelle del magazziniere con l’aggiunta di:

* Gestione stato ordini;
* Cancellazioni ordini.

Nel caso l’utilizzatore che ha effettuato l’accesso sia un Superutente, le feature offerte sono quelle dell’amministratore con l’aggiunta di:

* Modifica dei profili;
* Creazione di nuovi profili.

Il progetto è costituito da due parti: un database per la gestione di tutti i dati, utenti, giochi e un sito web. Per comodità organizzative è stato deciso di utilizzare Altervista.org come host per la parte web e di database. Il progetto è consultabile al seguente link: <http://nopaynogame.altervista.org/nopaynogame_/>.

## Scelta del Progetto

La traccia scelta per lo sviluppo del progetto è la seguente:

|  |
| --- |
| **NEGOZIO ELETTRONICO DI VIDEOGIOCHI (3-4)**  Base di dati:  **Videogiochi** (titolo, console, prezzo, quantità disponibile in magazzino). **Utenti** con nome e  cognome, indirizzo, telefono, login e password. **Ordini**: utente, titoli ordinati, data, tipo di  pagamento.  **Accesso senza login**:  Catalogo dei videogiochi, visualizzazione e ricerca. Registrazione di un utente.  **Accesso con login**:  Ordine.  Cancellazione di un'ordinazione.  **Accesso con login di amministratore**:  Inserimento e modifica dei dati, degli utenti e degli ordini. |

## Modifica e/o aggiunte alla consegna

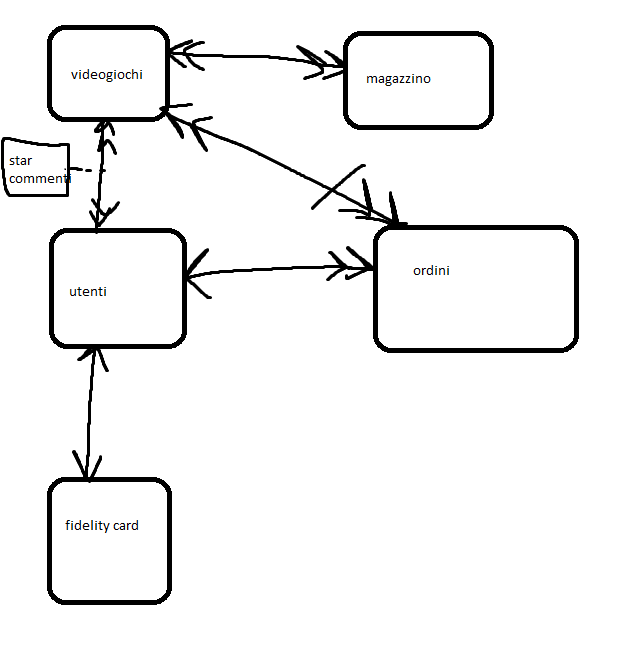
Abbiamo scelto di modificare la traccia del progetto ampliandone le funzionalità offerte come segue:

* È presente la gestione dei magazzini: è prevista una tabella magazzino che tiene traccia dei dati del magazzino stesso e delle quantità di ogni singolo gioco in esso presente. In questo modo è possibile gestire la presenza di molteplici magazzini;
* È presente un sistema di recensioni ai prodotti costituite da un voto (1-5 stelle) e un commento opzionale;
* È presente la gestione dell’elenco dei generi riferiti a ciascun gioco;
* È presente un campo portafoglio ed un campo punti fedeltà per gestire l’accredito verso l’utente in caso di eliminazione dell’ordine e per offrire al cliente un sistema di bonus acquistabili con i punti (ricaricati attraverso gli acquisti) che permettono di ricaricare il portafoglio online;
* È previsto un sistema di gestione dello stato dell’ordine (pagato - evaso):
  + Utente: quando un ordine risulta ancora non pagato, l’utente ha la possibilità di eliminarlo e/o modificarlo in base alle disponibilità dei giochi nel magazzino;
  + Amministratore: ha la possibilità di registrate il pagamento e solo a quel punto di poter evadere l’ordine. Nel caso in cui l’ordine sia non-pagato e/o non-evaso, l’amministratore ha la possibilità di eliminare l’ordine ed effettuare un eventuale rimborso;
* Sono presenti più ruoli rispetto a quello proposto: Utente normale, Magazziniere, Amministratore, Superuser. Ogni ruolo ha delle feature diverse, come spiegato nella Descrizione del Progetto.

# Database

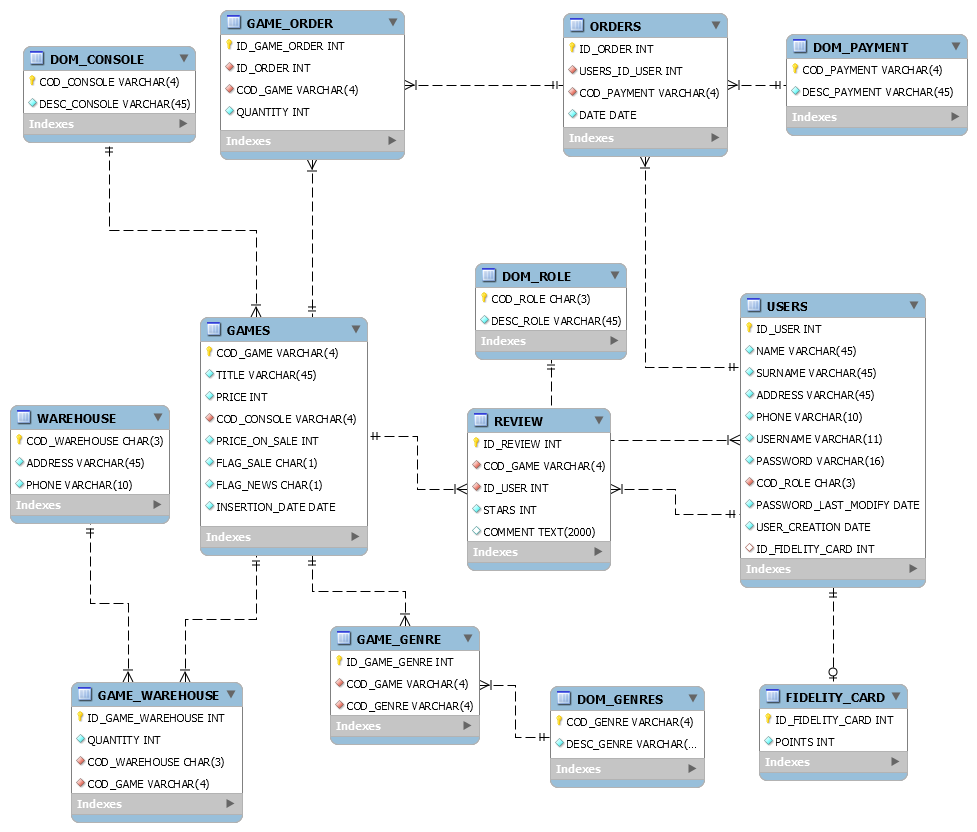
## Schema Concettuale

Di seguito lo schema concettuale del database:



## Schema Relazionale

Di seguito lo schema relazionale del database:



Descrizione dello schema:

## Creazione del Database

Per la creazione del database è stato creato un file DLL contenente tutte le create necessarie:

-- DATABASE: NOPAYNOGAME

-- TABLE STRUCTURE FOR TABLE DOM\_CONSOLE

CREATE TABLE DOM\_CONSOLE (

COD\_CONSOLE VARCHAR(4) NOT NULL,

DESC\_CONSOLE VARCHAR(45) NOT NULL,

CONSTRAINT DOM\_CONSOLE\_PK PRIMARY KEY (COD\_CONSOLE)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE DOM\_GENRE

CREATE TABLE DOM\_GENRE (

COD\_GENRE VARCHAR(4) NOT NULL,

DESC\_GENRE VARCHAR(45) NOT NULL,

CONSTRAINT DOM\_GENRE\_PK PRIMARY KEY (COD\_GENRE)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE DOM\_PAYMENT

CREATE TABLE DOM\_PAYMENT (

COD\_PAYMENT VARCHAR(4) NOT NULL,

DESC\_PAYMENT VARCHAR(45) NOT NULL,

CONSTRAINT DOM\_PAYMENT\_PK PRIMARY KEY (COD\_PAYMENT)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE DOM\_ROLE

CREATE TABLE DOM\_ROLE (

COD\_ROLE CHAR(3) NOT NULL,

DESC\_ROLE VARCHAR(45) NOT NULL,

CONSTRAINT DOM\_ROLE\_PK PRIMARY KEY (COD\_ROLE)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE GAMES

CREATE TABLE GAMES (

COD\_GAME VARCHAR(4) NOT NULL,

TITLE VARCHAR(45) NOT NULL,

PRICE INT(11) NOT NULL,

COD\_CONSOLE VARCHAR(4) NOT NULL,

PRICE\_ON\_SALE INT(11) NOT NULL,

FLAG\_SALE CHAR(1) NOT NULL,

FLAG\_NEWS CHAR(1) DEFAULT 'Y' NOT NULL,

INSERTION\_DATE TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL,

CONSTRAINT GAMES\_PK PRIMARY KEY (COD\_GAME),

CONSTRAINT GAMES\_CK01 CHECK (FLAG\_SALE IN ('Y', 'N')),

CONSTRAINT GAMES\_CK02 CHECK (FLAG\_NEWS IN ('Y', 'N')),

CONSTRAINT GAMES\_CK03 CHECK (PRICE >= 0),

CONSTRAINT GAMES\_CK04 CHECK (PRICE\_ON\_SALE >= 0),

CONSTRAINT GAMES\_FK01 FOREIGN KEY (COD\_CONSOLE) REFERENCES DOM\_CONSOLE(COD\_CONSOLE)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE GAME\_GENRE

CREATE TABLE GAME\_GENRE (

ID\_GAME\_GENRE INT(11) NOT NULL AUTO\_INCREMENT,

COD\_GAME VARCHAR(4) NOT NULL,

COD\_GENRE VARCHAR(4) NOT NULL,

CONSTRAINT GAME\_GENRE\_PK PRIMARY KEY (ID\_GAME\_GENRE),

CONSTRAINT GAME\_GENRE\_FK01 FOREIGN KEY (COD\_GAME) REFERENCES GAMES(COD\_GAME),

CONSTRAINT GAME\_GENRE\_FK02 FOREIGN KEY (COD\_GENRE) REFERENCES DOM\_GENRE(COD\_GENRE)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE USERS

CREATE TABLE USERS (

ID\_USER INT(11) NOT NULL AUTO\_INCREMENT,

NAME VARCHAR(45) NOT NULL,

SURNAME VARCHAR(45) NOT NULL,

ADDRESS VARCHAR(45) NOT NULL,

PHONE VARCHAR(10) NOT NULL,

USERNAME VARCHAR(11) UNIQUE NOT NULL,

PASSSWORD VARCHAR(16) NOT NULL,

COD\_ROLE CHAR(3) NOT NULL,

PASSWORD\_LAST\_MODIFY TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL,

FIDELITY\_POINT INT(11) DEFAULT 0 NOT NULL,

CONSTRAINT USERS\_PK PRIMARY KEY (ID\_USER),

CONSTRAINT USERS\_FK01 FOREIGN KEY (COD\_ROLE) REFERENCES DOM\_ROLE(COD\_ROLE),

CONSTRAINT USERs\_CK01 CHECK (FIDELITY\_POINT >= 0)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE ORDERS

CREATE TABLE ORDERS (

ID\_ORDER INT(11) NOT NULL AUTO\_INCREMENT,

ID\_USER INT(11) NOT NULL,

COD\_PAYMENT VARCHAR(4) NOT NULL,

DATE\_ORDER TIMESTAMP DEFAULT CURRENT\_TIMESTAMP NOT NULL,

CONSTRAINT ORDERS\_PK PRIMARY KEY (ID\_ORDER),

CONSTRAINT ORDERS\_FK01 FOREIGN KEY (ID\_USER) REFERENCES USERS(ID\_USER),

CONSTRAINT ORDERS\_FK02 FOREIGN KEY (COD\_PAYMENT) REFERENCES DOM\_PAYMENT(COD\_PAYMENT)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE GAME\_ORDER

CREATE TABLE GAME\_ORDER (

ID\_GAME\_ORDER INT(11) NOT NULL AUTO\_INCREMENT,

ID\_ORDER INT(11) NOT NULL,

COD\_GAME VARCHAR(4) NOT NULL,

QUANTITY INT(11) NOT NULL,

CONSTRAINT GAME\_ORDER\_PK PRIMARY KEY (ID\_GAME\_ORDER),

CONSTRAINT GAME\_ORDER\_FK01 FOREIGN KEY (COD\_GAME) REFERENCES GAMES(COD\_GAME),

CONSTRAINT GAME\_ORDER\_FK02 FOREIGN KEY (ID\_ORDER) REFERENCES ORDERS(ID\_ORDER),

CONSTRAINT GAME\_ORDER\_CK01 CHECK (QUANTITY >= 0)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE REVIEW

CREATE TABLE REVIEW (

ID\_REVIEW INT(11) NOT NULL AUTO\_INCREMENT,

COD\_GAME VARCHAR(4) NOT NULL,

ID\_USER INT(11) NOT NULL,

STARS INT(11) NOT NULL,

COMMENT\_TEXT VARCHAR(2000),

CONSTRAINT REVIEW\_PK PRIMARY KEY (ID\_REVIEW),

CONSTRAINT REVIEW\_FK01 FOREIGN KEY (COD\_GAME) REFERENCES GAMES(COD\_GAME),

CONSTRAINT REVIEW\_FK02 FOREIGN KEY (ID\_USER) REFERENCES USERS(ID\_USER),

CONSTRAINT REVIEW\_CK01 CHECK (STARS BETWEEN 1 AND 5)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE WAREHOUSE

CREATE TABLE WAREHOUSE (

COD\_WAREHOUSE CHAR(3) NOT NULL,

ADDRESS VARCHAR(45) NOT NULL,

PHONE VARCHAR(45) NOT NULL,

CONSTRAINT WAREHOUSE\_PK PRIMARY KEY (COD\_WAREHOUSE)

) ENGINE=InnoDB;

-- TABLE STRUCTURE FOR TABLE GAME\_WAREHOUSE

CREATE TABLE GAME\_WAREHOUSE (

ID\_GAME\_WAREHOUSE INT(11) NOT NULL AUTO\_INCREMENT,

QUANTITY INT(11) NOT NULL,

COD\_WAREHOUSE CHAR(3) NOT NULL,

COD\_GAME VARCHAR(4) NOT NULL,

CONSTRAINT GAME\_WAREHOUSE\_PK PRIMARY KEY (ID\_GAME\_WAREHOUSE),

CONSTRAINT GAME\_WAREHOUSE\_FK01 FOREIGN KEY (COD\_WAREHOUSE) REFERENCES WAREHOUSE(COD\_WAREHOUSE),

CONSTRAINT GAME\_WAREHOUSE\_FK02 FOREIGN KEY (COD\_GAME) REFERENCES GAMES(COD\_GAME),

CONSTRAINT GAME\_WAREHOUSE\_CK01 CHECK (QUANTITY >= 0)

) ENGINE=InnoDB;

## Popolamento del Database

Per il popolamento del database è stato creato un file DML con le insert necessarie per l’avvio corretto del sito web:

# Diagramma Applicazione Web

Qui ci sarà il diagramma

# Interfacce

Qui faremo gli screen

# Struttura del Progetto

Qui descriviamo la struttura del progetto -> vedi punto 4 e incolliamo tutti i file commentandoli

## File 1

## File 2

## File 3

## File 4

# Open Points

In **grassetto** le pagine.  
In *corsivo* le componenti.  
{xx} permessi.

## Struttura

* *Header*:
  + Home nel logo
  + Genere ->
    - Genere 1
    - Genere 2
    - Genere 3
    - …
  + Piattaforma ->
    - Ps4
    - Xbox
    - …
  + Catalogo completo
  + Ricerca ?
  + {Amministrazione}
  + Carrello
  + Accedi / Nome Utente
* **Homepage**: carousel / tutti i giochi con flag attive / newsletter
* *Gioco*: foto, descrizione, disponibilità, console, prezzo, aggiungi al carrello, more
  + Pagina del gioco: titolo, console, ecc ecc
  + Eventuale pop-up gioco come quickview (vedi opengames)
* **Genere**: Grid di giochi
* **Piattaforma**: Grid di giochi
* **Catalogo completo**: lista con collapse
* **Ricerca**: ricerca in %like% -> Grid di giochi
* {**Amministrazione**}:
  + Aggiungere un nuovo prodotto {AM, SU}
  + Modificare i prodotti esistenti {AM, SU}
  + Aggiungere amministratore {SU}
  + Bloccare utente {SU}
  + Aggiungere fidelity card {AM, SU}
* *Accedi*: popup login/registrazione
* **Registrazione**: pagina registrazione utente (inviamo una mail per confermare la registrazione cosi ci occupiamo della gestione degli account falsi? Volendo io ho già il modulo pronto con l'hashing)
* **Carrello**: elenco prodotti e tasto acquista modifica quantità del prodotto, eliminare prodotto o eliminare tutto il carrello
* **Nome Utente**:
  + dettagli utente
  + ordini completati , tutti gli ordini con lo stato(in lavorazione ecc)
  + modifica dei dati
* *Footer*:
  + Chi siamo
  + Contattaci
  + Iscriviti alla newsletter

## Dubbi

* Come gestiamo le newsletter? (vogliamo far arrivare le mail per davvero?)
* Come gestiamo la quantità nel magazzino?
  + Diminuisce all’acquisto o è real-time con il carrello? (diminuisce quando uno paga)
* PriceOnSale o Discount che gestisce la percentuale?
* Campo InsertionDate determina valore Flag\_News automaticamente da DB
  + Guardare Trigger <https://it.wikipedia.org/wiki/Trigger_(basi_di_dati)>
  + Events
* Campi tecnici: quali e dove (data ultima modifica, utente ultima modifica, ecc)? (data inserimento prodotto, ultima modifica password e/o email per l'utente)