



MEDITERRANEA  
UNIVERSITÀ DI REGGIO CALABRIA

# Progetto Basi di dati I

NEGOZIO DI VIDEOGIOCHI  
(GAMESTOP)  
GIOVANNI IANNIZZI

CORSO DI LAUREA TRIENNALE IN INGEGNERIA  
DELL'INFORMAZIONE A.A. 2015/16 |  
Docente: Domenico Ursino

# Sommario

## **Descrizione del Contesto di Riferimento**

Informazioni Generali .....	4
Descrizione della Realtà d'Interesse .....	5

## **Specifica e Analisi dei Requisiti**

Requisiti Funzionali .....	6
Requisiti Non Funzionali .....	7
Diagrammi dei Casi D'uso .....	9
Elenco dei Casi D'uso .....	14
Descrizione dei Casi D'Uso .....	15

## **Progettazione Concettuale**

Schema E/R.....	44
Dizionario delle Entità .....	45
Dizionario delle Relazioni .....	46
Dizionario dei Vincoli.....	47

## **Progettazione Logica**

Ristrutturazione dello Schema E/R .....	48
Schema E/R Ristrutturato .....	52
Traduzione verso il Modello Relazionale .....	53
Schema Relazionale.....	62
Vincoli di Integrità Referenziale .....	63

## **Progettazione Fisica**

Scelta degli Indici .....	64
Stima Spazi su Disco .....	65

## **Progettazione delle Applicazioni**

Mappa del sito .....	71
Process Flow .....	72
Mock-up livello 0 .....	77

Mock-up livello 2 .....	90
<b>Implementazione</b>	
Implementazione MySQL .....	94
Query.....	97

# Negozi di videogiochi

*GameStop*



# Descrizione della realtà d'interesse per realizzare un sistema informativo per la gestione di un negozio di videogiochi

## *Informazioni generali*

GameStop Corporation, con sede a Grapevine, in Texas, è il più grande rivenditore di videogiochi nuovi e usati del mondo, ma si occupa anche della vendita di accessori per videogiochi, console, figurine ed altro.

Da qualche anno, ritirano iPod touch, iPhone ed occasionalmente anche iPad e Tablet PC. La compagnia conta più di 6.000 negozi dislocati in 17 paesi.

I negozi operano sotto i marchi GameStop, Software Etc., FuncoLand, e EB Games (quest'ultimo, sigla di Electronics Boutique, è entrato in GameStop nel 2005 in seguito a una fusione).

GameStop è presente anche sul web attraverso gli e-shop GameStop.com e EBgames.com, tramite il magazine Game Informer (anche in formato cartaceo) e tramite il sito internet Kongregate. Dal novembre 2009 è attivo anche in Italia il sito di e-commerce GameStop.it

*La struttura, e le funzionalità che dovrà prevedere il sistema sono di seguito specificate.*

La filiale GameStop presa in considerazione ha sede in Reggio Calabria situata presso il Corso Giuseppe Garibaldi.

Ogni cliente ha la possibilità di possedere una tessera fedeltà tramite una registrazione fornendo le proprie generalità all'amministrazione, ad ogni tessera è associata un' identificativo e altre informazioni, a sua volta la tessera offre dei vantaggi come offerte(sconti) su giochi e console ed inviti speciali ad eventi.

La tessera è divisa in 4 livelli in ordine di importanza crescente, il livello 0 è gratis si dà al momento dalla registrazione, l'aumento di livello è a pagamento, tale scelta è dovuta al fatto che più si aumenta di livello e più le offerte sono alte.

Il cliente registrato, quindi possessore di una tessera ha la possibilità di portare in negozio i propri giochi, le proprie console usate, le quali sono visionate dall' amministrazione, il controllo dell'integrità è molto importante perché la vendita di un prodotto non funzionante o con problemi gravi abbasserebbero la fiducia tra cliente e negozio, se dovessero passare specifici controlli vengono ritirati come usato per poi essere rivenduti, ad ogni prodotto usato viene dato un codice ed altre informazioni tra cui il livello di integrità del prodotto.

Il cliente può prenotare giochi e console prima dell'uscita effettiva dalla Software House per poi ritirarlo in negozio, la prenotazione può essere effettuata solo da clienti registrati, questo per spingere il cliente alla registrazione.

Il sistema necessita durante l'orario di chiusura, il salvataggio di tutti i dati relativi ai clienti.

# Specifica e analisi dei requisiti

La prima tappa nella realizzazione di un buon sistema informatico consiste nel comprendere e razionalizzare i bisogni che il committente comunica, con la maggiore precisione possibile.

Analizzando con attenzione le parole del committente, in base alle proprie esigenze, procediamo con la formulazione dei requisiti.

## *Requisiti funzionali*

- **RF1:** il sistema dovrà essere in grado di gestire le attività CRUD sui Clienti;
- **RF2:** il sistema dovrà essere in grado di gestire le attività CRUD sui prodotti sia essi nuovi che usati;
- **RF3:** il sistema dovrà essere in grado di gestire le attività CRUD sulle Prenotazioni sia essi di videogiochi e console;
- **RF4:** il sistema dovrà essere in grado di gestire le attività CRUD sulle Software House;
- **RF5:** il sistema dovrà essere in grado di gestire le attività CRUD sugli Eventi;
- **RF6:** il sistema dovrà essere in grado di gestire le attività CRUD sui Documenti fiscali;
- **RF7:** il sistema dovrà essere in grado di gestire le attività CRUD sulle tessere;
- **RF8:** Il sistema dovrà consentire di effettuare calcoli statistici su tutti i dati contenuti nel database.
- **RF9:** Il sistema dovrà consentire da parte del cliente registrato di ricercare informazioni sulle prenotazioni effettuate.
- **RF10:** il sistema dovrà autorizzare la prenotazione a videogiochi e console solo a clienti registrati;

- **RF11:** il sistema dovrà garantire l'utilizzo degli sconti sui prodotti acquistati dai clienti registrati in fase di pagamento;
- **RF12:** Il sistema dovrà gestire un meccanismo di autenticazione e di accesso all' area riservata distinta per l'amministrazione e per i clienti registrati;
- **RF13:** Il sistema dovrà eseguire il backup nelle ore di chiusura;
- **RF14:** Il sistema dovrà consentire al generico cliente esclusivamente l'accesso ai propri dati.
- **RF15:** Il sistema dovrà consentire all'amministratore l'accesso, sia in lettura che in scrittura, a tutti i dati memorizzati nel database.
- **RF16:** il sistema dovrà garantire al generico cliente la ricerca degli eventi organizzati;
- **RF17:** il sistema dovrà garantire che i prodotti usati in consegna siano dati da clienti registrati;
- **RF18:** il sistema dovrà essere in grado di creare sconti per i clienti registrati al sistema;
- **RF19:** il sistema dovrà gestire il pagamento con le relative modalità, da parte del cliente e dall'amministrazione;
- **RF20:** il sistema dovrà gestire l'assegnamento delle offerte;
- **RF21:** il sistema dovrà garantire che alla registrazione del cliente si crei automaticamente una tessera di livello 0;

### *Requisiti non funzionali*

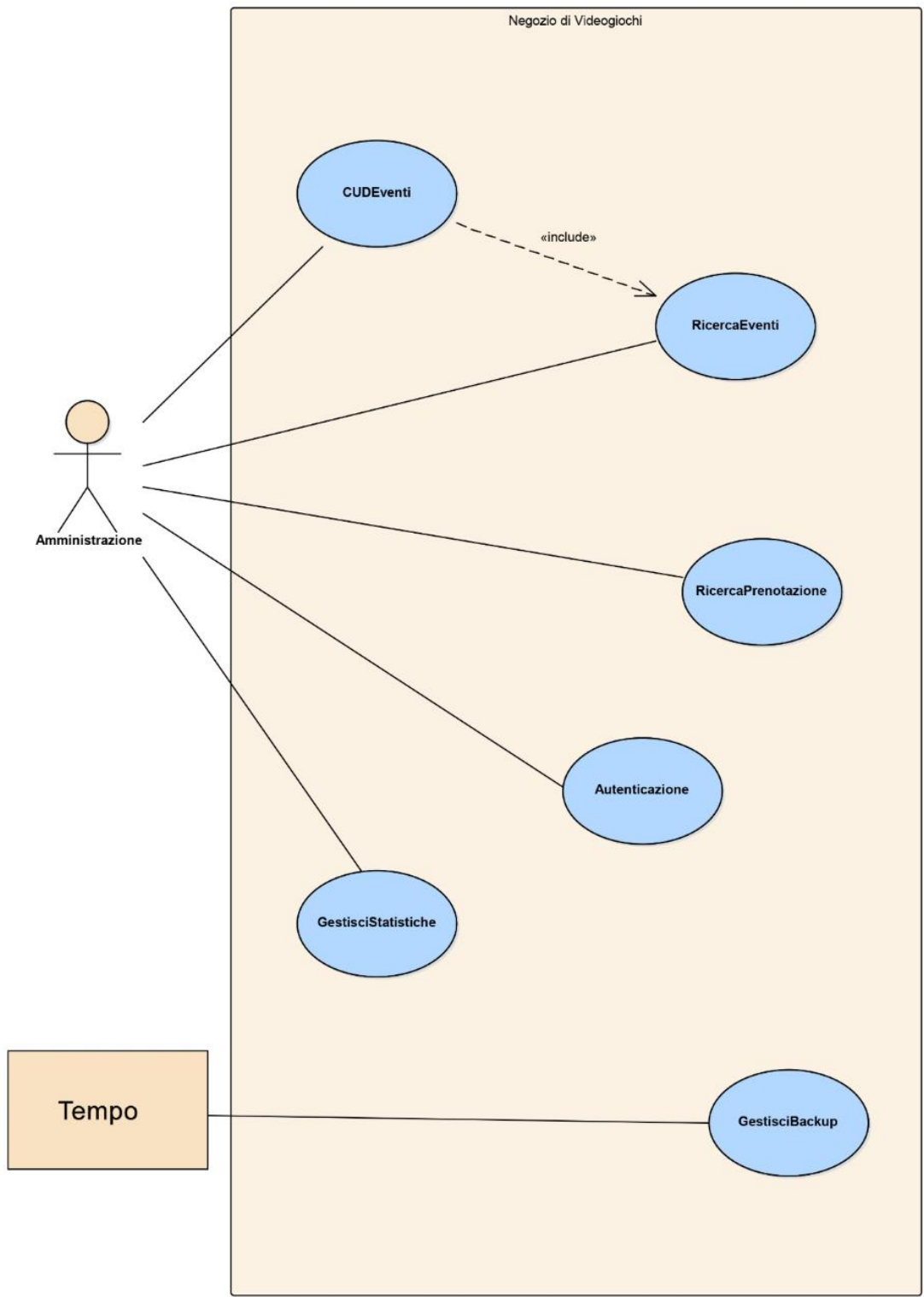
- **RNF1:** Il sistema dovrà essere realizzato in tecnologia Java per il web.
- **RNF2:** tutte le tecnologie come applicativi e software utilizzati per implementare il sistema devono essere open source;
- **RNF3:** Il sistema dovrà possedere una password di 8 caratteri associata a ciascun utente, contenenti almeno due caratteri tra numerico, maiuscole e minuscole;

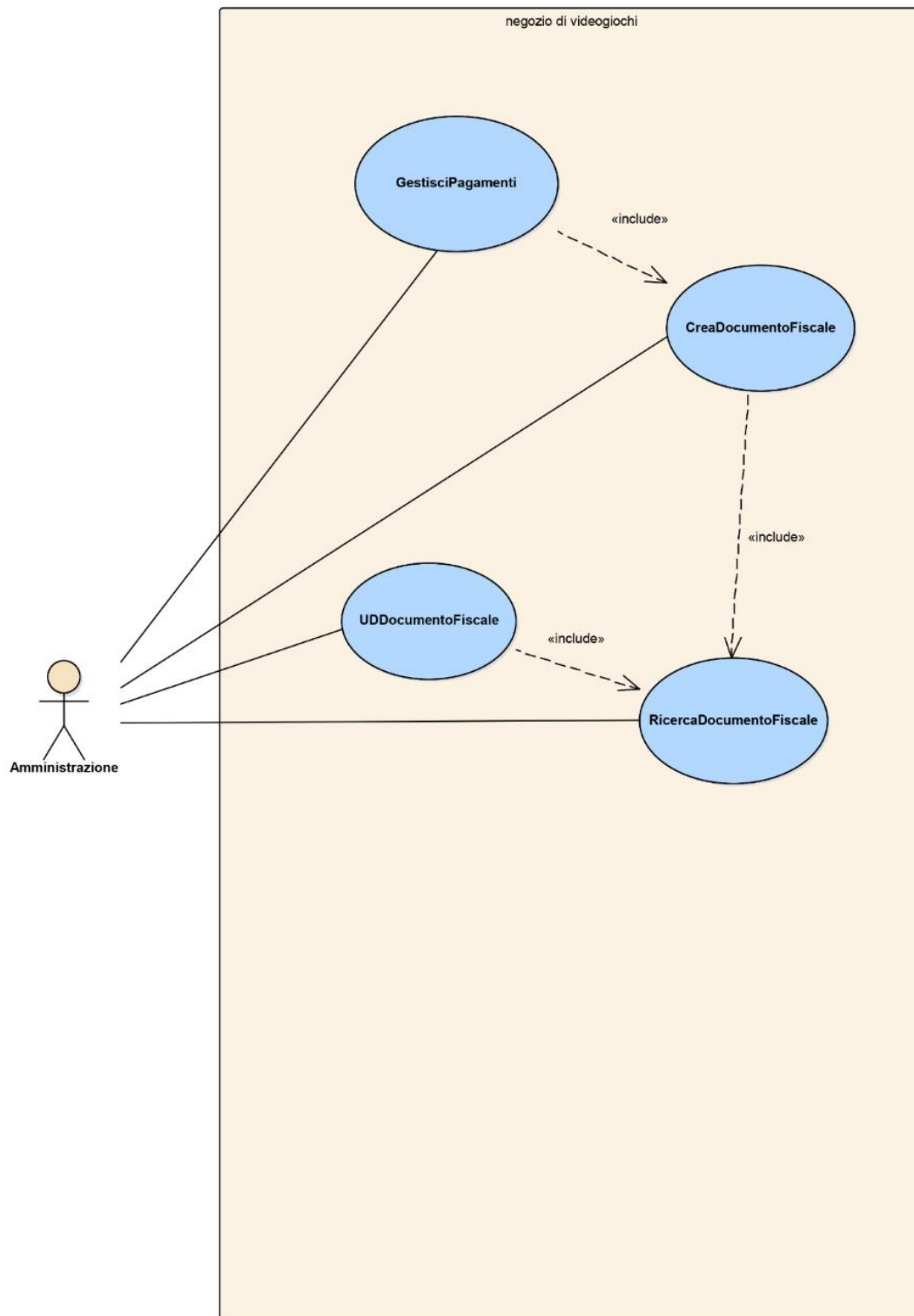


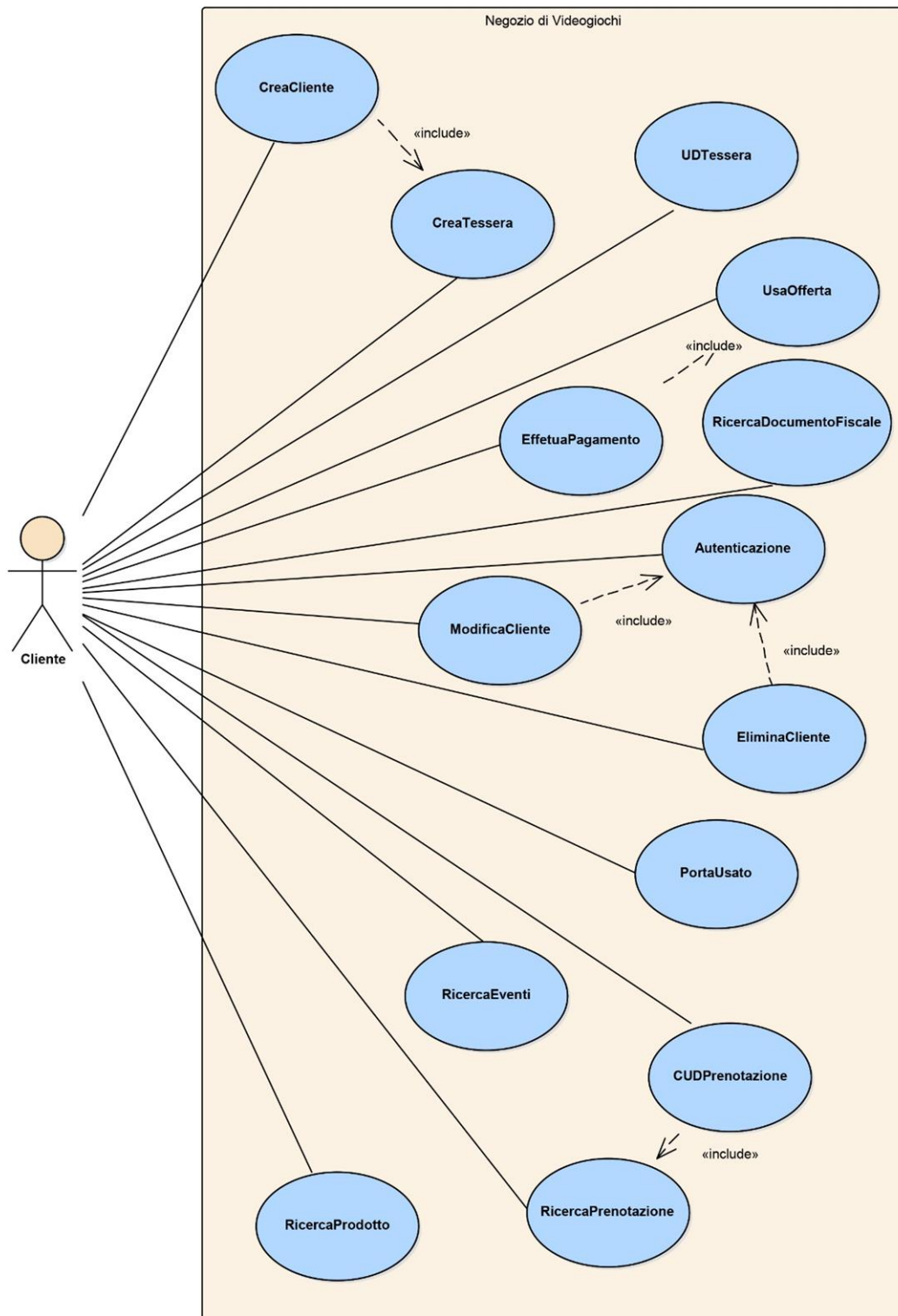
<b><i>Glossario dei termini</i></b>	<b><i>Descrizione</i></b>	<b><i>Sinonimi</i></b>
<i>Cliente</i>	è una persona fisica che effettua l'acquisto dal negozio	-----
<i>Amministrazione</i>	è il singolo o il gruppo di persone che gestisce il conseguimento del fine aziendale	Titolare, Amministratore
<i>Prodotto</i>	Sono i giochi, gadget, le console nuove e usate che il negozio rivende	Articolo
<i>Software House</i>	È un'azienda specializzata principalmente nella produzione di software e applicazioni	-----
<i>Documento Fiscale</i>	Documento rilasciato al cliente che include l'importo e la data di corretto pagamento	Ricevuta, Fattura
<i>News</i>	Notizie riguardanti l'azienda come: Eventi, uscite videogiochi, vip, ecc.; pubblicate nella home page del sito	Notizie
<i>Offerte</i>	Sconti, promozioni su alcuni tipi di prodotti forniti dal negozio stesso	Sconti, Promozioni

# Diagramma dei Casi d'uso









### *Individuazione dei Casi D'uso*

Nella costruzione di un sistema la prima cosa da fare è stabilire quali siano i suoi confini, a partire da requisiti funzionali possiamo distinguere:

Attori	Casi d'uso
✓ Cliente	○ <b>CU1:</b> RicercaCliente;
✓ Amministrazione	○ <b>CU2:</b> CUDProdotti;
✓ Tempo	○ <b>CU3:</b> RicercaProdotti;
	○ <b>CU4:</b> CUDSoftwareHouse;
	○ <b>CU5:</b> RicercaSoftwareHouse;
	○ <b>CU6:</b> CUDEventi;
	○ <b>CU7:</b> RicercaEventi;
	○ <b>CU8:</b> CreaOfferte;
	○ <b>CU9:</b> GestisciStatistiche;
	○ <b>CU10:</b> CreaCliente;
	○ <b>CU11:</b> Autenticazione;
	○ <b>CU12:</b> ModificaCliente;
	○ <b>CU13:</b> EliminaCliente;
	○ <b>CU14:</b> PortaUsato;
	○ <b>CU15:</b> CUDPrenotazione;
	○ <b>CU16:</b> RicercaPrenotazione;
	○ <b>CU17:</b> UsaOfferte;
	○ <b>CU18:</b> AssegnaOfferte;
	○ <b>CU19:</b> EffetuaPagamenti;
	○ <b>CU20:</b> GestisciBackup;
	○ <b>CU21:</b> GestisciPagamenti;
	○ <b>CU22:</b> CreaDocumentoFiscale;
	○ <b>CU23:</b> UDDocumentoFiscale;
	○ <b>CU24:</b> RicercaDocumentoFiscale;
	○ <b>CU25:</b> UDTessera
	○ <b>CU26:</b> CreaTessera
	○ <b>CU27:</b> RicercaTessera

## *Descrizione dei Casi d'uso*

---

### **Caso d'uso: RicercaCliente**

---

**ID:**

CU1

**Descrizione:**

*Questo caso permette la ricerca dei dati di un determinato cliente*

**Attori primari:**

*Amministrazione*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti di accesso*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia quando l'attore primario vuole eseguire la **Ricerca** di un cliente nel sistema;*
- 2. L'attore primario specifica una chiave, come nome e cognome o ID ad esempio;*
- 3. Il sistema effettua la ricerca del cliente nel database in base alle chiavi specificate;*
- 4. if il cliente non è presente nel sistema:  
4.1 Il sistema visualizza un opportuno messaggio di errore;*
- 5. else Il sistema visualizza i dati del cliente.*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---



---

## Caso d'uso: CUDProdotti

---

### ID:

CU2

### Descrizione:

Questo caso d'uso consente l'inserimento, la rimozione e l'aggiornamento dei dati riguardanti un prodotto.

### Attori primari:

Amministrazione

### Attori secondari:

Nessuno

### Precondizioni:

L'attore primario deve disporre di un account con appropriati diritti di accesso

### Sequenza degli eventi principale:

1. Il caso d'uso inizia quando l'attore vuole eseguire operazioni CRUD sui dati relativi ai prodotti;
2. **if** l'attore primario vuole **inserire** un nuovo prodotto:
  - 2.1. l'attore primario inserisce i dati relativi al nuovo prodotto;
  - 2.2. *Include(RicercaProdotto)*;
  - 2.3. **if** il prodotto appena descritto è già presente:
    - 2.3.1. il sistema visualizza un opportuno messaggio di errore;
  - 2.4. **else** il sistema memorizza i dati nel database;
3. **else if** l'attore primario vuole **rimuovere** i dati di un prodotto:
  - 3.1. l'attore inserisce i dati relativi al prodotto;
  - 3.2. *Include(RicercaProdotto)*;
  - 3.3. **if** il prodotto appena descritto è stato trovato:
    - 3.3.1. il sistema verifica i diritti di accesso per la rimozione;
    - 3.3.2. il sistema procede con la rimozione;
  - 3.4. **else** il sistema visualizza un opportuno messaggio di errore;
4. **else if** l'attore primario vuole **aggiornare** i dati relativi ad un prodotto:
  - 4.1. l'attore inserisce i dati relativi al prodotto;
  - 4.2. *Include(RicercaProdotto)*;
  - 4.3. **if** il prodotto appena descritto viene trovato:
    - 4.3.1. l'attore fornisce tutte le nuove informazioni relative al prodotto trovato;
    - 4.3.2. il sistema aggiorna i dati relativi al prodotto;
  - 4.4. **else** il sistema visualizza un opportuno messaggio di errore.

### Postcondizioni:

Nessuno

### Sequenza degli eventi alternative:

Nessuno

---

---

## Caso d'uso: RicercaProdotto

---

**ID:**

CU3

**Descrizione:**

*Questo caso d'uso consente la ricerca dei dati relativi ad un prodotto.*

**Attori primari:**

*Amministrazione, Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

1. *Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni di ricerca di un prodotto;*
2. *l'attore primario specifica una chiave di ricerca;*
3. *il sistema ricerca il prodotto all'interno del database;*
  - 3.1. **if** *il prodotto appena descritto non è presente nel database:*
    - 3.1.1. *il sistema visualizza un opportuno messaggio di errore;*
    - 3.2. *else il sistema visualizza tutte le informazioni riguardanti il prodotto.*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: CUDSoftwareHouse

---

### **ID:**

CU4

### **Descrizione:**

Questo caso d'uso consente l'inserimento, la rimozione e l'aggiornamento dei dati riguardanti una Software House

### **Attori primari:**

Amministrazione

### **Attori secondari:**

Nessuno

### **Precondizioni:**

L'attore primario deve disporre di un account con appropriati diritti di accesso

### **Sequenza degli eventi principale:**

1. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni CUD sui dati relativi alle Software House;
2. **if** l'attore primario vuole **inserire** dati relativi alla nuova Software House:
  - 2.1. l'attore primario inserisce i dati relativi alla nuova Software House;
  - 2.2. *Include*(RicercaSoftwareHouse);
  - 2.3. **if** la Software House appena descritta è già presente:
    - 2.3.1. il sistema visualizza un opportuno messaggio di errore;
  - 2.4. **else** il sistema memorizza i dati nel database;
3. **else if** l'attore primario vuole **rimuovere** i dati di una Software House:
  - 3.1. l'attore primario inserisce i dati relativi alla Software House;
  - 3.2. *Include*(RicercaSoftwareHouse);
  - 3.3. **if** la Software House appena descritto è stata trovata:
    - 3.3.1. il sistema verifica i diritti di accesso per la rimozione;
    - 3.3.2. il sistema procede con la rimozione;
  - 3.4. **else** il sistema visualizza un opportuno messaggio di errore;
4. **else if** l'attore primario vuole **aggiornare** i dati relativi ad una Software House:
  - 4.1. l'attore inserisce i dati relativi alla Software House;
  - 4.2. *Include*(RicercaSoftwareHouse);
  - 4.3. **if** la Software House appena descritta è stata trovata:
    - 4.3.1. l'attore fornisce tutte le nuove informazioni relative alla Software House;
    - 4.3.2. il sistema aggiorna i dati relativi alla Software House;
  - 4.4. **else** il sistema visualizza un opportuno messaggio di errore;

### **Postcondizioni:**

Nessuno

### **Sequenza degli eventi alternative:**

---

---

Nessuno

---

---

### **Caso d'uso: RicercaSoftwareHouse**

---

**ID:**

CU5

**Descrizione:**

*Questo caso d'uso consente la ricerca dei dati relativi ad una Software House*

**Attori primari:**

*Amministrazione, Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni di ricerca di una Software House;*
- 2. l'attore primario specifica una chiave di ricerca;*
- 3. il sistema ricerca la Software House all'interno del database;*
  - 3.1. **if** la Software House appena descritto non è presente nel database:*
    - 3.1.1. il sistema visualizza un opportuno messaggio di errore;*
    - 3.2. **else** il sistema visualizza tutte le informazioni riguardanti la Software House.*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: CUDEventi

---

### ID:

CU6

### Descrizione:

Questo caso d'uso consente l'inserimento, la rimozione e l'aggiornamento dei dati riguardanti un evento

### Attori primari:

Amministrazione

### Attori secondari:

Nessuno

### Precondizioni:

L'attore primario deve disporre di un account con appropriati diritti d'accesso

### Sequenza degli eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni CUD sui dati relativi alle Software House;
2. *if* l'attore primario vuole **inserire** dati relativi al nuovo evento:
  - 2.1. l'attore primario inserisce i dati relativi al nuovo evento;
  - 2.2. *Include*(RicercaEventi);
  - 2.3. *if* l'evento appena descritto è già presente:
    - 2.3.1. il sistema visualizza un opportuno messaggio di errore;
  - 2.4. *else* il sistema memorizza i dati nel database;
3. *else if* l'attore primario vuole **rimuovere** i dati ad un evento:
  - 3.1. l'attore primario inserisce i dati relativi all' evento;
  - 3.2. *Include*(RicercaEventi);
  - 3.3. *if* l'evento appena descritto è stata trovata:
    - 3.3.1. il sistema verifica i diritti di accesso per la rimozione;
    - 3.3.2. il sistema procede con la rimozione;
  - 3.4. *else* il sistema visualizza un opportuno messaggio di errore;
4. *else if* l'attore primario vuole **aggiornare** i dati relativi ad un evento:
  - 4.1. l'attore inserisce i dati relativi all' evento;
  - 4.2. *Include*(RicercaEventi);
  - 4.3. *if* l'evento appena descritto è stata trovata:
    - 4.3.1. l'attore fornisce tutte le nuove informazioni relative all'evento;
    - 4.3.2. il sistema aggiorna i dati relativi all'evento;
  - 4.4. *else* il sistema visualizza un opportuno messaggio di errore;

### Postcondizioni:

Nessuno

### Sequenza degli eventi alternative:

Nessuno

---

---

## Caso d'uso: RicercaEvento

---

**ID:**

CU7

**Descrizione:**

*Questo caso d'uso consente la ricerca dei dati relativi ad un evento*

**Attori primari:**

*Amministrazione, Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni di **ricerca** di un' evento;*
- 2. l'attore primario specifica una chiave di ricerca;*
- 3. il sistema ricerca la l'evento all'interno del database;*
  - 3.1. **if** l'evento appena descritto non è presente nel database:*
    - 3.1.1. il sistema visualizza un opportuno messaggio di errore;*
    - 3.2. **else:** il sistema visualizza tutte le informazioni riguardanti l'evento;*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: CreaOfferte

---

**ID:**

CU8

**Descrizione:**

*il caso d'uso consente la creazione di offerte da assegnare ai clienti*

**Attori primari:**

*Amministrazione*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia quando l'attore primario desidera **creare** un offerta;*
- 2. L'attore primario inserisce i dati relativi l'offerta;*
- 3. Viene creata l'offerta con i dati inseriti dall'attore primario;*
- 4. Il sistema memorizza i dati relativi all'offerta nel database;*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: GestisciStatistiche

---

**ID:**

CU9

**Descrizione:**

*Questo caso d'uso consente di effettuare statistiche di rilevante interesse per l'amministrazione al fine di indagini di mercato ecc.*

**Attori primari:**

*Amministrazione*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

1. *Il caso d'uso inizia quando l'attore primario richiede al sistema di eseguire alcune statistiche;*
2. *if l'attore primario richiede statistiche sui Clienti:*
  - 2.1. *Il sistema preleva i dati sui clienti;*
  - 2.2. *Il sistema elabora i dati prelevati;*
  - 2.3. *Il sistema visualizza i dati elaborati;*
3. *if l'attore primario richiede statistiche sugli Eventi:*
  - 3.1. *il sistema preleva i dati sugli eventi;*
  - 3.2. *il sistema elabora i dati prelevati;*
  - 3.3. *il sistema visualizza i dati elaborati;*
4. *if l'attore primario richiede statistiche sui Prodotti:*
  - 4.1. *il sistema preleva i dati sui prodotti;*
  - 4.2. *il sistema elabora i dati prelevati;*
  - 4.3. *il sistema visualizza i dati elaborati;*
5. *if l'attore primario richiede statistiche sulle Prenotazioni:*
  - 5.1. *il sistema preleva i dati sulle prenotazioni;*
  - 5.2. *il sistema elabora i dati prelevati;*
  - 5.3. *il sistema visualizza i dati elaborati;*

**Postcondizioni:**

*L'amministrazione ottiene le statistiche desiderate*

**Sequenza degli eventi alternative:**

*Nessuno*

---



---

## Caso d'uso: CreaCliente

---

**ID:**

CU10

**Descrizione:**

*Questo caso d'uso consente la creazione dell'account personale da parte del cliente*

**Attori primari:**

*Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario non deve già essere registrato*

**Sequenza degli eventi principale:**

- 1. il caso d'uso inizia quando un nuovo cliente vuole eseguire l'operazione di **creazione** del proprio account;*
- 2. l'attore primario inserisce i propri dati;*
- 3. il sistema memorizza i dati;*
- 4. **Include**(CreaTessera);*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: Autenticazione

---

**ID:**

CU11

**Descrizione:**

Questo caso d'uso consente l'accesso all'area riservate di amministrazione e clienti

**Attori primari:**

*Amministrazione, Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. il caso d'uso inizia quando l'attore primario vuole effettuare l'accesso al sistema;*
- 2. l'attore primario specifica il nome utente e password;*
- 3. il sistema ricerca nel database le chiavi sopra specificate;*
- 4. if i dati sono presenti nel database:*
  - 4.1.il sistema autentica l'attore primario;*
- 5. else il sistema visualizza un opportuno messaggio di errore;*

**Postcondizioni:**

*L'attore primario è stato autenticato dal sistema*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: ModificaCliente

---

**ID:**

CU12

**Descrizione:**

*Questo caso d'uso consente ad un cliente di modificare i propri dati*

**Attori primari:**

*Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

1. *il caso d'uso inizia quando l'attore primario vuole modificare i propri dati:*
  2. *if l'attore primario vuole modificare il proprio nome:*
    - 2.1. *l'attore inserisce il nuovo nome;*
    - 2.2. *if è già presente nel sistema:*
      - 2.2.1. *il sistema visualizza un opportuno messaggio di errore;*
    - 2.3. *else il sistema procede con la memorizzazione del nuovo nome;*
  3. *if l'attore primario vuole modificare il proprio cognome:*
    - 3.1. *l'attore inserisce il nuovo cognome;*
    - 3.2. *if è già presente nel sistema:*
      - 3.2.1. *il sistema visualizza un opportuno messaggio di errore;*
  4. *else il sistema procede con la memorizzazione del nuovo cognome;*
  5. *if l'attore primario vuole modificare la propria mail:*
    - 5.1. *l'attore inserisce la nuova mail;*
    - 5.2. *if è già presente nel sistema:*
      - 5.2.1. *il sistema visualizza un opportuno messaggio di errore;*
  6. *else il sistema procede con la memorizzazione della nuova mail;*
  7. *if l'attore primario vuole modificare la propria password:*
    - 7.1. *l'attore inserisce la nuova password;*
    - 7.2. *if la password non rispetta le specifiche:*
      - 7.2.1. *il sistema visualizza un opportuno messaggio di errore;*
  8. *else il sistema procede con la memorizzazione della nuova password;*
  9. *if l'attore primario vuole modificare il proprio codice fiscale:*
    - 9.1. *l'attore primario inserisce il nuovo codice fiscale;*
    - 9.2. *il sistema procede con la memorizzazione del nuovo codice fiscale;*
  10. *if l'attore primario vuole modificare il proprio CAP:*
    - 10.1. *l'attore primario inserisce il nuovo CAP;*
    - 10.2. *il sistema procede con la memorizzazione del nuovo CAP;*
  11. *if l'attore primario vuole modificare il proprio civico:*
    - 11.1. *l'attore primario inserisce il nuovo civico;*
-

- 
- 11.2. *il sistema procede con la memorizzazione del nuovo civico;*
  - 12. *if l'attore primario vuole modificare la propria città:*
    - 12.1. *l'attore primario inserisce la nuova città;*
    - 12.2. *il sistema procede con la memorizzazione della nuova città;*
  - 13. *if l'attore primario vuole modificare la propria via:*
    - 13.1. *l'attore primario inserisce la nuova via;*
    - 13.2. *il sistema procede con la memorizzazione della nuova via;*

**Postcondizioni:**

*I dati del cliente sono stati modificati*

**Sequenza degli eventi alternative:**

*Nessuno*

---

**Caso d'uso: EliminaCliente**

**ID:**

*CU13*

**Descrizione:**

*Questo caso d'uso consente ad un cliente la rimozione del proprio account*

**Attori primari:**

*Cliente,*

**Attori secondari:**

*Amministrazione*

**Precondizioni:**

1. *L'attore primario e l'attore secondario devono disporre di un account con appropriati diritti d'accesso;*
2. *L'attore secondario deve disporre di adeguate motivazioni per la rimozione;*

**Sequenza degli eventi principale:**

**Attore primario:**

1. *il caso d'uso inizia quando l'attore primario vuole rimuovere il proprio account cliente dal sistema*
2. *;il sistema procede con la rimozione cliente;*

**Attore secondario:**

1. *il caso d'uso inizia quando l'attore secondario vuole rimuovere un account Cliente;*
2. *Include(RicercaCliente);*
3. *If il cliente è presente:*
  - 3.1. *Il sistema elimina il cliente;*
4. *else: Il sistema visualizza un adeguato messaggio di errore;*

**Postcondizioni:**

*I dati del cliente e il suo profilo sono stati rimossi dal database*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: PortaUsato

---

**ID:**

CU14

**Descrizione:**

*Questo caso d'uso consente di gestire gli usati dei clienti registrati*

**Attori primari:**

*Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. il caso d'uso inizia quando l'attore primario porta il proprio prodotto usato;*
- 2. il sistema verifica il livello d'integrità;*
- 3. if livello d'integrità è uguale o superiore allo standard:*
  - 3.1. il sistema memorizza i dati relativi al prodotto usato;*
- 4. else il sistema visualizza un opportuno messaggio di errore;*

**Postcondizioni:**

*I dati relativi al prodotto usato sono stati memorizzati nel database*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: CUDPrenotazioni

---

### ID:

CU15

### Descrizione:

Questo caso d'uso consente l'inserimento, la rimozione e l'aggiornamento dei dati relativi le prenotazioni

### Attori primari:

Clienti

### Attori secondari:

Nessuno

### Precondizioni:

L'attore primario deve disporre di un account con appropriati diritti d'accesso

### Sequenza degli eventi principale:

1. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni CUD sui dati relativi alle prenotazioni
2. *if* l'attore primario vuole **inserire** dati relativi alla nuova prenotazione:
  - 2.1. l'attore primario inserisce i dati relativi alla nuova prenotazione;
  - 2.2. *Include*(RicercaPrenotazioni);
  - 2.3. *if* la prenotazione appena descritta è già presente:
    - 2.3.1. il sistema visualizza un opportuno messaggio di errore;
  - 2.4. *else* il sistema memorizza i dati nel database;
3. *else if* l'attore primario vuole **rimuovere** i dati di una prenotazione:
  - 3.1. l'attore primario inserisce i dati relativi alla prenotazione;
  - 3.2. *Include*(RicercaPrenotazioni);
  - 3.3. *if* la prenotazione appena descritto è stata trovata:
    - 3.3.1. il sistema verifica i diritti di accesso per la rimozione;
    - 3.3.2. il sistema procede con la rimozione;
  - 3.4. *else* il sistema visualizza un opportuno messaggio di errore;
4. *else if* l'attore primario vuole **aggiornare** i dati relativi ad una prenotazione:
  - 4.1. l'attore inserisce i dati relativi alla prenotazione;
  - 4.2. *Include*(RicercaPrenotazioni);
  - 4.3. *if* la prenotazione appena descritta è stata trovata:
    - 4.3.1. l'attore fornisce tutte le nuove informazioni relative alla prenotazione;
    - 4.3.2. il sistema aggiorna i dati relativi alla prenotazione;
  - 4.4. *else* il sistema visualizza un opportuno messaggio di errore;

### Postcondizioni:

Nessuno

### Sequenza degli eventi alternative:

Nessuno

---

---

## Caso d'uso: RicercaPrenotazioni

---

**ID:**

CU16

**Descrizione:**

*Questo caso d'uso consente la ricerca di una specifica prenotazione*

**Attori primari:**

*Amministrazione, Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 4. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni di ricerca di una prenotazione;*
- 5. l'attore primario specifica una chiave di ricerca;*
- 6. il sistema ricerca la prenotazione all'interno del database;*
  - 6.1. **if** la prenotazione appena descritto non è presente nel database:*
    - 6.1.1. il sistema visualizza un opportuno messaggio di errore;*
    - 6.2. **else** il sistema visualizza tutte le informazioni riguardanti la prenotazione.*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: UsaOfferte

---

**ID:**

CU17

**Descrizione:**

*Questo caso d'uso consente al cliente registrato di usufruire di una offerta a lui concessa*

**Attori primari:**

*Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

- 1. L'attore primario deve disporre di un account con appropriati diritti d'accesso;*
- 2. L'attore primario deve essere in possesso di un codice offerta relativo ad una offerta;*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia quando l'attore primario vuole usare una offerta;*
- 2. L'attore primario sceglie di usare una offerta durante il pagamento di un prodotto inserendo il codice offerta ad essa relativa;*
- 3. Il sistema verifica la correttezza del codice offerta inserito dall'attore primario;*
- 4. If codice offerta corretto:*
  - 4.1. Il sistema memorizza il dato nel database e riduce il totale del costo del prodotto di una quantità pari al valore dell'offerta;*
- 5. else il sistema visualizza un opportuno messaggio di errore;*

**Postcondizioni:**

*Se il cliente ha usato il codice offerta, nel saldo del documento fiscale si terrà conto del buono utilizzato;*

**Sequenza degli eventi alternative:**

*Nessuno*

---



---

## Caso d'uso: AssegnaOfferte

---

**ID:**

CU18

**Descrizione:**

*Questo caso d'uso consente l'assegnamento delle offerte ai clienti*

**Attori primari:**

*Amministrazione*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

1. *il caso d'uso inizia quando l'attore primario desidera assegnare una offerta ad un cliente;*
2. *include(RicercaTessera);*
3. *if l'attore primario vuole assegna un offerta ad una tessera specifico:*
  - 3.1. *l'attore primario specifica una chiave di ricerca;*
  - 3.2. *if la tessera è stato trovata:*
    - 3.2.1. *Include (CreaOfferta);*
    - 3.2.2. *il sistema il sistema stampa il codice offerta e tutti i dati relativi;*
  - 3.3. *else il sistema visualizza un opportuno messaggio di errore;*
4. *else if l'attore primario vuole assegnare un offerta ad un cliente a caso:*
  - 4.1. *include (CreaOfferta);*
  - 4.2. *include(RicercaTessera);*
  - 4.3. *il sistema sceglie la tessera senza nessuna politica di scelta;*
  - 4.4. *il sistema stampa il codice offerta e tutti i dati relativi;*

**Postcondizioni:**

*Nessuna*

**Sequenza degli eventi alternative:**

*Nessuna*

---

---

## Caso d'uso: EffetuaPagamenti

---

**ID:**

CU19

**Descrizione:**

*Questo caso d'uso consente di effettuare il pagamento di uno o più prodotti*

**Attori primari:**

*Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. il caso d'uso inizia quando l'attore primario vuole effettuare il pagamento di uno o più prodotti;*
- 2. l'attore primario sceglie la modalità di pagamento tra PayPal, carta di credito, pagamento in contrassegno;*
- 3. l'attore primario può scegliere se usare o meno una offerta assegnata in precedenza;*
- 4. l'attore primario inserisce i dati richiesti dal sistema in base alla scelta fatta;*
- 5. il sistema verifica la correttezza dei dati;*
- 6. **if** la verifica ha esito positivo:*
  - 6.1. il sistema invia un riepilogo dei dati al cliente rilasciando una copia del Documento Fiscale via email ;*
- 7. **else** il sistema visualizza un opportuno messaggio di errore;*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: GestisciBackup

---

**ID:**

CU20

**Descrizione:**

*Questo caso d'uso consente di effettuare periodicamente il backup di tutti i dati del sistema*

**Attori primari:**

*Tempo*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*Nessuna*

**Sequenza degli eventi principale:**

- 1. il caso d'uso inizia quando l'attore primario è programmato per effettuare il salvataggio dei dati;*
- 2. if il giorno corrente è il giorno della settimana previsto per il backup:*
  - 2.1. l'attore primario ricerca tutte le informazioni relative agli ultimi 7 giorni;*
  - 2.2. l'attore primario memorizza i dati nel terminale;*
  - 2.3. il sistema stampa a video un messaggio di avvenuto backup;*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: GestisciPagamenti

---

**ID:**

CU21

**Descrizione:**

*Questo caso d'uso consente di gestire uno o più pagamenti*

**Attori primari:**

*Amministrazione*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia l'attore primario vuole effettuare il pagamento di uno o più prodotti;*
- 2. l'attore primario inserisce i dati del pagamento;*
- 3. il sistema verifica la correttezza dei dati;*
- 4. if la verifica ha esito positivo:*
  - 4.1. il sistema memorizza i dati relativi al pagamento;*
  - 4.2. Include (CreaDocumentoFiscale);*
  - 4.3. l'attore primario inserisce i dati relativi al cliente;*
  - 4.4. Il sistema verifica la correttezza dei dati;*
  - 4.5. if la verifica ha esito positivo:*
    - 4.5.1. il sistema rilascia al cliente ed all'attore primario la copia del documento fiscale;*
  - 4.6. else il sistema visualizza un opportuno messaggio di errore;*
- 5. else il sistema visualizza un opportuno messaggio di errore;*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: CreaDocumentoFiscale

---

**ID:**

CU22

**Descrizione:**

*Questo caso d'uso consente all'attore primario di inserire un nuovo Documento Fiscale*

**Attori primari:**

*Amministrazione*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia quando l'attore primario vuole **creare** un nuovo Documento Fiscale;*
- 2. l'attore primario inserisce i dati relativi ad un nuovo Documento Fiscale;*
- 3. **Include** (RicercaDocumentoFiscale);*
- 4. **if** il Documento Fiscale descritto è già presente:
  - 4.1. il sistema visualizza un opportuno messaggio di errore;**
- 5. **else** il sistema memorizza i dati relativi al Documento Fiscale;*

**Postcondizioni:**

*Nessuna*

**Sequenza degli eventi alternative:**

*Nessuna*

---

---

## Caso d'uso: UDDocumentoFiscale

---

### **ID:**

CU23

### **Descrizione:**

*Questo caso d'uso consente la rimozione e l'aggiornamento dei dati riguardanti un Documento Fiscale*

### **Attori primari:**

*Amministrazione*

### **Attori secondari:**

*Nessuno*

### **Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

### **Sequenza degli eventi principale:**

- 1.** *Il caso d'uso inizia quando l'attore primario vuole effettuare operazioni **UD** sui dati relativi ad un Documento Fiscale;*
- 2.** *if l'attore primario vuole **aggiornare** i dati relativi ad un Documento Fiscale:*
  - 2.1.** *l'attore inserisce i dati relativi al Documento Fiscale;*
  - 2.2.** *Include (RicercaDocumentoFiscale);*
  - 2.3.** *if il Documento Fiscale appena descritto è stato trovato:*
    - 2.3.1.** *l'attore fornisce tutte le nuove informazioni relative al Documento Fiscale;*
    - 2.3.2.** *il sistema aggiorna i dati relativi al Documento Fiscale;*
  - 2.4.** *else il sistema visualizza un opportuno messaggio di errore;*
- 3.** *else if l'attore primario vuole **rimuovere** i dati di un Documento Fiscale:*
  - 3.1.** *l'attore primario inserisce i dati relativi al Documento Fiscale;*
  - 3.2.** *Include (RicercaDocumentoFiscale);*
  - 3.3.** *if il Documento Fiscale appena descritto è stato trovato:*
    - 3.3.1.** *il sistema verifica i diritti di accesso per la rimozione;*
    - 3.3.2.** *il sistema procede con la rimozione;*
  - 3.4.** *else il sistema visualizza un opportuno messaggio di errore;*

### **Postcondizioni:**

*Nessuno*

### **Sequenza degli eventi alternative:**

*Nessuna*

---

---

## Caso d'uso: RicercaDocumentoFiscale

---

**ID:**

CU24

**Descrizione:**

*Questo caso d'uso consente la ricerca dei dati relativi ad un Documento Fiscale.*

**Attori primari:**

*Amministrazione, Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni di **ricerca** di un Documento Fiscale;*
- 2. l'attore primario specifica una chiave di ricerca;*
- 3. il sistema ricerca il Documento Fiscale all'interno del database;*
  - 3.1. **if** il Documento Fiscale appena descritto non è presente nel database:*
    - 3.1.1. il sistema visualizza un opportuno messaggio di errore;*
  - 3.2. **else** il sistema visualizza tutte le informazioni riguardanti il Documento Fiscale.*

**Postcondizioni:**

*Nessuna*

**Sequenza degli eventi alternative:**

*Nessuna*

---

---

## Caso d'uso: UDTessera

---

### **ID:**

CU25

### **Descrizione:**

Questo caso d'uso consente la rimozione, e l'aggiornamento dei dati riguardanti una tessera.

### **Attori primari:**

Cliente

### **Attori secondari:**

Amministrazione

### **Precondizioni:**

L'attore primario e secondario deve disporre di un account con appropriati diritti d'accesso;

L'attore secondario deve disporre di adeguata motivazione per la rimozione di una tessera

### **Sequenza degli eventi principale:**

#### **Attore primario:**

1. il caso d'uso inizia quando l'attore primario vuole eseguire operazioni UD sulla propria tessera;
3. **if** l'attore primario vuole rimuovere i dati relativi alla propria tessera dal sistema:
  - 3.1. il sistema procede con la rimozione della tessera;
4. **else** l'attore primario vuole aggiornare i dati relativi alla propria tessera dal sistema:
  - 4.1. il sistema procede con l'aggiornamento della tessera;

#### **Attore secondario:**

1. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni UD sui dati relativi alle tessere;
  2. **if** l'attore primario vuole rimuovere i dati di una tessera:
    - 2.1. l'attore primario inserisce i dati relativi alla tessera;
    - 2.2. *Include*(RicercaTessera);
    - 2.3. **if** la tessera appena descritto è stata trovata:
      - 2.3.1. il sistema verifica i diritti di accesso per la rimozione;
      - 2.3.2. il sistema procede con la rimozione;
    - 2.4. **else** il sistema visualizza un opportuno messaggio di errore;
  3. **else if** l'attore primario vuole aggiornare i dati relativi ad una tessera:
    - 3.1. l'attore inserisce i dati relativi alla tessera;
    - 3.2. *Include*(RicercaTessera);
    - 3.3. **if** la tessera appena descritta è stata trovata:
      - 3.3.1. l'attore fornisce tutte le nuove informazioni relative alla tessera;
      - 3.3.2. il sistema aggiorna i dati relativi alla tessera;
-



---

3.4. *else il sistema visualizza un opportuno messaggio di errore;*

**Postcondizioni:**

*Nessuna*

**Sequenza degli eventi alternative:**

*Nessuna*

---

---

**Caso d'uso: CreaTessera**

**ID:**

CU26

**Descrizione:**

*Questo caso d'uso consente la creazione di una tessera da parte del cliente una volta registrato*

**Attori primari:**

*Cliente*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con adeguati diritti di accesso;*

**Sequenza degli eventi principale:**

5. *il caso d'uso inizia quando un nuovo cliente vuole eseguire l'operazione di **creazione** della propria tessera;*
6. *l'attore primario inserisce i dati richiesti;*
  - 6.1. *il sistema crea una tessere di livello 0;*
  - 6.2. *il sistema memorizza i dati;*

**Postcondizioni:**

*Nessuno*

**Sequenza degli eventi alternative:**

*Nessuno*

---

---

## Caso d'uso: Ricerca Tessera

---

**ID:**

CU27

**Descrizione:**

*Questo caso d'uso consente la ricerca di una tessera.*

**Attori primari:**

*Amministrazione*

**Attori secondari:**

*Nessuno*

**Precondizioni:**

*L'attore primario deve disporre di un account con appropriati diritti d'accesso*

**Sequenza degli eventi principale:**

- 1. Il caso d'uso inizia quando l'attore primario vuole eseguire operazioni di **ricerca** di una tessera;*
- 2. l'attore primario specifica una chiave di ricerca;*
- 3. il sistema ricerca all'interno del database;*
  - 3.1. **if** la tessera appena descritto non è presente nel database:*
    - 3.1.1. il sistema visualizza un opportuno messaggio di errore;*
    - 3.2. **else** il sistema visualizza tutte le informazioni riguardanti la tessera.*

**Postcondizioni:**

*Nessuna*

**Sequenza degli eventi alternative:**

*Nessuna*

---

# Progettazione Concettuale

La Progettazione Concettuale di una Base di Dati significa individuare gli oggetti (o entità) che la costituiscono e le relazioni (o operazioni o associazioni) tra un oggetto e l'altro.

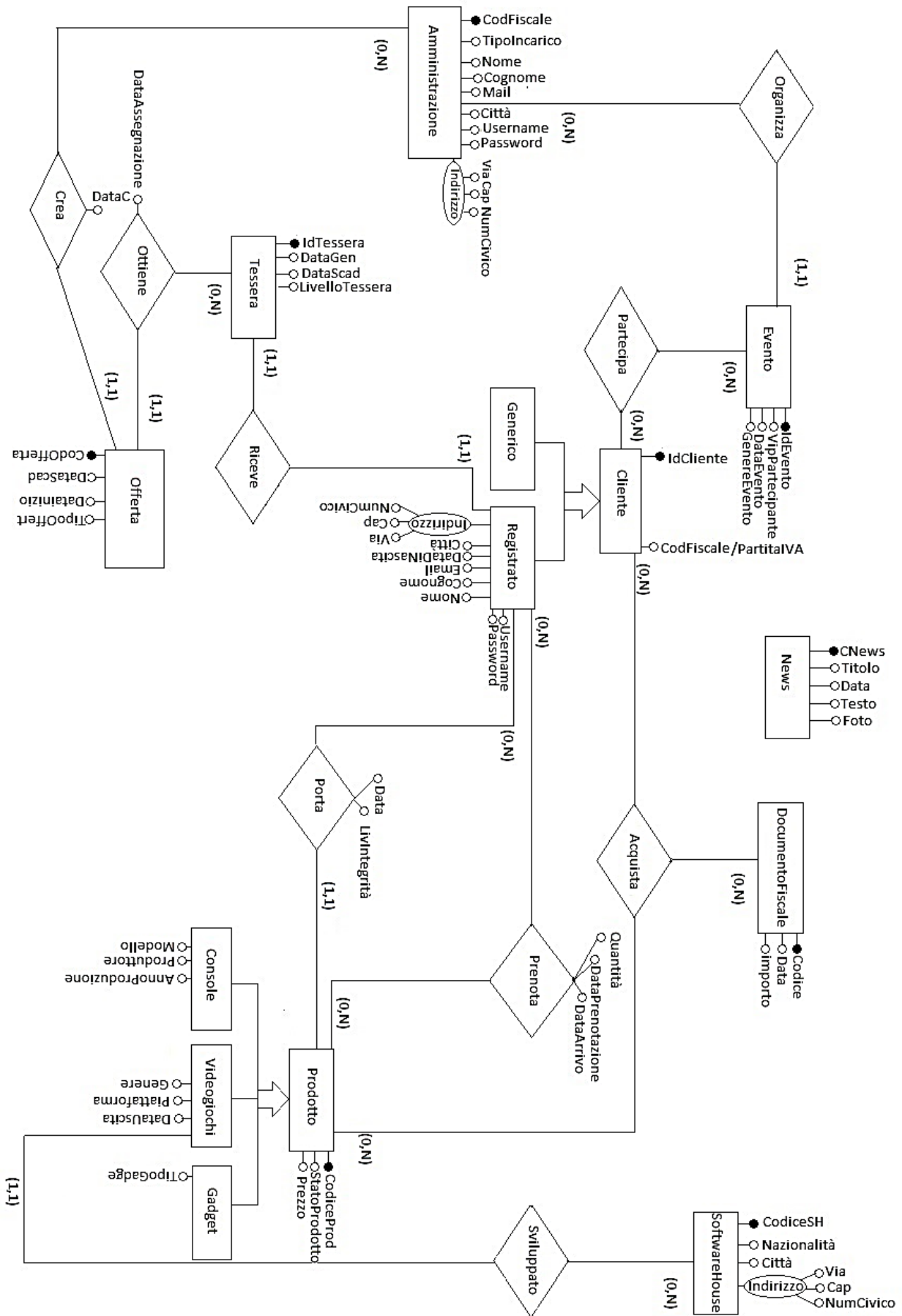
Si tratta del livello più alto della progettazione di un Database, quello più vicino all'uomo e più lontano dalla macchina (hardware). Deve quindi essere realizzata con strumenti e linguaggi comprensibili a tutti, non solo agli specialisti, e indipendenti dal sistema di data-base.

Il Modello Concettuale su cui si Basa la nostra Progettazione è il Modello Entità-Relazione ( E/R )

In questa fase cercheremo di costruire di uno schema in grado di descrivere al meglio la realtà di interesse

Definiamo adesso lo schema concettuale del sistema informativo, per la gestione di un Negozio di Videogiochi(GameStop).

# Diagramma Entità-Relazione



## *Dizionario delle Entità*

Descriviamo le Entità che sono presenti nello schema E/R:

<b>Entità</b>	<b>Descrizione</b>	<b>Attributi</b>	<b>Identificatore</b>
Cliente	Colui che effettua acquisti nel negozio; può possedere una tessera fedeltà con la quale può portare i prodotti usati.	IdCliente, CodFiscale/PartitaIVA.	IdCliente
Generico	Sottocategoria di cliente; cliente in mancanza di tessera fedeltà.	IdCliente, CodFiscale.	IdCliente
Registrato	Sottocategoria di cliente; il cliente possiede una tessera fedeltà con la quale può portare l'usato, avere accesso ad offerte esclusive;	IdCliente, CodFiscale, Nome, Cognome, Email, Username, Password, DataDiNascita, Città, Indirizzo.	IdCliente
Prodotto	Ciò che viene esposto e messo in vendita nel negozio.	CodiceProd, StatoProdotto, Prezzo.	CodiceProd
Console	Sottocategoria di prodotto.	CodiceProd, StatoProdotto, Prezzo, AnnoProduzione, Produttore, Modello.	CodiceProd
Videogiochi	Sottocategoria di prodotto	CodiceProd, StatoProdotto, Prezzo, Piattaforma, Genere, DataUscita.	CodiceProd
Gadget	Sottocategoria di prodotto	CodiceProd, StatoProdotto, Prezzo, TipoGadget.	CodiceProd
Evento	Evento pubblico con presenza di vip del mondo di videogiochi organizzato dal negozio.	IdEvento, DataEvento, GenereEvento, VipPartecipante.	IdEvento

SoftwareHouse	È un'azienda specializzata principalmente nella produzione di software e applicazioni.	CodiceSH, Nazionalità, Città, Indirizzo.	CodiceSH
Offerta	Servizio di sconto, supervalutazioni e inviti speciali forniti dal negozio.	CodOfferta, TipoOfferta, DataInizio, DataScad.	CodOfferta
Amministrazione	Persona fisica che amministra il sistema e gestisce il negozio.	CodiceFiscale, TipoIncarico, Nome, Cognome, Username, Password, Email, Città, Indirizzo.	CodiceFiscale
Tessera	Tessera fedeltà di un cliente registrato.	IdTessera, DataGen, DataScad, LivelloTessera.	IdTessera
News	Notizie riguardanti il negozio pubblicate nella homepage del sito.	CNews, Titolo, Data, Testo, Foto	Cnews
DocumentoFiscale	Documento rilasciato al cliente ad ogni acquisto	Codice, Importo, Data	Codice

### *Dizionario delle Relazioni*

Procediamo descrivendo le Relazioni

<b>Relazione</b>	<b>Descrizione</b>	<b>Entità Coinvolte</b>	<b>Attributi</b>
Acquista	Indica che ogni cliente può acquistare un certo numero di prodotti.	Cliente(o,N) Prodotto(o,N) DocumentoFiscale(o,N)	---
Prenota	Indica che il cliente con tessera fedeltà prenota una serie di prodotti.	Registrato(o,N) Prodotto(o,N)	DataPrenotazione, DataArrivo, Quantità
Porta	Indica che il cliente con tessera fedeltà porta dei prodotti usati in negozio	Registrato(o,N) Prodotto(1,1)	Data, LivelloIntegrità

	pronti per essere valutati.		
Riceve	Associa un cliente appena registrato con una tessera fedeltà unica.	Registrato(1,1) Tessera(1,1)	---
Partecipa	Associa un qualsiasi cliente ad un evento.	Evento(0,N) Cliente(0,N)	---
Organizza	Indica che l'amministrazione organizza un evento alla volta.	Amministrazione(0,N) Evento(1,1)	---
Crea	Indica che l'amministrazione genera offerte.	Amministrazione(0,N) Offerte(1,1)	DataC
Ottiene	Indica che ad ogni tessera fedeltà può essere data un offerta.	Tessera(0,N) Offerta(1,1)	DataAssegnazione
Sviluppato	Indica un qualsiasi videogioco da quale Software House è stato creato.	Prodotto(1,1) SoftwareHouse(0,N)	---

### *Dizionario dei Vincoli*

Descriviamo i vincoli

<b>Vincolo</b>	<b>Regola</b>
V1	Un acquisto per essere tale deve riferirsi ad almeno un prodotto presente in negozio
V2	Un prodotto usato di un cliente per essere aggiunto nel negozio deve avere il livello di integrità maggiore di 3
V3	Il prezzo di un prodotto deve essere maggiore di 0
V4	La quantità della prenotazioni per cliente registrato non deve superare i 4 prodotti
V5	L'offerta non si può usare dopo la data di scadenza
V5	L'offerta può essere usata solo per un acquisto alla volta
V6	L'offerta deve essere ottenuta da clienti registrati presenti nel sistema
V7	La tessera non può superare il livello 4



# Progettazione Logica

La progettazione logica costituisce la base per l'effettiva realizzazione e deve tener conto, per quanto possibile, delle sue prestazioni.

La progettazione logica si suddivide in due fasi:

- ✓ Ristrutturazione dello schema E/R
- ✓ Traduzione verso il modello logico

Quindi la progettazione Logica di una Base di Dati consiste nella traduzione dello schema concettuale dei dati in uno schema logico che rispecchia il modello dei dati scelto, cioè, nel nostro caso, il modello relazionale.

La semplificazione dello schema si rende necessaria perché non tutti i costrutti del modello E/R hanno una traduzione naturale nei modelli logici.

## *Ristrutturazione dello Schema E/R*

La fase di ristrutturazione di uno schema E/R si può suddividere in una serie di passi da effettuare in sequenza:

### **1. Analisi delle ridondanze:**

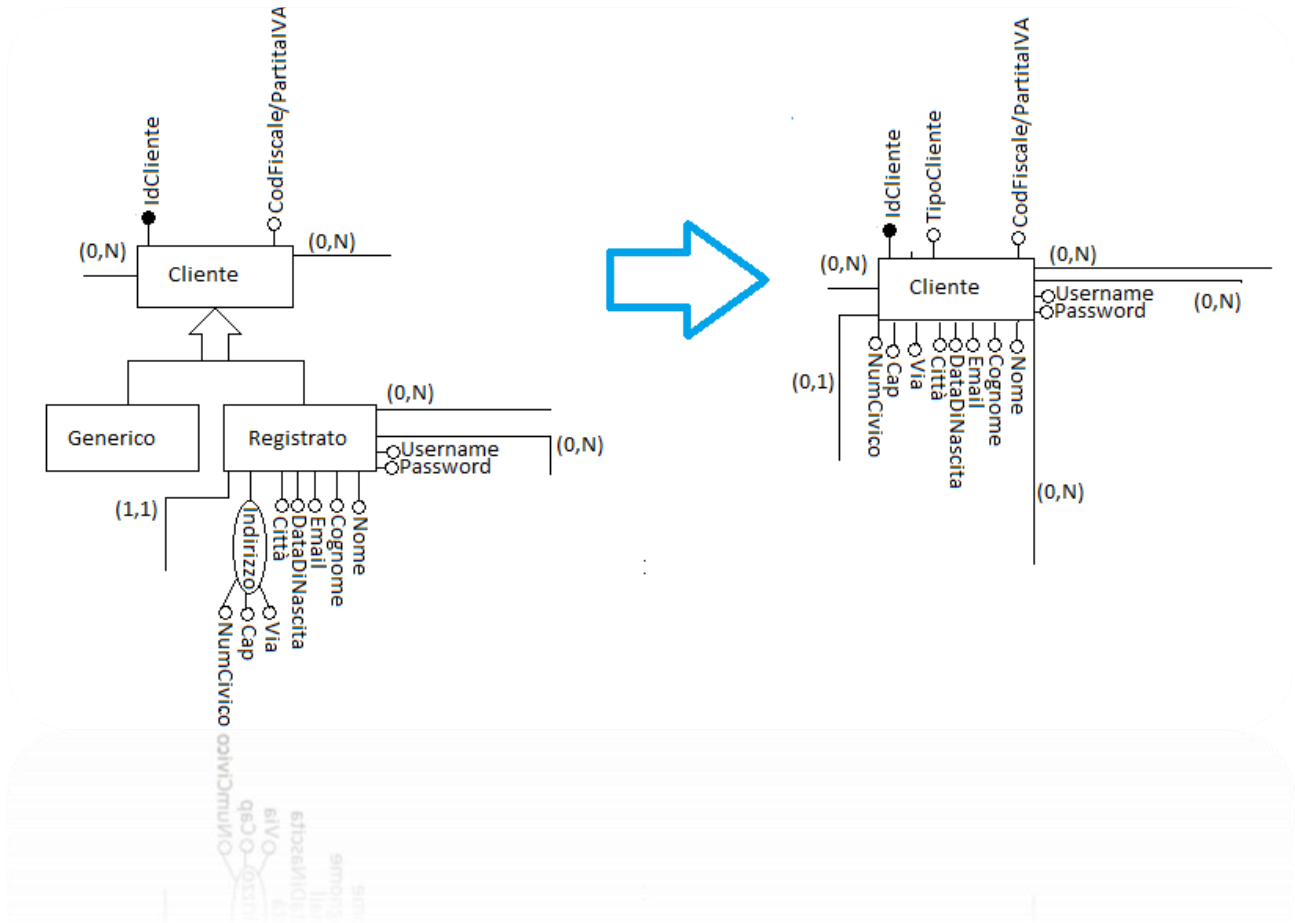
In questo caso, non essendo presente alcuna ridondanza non dobbiamo apportare alcuna modifica al modello E/R.

### **2. Eliminazione delle generalizzazioni:**

- a.** La generalizzazione tra Cliente registrato e generico è stata eliminata accorpando le entità figlie nell'entità padre ereditando tutte le associazioni e tutti gli attributi, questo comporta:
  - i.** Vi è l'aggiunta di un nuovo attributo per distinguere il tipo di una occorrenza dell'entità padre, **TipoCliente**;
  - ii.** L'entità padre eredita tutte le associazioni dei figli avendo come cardinalità minima 0;
  - iii.** L'eliminazione delle entità figlie, il padre eredita gli attributi dei figli:

Registrato: **Nome, Cognome, Email, DataDiNascita, Città, Indirizzo, Username, Password**;

b. La generalizzazione tra prodotto console, videogiochi e gadget è



stata eliminata intraprendendo un accorpamento delle entità figlie nell'entità padre, questo avviene perché l'entità padre è più importante delle entità figlie, questo comporta:

i. L'eliminazione delle entità figlie, il padre eredita gli attributi dei figli:

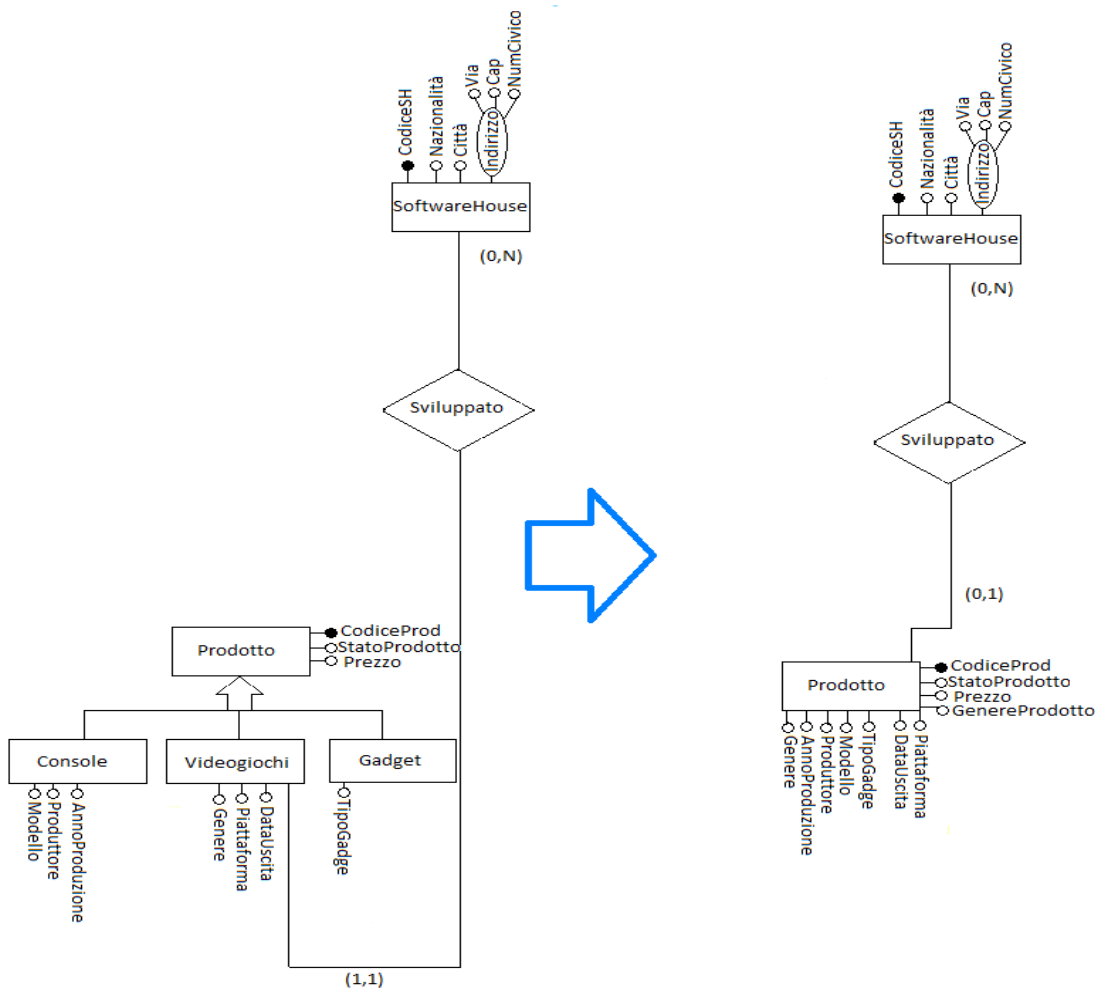
Console: **produttore, Modello, AnnoProduzione;**

Videogiochi: **DataUscita, Piattaforma, Genere;**

Gadget: **TipoGadget;**

ii. L'entità padre eredita l'associazione dei figli e mettendo come cardinalità minima 0;

iii. Vi è l'aggiunta di un nuovo attributo per distinguere il tipo di una occorrenza dell'entità padre, **GenereProdotto;**



### 3. Partizione/accorpamento di entità e associazioni:

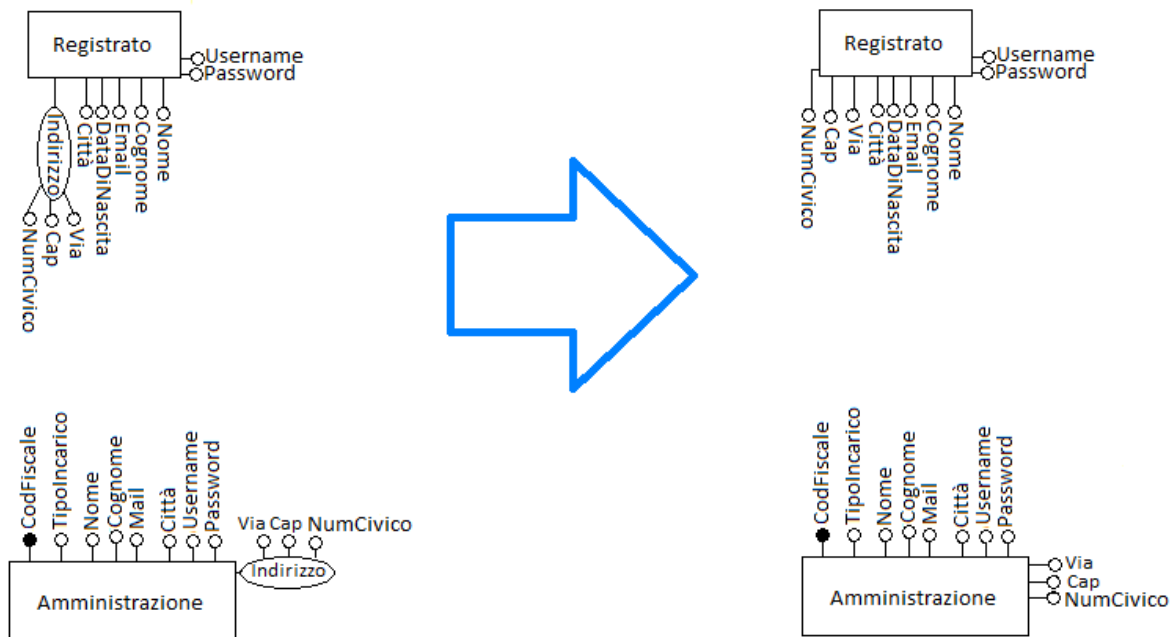
Il modello E/R realizzato non presenta concetti che debbano essere partizionati o accorpati;

### 4. Eliminazione degli attributi multivalore:

Non essendo presenti attributi multivalore non apporteremo nessuna modifica;

### 5. Eliminazione degli attributi composti:

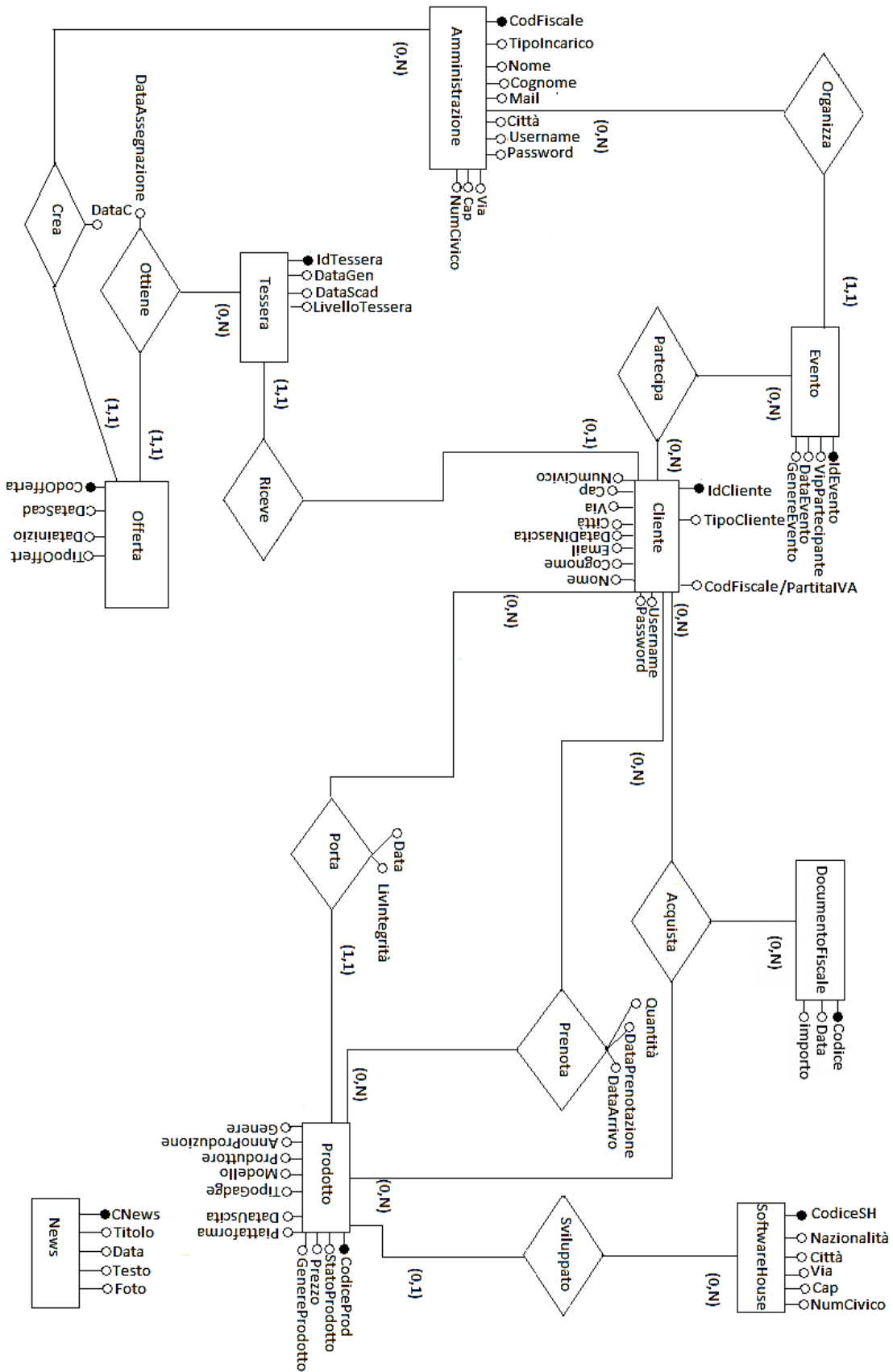
Il modello relazionale non è in grado di gestire gli attributi composti, se lo schema E/R presenta tali attributi è necessario eliminarli, è molto semplice basta sostituire all'attributo stesso gli attributi semplici che lo compongono:



## 6. Scelta degli identificatori primari:

Le chiavi primarie sono già tutte ben definite e non potranno assumere valori nulli, sono tutte chiavi primarie semplici (costituite da un solo attributo) e tutte costituite da attributi interni;

*Schema ER dopo la Ristrutturazione:*



## Traduzione verso il modello Relazionale

Adesso dobbiamo esaminare la traduzione verso il modello Relazionale, andiamo quindi a tradurre tutte le relazioni:

### I. Relazione: **Acquista**

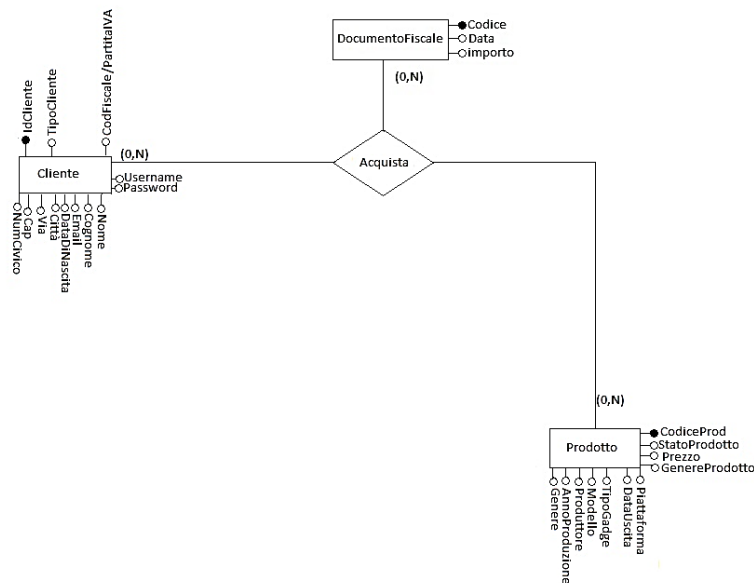
Lo schema relazione che ne deriva sarà:

**CLIENTE**(IdCliente, Username, Password, TipoCliente, Nome, Cognome, CodFiscale/PartitaIVA, Email, DataDiNascita, Città, Via, Cap, NumCivico)

**ACQUISTA**(IdCliente, CodiceProd, CodiceFattura, DataPagamento)

**PRODOTTO**(CodiceProd, StatoProdotto, Prezzo, GenereProdotto, Piattaforma, DataUscita, TipoGadget, Modello, Produttore, AnnoProduzione, Genere)

**DOCUMENTOFISCALE**(Codice, Importo, Data)



Possiamo quindi definire adesso i vincoli di integrità referenziale che coinvolgono gli attributi IdCliente, CodiceProd e CodiceFat della relazione ACQUISTA con gli attributi aventi lo stesso nome di CLIENTE, PRODOTTO e DOCUMENTOFISCALE.

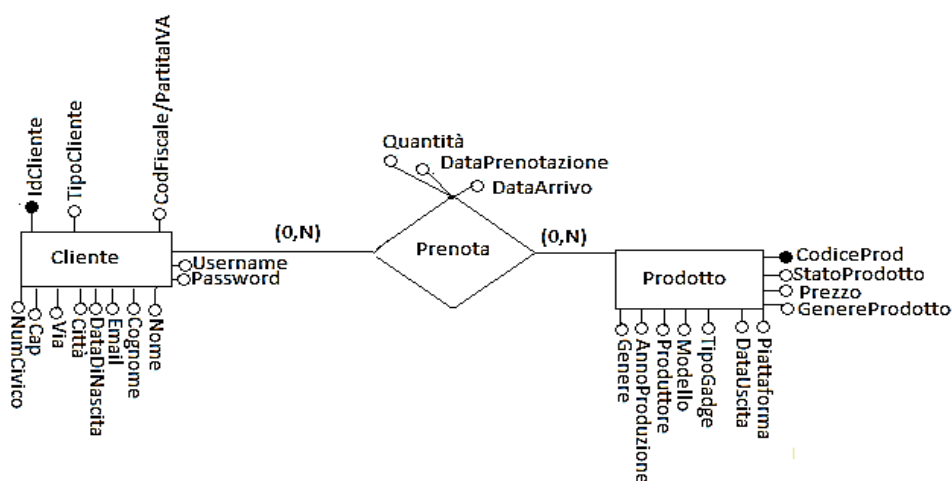
## II. Relazione: Prenota

Lo schema relazionale che ne deriva sarà:

**CLIENTE**(IdCliente, Username, Password, TipoCliente, Nome, Cognome, CodFiscale/PartitaIVA, Email, DataDiNascita, Città, Via, Cap, NumCivico)

**PRENOTA**(IdCliente, CodiceProd, Quantità, DataPrenotazione, DataArrivo)

**PRODOTTO**(CodiceProd, StatoProdotto, Prezzo, GenereProdotto, Piattaforma, DataUscita, TipoGadget, Modello, Produttore, AnnoProduzione, Genere)



Possiamo quindi definire adesso i vincoli di integrità referenziale che coinvolgono gli attributi IdCliente e CodiceProd della relazione PRENOTA con gli attributi aventi lo stesso nome di CLIENTE e PRODOTTO.

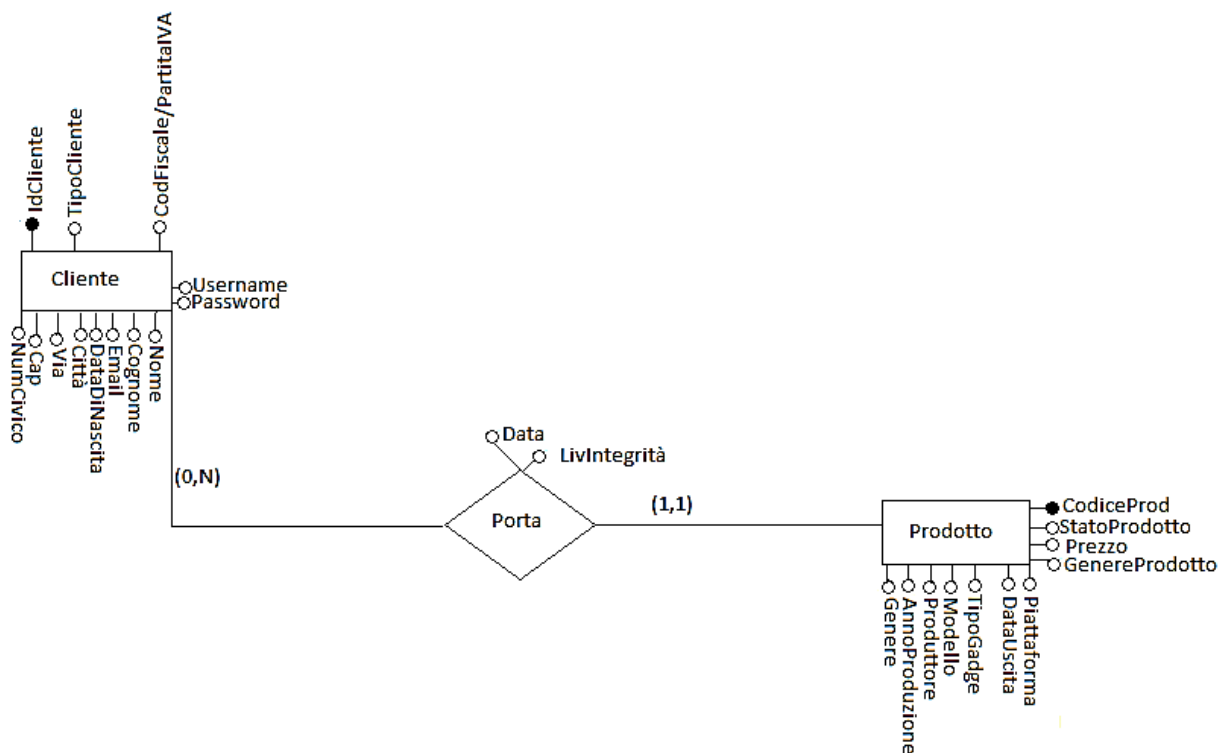
### III. Relazione: Porta

**\*Questa è una relazione uno a molti quindi applicheremo le dovute regole. (Chiave dell'entità con cardinalità massima molti, lo metto come attributo dell'entità con cardinalità massima 1)**

Lo schema relazione che ne deriva sarà:

**CLIENTE** (IdCliente, Username, Password, TipoCliente, Nome, Cognome, CodFiscale/PartitaIVA, Email, DataDiNascita, Città, Via, Cap, NumCivico)

**PRODOTTO** (CodiceProd, StatoProdotto, IdCliente, Data, LivIntegrità, Prezzo, GenereProdotto, Piattaforma, DataUscita, TipoGadget, Modello, Produttore, AnnoProduzione, Genere)



Per lo schema ottenuto esiste un vincolo di integrità referenziale tra l'attributo IdCliente della relazione CLIENTE con l'attributo avente lo stesso nome della relazione PRODOTTO.



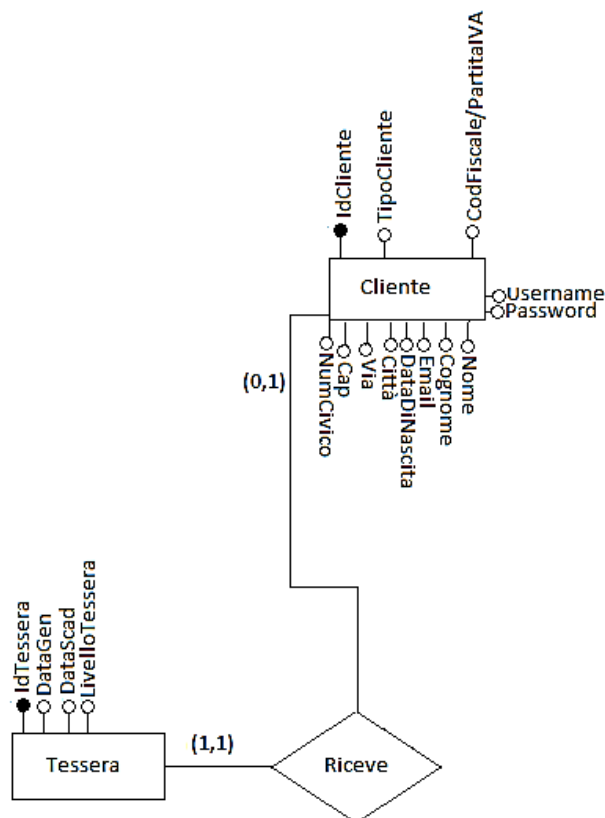
#### IV. Relazione: **Riceve**

**\*Questa è una relazione uno a uno quindi applicheremo le dovute regole. (Chiave dell'entità con cardinalità (0,1) , lo metto come attributo dell'entità con cardinalità (1,1))**

Lo schema relazionale che ne deriva sarà:

**CLIENTE**(IdCliente, Username, Password, TipoCliente, Nome, Cognome, CodFiscale/PartitaIVA, Email, DataDiNascita, Città, Via, Cap, NumCivico)

**TESSERA**(IdTessera, IdCliente, DataGen, DataScad, LivelloTessera)



Per lo schema ottenuto esiste un vincolo di integrità referenziale tra l'attributo IdCliente della relazione TESSERA con l'attributo avente lo stesso nome della relazione CLIENTE.

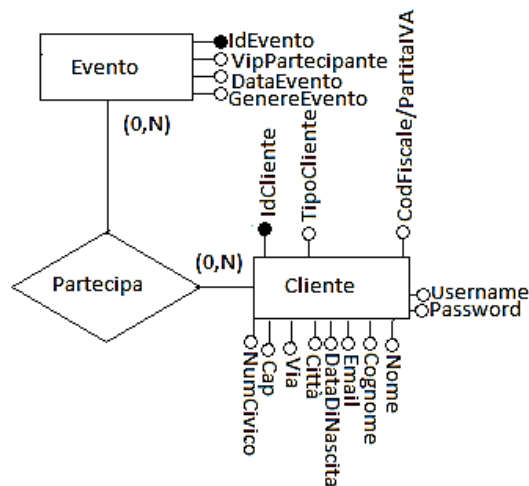
## V. **Relazione: Partecipa**

Lo schema relazionale che ne deriva sarà:

**CLIENTE**(IdCliente, Username, Password, TipoCliente, Nome, Cognome, CodFiscale/PartitaIVA, Email, DataDiNascita, Città, Via, Cap, NumCivico)

**PARTECIPA**(IdCliente, IdEvento)

**EVENTO**(IdEvento, VipPartecipante, DataEvento, GenereEvento)



Possiamo quindi definire adesso i vincoli di integrità referenziale che coinvolgono gli attributi IdCliente e IdEvento della relazione PARTECIPA con gli attributi aventi lo stesso nome di CLIENTE ed EVENTO.

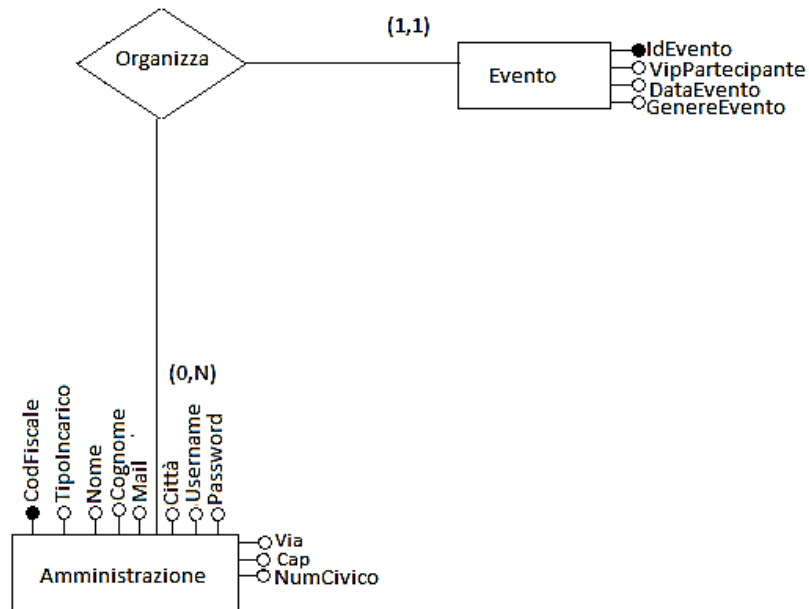
## VI. Relazione: Organizza

**\*Questa è una relazione uno a molti quindi applicheremo le dovute regole. (Chiave dell'entità con cardinalità massima molti, lo metto come attributo dell'entità con cardinalità massima 1)**

Lo schema relazionale che ne deriva sarà:

**EVENTO**(IdEvento, CodFiscale, VipPartecipante, DataEvento, GenereEvento)

**AMMINISTRAZIONE**(CodFiscale, TipoIncarico, Username, Password, Nome, Cognome, Mail, Città, Via, Cap, NumCivico)



Per lo schema ottenuto esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della relazione AMMINISTRAZIONE con l'attributo avente lo stesso nome della relazione EVENTO.

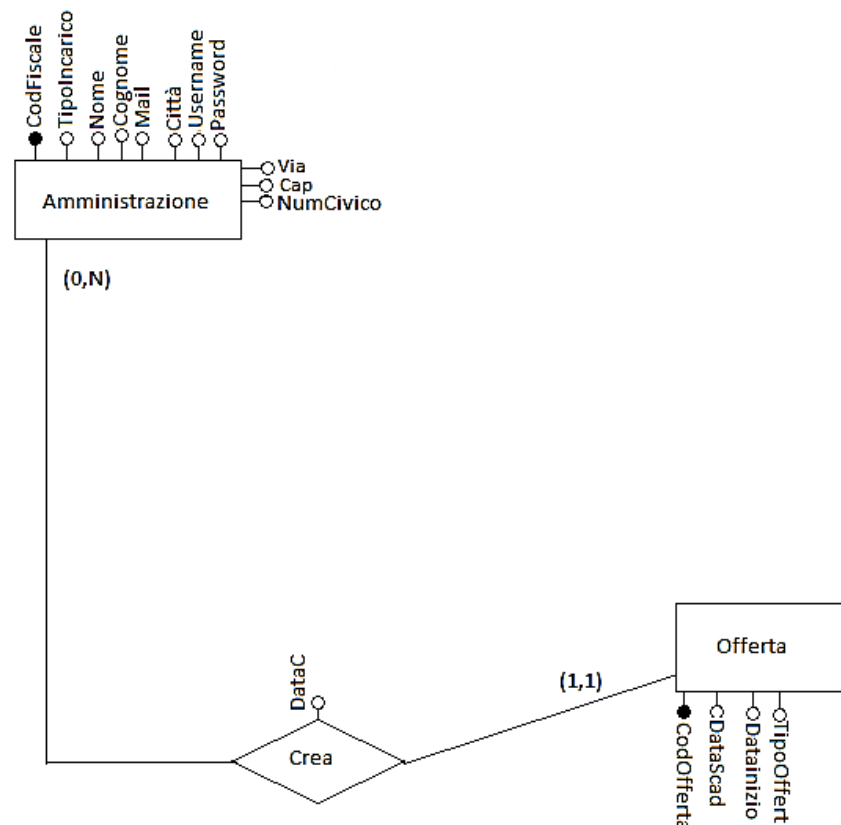
## VII. Relazione: Crea

**\*Questa è una relazione uno a molti quindi applicheremo le dovute regole. (Chiave dell'entità con cardinalità massima molti, lo metto come attributo dell'entità con cardinalità massima 1)**

Lo schema relazionale che ne deriva sarà:

**AMMINISTRAZIONE**(CodFiscale, TipoIncarico, Username, Password, Nome, Cognome, Mail, Città, Via, Cap, NumCivico)

**OFFERTA**(CodOfferta, CodFiscale, DataC, DataInizio, DataScad, TipoOfferta)



Per lo schema ottenuto esiste un vincolo di integrità referenziale tra l'attributo CodFiscale della relazione AMMINISTRAZIONE con l'attributo avente lo stesso nome della relazione OFFERTA.

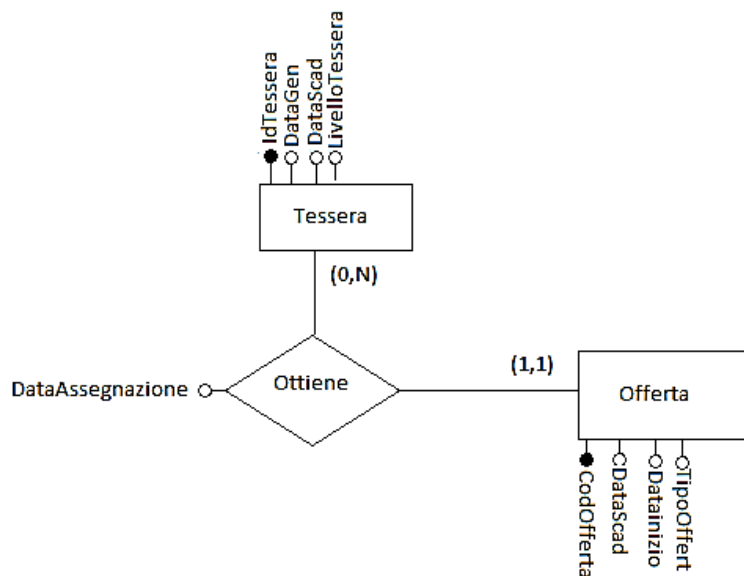
## VIII. Relazione: **Ottiene**

**\*Questa è una relazione uno a molti quindi applicheremo le dovute regole. (Chiave dell'entità con cardinalità massima molti, lo metto come attributo dell'entità con cardinalità massima 1)**

Lo schema relazionale che ne deriva sarà:

**TESSERA**(IdTessera, DataGen, DataScad, LivelloTessera)

**OFFERTA**(CodOfferta, DataInizio, DataScad, TipoOfferta, IdTessera, DataAssegnazione)



Possiamo quindi definire adesso i vincoli di integrità referenziale che coinvolge l'attributi IdTessera della relazione TESSERA con gli attributi aventi lo stesso nome della relazione OFFERTA.

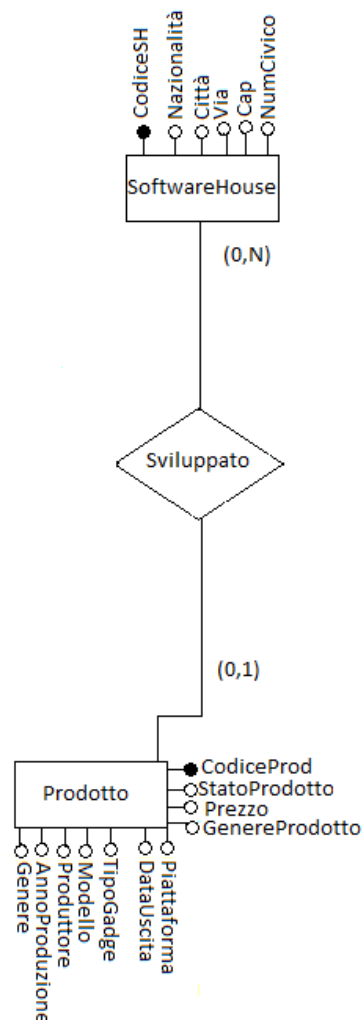
## IX. Relazione: **Sviluppato**

**\*Questa è una relazione uno a molti quindi applicheremo le dovute regole. (Chiave dell'entità con cardinalità massima molti, lo metto come attributo dell'entità con cardinalità massima 1)**

Lo schema che ne deriva sarà:

**PRODOTTO**(CodiceProd, CodiceSH, StatoProdotto, Prezzo, GenereProdotto, Piattaforma, DataUscita, TipoGadget, Modello, Produttore, AnnoProduzione, Genere)

**SOFTWAREHOUSE**(CodiceSH, Nazionalità, Città, Via, Cap, NumCivico)



Per lo schema ottenuto esiste un vincolo di integrità referenziale tra l'attributo CodiceSH della relazione SOFTWAREHOUSE con l'attributo avente lo stesso nome della relazione PRODOTTO.

## Schema Relazionale

Con le traduzioni apportate per la trasformazione otteniamo il seguente schema relazionale

<b>SCHEMA RELAZIONALE</b>
<b>PRODOTTO</b> ( <u>CodiceProd</u> , CodiceSH, StatoProdotto, IdCliente, Data, LivIntegrità, Prezzo, GenereProdotto, Piattaforma, DataUscita, TipoGadget, Modello, Produttore, AnnoProduzione, Genere)
<b>CLIENTE</b> ( <u>IdCliente</u> , Username, Password, TipoCliente, Nome, Cognome, CodFiscale/PartitaIVA, Mail, DataDiNascita, Città, Via, Cap, NumCivico)
<b>AMMINISTRAZIONE</b> ( <u>CodFiscale</u> , TipoIncarico, Username, Password, Nome, Cognome, Email, Città, Via, Cap, NumCivico)
<b>DOCUMENTOFISCALE</b> ( <u>Codice</u> , Importo, Data)
<b>ACQUISTA</b> ( <u>IdCliente</u> , <u>CodiceProd</u> , <u>CodiceFattura</u> , <u>DataPagamento</u> )
<b>TESSERA</b> ( <u>IdTessera</u> , <u>IdCliente</u> , <u>DataGen</u> , <u>DataScad</u> , <u>LivelloTessera</u> )
<b>SOFTWAREHOUSE</b> ( <u>CodiceSH</u> , Nazionalità, Città, Via, Cap, NumCivico)
<b>OFFERTA</b> ( <u>CodOfferta</u> , <u>CodFiscale</u> , <u>DataC</u> , <u>DataInizio</u> , <u>DataScad</u> , <u>TipoOfferta</u> , <u>IdTessera</u> , <u>DataAssegnazione</u> )
<b>EVENTO</b> ( <u>IdEvento</u> , <u>CodFiscale</u> , <u>VipPartecipante</u> , <u>DataEvento</u> , <u>GenereEvento</u> )
<b>PARTECIPA</b> ( <u>IdCliente</u> , <u>IdEvento</u> )
<b>PRENOTA</b> ( <u>IdCliente</u> , <u>CodiceProd</u> , <u>Quantità</u> , <u>DataPrenotazione</u> , <u>DataArrivo</u> )

## Dizionario dei vincoli di integrità referenziale

Attributo vincolato	Relazione vincolante
Prodotto.CodiceSH	SoftwareHouse
Prodotto.IdCliente	Cliente
Tessera.IdCliente	Cliente
Acquista.IdCliente	Cliente
Acquista.CodiceProd	Prodotto
Acquista.CodiceFattura	DocumentoFiscale
Offerta.IdTessera	Tessera
Offerta.CodFiscale	Amministrazione
Evento.CodFiscale	Amministrazione
Partecipa.IdCliente	Cliente
Partecipa.IdEvento	Evento
Prenota.IdCliente	Cliente
Prenota.CodiceProd	Prodotto



## Progettazione fisica

Partendo dalla base costruita attraverso la progettazione logica e quella concettuale analizziamo adesso la progettazione fisica che per noi è costituita da 2 punti fondamentali: la scelta degli indici e la stima delle richieste di spazio su disco.

### *Scelta degli indici*

<b>Tabella</b>	<b>Indice</b>
Cliente	IdCliente
Prodotto	CodiceProd
Acquista	IdCliente,CodiceProd, CodiceFat
Tessera	IdTessera
SoftwareHouse	CodiceSH
Offerta	CodOfferta
Evento	IdEvento
Partecipa	IdCliente,IdEvento
Prenota	IdCliente,CodiceProd
DocumentoFiscale	Codice
Amministrazione	CodFiscale

Per la tabella **Prodotto** può rivelarsi molto utile la creazione di un altro indice formato dal campo “GenereProdotto” questo perché ricorre spesso la ricerca di un particolare tipo di prodotto. Ma considerando i costi di aggiornamento degli indici e anche il ridotto numero di generi possiamo evitare l’aggiunta di tale indice.

### *Stime delle richieste di spazio su disco*

Supponiamo di dover dare al nostro database, dal punto di vista dello spazio necessario su disco, un'autonomia di 10 anni. Andiamo quindi adesso a fare una stima sulle singole tabelle per la determinazione dello spazio minimo al quale sommeremo un margine del 5-10% sul totale delle dimensioni espresse in byte.

**Tabella: CLIENTE**

<b>Attributo</b>	<b>Tipo di dato</b>	<b>Dimensioni</b>
IdCliente	varchar(10)	10 byte
Username	varchar(20)	20 byte
Password	varchar(8)	8 byte
Nome	varchar(25)	25 byte
Cognome	varchar(25)	25 byte
TipoCliente	varchar(15)	15 byte
CodFiscale/PartitaIVA	varchar(16)	16 byte
Email	varchar(25)	25 byte
DataDiNascita	date	10 byte
Città	varchar(20)	20 byte
Via	varchar(50)	50 byte
Cap	varchar(5)	5 byte
NumCivico	varchar(3)	3 byte
<b>Totale</b>		<b>232 byte</b>

Considerando che la provincia di Reggio conta 565.819 abitanti aggiornati alla data odierna, considerando che i possibili clienti sono di tutte le età in maggioranza dai 7 ai 35 anni, consideriamo il 60%, che di questa fetta solo il 8% non usa videogame, console o gadget, considerando del restante è stimato che andremo a conquistare in 10 anni tra il 13 e 15% del mercato ricaviamo che i possibili clienti saranno un massimo di 35.000 clienti, tale numero deriva dalla quantità di giovani sempre crescente che usano tale tipologia di prodotti ed inoltre anche il posto strategico del negozio cioè sul corso Giuseppe Garibaldi di Reggio Calabria. Di questi solo il 30% effettuerà una registrazione avendo così un totale massimo di 10.500 elementi nella tabella in 10 anni che andranno a richiedere **2.436.000 byte**.

**Tabella: PRODOTTO**

Attributo	Tipo di dato	Dimensione
CodiceProd	varchar(10)	10 byte
CodiceSH	varchar(10)	10 byte
IdCliente	varchar(10)	10 byte
Data	date	10 byte
LivIntegrità	int	4 byte
StatoProdotto	varchar(5)	5 byte
Prezzo	double	8 byte
GenereProdotto	varchar(25)	25 byte
Piattaforma	varchar(25)	25 byte
DataUscita	date	10 byte
TipoGadget	varchar(25)	25 byte
Modello	varchar(25)	25 byte
Produttore	varchar(50)	50 byte
AnnoProduzione	date	10 byte
Genere	varchar(20)	20 byte
<b>Totale</b>		<b>247 byte</b>

Si può ipotizzare che in 10 anni di servizio, essendo un rivenditore di molteplici prodotti sia nuovi che usati, verranno memorizzati più di 50.000 prodotti di tipo diverso. Così possiamo calcolare le dimensioni da riservare a questa tabella:

$50.000 * 247 \text{ byte} = \mathbf{12.350.000 \text{ byte}}$ .

**Tabella: AMMINISTRAZIONE**

Attributo	Tipo di dato	Dimensione
CodFiscale	varchar(16)	16 byte
TipoIncarico	varchar(20)	20 byte
Username	varchar(20)	20 byte
Password	char(8)	8 byte
Nome	varchar(25)	25 byte
Cognome	varchar(25)	25 byte
Mail	varchar(25)	25 byte
Città	varchar(20)	20 byte
Via	varchar(50)	50 byte
Cap	varchar(5)	5 byte
NumCivico	varchar(3)	3 byte
<b>Totale</b>		<b>217 byte</b>

La tabella amministrazione non tiene conto solo dei proprietari ma anche dei commessi, quindi in 10 anni può succedere di cambiare spesso commessi, ipotizzando che i commessi possono cambiare 1 all'anno, e che resti sempre lo

stesso proprietario. Così possiamo calcolare le dimensioni da riservare a questa tabella:

$$11 * 217 \text{ byte} = \mathbf{2.170 \text{ byte}}$$

*Tabella: ACQUISTA*

Attributo	Tipo di dato	Dimensione
IdCliente	varchar(10)	10 byte
CodiceProd	varchar(10)	10 byte
CodiceFattura	int	4 byte
DataPagamento	date	10 byte
<b>Totale</b>		<b>34 byte</b>

Si può ipotizzare che al giorno si ha una media di 10 acquisti, per 10 anni di servizio si arriva ad avere, 36.500 acquisti. Così possiamo calcolare le dimensioni da riservare a questa tabella:

$$36.500 * 34 \text{ byte} = \mathbf{1.241.000 \text{ byte.}}$$

*Tabella: DOCUMENTOFISCALE*

Attributo	Tipo di dato	Dimensione
Codice	int	4 byte
Data	date	10 byte
Importo	double	8 byte
<b>Totale</b>		<b>22 byte</b>

Si può ipotizzare che al giorno si ha una media di 10 acquisti, per 10 anni di servizio si arriva ad avere, 36.500 acquisti. Così possiamo calcolare le dimensioni da riservare a questa tabella:

$$36.500 * 22 \text{ byte} = \mathbf{803.000 \text{ byte}}$$

*Tabella: TESSERA*

Attributo	Tipo di dato	Dimensione
IdTessera	varchar(10)	10 byte
IdCliente	varchar(10)	10 byte
DataGen	date	10 byte
LivelloTessera	int	4 byte
DataScad	date	10 byte
<b>Totale</b>		<b>44 byte</b>

Si può ipotizzare che avendo 10.500 clienti registrati allora avremo 1 tessera per ogni cliente. Così possiamo calcolare le dimensioni da riservare a questa tabella:

$$10.500 * 44 \text{ byte} = \mathbf{462.000 \text{ byte.}}$$

*Tabella: SOFTWAREHOUSE*

Attributo	Tipo di dato	Dimensione
CodiceSH	varchar(10)	10 byte
Nazionalità	varchar(25)	25 byte
Città	varchar(20)	20 byte
Cap	varchar(5)	5 byte
Via	varchar(50)	50 byte
NumCivico	varchar(3)	3 byte
<b>Totale</b>		<b>113 byte</b>

Si può ipotizzare che dei 50.000 prodotti presenti in negozio il 60% sono videogiochi, quindi sviluppati da una Software House. Così possiamo calcolare le dimensioni da riservare a questa tabella:

$$30.000 * 113 \text{ byte} = \mathbf{3.390.000 \text{ byte}}$$

*Tabella: OFFERTA*

Attributo	Tipo di dato	Dimensione
CodOfferta	varchar(10)	10 byte
CodFiscale	varchar(16)	16 byte
DataC	date	10 byte
DataInizio	date	10 byte
DataScad	date	10 byte
TipoOfferta	varchar(20)	20 byte
IdTessera	varchar(10)	10 byte
DataAssegnazione	date	10 byte
<b>Totale</b>		<b>96 byte</b>

Si può ipotizzare che le offerte create in 10 anni di servizio, dall'amministrazione, per i clienti che si registrano, sono 8.000. Così possiamo calcolare le dimensioni da riservare a questa tabella:

$$8.000 * 96 \text{ byte} = \mathbf{768.000 \text{ byte}}$$

*Tabella: EVENTO*

Attributo	Tipo di dato	Dimensione
IdEvento	varchar(10)	10 byte
CodFiscale	varchar(16)	16 byte
DataEvento	date	10 byte
GenereEvento	varchar(25)	25 byte
VipPartecipante	varchar(50)	50 byte
<b>Totale</b>		<b>111 byte</b>

Si può ipotizzare che l'amministrazione organizza circa 2 eventi al mese, considerando che nei mesi invernali il numero di eventi raddoppia avremo quindi in 1 anno avremo:

12 mesi \* 2(eventi al mese) + 3\*2(eventi estivi aggiuntivi)= **30 eventi in 1 anno;**

30 eventi\* 111 byte= **3.330** \* 10 anni= **33.300 byte** è la dimensione da riservare a questa tabella.

*Tabella: PARTECIPA*

Attributo	Tipo di dato	Dimensione
IdCliente	varchar(10)	10 byte
IdEvento	varchar(10)	10 byte
Totale		20 byte

Si può ipotizzare che sui 10.500 clienti registrati, solo il 20% parteciperà agli eventi e che in media ogni cliente di tale 20 % parteciperà solo a 5 eventi su 30 eventi organizzati, così possiamo calcolare le dimensioni da riservare a questa tabella:

$[10.500 \% 20] * [5 \text{ eventi} * 10 \text{ anni}] * 2 \text{ byte} = \mathbf{2.100.000 \text{ byte.}}$

*Tabella: PRENOTA*

Attributo	Tipo di dato	Dimensione
IdCliente	varchar(10)	10 byte
CodiceProd	varchar(10)	10 byte
DataPrenotazione	date	10 byte
Quantità	int	4 byte
DataArrivo	date	10 byte
Totale		44 byte

Si può ipotizzare che dei 10.500 clienti registrati in 10 anni di servizio, il 60% effettuerà una prenotazione su un prodotto. Così possiamo calcolare le dimensioni da riservare a questa tabella:

$[10.500 \% 60] * 44 \text{ byte} = \mathbf{277.200 \text{ byte.}}$

Facciamo adesso una somma per determinare le dimensioni totali necessarie su disco:

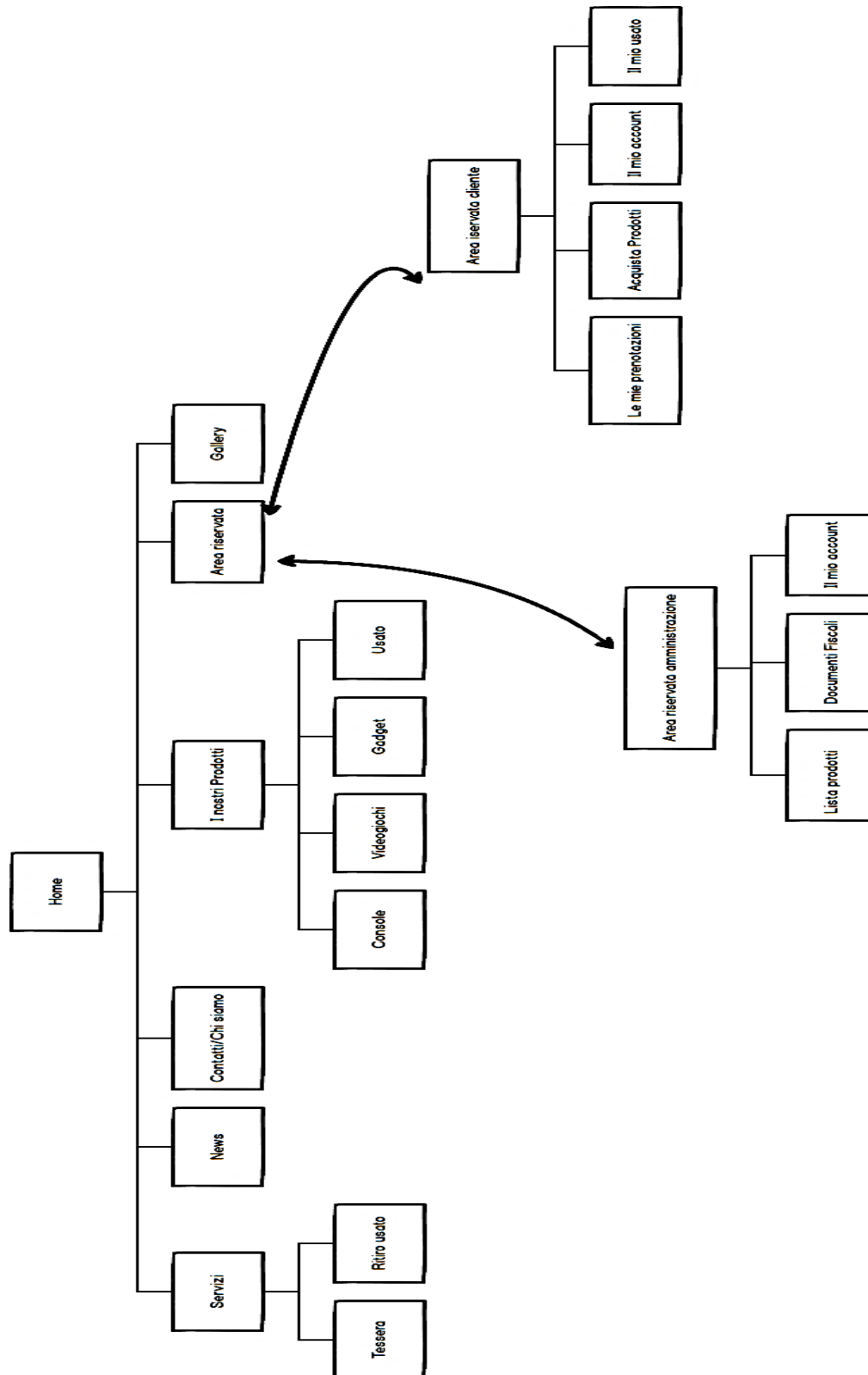
<b>Riassunto delle dimensioni totali richieste su disco:</b>	
<b>CLIENTE</b>	<b>1.436.000 byte</b>
<b>AMMINISTRAZIONE</b>	<b>2.170 byte</b>
<b>PRODOTTO</b>	<b>12.350.000 byte</b>
<b>ACQUISTA</b>	<b>1.241.000 byte</b>
<b>DOCUMENTOFISCALE</b>	<b>803.000 byte</b>
<b>TESSERA</b>	<b>462.000 byte</b>
<b>SOFTWAREHOUSE</b>	<b>3.390.000 byte</b>
<b>OFFERTA</b>	<b>768.000 byte</b>
<b>EVENTO</b>	<b>33.300 byte</b>
<b>PARTECIPA</b>	<b>2.100.000 byte</b>
<b>PRENOTA</b>	<b>277.200 byte</b>
<b>Totale</b>	<b>22.862.670 byte</b>
<b>Margine del 10 %</b>	<b>2.286.260 byte</b>
<b>TOTALE DEFINITIVO</b>	<b>25.148.930 byte</b>

Il totale in byte risulta essere 25.148.930 byte che convertito in MB che arrotonderemo a 25 MB.

# Progettazione delle applicazioni

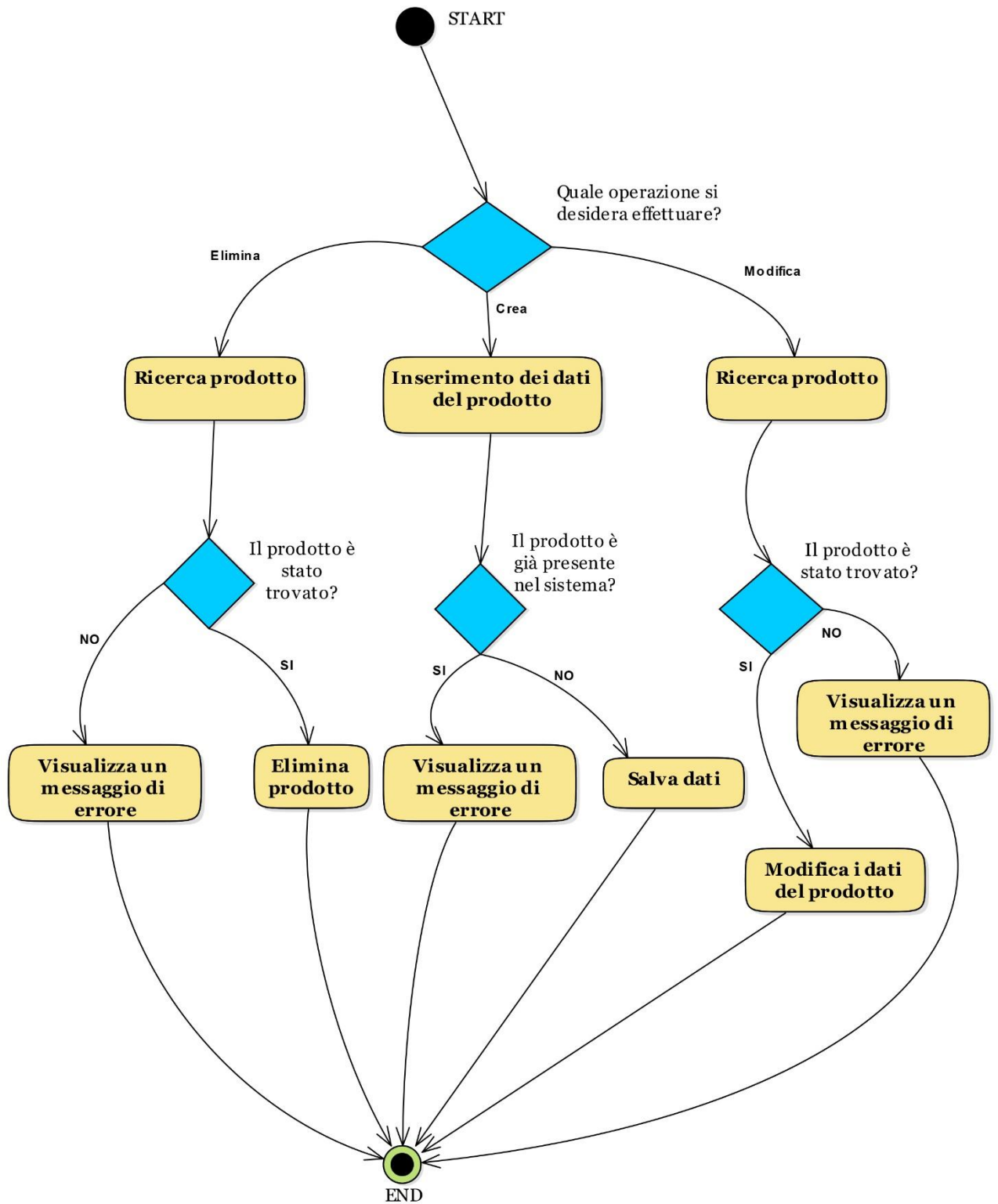
Per progettazione delle applicazioni si intende quella fase che si occupa di progettare tutti i programmi che dovranno operare sui dati relativi al sistema informativo.

✓ *Mappa del sito*

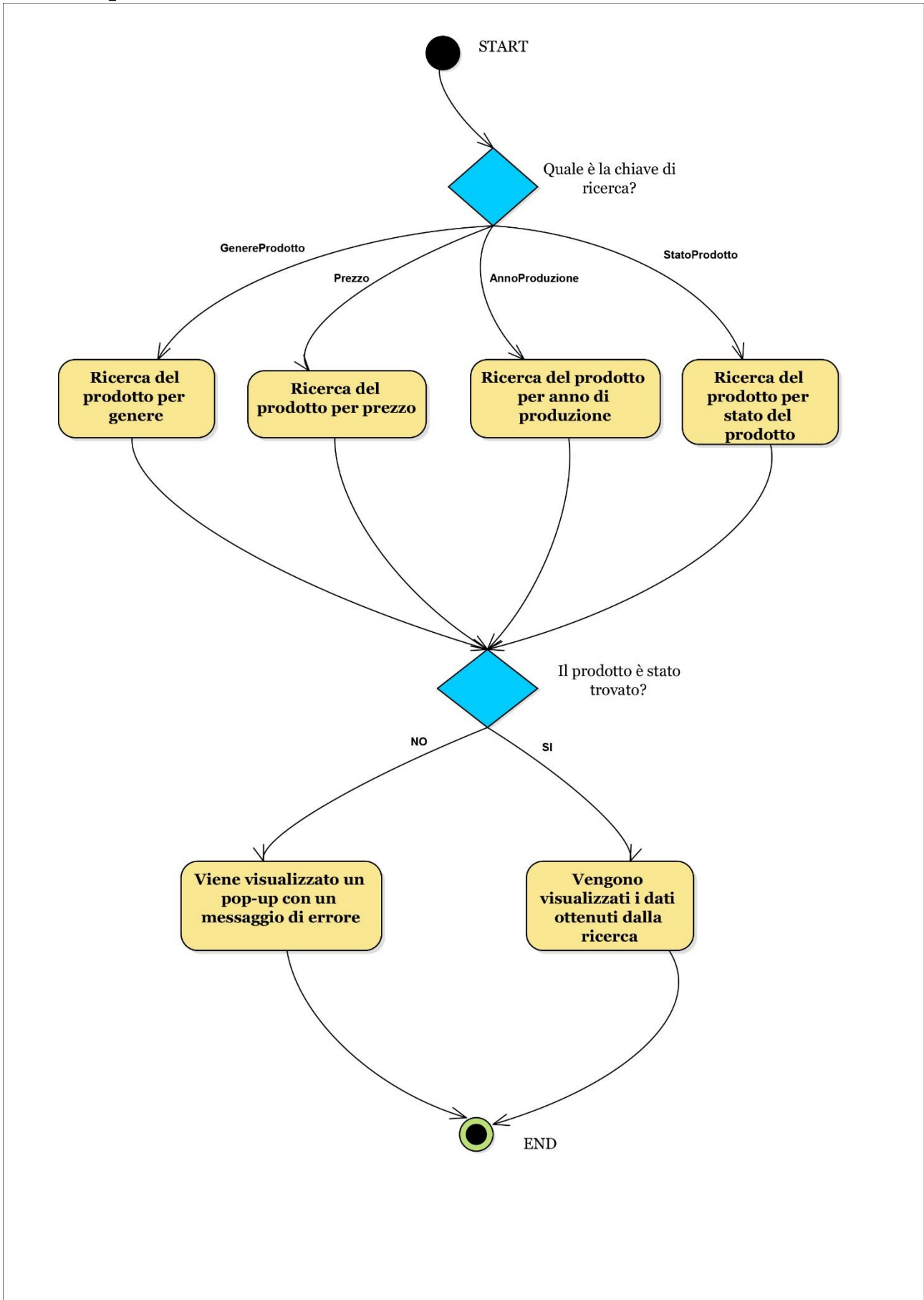




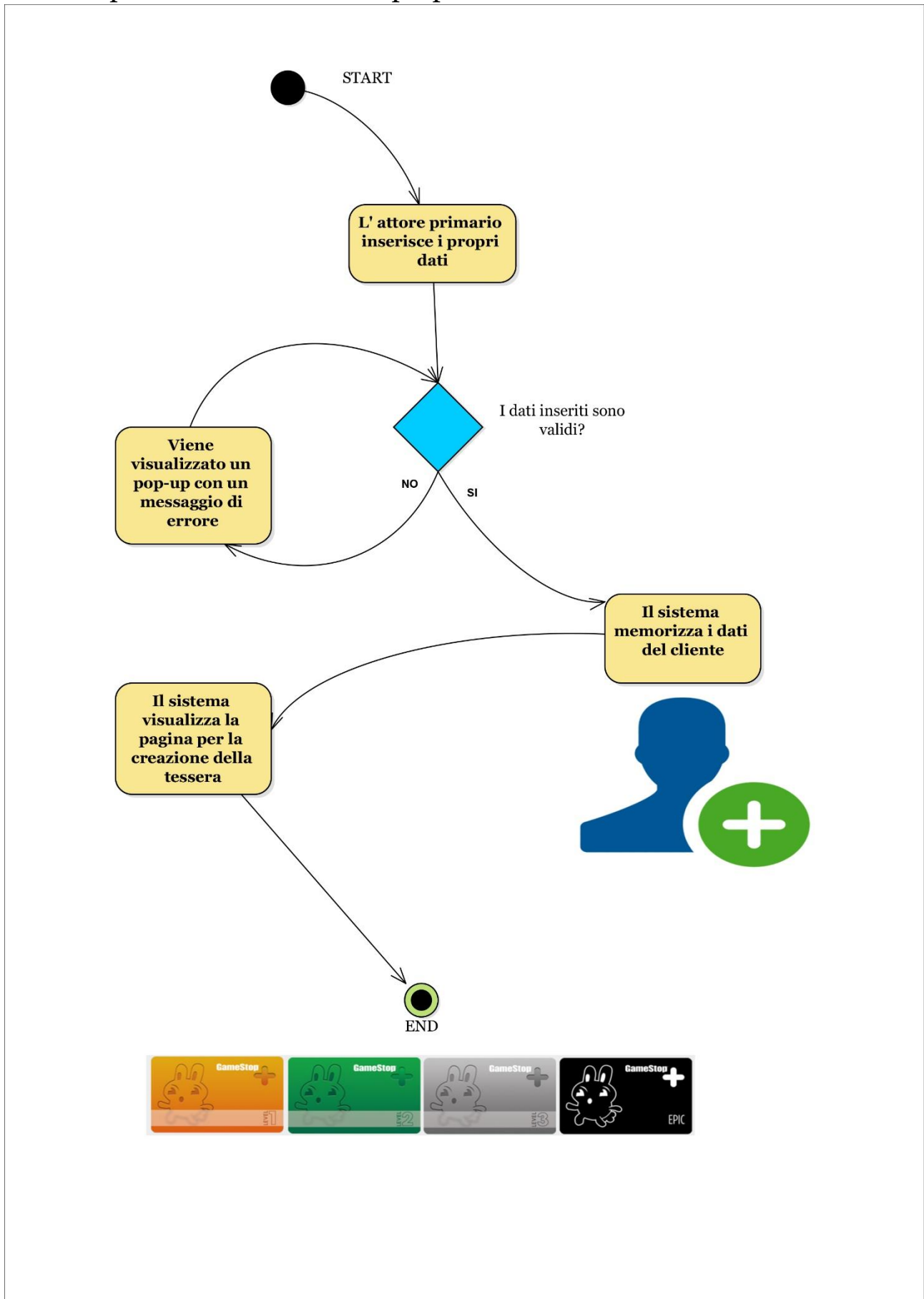
- ✓ Process flow:  
L'attore primario vuole effettuare un operazione CRUD sui prodotti



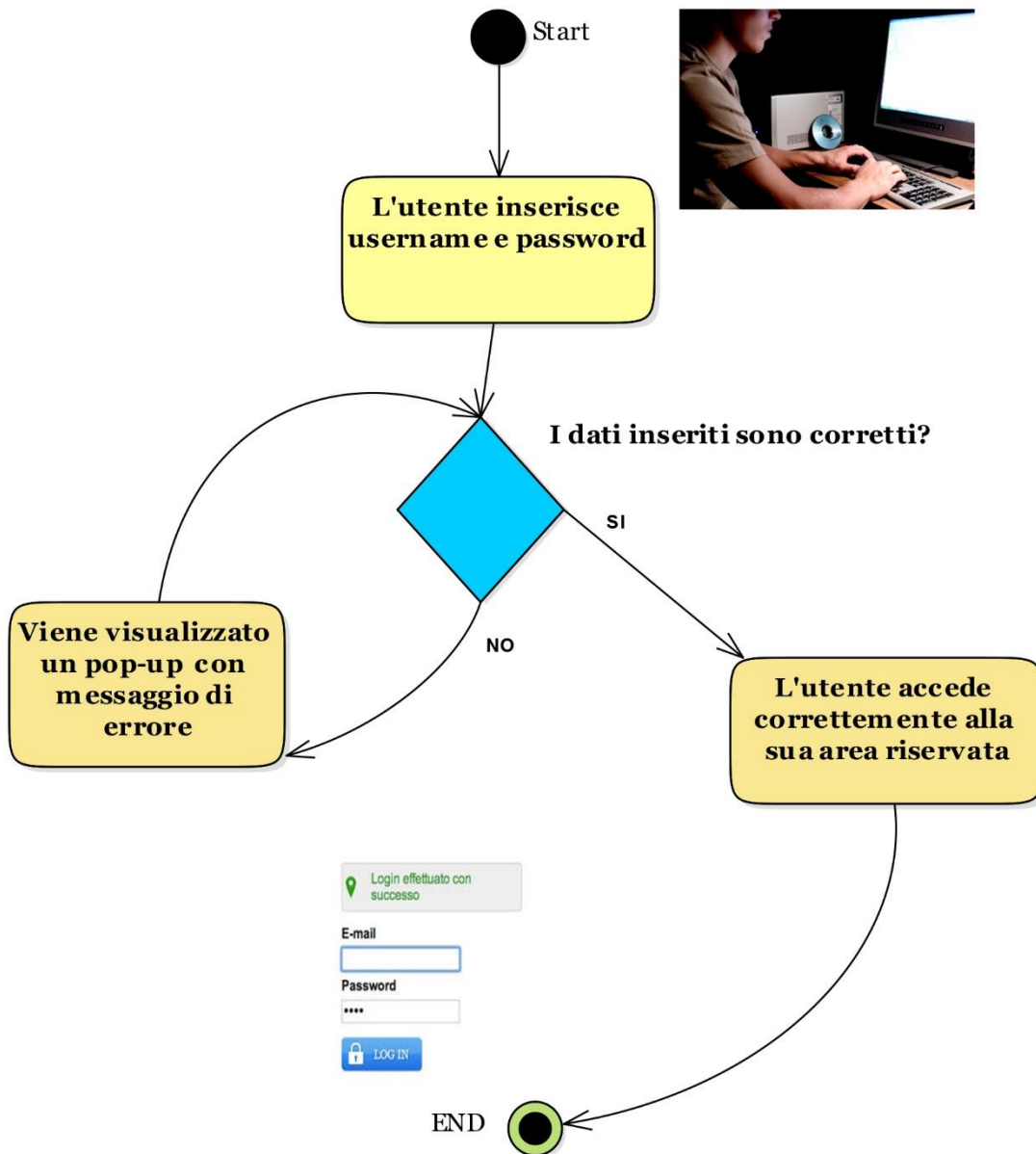
L'attore primario vuole effettuare una RicercaProdotto:



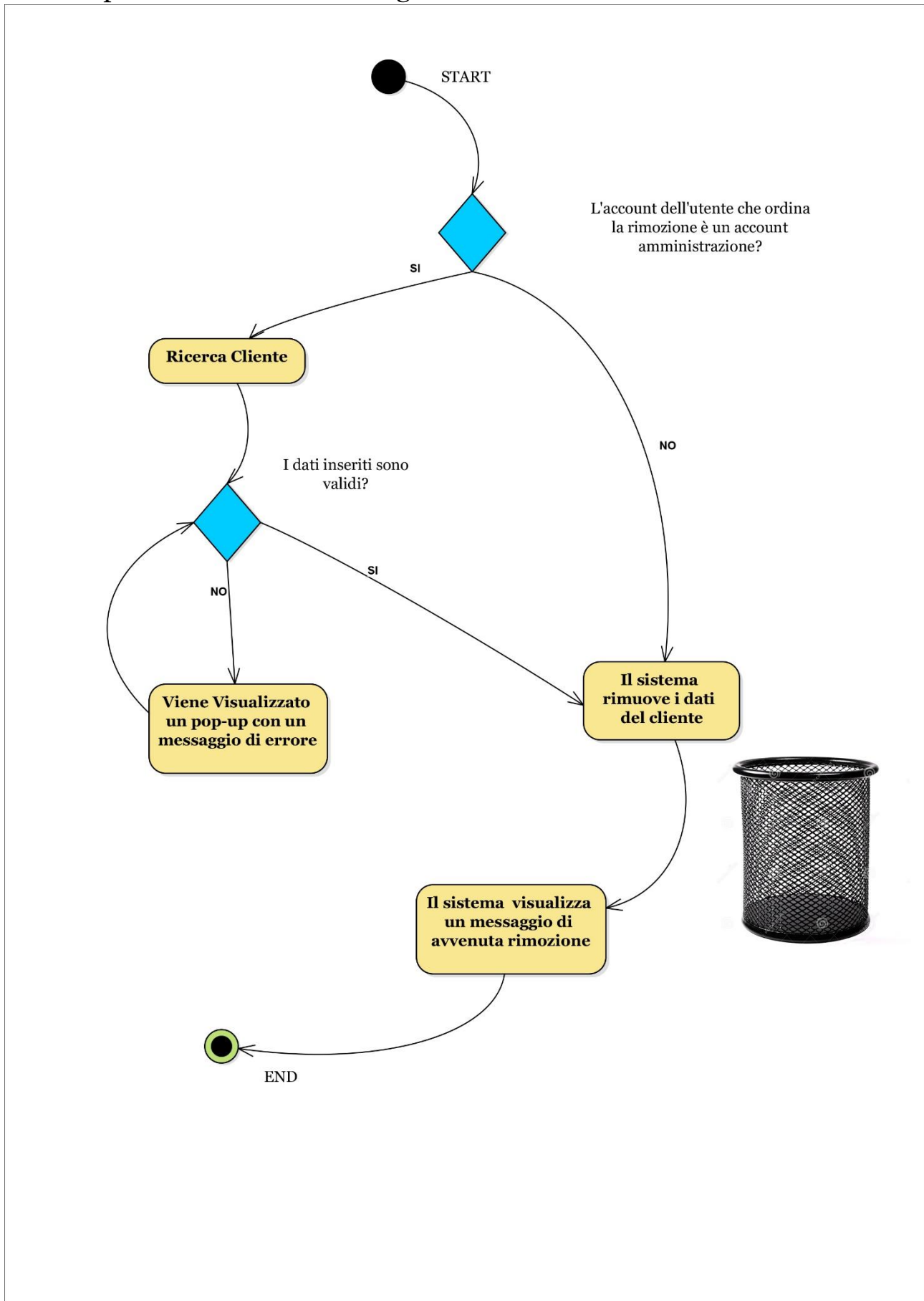
L' attore primario vuole creare il proprio account:



L'attore primario vuole accedere alla propria area riservata:

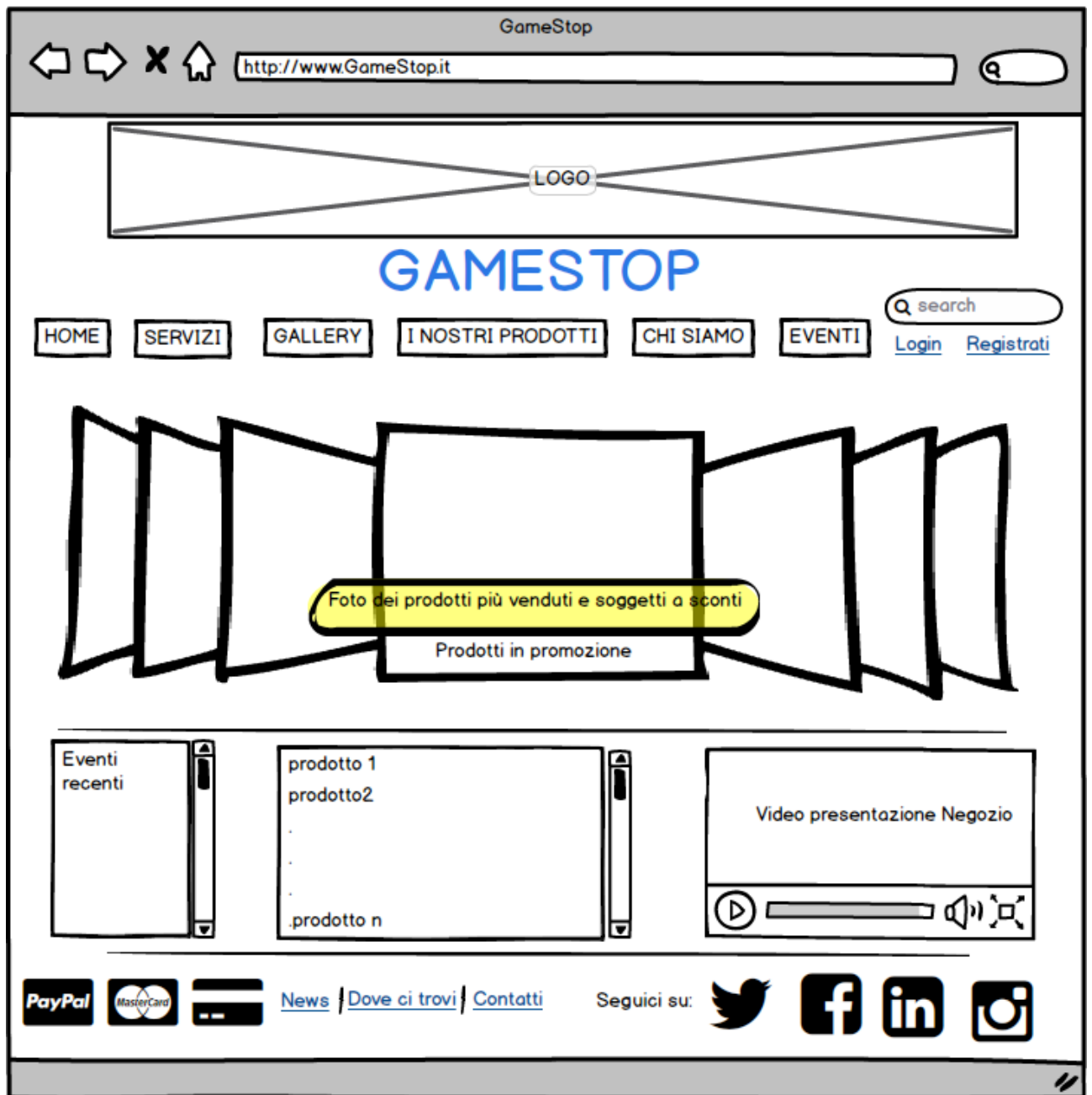


## L'attore primario e secondario vogliono Eliminare un cliente:



✓ Mock-up di livello 0:

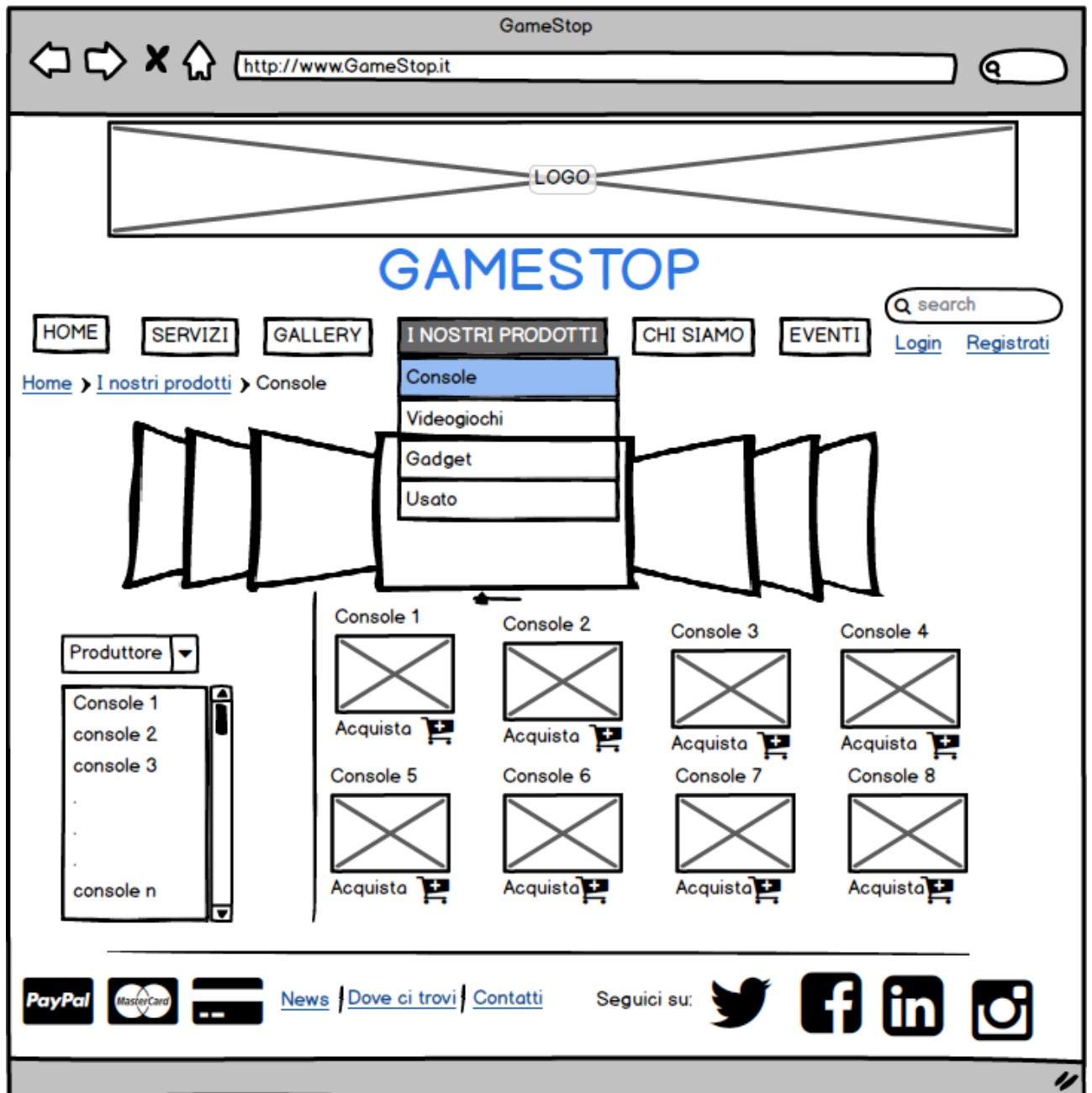
## MOCK-UP DI LIVELLO 0



# MOCK-UP DI LIVELLO 0

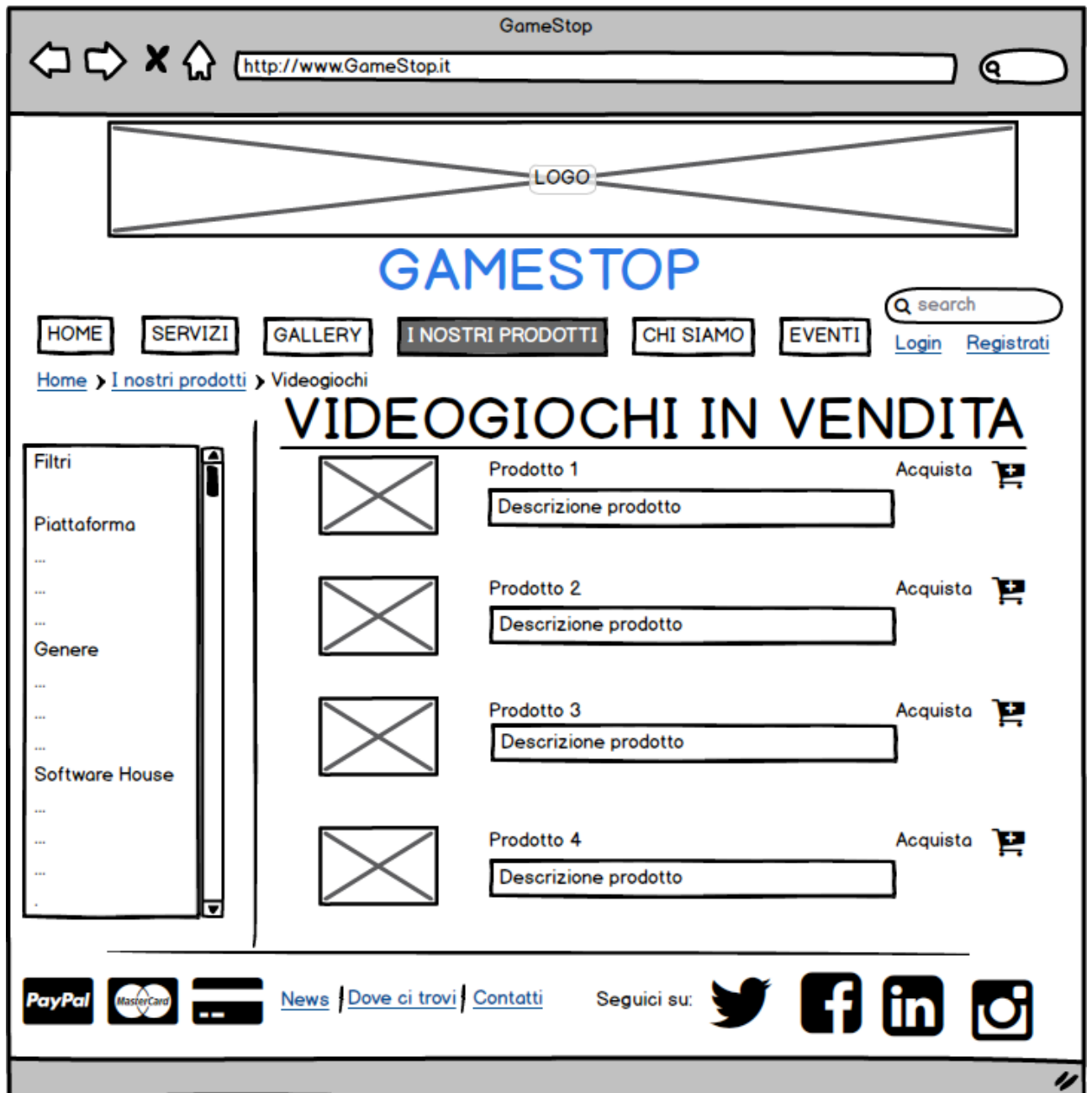


# MOCK-UP DI LIVELLO 0

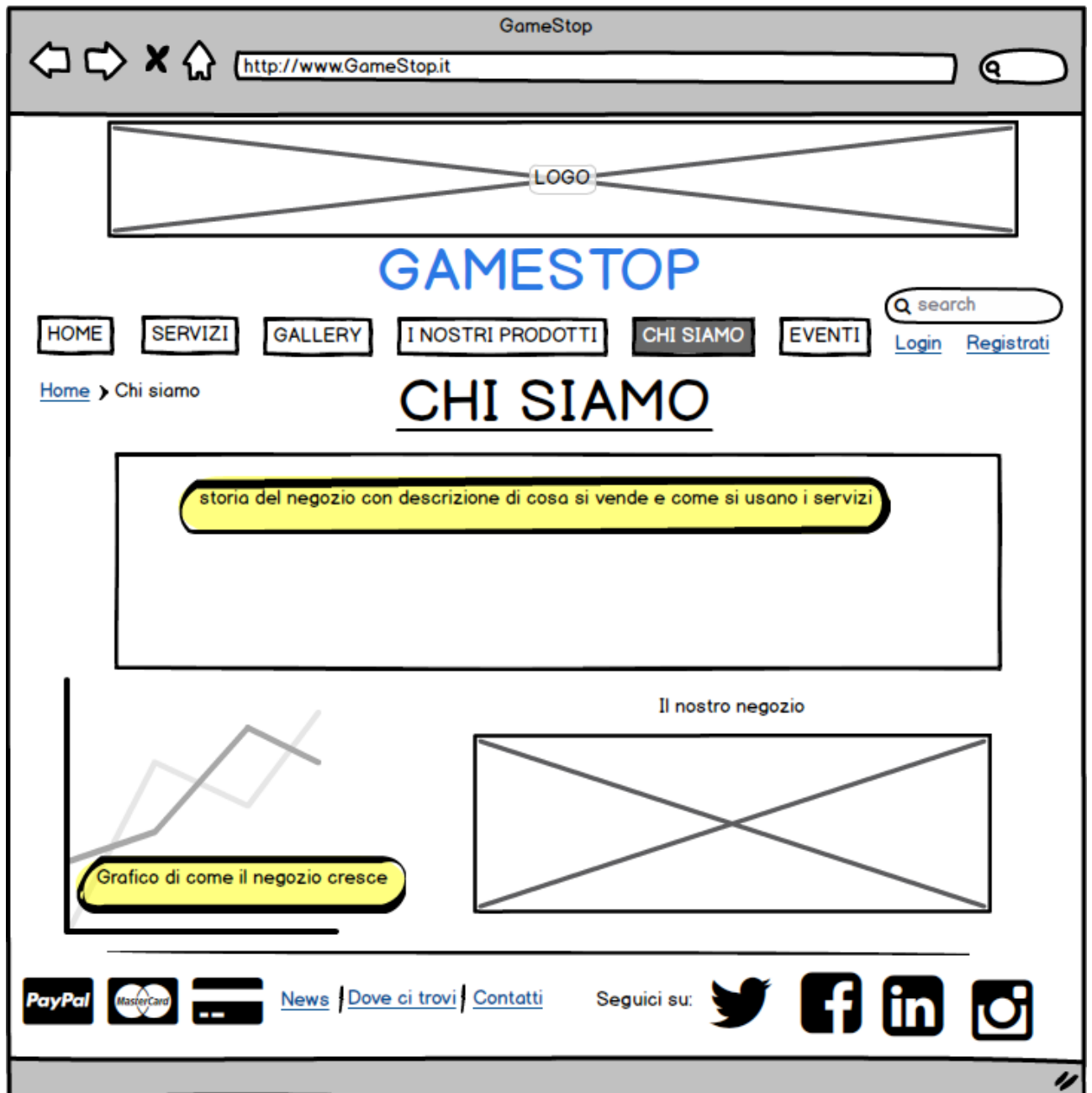




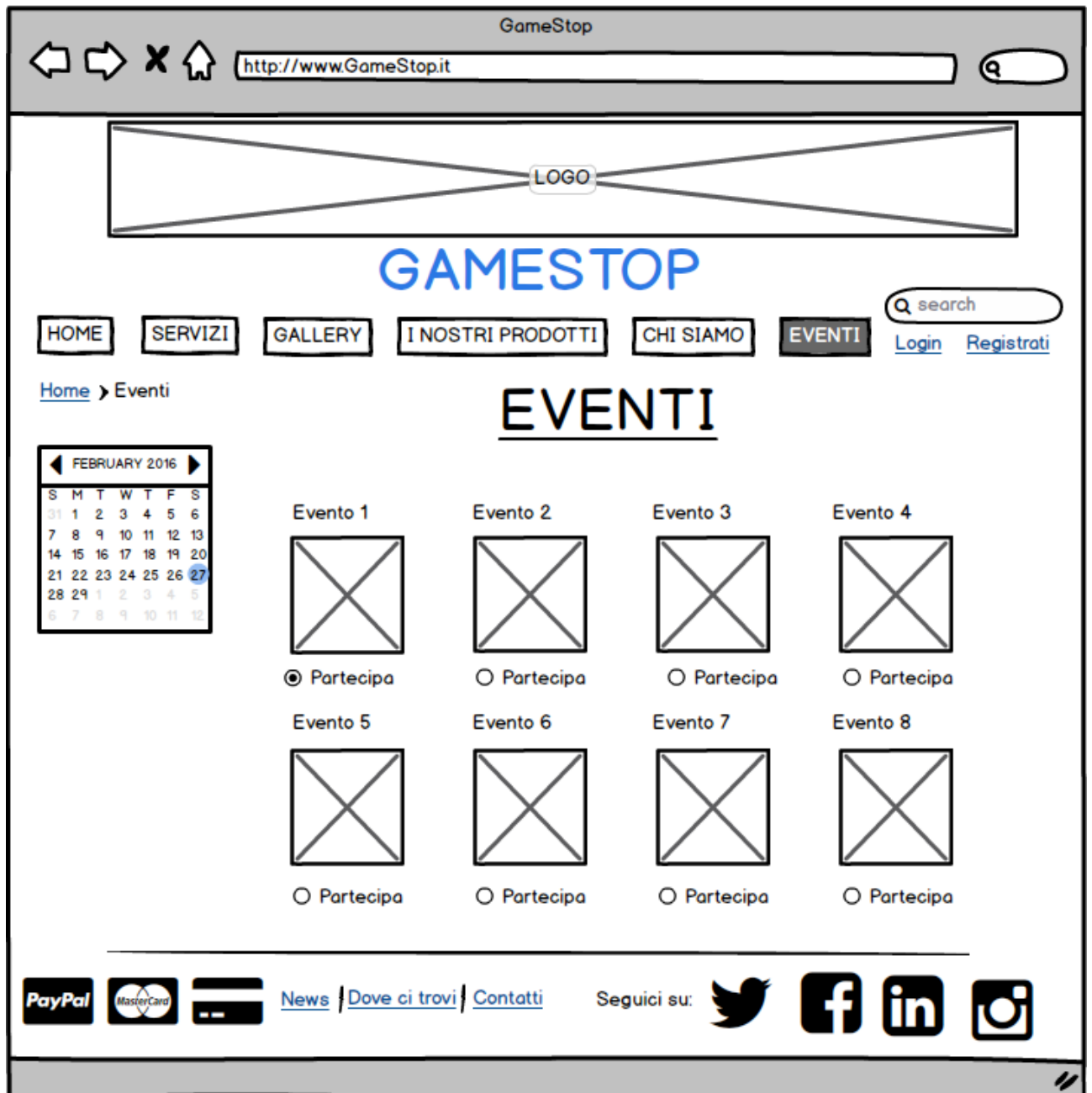
# MOCK-UP DI LIVELLO 0



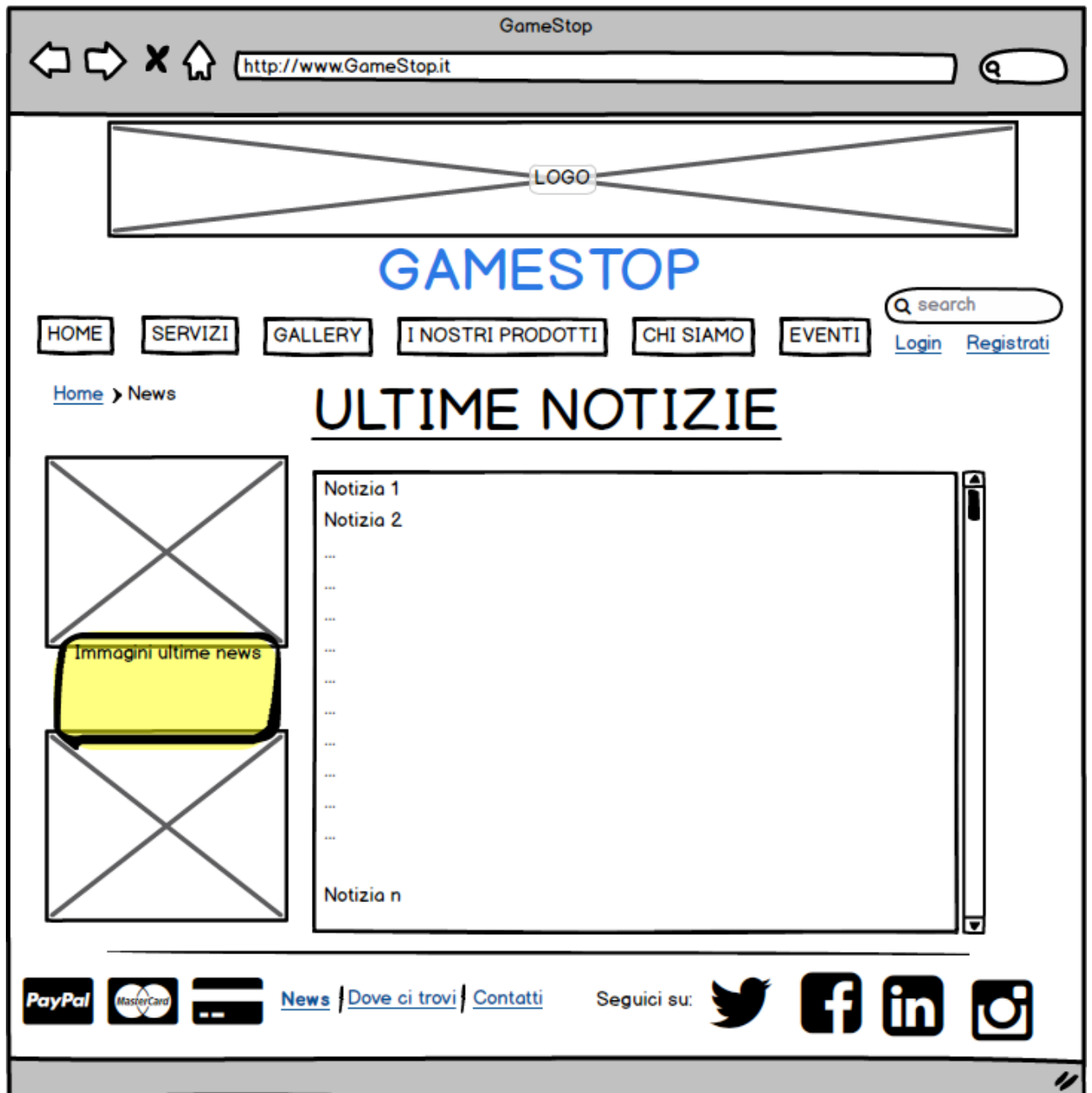
# MOCK-UP DI LIVELLO 0



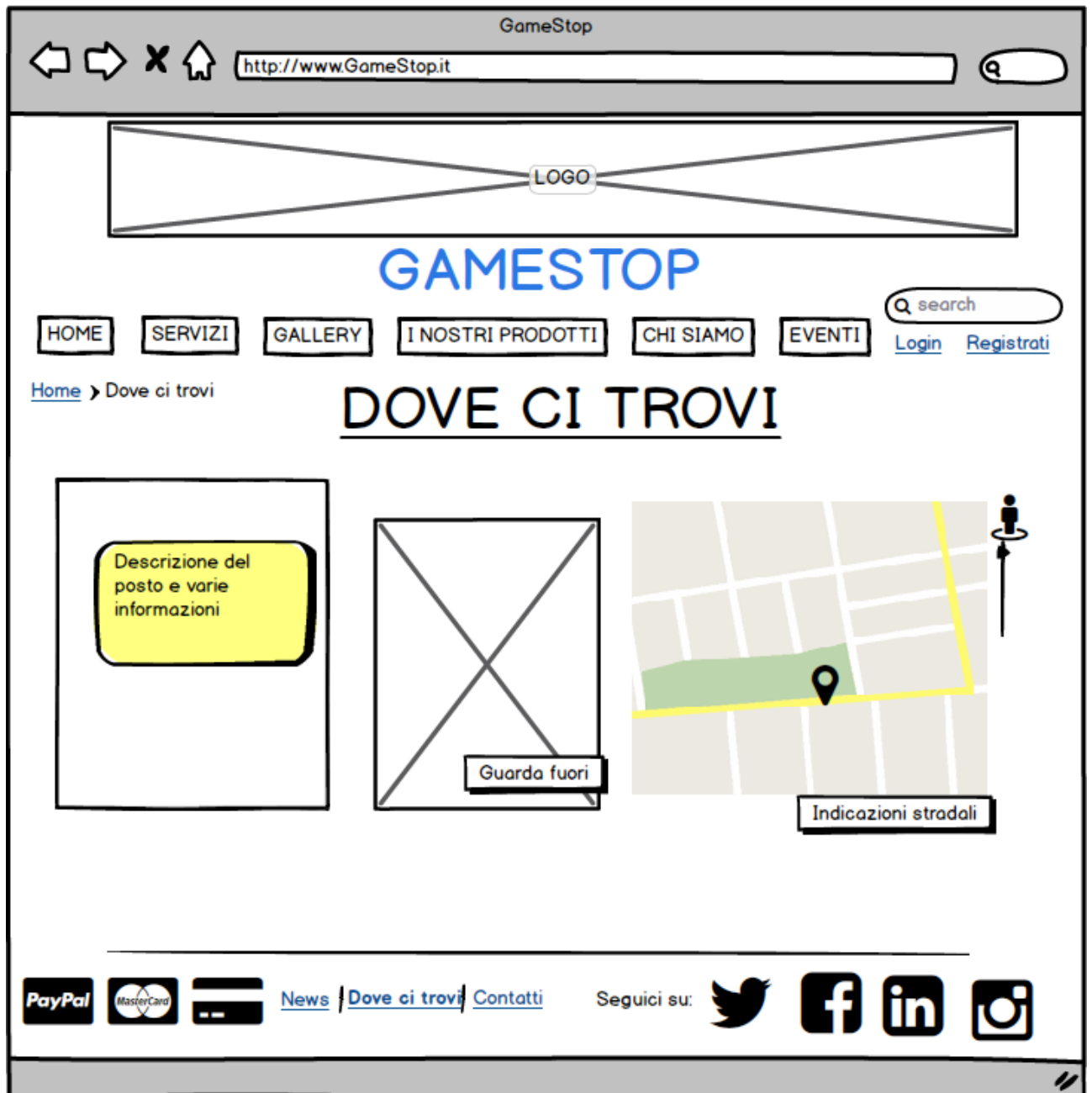
# MOCK-UP DI LIVELLO 0



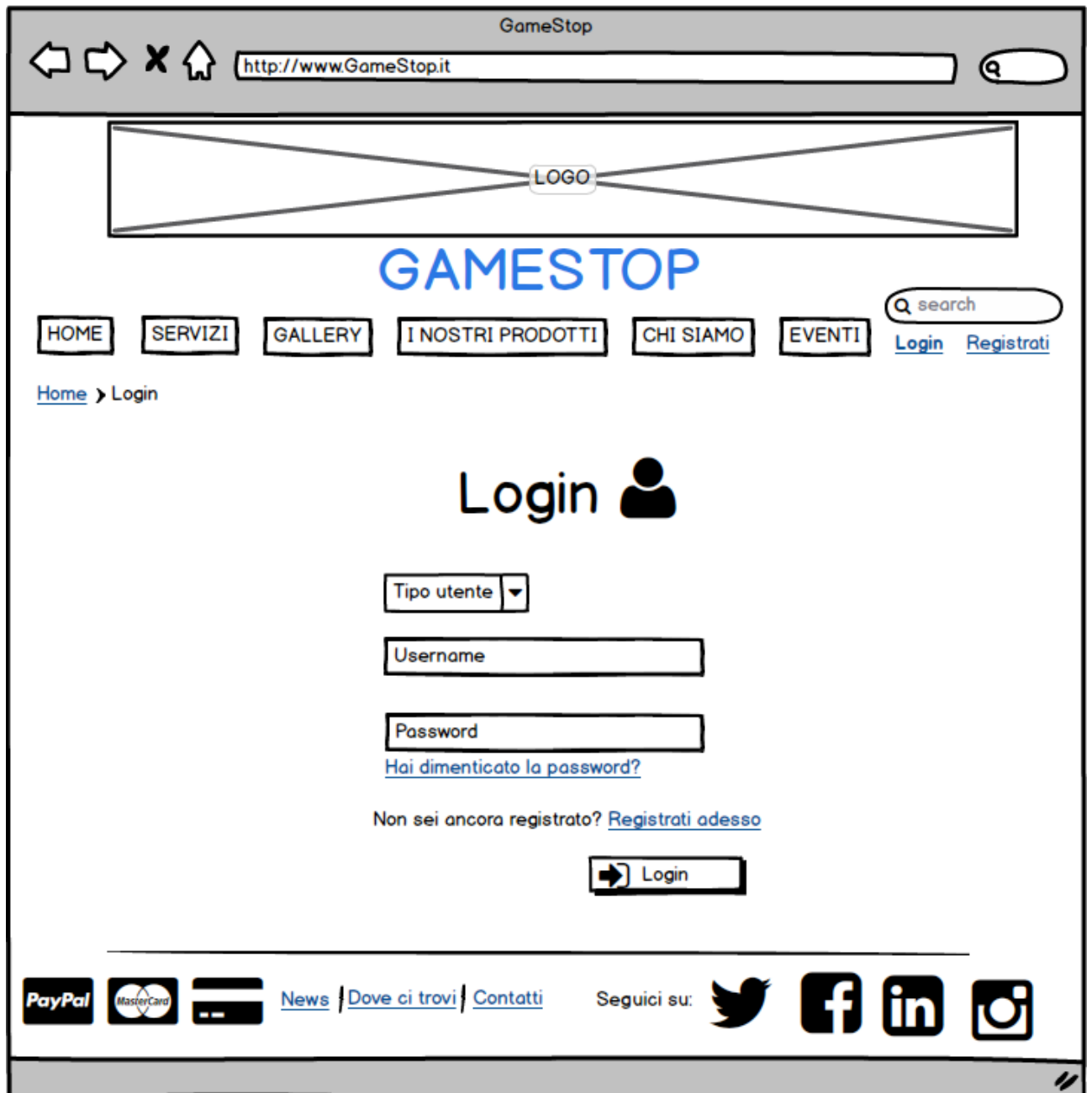
# MOCK-UP DI LIVELLO 0



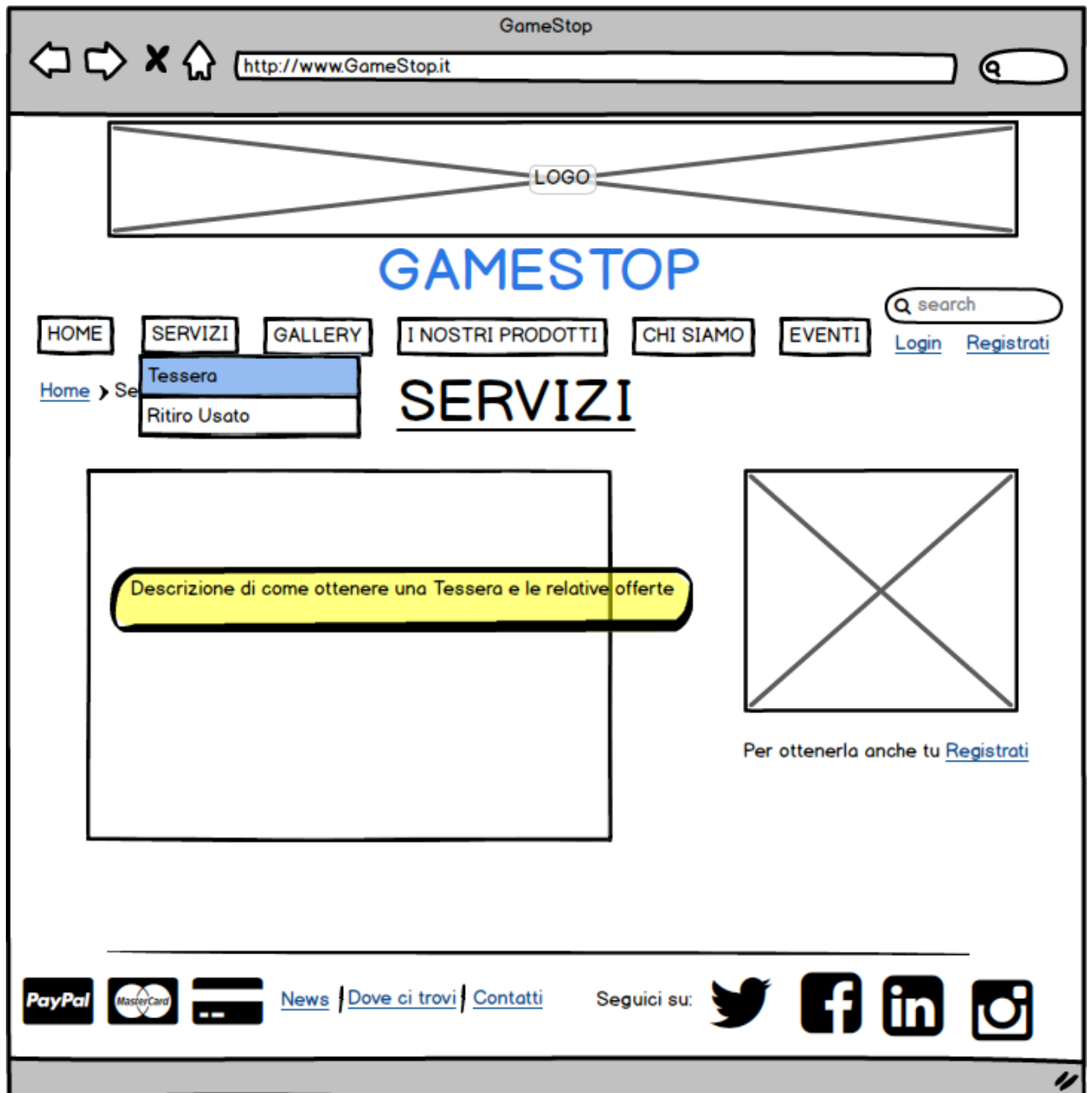
# MOCK-UP DI LIVELLO 0



# MOCK-UP DI LIVELLO 0



# MOCK-UP DI LIVELLO 0



# MOCK-UP DI LIVELLO 0

GameStop

http://www.GameStop.it

LOGO

## GAMESTOP

HOME SERVIZI GALLERY I NOSTRI PRODOTTI CHI SIAMO EVENTI

Q search

[Login](#) [Registrati](#)

[Home](#) > Registrazione

### Registrazione

Nome\* Username\*

Cognome Password\*

CodFiscale/PartitaIVA\* Ripeti Password\*

Email\*

Data di nascita

Via n° Civico

Città CAP

➔ Registrati

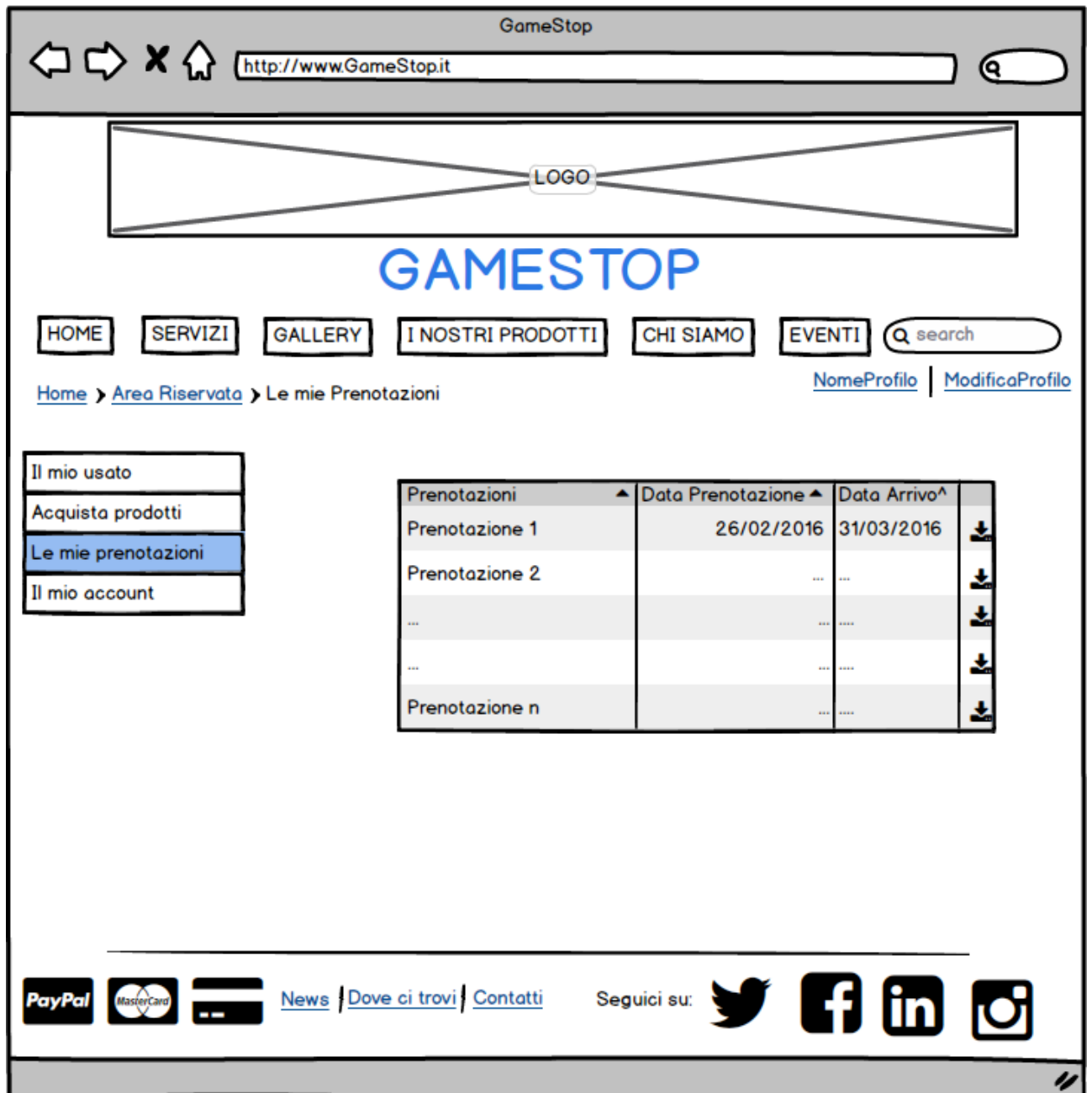
PayPal MasterCard

[News](#) | [Dove ci trovi](#) | [Contatti](#)

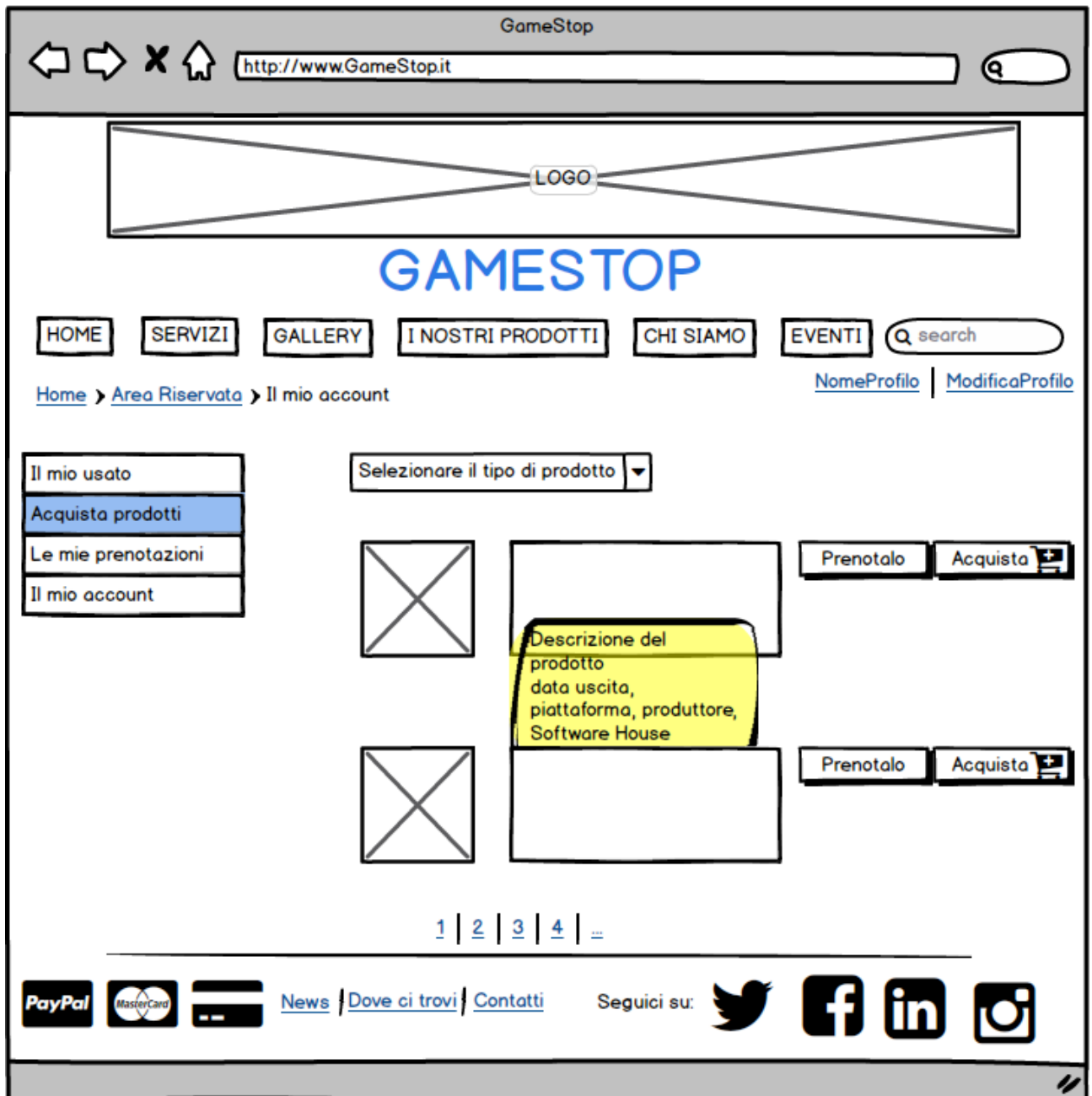
Seguici su:



# MOCK-UP DI LIVELLO 0



# MOCK-UP DI LIVELLO 0




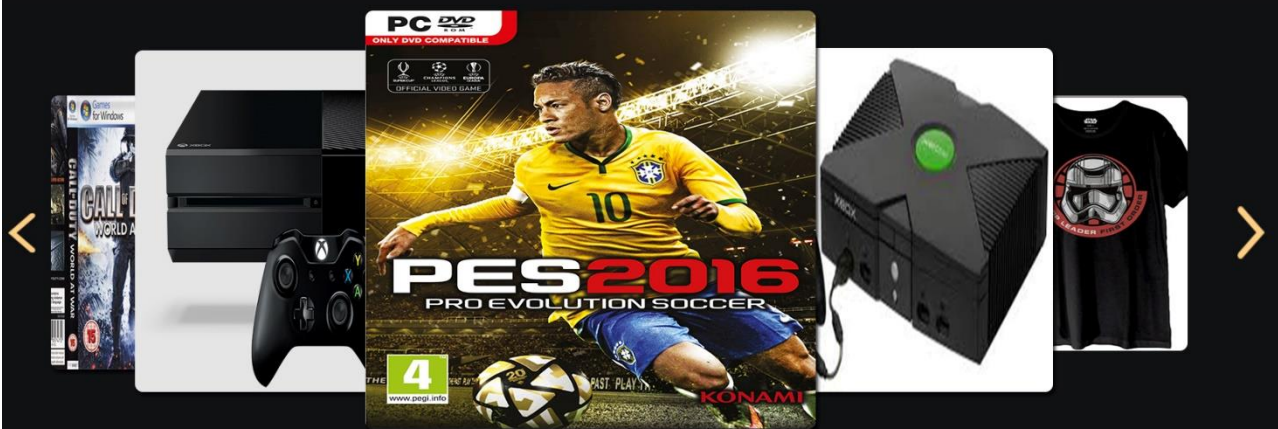
✓ Mock-up di livello 2:

# GameStop®

power to the players™

Home Servizi Gallery I nostri prodotti Chi siamo Eventi

  
[Login](#) [Registrati](#)



**02-03-2016:** grande evento alle 18:30 uscita videogiochi 2016, con la partecipazione di Dexter

**10-12-2015:** presentazione della nuova console di casa Microsoft.

**Playstation 4** a 369 euro ([Clicca qui](#))

**Xbox one** a 329 euro ([Clicca qui](#))

**Need for speed** per pc a 59,90 euro ([Clicca qui](#))

**Final fantasy type 0** per pc a 19,90 euro ([Clicca qui](#))



[News](#) | [Dove ci trovi](#) | [Contatti](#)



Seguici su:



# GameStop®

power to the players™

[Home](#) [Servizi](#) [Gallery](#) [I nostri prodotti](#) [Chi siamo](#) [Eventi](#)

[Login](#) [Registrati](#)

## Prodotti in vendita

### Filtri:

#### Piattaforma

Playstation 4

wii

Xbox one

Xbox 360

wiiu

#### Genere

Arcade

Azione

Fantasy

Guida

Picchiaduro

Sparatutto

Strategia

Sport

Simulazione

#### Software house

Interactive

Activision



### Fifa 16

[BUY NOW](#)

**data uscita:** 24/09/15

**Piattaforma:** Playstation 4

**Genere:** Sport

**Codice:** 1222547896

**Stato:** Nuovo

**Prezzo:** € 70.98



### The heavy rain and beyond

[BUY NOW](#)

**data uscita:** 02/03/16

**Piattaforma:** Playstation 4

**Genere:** Avventura

**Codice:** 1000293145

**Stato:** Nuovo

**Prezzo:** € 40.98



### Deadpool

[BUY NOW](#)

**data uscita:** 02/03/16

**Piattaforma:** Playstation 4

**Genere:** Avventura

**Codice:** 5444752226

[News](#) [Dove ci trovi](#) [Contatti](#)



Seguici su:



# GameStop®

power to the players™

[Home](#) [Servizi](#) [Gallery](#) [I nostri prodotti](#) [Chi siamo](#) [Eventi](#)

[Login](#) [Registrati](#)

## Servizi

### Che vantaggi mi dà la tessera Gamestop?

GameStop ti offre un mondo di vantaggi: sconti, promozioni e supervalutazioni su giochi e consoles. Solo per i possessori della tessera GameStop offre uno sconto sul prezzo dei giochi e accessori usati e nuovi.

L'attivazione di questo servizio non comporta in alcun modo obblighi successivi e ti offrirà l'opportunità di rimanere sempre aggiornato sulle novità del mondo dei videogiochi, con news, anteprime e recensioni esclusive. Infine solo i possessori della Tessera posso usufruire del servizio Ritiro usato, ricevendo una super valutazione sui propri prodotti.

News | [Dove ci trovi](#) | [Contatti](#)

PayPal™ | MasterCard | AMERICAN EXPRESS

Seguici su:

# GameStop®

power to the players™

[Home](#) [Servizi](#) [Gallery](#) [I nostri prodotti](#) [Chi siamo](#) [Eventi](#)

  
[Login](#) [Registrati](#)

## Login

Tipo utente



Username

Password

[Hai dimenticato la password?](#)

non sei ancora registrato? [Registrati adesso](#)

**Login**

[News](#) | [Dove ci trovi](#) | [Contatti](#)

PayPal™



Seguici su:



## Implementazione in MySQL:

Ora andiamo a vedere come sono strutturate le tabelle ed alcune query su un BDMS:

- ✓ Creazione delle tabelle e dichiarazione dei vincoli di integrità referenziali (DDL):

```
1 CREATE TABLE `gamestop`.`cliente` (  
2   `idCliente` VARCHAR(10) NOT NULL,  
3   `nome` VARCHAR(25) NOT NULL,  
4   `cognome` VARCHAR(25) NOT NULL,  
5   `codfiscale/partitaiva` VARCHAR(16) NOT NULL,  
6   `tipocliente` VARCHAR(15) NOT NULL,  
7   `username` VARCHAR(20) NULL,  
8   `password` CHAR(8) NULL,  
9   `datadinascita` DATE NULL,  
10  `città` VARCHAR(20) NULL,  
11  `via` VARCHAR(50) NULL,  
12  `cap` VARCHAR(5) NULL,  
13  `numcivico` VARCHAR(3) NULL,  
14  PRIMARY KEY (`idCliente`));  
15
```

```
1 CREATE TABLE `gamestop`.`prodotto` (  
2   `codiceprod` VARCHAR(10) NOT NULL,  
3   `codicesh` VARCHAR(10) NULL,  
4   `idcliente` VARCHAR(10) NULL,  
5   `data` DATE NULL,  
6   `livintegrità` INT NULL,  
7   `statoprodotto` VARCHAR(5) NOT NULL,  
8   `prezzo` DOUBLE NOT NULL,  
9   `genereprodotto` VARCHAR(25) NOT NULL,  
10  `piattaforma` VARCHAR(25) NULL,  
11  `datauscita` DATE NULL,  
12  `tipogadget` VARCHAR(25) NULL,  
13  `modello` VARCHAR(25) NULL,  
14  `produttore` VARCHAR(50) NULL,  
15  `annoproduzione` DATE NULL,  
16  `genere` VARCHAR(20) NULL,  
17  PRIMARY KEY (`codiceprod`),  
18  FOREIGN KEY (`idcliente`)  
19  REFERENCES `gamestop`.`cliente` (`idCliente`)  
20  FOREIGN KEY (`codicesh`)  
21  REFERENCES `gamestop`.`softwarehouse` (`codicesh`));  
22
```

```
1 CREATE TABLE `gamestop`.`documentofiscale` (  
2   `codice` INT NOT NULL AUTO_INCREMENT,  
3   `data` DATE NOT NULL,  
4   `importo` DOUBLE NOT NULL,  
5   PRIMARY KEY (`codice`));  
6
```

```
1 CREATE TABLE `gamestop`.`amministrazione` (  
2   `codfiscale` VARCHAR(16) NOT NULL,  
3   `nome` VARCHAR(25) NOT NULL,  
4   `cognome` VARCHAR(25) NOT NULL,  
5   `tipoincarico` VARCHAR(20) NOT NULL,  
6   `username` VARCHAR(20) NULL,  
7   `password` CHAR(8) NULL,  
8   `email` VARCHAR(25) NULL,  
9   `città` VARCHAR(20) NULL,  
10  `via` VARCHAR(50) NULL,  
11  `cap` VARCHAR(5) NULL,  
12  `numcivico` VARCHAR(3) NULL,  
13  PRIMARY KEY (`codfiscale`));  
14
```

```
1 CREATE TABLE `gamestop`.`softwarehouse` (  
2   `codicesh` VARCHAR(10) NOT NULL,  
3   `nazionalità` VARCHAR(25) NULL,  
4   `città` VARCHAR(20) NULL,  
5   `cap` VARCHAR(5) NULL,  
6   `via` VARCHAR(50) NULL,  
7   `numcivico` VARCHAR(3) NULL,  
8   PRIMARY KEY (`codicesh`));  
9
```

```
1 CREATE TABLE `gamestop`.`tessera` (  
2   `idtessera` VARCHAR(10) NOT NULL,  
3   `idcliente` VARCHAR(10) NOT NULL,  
4   `datagen` DATE NULL,  
5   `livellotessera` INT NULL,  
6   `datascad` DATE NULL,  
7   PRIMARY KEY (`idtessera`),  
8   FOREIGN KEY (`idcliente`)  
9   REFERENCES `gamestop`.`cliente` (`idCliente`));  
10
```



```

1 CREATE TABLE `gamestop`.`prenota` (
2   `idcliente` VARCHAR(10) NOT NULL,
3   `codiceprod` VARCHAR(10) NOT NULL,
4   `dataprenotazione` DATE NULL,
5   `quantita` INT NOT NULL,
6   `dataarrivo` DATE NULL,
7   PRIMARY KEY (`idcliente`,`codiceprod`),
8   FOREIGN KEY (`idcliente`)
9   REFERENCES `gamestop`.`cliente` (`idCliente`),
10  FOREIGN KEY (`codiceprod`)
11  REFERENCES `gamestop`.`prodotto` (`codiceprod`));
12

```

```

1 CREATE TABLE `gamestop`.`offerta` (
2   `codofferta` VARCHAR(10) NOT NULL,
3   `codfiscale` VARCHAR(16) NOT NULL,
4   `datac` DATE NULL,
5   `datainizio` DATE NULL,
6   `datascad` DATE NULL,
7   `tipoefferta` VARCHAR(20) NULL,
8   `idtessera` VARCHAR(10) NOT NULL,
9   `dataassegnazione` VARCHAR(45) NULL,
10  PRIMARY KEY (`codofferta`),
11  FOREIGN KEY (`codfiscale`)
12  REFERENCES `gamestop`.`amministrazione` (`codfiscale`),
13  FOREIGN KEY (`idtessera`)
14  REFERENCES `gamestop`.`tessera` (`idtessera`));
15

```

```

1 CREATE TABLE `gamestop`.`evento` (
2   `idevento` VARCHAR(10) NOT NULL,
3   `codfiscale` VARCHAR(16) NOT NULL,
4   `dataevento` DATE NULL,
5   `genereevento` VARCHAR(25) NULL,
6   `vippartecipante` VARCHAR(50) NULL,
7   PRIMARY KEY (`idevento`),
8   FOREIGN KEY (`codfiscale`)
9   REFERENCES `gamestop`.`amministrazione` (`codfiscale`));
10

```

```

1 CREATE TABLE `gamestop`.`partecipa` (
2   `idcliente` VARCHAR(10) NOT NULL,
3   `idevento` VARCHAR(10) NOT NULL,
4   PRIMARY KEY (`idcliente`,`idevento`),
5   FOREIGN KEY (`idcliente`)
6   REFERENCES `gamestop`.`cliente` (`idCliente`),
7   FOREIGN KEY (`idevento`)
8   REFERENCES `gamestop`.`evento` (`idevento`));
9

```

```

1 CREATE TABLE `gamestop`.`participa` (
2   `idcliente` VARCHAR(10) NOT NULL,
3   `idevento` VARCHAR(10) NOT NULL,
4   PRIMARY KEY (`idcliente`, `idevento`),
5   FOREIGN KEY (`idcliente`)
6   REFERENCES `gamestop`.`cliente` (`idCliente`),
7   FOREIGN KEY (`idevento`)
8   REFERENCES `gamestop`.`evento` (`idevento`));
9

```

```

1 CREATE TABLE `gamestop`.`acquisto` (
2   `idcliente` VARCHAR(10) NOT NULL,
3   `codiceprod` VARCHAR(10) NOT NULL,
4   `codicefattura` INT NOT NULL,
5   `datapagamento` DATE NULL,
6   PRIMARY KEY (`idcliente`, `codiceprod`, `codicefattura`),
7   FOREIGN KEY (`idcliente`)
8   REFERENCES `gamestop`.`cliente` (`idCliente`),
9   FOREIGN KEY (`codiceprod`)
10  REFERENCES `gamestop`.`prodotto` (`codiceprod`),
11  FOREIGN KEY (`codicefattura`)
12  REFERENCES `gamestop`.`documentofiscale` (`codice`));
13

```

```

1 create assertion prodottousato check(
2 not exists(
3 select *
4 from prodotto
5 where (statoprodotto='usato')and(idcliente is null));

```

✓ Query:

1. selezionare il nome e il cognome dei clienti che hanno portato videogiochi di genere sparattutto sviluppati da una softwarehouse francese;

```

1 select c.idcliente, c.nome, c.cognome
2 from cliente c, prodotto p, softwarehouse s
3 where (c.idcliente=p.idcliente)and(p.codicesh=s.codicesh)
4 and(p.genereprodotto='videogioco')and(p.genere='sparattutto')
5 and(s.nazionalità='francia');
6

```

2. selezionare i nomi dei clienti registrati che hanno acquistato almeno una console prodotta dalla SONY e non hanno mai portato un videogioco uscito nel 2014 con livintegrità maggiore di 4;

```
1 • select c.idcliente, c.nome
2   from cliente c, acquista a, prodotto p
3   where(c.idcliente=a.idcliente)and(a.codiceprod=p.codiceprod)and(c.tipocliente='registrato')
4     and(p.genereprodotto='console')and(p.prodotto='SONY')
5     and (c.codice) not in(select c1.idcliente,c1.nome
6   from cliente c1, prodotto p1
7   where (c1.idcliente=p1.idcliente)and(p1.livintegrità>4)and(p1.genereprodotto='videogioco')
8   and(p1.datauscita>'31-12-2013')and(p1.datauscita<'1-1-2015'));
```

3. selezionare il cliente con il massimo livello di tessera;

```
1 • select c.idcliente, c.nome, c.cognome
2   from cliente c, tessera t
3   where (c.idcliente=t.idcliente)and(livtessera=(select max(t1.livtessera)
4   |from tessera t1));
```

4. selezionare il nome e il cognome del cliente che ha acquistato almeno un videogioco usato di genere sportivo sviluppato da una software house di nazionalità americana e hanno acquistato solo console playstation 1;

**Parafrasi:** selezionare il nome e il cognome del cliente che ha acquistato almeno un videogioco usato di genere sportivo sviluppato da una software house di nazionalità americana tale che non esista un acquisto riferito a quel cliente e che non esista una console playstation 1 riferito a quell' acquisto;

```
1 • select c.idcliente, c.nome, c.cognome
2   from cliente c, acquista a, prodotto p, softwarehouse s
3   where(c.idcliente=a.idcliente)and(a.codiceprod=p.codiceprod)and(p.codicesh=s.codicesh)
4     and(p.genereprodotto='videogioco')and(p.statoprodotto='usato')
5     and(p.genere='sport')and(s.nazionalità='usa')
6     and NOT EXISTS(select *
7   from acquista a1
8   where (a1.idcliente=c.idcliente)and NOT EXISTS(select *
9   from prodotto p1
10  where (p1.genereprodotto='console')
11  and(p1.modelo='playstation1')
12  and(p1.codiceprod=a1.codiceprod)));
```

5. selezionare il nome e l'incarico degli amministratori ordinati per numero di eventi creati in modo decrescente;

```
1 • select a.codfiscale, a.nome, a.tipoincarico, count(e.idevento)
2 from amministrazione a, evento e
3 where (a.codfiscale=e.codfiscale)
4 group by a.codfiscale, a.nome, a.tipoincarico
5 order by count(e.idevento) desc;
```

6. selezionare il nome e cognome del cliente che ha acquistato il massimo numero di videogiochi;

```
1 • select c.idcliente, c.nome, c.cognome
2 from cliente c, acquista a, prodotto p
3 where (c.idcliente=a.idcliente)and(a.codiceprod=p.codiceprod)and(p.genereprodotto='videogioco')
4 group by c.idcliente, c.nome, c.cognome
5 having count(a.codiceprod)>=all (select count(a1.codiceprod)
6 from acquista a1,prodotto p
7 where (a1.codiceprod=p.codiceprod)and(p.genereprodotto='videogioco')
8 group by a1.idcliente);
```

7. estrarre il numero medio dei prodotti per genere;

```
1 • create view categoriaprodotti(genereprodotto, numprodotti) as
2 select p.genereprodotto, count(p.codiceprod)
3 from prodotto p
4 group by p.genereprodotto;
5
6 • select avg(numprodotti)
7 from categoriaprodotti;
```