**Progetto Sistema Informativo Game Life**

**1. Analisi e Progettazione Concettuale**

**Entità principali:**

* **Cliente**: nome, email, data\_registrazione
* **Gioco**: titolo, descrizione, data\_uscita, prezzo, genere
* **CasaProduzione**: nome, anno\_fondazione, sede
* **Venditore**: nome, sito\_web, contatto\_email
* **Recensione**: contenuto, voto, data\_pubblicazione
* **Acquisto**: data, metodo\_pagamento

**Relazioni:**

* Un **gioco** può essere prodotto da più **case di produzione** (n:m)
* Un **gioco** può essere venduto da più **venditori** (n:m)
* Un **cliente** può scrivere più **recensioni**, ogni recensione è associata a un solo **gioco** (1:n)
* Un **cliente** può acquistare più **giochi** (1:n)

Non ci sono generalizzazioni nel modello, ma sono state gestite relazioni molti-a-molti tramite Django con campi ManyToManyField.

**2. Progettazione Logica**

**Tabelle principali:**

* Cliente(id, nome, email, data\_registrazione)
* Gioco(id, titolo, descrizione, data\_uscita, prezzo, genere)
* CasaProduzione(id, nome, anno\_fondazione, sede)
* Venditore(id, nome, sito\_web, contatto\_email)
* Recensione(id, contenuto, voto, cliente\_id, gioco\_id, data\_pubblicazione)
* Acquisto(id, cliente\_id, gioco\_id, data, metodo\_pagamento)
* Tabelle di relazione many-to-many:
  + gioco\_case\_produzione(gioco\_id, casaproduzione\_id)
  + gioco\_venditori(gioco\_id, venditore\_id)

**Vincoli:**

* email è univoco per ogni cliente
* titolo del gioco è univoco
* metodo\_pagamento è vincolato a valori: 'Carta', 'PayPal', 'Bonifico'
* Le relazioni sono impostate con chiavi esterne per mantenere l'integrità referenziale

**3. Implementazione del Sistema Informativo (Django)**

L'applicazione è realizzata con Django e utilizza:

* **Backend Django** (models, views, forms)
* **Template system** Django per generare pagine HTML
* **Bootstrap CSS** per uno stile responsive e moderno

**Funzionalità implementate:**

1. **Homepage dinamica con recensioni recenti**
2. **Catalogo giochi** con schede dettagliate e possibilità di navigare per titolo
3. **Pagina dedicata alle recensioni**: visualizzazione e inserimento
4. **Sezione acquisti**: possibilità di acquistare un gioco selezionando il metodo di pagamento
5. **Gestione relazioni tra giochi, case di produzione e venditori**

Ogni entità ha un modello Django dedicato, accessibile dal pannello admin. Le relazioni sono mappate tramite chiavi esterne e campi ManyToManyField.

Immagine che contiene schermata, diagramma, spazio

Il contenuto generato dall'IA potrebbe non essere corretto.