

DataBase Foundations

Project Report - Virtual Video Game Store Database Design

Laura Marcela Santana Sánchez
Juan Pablo Borja Espitia
Universidad Distrital, Bogotá, Colombia.

Introduction

In this project, we are designing a database for a virtual video game store. A virtual store is an online place where people can buy things, like video games. The idea of virtual stores started in the 1990s when the internet became popular, and people wanted to shop online [1]. One famous platform for buying and playing games is Steam. We are using Steam as a reference to help us create the database.

This report will explain how the business model of a virtual game store works. We will also talk about the information the store needs, and how we designed the database. Lastly, we will describe the steps we followed to create the Entity-Relationship Model (ERM), and show the final design of the database.

Business Model

The business model of a virtual game store like Steam is simple. The platform makes money by selling video games. Users can buy games, download them, and play them. Sometimes, they can also buy extra content for games, like new levels or special items, which is called DLC (Downloadable Content) [2]. Steam also lets people join communities, write reviews, and unlock badges and achievements, making the platform more fun and interactive [3].

Processes and Information Required

To make a virtual store work, the platform needs to manage different types of information. Here are some examples of important information:

- **User information:** like name, email, country, and payment details.
- **Game information:** like name, price, release date and genre.
- **Community Information:** like forum posts, reviews and badges.

The main processes the platform needs are:

- User create accounts to buy and play games.
- User create libraries for storing Video games.
- User can search games.
- User can review games and join communities.

Steps to Create the Entity-Relationship Model (ERM)

1. **Identify entities:** We first found the important parts of the platform, like users, games and forums.
2. **Define attributes:** We gave each entity important details, like a user's nickname a game's price, or a review's rating.
3. **Find relationships:** We looked at how the entities are connected. For example, a user can buy many games, and a game can have many reviews.
4. **Draw the ERM:** Finally, we made the Entity-Relationship Model, showing the entities, their attributes, and how they relate to each other.
5. **Adjusting many-to-many relationships:** When we found many-to-many relationships between the entities, we made adjustments to normalize the database. This helped us remove redundancies and organize the information better.

Database design:

1. Components:

- Users.
- Games.
- Forums.

2. Entities:

- User.
- Game.
- Genre.
- Category.
- Badge.
- Message.
- Community.

- Developer.
- Achievement.
- Forum.
- Review.
- DLC.

3. Attributes per entity:

- E_1 **User:** Nickname, Friends, Shopping Cart, Downloads, Notifications, Messages, Wallet, Wish List, Email, Password, Level, Country, Creation Date, Steam ID, Status, Phone, Subscriptions, Name.
- E_2 **Game:** ID, Price, Release Date, Update History, Name, Description, Languages, Achievements, Category, Genre, Review, DLC (Downloadable Content), System Requirements, Community, Developer, Age Restriction, Size.
- E_3 **Genre:** ID, Name.
- E_4 **Category:** ID, Name.
- E_5 **Badge:** ID, Name, Experience, Description, Icon.
- E_6 **Message:** ID, Title, Content, Author, Date.
- E_7 **Community:** ID, Subscribers, Posts, Associated Game.
- E_8 **Developer:** ID, Biography, Associated Games
- E_9 **Achievement:** Name, ID, Description, Icon.
- E_{10} **Forum:** Messages, Title, Author, Date.
- E_{11} **Review:** Rating, Comment, User, Date.
- E_{12} **DLC:** Name, Price, Associated Game, Size.

4. Define relationships:

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
E1												
E2												
E3												
E4												
E5												
E6												
E7												
E8												
E9												
E10												
E11												
E12												

Figure 1: Relationships

5. Define relationships types:

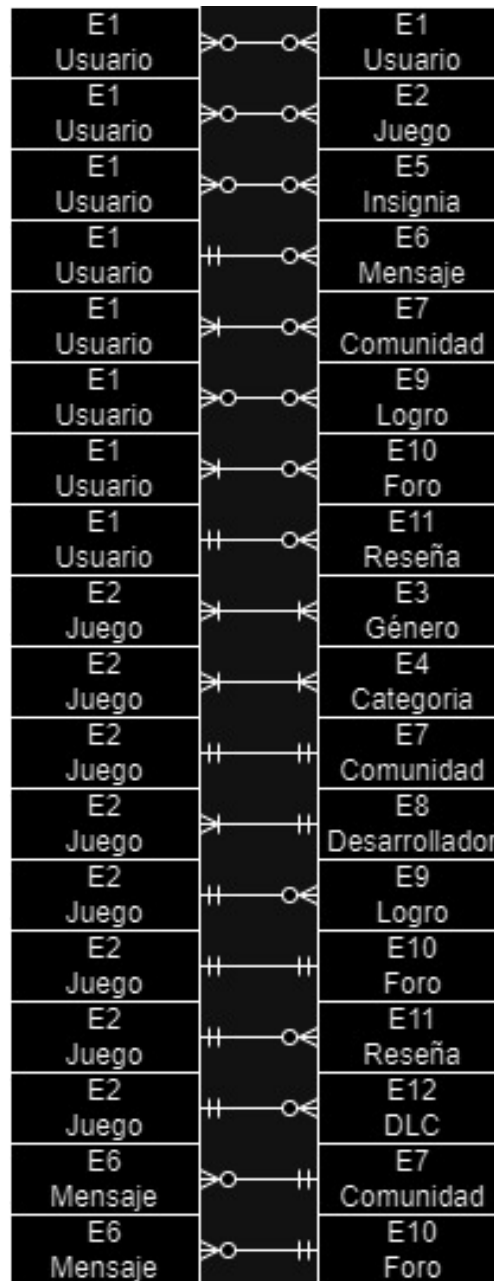


Figure 2: Relationships types

6. First Entity-relationships Draw:

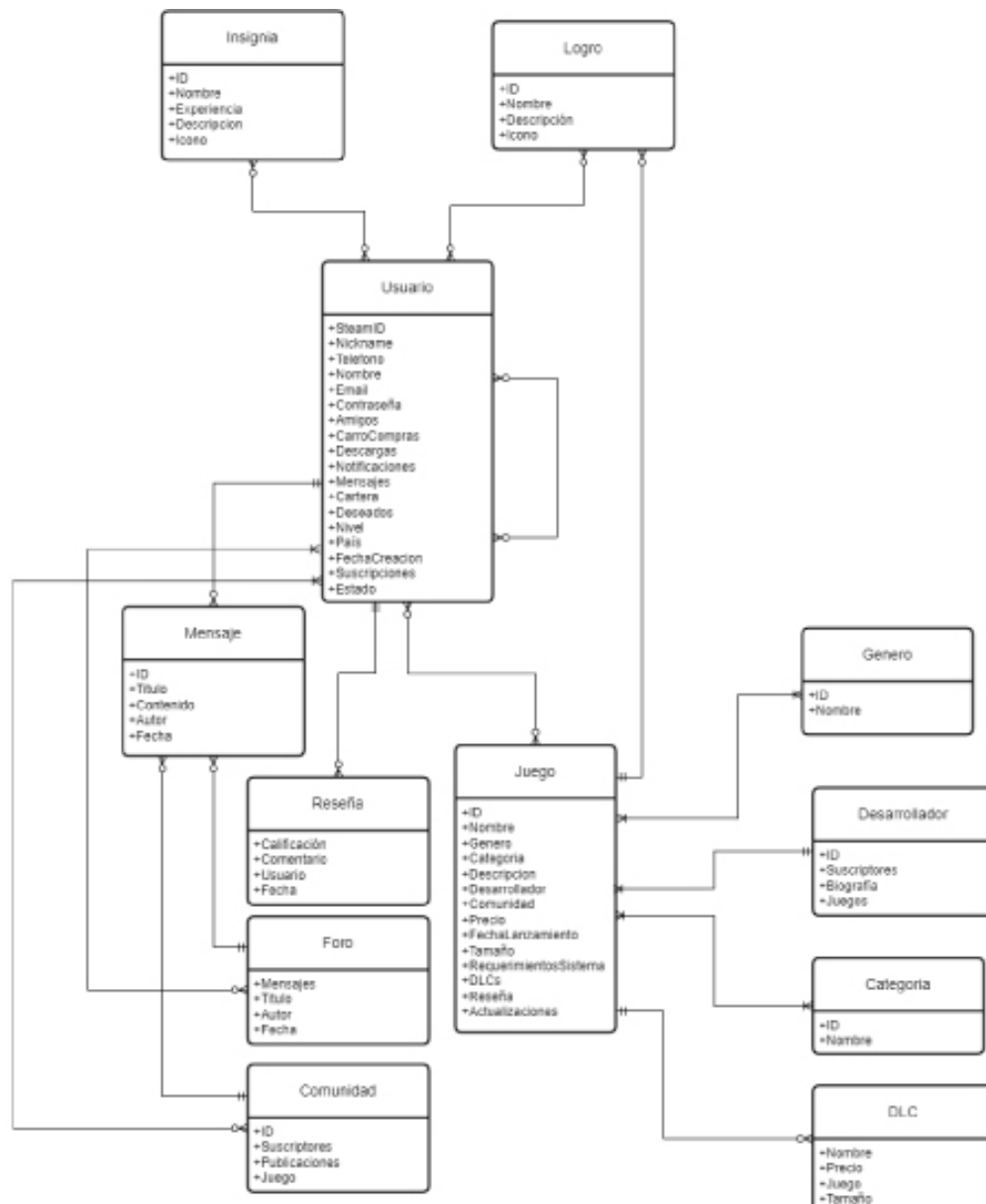


Figure 3: First Entity Relationships Draw

7. First split many to many relationships

8. Second Entity-relationships Draw:

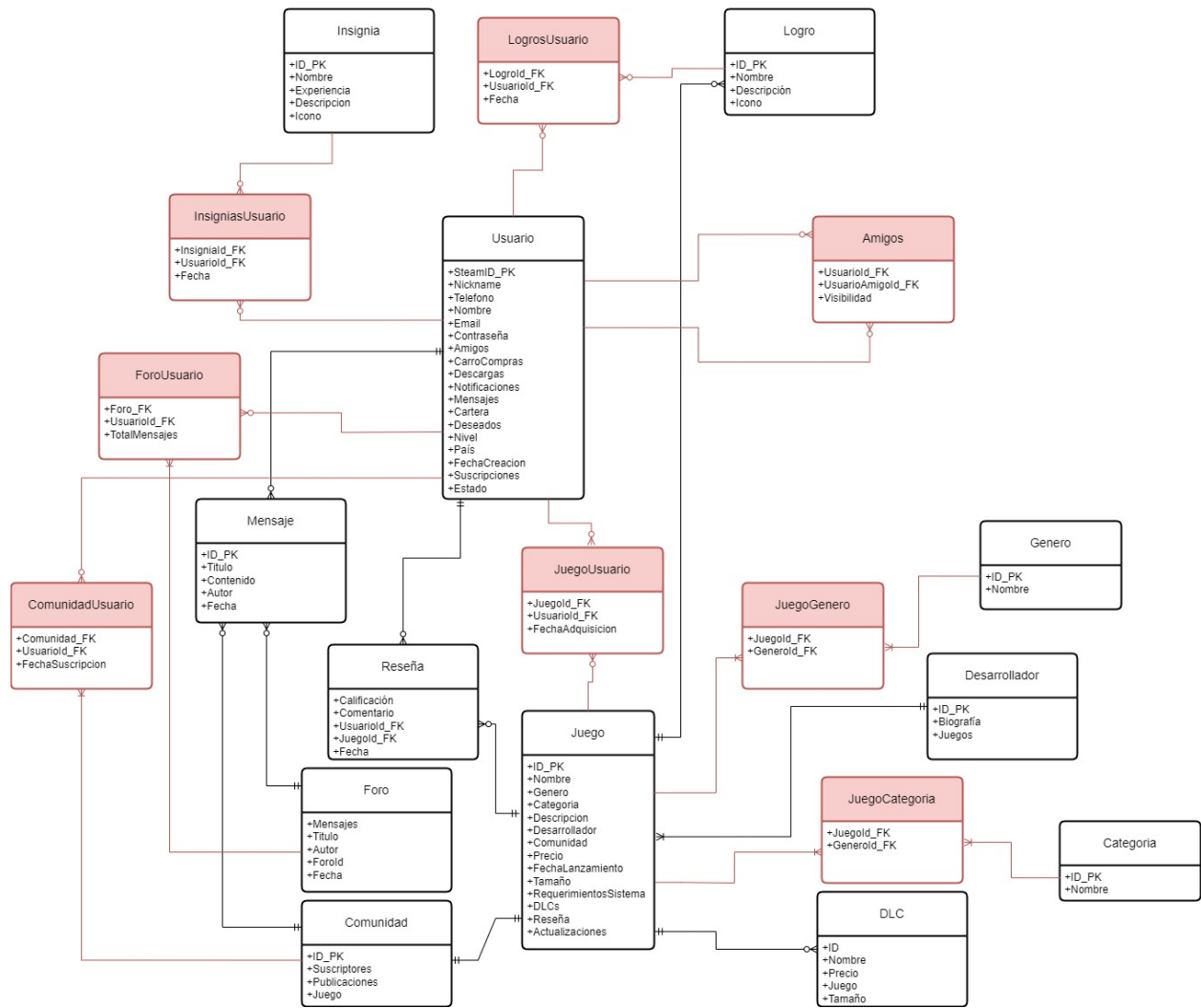


Figure 4: split many to many

9. Get data-structure ERM

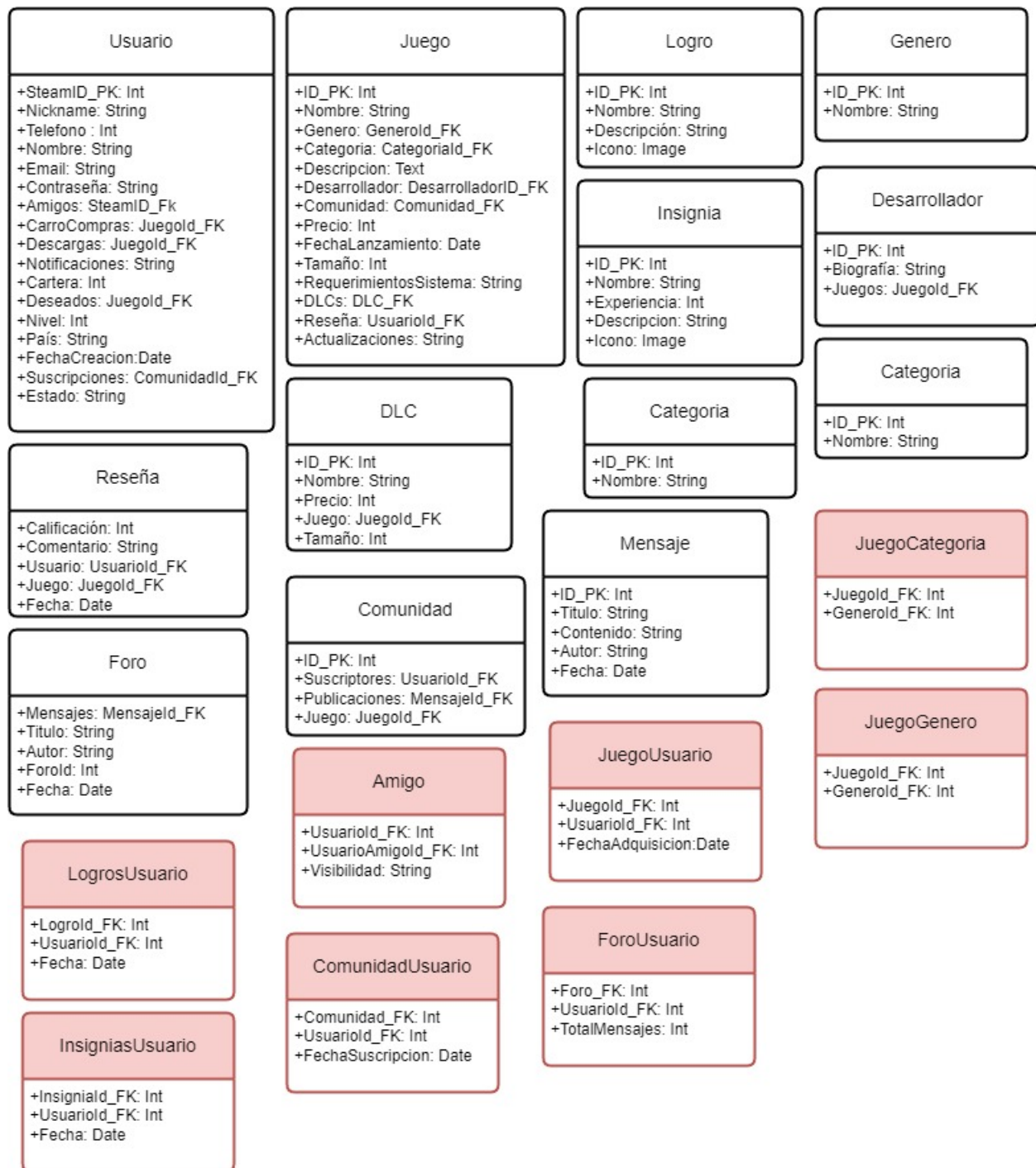


Figure 5: Get data structure ERM

10. Define constrains properties of data

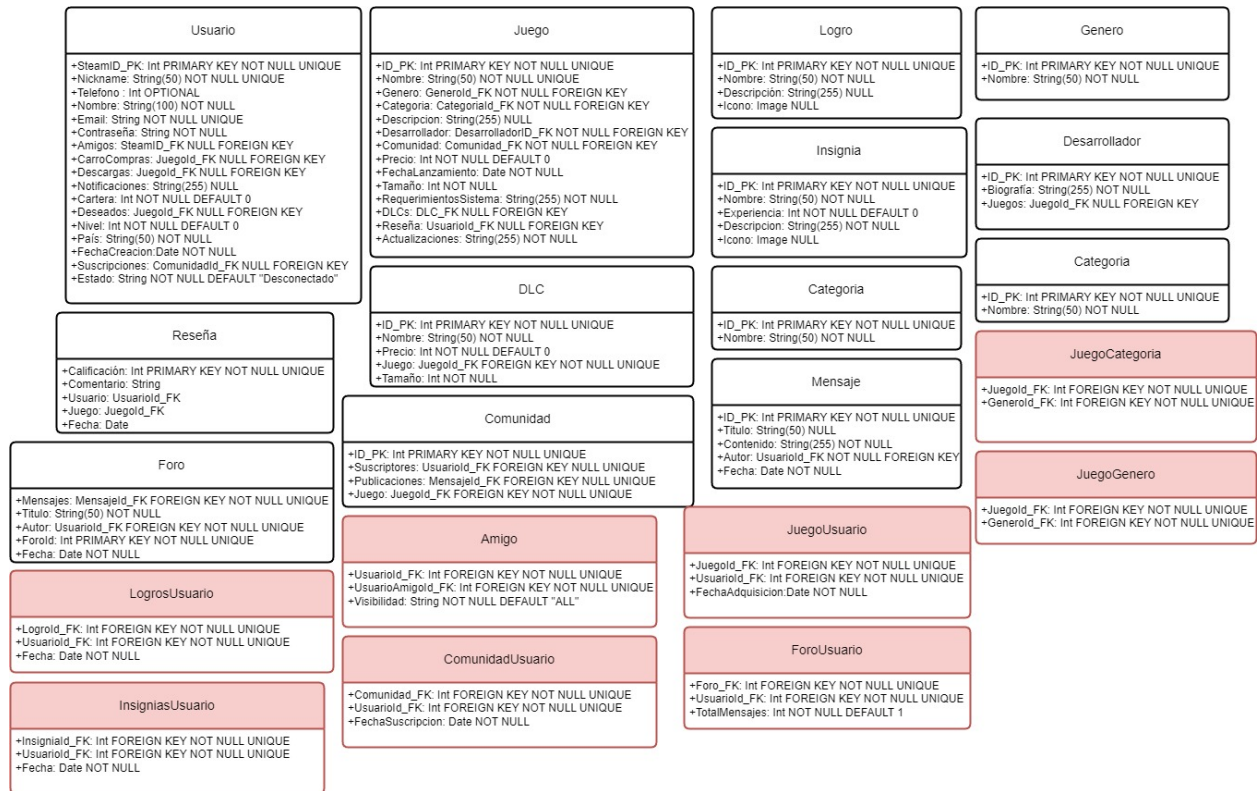


Figure 6: Caption

References

- [1] "E-Commerce History," E-commerce Nation, 2024.
- [2] "DLC - Downloadable Content," Steam Support, Accessed October 2024.
- [3] Valve Corporation, "Steam Community Features," Steam, 2024.