

SQL

Proyecto Final

Comportamiento de los usuarios de Steam

Francisco Paz

Comisión: 39795 – Junio 2023





Proyecto Final “Comportamiento de usuarios de Steam”

Por Francisco Paz

Contenido

1. Descripción de la temática de la BD.....	1.
2. Hipótesis.....	2.
3. Objetivo.....	3.
4. Diagrama Entidad-Relación.....	4
5. Listado de Tablas.....	5.
6. EER.....	6.
7. Archivo SQL.....	7.
8. Importación de registros.....	8.
9. Creación de vistas.....	9.
10. Creación de funciones.....	10.
11. Creación de Stored Procedures.....	11.
12. Creación de tabla bitácora.....	12.
13. Creación de Triggers.....	13.
14. Alta de usuarios.....	14.
15. Sentencias del sublenguaje TCL.....	15.
16. Herramientas y tecnologías usadas.....	16.
17. Futuras líneas.....	17.

1. Descripción de la temática de la base de datos

La utilización de los datos en el del mercado del gaming es muy importante para su desarrollo exponencial. Algunos datos para tener en cuenta:

Un reporte de la firma de análisis de mercado Statista afirma que los videojuegos en 2021 -con un boom especial durante la pandemia- registraron en todo el mundo aproximadamente 2.700 millones de jugadores que, en total, generaron ingresos de hasta u\$u93.200 millones en el último año.

En 2020 el gaming cerró con un valor de alrededor de los u\$s174.900 millones con 3.100 millones de consumidores. Con estos números superó a los sectores del cine y el deporte juntos. Para 2027 se calcula que alcanzará los u\$s340.000 millones. Hoy en la Argentina es una industria de u\$s87 millones, emplea a casi 2.000 desarrolladores de manera directa, y sigue creciendo. [\[Fuente: Ambito.com\]](#)

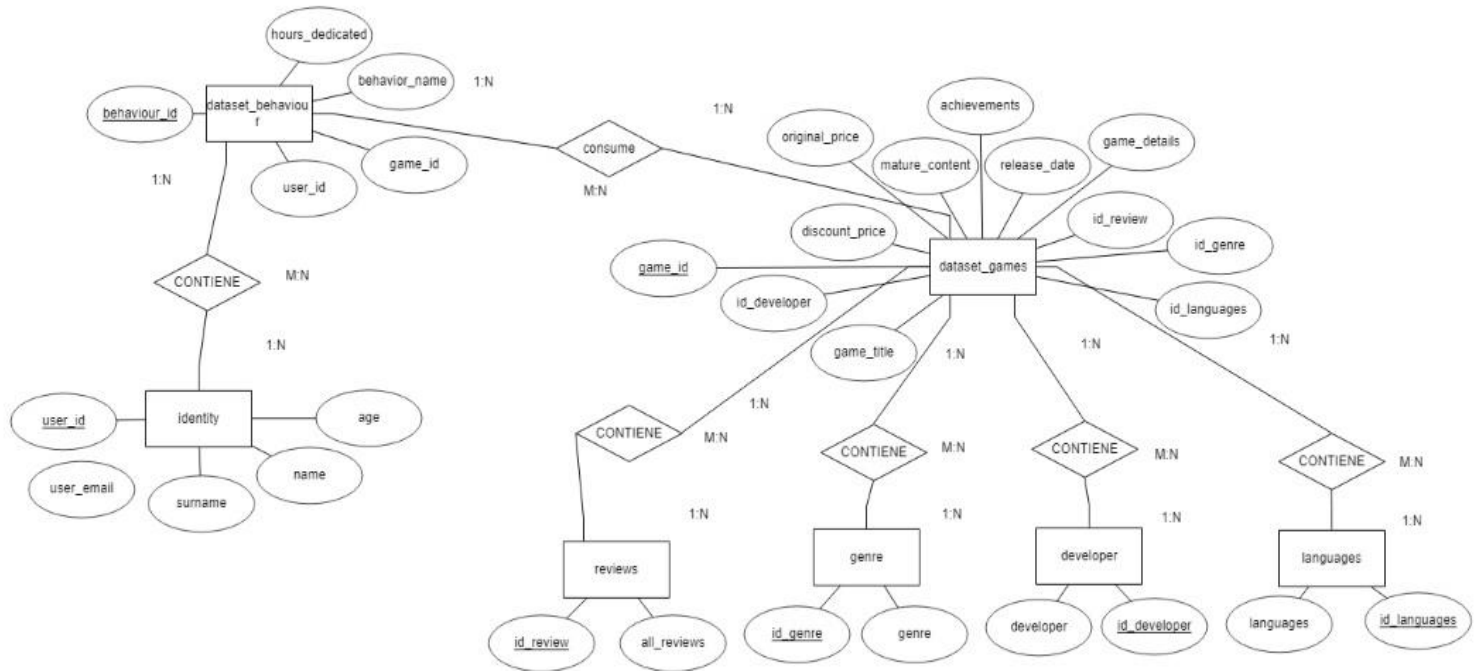
2. Hipótesis

Mediante el resguardo de datos sobre, ventas, comportamiento de usuarios con respecto a los juegos. Los datos pueden brindarnos información valiosa con respecto a futuras ventas dentro de la empresa.

3. Objetivo

Planificar, desarrollar y mantener una base de datos para la firma Steam, la cual pueda brindarle la disponibilidad de los mismos para futuros análisis de mercado.

4. Diagrama Entidad-Relación



5. Listado de columnas de Tablas

5.1 Tabla “dataset_games”: Esta tabla corresponde a los datos de cada juego dentro de la plataforma Steam, con sus detalles más importantes.

dataset_games			
Campo	Tipo de campo	Tipo de clave	Detalle
id_game	INT	PK - index	Codigo de identificacion de Juego - Esta sera la clave primaria NOT NULL AUTO_INCREMENT
id_developer	INT	FK	Codigo de identificacion de Desarrollador - Clave Foranea - Permite la relacion con la tabla "developer" NOT NULL
id_review	INT	FK	Codigo de identificacion de Reseñas generales del juego - Clave Foranea - Permite la relacion con la tabla "reviews" NOT NULL
id_languages	INT	FK	Codigo de identificacion de Languages disponibles del juego - Clave Foranea - Permite la relacion con la tabla "languages" NOT NULL
id_genre	INT	FK	Codigo de identificacion de Genero de juego - Clave Foreanea - Permite la relacion con la tabla "genre" NOT NULL
game_title	Varchar(200)	-	Titulo del juego NOT NULL
release_date	Datetime	-	Fecha de lanzamiento NOT NULL
game_details	Varchar(200)	-	Detalles generales del juego
achievements	INT	-	Logros del juego
id_pegi	INT	-	Categoria para adulto - Se diferenciarian por su id_pegi NOT NULL
original_price	Decimal(18,2)	-	Precios de los juegos, en decimales
discount_price	Decimal(18,2)	-	Descuento en decimales

5.2 Tabla “genre”: Esta tabla corresponde a los géneros detallados de cada juego.

genre			
Campo	Tipo de campo	Tipo de clave	Detalle
id_genre	INT	PK - Index	Codigo de identificacion de Genero de juego - Esta sera la clave primaria. NOT NULL AUTO_INCREMENT
genre	Varchar(100)	-	Genero al que corresponde cada juego. NOT NULL

5.3 Tabla “developer”: Esta tabla corresponde a los nombres listados de los desarrolladores de cada juego.

developer			
Campo	Tipo de campo	Tipo de clave	Detalle
id_developer	INT	PK - index	Codigo de identificacion de Desarrollador - Esta sera la clave primaria. NOT NULL AUTO_INCREMENT
developer	Varchar(200)	-	Nombre del desarrollador del juego. NOT NULL

5.4 Tabla “dataset_behaviour”: Esta tabla corresponde al comportamiento de los usuarios con respecto al consumo de horas en cada juego.

dataset_behaviour			
Campo	Tipo de campo	Tipo de clave	Detalle
id_behaviour	INT	PK - index	Codigo de identificacion que nace de la relacion entre usuario-idjuego consumido - Esta sera la clave primaria NOT NULL AUTO_INCREMENT
id_user	INT	FK	Codigo de identificacion de usuario - Clave foranea - Permite la relacion con la tabla "Identity" NOT NULL
id_game	INT	FK	Codigo de identificacion de juego - Clave foranea - Permite la relacion con la tabla "dataset_games" NOT NULL
hours_dedicated	Decimal(18,2)	-	Valores de horas consumidas por el usuario

5.5 Tabla “review”: Esta tabla corresponde a la recopilación de reseñas hechas por los usuarios a cada juego.

review			
Campo	Tipo de campo	Tipo de clave	Detalle
id_review	INT	PK - index	Codigo de identificacion de Reseñas generales del juego - Esta sera la clave primaria. NOT NULL AUTO_INCREMENT
all_reviews	Varchar(200)	-	Reseñas de los juegos recopilada de opiniones de los usuarios. NOT NULL

5.6 Tabla “languages”: Esta tabla corresponde a los idiomas disponibles para cada juego.

languages			
Campo	Tipo de campo	Tipo de clave	Detalle
id_languages	INT	PK - index	Codigo de identificacion de Languages disponibles del juego - Esta sera la clave primaria. NOT NULL AUTO_INCREMENT
languages	Varchar(200)	-	Lenguajes disponibles para cada juego. NOT NULL

5.7 Tabla “identity”: Esta tabla corresponde a los datos personales de los usuarios.

identity			
Campo	Tipo de campo	Tipo de clave	Detalle
user_id	INT	PK - index	Codigo de identificacion de usuario - Esta sera la clave primaria. NOT NULL AUTO_INCREMENT
id_country	INT	FK	ID del pais usuario. Clave Foreanea NOT NULL
surname	Varchar(200)	-	Apellido del usuario. NOT NULL
name	Varchar(200)	-	Nombre del usuario. NOT NULL
age	Int	-	Edad del usuario. NOT NULL
email	Varchar(200)	-	Email del usuario. NOT NULL

5.8 Tabla “sales”: Esta tabla corresponde a los datos de ventas.

sales			
Campo	Tipo de campo	Tipo de clave	Detalle
id_sale	INT	PK - index	Codigo de identificacion de ventas NOT NULL AUTO_INCREMENT
id_game	INT	FK	Codigo de identificacion de Juego - Clave foránea NOT NULL
id_user	INT	FK	Codigo de identificacion de usuario - Clave foranea NOT NULL
id_seller	INT	FK	Codigo de identificacion de vendedor- Clave foranea NOT NULL
sale_date	DATETIME	-	Fecha de venta NOT NULL
units	INT	-	Cantidad de unidades NOT NULL

5.9 Tabla “seller”: Esta tabla corresponde a los datos de los vendedores.

seller			
Campo	Tipo de campo	Tipo de clave	Detalle
id_seller	INT	PK - index	Codigo de identificacion de vendedor - Clave Primaria NOT NULL AUTO_INCREMENT
name	Varchar(200)		Nombre del vendedor NOT NULL
lastname	Varchar(200)		Apellido del vendedor NOT NULL
email	Varchar(200)		Email del vendedor NOT NULL
age	INT		Edad del vendedor NOT NULL

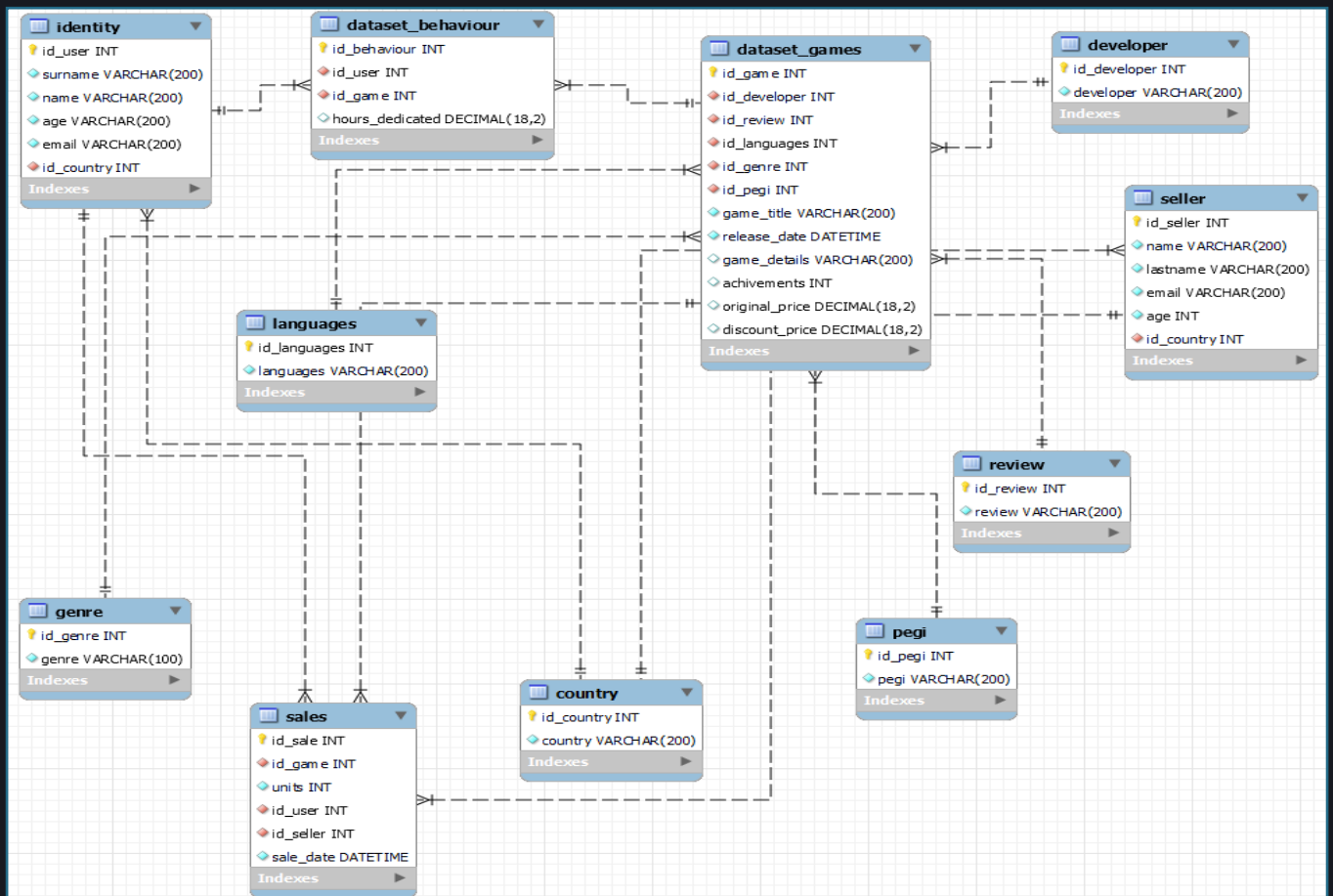
5.10 Tabla “country”: Esta tabla corresponde a los datos de los paises.

country			
Campo	Tipo de campo	Tipo de clave	Detalle
id_country	INT	PK - index	Codigo de identificacion de Pais - Clave Primaria AUTO_INCREMENT
country	Varchar(30)		Nombre del país NOT NULL

5.11 Tabla “pegi”: Esta tabla corresponde a los datos de la clasificación PEGI (sistema de clasificación de videojuegos por edades) .

pegi			
Campo	Tipo de campo	Tipo de clave	Detalle
id_pegi	INT	PK - index	Codigo de identificacion de Clasificacion PEGI - Clave Primaria NOT NULL AUTO_INCREMENT
pegi	Varchar(200)		Clasificacion PEGI NOT NULL

6. EER.

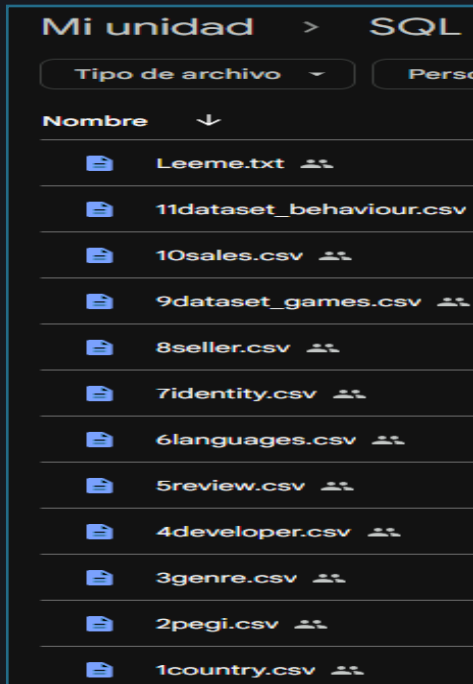


7. Archivo SQL

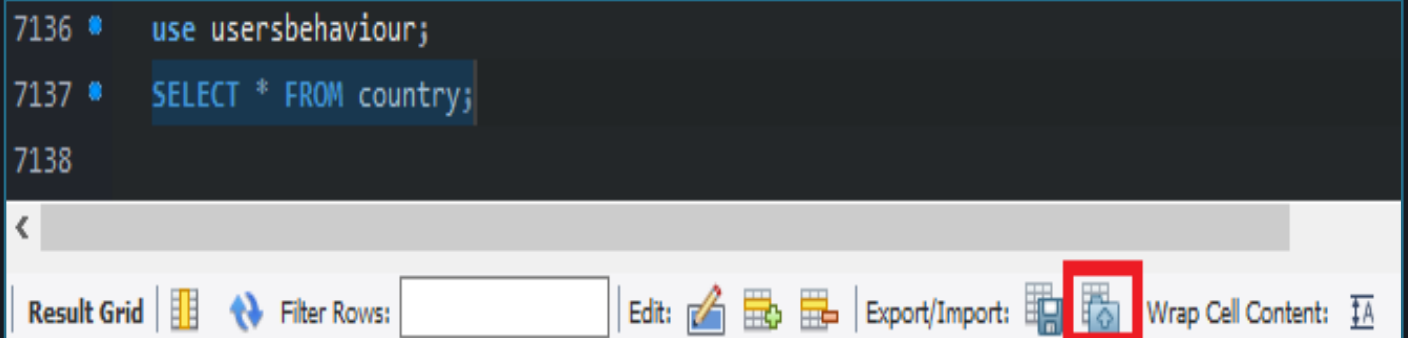
[CLICK ACA.](#)

8. Importación de Registros.

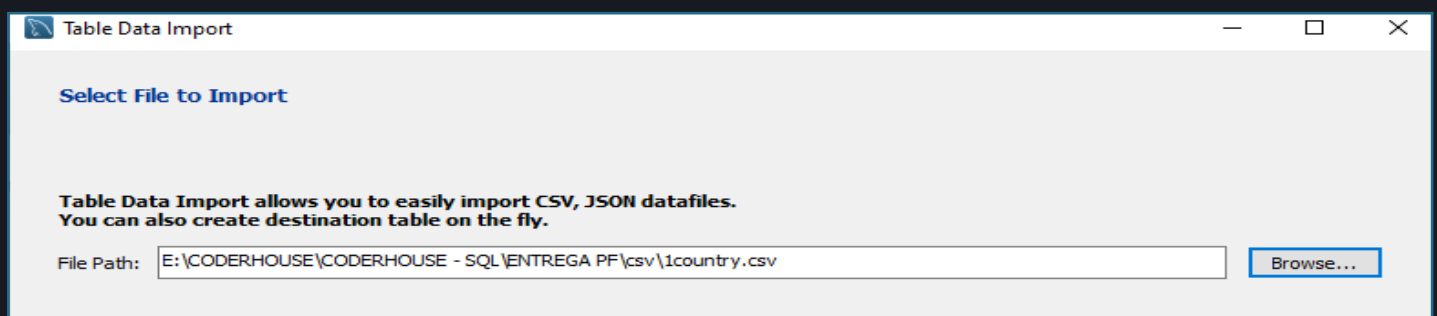
Desde la carpeta de drive, descargamos los archivos de la carpeta “csv”.



Luego desde workbench seleccionamos la tabla a la que insertaremos los datos.



Hacemos click en “import +”
Seleccionaremos la ruta del archivo:



Utilizaremos la opción de tabla existente, le damos a “Next”


Select destination table and additional options.

☒ Use existing table:

☐ Create new table: .

☐ Truncate table before import

Le damos a “Next” hasta completar con la tarea, los archivos están renombrados con números consecutivos para que se identifiquen en el orden que deben cargarse.

Detected file format: csv 

Encoding:

Columns:

<input checked="" type="checkbox"/>	Source Column	Dest Column
<input checked="" type="checkbox"/>	id_country	<input type="text" value="id_countr"/>
<input checked="" type="checkbox"/>	country	<input type="text" value="country"/>

9. Creación de vistas.

```
-- VISTA DE TITULO, FECHA DE LANZAMIENTO Y PRECIO ORIGINAL
CREATE VIEW games_info AS
SELECT game_title, release_date, original_price
FROM dataset_games;

select * from games_info;
```

Result Grid				Filter Rows:	Export:	Wrap Cell Content:
	game_title	release_date	original_price			
►	1... 2... 3... KICK IT! (Drop That Beat Like an U...	2021-09-05 00:00:00	14.29			
	Shadow Warrior Classic (1997)	2022-12-22 00:00:00	104.01			
	Bio Menace	2021-05-07 00:00:00	14.29			
	Crystal Caves	2022-07-23 00:00:00	7.14			14
	Major Stryker	2021-01-30 00:00:00	4.28			
	Monster Bash	2021-12-27 00:00:00	7.14			
	Secret Agent	2022-08-06 00:00:00	7.14			
	Shadow Warrior (Classic)	2022-08-03 00:00:00	14.29			
	Eradicator	2022-04-21 00:00:00	79.96			
	Max Payne	2022-01-27 00:00:00	14.29			
	Alien Carnage / Halloween Harry	2022-03-10 00:00:00	7.14			
	Alien Swarm	2021-04-14 00:00:00	63.21			



```
-- TOP 10 JUEGOS MAS CAROS
CREATE VIEW top_10_expensive_games AS
SELECT id_game, id_developer, id_review, id_languages, id_genre, id_pegi, game_title, release_date, game_details, achievements, original_price, discount_price
FROM dataset_games
ORDER BY original_price DESC
LIMIT 10;

select * from top_10_expensive_games;
```


Result Grid												
Filter Rows: <input type="text"/>												
Export: <input type="button" value="CSV"/> Wrap Cell Content: <input type="button" value="A"/>												
	id_game	id_developer	id_review	id_languages	id_genre	id_pegi	game_title	release_date	game_details	achievements	original_price	
▶	100	305	6	9	5	2	Hollywood Visionary	2022-09-11 00:00:00	Single-player,Steam Achievements,Captions av...	64	717.68	3
	99	360	3	3	10	5	Choice of the Deathless	2021-08-27 00:00:00	Single-player,Steam Achievements,Captions av...	32	717.68	7
	98	643	9	11	2	5	Choice of Robots	2022-10-28 00:00:00	Single-player,Steam Achievements,Captions av...	72	717.68	9
	572	500	4	11	8	2	Cities in Motion	2022-08-31 00:00:00	Single-player,Steam Achievements	47	235.56	4
	567	249	8	10	6	4	CAFE 0 ~The Drowned Mermaid~	2021-09-20 00:00:00	Single-player,Steam Achievements,Steam Tradi...	21	199.95	7
	568	139	5	6	12	2	CAFE 0 ~The Drowned Mermaid~ - Japanese V...	2022-07-03 00:00:00	Single-player,Downloadable Content,Steam Tra...	30	199.95	9
	44	641	8	2	6	4	Chains	2022-10-11 00:00:00	Single-player	30	191.35	6
	542	369	2	8	4	4	Clickteam Fusion 2.5	2022-10-19 00:00:00	Single-player,Multi-player,MMO,Co-op,Shared/...	30	142.99	6
	580	30	6	10	8	2	Car Mechanic Simulator 2015	2021-03-01 00:00:00	Single-player	30	127.18	3
	322	217	8	5	11	3	Gary Grigsby's War in the East	2021-11-25 00:00:00	Single-player,Multi-player	30	114.39	9

```
-- JUEGOS MAS VENDIDOS
```

```
• CREATE VIEW top_selling_games AS  
  SELECT id_game, SUM(units) AS total_units_sold  
  FROM sales  
  GROUP BY id_game  
  ORDER BY total_units_sold DESC;  
  
• select * from top_selling_games;
```

Result Grid |   Filter Rows:

	id_game	total_units_sold
	557	9
	166	8
	104	6
	394	6
	518	6
	201	5
	344	5
	23	5
	485	5
	70	5
	216	5
	232	5
	348	5



top_selling_games 3 x 

Output:


```
-- TOP 5 USUARIOS QUE MAS COMPRARON
```

```
CREATE VIEW top_users AS  
SELECT u.surname, u.name, SUM(s.units) as total_units  
FROM sales s  
JOIN identity u ON s.id_user = u.id_user  
GROUP BY s.id_user  
ORDER BY total_units DESC  
LIMIT 5;
```

```
select * from top_users;
```



Result Grid   Filter Rows: Export

	surname	name	total_units
▶	BOIX GONZALEZ	MARIO	7
	BERENGUERAS CULLERES	MARTA	6
	AGUILAR SUNYE	GERARD	6
	AGUILAR RAMOS	MARTA	5
	AGUILAR RODRIGUEZ	GERARD	5

```
-- TOP 3 VENDEDORES

CREATE VIEW top_sellers AS
SELECT s.id_seller, s.name, s.lastname, SUM(sl.units) AS total_units_sold
FROM seller s
JOIN sales sl ON s.id_seller = sl.id_seller
GROUP BY s.id_seller
ORDER BY total_units_sold DESC
LIMIT 3;

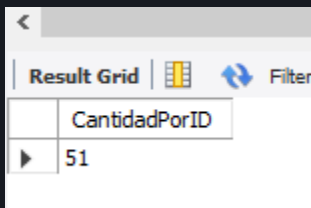
SELECT * from top_sellers;
```

Result Grid   Filter Rows: <input type="text"/> Export				
	id_seller	name	lastname	total_units_sold
▶	18	Madlen	Menis	28
	21	Dianemarie	Stilgo	23
	25	Thedrick	Satterley	21

10. Creación de funciones:

```
-- FUNCION PARA CONTAR CLIENTES POR ID DE PAIS
DELIMITER //
CREATE FUNCTION buscarClientes(country INT)
RETURNS INT
DETERMINISTIC
BEGIN
RETURN (SELECT COUNT(*) FROM identity WHERE country = id_country);
END //

select buscarClientes(5) as CantidadPorID;
```



The screenshot shows a SQL client interface with a 'Result Grid' tab. The grid has one column named 'CantidadPorID' and one row with the value '51'. There are navigation icons and a 'Filter' button above the grid.

	CantidadPorID
▶	51



```

-- FUNCION PARA MOSTRAR RANGO ETARIO
drop function if exists MostraRetario;

DELIMITER //
CREATE FUNCTION MostraRetario(Rango INT)
RETURNS VARCHAR(50)
DETERMINISTIC
BEGIN
CASE
when Rango<11 then
return 'Niñez';
when rango>=11 and rango<18 then
return 'adolescencia';
when rango>=18 and rango<59 then
return 'adultez';
when rango>=59 then
return 'vejez';
end case;
END //

SELECT id_seller, name, lastname, MostraRetario(age) from seller;

```

Result Grid			 Filter Rows:		Ex
	id_seller	name	lastname	MostraRetario(age)	
▶	1	Leesa	Lindenblatt	adultez	
	2	Damian	Quenby	adultez	
	3	Pablo	Le Fleming	adultez	
	4	Urbanus	Sayles	adultez	
	5	Eddie	Matyatin	adultez	
	6	Yalonda	Densell	adultez	
	7	Bartlet	Hatherell	adultez	
	8	Alica	Salvadori	adultez	
	9	Zabrina	Calrow	adultez	
	10	Aigneis	Rosenau	adultez	
	11	Violetta	Slyman	adultez	
	12	Dagmar	Bowser	adultez	
	13	Tara	Barker	adultez	

11. Creación de SP:

```
-- SP DE ORDENAMIENTO DE TABLA CON PRIMER CAMPO, CON EL SEGUNDO CAMPO (DANDO CUALQUIER VALOR ENTRE '') SE ORDENARA DE MANERA DESCENDENTE
-- CASO CONTRARIO, SI NO PUSIERA NINGUN VALOR (SOLO COMILLAS) QUEDARIA COMO VIENE POR DEFECTO EN VALOR ASCENDENTE
DELIMITER ??

CREATE PROCEDURE `orden_param` (IN field CHAR(20), IN param2 VARCHAR(20))
BEGIN
IF field <> '' THEN
    SET @game_order = concat('ORDER BY ', field);
else
    SET @game_order = '';
END IF;
IF param2 <> '' THEN
    SET @game_order2 = ' DESC';
else
    SET @game_order2 = '';
END IF;
SET @clausula = concat('SELECT * FROM dataset_games ', @game_order, @game_order2);
PREPARE runSQL FROM @clausula;
EXECUTE runSQL;
DEALLOCATE PREPARE runSQL;
END ??

CALL orden_param('game_title','');
```

	id_game	id_developer	id_review	id_languages	id_genre	id_peg	game_title	release_date	game_details	achivements	original_price
▶	1	274	9	7	9	3	1... 2... 3... KICK IT! (Drop That Beat Like an U...	2021-09-05 00:00:00	Single-player,Steam Achievements,Steam Tradi...	1	14.29
	90	113	1	9	1	1	10 Second Ninja	2022-07-11 00:00:00	Single-player,Steam Achievements,Steam Tradi...	30	2.85
	127	68	7	1	7	4	100% Orange Juice	2022-07-10 00:00:00	Single-player,Multi-player,Steam Achievements,...	165	10.00
	189	79	9	4	10	5	100000000	2021-10-28 00:00:00	Single-player,Steam Achievements,Steam Tradi...	14	7.14
	91	288	4	5	1	5	140	2021-08-21 00:00:00	Single-player,Steam Achievements,Full controle...	8	7.14
	256	207	6	6	9	3	16bit Trader	2022-07-14 00:00:00	Single-player,Steam Achievements,Steam Tradi...	16	4.28
	294	290	9	1	6	5	1954 Alcatraz	2021-10-01 00:00:00	Single-player,Steam Achievements,Steam Tradi...	25	14.29
	39	56	1	10	8	3	3 Stars of Destiny	2022-05-12 00:00:00	Single-player,Steam Achievements,Full controle...	50	21.44
	303	369	5	1	8	1	3DMark	2021-04-03 00:00:00	Steam Achievements,Steam Leaderboards	15	42.89
	308	16	7	12	7	3	3DMark 11	2022-06-18 00:00:00	Steam Leaderboards	30	28.59
	304	623	5	1	6	3	3DMark API Overhead feature test	2021-11-30 00:00:00	Downloadable Content	30	11.44

```

-- ACTUALIZAR PRECIO DE MANERA PORCENTUAL EN LA TABLA DE DATASET_GAMES DONDE SE ENCUENTRAN LOS VALORES
DELIMITER &&
CREATE PROCEDURE `actualizarPrecio` (IN field decimal(5,2))
BEGIN
  IF field <> '' THEN
    set @game_order = concat('SET original_price = original_price * ', field);
  else
    SET @game_order= '';
  END IF;
  set @clausula = concat('UPDATE dataset_games ', @game_order);
  PREPARE runSQL FROM @clausula;
  EXECUTE runSQL;
  DEALLOCATE PREPARE runSQL;
END &&

-- SI QUISIERA POR EJEMPLO ACTUALIZAR EL PRECIO ORIGINAL UN 10% LLAMARIA AL SP CON EL VALOR 1.1 EJEMPLO:
CALL actualizarPrecio(1.1);

```

12. Creación de tabla bitácora.

```

-- CREACION DE TABLA BITACORA PARA USUARIOS NUEVOS/EDITADOS/ELIMINADOS

drop table if exists new_users;
CREATE TABLE new_users(
  id_action int primary key AUTO_INCREMENT,
  id_user int,
  name varchar(100),
  surname varchar(100),
  action varchar(100),
  dtime datetime,
  admin varchar(100)
);

```


13. Creación de Triggers.

```
-- TRIGGER DE INSERCIÓN DE USUARIO EN TABLA IDENTITY
drop trigger if exists `tr_add_new_user`;
DELIMITER $$
CREATE TRIGGER `tr_add_new_user`
AFTER INSERT ON `identity`
FOR EACH ROW
BEGIN
INSERT INTO `new_users` (id_user, name, surname, action, dtime, admin) VALUES (NEW.id_user, NEW.name, NEW.surname, 'INSERT', NOW(), USER());
END $$
DELIMITER ;

INSERT INTO identity VALUES (1722, 'cece', 'ohhh', 33, 'imbaimba@tutu.com', 5);
select * FROM identity ORDER BY id_user DESC;

select * FROM new_users;
```

```
-- TRIGGER ACTUALIZACIÓN USUARIO EN TABLA IDENTITY
drop trigger if exists tr_update_user;
DELIMITER $$
CREATE TRIGGER `tr_update_user`
BEFORE UPDATE ON `identity`
FOR EACH ROW
BEGIN
INSERT INTO `new_users` (id_user, name, surname, action, dtime, admin) VALUES (NEW.id_user, NEW.name, NEW.surname, 'UPDATE', NOW(), USER());
END $$
DELIMITER ;

UPDATE identity set email = 'ejemplo@ejemplo.com' where id_user = 943;
select * FROM new_users;

-- TRIGGER ELIMINACIÓN DE USUARIO EN TABLA IDENTITY
drop trigger if exists tr_delete_user;
DELIMITER $$
CREATE TRIGGER `tr_delete_user`
AFTER DELETE ON `identity`
FOR EACH ROW
BEGIN
INSERT INTO `new_users` (id_user, name, surname, action, dtime, admin) VALUES (OLD.id_user, OLD.name, OLD.surname, 'DELETE', NOW(), user());
END $$
DELIMITER ;

DELETE FROM identity where id_user = 943;
select * FROM new_users;
```

14. Alta de usuarios

```
-- ALTA DE USUARIOS

-- creacion de usuario 1
drop user if exists usuario1@localhost;
create user usuario1@localhost identified by 'pass123word';

-- creacion de usuario 2
drop user if exists usuario2@localhost;
create user usuario2@localhost identified by 'word123pass';

-- se le otorga al usuario 1 permiso de seleccion, insercion y actualizacion de datos. Sin permiso para borrar
GRANT SELECT, INSERT, UPDATE ON usersbehaviour.* to usuario1@localhost;

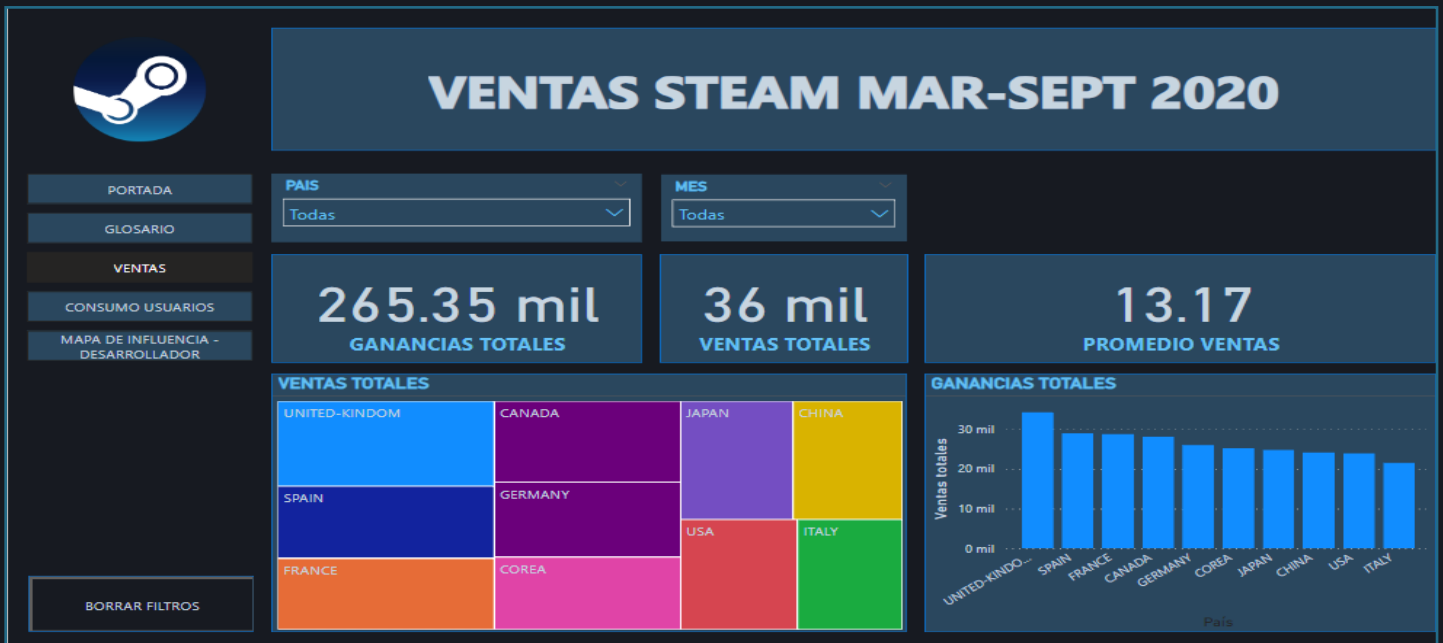
-- se le otorga permisos solo de lectura al usuario 2
GRANT SELECT ON usersbehaviour.* to usuario2@localhost;
```

15. Sentencias del sublenguaje TCL

```
440 -- SENTENCIAS DEL SUBLenguaje TCL
441 -- SE INICIA LA TRANSACCION INSERTANDO UN LENGUAJE MAS EN LA TABLA LANGUAGES, SE COMMITTEA Y SE CONSULTA PARA CORROBORAR.
442
443 • START TRANSACTION;
444 • INSERT INTO LANGUAGES (id_languages, languages)
445   VALUES (NULL,'Czech');
446 • COMMIT;
447 • SELECT * FROM LANGUAGES;
448
449 -- SE INICIA LA TRANSACCION INSERTANDO UN LANGUAGE, PERO SE HACE ROLLBACK, SE CONSULTA LA TABLA PARA CORROBORAR EL ROLLBACK.
450 • START TRANSACTION;
451 • INSERT INTO LANGUAGES (id_languages, languages)
452   VALUES (NULL,'Mandarin');
453 • rollback;
454 • SELECT * FROM LANGUAGES;
455
456 -- SE INICIA LA TRANSACCION PARA INSERTAR DATOS DE UNA VENTA EN LA TABLA SALES, SE COMMITTEA Y SE CONSULTA LA TABLA PARA CORROBORAR.
457
458 • START TRANSACTION;
459 • INSERT INTO sales (id_sale, id_game, units, id_user, id_seller, sale_date)
460   VALUES(NULL, 3, 1,5,1,'2023-05-16');
461 • COMMIT;
462 • SELECT * FROM sales ORDER BY id_sale DESC;
```

16. Informes generados en base a la información de la BD.

Teniendo en cuenta el volumen de información, la sensibilidad de los datos para un análisis de mercado mas acertado, se pueden generar tableros como los siguientes (imágenes abajo), que de hecho fueron generados por mi. Con otro dataset, que por cuestiones de practicidad no eh incluido a la base en si.





17. Herramientas y tecnologías usadas

- .MySQL Workbench 8.0
- .Microsoft 365 (Word-Excel)
- .Adobe Acrobat Reader

18. Futuras líneas

El futuro de mi DB se espera que sea el cual por lo que fue desarrollada, que sea útil para maximizar futuras ventas, teniendo en cuenta toda la información recopilada. Uno espera que los datos seleccionados a capturar sean de utilidad para la gente encargada del marketing de la empresa.