PONTIFICIA UNIVERSIDAD CATÓLICA MADRE Y MAESTRA

Proyecto 1

Análisis de la relación costo-calidad de los juegos de la plataforma de steam a través del tiempo

Presentado por:

Rodolfo Fernandez Fortuna [10135653]

Curso:

Base de Datos II ISC-349

Fecha de entrega:

24 de marzo de 2023

SANTIAGO DE LOS CABALLEROS; REPÚBLICA DOMINICANA

INDICE

PROBLEMÁTICA Y OBJETIVOS

Problemática

Los videojuegos tuvieron un inicio relativamente humilde, siendo vendidos en un nicho reducido como no más que juguetes para niños, y como tal, debían responder a un rango de precio dedicado al sector. Sin embargo, al ser estos tan íntimamente relacionados con la tecnologías, era solo natural que el avance de esta significaba también el avance de los videojuegos. Con estos avances, era inevitable que los videojuegos pasasen a atraer un público mayor, tanto en edad como en tamaño, y a convertirse en un mercado cada vez más competitivo. Por pura ley del mercado, en una competencia emergente, era necesario una mayor inversión, resultando en un aumento en el costo promedio para la creación de un videojuego, y por ende, en el aumento del costo de venta.

En un ambiente así, un 24 de agosto de 1996, fue cuando Gabe Newell, un desarrollador que trabajó para Microsoft durante 13 años, creó Valve como una compañía dedicada a la creación de juegos para PC (para los que existían muy pocos para el momento), y la creación de ports para juegos ya existentes. Tomando esto en mente, y también tomando el hecho de que fueron prácticamente la primera empresa en dedicarse a esto, el aumento económico de Valve se hizo veloz, estableciendo un antes y un después en los videojuegos, pues ya estos nunca más dejarán de considerarse "jueguetes" para ser considerados ahora "softwares".

Con un nuevo mercado emergente, y un nuevo enfoque para el producto, un nuevo problema empezó a surgir, y era que los videojuegos requerían actualizaciones y estos, al ser vendidos en físico, no podían recibirlos. Así fue como en el 2003, desde Valve se creó Steam, como una plataforma multipropósito dedicada a la actualización de software, venta y distribución de juegos y combatir la piratería, siendo, de todos estos aspectos, el de la venta el más importante. Tan solo era de esperarse que una plataforma a la que solo hacía falta conexión a internet para vender un juego superarse rápidamente a las tiendas físicas que requerían de varios materiales y costos de distribución para poder vender su juego.

Con la nueva existencia de Steam, el mercado de los videojuegos empezó aumentar de manera exponencial. Curiosamente, la competencia no aumentó en el área de las plataformas, sino en la del software, dejando a steam como la plataforma dominante por un amplio margen. Gracias a esto es que vaios datos han podido ser obtenidos, que reflejan con un reducido margen de error el comportamiento del mercado de los videojuegos para la PC.

Es así como nació la pregunta que da vida a este análisis, puesto que el aumento de la producción de un videojuego ha ido aumentando considerablemente desde los inicios de Steam, llegando a tener costos actuales que superan al de las películas de hollywood. Siendo esto así, y viendo como la percepción del publico sobre los videojuegos pasó de la de ser un juguete a la de ser un software, ¿cómo ha cambiado el costo de venta de los videojuegos con respecto a su cálidad a traves del tiempo?

Objetivos

Objetivo general:

Entender, por medio de un estudio a los datos de venta públicos de la plataforma de Steam, el comportamiento del mercado de los videojuegos para ordenador y su evolución a lo largo de los años, resultante en su imposición como una gran industria responsable del movimiento de enormes cantidades de dinero.

Objetivos específicos:

- Descubrir la existencia de algún patrón de comportamiento en el costo de venta de los videojuegos en relación a su calidad durante la evolución de su mercado en la plataforma de Steam, tomando como factor de calidad la valoración de los usuarios en el rating dado en la plataforma.

- Demostrar la evolución del comportamiento del mercado de los videojuegos en la plataforma de Steam desde su creación, haciendo énfasis en las compras de videojuegos por género y por temporada.

- Obtener diversas deducciones de los datos obtenidos para sujetarlos a diversos tipos de análisis y conseguir las conclusiones más acertadas posibles.

Diseño y modelado de datos

Descripción del proceso a modelar

Como estudiantes de ingeniería en sistemas y computación, es tan solo natural para nosotros pasar una gran cantidad de nuestro tiempo frente a un ordenador. Bien sea para actividades académicas o para actividades de ocio, nuestros ojos frecuentan miles de pixeles cada día, y el tiempo frente al monitor no hace más que ser mayor y mayor. Es así, como casi por instinto, predominante en la gran mayoría de nuestros colegas, y por supuesto, en nosotros, acudimos a los videojuegos como método principal de distracción.

De cierta forma podría decirse que contamos con una cierta cantidad de experiencia en el ámbito, pues todos los miembros de este estudio han comprado al menos un videojuego en su vida, y sobra decir, este proveniente de Steam. Y así como todos hemos dispuesto de nuestros recursos económicos para adquirir algo que muchos describen como no más que un software -a veces hasta incompleto-, también hemos sido partícipes de la percepción del cambio de los precios de esos bienes que tanto buscamos.

Así es como nace la idea de este estudio, con un dataset de las ventas de Steam hecho público con los valores por juego tan útiles como la cantidad de personas que lo poseen, la valoración en forma de rating que le han dado, la fecha de lanzamiento, el coste inicial, entre otros; y usando varias transformaciones de estos, se busca comprobar si nuestra percepción del aumento de los precios es real o tan solo mera consecuencia de nuestra economía.

Como método principal de modelado tomaremos la relación entre el coste de un videojuego y la valoración del mismo, pues es frecuente que los juegos reciban malas valoraciones si el costo es demasiado alto, pero también que, debido a una barata producción, la valoración también sea baja, por lo que una buena valoración por lo general es indicio de un buen juego en un buen rango de precio.

Como parte de otro método de gran importancia, tomaremos el cambio de los videojuegos en el tiempo, dividiendo este valor en años, y en temporadas, a modo de poder deducir el comportamiento de esta variación. Estos y otros métodos fueron los que usamos para procesar estos datos.

Descripción del nivel de grano

El nivel de grano en el proceso analítico es alto debido a que los datos contenidos en el dataset no contaban con la suficiente granularidad para poder implementar de manera más detallada cada una de las implementaciones

ETL

Datasets usados

Steam es el nombre del dataset que conseguimos donde se encuentran los datos de los registros de cada juego dentro de las plataformas, donde algunos de ellos son el id, el nombre, fecha de lanzamiento, idioma, desarrolladores, entre otras. El dataset fue conseguido aquí: https://www.kaggle.com/datasets/nikdavis/steam-store-games

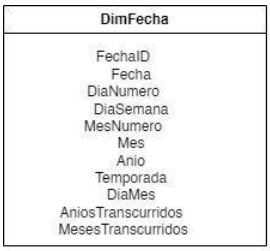
Descripción de la fuente de datos

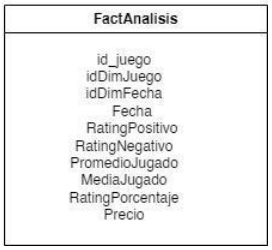
appid: se refiere al id del video juego.
name: se refiere al nombre del video juego.
release_date: se refiere a la fecha de lanzamiento.
english: se refiere al idioma del video juego.
developer: se refiere al nombre de los desarrolladores.
publisher: se refiere a quien lo publico
plataforms: se refiere al sistema operativo que fue lanzado.
required_age: se refiere a la edad mínima requerida para jugar el video juego.
categories: se refiere a la categoría que pertenece.
genres: se refiere al género.

steamspy_tags: se refiere a la etiqueta con los que ellos conocen el video juego.
achievements: se refiere a los logros.
positive_ratings: se refiere al porcentaje de aprobación.
negative_ratings: se refiere al porcentaje desaprobación.
average_ratings: se refiere al promedio de los ratings positivos y negativos.
median_playtime: se refiere al tiempo promedio que cada jugador consume.
owners: se refiere al dueño.
price: se refiere al precio que salió al mercado.

Lista de las estructuras de datos

Nombre	Descripción
steam	Tabla creada para cargar los datos en el dataset
DimFecha	Es la tabla de dimensión creada para generar todas las posibles transformaciones de fecha
vw_DimFecha	Vista que realiza la consulta de los datos existentes en la tabla de fecha
DimJuego	Es la tabla de demisión creada para almacenar todos los datos y posibles transformaciones de los juegos
vw_DimJuego	Vista que realiza la consulta de los datos existentes en la tabla de juego
FactAnalisis	Es la tabla de hechos donde se realizarán los análisis
vw_FactAnalisis	Vista que realiza la consulta de los datos existentes en la tabla de Hechos.





Tablas del modelo analítico

```
select * from DimFecha
```

Results | Messages

FechaID	Fecha	DiaNumero	DiaSemana	MesNumero	Mes	Anio	Temporada	DiaMes	AniosTranscurridos	MesesTranscurridos
1	1950-01-01	1	Sunday	1	January	1950	Invierno	Principio de Mes	73	879
2	1950-01-02	2	Monday	1	January	1950	Invierno	Principio de Mes	73	879
3	1950-01-03	3	Tuesday	1	January	1950	Invierno	Principio de Mes	73	879
4	1950-01-04	4	Wednesday	1	January	1950	Invierno	Principio de Mes	73	879
5	1950-01-05	5	Thursday	1	January	1950	Invierno	Principio de Mes	73	879
6	1950-01-06	6	Friday	1	January	1950	Invierno	Principio de Mes	73	879
7	1950-01-07	7	Saturday	1	January	1950	Invierno	Principio de Mes	73	879
8	1950-01-08	8	Sunday	1	January	1950	Invierno	Principio de Mes	73	879
9	1950-01-09	9	Monday	1	January	1950	Invierno	Principio de Mes	73	879
10	1950-01-10	10	Tuesday	1	January	1950	Invierno	Principio de Mes	73	879
11	1950-01-11	11	Wednesday	1	January	1950	Invierno	Mediados de Mes	73	879
12	1950-01-12	12	Thursday	1	January	1950	Invierno	Mediados de Mes	73	879
13	1950-01-13	13	Friday	1	January	1950	Invierno	Mediados de Mes	73	879
14	1950-01-14	14	Saturday	1	January	1950	Invierno	Mediados de Mes	73	879
15	1950-01-15	15	Sunday	1	January	1950	Invierno	Mediados de Mes	73	879
16	1950-01-16	16	Monday	1	January	1950	Invierno	Mediados de Mes	73	879
17	1950-01-17	17	Tuesday	1	January	1950	Invierno	Mediados de Mes	73	879
18	1950-01-18	18	Wednesday	1	January	1950	Invierno	Mediados de Mes	73	879
19	1950-01-19	19	Thursday	1	January	1950	Invierno	Mediados de Mes	73	879
20	1950-01-20	20	Friday	1	January	1950	Invierno	Mediados de Mes	73	879
21	1950-01-21	21	Saturday	1	January	1950	Invierno	Finales de Mes	73	879
22	1950-01-22	22	Sunday	1	January	1950	Invierno	Finales de Mes	73	879
23	1950-01-23	23	Monday	1	January	1950	Invierno	Finales de Mes	73	879
24	1950-01-24	24	Tuesday	1	January	1950	Invierno	Finales de Mes	73	879
25	1950-01-25	25	Wednesday	1	January	1950	Invierno	Finales de Mes	73	879

```
select * from DimJuego
```

Results | Messages

id	id_juego	Nombre	Desarrollador	Editor	Fecha_lanzamiento	Categorias	Generos
1	10	Counter-Strike	Valve	Valve	2000-11-01	Multi-player;Online Multi-Player;Local Multi-Player;V...	Action
2	20	Team Fortress Classic	Valve	Valve	1999-04-01	Multi-player;Online Multi-Player;Local Multi-Player;V...	Action
3	30	Day of Defeat	Valve	Valve	2003-05-01	Multi-player;Valve Anti-Cheat enabled	Action
4	40	Deathmatch Classic	Valve	Valve	2001-06-01	Multi-player;Online Multi-Player;Local Multi-Player;V...	Action
5	50	Half-Life: Opposing Force	Gearbox Software	Valve	1999-11-01	Single-player;Multi-player;Valve Anti-Cheat enabled	Action
6	60	Ricochet	Valve	Valve	2000-11-01	Multi-player;Online Multi-Player;Valve Anti-Cheat ena...	Action
7	70	Half-Life	Valve	Valve	1998-11-08	Single-player;Multi-player;Online Multi-Player;Steam ...	Action
8	80	Counter-Strike: Condition Zero	Valve	Valve	2004-03-01	Single-player;Multi-player;Valve Anti-Cheat enabled	Action
9	130	Half-Life: Blue Shift	Gearbox Software	Valve	2001-06-01	Single-player	Action
10	220	Half-Life 2	Valve	Valve	2004-11-16	Single-player;Steam Achievements;Steam Trading C...	Action
11	240	Counter-Strike: Source	Valve	Valve	2004-11-01	Multi-player;Cross-Platform Multiplayer;Steam Achiev...	Action
12	280	Half-Life: Source	Valve	Valve	2004-06-01	Single-player	Action
13	300	Day of Defeat: Source	Valve	Valve	2010-07-12	Multi-player;Cross-Platform Multiplayer;Steam Achiev...	Action
14	320	Half-Life 2: Deathmatch	Valve	Valve	2004-11-01	Multi-player;Valve Anti-Cheat enabled;Includes Sourc...	Action
15	340	Half-Life 2: Lost Coast	Valve	Valve	2005-10-27	Single-player;Commentary available	Action
16	360	Half-Life Deathmatch: Source	Valve	Valve	2006-05-01	Multi-player;Valve Anti-Cheat enabled	Action
17	380	Half-Life 2: Episode One	Valve	Valve	2006-06-01	Single-player;Steam Achievements;Captions availabl...	Action
18	400	Portal	Valve	Valve	2007-10-10	Single-player;Steam Achievements;Captions availabl...	Action
19	420	Half-Life 2: Episode Two	Valve	Valve	2007-10-10	Single-player;Steam Achievements;Captions availabl...	Action
20	440	Team Fortress 2	Valve	Valve	2007-10-10	Multi-player;Cross-Platform Multiplayer;Steam Achiev...	Action;Free to Play
21	500	Left 4 Dead	Valve	Valve	2008-11-17	Single-player;Multi-player;Co-op;Steam Achievement...	Action
22	550	Left 4 Dead 2	Valve	Valve	2009-11-19	Single-player;Multi-player;Co-op;Steam Achievement...	Action
23	570	Dota 2	Valve	Valve	2013-07-09	Multi-player;Co-op;Steam Trading Cards;Steam Wor...	Action;Free to Play;Strateg
24	620	Portal 2	Valve	Valve	2011-04-18	Single-player;Co-op;Steam Achievements;Full contro...	Action;Adventure
25	630	Alien Swarm	Valve	Valve	2010-07-19	Single-player;Multi-player;Co-op;Steam Achievement...	Action

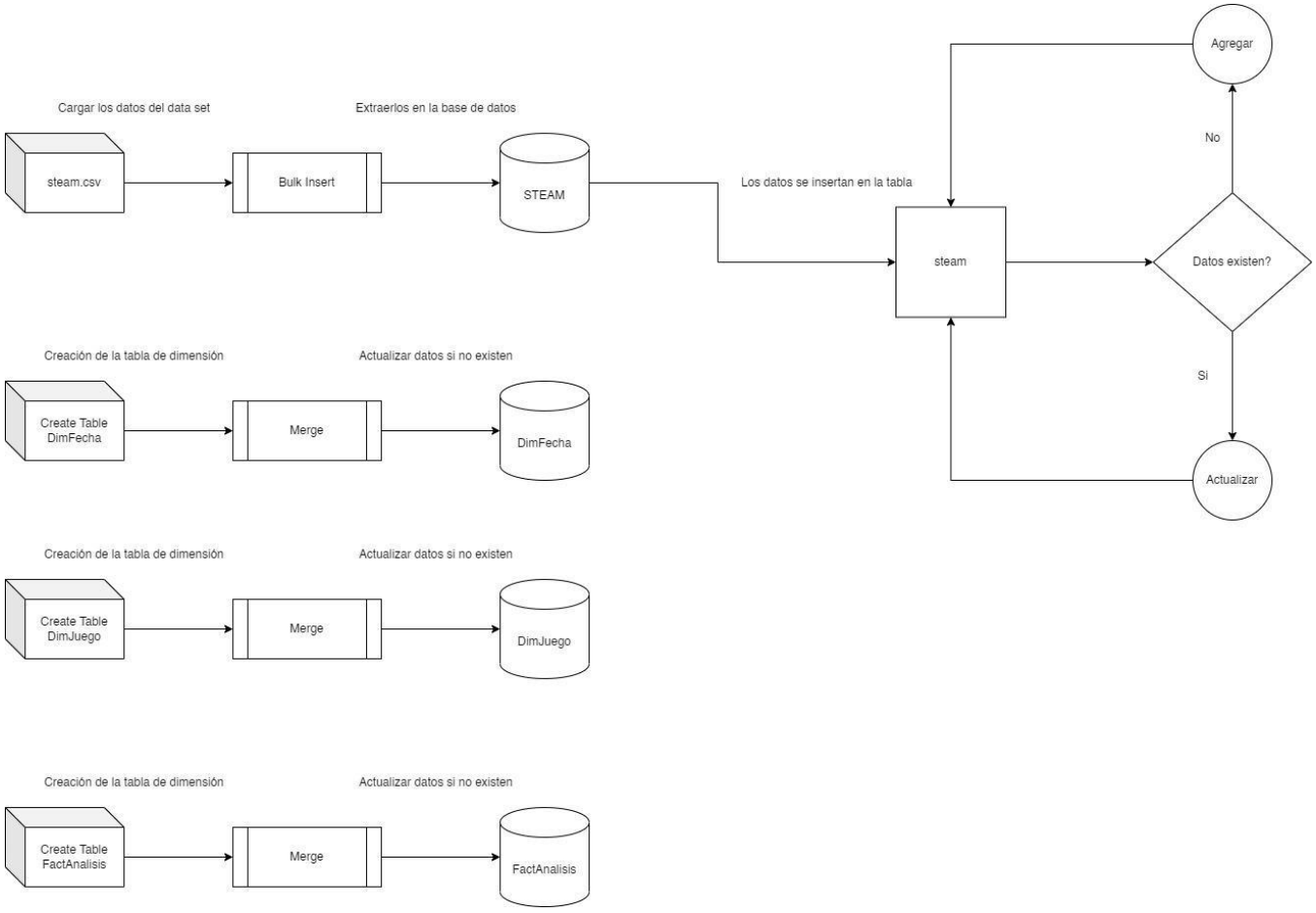
```
select * from FactAnalisis
```

%

Results | Messages

id_juego	idDimJuego	idDimFecha	Fecha	RatingPositivo	RatingNegativo	PromedioJugado	MediaJugado	RatingPorcentaje	Precio
1	10	1	2000-11-01	124534	3339	17612	317	97	7.19
2	20	2	1999-04-01	3318	633	277	62	83	3.99
3	30	3	2003-05-01	3416	398	187	34	89	3.99
4	40	4	2001-06-01	1273	267	258	184	82	3.99
5	50	5	1999-11-01	5250	288	624	415	94	3.99
6	60	6	2000-11-01	2758	684	175	10	80	3.99
7	70	7	1998-11-08	27755	1100	1300	83	96	7.19
8	80	8	2004-03-01	12120	1439	427	43	89	7.19
9	130	9	2001-06-01	3822	420	361	205	90	3.99
10	220	10	2004-11-16	67902	2419	691	402	96	7.19
11	240	11	2004-11-01	76640	3497	6842	400	95	7.19
12	280	12	2004-06-01	3767	1053	190	214	78	0.00
13	300	13	2010-07-12	10489	1210	1356	134	89	7.19
14	320	14	2004-11-01	6020	787	311	32	88	3.99
15	340	15	2005-10-27	5783	1020	46	29	85	0.00
16	360	16	2006-05-01	1362	473	102	81	74	0.00
17	380	17	2006-06-01	7908	517	281	184	93	5.79
18	400	18	2007-10-10	51801	1080	288	137	97	7.19
19	420	19	2007-10-10	13902	696	354	301	95	5.79
20	440	20	2007-10-10	515879	34036	8495	623	93	0.00
21	500	21	2008-11-17	17951	948	897	278	94	7.19
22	550	22	2009-11-19	251789	8418	1615	566	96	7.19
23	570	23	2013-07-09	863507	142079	23944	801	85	0.00
24	620	24	2011-04-18	138220	1891	1102	520	98	7.19
25	630	25	2010-07-19	17435	941	371	83	94	0.00

Esquema de flujo general del ETL



Código y documentación

Es necesario destacar que todos los fragmentos de códigos cargados son ejecutados en script distintos para la eficiente implementación.

Esta primera función realiza el procedimiento de crear la tabla de dimensión "DimFecha" que tiene cómo objetivo realizar las transformaciones relacionadas a las fechas y luego cargar todos los datos generados a la tabla ya creada.

```
CREATE TABLE DimFecha (
   FechaID INT PRIMARY KEY NOT NULL,
   Fecha DATE,
        DiaNumero INT,
   DiaSemana VARCHAR(100),
        MesNumero INT,
```

```
   Mes VARCHAR(100),
   Anio INT,
   Temporada VARCHAR(200),
   DiaMes VARCHAR(200),
         AniosTranscurridos INT,
         MesesTranscurridos INT
);

/*AGREGANDO TODOS LOS DATOS LA PRIMERA VEZ*/
DECLARE @FechaInicio DATE = '19500101';
DECLARE @FechaFin DATE = GETDATE();
DECLARE @FechaActual DATE = @FechaInicio;
DECLARE @FechaID INT = 1;

WHILE @FechaActual <= @FechaFin
BEGIN
      INSERT INTO DimFecha (FechaID, Fecha, DiaNumero, DiaSemana, MesNumero, Mes, Anio,
Temporada, DiaMes, AniosTranscurridos, MesesTranscurridos)
            SELECT @FechaID, @FechaActual, Day(@FechaActual), DATENAME(WEEKDAY,
@FechaActual), MONTH(@FechaActual),
                        DATENAME(MONTH, @FechaActual), YEAR(@FechaActual),
        CASE
            WHEN MONTH(@FechaActual) IN (12,1,2) THEN 'Invierno'
            WHEN MONTH(@FechaActual) IN (3,4,5) THEN 'Primavera'
            WHEN MONTH(@FechaActual) IN (6,7,8) THEN 'Verano'
            ELSE 'Otoño'
        END,
        CASE
            WHEN DAY(@FechaActual) <= 10 THEN 'Principio de Mes'
            WHEN DAY(@FechaActual) <= 20 THEN 'Mediados de Mes'
            ELSE 'Finales de Mes'
        END,
                  DATEDIFF(YEAR, @FechaActual, GETDATE()),
                DATEDIFF(MONTH, @FechaActual, GETDATE());

   SET @FechaID += 1;
   SET @FechaActual = DATEADD(day, 1, @FechaActual);
END
```

Luego se creó en otro script donde se realizó el proceso de crear la tabla de dimensiones DimJuego y posterior a su creación se cargó con todos los datos generados.

```
/*CREANDO LA TABLA DE DIMENSION DE JUEGOS*/
CREATE TABLE DimJuego (
         id INT PRIMARY KEY,
   id_juego INT,
```

```
    Nombre VARCHAR(MAX),
    Desarrollador VARCHAR(MAX),
    Editor VARCHAR(MAX),
    Fecha_lanzamiento DATE,
    Categorias VARCHAR(MAX),
    Generos VARCHAR(MAX),
        PosRating INT,
        NegRating INT,
        AVGJugado INT,
        MediaJugado INT,
        Precio Decimal(18, 2)
)

/*INSERTA  LA PRIMERA VEZ TODO LOS DATOS EN LA TABLA DE DIMENSION DE
JUEGOS*/
INSERT INTO DimJuego (id, id_juego, Nombre, Desarrollador, Editor, Fecha_lanzamiento, Categorias,
Generos, PosRating, NegRating, AVGJugado, MediaJugado, precio)
SELECT
        id,
        appid,
        name,
        developer,
        publisher,
        release_date,
        categories,
        genres,
        positive_ratings,
        negative_ratings,
        average_playtime,
        median_playtime,
        price
FROM Steam;
```

Por último se ejecuta al inicio la creación de la tabla de hechos llamada “FactAnalisis” que es donde se realizan los análisis y se cargan todos los datos ya existentes en las tablas de dimensiones que sean relevantes para esta tabla.

```
/*CREANDO LA TABLA DE HECHOS DE ANALISIS*/
CREATE TABLE FactAnalisis (
    id_juego INT PRIMARY KEY IDENTITY,
        idDimJuego INT,
        idDimFecha INT,
    Fecha DATE,
    RatingPositivo INT,
        RatingNegativo INT,
```

```
    PromedioJugado INT,
    MediaJugado INT,
    RatingPorcentaje INT,
    Precio DECIMAL(18, 2)
)

/*INSERTANDO LA PRIMERA VEZ TODOS LOS DATOS EN LA TABLA DE HECHOS DE ANALISIS*/
INSERT INTO FactAnalisis (
    idDimJuego, idDimFecha, Fecha, RatingPositivo, RatingNegativo, PromedioJugado, MediaJugado, RatingPorcentaje, Precio)
SELECT DimJuego.id_juego, DimFecha.FechaID, DimJuego.Fecha_lanzamiento, DimJuego.PosRating, DimJuego.NegRating,
    DimJuego.AVGJugado, DimJuego.MediaJugado,
    (DimJuego.PosRating * 100 / (DimJuego.PosRating + DimJuego.NegRating)) AS RatingPorcentaje, DimJuego.Precio
FROM DimJuego
INNER JOIN DimFecha ON DimJuego.id = DimFecha.FechaID
```

Creación de la vista de la tabla de dimensiones DimFecha

```
CREATE VIEW vw_DimFecha AS
    SELECT FechaID, Fecha, DiaNumero, DiaSemana, MesNumero, Mes, Anio, Temporada, DiaMes, AniosTranscurridos, MesesTranscurridos
FROM DimFecha
```

Creación de la vista de la tabla de dimensiones DimFecha

```
CREATE VIEW vw_DimJuego AS
    SELECT appid, name, developer, publisher, release_date, categories, genres
FROM steam
```

Creación de la vista de la tabla de dimensiones FactAnalisis

```
CREATE VIEW vw_FactAnalisis AS
    SELECT id, release_date, positive_ratings, negative_ratings, average_playtime, median_playtime,
        (positive_ratings * 100 / (positive_ratings + negative_ratings)) AS RatingPorcentaje, price
FROM steam
```

Actualización de tabla DimFecha

```
MERGE DimFecha AS DF
USING (SELECT * FROM vw_DimFecha) AS view_DF
ON DF.FechaID = view_DF.FechaID
WHEN NOT MATCHED THEN
   INSERT (
                FechaID,
                Fecha,
                DiaNumero,
                DiaSemana,
                MesNumero,
                Mes,
                Anio,
                Temporada,
                DiaMes,
                AniosTranscurridos,
                MesesTranscurridos)
                VALUES (
                view_DF.FechaID,
                view_DF.Fecha,
                view_DF.DiaNumero,
                view_DF.DiaSemana,
                view_DF.MesNumero,
                view_DF.Mes,
                view_DF.Anio,
                view_DF.Temporada,
                view_DF.DiaMes,
                view_DF.AniosTranscurridos,
                view_DF.MesesTranscurridos
);
```

Actualización de la tabla DimJuego

```
MERGE DimJuego AS DJ
USING (SELECT * FROM vw_DimJuego) AS view_DJ
ON DJ.id_juego = view_DJ.appid
WHEN NOT MATCHED THEN
   INSERT (
                id_juego,
                Nombre,
                Desarrollador,
                Editor,
                Fecha_lanzamiento,
                Categorias,
                Generos)
```

```
                VALUES (
                view_DJ.appid,
                view_DJ.name,
                view_DJ.developer,
                view_DJ.publisher,
                view_DJ.release_date,
                view_DJ.categories,
                view_DJ.genres
);
```

Actualización de la tabla de FactAnalisi

```
MERGE FactAnalisis AS FA
USING (SELECT * FROM vw_FactAnalisis) AS view_FA
ON FA.id_juego = view_FA.appid
WHEN NOT MATCHED THEN
   INSERT (
                Fecha,
                RatingPositivo,
                RatingNegativo,
                PromedioJugado,
                MediaJugado,
                RatingPorcentaje,
                Precio)
                VALUES (
                view_FA.release_date,
                view_FA.positive_ratings,
                view_FA.negative_ratings,
                view_FA.average_playtime,
                view_FA.median_playtime,
                (view_FA.positive_ratings 100 / (view_FA.positive_ratings +
view_FA.negative_ratings)),
                view_FA.price
);
```

Principales situaciones

Como era de esperarse, al ser una herramienta completamente nueva para nosotros, tuvimos bastantes dificultades iniciales en el proceso de entender su funcionamiento. No fue sino hasta que pudimos montar los primeros datos de prueba que supimos que el verdadero desafío estaba por comenzar.

No tomó mucho para enfrentarnos a nuestra segunda dificultad, pues muchos de los comandos de SQL no eran reconocidos en power BI, por lo que tuvimos que dedicar tiempo extra al estudio de tutoriales y a la búsqueda de soluciones.

Aplicación y análisis

Casos de análisis

Caso 1

Nombre: Variación del costo de venta promedio de un software en Steam a través del tiempo.

Objetivo: Evidenciar la tendencia económica en el mercado de los videojuegos.

Pregunta: ¿Qué tanto ha aumentado o disminuido los costos promedios desde que se creó Steam?

```
CREATE VIEW P1 AS
SELECT
   YEAR(primer.Fecha) AS AñoCreacion,
   YEAR(ultimo.Fecha) AS AñoUltimo,
   AVG(ultimo.Precio) - AVG(primer.Precio) AS AumentoCostoPromedio
FROM
   FactAnalisis primer
   JOIN FactAnalisis ultimo
      ON primer.id_juego = (SELECT MIN(id_juego) FROM FactAnalisis)
      AND ultimo.id_juego = (SELECT MAX(id_juego) FROM FactAnalisis)
GROUP BY
   YEAR(primer.Fecha),
   YEAR(ultimo.Fecha)
```

1. Se hace un JOIN de la tabla FactAnalisis consigo misma para unir el primer juego y el último juego en una misma consulta.
2. Se utiliza la función YEAR para obtener el año de la fecha del primer juego y del último juego.
3. Se utiliza la función AVG para obtener el costo promedio de todos los juegos en la fecha del primer juego y del último juego.
4. Se resta el costo promedio de la fecha del primer juego al costo promedio de la fecha del último juego, para obtener el aumento de los costos promedios.
5. Se agrupa por año de la fecha del primer juego y del último juego para obtener los resultados deseados.

Caso 2

Nombre: Relación coste-rating.

Objetivo: Tratar de conseguir y entender la relación existente entre el coste de venta de un videojuego y su valoración.

Pregunta 2. ¿Cuál es el monto promedio de los juegos basado en sus ratings?

```
CREATE VIEW P2 AS
SELECT AVG(Precio) AS PromedioPrecio, AVG(RatingPorcentaje) AS PromedioRatingPorcentaje
FROM FactAnalisis
```

1. Para obtener el promedio de los precios de todos los juegos en la tabla:

```
SELECT AVG(Precio) FROM FactAnalisis;
```

2. Para mostrar la columna RatingPorcentaje haciendo un promedio de esta:

```
SELECT AVG(RatingPorcentaje) FROM FactAnalisis;
```

Ambas consultas devuelven un valor numérico, que es el promedio de los precios y el promedio de rating de todos los juegos.

Caso 3

Nombre: Géneros con mayor aceptación entre los usuarios

Objetivo: Identificar si existe alguna inclinación por parte de los usuarios hacia cierto tipo de juegos sobre otros y si esto puede influenciar en su valoración.

Pregunta: ¿Qué géneros de juegos generan mayor aceptación en base a los costos?

```
CREATE PROCEDURE P3 AS
BEGIN
    SELECT dj.Generos, AVG(fa.Precio) as PrecioPromedio, AVG(fa.RatingPorcentaje) as RatingPromedio
    FROM FactAnalisis fa
    JOIN DimJuego dj ON fa.idDimJuego = dj.id_juego
    GROUP BY dj.Generos
    ORDER BY PrecioPromedio DESC
END
```

Esta consulta une ambas tablas en la columna idDimJuego, luego agrupa los resultados por el género de juego
(columna Generos de la tabla DimJuego) y calcula el promedio del Precio (columna Precio de la tabla FactAnalisis)
y el RatingPorcentaje (columna RatingPorcentaje de la tabla FactAnalisis). Luego, ordena los resultados por el
PrecioPromedio de manera descendente para que los géneros con mayores precios promedios aparezcan primero.

Caso 4

Nombre: Popularidad de los géneros acorde al tiempo.

Objetivo: Evidenciar la variabilidad del mercado, y sus cambios a través del tiempo en sus preferencias.

Pregunta 4: ¿Qué géneros tiene mejor aceptación en estos tiempos?

```
CREATE PROCEDURE P4 AS
BEGIN
	SELECT DimJuego.Generos, DimFecha.Fecha, AVG(FactAnalisis.RatingPorcentaje) AS Aceptacion
	FROM FactAnalisis
	JOIN DimJuego ON FactAnalisis.idDimJuego = DimJuego.id_juego
	JOIN DimFecha ON FactAnalisis.idDimFecha = DimFecha.FechaID
	GROUP BY DimJuego.Generos, DimFecha.Fecha
	ORDER BY Aceptacion DESC;
END
```

1. Se seleccionan los atributos Generos y Fecha de las tablas DimJuego y DimFecha, respectivamente, junto con la columna RatingPorcentaje de la tabla FactAnalisis.
2. Se realiza un JOIN entre las tres tablas, utilizando los id correspondientes.
3. Se agrupan los datos por Generos y Fecha.
4. Se utiliza la función AVG para calcular la aceptación promedio de los juegos de cada género en cada fecha.
5. Se ordenan los resultados en orden descendente de acuerdo con la aceptación promedio.

Caso 5

Nombre: Temporada de mejor venta para los videojuegos.

Objetivo: Tratar de descubrir si existe alguna tendencia del mercado inclinada a ciertos momentos del año.

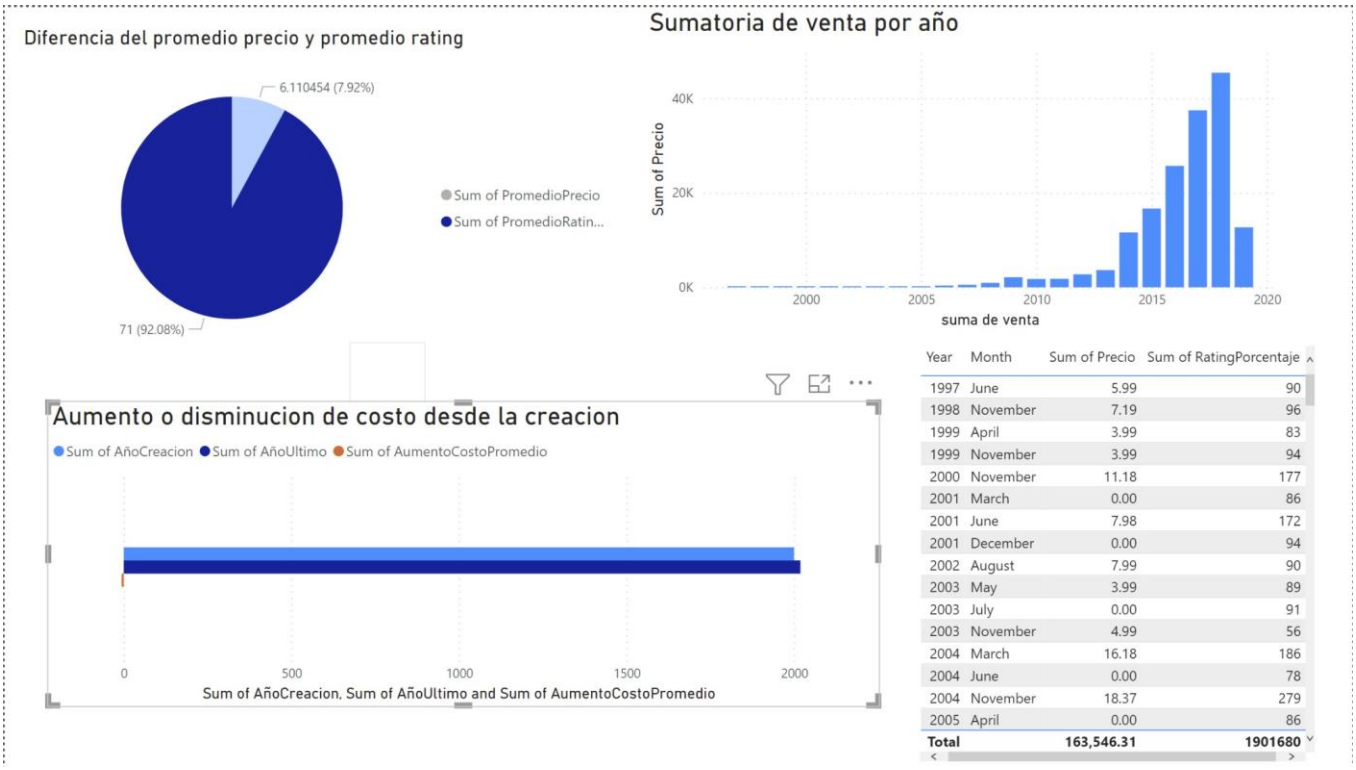
Pregunta: ¿Cuál es la temporada en la que los videojuegos tienen una mayor venta?

```
CREATE PROCEDURE P5 AS
BEGIN
	SELECT TOP (1) df.Temporada, SUM(fa.Precio) AS TotalVentas
	FROM FactAnalisis fa
	JOIN DimFecha df ON fa.idDimFecha = df.FechaID
	GROUP BY df.Temporada
	ORDER BY TotalVentas DESC
END
```

Esta consulta une las tablas FactAnalisis y DimFecha utilizando un join en la columna idDimFecha,

y luego agrupa los resultados por temporada en la tabla DimFecha. La función SUM() se utiliza para calcular la cantidad total vendida (suma de la columna Precio) para cada temporada. Luego, se ordenan los resultados por el total de ventas en orden descendente y se utiliza la cláusula TOP (1) para mostrar sólo la temporada con el mayor total de ventas.

Dashboard de análisis



3 posibles decisiones significativas

1. Considerar si es factible comprar un determinado videojuego tomando como referencia la aceptación basado en el porcentaje de rating.

2. Analizar una posible inversión en una compañía de videojuegos basado en el progreso económico que ha tenido en el paso del tiempo.

3. Las compañías pueden realizar análisis para validar si es rentable para ellos sacar una secuela de un videojuego tomando como referencia la aceptación que ha tenido.

Reflexiones y conclusiones

Todo el proceso de realizar el desarrollo del proyecto junto con todo lo que habíamos visto durante las clases fue muy provechoso pero a la vez muy agotador, todo por lo que fue las investigaciones de comandos y sentencias a utilizar para poder realizar el desarrollo del proyecto, pero también en lo que fue la organización de los datos que se encontraban en el dataset, que no todos estaban de una manera coherente, pues en algunos de los casos se encontraban datos de valores numéricos junto con texto lo que generaba conflicto al momento de realizar la carga.

Otra de las cosas que me llevó un buen tiempo de aprendizaje, búsqueda y comprensión de la situación fue el cargar los datos del dataset al SQLServer. Este siempre me daba error y no comprendía el porqué, hasta después de mucho investigar me di cuenta que se trataba de un bloque por parte de los permisos que se veían configurar.

En cuanto al proyecto como tal conocer cómo ha ido variando con el tiempo las estadísticas de los juegos me pareció bien interesante, ya que en el pasado nunca me detenía a observar este tipo de informaciones y que ahora la comprendo mucho mejor y puedo incluso tomar decisiones basado en lo que ya conozco por el análisis que realizamos.