

Universidad Finis Terrae
Facultad de Ingeniería
Bases de Datos
NRC: 78092

Implementación de una base de datos en la tienda Finis TCGrrae

Autores

José Campodonico
Nicolás Vergara

Profesor

Fernando San Martín

Ayudante

Juan Rivera

Índice

Introducción	3
Requisitos Funcionales	4
Requisitos No Funcionales	4
Modelo Conceptual	5
Modelo Relacional	7
Normalización	7
Primera Forma Normal (1FN)	8
Segunda Forma Normal (2FN)	8
Tercera Forma Normal (3FN)	8
Definición de Claves, Restricciones de Integridad y Optimizaciones	9
Definición de Claves	9
Restricciones de Integridad	9
Optimizaciones	10
Implementación en la base de datos	10
Creación de Tablas	10
Creación de Índices	11
Creación de Vistas	11
Restricciones	11
Carga de Datos	12
Operaciones CRUD	12
Interfaz Gráfica	13

Índice de figuras

1. Modelo Conceptual	5
2. Modelo Relacional	7
3. Tabla normalizada 3FN	9
4. Productos más caros del estante C-03	13
5. Inventario tienda central	14
6. Carrito con 3 productos antes de finalizar	14

7.	Historial de ventas con productos detallados	15
8.	Recepción de nuevo stock	15

Índice de tablas

1.	Requisitos Funcionales	4
2.	Requisitos No Funcionales	4
3.	Resumen de registros insertados	12

Introducción

El presente proyecto está enfocado en un sistema de gestión para una tienda de cartas de One Piece Trading Card Game (OPTCG) en el cual se implementarán los conocimientos del curso de Bases de datos implementando conocimientos de modelos relacionales, normalización e implementación de SQL.

La tienda de Finis TCGrrae es una tienda minorista especializada en juegos de cartas de OPTCG que posee tienda física y digital, donde los clientes pueden comprar diferentes cosas entre ellas: fundas de cartas (sleeves), porta mazos, carpetas, sobres, mazos st (starter deck), cajas (display, illustration box), entre otros. La cual enfrenta desafíos en la gestión de inventarios debido a la gran cantidad de coleccio-nables que posee la franquicia y la organización de torneos competitivos.

Actualmente, el inventario y las ventas se manejan manualmente mediante hojas de cálculo en Excel o cuadernos, lo que limita el acceso remoto y genera inconsistencias en los datos manejados. Los pro-veedores dan a la tienda fechas para liberar los productos, ya que se deben seguir reglas para poder vender.

Los actores principales involucrados son:

- **Clientes:** Jugadores que compran cartas sueltas (singles), display y accesorios.
- **Empleados:** Vendedores y administradores que gestionan stock y eventos.
- **Proveedores:** Distribuidores que envían paquetes de cartas y productos.

Algunos de los desafíos más problemáticos que fueron identificados son:

- Errores en el conteo de stock.
- Errores en el registro de clientes.
- Inscripciones duplicadas en torneos.
- Productos liberados por error debido a fechas mal guardadas.

Estos problemas son comunes en tiendas TCG debido a la complicada gestión manual. Mediante la im-plementación de una solución basada en base de datos relacional se espera que:

- Reducción de errores en inventario
- Reportes semanales para optimizar compras
- Satisfacción del cliente debido a la mejora del sistema

Requisitos Funcionales

ID	Requisito Funcional
RF1	Registrar Producto: Ingresar nuevo producto con atributos: ID único, nombre, precio.
RF2	Gestionar Inventory: cantidad disponible, ubicación.
RF3	Procesar Venta: vincular cliente, lista de productos vendidos, actualización de stock.
RF4	Inscripción de Torneos: Registro del cliente, fecha del evento, tipo torneo y asignar premios.
RF5	Generar Reportes: registro de ventas o compras, torneos realizados.
RF6	Búsqueda de Productos: Filtro al stock nombre/set/rareza/tipo para ventas más rápidas.
RF7	Gestionar Cliente: registrar/actualizar/eliminar clientes con historial.
RF8	Gestionar Empleado: asignar ventas y roles.
RF9	Verificar Liberación: Bloquear ventas antes de fecha liberación.

Tabla 1: Requisitos Funcionales

Requisitos No Funcionales

ID	Requisito No Funcional
RNF1	Rendimiento: Consultas <2 s para hasta 5.000 productos.
RNF2	Usabilidad: Interfaz intuitiva para empleados y clientes.
RNF3	Seguridad: Restricciones para stock negativo y fechas; backups semanales.
RNF4	Escalabilidad: Soporte inicial para 1.000 ítems y que sea expandible.

Tabla 2: Requisitos No Funcionales

Modelo Conceptual

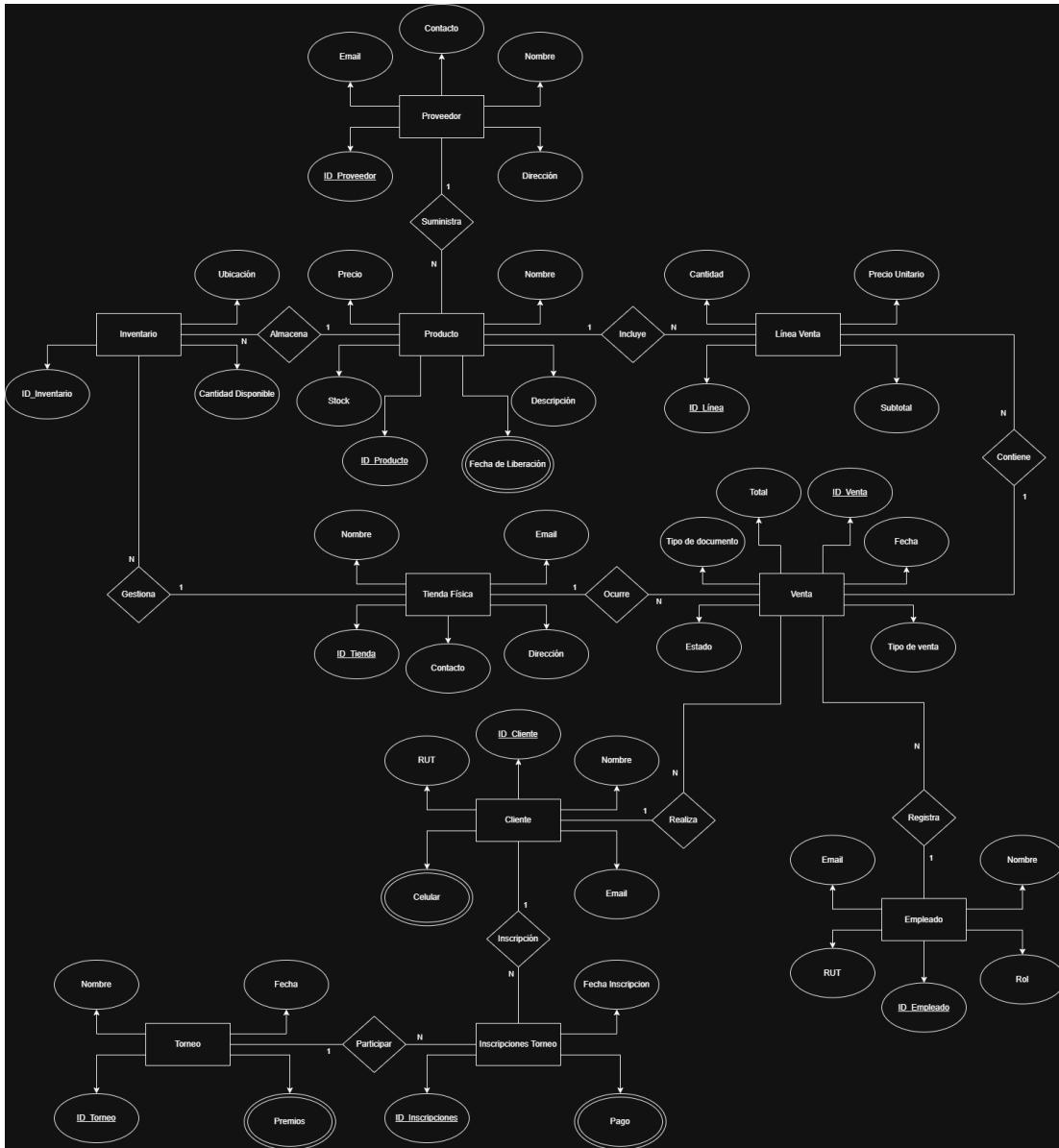


Figura 1: Modelo Conceptual

El diagrama incluye las siguientes entidades principales, cada una con sus atributos y claves primarias subrayadas:

- **Cliente**: Representa a los clientes registrados o anónimos (presenciales).
- **Producto**: Representa los productos TCG (cartas, accesorios, etc.).
- **Venta**: Representa las transacciones realizadas.

-
- **Tienda_Física**: Representa las sucursales físicas.
 - **Inventario**: Gestiona el stock por tienda.
 - **Proveedor**: Representa a los distribuidores.
 - **Empleado**: Representa a vendedores y administradores.
 - **Inscripción_Torneo**: Registra la participación en torneos.
 - **Torneo**: Representa eventos gratuitos o de pago.
 - **Línea_Venta**: Detalla los productos en cada venta.

Las relaciones entre las entidades se representan mediante rombos, con las siguientes cardinalidades:

- **Cliente (1:N) Venta**: Un cliente puede realizar muchas ventas, pero cada venta está asociada a un solo cliente.
- **Producto (1:N) Línea_Venta**: Un producto puede estar en muchas líneas de venta, pero cada línea se refiere a un solo producto.
- **Venta (1:N) Línea_Venta**: Una venta puede contener muchas líneas, pero cada línea pertenece a una sola venta.
- **Cliente (1:N) Inscripción (1:N) Torneo**: Un cliente puede participar en varios torneos pero solamente inscribirse una vez en cada torneo y un torneo puede tener varios clientes.
- **Empleado (1:N) Venta**: Un empleado puede registrar muchas ventas, pero cada venta la registra un solo empleado.
- **Proveedor (1:N) Producto**: Un proveedor puede suministrar muchos productos, pero cada producto tiene un solo proveedor.
- **Producto (1:N) Inventario**: Un producto puede tener varios registros de inventario, pero cada registro se refiere a un solo producto.
- **Tienda_Física (1:N) Inventario**: Una tienda puede gestionar muchos registros de inventario, pero cada registro pertenece a una tienda.
- **Tienda_Física (1:N) Venta**: Una tienda puede tener muchas ventas presenciales, pero cada venta presencial está asociada a una tienda.

Modelo Relacional

Con el modelo conceptual creado se llevó a cabo la transformación de cada entidad y relación en tablas relacionales, definiendo claves primarias (PK) y foráneas (FK) según sus cardinalidades.

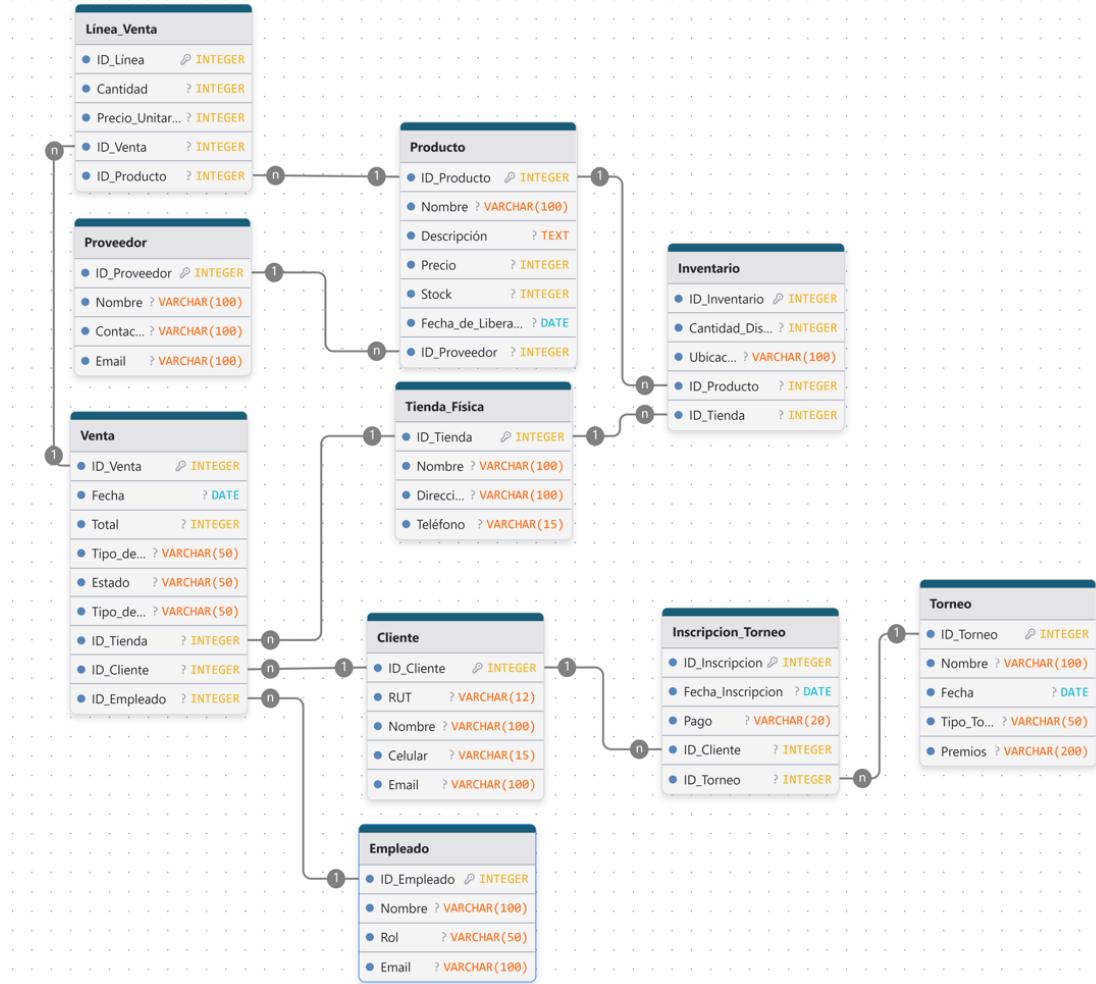


Figura 2: Modelo Relacional

Normalización

El proceso de normalización se realizó sobre el modelo inicial para garantizar la integridad de los datos y optimizar el almacenamiento, alcanzando la Tercera Forma Normal (3FN).

Primera Forma Normal (1FN)

Eliminación de grupos repetitivos y atomicidad de atributos.

Se analizó la entidad **Venta**. Inicialmente, se podría haber considerado almacenar la lista de productos vendidos en un solo campo de texto (ej. "2x Mazos, 1x Caja"). Para cumplir con 1FN, se separaron estos datos creando la entidad débil **Línea_de_venta**.

Cada registro en **Línea_de_venta** contiene un único producto asociado a una venta, garantizando la atomicidad.

Segunda Forma Normal (2FN)

Eliminación de dependencias parciales (atributos que dependen solo de una parte de una clave primaria compuesta).

En nuestro modelo, todas las tablas principales (**Producto**, **Cliente**, **Venta**, etc.) utilizan una clave primaria subrogada artificial (**AUTOINCREMENT**) y simple (ej. **ID_Producto**, **ID_Venta**) en lugar de claves compuestas.

Al no existir claves primarias compuestas en las entidades fuertes, no es posible que existan dependencias parciales. Por ejemplo, en **Línea_de_venta**, aunque conceptualmente vincula **Venta** y **Producto**, su PK es **ID_Línea_Venta**, por lo que el precio unitario depende totalmente de esta ID y no parcialmente de la venta o el producto.

Tercera Forma Normal (3FN)

Eliminación de dependencias transitivas (atributos que dependen de otros atributos no clave).

Se identificó una dependencia transitiva crítica en la gestión de productos.

Si la tabla **Producto** incluyera campos como **Nombre_Proveedor** o **Contacto_Proveedor**, existiría la dependencia: $\text{ID_Producto} \rightarrow \text{ID_Proveedor} \rightarrow \text{Contacto_Proveedor}$.

Se extrajo la información del proveedor a su propia tabla independiente **Proveedor**. La tabla **Producto** mantiene únicamente la clave foránea **ID_Proveedor**.

Esto permite actualizar el contacto o email de un distribuidor (ej. "TCG Distributor LATAM") en un solo registro, sin tener que modificar los 42 productos asociados en el inventario.

Linea de venta				
ID_Linea_Venta	Cantidad	Precio Unitario	ID_Venta	ID_Producto
Proveedor				
ID_Proveedor	Nombre	Contacto	Email	
Cliente				
ID_Cliente	Rut	Nombre	Celular	Email
Empleado				
ID_Empleado	Nombre	Rol	Email	
Producto				
ID_Producto	Nombre	Descripcion	Precio	Stock
Tienda física				
ID_Tienda	Nombre	Direccion	Telefono	
Inscripcion Torneo				
ID_Incripcion_Torneo	Fecha inscripcion	Pago	ID_Cliente	ID_Torneo
Torneo				
ID_Torneo	Nombre	Fecha	Tipo Torneo	Premios
Inventario				
ID_Inventario	Cantidad disponible	Ubicacion	ID_Producto	ID_Tienda
Venta				
ID_Venta	Fecha	Total	Tipo de venta	Estado
ID_Venta	Fecha	Total	Tipo de documento	ID_Tienda
ID_Venta	Fecha	Total	ID_Cliente	ID_Empleado

Figura 3: Tabla normalizada 3FN

Definición de Claves, Restricciones de Integridad y Optimizaciones

Tras normalizar el modelo relacional hasta la Tercera Forma Normal (3FN), se definieron las claves, se establecieron restricciones de integridad y se propusieron optimizaciones para garantizar la consistencia, seguridad y eficiencia de la base de datos. A continuación, se detalla el proceso por cada tabla.

Definición de Claves

Se identificaron las claves primarias (PK) y foráneas (FK) para cada tabla, asegurando la unicidad y las relaciones entre entidades:

Restricciones de Integridad

Se definieron restricciones para garantizar la consistencia y validez de los datos:

- **Integridad de entidad:** Cada clave primaria (**ID_***) debe ser NOT NULL y UNIQUE.
- **Integridad referencial:** Las claves foráneas deben referenciar claves primarias existentes o aceptar NULL donde sea aplicable (ej. ID_Cliente en Venta puede ser NULL para ventas anónimas).

-
- **Restricciones específicas por tabla:**
 - **Línea de venta:** **Cantidad** >0 , **Precio Unitario** ≥ 0 .
 - **Producto:** **Precio** >0 , **Stock** ≥ 0 , **Fecha liberación** debe ser una fecha válida.
 - **Inventario:** **Cantidad disponible** ≥ 0 .
 - **Venta:** **Total** ≥ 0 , **Fecha** debe ser una fecha válida y no futura respecto a la fecha actual.
 - **Inscripción Torneo:** **Pago** ≥ 0 , **Fecha inscripción** debe ser anterior o igual a la fecha del torneo.
 - **Restricciones de dominio:** Atributos como **Rut**, **Celular**, **Email** deben cumplir formatos específicos (ej. Rut chileno, número de teléfono válido, email con @).

Optimizaciones

Se propusieron optimizaciones para mejorar el rendimiento en donde se crearon índices en claves foráneas y atributos frecuentemente consultados (ej. ID_Cliente, ID_Producto, ID_Venta) para acelerar consultas JOIN y búsquedas.

Implementación en la base de datos

Se implementó el modelo relacional normalizado en **SQLite**, definiendo tablas, vistas, índices y restricciones para garantizar la integridad, consistencia y eficiencia de la base de datos. Se utilizaron los tipos de datos nativos de SQLite (**INTEGER**, **TEXT**, **REAL**) y se configuraron claves primarias autoincrementales.

Creación de Tablas

Se crearon las siguientes tablas con claves primarias autoincrementales, foráneas y restricciones básicas:

- **Proveedor** (ID_Proveedor, Nombre, Contacto, Email)
- **Cliente** (ID_Cliente, Rut, Nombre, Celular, Email)
- **Empleado** (ID_Emppleado, Nombre, Rol, Email)
- **Producto** (ID_Producto, Nombre, Descripción, Precio, Stock, Fecha_liberación, ID_Proveedor)
- **Tienda_física** (ID_Tienda, Nombre, Dirección, Teléfono)
- **Torneo** (ID_Torneo, Nombre, Fecha, Tipo_Torneo, Premios)
- **Inscripción_Torneo** (ID_Inscripción_Torneo, Fecha_inscripción, Pago, ID_Cliente, ID_Torneo)

-
- **Inventario** (ID_Inventario, Cantidad_disponible, Ubicación, ID_Producto, ID_Tienda)
 - **Venta** (ID_Venta, Fecha, Total, Tipo_de_venta, Estado, Tipo_de_documento, ID_Tienda, ID_Cliente, ID_Emppleado)
 - **Línea de venta** (ID_Línea_Venta, Cantidad, Precio_Unitario, ID_Venta, ID_Producto)

Las claves primarias se definieron como **INTEGER PRIMARY KEY AUTOINCREMENT** para generar identificadores únicos automáticamente. Los tipos de datos se ajustaron a las capacidades de SQLite: **INTEGER** para IDs y cantidades, **REAL** para valores monetarios, y **TEXT** para fechas y cadenas de texto.

Creación de Índices

Se crearon índices para optimizar el rendimiento en consultas frecuentes:

- **idx_venta_cliente** en **Venta(ID_Cliente)**
- **idx_venta_tienda** en **Venta(ID_Tienda)**
- **idx_venta_empleado** en **Venta(ID_Emppleado)**
- **idx_linea_venta** en **Línea de venta(ID_Venta, ID_Producto)**
- **idx_inventario_producto** en **Inventario(ID_Producto)**
- **idx_inscripcion_cliente** en **Inscripción_Torneo(ID_Cliente)**
- **idx_inscripcion_torneo** en **Inscripción_Torneo(ID_Torneo)**

Creación de Vistas

Se diseñaron vistas para facilitar reportes frecuentes:

- **Vista_Ventas_Cliente**: Muestra el número de ventas y monto total por cliente.
- **Vista_Inventario_Tienda**: Muestra el stock disponible por tienda y producto.

Restricciones

Se implementaron restricciones básicas soportadas por SQLite:

- **NOT NULL**: En atributos obligatorios como Nombre, Rut, Precio.
- **UNIQUE**: En Email y Rut para evitar duplicados.
- **FOREIGN KEY**: Vincula tablas relacionadas (ej. **ID_Proveedor** en **Producto**).
- **DEFAULT**: Para valores por defecto como **Stock = 0, Pago = 0**.

Carga de Datos

Se realizó la inserción de datos de prueba en todas las tablas de la base de datos **finis_tcgrae.db**, asegurando la coherencia entre las relaciones y la integridad referencial.

Se utilizaron 42 productos, 10 clientes, 5 ventas, 14 inscripciones en torneos y un inventario organizado en tres estantes físicos.

Tabla	Registros	Notas
Cliente	10	9 con RUT + 1 presencial
Empleado	3	Roles: Reponedor, Encargado, Vendedor
Proveedor	2	TCG Distributor y Bandai Namco
Tienda_fisica	1	Finis TCGrae Central
Producto	42	Accesorios, cajas, ST y singles
Inventario	42	Estantes A-01, B-02, C-03
Torneo	2	Pago y Gratuito
Inscripcion_Torneo	14	5 pagas, 9 gratuitas
Venta	5	Online y presencial
Línea_de_venta	7	Detalle de productos vendidos

Tabla 3: Resumen de registros insertados

El inventario se organizó en tres estantes para optimizar el espacio físico:

- **Estante A-01:** Accesorios (sleeves, carpetas, portamazos) – 10 productos
- **Estante B-02:** Cajas selladas (OP, EB, PRB) – 12 productos
- **Estante C-03:** Starter Decks y singles – 20 productos

Operaciones CRUD

Se implementaron las operaciones básicas de **Create, Read, Update y Delete** sobre las principales entidades del sistema. A continuación, se presentan ejemplos de sus aplicaciones en **DB Browser for SQLite**.

Create (INSERT)

Insención de un nuevo cliente Online

```
INSERT INTO "Cliente" ("Nombre", "Rut", "Celular", "Email")
VALUES ("Leoncio\u00d1Gutierrez", "19872323-2", "+56922222111", "leoguti@gmail.com");
```

Read (SELECT)

Listar productos en el estante C-03 (singles y ST)

```
SELECT p."ID_Producto", p."Nombre", p."Precio", i."Cantidad_disponible"
FROM "Producto" p
JOIN "Inventario" i ON p."ID_Producto" = i."ID_Producto"
WHERE i."Ubicacion" = 'Estante\u2022C-03'
ORDER BY p."Precio" DESC
LIMIT 5;
```

	ID_Producto	Nombre	Precio	Cantidad_disponible
1	23	Starter Deck ST-11 - Uta	12900.0	20
2	24	Starter Deck ST-12 - Zoro & Sanji	12900.0	18
3	25	Starter Deck ST-13 - The Three ...	12900.0	15
4	26	Starter Deck ST-14 - 3D2Y	12900.0	14
5	27	Starter Deck ST-15 - Red Edward ...	12900.0	12

Figura 4: Productos más caros del estante C-03

Update (UPDATE)

Actualizar stock tras una venta

```
-- Venta de 1 unidad del producto ID 11 (Booster Box OP06)
UPDATE "Producto"
SET "Stock" = "Stock" - 1
WHERE "ID_Producto" = 11;
```

-- También se actualiza el inventario

```
UPDATE "Inventario"
SET "Cantidad_disponible" = "Cantidad_disponible" - 1
WHERE "ID_Producto" = 11;
```

Delete (DELETE)

Eliminar una inscripción cancelada

```
-- Cliente 9 cancela su inscripción al torneo gratuito
DELETE FROM "Inscripcion_Torneo"
WHERE "ID_Cliente" = 9 AND "ID_Torneo" = 2;
```

Interfaz Gráfica

Se desarrolló una interfaz gráfica con **Python** y **Tkinter** que permite gestionar las operaciones principales del sistema: inventario, ventas, historial y recepción de stock. El archivo de base de datos **w1.db** se conecta directamente desde la misma carpeta del script.

Las principales funcionalidades de la interfaz que fueron implementadas son:

- **Inventario:** Visualización por estante (A-01, B-02, C-03)
- **Carrito de compras:** Múltiples productos en una sola venta
- **Ventas realizadas:** Historial con detalle de productos
- **Recepción de stock:** Ingreso de productos del proveedor

Finis TCGrae - Gestión Completa

Inventario | Carrito de Compras | Ventas Realizadas | Recepción de Stock

Inventario por Estante

ID	Nombre	Precio	Stock	Estante
1	Sleeves Perfect Fit (60 und)	3500,0	50	Estante A-01
2	Sleeves Standard (60 und)	4500,0	40	Estante A-01
3	Sleeves Premium (60 und)	6500,0	30	Estante A-01
4	Carpeta 4x (160 cartas)	6500,0	25	Estante A-01
5	Carpeta 12x (480 cartas)	14900,0	15	Estante A-01
6	Carpeta 9x (360 cartas)	11900,0	20	Estante A-01
7	Portamazo Básico	2300,0	35	Estante A-01
8	Portamazo Magnético	13900,0	18	Estante A-01
9	Portamazo Premium	12900,0	22	Estante A-01
10	Portamazo Dual	22900,0	12	Estante A-01
11	Booster Box OP06 - Wings of the Captain	125000,0	5	Estante B-02
12	Booster Box OP07 - 500 Years in the Future	125000,0	6	Estante B-02
13	Booster Box OP08 - Two Legends	125000,0	4	Estante B-02
14	Booster Box OP09 - Emperors in the New World	125000,0	3	Estante B-02
15	Booster Box OP10 - Royal Blood	125000,0	7	Estante B-02
16	Booster Box OP11 - A Fist of Divine Speed	125000,0	2	Estante B-02
17	Booster Box OP12 - Legacy of the Master	125000,0	5	Estante B-02
18	Booster Box OP13 - Carrying On His Will	125000,0	4	Estante B-02
19	Extra Booster EB01 - Memorial Collection	140000,0	6	Estante B-02
20	Extra Booster EB02 - Anime 25th Collection	140000,0	4	Estante B-02
21	Premium Booster PRB01 - The Best	130000,0	7	Estante B-02
22	Premium Booster PRB02 - The Best Vol. 2	130000,0	5	Estante B-02
23	Starter Deck ST-11 - Uta	12900,0	20	Estante C-03
24	Starter Deck ST-12 - Zoro & Sanji	12900,0	18	Estante C-03
25	Starter Deck ST-13 - The Three Captains	12900,0	15	Estante C-03
26	Starter Deck ST-14 - 3DZY	12900,0	14	Estante C-03
27	Starter Deck ST-15 - Red Edward Newgate	12900,0	12	Estante C-03
28	Starter Deck ST-16 - Black Smoker	12900,0	13	Estante C-03
29	Starter Deck ST-17 - Blue Donquixote Doflamingo	12900,0	11	Estante C-03
30	Starter Deck ST-18 - Purple Monkey D. Luffy	12900,0	10	Estante C-03
31	Starter Deck ST-19 - Black Monkey D. Luffy	12900,0	9	Estante C-03

[Actualizar Inventario](#)

Figura 5: Inventario tienda central

Finis TCGrae - Gestión Completa

Inventario | Carrito de Compras | Ventas Realizadas | Recepción de Stock

Carrito de Compras

Cliente

ID Cliente: 10 - Juan Silva

Agregar Producto

Producto: 3 - Sleeves Premium (60 und) Cantidad: 1

ID	Producto	Cant	Precio	Subtotal
4	Carpeta 4x (160 cartas)	2	\$6,500,0	\$13,000,0
8	Portamazo Magnético	1	\$13,900,0	\$13,900,0
3	Sleeves Premium (60 und)	2	\$6,500,0	\$13,000,0

Total: \$39,900,0

[Agregar al Carrito](#) [Finalizar Venta](#) [Limpiar Carrito](#)

Figura 6: Carrito con 3 productos antes de finalizar

Finis TCGrae - Gestión Completa

Inventario | Carrito de Compras | Ventas Realizadas | Recepción de Stock

Historial de Ventas

ID Venta	Fecha	Cliente	Total	Productos
6	2025-10-31	Juan Silva	\$39,900.0	2x Sleeves Premium (60 und), 2x Carpeta 4x (160 cartas), 1x Portamazo Magnético
5	2025-10-28	Cínteres Presenciales	\$1,000.0	1x Brook OP06-092 R, 1x Absalom OP06-061 R
4	2025-10-27	Camila Vega	\$22,900.0	1x Portamazo Dual
3	2025-10-27	Cínteres Presenciales	\$500.0	1x Uta OP06-001 L
2	2025-10-26	Cínteres Presenciales	\$12,900.0	1x Starter Deck ST-11 - Uta
1	2025-10-26	Ana Torres	\$128,500.0	1x Sleeves Perfect Fit (60 und), 1x Booster Box OP06 - Wings of the Captain

[Actualizar Ventas](#)

Figura 7: Historial de ventas con productos detallados

Finis TCGrae - Gestión Completa

Inventario | Carrito de Compras | Ventas Realizadas | Recepción de Stock

Recepción de Stock

Ingresar Productos al Inventario

Producto:

Cantidad Recibida:

Proveedor (opcional):

[Recibir Stock](#)

Figura 8: Recepción de nuevo stock