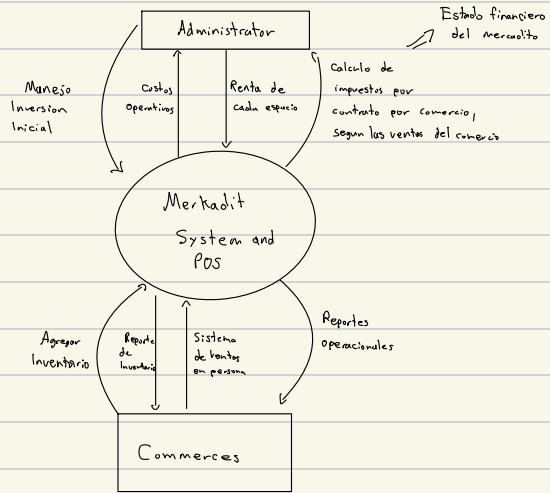
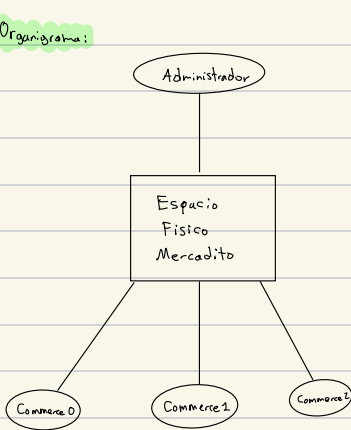


Caso #1 Bases de Datos

Merkadit = Sistema Interno de manejo de mercados gastronomicos y tiendas de conveniencia.

Organograma:



Esto ayuda a
definir los contratos
o settlements por
mes segun las ventas
del negocio.

Nota: Todas las ventas y reportes del negocio independiente
Son vistas por el administrador para tomarlos en cuenta
Segun el contrato.

Requerimientos funcionales:

1. Manejo de Espacio y Contrato
2. Manejo financiero del administrador
3. Sistema de ventas (POS)
4. Fee and Revenue Calculation
5. Reportes y Monitoreo.
6. Roles de Usuario y control de acceso

Derivables

1. Investigar el dominio del problema
2. Diseño de la base de datos
3. Inserts en Queries

(Alquiler)
↳ 2 Edificios ⇒ 2 Espacios ⇒ 4 a 7 Comercios

4. Creación de un Rest API

Investigación:

- Que es una API?
- Que es un rest API?
- Que es una tecnología de API? Node Js + Express etc..

Architectural Layers:

- Handler Layer (API Routes / EndPoints): → Que son? Para que se utilizan? Como funcionan?
- Controller Layer (Business Logic Orchestration): → Que son? Para que se utilizan? Como funcionan?
- Service Layer (Complex Business Logic): → Que son? Para que se utilizan? Como funcionan?
- Repository Layer (Data Access): → Que son? Para que se utilizan? Como funcionan?

Stored Procedures: → Que son? Para que se utilizan? Como funcionan?

Stored Procedure 1: registerSale

- Purpose: To record the sale of an item at a store.
- Input Parameters must have at least:
 - Product name
 - Store name (comercio)
 - Quantity sold
 - Amount paid (monto pagado)
 - Payment method (medio de pago)
 - Payment confirmations (confirmaciones de pago)
 - Reference numbers (numeros de referencia)
 - Invoice number (numero de factura)
 - Customer (cliente)
 - Applied discounts (descuentos aplicados)
- Implementation Requirements:
 - Must include comprehensive exception handling.
 - Must log details of the operation (e.g., computer, user, checksum).

→ Esto como se conecta con mi BD?

→ Los resultados de este procedure tienen un efecto en el diseño de mis tablas?

Postman test. Que es un postman collection para testear la API?

S. Queries para reportes

Stored Procedure 2: settleAccounts

- Purpose: To settle the accounts for a store for the current month.
- Input Parameters: It must have at least:
 - Name of the store (comercio)
 - Name of the location/point of sale (local)
- Logic: It must perform:
 - Check if the store has already been settled for the current month.
 - If it has NOT been settled:
 - Calculate the total fees owed to the previous administrator based on all sales.
 - Calculate the incoming amount that belongs to the store.
 - Insert the respective financial transaction log, adjusting balances.
 - Record the settlement to prevent it from being done a second time.
 - If it has been settled, it should handle this scenario appropriately.
- Implementation Requirements:
 - Must include comprehensive exception handling and transaction management to ensure data integrity.
 - Must log details of the operation (e.g., computer, user, checksum).

5. Write SQL queries for reporting

- Create a query that generates a business report as of the current date.
- The report should include:
 - Customer name
 - Store name
 - Building name
 - Date of the last sale in the current month
 - Number of items sold
 - Total sale amount
 - Percentage and monetary amount due to the store space owner
 - Retail fee amount to be paid by the business
- Transform the query into a view and use it as data source for a report.
- Create a professional looking report that includes:
 - Title and subtitle
 - Current date
 - A table with the query results
 - Subtotals grouped by business
 - A final total at the end of the report
- You may use any of the following tools for a desktop and/or Power BI Cloud Reports: Tableau, Kibana.
- The report should demonstrate clear data visualization and professional presentation.

→ Que es Power BI, como lo conecto con mi BD?

Diseño de la BD:

- Todos las tablas inician con la etiqueta "MK_" de Morkadit
- División de Tablas:

Perfiles

• Users ↔ • User Roles ↔ • Roles ↔ • RolePermissions ↔ • Permissions

Usuario x Mercado = Un mercado tiene varios user y un user puede administrar varios mercados. Caso empresas.

Espacio y Comercios

- Mercado (Un administrador puede tener varios mercados)
- Building x Mercado (Multiplaza puede tener 2 mercados y 1 mismo mercado puede estar en varias malls)
- Building (Ejemplo Multiplaza) => FK Addresses
- Spaces (Un local que se alquila para el espacio del mercado)
- Commerces (Comercios independientes que alquilan en el mercadito)
- Commerce Category (Categoría del Comercio del mercadito)

Localizaciones

• Countries • States • Cities • Address

Contratos con Comercios y Building

• Contract ↔ • Contract Per Commerce ↔ • Contract Per Building ↔ • Contract Renewals • Commerce Settlement • Commerce Settlement detail

Logs

- Logs
- Log Types
- Log Sources
- Log Severities