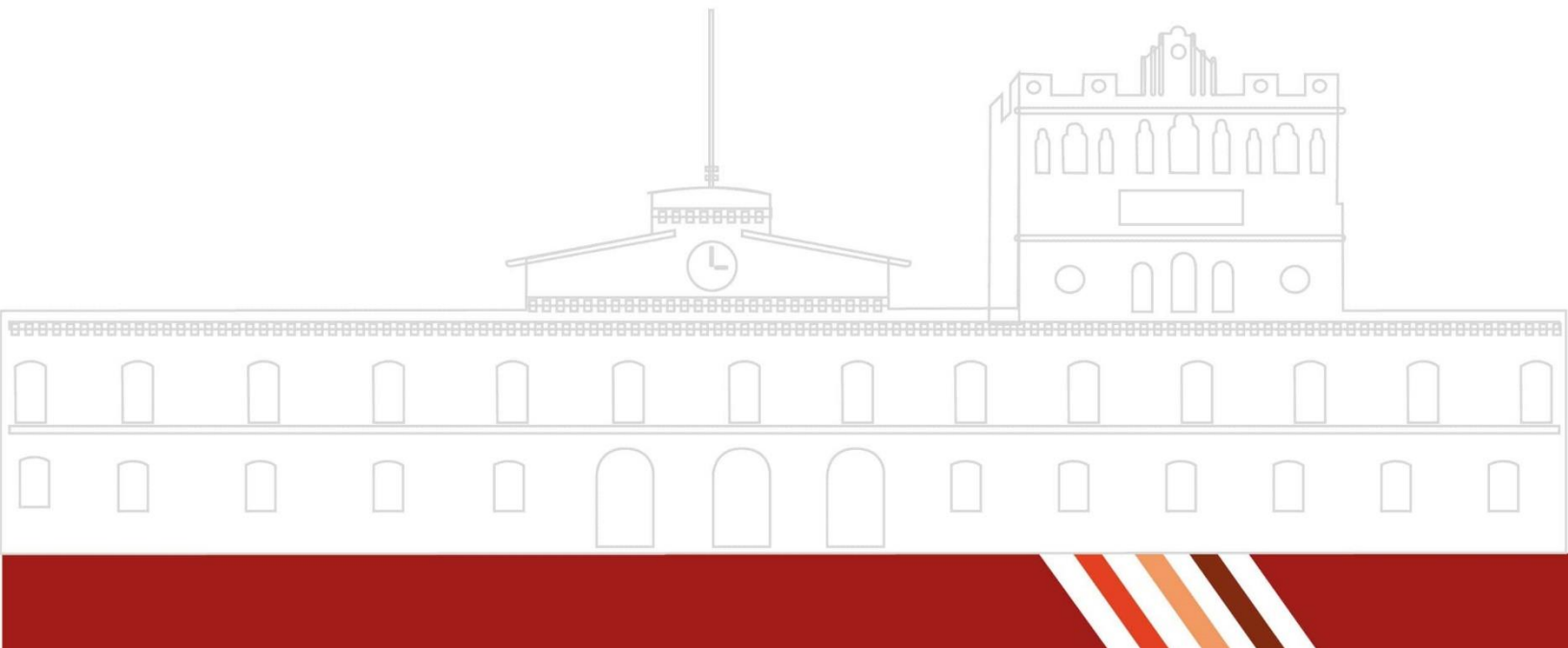


2.3 Práctica. Inventario herramientas

2.3 Práctica. Inventario herramientas

ALUMNO: Chavez Quiroz Kevin Axel

Dr. Eduardo Cornejo-Velázquez



1. Introducción

En este reporte abordaremos la importancia de las gramáticas independientes del contexto (GIC) en el desarrollo de lenguajes de programación. Estas gramáticas permiten definir de manera precisa estructuras recursivas que son fundamentales para la correcta interpretación y compilación de los códigos fuente. Además, se presentará un análisis de los requerimientos del sistema, diseño de vistas, y un modelo entidad-relación con sus respectivas implementaciones en SQL, ofreciendo un panorama completo desde la teoría hasta la práctica en la gestión de bases de datos.

2. Marco teórico

Análisis de requerimientos

- Descripción de los requisitos del sistema.
- Justificación de los mismos.

Diseño de vistas

- Definición y diseño de las interfaces de usuario.
- Ejemplos de vistas: formularios, reportes, etc.

Diseño conceptual

- **Análisis de entidades:** Identificación de las entidades del sistema.
- **Análisis funcional:** Descripción de las funciones que realizará el sistema.
- **Integración de vistas:** Cómo se integran las diferentes vistas en el sistema.
- **Esquema conceptual global:** Diagrama y descripción del esquema general del sistema.

3. Herramientas utilizadas

1. MySQL Server

MySQL Server es un sistema de gestión de bases de datos relacional de código abierto, usado para almacenar y manejar datos. Permite realizar consultas SQL, gestionar la seguridad y control de acceso, y es conocido por su velocidad y escalabilidad. Es popular en el desarrollo web para manejar bases de datos en aplicaciones como sitios de comercio electrónico y sistemas de gestión de contenido.

2. Overleaf

Overleaf es una plataforma en línea para crear y colaborar en documentos LaTeX. Permite la edición colaborativa en tiempo real y la compilación automática del documento. Ofrece plantillas y se integra con herramientas de referencias bibliográficas y control de versiones. Es ampliamente usado en entornos académicos y científicos para escribir artículos, tesis y presentaciones de alta calidad.

4. Desarrollo

Modelo Entidad-Relación

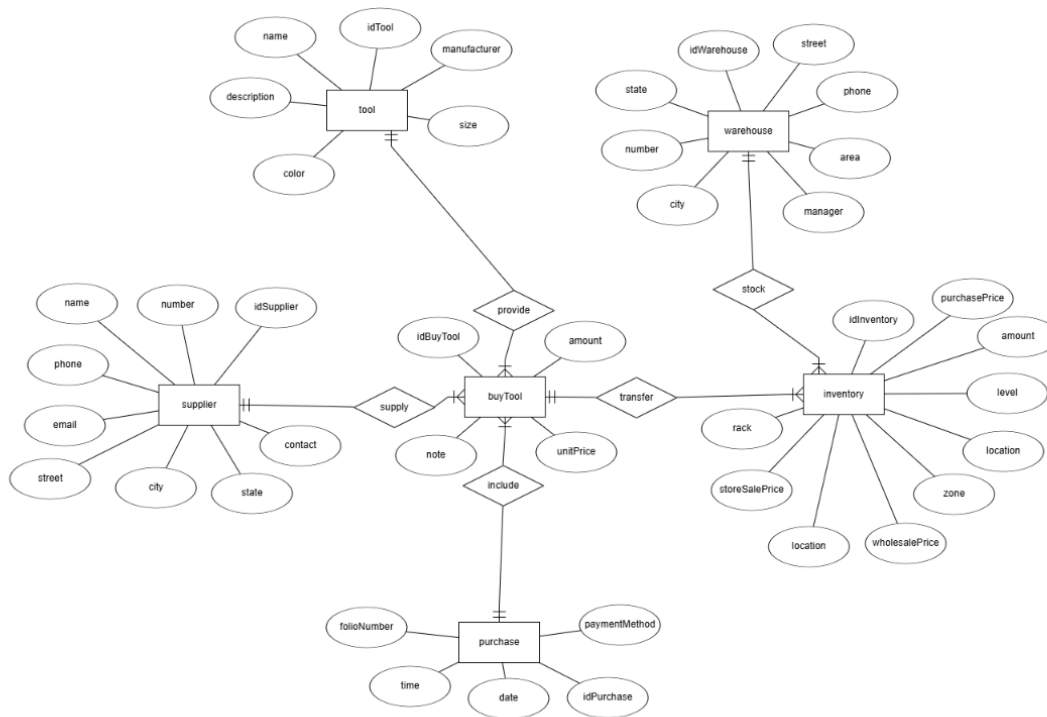
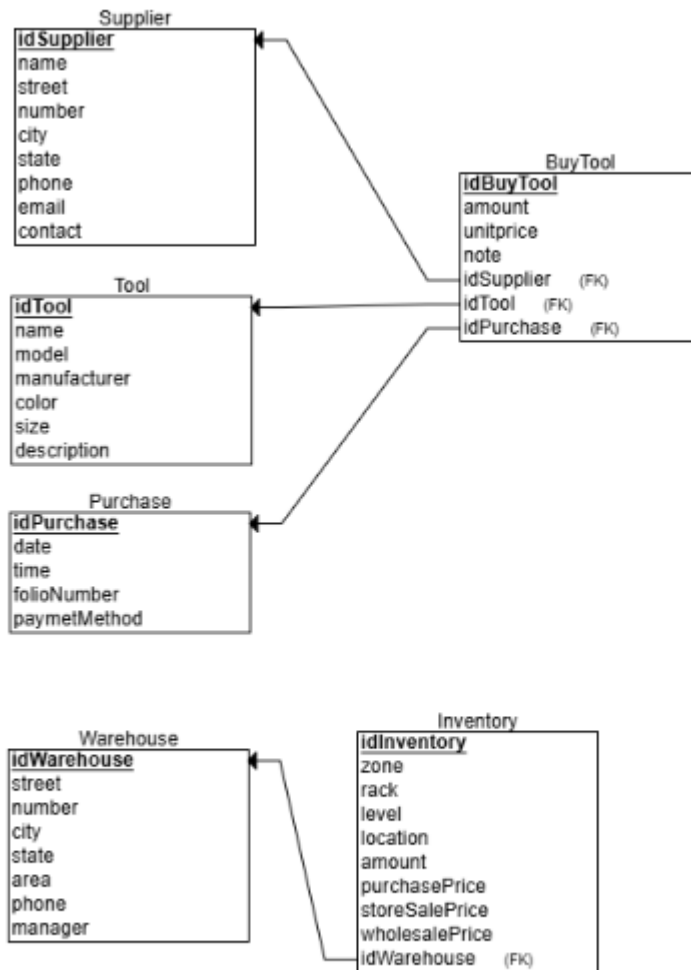


Tabla de mapeo de atributos

Entidad	Atributo	Clave
Supplier	idSupplier	Primary Key
name	VARCHAR(50)	
street	VARCHAR(50)	
number	INT	
city	VARCHAR(50)	
state	VARCHAR(50)	
phone	VARCHAR(15)	
email	VARCHAR(50)	
contact	VARCHAR(50)	
Tool	idTool	Primary Key
name	VARCHAR(50)	
model	VARCHAR(50)	
manufacturer	VARCHAR(50)	
color	VARCHAR(20)	
size	VARCHAR(10)	
description	TEXT	
BuyTool	idBuyTool	Primary Key
idSupplier	INT	
idPurchase	INT	

idTool	INT	
amount	INT	
unitPrice	DECIMAL(10, 2)	
note	TEXT	
Purchase	idPurchase	Primary Key
date	DATE	
time	TIME	
folioNumber	VARCHAR(50)	
paymentMethod	VARCHAR(20)	
Inventory	idInventory	Primary Key
idWarehouse	INT	
zone	VARCHAR(20)	
rack	INT	
level	INT	
location	VARCHAR(50)	
amount	INT	
purchasePrice	DECIMAL(10, 2)	
storeSalePrice	DECIMAL(10, 2)	
wholesalePrice	DECIMAL(10, 2)	
Warehouse	idWarehouse	Primary Key
street	VARCHAR(50)	
number	INT	
city	VARCHAR(50)	
state	VARCHAR(50)	
area	DECIMAL(10, 2)	

Modelo relacional



Sentencias SQL

CREATE TABLE Supplier (

idSupplier INT PRIMARY KEY,

name VARCHAR(50),

street VARCHAR(50),

number INT,

city VARCHAR(50),

state VARCHAR(50),

phone VARCHAR(15),

email VARCHAR(50),

contact VARCHAR(50)

);

```
CREATE TABLE Tool (  
    idTool INT PRIMARY KEY,  
    name VARCHAR(50),  
    model VARCHAR(50),  
    manufacturer VARCHAR(50),  
    color VARCHAR(20),  
    size VARCHAR(10),  
    description TEXT  
);
```

```
CREATE TABLE BuyTool (  
    idBuyTool INT PRIMARY KEY,  
    idSupplier INT,  
    idPurchase INT,  
    idTool INT,  
    amount INT,  
    unitPrice DECIMAL(10,2),  
    note TEXT,  
    FOREIGN KEY (idSupplier) REFERENCES Supplier(idSupplier),  
    FOREIGN KEY (idPurchase) REFERENCES Purchase(idPurchase),  
    FOREIGN KEY (idTool) REFERENCES Tool(idTool)  
);
```

```
CREATE TABLE Purchase (  
    idPurchase INT PRIMARY KEY,  
    date DATE,  
    time TIME,  
    folioNumber VARCHAR(50),
```

```
paymentMethod VARCHAR(20)
);
```

```
CREATE TABLE Inventory (
    idInventory INT PRIMARY KEY,
    idWarehouse INT,
    zone VARCHAR(20),
    rack INT,
    level INT,
    location VARCHAR(50),
    amount INT,
    purchasePrice DECIMAL(10,2),
    storeSalePrice DECIMAL(10,2),
    wholesalePrice DECIMAL(10,2),
    FOREIGN KEY (idWarehouse) REFERENCES Warehouse(idWarehouse)
);
```

```
CREATE TABLE Warehouse (
    idWarehouse INT PRIMARY KEY,
    street VARCHAR(50),
    number INT,
    city VARCHAR(50),
    state VARCHAR(50),
    area DECIMAL(10,2),
    phone VARCHAR(15),
    manager VARCHAR(50)
);
```


-- Insertar registros en Supplier

```
INSERT INTO Supplier VALUES (1, 'Proveedor 1', 'Calle 1', 100, 'Ciudad 1', 'Estado 1', '1234567890', 'prov1@mail.com', 'Contacto 1');
```

```
INSERT INTO Supplier VALUES (2, 'Proveedor 2', 'Calle 2', 200, 'Ciudad 2', 'Estado 2', '0987654321', 'prov2@mail.com', 'Contacto 2');
```

-- Insertar registros en Tool

```
INSERT INTO Tool VALUES (1, 'Herramienta 1', 'Modelo 1', 'Fabricante 1', 'Rojo', 'M', 'Descripción 1');
```

```
INSERT INTO Tool VALUES (2, 'Herramienta 2', 'Modelo 2', 'Fabricante 2', 'Azul', 'L', 'Descripción 2');
```

-- Insertar registros en BuyTool

```
INSERT INTO BuyTool VALUES (1, 1, 1, 1, 10, 50.00, 'Nota 1');
```

```
INSERT INTO BuyTool VALUES (2, 2, 2, 2, 5, 100.00, 'Nota 2');
```

-- Insertar registros en Purchase

```
INSERT INTO Purchase VALUES (1, '2024-01-15', '10:00:00', 'F1234', 'Efectivo');
```

```
INSERT INTO Purchase VALUES (2, '2024-01-20', '15:00:00', 'F5678', 'Tarjeta');
```

-- Insertar registros en Inventory

```
INSERT INTO Inventory VALUES (1, 1, 'Zona 1', 1, 1, 'Ubicación 1', 10, 500.00, 700.00, 650.00);
```

```
INSERT INTO Inventory VALUES (2, 1, 'Zona 2', 2, 2, 'Ubicación 2', 5, 1000.00, 1500.00, 1400.00);
```

-- Insertar registros en Warehouse

```
INSERT INTO Warehouse VALUES (1, 'Calle Bodega 1', 123, 'Ciudad Bodega 1', 'Estado Bodega 1', 500.00, '1112223333', 'Manager 1');
```

```
SELECT p.date, s.name AS supplier, t.name AS tool, b.amount, b.unitPrice, (b.amount * b.unitPrice)
AS totalPrice

FROM BuyTool b

JOIN Purchase p ON b.idPurchase = p.idPurchase

JOIN Supplier s ON b.idSupplier = s.idSupplier

JOIN Tool t ON b.idTool = t.idTool

WHERE MONTH(p.date) = 1

ORDER BY p.date;
```

```
SELECT t.name AS tool, i.amount, (i.amount * i.purchasePrice) AS totalCost

FROM Inventory i

JOIN Tool t ON i.idInventory = t.idTool

WHERE i.idWarehouse = 1;
```

5. Conclusión

En resumen, hemos explorado cómo las gramáticas independientes del contexto son fundamentales para definir estructuras recursivas en lenguajes de programación, permitiendo un análisis y traducción eficiente en compiladores. Hemos cubierto desde conceptos teóricos hasta la implementación práctica de modelos relacionales y sentencias SQL para la gestión de datos, concluyendo con ejemplos de consultas específicas.