

28 DE MAYO, 2023

# TRABAJO PRACTICAS

DESAROLLO  
DIRIGIDO  
POR  
MODELOS

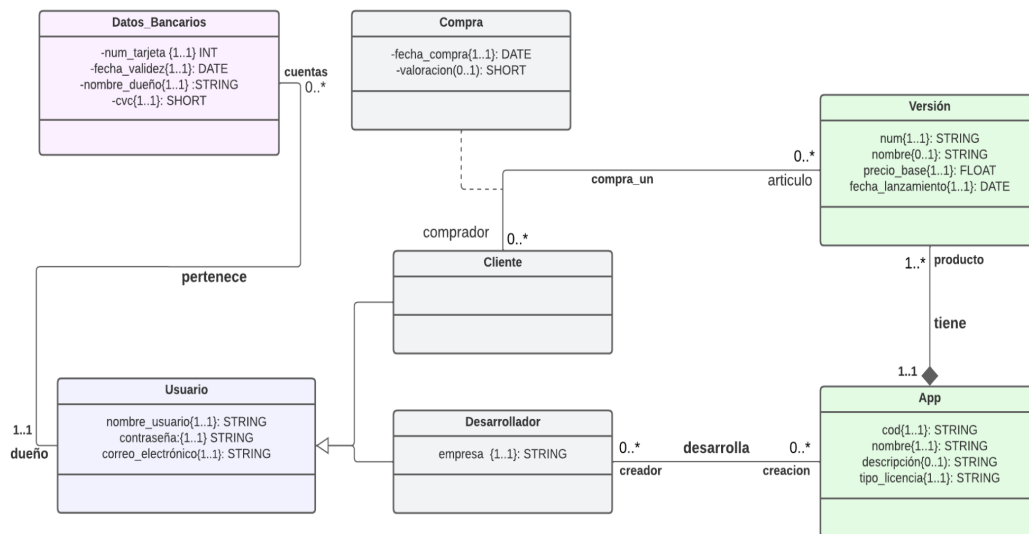
EDWIN MAKOVEEV ROUTSKAIA  
IVAN ABAD GARCIA

## INTRODUCCIÓN

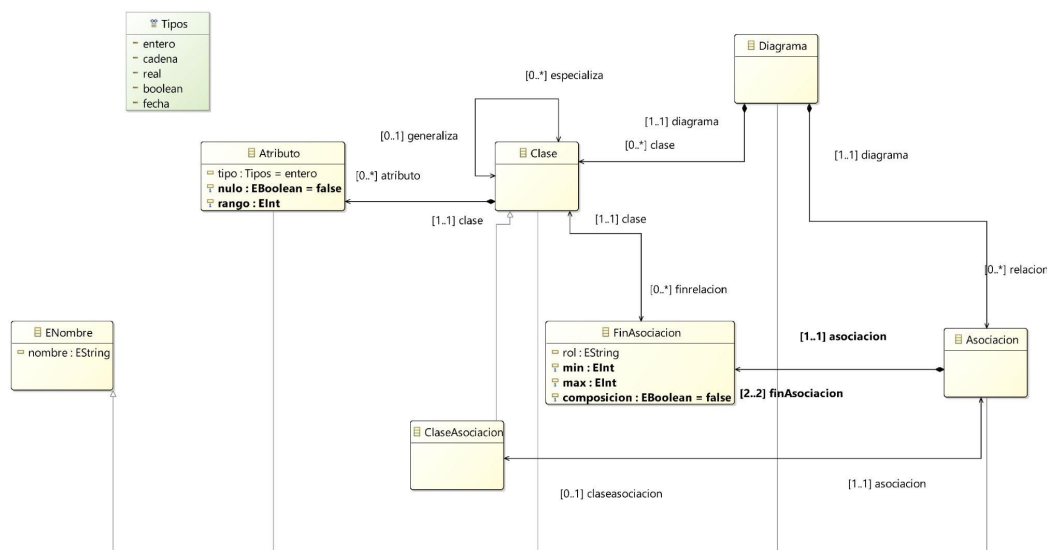
Este proyecto consiste a partir de un fichero .xmi crear un script SQL que incluya todas sus tablas, atributos y relaciones, cabe recalcar que, debido a que en la entrega pasada se hizo mal la **asociacion2ClaveAjena**, **herencia2tablas** y **clavesAjenas2ClasesAsociacion** arrastramos esos errores a esta entrega, ya que se utiliza ese fichero .xmi generado. Por otra parte, para la realización de este proyecto hemos tenido que consultar los pdfs de la asignatura de **bases de datos (BDA)**, ya que no sabiamos muy bien el formato de como debia generarse, y a su vez [SQL Fiddle](#) para asegurarnos que el OUTPUT de Acceleo era correcto.

## DIAGRAMAS Y XMI UTILIZADOS EN EL PROYECTO

UML:



DC:



ER:

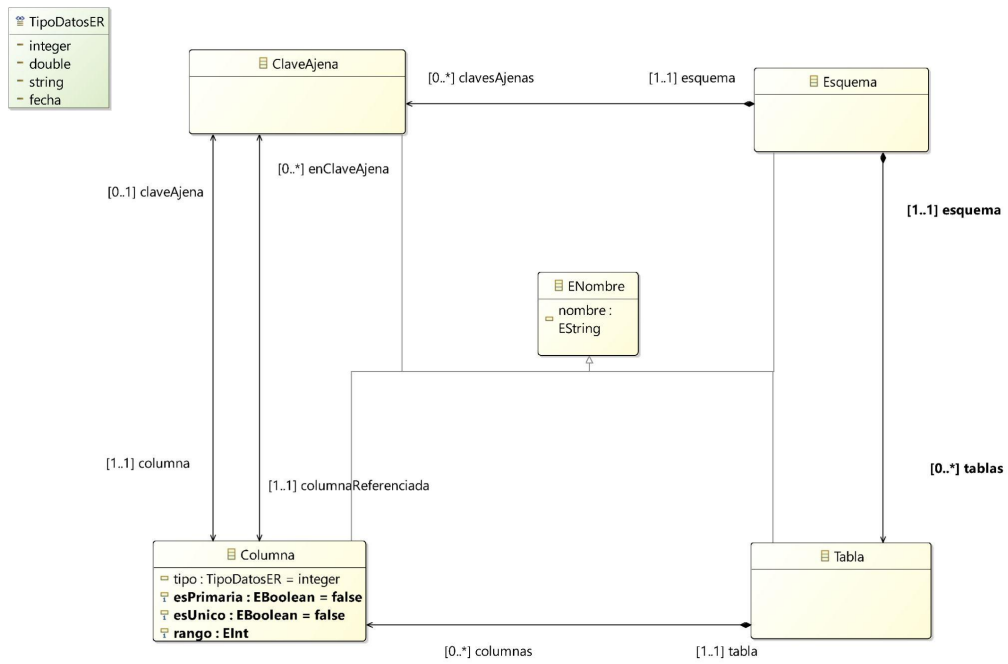


DIAGRAMA .XMI INPUT ('Diagrama.xmi'):








































































- ▼  platform:/resource/UMLRefining/Transformation/Diagrama.xmi
  - ▼  Diagrama
    - ▼  Clase Usuario
      -  Atributo contraseña
      -  Atributo nombre\_usuario
      -  Atributo correo\_electronico
    -  Clase Cliente
    - ▼  Clase Desarrollador
      -  Atributo empresa
    - ▼  Clase App
      -  Atributo cod
      -  Atributo nombre
      -  Atributo descripcion
      -  Atributo tipo\_licencia
    - ▼  Clase Version
      -  Atributo num
      -  Atributo nombre
      -  Atributo precio\_base
      -  Atributo fecha\_lanzamiento
    - ▼  Clase Datos\_Bancarios
      -  Atributo num\_tarjeta
      -  Atributo fecha\_validez
      -  Atributo nombre\_dueño
      -  Atributo cvc
    -  Clase Asociacion Compra
    - ▼  Asociacion compra\_un
      -  Fin Asociacion comprador
      -  Fin Asociacion articulo
    - ▼  Asociacion pertenece
      -  Fin Asociacion dueño
      -  Fin Asociacion cuentas
    - ▼  Asociacion desarrolla
      -  Fin Asociacion creador
      -  Fin Asociacion creacion
- >  platform:/resource/UMLRefining/Metamodels/DC\_Uml.ecore

DIAGRAMA .XMI GENERADO ('DiagramaOUT.xmi'):

- ▼  platform:/resource/UMLRefining/Transformation/DiagramaOUT.xmi
  - ▼  Esquema
    - ▼  Tabla Usuario
      -  Columna id\_Usuario
      -  Columna contraseña
      -  Columna nombre\_usuario
      -  Columna correo\_electronico
    - ▼  Tabla Cliente
      -  Columna id\_Cliente
    - ▼  Tabla Desarrollador
      -  Columna id\_Developador
      -  Columna empresa
    - ▼  Tabla App
      -  Columna id\_App
      -  Columna cod
      -  Columna nombre
      -  Columna descripcion
      -  Columna tipo\_licencia
    - ▼  Tabla Version
      -  Columna id\_Version
      -  Columna num
      -  Columna nombre
      -  Columna precio\_base
      -  Columna fecha\_lanzamiento
    - ▼  Tabla Compra
      -  Columna id\_Compra
      -  Columna fecha\_compra
      -  Columna valoracion
    - ▼  Tabla Datos\_Bancarios
      -  Columna id\_Datos\_Bancarios
      -  Columna num\_tarjeta
      -  Columna fecha\_validez
      -  Columna nombre\_dueño
      -  Columna cvc
      -  Columna id\_FK\_dueño
    -  Clave Ajena FK\_cuentas\_dueño

# SQL GENERADO

Usando como base el fichero .xml ‘Diagrama.OUT’ hemos creado el siguiente fichero .mtl:

```
[comment encoding = UTF-8 /]
[module generate('ER')]

[template public generateElement(anEsquema : Esquema)]
[comment @main/]

[file (anEsquema.nombre.concat('.sql'), false, 'UTF-8')]
DROP DATABASE IF EXISTS [anEsquema.nombre/];
CREATE DATABASE [anEsquema.nombre/];
USE [anEsquema.nombre/];
CREATE SCHEMA [anEsquema.nombre/];

[for (t: Tabla | anEsquema.tablas) separator ('\n')]
DROP TABLE IF EXISTS [t.nombre/];
[/for]

[for (t: Tabla | anEsquema.tablas) separator ('\n')]
CREATE TABLE [t.nombre/]
(
  [for (c:Columna | t.columnas) separator ('\n')]
  [if not (t.columnas -> last() = c)]
  [c.nombre/] [obtenerTipo (c.tipo)/] [esPrimaria (c)/],
  [/if]
[/for]
[t.columnas->last().nombre/] [obtenerTipo(t.columnas->last().tipo)/] [esPrimaria(t.columnas->last())]/[for (c:Columna | t.columnas)][for (fk: ClaveAjena | c.enClaveAjena)],['\n'/]
CONSTRAINT [fk.nombre/] FOREIGN KEY ([c.nombre/]) REFERENCES [fk.columnaReferenciada.tabla.nombre/] ([fk.columnaReferenciada.nombre/])
[/for]
);
[/for]

[/file]
[/template]
[query public obtenerTipo(tipo : TipoDatosER) : String =
if tipo = TipoDatosER::integer then 'INTEGER' else
if tipo = TipoDatosER::double then 'REAL' else
if tipo = TipoDatosER::fecha then 'DATE' else 'VARCHAR(75)'
endif
endif
endif
/]

[query public esPrimaria (c: Columna): String =
if c.esPrimaria = true then 'PRIMARY KEY' else
if c.rango = 1 and c.esUnico then 'NOT NULL UNIQUE' else
if c.rango = 1 then 'NOT NULL' else ''
endif
endif
endif
/]
/]
```

El cual al ejecutar nos da lo siguiente:

```
DROP DATABASE IF EXISTS UML;

CREATE DATABASE UML;

USE UML;

DROP SCHEMA IF EXISTS UML;

CREATE SCHEMA UML;


DROP TABLE IF EXISTS Usuario;


DROP TABLE IF EXISTS Cliente;


DROP TABLE IF EXISTS Desarrollador;


DROP TABLE IF EXISTS App;


DROP TABLE IF EXISTS Version;
```

```
DROP TABLE IF EXISTS Compra;

DROP TABLE IF EXISTS Datos_Bancarios;

CREATE TABLE Usuario
(
    id_Usuario INTEGER PRIMARY KEY,

    contraseña VARCHAR(75) NOT NULL,

    nombre_usuario VARCHAR(75) NOT NULL,

    correo_electronico VARCHAR(75) NOT NULL,

    CONSTRAINT FK_cuentas_dueño FOREIGN KEY (id_Usuario) REFERENCES
Usuario (id_Usuario)
);

CREATE TABLE Cliente
(
    id_Cliente INTEGER PRIMARY KEY);

CREATE TABLE Desarrollador
(
    id_Desarrollador INTEGER PRIMARY KEY,

    empresa VARCHAR(75) NOT NULL);

CREATE TABLE App
(
    id_App INTEGER PRIMARY KEY,

    cod VARCHAR(75) NOT NULL,

    nombre VARCHAR(75) NOT NULL,

    descripcion VARCHAR(75) ,

    tipo_licencia VARCHAR(75) NOT NULL);

CREATE TABLE Version
(
    id_Version INTEGER PRIMARY KEY,
```

```

    num VARCHAR(75) NOT NULL,

    nombre VARCHAR(75) ,

    precio_base REAL NOT NULL,

    fecha_lanzamiento DATE NOT NULL);

CREATE TABLE Compra
(
    id_Compra INTEGER PRIMARY KEY,

    fecha_compra DATE NOT NULL,

    valoracion INTEGER );

CREATE TABLE Datos_Bancarios
(
    id_Datos_Bancarios INTEGER PRIMARY KEY,

    num_tarjeta INTEGER NOT NULL,

    fecha_validez DATE NOT NULL,

    nombre_dueño VARCHAR(75) NOT NULL,

    cvc INTEGER ,

    id_FK_dueño INTEGER );

```

Cabe recalcar que, en clase se menciono que habia que dropear las tablas si existian, ya que si no se hiciese podria generar algún error, lo que no sabiamos era si hacia falta tambien dropear el esquema general, por lo que ante las dudas lo hicimos, por otra parte, hemos como tenemos columnas que pueden o no ser nulas, ser unicas o incluso ser claves primarias, nos hemos asegurado que eso se tome en cuenta con el metodo de **query esPrimaria**.

En la validación de sqlFiddle las 5 primeras lineas el programa dice que no tiene permisos, si se quita, no habria problemas excepto con la CONSTRAINT de Usuario diciendo '**Cannot add foreign key constraint**', si se quita el CONSTRAINT pone que '**Schema Ready**'.