

Sede Arequipa UTP

**FACULTAD DE INGENIERIA**

**Carrera Profesional de Ingeniería de Sistemas e Informática**

CURSO:

**BASE DE DATOS**

**Trabajo Final**

**Integrantes**

* Bazan Ramos, Axel Jean Piero U20202870
* Sharom Mitzey Atencio Vilca U19217210

**Docente**

Maribel Carmen Urquizo Abril

Arequipa, Perú

2023

Introducción

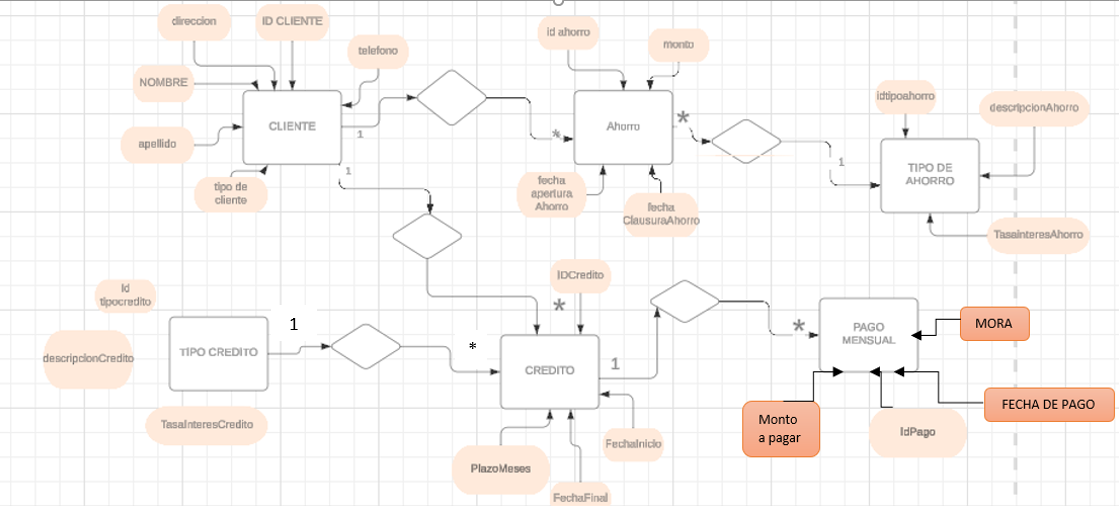
En el contexto actual de las cooperativas financieras, la gestión eficiente de los ahorros y créditos es fundamental para mantener la estabilidad y satisfacer las necesidades de los clientes. La cooperativa de Ahorro y Crédito “Amaru” se especializa en ofrecer servicios tanto de ahorro como de crédito a sus clientes, los cuales pueden ser personas naturales o jurídicas. Entre sus productos de ahorro destacan la Cuenta Simple con una tasa de interés del 3%, la Super Tasa con un 6.5%, y el Depósito a Plazo con un 8%. Para los créditos, la cooperativa ofrece una variedad de opciones como préstamos personales, para vivienda, apoyo a negocios, estudios, y disposición de efectivo, cada uno con sus condiciones específicas de pago y tasas de interés.

Caso

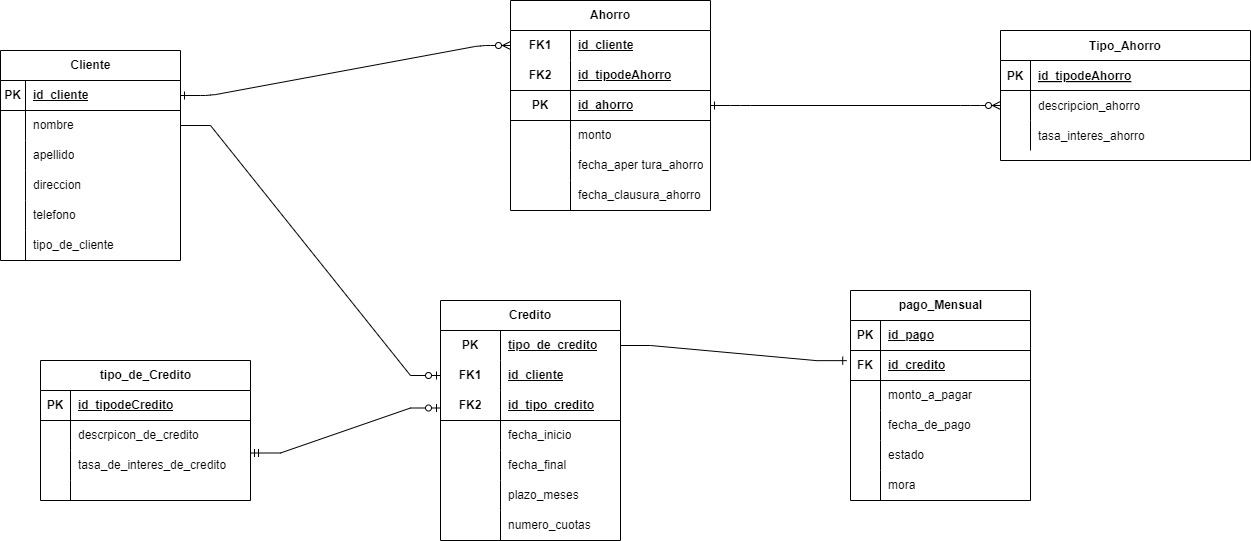
**Sistema de ahorros y créditos**

La cooperativa de Ahorro y Crédito “Amaru” tiene operaciones de ahorro y crédito para sus clientes (es decir registran datos de sus clientes, que pueden ser naturales o jurídicos). Así mismo se registran ahorros en los diferentes tipos como: Cuenta Simple 3%, Super Tasa 6.5%, Depósito a plazo 8%. Así también tipos de crédito como: préstamo personal, préstamo para vivienda, préstamo apoya tu negocio, préstamo para estudios, disposición de efectivo. A nivel de créditos el cliente se presta un monto, escogiendo un tipo de préstamo y el tiempo determinado a pagar que va de 12, 24 y 36 meses, generando una tasa de interés en cada cuota y las fechas de pago mensuales. Si el cliente no paga su cuota en la fecha límite mensual (calendario de pagos) que establece la cooperativa, se le otorga una mora adicional del 2% del monto a pagar respectivo. Se desea que el banco registre los datos de ahorros de sus clientes, sus créditos y el monto a pagar mes a mes, también los clientes que son morosos. Considerar que un cliente puede tener varios tipos de ahorro, pero solo un tipo de préstamo.

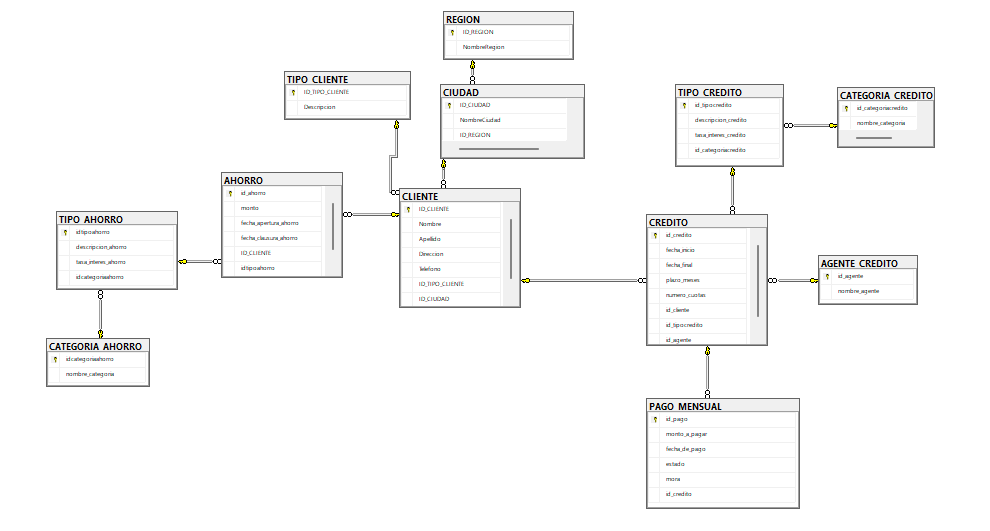
Diagrama E-R



1. Modelo Lógico



Normalizacion:



Comandos de Creación

CREATE DATABASE PROYECTOFINAL2;

USE PROYECTOFINAL2;

-- Tabla TIPO\_CLIENTE

CREATE TABLE TIPO\_CLIENTE (

ID\_TIPO\_CLIENTE INT PRIMARY KEY,

Descripcion VARCHAR(50)

);

-- Tabla REGION

CREATE TABLE REGION (

ID\_REGION INT PRIMARY KEY,

NombreRegion VARCHAR(50)

);

-- Tabla CIUDAD

CREATE TABLE CIUDAD (

ID\_CIUDAD INT PRIMARY KEY,

NombreCiudad VARCHAR(50),

ID\_REGION INT,

FOREIGN KEY (ID\_REGION) REFERENCES REGION(ID\_REGION)

);

-- Tabla CLIENTE

CREATE TABLE CLIENTE (

ID\_CLIENTE INT PRIMARY KEY,

Nombre VARCHAR(50),

Apellido VARCHAR(50),

Direccion VARCHAR(100),

Telefono VARCHAR(15),

ID\_TIPO\_CLIENTE INT,

ID\_CIUDAD INT,

FOREIGN KEY (ID\_TIPO\_CLIENTE) REFERENCES TIPO\_CLIENTE(ID\_TIPO\_CLIENTE),

FOREIGN KEY (ID\_CIUDAD) REFERENCES CIUDAD(ID\_CIUDAD)

);

-- Tabla CATEGORIA\_AHORRO

CREATE TABLE CATEGORIA\_AHORRO (

idcategoriaahorro INT PRIMARY KEY,

nombre\_categoria VARCHAR(50)

);

-- Tabla TIPO\_AHORRO

CREATE TABLE TIPO\_AHORRO (

idtipoahorro INT PRIMARY KEY,

descripcion\_ahorro VARCHAR(100),

tasa\_interes\_ahorro DECIMAL(5,2),

idcategoriaahorro INT,

FOREIGN KEY (idcategoriaahorro) REFERENCES CATEGORIA\_AHORRO(idcategoriaahorro)

);

-- Tabla AHORRO

CREATE TABLE AHORRO (

id\_ahorro INT PRIMARY KEY,

monto DECIMAL(10,2),

fecha\_apertura\_ahorro DATE,

fecha\_clausura\_ahorro DATE,

ID\_CLIENTE INT,

idtipoahorro INT,

FOREIGN KEY (ID\_CLIENTE) REFERENCES CLIENTE(ID\_CLIENTE),

FOREIGN KEY (idtipoahorro) REFERENCES TIPO\_AHORRO(idtipoahorro)

);

-- Tabla CATEGORIA\_CREDITO

CREATE TABLE CATEGORIA\_CREDITO (

id\_categoriacredito INT PRIMARY KEY,

nombre\_categoria VARCHAR(50)

);

-- Tabla TIPO\_CREDITO

CREATE TABLE TIPO\_CREDITO (

id\_tipocredito INT PRIMARY KEY,

descripcion\_credito VARCHAR(100),

tasa\_interes\_credito DECIMAL(5,2),

id\_categoriacredito INT,

FOREIGN KEY (id\_categoriacredito) REFERENCES CATEGORIA\_CREDITO(id\_categoriacredito)

);

-- Tabla AGENTE\_CREDITO

CREATE TABLE AGENTE\_CREDITO (

id\_agente INT PRIMARY KEY,

nombre\_agente VARCHAR(100)

);

-- Tabla CREDITO

CREATE TABLE CREDITO (

id\_credito INT PRIMARY KEY,

fecha\_inicio DATE,

fecha\_final DATE,

plazo\_meses INT,

numero\_cuotas INT,

id\_cliente INT,

id\_tipocredito INT,

id\_agente INT,

FOREIGN KEY (id\_cliente) REFERENCES CLIENTE(ID\_CLIENTE),

FOREIGN KEY (id\_tipocredito) REFERENCES TIPO\_CREDITO(id\_tipocredito),

FOREIGN KEY (id\_agente) REFERENCES AGENTE\_CREDITO(id\_agente)

);

-- Tabla PAGO\_MENSUAL

CREATE TABLE PAGO\_MENSUAL (

id\_pago INT PRIMARY KEY,

monto\_a\_pagar DECIMAL(10,2),

fecha\_de\_pago DATE,

estado VARCHAR(20),

mora DECIMAL(5,2),

id\_credito INT,

FOREIGN KEY (id\_credito) REFERENCES CREDITO(id\_credito)

);

-- Inserciones para TIPO\_CLIENTE

INSERT INTO TIPO\_CLIENTE (ID\_TIPO\_CLIENTE, Descripcion) VALUES

(1, 'Natural'),

(2, 'Jurídico');

-- Inserciones para REGION

INSERT INTO REGION (ID\_REGION, NombreRegion) VALUES

(1, 'Lima'),

(2, 'Arequipa'),

(3, 'Cusco');

-- Inserciones para CIUDAD

INSERT INTO CIUDAD (ID\_CIUDAD, NombreCiudad, ID\_REGION) VALUES

(1, 'Lima Ciudad', 1),

(2, 'Arequipa Ciudad', 2),

(3, 'Cusco Ciudad', 3);

-- Inserciones para CLIENTE

INSERT INTO CLIENTE (ID\_CLIENTE, Nombre, Apellido, Direccion, Telefono, ID\_TIPO\_CLIENTE, ID\_CIUDAD) VALUES

(1, 'Juan', 'Perez', 'Av. Primavera 123', '987654321', 1, 1),

(2, 'María', 'Gomez', 'Jr. Huancavelica 456', '999888777', 2, 2);

-- Inserciones para CATEGORIA\_AHORRO

INSERT INTO CATEGORIA\_AHORRO (idcategoriaahorro, nombre\_categoria) VALUES

(1, 'Cuenta Simple'),

(2, 'Super Tasa'),

(3, 'Depósito a Plazo');

-- Inserciones para TIPO\_AHORRO

INSERT INTO TIPO\_AHORRO (idtipoahorro, descripcion\_ahorro, tasa\_interes\_ahorro, idcategoriaahorro) VALUES

(1, 'Cuenta Simple con 3% de interés', 3.00, 1),

(2, 'Super Tasa con 6.5% de interés', 6.50, 2),

(3, 'Depósito a Plazo con 8% de interés', 8.00, 3);

-- Inserciones para AHORRO

INSERT INTO AHORRO (id\_ahorro, monto, fecha\_apertura\_ahorro, fecha\_clausura\_ahorro, ID\_CLIENTE, idtipoahorro) VALUES

(1, 5000.00, '2024-07-15', NULL, 1, 1),

(2, 10000.00, '2024-07-10', NULL, 2, 2);

-- Inserciones para CATEGORIA\_CREDITO

INSERT INTO CATEGORIA\_CREDITO (id\_categoriacredito, nombre\_categoria) VALUES

(1, 'Préstamo Personal'),

(2, 'Préstamo para Vivienda'),

(3, 'Préstamo Apoya Tu Negocio');

-- Inserciones para TIPO\_CREDITO

INSERT INTO TIPO\_CREDITO (id\_tipocredito, descripcion\_credito, tasa\_interes\_credito, id\_categoriacredito) VALUES

(1, 'Préstamo Personal hasta 12 meses', 12.00, 1),

(2, 'Préstamo para Vivienda hasta 24 meses', 14.50, 2),

(3, 'Préstamo Apoya Tu Negocio hasta 36 meses', 16.00, 3);

-- Inserciones para AGENTE\_CREDITO

INSERT INTO AGENTE\_CREDITO (id\_agente, nombre\_agente) VALUES

(1, 'Pedro García'),

(2, 'Ana Rodríguez');

-- Inserciones para CREDITO

INSERT INTO CREDITO (id\_credito, fecha\_inicio, fecha\_final, plazo\_meses, numero\_cuotas, id\_cliente, id\_tipocredito, id\_agente) VALUES

(1, '2024-07-01', '2024-07-01', 12, 12, 1, 1, 1),

(2, '2024-06-15', '2024-06-15', 24, 24, 2, 2, 2);

-- Inserciones para PAGO\_MENSUAL

INSERT INTO PAGO\_MENSUAL (id\_pago, monto\_a\_pagar, fecha\_de\_pago, estado, mora, id\_credito) VALUES

(1, 500.00, '2024-07-15', 'Pendiente', 0.00, 1),

(2, 750.00, '2024-06-30', 'Pagado', 0.00, 2);

**Realice una consulta con JOIN en donde emplee como mínimo 3 tablas, también considerar el uso de la clausula where (con between, in o like) yorder by. Coloque su enunciado (2 puntos)**

Realice una consulta que muestre los pagos mensuales, el nombre del cliente, la descripción del tipo de crédito y el nombre del agente de crédito para aquellos pagos cuya fecha de pago esté entre dos fechas específicas. Ordene los resultados por la fecha de pago.

SELECT

PAGO\_MENSUAL.id\_pago,

PAGO\_MENSUAL.monto\_a\_pagar,

PAGO\_MENSUAL.fecha\_de\_pago,

CLIENTE.Nombre,

CLIENTE.Apellido,

TIPO\_CREDITO.descripcion\_credito,

AGENTE\_CREDITO.nombre\_agente

FROM

PAGO\_MENSUAL

JOIN

CREDITO ON PAGO\_MENSUAL.id\_credito = CREDITO.id\_credito

JOIN

CLIENTE ON CREDITO.id\_cliente = CLIENTE.ID\_CLIENTE

JOIN

TIPO\_CREDITO ON CREDITO.id\_tipocredito = TIPO\_CREDITO.id\_tipocredito

JOIN

AGENTE\_CREDITO ON CREDITO.id\_agente = AGENTE\_CREDITO.id\_agente

WHERE

PAGO\_MENSUAL.fecha\_de\_pago BETWEEN '2024-01-01' AND '2024-12-31'

ORDER BY

PAGO\_MENSUAL.fecha\_de\_pago;



**Realice una subconsulta de 2 niveles. Coloque su enunciado (2 puntos)**

Obtener los clientes que tienen créditos activos y mostrar los pagos mensuales pendientes de esos créditos

SELECT ID\_CLIENTE, Nombre, Apellido

FROM CLIENTE

WHERE ID\_CLIENTE IN (

SELECT id\_cliente

FROM CREDITO

WHERE id\_credito IN (

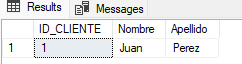
SELECT id\_credito

FROM PAGO\_MENSUAL

WHERE estado = 'Pendiente'

)

);

****

**Realice una consulta utilizando Group by y Having. Coloque su enunciado (2 puntos)**

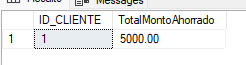
Obtener el total de monto ahorrado por cada cliente y mostrar solo aquellos clientes que tienen un total de ahorro superior a 10,000.

SELECT ID\_CLIENTE, SUM(monto) AS TotalMontoAhorrado

FROM AHORRO

GROUP BY ID\_CLIENTE

HAVING SUM(monto) < 10000;



**Realice función definida por el usuario. Coloque su enunciado (2 puntos)**

La función fn\_TotalSueldoPuesto calcula el sueldo total acumulado para un puesto específico (@codpuesto) en base a contratos existentes, ajustando el total según condiciones especiales para ciertos códigos de puesto ('PU03' y 'PU05').

CREATE FUNCTION dbo.fn\_TotalSueldoPuesto (@codpuesto CHAR(4))

RETURNS DECIMAL(10, 2)

AS

BEGIN

DECLARE @totalSueldo DECIMAL(10, 2)

DECLARE @tempSueldo DECIMAL(10, 2)

-- Total acumulado de sueldo para el puesto dado

SELECT @totalSueldo = SUM(sueldo)

FROM contrato

WHERE codpuesto = @codpuesto

-- Condiciones especiales

IF @codpuesto = 'PU03' AND (

SELECT COUNT(\*)

FROM empleado

WHERE codpuesto = 'PU03' AND DATEDIFF(YEAR, fecnac, GETDATE()) < 30

) <= 5

BEGIN

SET @totalSueldo = @totalSueldo + (@totalSueldo \* 0.3)

END

ELSE IF @codpuesto = 'PU05' AND (

SELECT COUNT(\*)

FROM empleado

WHERE codpuesto = 'PU05' AND DATEDIFF(YEAR, fecnac, GETDATE()) > 30

) > 5

BEGIN

-- Recalcular sueldo total excluyendo el puesto 'PU05'

SELECT @tempSueldo = SUM(sueldo)

FROM contrato

WHERE codpuesto != 'PU05'

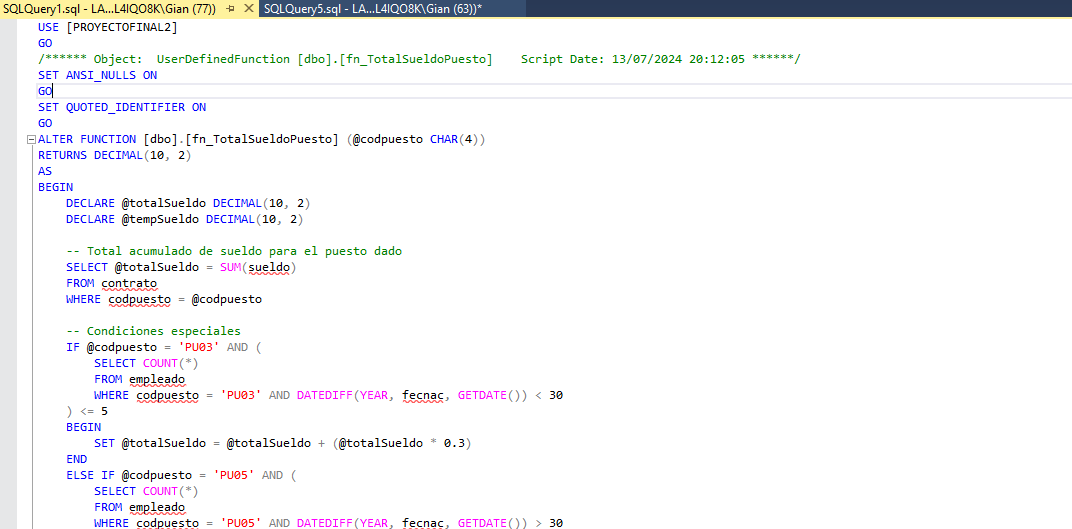
-- Asignar el sueldo recalculado a @totalSueldo

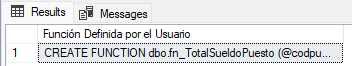
SET @totalSueldo = @tempSueldo

END

RETURN @totalSueldo

END

****

****

**Realice procedimiento almacenado. Coloque su enunciado (2puntos)**

El procedimiento almacenado usp\_InsertarEmpleado permite insertar un nuevo registro de empleado en la tabla Empleado, con los siguientes parámetros de entrada:

* @Nombre: Nombre del empleado (VARCHAR).
* @Apellido: Apellido del empleado (VARCHAR).
* @FechaNacimiento: Fecha de nacimiento del empleado (DATE).
* @CodPuesto: Código del puesto del empleado (CHAR(4)).

CREATE PROCEDURE usp\_InsertarEmpleado

@Nombre VARCHAR(100),

@Apellido VARCHAR(100),

@FechaNacimiento DATE,

@CodPuesto CHAR(4)

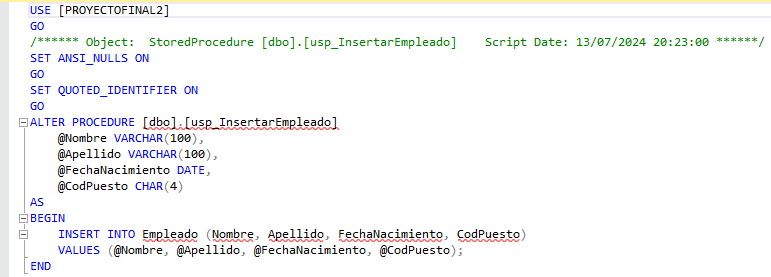
AS

BEGIN

INSERT INTO Empleado (Nombre, Apellido, FechaNacimiento, CodPuesto)

VALUES (@Nombre, @Apellido, @FechaNacimiento, @CodPuesto);

END





El trabajo final será presentado en un documento en Word y subido a su aula virtual en la semana 17, así mismo se realizará una exposición presencial en clase (en caso no se responda bien se penaliza con la disminución del puntaje).

Considerar que, si el modelo lógico propuesto está mal planteado, puede acarrear que las demás respuestas a las preguntas estén mal planteadas, penalizándose en el puntaje de la pregunta.