

Proyecto

Integrantes: Hernán Sánchez, Galo Tarapués

Materia: Aplicaciones Móviles

Fecha: 09/12/2024

Tema Proyecto: Control de Gastos Personales

El diagrama de abajo muestra las siguientes relaciones principales:

1. USUARIO:

- Es la entidad central del sistema
- Puede tener múltiples cuentas, metas, categorías y notificaciones
- Se relaciona con grupos familiares
- Crea planes financieros

2. CUENTA:

- Pertenece a un usuario
- Registra múltiples transacciones
- Puede tener servicios recurrentes asociados

3. TRANSACCIÓN:

- Está asociada a una cuenta
- Está clasificada en una categoría
- Registra movimientos financieros

4. CATEGORÍA:

- Pertenece a un usuario
- Clasifica transacciones
- Puede estar asociada a metas financieras

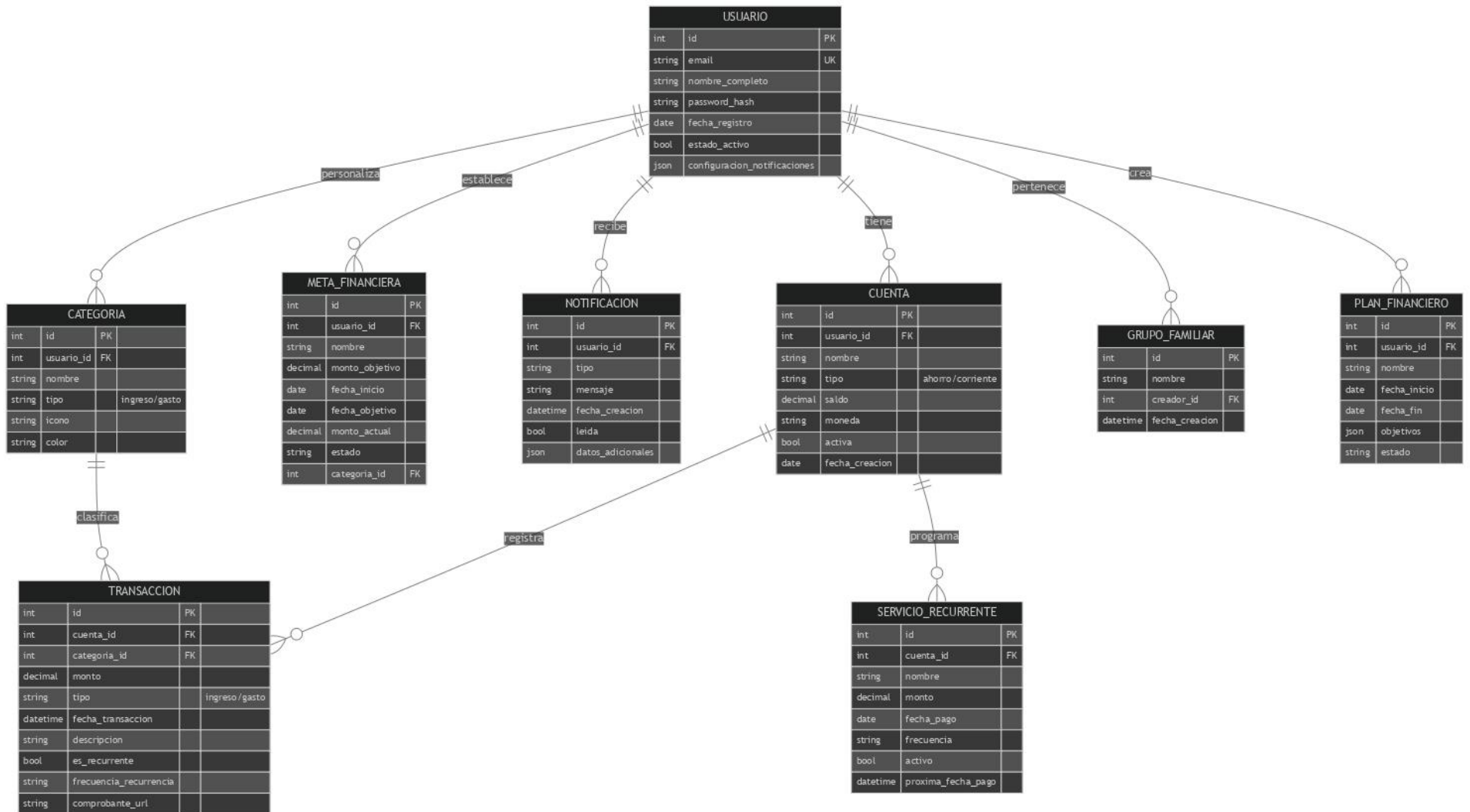
5. Elementos adicionales:

- **SERVICIO_RECURRENTE:** Para pagos programados
- **META_FINANCIERA:** Para objetivos de ahorro
- **PLAN_FINANCIERO:** Para estrategias financieras
- **GRUPO_FAMILIAR:** Para gestión familiar de finanzas
- **NOTIFICACION:** Para sistema de alertas

Las cardinalidades muestran que:

- Un usuario puede tener muchas cuentas, pero una cuenta pertenece a un solo usuario
- Una cuenta puede tener muchas transacciones
- Una categoría puede clasificar muchas transacciones
- Un usuario puede tener muchas metas y notificaciones

Diagrama Entidad Relación



Script:

-- Crear la base de datos

```
CREATE DATABASE sistema_financiero;
```

```
USE sistema_financiero;
```

-- Tabla de Usuarios

```
CREATE TABLE usuario (
```

```
    id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    email VARCHAR(100) UNIQUE NOT NULL,
```

```
    nombre_completo VARCHAR(200) NOT NULL,
```

```
    password_hash VARCHAR(255) NOT NULL,
```

```
    fecha_registro DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
    estado_activo BOOLEAN DEFAULT TRUE,
```

```
    configuracion_notificaciones JSON,
```

```
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
    updated_at    TIMESTAMP    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE  
CURRENT_TIMESTAMP  
);
```

-- Tabla de Cuentas

```
CREATE TABLE cuenta (
```

```
    id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    usuario_id INT NOT NULL,
```

```
    nombre VARCHAR(100) NOT NULL,
```

```
    tipo ENUM('ahorro', 'corriente') NOT NULL,
```

```
    saldo DECIMAL(15,2) DEFAULT 0.00,
```

```
    moneda VARCHAR(3) DEFAULT 'USD',
```

```
    activa BOOLEAN DEFAULT TRUE,
```

```
    fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP,
```

```
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
```

```
    updated_at    TIMESTAMP    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE  
CURRENT_TIMESTAMP,  
    FOREIGN KEY (usuario_id) REFERENCES usuario(id)  
);
```

-- Tabla de Categorías

```
CREATE TABLE categoria (
```

```
    id INT PRIMARY KEY AUTO_INCREMENT,
```

```
    usuario_id INT NOT NULL,
```

```
    nombre VARCHAR(50) NOT NULL,
```

```
    tipo ENUM('ingreso', 'gasto') NOT NULL,
```

```

    icono VARCHAR(50),
    color VARCHAR(7),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at    TIMESTAMP    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
    CURRENT_TIMESTAMP,
    FOREIGN KEY (usuario_id) REFERENCES usuario(id)
);

```

-- Tabla de Transacciones

```

CREATE TABLE transaccion (
    id INT PRIMARY KEY AUTO_INCREMENT,
    cuenta_id INT NOT NULL,
    categoria_id INT NOT NULL,
    monto DECIMAL(15,2) NOT NULL,
    tipo ENUM('ingreso', 'gasto') NOT NULL,
    fecha_transaccion DATETIME NOT NULL,
    descripcion TEXT,
    es_recorrente BOOLEAN DEFAULT FALSE,
    frecuencia_recurrencia VARCHAR(50),
    comprobante_url VARCHAR(255),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at    TIMESTAMP    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
    CURRENT_TIMESTAMP,
    FOREIGN KEY (cuenta_id) REFERENCES cuenta(id),
    FOREIGN KEY (categoria_id) REFERENCES categoria(id)
);

```

-- Tabla de Metas Financieras

```

CREATE TABLE meta_financiera (
    id INT PRIMARY KEY AUTO_INCREMENT,
    usuario_id INT NOT NULL,
    nombre VARCHAR(100) NOT NULL,
    monto_objetivo DECIMAL(15,2) NOT NULL,
    fecha_inicio DATE NOT NULL,
    fecha_objetivo DATE NOT NULL,
    monto_actual DECIMAL(15,2) DEFAULT 0.00,
    estado ENUM('pendiente', 'en_progreso', 'completada', 'cancelada') DEFAULT 'pendiente',
    categoria_id INT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at    TIMESTAMP    DEFAULT    CURRENT_TIMESTAMP    ON    UPDATE
    CURRENT_TIMESTAMP,
    FOREIGN KEY (usuario_id) REFERENCES usuario(id),

```

```

FOREIGN KEY (categoria_id) REFERENCES categoria(id)
);

-- Tabla de Notificaciones
CREATE TABLE notificacion (
  id INT PRIMARY KEY AUTO_INCREMENT,
  usuario_id INT NOT NULL,
  tipo VARCHAR(50) NOT NULL,
  mensaje TEXT NOT NULL,
  fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP,
  leida BOOLEAN DEFAULT FALSE,
  datos_adicionales JSON,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  FOREIGN KEY (usuario_id) REFERENCES usuario(id)
);

-- Tabla de Servicios Recurrentes
CREATE TABLE servicio_recurrente (
  id INT PRIMARY KEY AUTO_INCREMENT,
  cuenta_id INT NOT NULL,
  nombre VARCHAR(100) NOT NULL,
  monto DECIMAL(15,2) NOT NULL,
  fecha_pago DATE NOT NULL,
  frecuencia ENUM('diario', 'semanal', 'mensual', 'anual') NOT NULL,
  activo BOOLEAN DEFAULT TRUE,
  proxima_fecha_pago DATE,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
  FOREIGN KEY (cuenta_id) REFERENCES cuenta(id)
);

-- Tabla de Planes Financieros
CREATE TABLE plan_financiero (
  id INT PRIMARY KEY AUTO_INCREMENT,
  usuario_id INT NOT NULL,
  nombre VARCHAR(100) NOT NULL,
  fecha_inicio DATE NOT NULL,
  fecha_fin DATE NOT NULL,
  objetivos JSON,

```

```

    estado ENUM('activo', 'completado', 'cancelado') DEFAULT 'activo',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (usuario_id) REFERENCES usuario(id)
);

```

-- Tabla de Grupos Familiares

```

CREATE TABLE grupo_familiar (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    creador_id INT NOT NULL,
    fecha_creacion DATETIME DEFAULT CURRENT_TIMESTAMP,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    FOREIGN KEY (creador_id) REFERENCES usuario(id)
);

```

-- Tabla de relación entre Usuarios y Grupos Familiares

```

CREATE TABLE usuario_grupo_familiar (
    usuario_id INT NOT NULL,
    grupo_id INT NOT NULL,
    rol ENUM('admin', 'miembro') DEFAULT 'miembro',
    fecha_union DATETIME DEFAULT CURRENT_TIMESTAMP,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    PRIMARY KEY (usuario_id, grupo_id),
    FOREIGN KEY (usuario_id) REFERENCES usuario(id),
    FOREIGN KEY (grupo_id) REFERENCES grupo_familiar(id)
);

```

-- Crear índices para optimizar búsquedas

```

CREATE INDEX idx_usuario_email ON usuario(email);
CREATE INDEX idx_transaccion_fecha ON transaccion(fecha_transaccion);
CREATE INDEX idx_cuenta_usuario ON cuenta(usuario_id);
CREATE INDEX idx_categoria_usuario ON categoria(usuario_id);
CREATE INDEX idx_meta_usuario ON meta_financiera(usuario_id);

```

