CÁTEDRA DE BASE DE DATOS AVANZADA

Universidad del Aconcagua

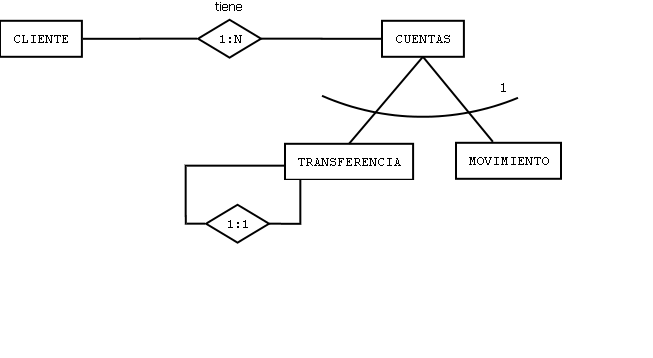
TRABAJO PRÁCTICO

N° 4

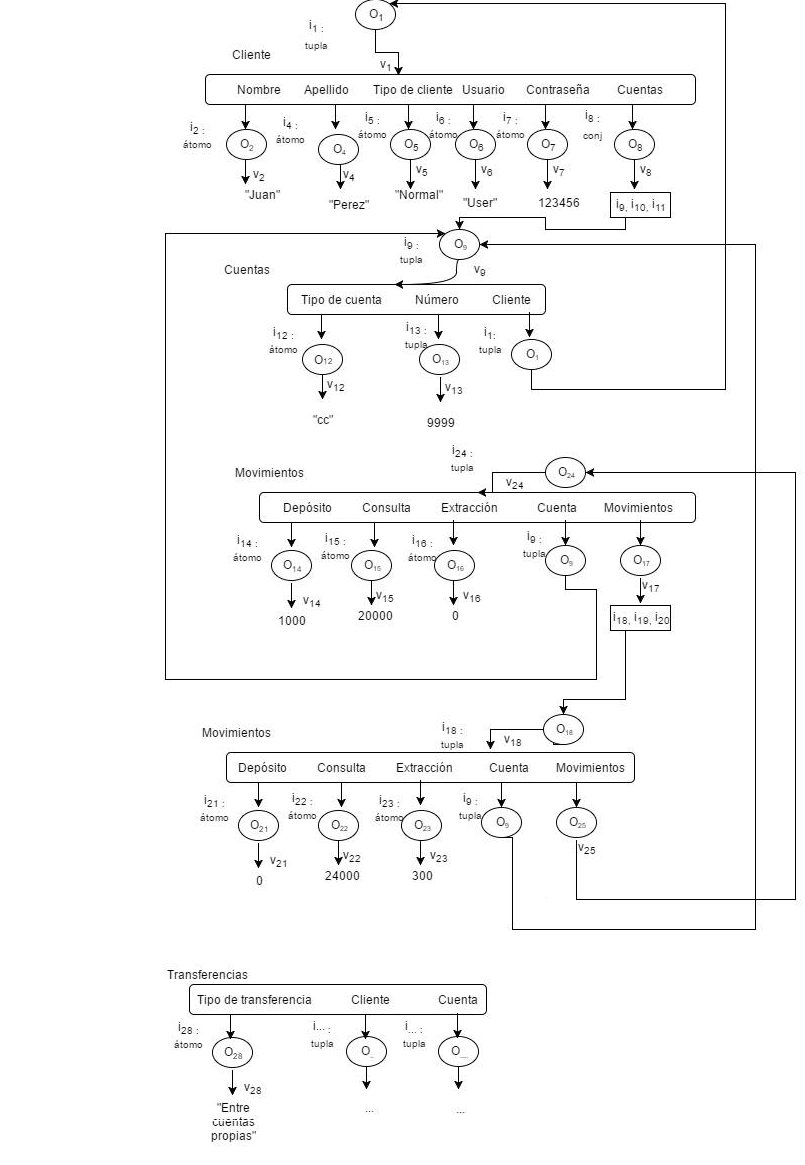
Sigal Nicolás – 110994

**CAJERO AUTOMÁTICO**

**DIAGRAMA ENTIDAD-RELACION**



**GRAFICO DE OBJETOS**



**DESCRIPCION LITERAL DE OBJETOS**

**O1 =** ( i1, tupla,< Nombre:i2;Apellido:i3;Tipo de cliente:i5;Usuario:i6;Contraseña:i7;Cuentas:i8>)

**O2=**(i2,atomo,"Juan")

**O4=**(i4,atomo,"Perez")

**O5=**(i5,atomo,"Normal")

**O6=**(i6,atomo,"User")

**O7=**(i7,atomo,"123456")

**O8=**(i8,conjunto,{i9,i10,i11})

**O9=**(i9,tupla, <Tipo de cuenta:i12;numero:i13;cliente:i1>)

**O12=**(i2,atomo,"cc")

**O13=**(i13,atomo,"9999")

**O14=**(i14,atomo,"1000")

**O15=**(i15,atomo,"20000")

**O16=**(i16,atomo,"0")

**O17=**(i17,conjunto,{i18,i19,i20})

**O18=**(i18,tupla ,< Deposito:i21;Consulta:i22;Extraccion:i23;Cuenta:i9,Movimientos:i25>)

**O25**=(i25,tupla, ,< Deposito:i21;Consulta:i22;Extraccion:i23;Cuenta:i9,Movimientos:i25>)

**O28=**(i28,atomo,"entre cuentas propias")

**ESQUEMA ODL- DEFINICION DE OBJETOS**

**CLIENTE**

Define Class Cliente

Tupla( Nombre: String;

Apellido: String;

tipo de cliente: String;

usuario: String;

contraseña: int;

cuentas: set(Cuentas););

operation ( crear\_cliente: Cliente;

eliminar\_cliente: boolean;

asignar\_cuenta(e:Cuenta):boolean;

quitar\_cuenta(e:Cuenta):boolean;

End Cliente;

**CUENTA**

Define Class Cuenta

Tupla( Tipo de cuenta: String;

Numero: int;

cliente: tupla(cliente:Cliente);

operation ( crear\_cuenta:cuenta;

eliminar\_cuenta:boolean;

asignar\_cliente(e:Cliente):boolean;

quitar\_cliente(e:Cliente):boolean;);

End Cuenta;

**MOVIMIENTOS**

Define Class Movimiento

Tupla( Deposito: int;

Consulta:int;

Extraccion: int;

Cuenta: int;

movimientos: tupla(movimientos:Movimientos);

cuenta:tupla(cuenta:Cuenta));

operation ( crear\_movimiento:Movimiento;

eliminar\_movimiento:boolean;

asignar\_cuenta (e:Cuenta):boolean;

quitar\_cuenta(e:Cuenta):boolean;);

crear\_movimiento:(e:Movimiento):boolean;

eliminar\_movimiento:(e:Movimiento):boolean;

End Movimiento;

**TRANSFERENCIA**

Define Class Transferencia

Tupla( tipo de tranferencia: String;

cliente:tupla (cliente:Cliente);

cuenta: tupla (cuenta:Cuenta););

operation ( crear\_tranferencia:transferencia;

eliminar\_transferencia:boolean;

asignar\_cliente(e:Cliente):boolean;

asignar\_cuenta (e:Cuenta):boolean;

quitar\_cliente(e:Cliente):boolean;

quitar\_cuenta(e:Cuenta):boolean;

End Transferencia;

**PERSISTENACIA**

**CLIENTE**

Define ClassConjuntoClientes

Type Set (Cliente);

Operations añadir\_clte(c:Cliente):boolean;

quitar\_clte(c:Cliente):boolean;

crear\_conj\_clte:Conjunto Clientes;

borrar\_conj\_clte: boolean;

EndConjuntoClientes

PersistentNameTodos\_Cliente:ConjuntoClientes

**MOVIMIENTOS**

Define ClassConjuntoMovimientos

Type Set (Movimiento);

Operationsañadir\_mov(m:Movimiento):boolean;

quitar\_mov(m:Movimiento):boolean;

crear\_conj\_mov:Conjunto Movimientos;

borrar\_conj\_mov: boolean;

EndConjuntoMovimientos

PersistentNameTodos\_Movimientos:ConjuntoMovimientos

**CUENTA**

Define ClassConjuntoCuentas

Type Set (Cuenta);

Operationsañadir\_cuen(c:Cuentas):boolean;

quitar\_cuen(c:Cuentas):boolean;

crear\_conj\_cuen:Conjunto Cuentas;

borrar\_conj\_cuen: boolean;

EndConjuntoCuentas

PersistentNameTodos\_Cuentas:ConjuntoCuentas

**TRANSFERENCIA**

Define ClassConjuntoTransferencia

Type Set (Transferencia);

Operationsañadir\_trans(t:Transferencia):boolean;

quitar\_trans(t:Transferencia):boolean;

crear\_conj\_transf:Conjunto Transferencias;

borrar\_conj\_transf: boolean;

EndConjuntoTransferencia

PersistentNameTodos\_Transferencias:ConjuntoTransferencia