

Proyecto #1

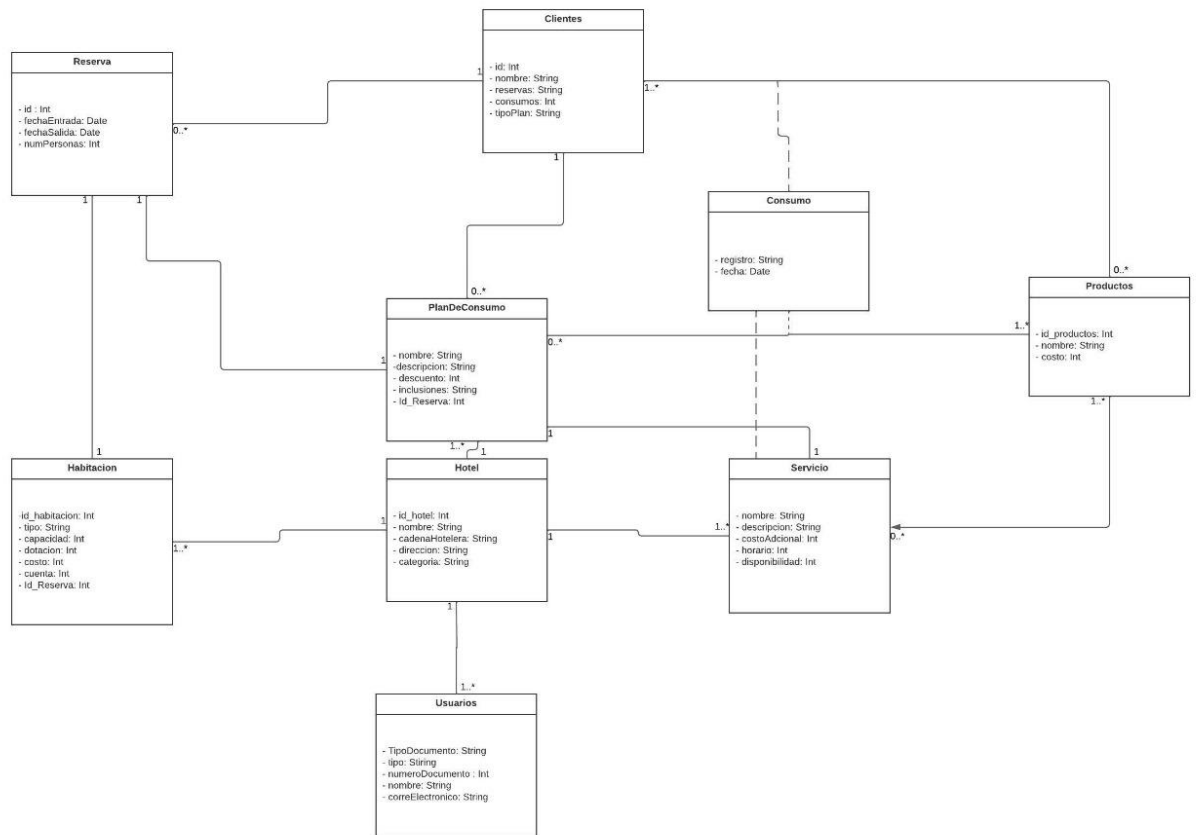
Sistemas Transaccionales

Juan Ortega - 202113442

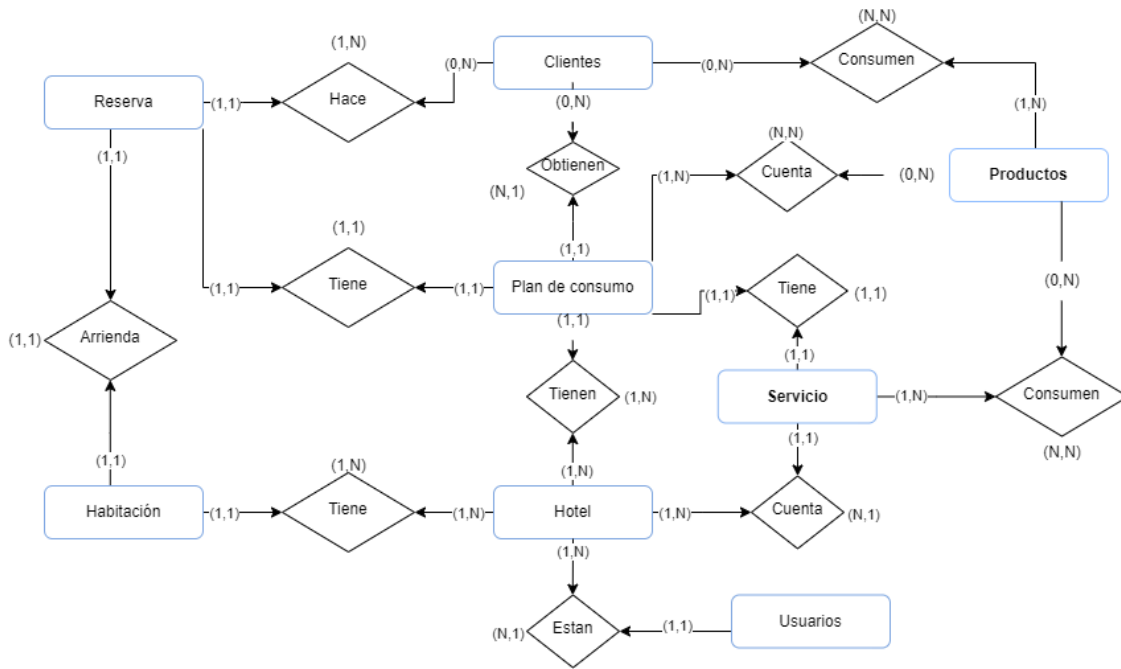
Santiago Tenjo - 202113965

Jose David Martinez - 202116677

1. Modelo Conceptual UML

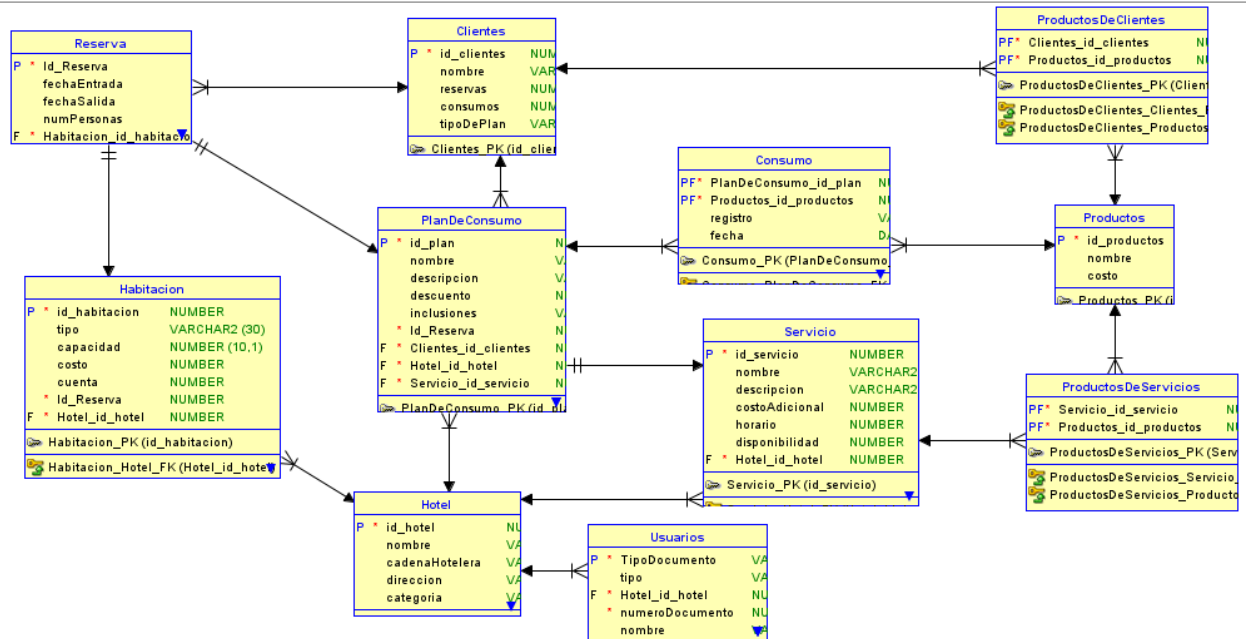


2. Modelo Conceptual Entidad Relación (E/R)



3. Modelo de Datos Relacional

- Modelo en Data Modeler



- Modelo manual

Reserva						
fechaEntrada	fechaSalida	numPersonas	Id_reserva	habAsignada	planAsignado	ReservaCliente
CK, NN	CK, NN	CK, NN	PK	FK<Habitacion, idhabitacion>	FK<PlanConsumo, idPlan>	FK<Cliente, idCliente>

Cliente				
idCliente	nombre	reservas	consumos	TipoDePlan
PK, UA	NN, ND	NN	NN	NN

PlanDeConsumo							
idPlan	nombre	descripcion	descuento	inclusiones	PlanCliente	PlanHoteles	PlanServicio
PK, SA	NN, ND	NN	NN	NN	FK<Cliente, idCliente>	FK<Hotel, idHotel>	FK<Servicio, idServicio>

Habitacion					
idHabitacion	tipo	capacidad	costo	cuenta	HabHotel
PK, SA	NN	NN	NN	NN	FK<Hotel, idHotel>

Hotel					ProductoServicio		Consumo		
idHotel	nombre	cadenaHotelera	direccion	categoria	CodigoServ	CodigoProd	Registro	Fecha	CodigoProd
PK, SA	NN, ND	NN	NN	NN	FK<Servicio, idServicio>	FK<Producto, idProducto>	NN, ND	CK, NN	FK<Producto, idProducto>
							PlanConsumo		
							FK<PlanConsumo, idPlan>		
Servicio						ProductoCliente			
idServicio	nombre	descripcion	costoAdicional	horario	disponibilidad	ServicioHotel	CodigoProd	idCliente	
PK, SA	NN	NN, ND	NN	NN	NN	FK<Hotel, idHotel>	FK<Producto, idProducto>	FK<Cliente, idCliente>	
Producto			Usuarios						
idProducto	nombre	costo	numDocumento	tipoDocumento	tipoUsuario	nombre	correoElectronico	HotelUsuario	
PK, SA	NN	CK	PK, UA	NN	NN	NN, ND	NN, ND	FK<Hotel, idHotel>	

4. Normalización

- **Reserva:** FechaEntrada(FE), FechaSalida(FS), numPersonas(P), idReserva(ID)
ID -> Fe ; ID -> Fs ; ID -> P
Primos: ID NoPrimos: Fe, Fs, P
 - 1ra forma: Si cumple, dado que no hay datos multivariados
 - 2da forma: Si cumple dado que no hay dependencias parciales, todos dependen de IdReserva.
 - 3ra forma: Si cumple dado que no hay dependencias transitivas, todas dependen de IdReserva, no de algún otro atributo NoPrimo.
 - BCNF: Si está en Boyce-Codd dado que no existe llave candidata compuesta.
- **Cliente:** Nombre(N), Reservas(R), Consumos(C), idCliente(ID), TipoDePlan(TP)
ID -> N ; ID -> R ; ID -> C ; ID -> TP ; N -> R
Primos: ID, N NoPrimos: R, C, TP
 - 1ra forma: Si cumple, dado que no hay datos multivariados
 - 2da forma: Si cumple dado que no hay dependencias parciales, todos dependen de IdCliente y Nombre.
 - 3ra forma: Si cumple dado que no hay dependencias transitivas, todas dependen de IdReserva o Nombre y no de algún otro atributo NoPrimo.
 - BCNF: Si está en Boyce-Codd dado que no tiene llaves candidatas compuestas.
- **PlanDeConsumo:** Nombre(N), Descripción(D), Descuento(DC), idPlan(ID), Inclusiones(I)
ID -> N ; ID -> D ; ID -> DC ; ID -> I
Primos: ID NoPrimos: N, D, DC, I
 - 1ra forma: Si cumple, dado que no hay datos multivariados
 - 2da forma: Si cumple dado que no hay dependencias parciales, todos dependen de IdPlan.
 - 3ra forma: Si cumple dado que no hay dependencias transitivas, todas dependen de IdPlan y no de algún otro atributo NoPrimo.
 - BCNF: Si está en Boyce-Codd dado que no tiene llaves candidatas compuestas.
- **Habitación:** Tipo(T), Capacidad(C), Costo(CO), Cuenta(CU), idHabitacion(ID)
ID -> C ; ID -> CO ; ID -> CU ; ID -> T ; T -> C ; T -> CO
Primos: ID, T NoPrimos: C, CO, CU
 - 1ra forma: Si cumple, dado que no hay datos multivariados
 - 2da forma: No cumple dado que hay dependencias parciales, porque Tipo por si solo puede determinar capacidad y costo. Se podía arreglar separando Tipo en otra tabla con su respectiva capacidad y costo. (Resultado: Habitación (T, CU, ID), Tipo (T, C, CO))
 - 3ra forma: No cumple.
 - BCNF: No cumple.

- **Hotel:** Nombre(N), CadenaHotelera(CH), Direccion(D), Categoria(C), idHotel(ID)
 $ID \rightarrow N ; ID \rightarrow CH ; ID \rightarrow D ; ID \rightarrow C$
 Primos: ID NoPrimos: N, CH, D, C
 - 1ra forma: Si cumple, dado que no hay datos multivariados
 - 2da forma: Si cumple dado que no hay dependencias parciales, todos dependen de IdHotel.
 - 3ra forma: Si cumple dado que no hay dependencias transitivas, todas dependen de IdHotel y no de algún otro atributo NoPrimo.
 - BCNF: Si está en Boyce-Codd dado que no tiene llaves candidatas compuestas.

- **Servicio:** Nombre(N), Descripción(D), CostoAdicional(CA), Horario(H), Disponibilidad(DS), idServicio(ID)
 $ID \rightarrow N ; ID \rightarrow D ; ID \rightarrow CA ; ID \rightarrow H ; ID \rightarrow DS$
 Primos: ID NoPrimos: N, D, CA, H, DS
 - 1ra forma: Si cumple, dado que no hay datos multivariados
 - 2da forma: Si cumple dado que no hay dependencias parciales, todos dependen de IdServicio.
 - 3ra forma: Si cumple dado que no hay dependencias transitivas, todas dependen de idServicio y no de algún otro atributo NoPrimo.
 - BCNF: Si está en Boyce-Codd dado que no tiene llaves candidatas compuestas.

- **Producto:** Nombre(N), Costo(C), idProducto(ID)
 $ID \rightarrow N ; ID \rightarrow C$
 Primos: ID NoPrimos: N, C
 - 1ra forma: Si cumple, dado que no hay datos multivariados
 - 2da forma: Si cumple dado que no hay dependencias parciales, todos dependen de IdProducto.
 - 3ra forma: Si cumple dado que no hay dependencias transitivas, todas dependen de IdProducto y no de algún otro atributo NoPrimo.
 - BCNF: Si está en Boyce-Codd dado que no tiene llaves candidatas compuestas.

- **Usuario:** Nombre(N), numDocumento(ND), tipoDocumento(TD), tipoUsuario(TU), CorreoElectronico(CE)
 $ID \rightarrow N ; ID \rightarrow R ; ID \rightarrow C ; ID \rightarrow TP ; N \rightarrow R$
 Primos: ND NoPrimos: N, TD, TU, CE
 - 1ra forma: Si cumple, dado que no hay datos multivariados
 - 2da forma: Si cumple dado que no hay dependencias parciales, todos dependen de NumeroDocumento.
 - 3ra forma: Si cumple dado que no hay dependencias transitivas, todas dependen de IdReserva o Nombre y no de algún otro atributo NoPrimo.
 - BCNF: Si está en Boyce-Codd dado que no tiene llaves candidatas compuestas.

5. Creación de tablas

```
CREATE TABLE clientes (  
    id_clientes NUMBER NOT NULL,  
    nombre     VARCHAR2(20),  
    reservas  NUMBER,  
    consumos  NUMBER,  
    tipodeplan VARCHAR2(20)  
);
```

```
ALTER TABLE clientes ADD CONSTRAINT clientes_pk PRIMARY KEY ( id_clientes );  
                    productos_id_productos );
```

```
CREATE TABLE habitacion (  
    id_habitacion NUMBER NOT NULL,  
    tipo          VARCHAR2(30),  
    capacidad     NUMBER(10, 1),  
    costo         NUMBER,  
    cuenta       NUMBER  
);
```

```
ALTER TABLE habitacion ADD CONSTRAINT habitacion_pk PRIMARY KEY ( id_habitacion  
);
```

```
CREATE TABLE hotel (  
    id_hotel     NUMBER NOT NULL,  
    nombre       VARCHAR2(20),  
    cadenahotelera VARCHAR2(40),  
    direccion    VARCHAR2(20),  
    categoria    VARCHAR2(20)  
);
```

```
ALTER TABLE hotel ADD CONSTRAINT hotel_pk PRIMARY KEY ( id_hotel );
```

```
CREATE TABLE plandeconsumo (  
    id_plan      NUMBER NOT NULL,  
    nombre       VARCHAR2(30),  
    descripcion  VARCHAR2(100),  
    descuento   NUMBER,  
    inclusiones  VARCHAR2(100)  
);
```

```
ALTER TABLE plandeconsumo ADD CONSTRAINT plandeconsumo_pk PRIMARY KEY ( id_plan );
```

```
CREATE TABLE productos (  
    id_productos NUMBER NOT NULL,  
    nombre       VARCHAR2(20),
```

```
    costo    NUMBER
);
```

```
ALTER TABLE productos ADD CONSTRAINT productos_pk PRIMARY KEY ( id_productos );
```

```
CREATE TABLE reserva (
    id_reserva    NUMBER NOT NULL,
    fechaentrada  VARCHAR2(10),
    fechasalida   VARCHAR2(10),
    numpersonas   NUMBER
);
```

```
ALTER TABLE reserva ADD CONSTRAINT reserva_pk PRIMARY KEY ( id_reserva );
```

```
CREATE TABLE servicio (
    id_servicio   NUMBER NOT NULL,
    nombre        VARCHAR2(20),
    descripcion   VARCHAR2(100),
    costoadicional NUMBER,
    horario       NUMBER,
    disponibilidad NUMBER
);
```

```
ALTER TABLE servicio ADD CONSTRAINT servicio_pk PRIMARY KEY ( id_servicio );
```

```
CREATE TABLE usuarios (
    tipodocumento VARCHAR2(20),
    tipo            VARCHAR2(20),
    numerodocumento NUMBER NOT NULL,
    nombre          VARCHAR2(20),
    correoelectronico VARCHAR2(40)
);
```

```
ALTER TABLE usuarios ADD CONSTRAINT usuarios_pk PRIMARY KEY (
    numerodocumento );
```

```
ALTER TABLE habitacion
    ADD CONSTRAINT habitacion_hotel_fk FOREIGN KEY ( hotel_id_hotel )
    REFERENCES hotel ( id_hotel );
```

```
ALTER TABLE plandeconsumo
    ADD CONSTRAINT plandeconsumo_clientes_fk FOREIGN KEY ( clientes_id_clientes )
```

```

REFERENCES clientes ( id_clientes );

ALTER TABLE plandeconsumo
ADD CONSTRAINT plandeconsumo_hotel_fk FOREIGN KEY ( hotel_id_hotel )
REFERENCES hotel ( id_hotel );

ALTER TABLE plandeconsumo
ADD CONSTRAINT plandeconsumo_servicio_fk FOREIGN KEY ( servicio_id_servicio )
REFERENCES servicio ( id_servicio );

ALTER TABLE reserva
ADD CONSTRAINT reserva_clientes_fk FOREIGN KEY ( clientes_id_clientes )
REFERENCES clientes ( id_clientes );

ALTER TABLE reserva
ADD CONSTRAINT reserva_habitacion_fk FOREIGN KEY ( habitacion_id_habitacion )
REFERENCES habitacion ( id_habitacion );

ALTER TABLE reserva
ADD CONSTRAINT reserva_plandeconsumo_fk FOREIGN KEY ( plandeconsumo_id_plan )
REFERENCES plandeconsumo ( id_plan );

ALTER TABLE servicio
ADD CONSTRAINT servicio_hotel_fk FOREIGN KEY ( hotel_id_hotel )
REFERENCES hotel ( id_hotel );

ALTER TABLE usuarios
ADD CONSTRAINT usuarios_hotel_fk FOREIGN KEY ( hotel_id_hotel )
REFERENCES hotel ( id_hotel );

```

6. Escenarios de pruebas

```

-- Insertar tupla 1
INSERT INTO Productos (id_productos, nombre, costo) VALUES ('1', 'Llavero', '5500');

-- Intentar insertar tupla 2 con la misma PK
INSERT INTO Productos (id_productos, nombre, costo) VALUES ('1', 'Pulser', '3000');

-- Insertar tupla 1 con FK existente
INSERT INTO Reservas (id_Reserva, fechaEntrada, fechaSalida, numPersonas,
Habitacion_id_habitacion) VALUES ('1', '1/10/23', '2/12/23', 4, 'FK_existente');

-- Intentar insertar tupla 1 con FK no existente
INSERT INTO tabla (id_Reserva, fechaEntrada, fechaSalida, numPersonas,
Habitacion_id_habitacion) VALUES ('2', '2/10/23', '3/12/23', 6, 'FK_no_existente');

```



```
-- Insertar tupla que viola restricciones de chequeo
INSERT INTO Hotel (id_hotel, nombre, cadenaHotelera, direccion, categoria) VALUES ('1',
Null, 'Decameron','Carrer 4a con 14', 'Resort' );
```