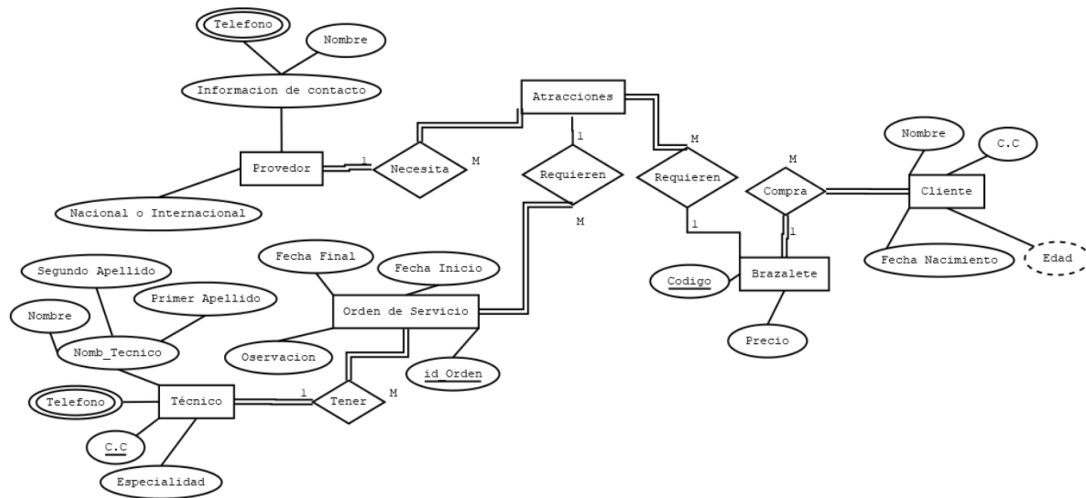


## Modelo Entidad-Relacion de Nervardo



Revisando el modelo entidad de mi compañero veo que la estructura está bien, solo cambiaria unos verbos de las relaciones, algunas cardinalidades, agregaría algunos atributos y entidades.

### Entidades y atributos:

Operador: Agrego la entidad Operador con sus respectivos atributos y la relacionaría con atracciones, ya que el operador es el que maneja una atracción.

Proveedor: Agrego el Id al proveedor para que tenga llave primaria.

Atracciones: Le agrego los atributos IdAtraccion y nombre.

Técnico: Especialidad la vuelvo un atributo multivaluado.

### Relaciones:

Creo la relación entre operador y atracción llamada "Manipula".

Cambio la relación entre proveedor y atracción de "Necesita" a "Vende".

Cambio la relación entre técnico y orden de servicio de "Tener" a "Lleva a cabo".

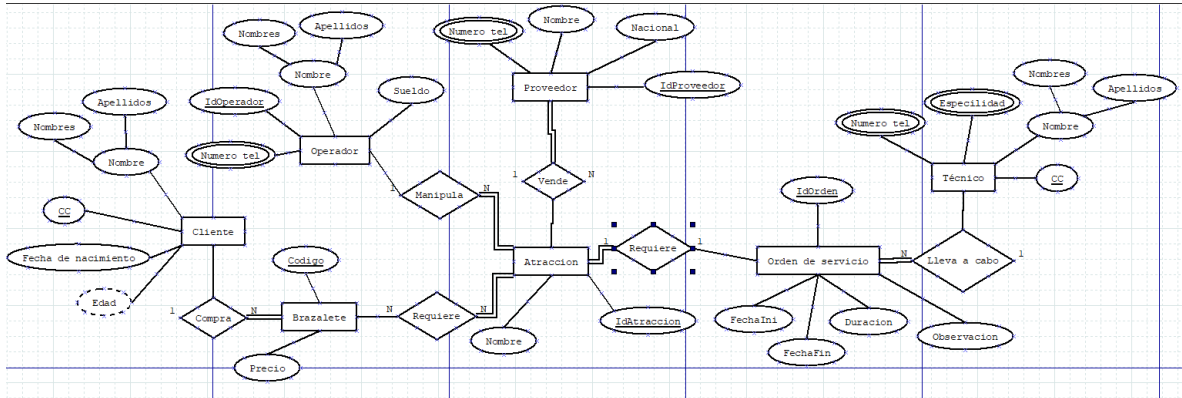
### Cardinalidades y participaciones:

Arreglo la cardinalidad de clientes con brazaletes de N:1 a 1:N, ya que considero que es más lógico que un cliente pueda comprar varios brazaletes a que un brazalete sea comprado por varios clientes.

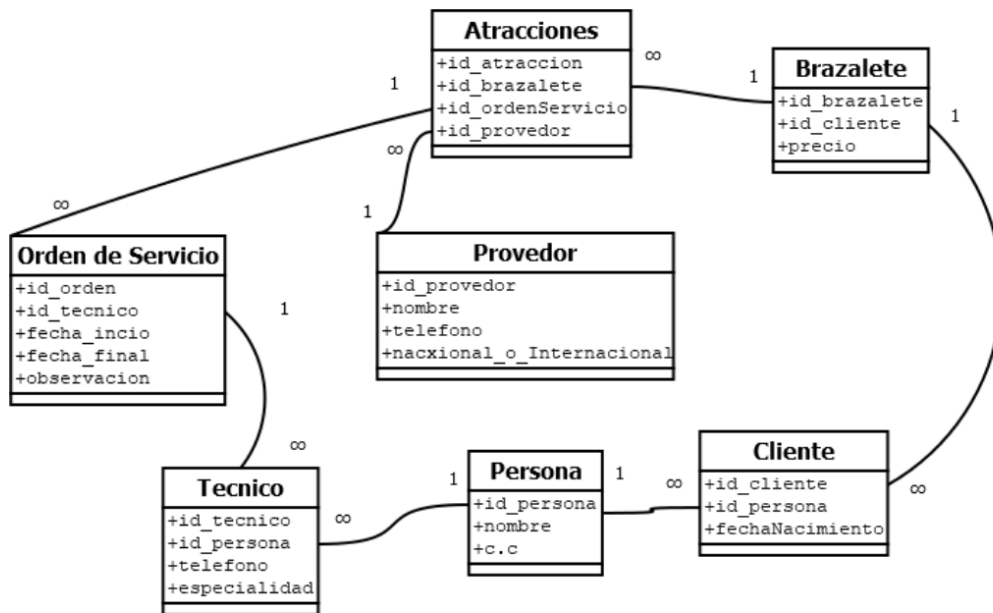
Arreglo la cardinalidad de brazalete con atracciones de N:M, ya que considero que un brazalete sirve para entrar a varias atracciones y a una atracción se puede entrar con distintos brazaletes.

Agrego la cardinalidad de proveedor con atracción 1:N, un proveedor puede proveer uno o varias atracciones, pero una atracción solo la puede proveer un proveedor.

**Resultado al aplicar las correcciones:**



### Modelo Relacional de Nevardo:

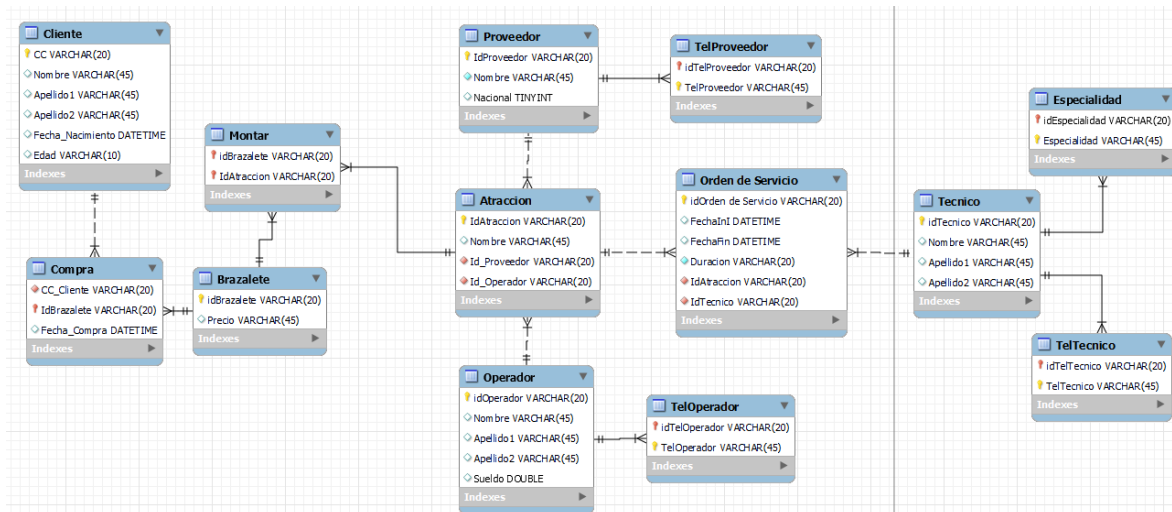


El modelo relacional del compañero no se fija en sus Cardinalidades, por ende, no aplicar las reglas para relacionar las tablas, es decir, no crea llaves foráneas que conecten las tablas.

También crea una nueva entidad “Persona” en la que trata de hacer una herencia, lo cual no es recomendable.

Para corregir el modelo relacional me voy a basar en el modelo entidad relación después de las correcciones que creí necesarias.

**Resultado al aplicar las correcciones:**



## Código generado:

```
1  -- MySQL Workbench Forward Engineering
2
3  SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
4  SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
5  SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
6
7  -- Schema mydb
8  -----
9
10
11
12  -- Schema mydb
13  -----
14  CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;
15  USE `mydb` ;
16
17  -----
18  -- Table `mydb`.`Proveedor`
19  -----
20  CREATE TABLE IF NOT EXISTS `mydb`.`Proveedor` (
21    `IdProveedor` VARCHAR(20) NOT NULL,
22    `Nombre` VARCHAR(45) NOT NULL,
23    `Nacional` TINYINT NULL,
24    PRIMARY KEY (`IdProveedor`))
25  ENGINE = InnoDB;
26
27  -----
28  -- Table `mydb`.`Operador`
29  -----
30
31  CREATE TABLE IF NOT EXISTS `mydb`.`Operador` (
32    `IdOperador` VARCHAR(20) NOT NULL,
```

## Schema generado:



Crear modelo de forma manual:

Creación del schema:

32

33 • `CREATE SCHEMA `parque` ;`

34

Creación de tablas:

The screenshot displays a database management interface. On the left, a tree view shows the database structure. The 'parque' schema is expanded, showing tables: 'atraccion', 'montar', 'operador', 'proveedor', and 'telproveedor'. The 'sys' schema is also visible. The main pane on the right shows the SQL code for creating the schema and tables. The code is as follows:

```
33 • CREATE SCHEMA `parque` ;
34
35 • CREATE TABLE `parque`.`atraccion` (
36     `idAtraccion` VARCHAR(20) NOT NULL,
37     `Nombre` VARCHAR(45) NULL,
38     `IdProveedor` VARCHAR(45) NOT NULL,
39     `IdOperador` VARCHAR(45) NOT NULL,
40     PRIMARY KEY (`idAtraccion`));
41
42 • CREATE TABLE `parque`.`proveedor` (
43     `idProveedor` VARCHAR(20) NOT NULL,
44     `Nombre` VARCHAR(45) NULL,
45     `Nacional` BOOLEAN NOT NULL,
46     PRIMARY KEY (`idProveedor`));
47
48 • CREATE TABLE `parque`.`telproveedor` (
49     `idProveedor` VARCHAR(20) NOT NULL,
50     `telProveedor` VARCHAR(45) NOT NULL,
51     PRIMARY KEY (`idProveedor`,`telProveedor`));
52
53 • CREATE TABLE `parque`.`operador` (
54     `idOperador` VARCHAR(20) NOT NULL,
55     `Nombre` VARCHAR(45) NULL,
56     `Apellido1` VARCHAR(45) NULL,
57     `Apellido2` VARCHAR(45) NULL,
58     `Sueldo` DOUBLE NOT NULL,
59     PRIMARY KEY (`idOperador`));
60
61 • CREATE TABLE `parque`.`teloperador` (
62     `idOperador` VARCHAR(20) NOT NULL,
63     `telOperador` VARCHAR(45) NOT NULL,
64     PRIMARY KEY (`idOperador`,`telOperador`));
65
```

De esta manera fueron creadas todas las tablas con sus respectivos atributos y tipos de datos, y las relaciones entre tablas se hicieron de mediante el programa como se vio en clases.