

Primera Actividad

Para dar solución a esta actividad, se realiza la lectura de los requerimientos y condiciones que tiene la librería y se adicionan algunos atributos que no se consideran en el texto inicial. Así, se obtiene el MER que da solución a este problema.

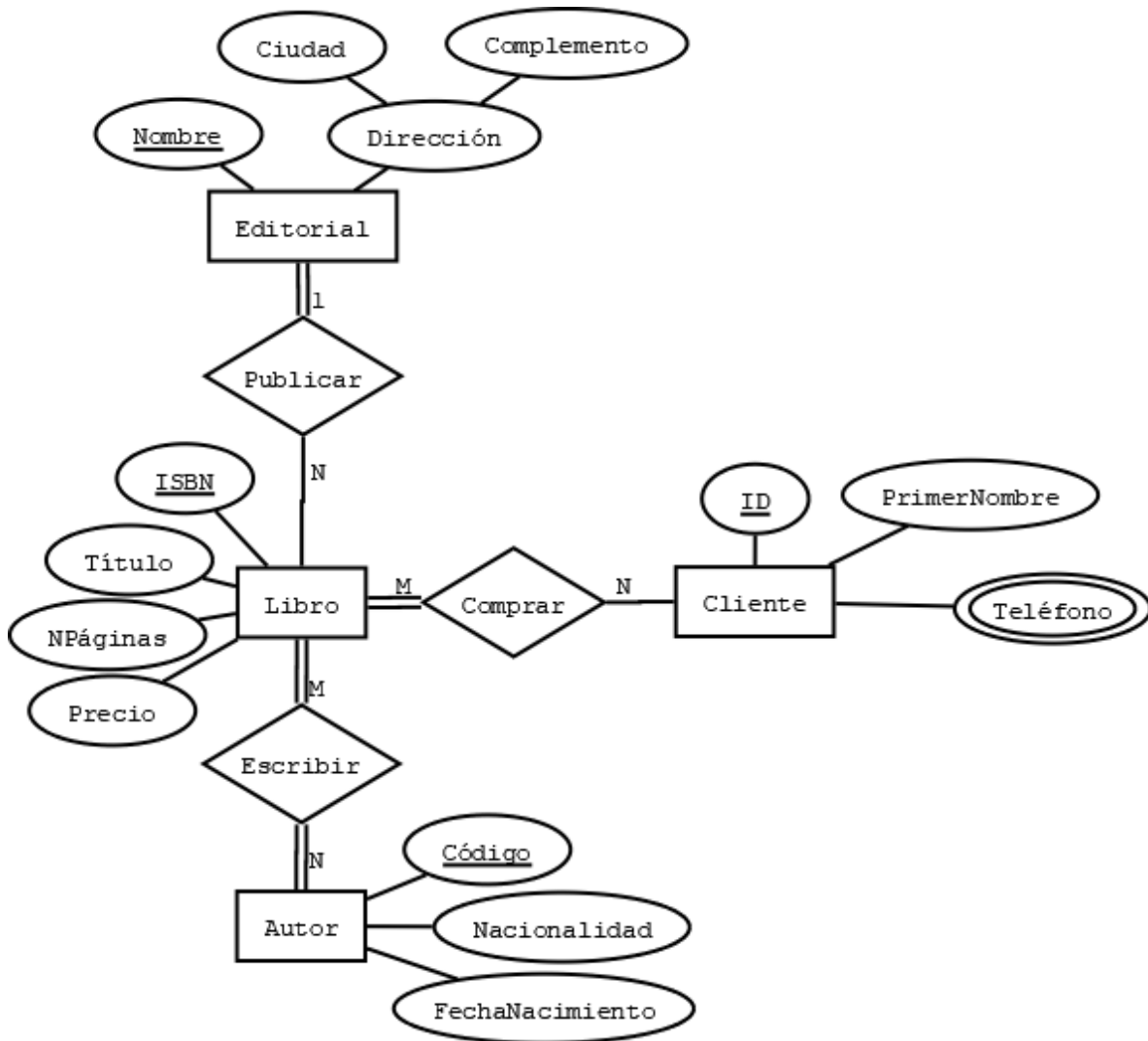


Figura 1 MER Librería BuscaLibre.

Luego, se realiza la transformación a MR directamente en Workbench, ya que así se evita realizar doble trabajo. El resultado se aprecia en la Figura 2.

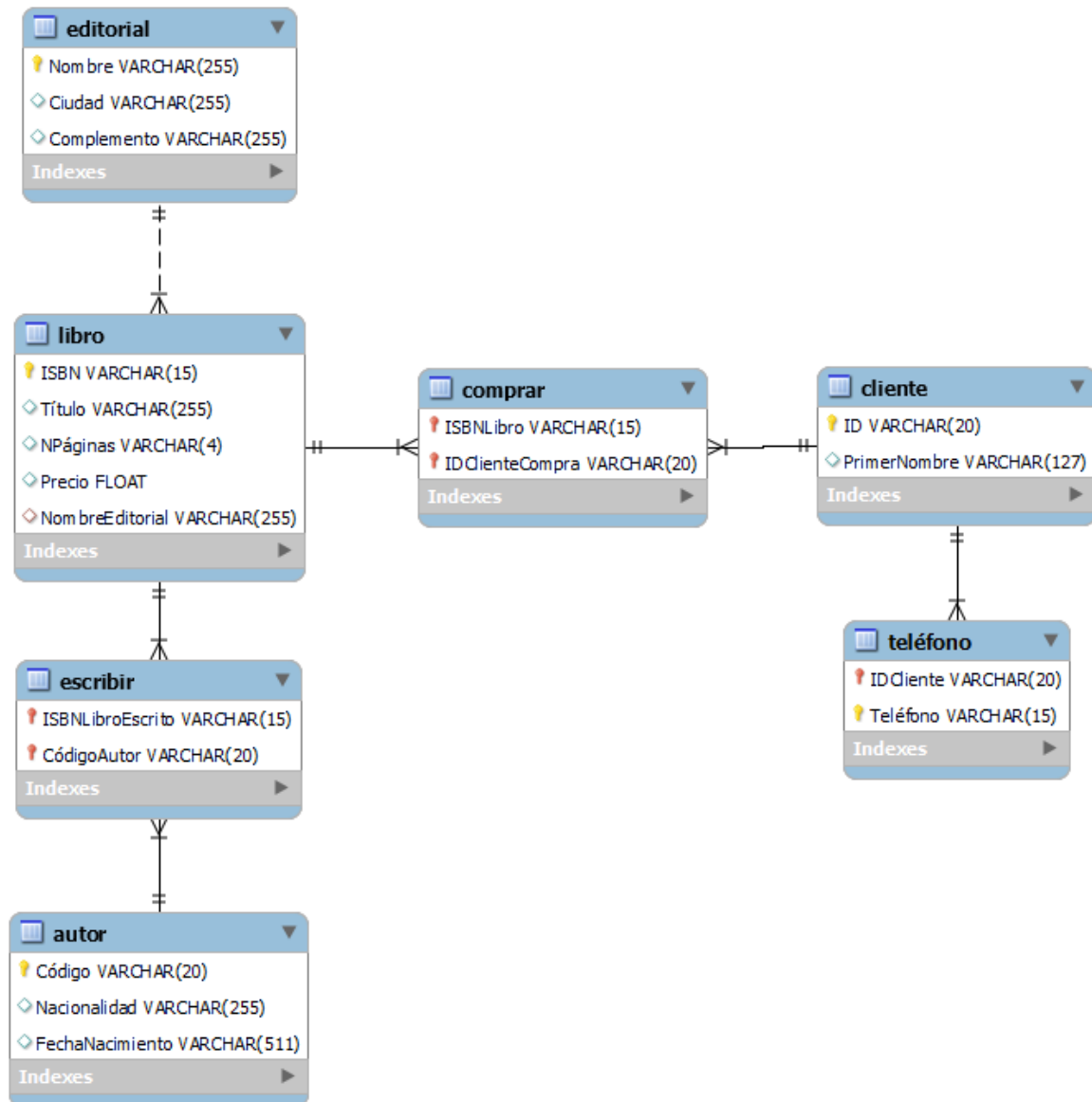


Figura 2 MR Librería BuscaLibre.

Finalmente, se realiza la traducción de este modelo a sentencias SQL, de la siguiente manera.


```

CREATE SCHEMA `primeraActividad`;
CREATE TABLE `primeraActividad`.`Editorial` (
  Nombre varchar(255) primary key,
  Ciudad varchar(255),
  Complemento varchar(255)
);
CREATE TABLE `primeraActividad`.`Libro` (
  ISBN varchar(15) primary key,
  Título varchar(255),
  NPáginas varchar(4),
  Precio float(15),
  NombreEditorial varchar(255),
  foreign key(NombreEditorial) references Editorial(Nombre)
);
CREATE TABLE `primeraActividad`.`Autor` (
  Código varchar(20) primary key,
  Nacionalidad varchar(255),
  FechaNacimiento varchar(511)
);
CREATE TABLE `primeraActividad`.`Cliente` (
  ID varchar(20) primary key,
  PrimerNombre varchar(127)
);
CREATE TABLE `primeraActividad`.`Teléfono` (
  IDCliente varchar(20),
  Teléfono varchar(15),
  primary key(IDCliente, Teléfono),
  foreign key(IDCliente) references Cliente(ID)
);
CREATE TABLE `primeraActividad`.`Comprar` (
  ISBNLibro varchar(15),
  IDClienteCompra varchar(20),
  primary key(ISBNLibro, IDClienteCompra),
  foreign key(ISBNLibro) references Libro(ISBN),
  foreign key(IDClienteCompra) references Cliente(ID)
);
CREATE TABLE `primeraActividad`.`Escribir` (
  ISBNLibroEscrito varchar(15),
  CódigoAutor varchar(20),
  primary key(ISBNLibroEscrito, CódigoAutor),
  foreign key(ISBNLibroEscrito) references Libro(ISBN),
  foreign key(CódigoAutor) references Autor(Código)
);

```

Segunda Actividad

Inicialmente se realiza la descarga del repositorio completo del compañero asignado, Miguel Núñez (no clonar, solo descarga para revisar el contenido en local) y se realiza la revisión total de la documentación en el archivo PDF. De esta revisión se extraen, con criterio subjetivo, algunos errores que se presentan a continuación.

Diagrama ER Parque Norte: 

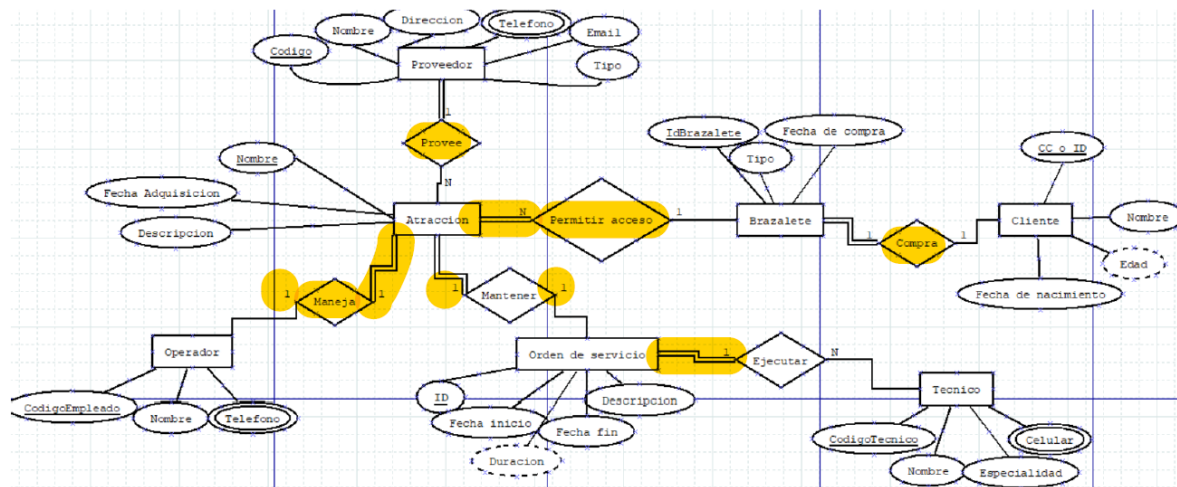


Figura 3 MER con correcciones.

Los nombres de los atributos varían de formato. Se establece formato Camel case.
Se re-nombran algunos atributos para tener atributos atómicos directamente.
Se añade atributo Precio al Brazalete.

No se revisa el MR ya que parte de un MER que no está claro. Se creará un MR nuevo.

Falta especificar la normalización.
Falta rellenar tablas con por lo menos un campo.
La documentación no me explica nada, por lo que las participaciones, cardinalidades y verbos no son claras.

Figura 4 Anotaciones de errores en el archivo PDF.

A partir de los errores encontrados, como lo es una declaración de verbos para las relaciones que no están en infinitivo, participaciones no claras y que al ser un aspecto subjetivo se consideran erróneas; y unas cardinalidades no explicadas que de igual manera pueden ser subjetivas. Luego, se reestructura el MER como en la Figura 5.

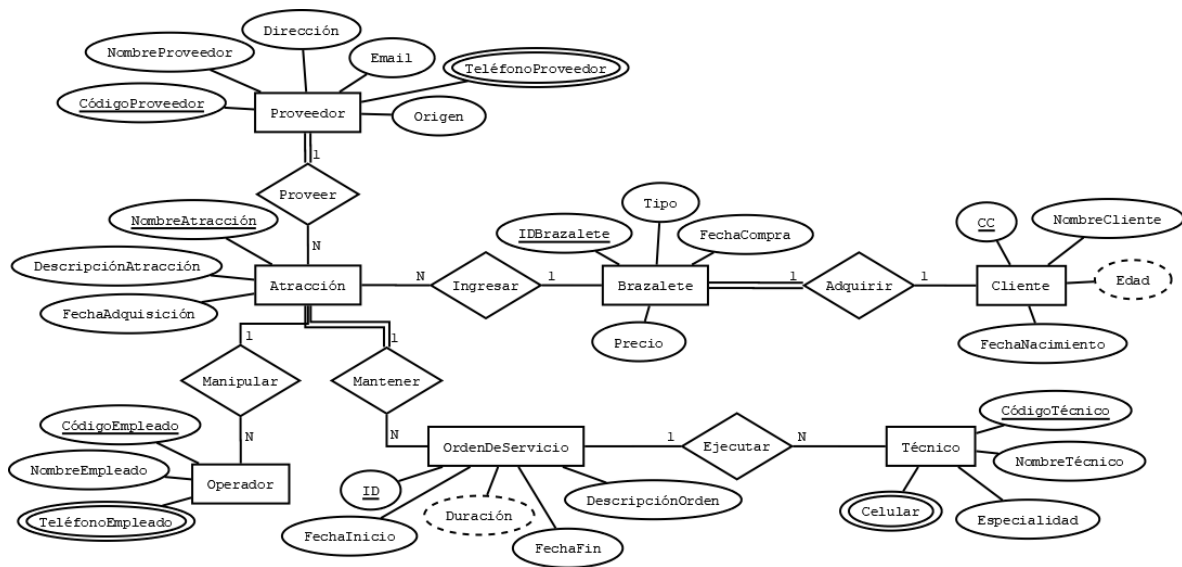


Figura 5 MER corregido.

Se ordena la estructura planteada por el compañero, se reemplazan los verbos usados a verbos en infinitivo y acorde a la directriz inicial. También, se renombran algunos atributos con el formato *Camel case*, y se realiza una atomización de los atributos (todos los atributos “Nombre” ahora se llaman con el identificador de su entidad, es decir, “NombreCliente”). Las participaciones que eran totales y marcadas en la Figura 3, pasan a ser parciales, así: para que un brazalete exista, no es necesario que exista una atracción porque los clientes pueden entrar y no usar una atracción del parque. Para que un técnico exista, no es necesario que se haya generado una orden de servicio, porque dado el caso se presente una novedad, habría que generar la orden y luego registrar el técnico, en lugar de simplemente asignarlo porque ya está en la base de datos. Igualmente, para el caso de la existencia del operador, no se debe esperar que haya una atracción para registrar un operador en la BD.

Para la cardinalidad entre operador y atracción, se observa que la lógica del compañero indica que solo una persona puede manejar una atracción (1:1), lo cual haría que fuera imposible realizar cambios de turno. Así, se modifica a una cardinalidad muchos a uno (1:N) para que varios operadores puedan manipular una sola atracción, y que una atracción solo sea manipulada por un operario a la vez. También, no tiene sentido que una atracción solo pueda ser mantenida a través de una sola orden de servicio (1:1), así que se cambia para que una orden pueda generar un mantenimiento para solo una atracción, pero que una atracción pueda ser mantenida a través de varias órdenes de servicio (1:N).

De esta manera, se procede a realizar la transformación a MR directamente en Workbench, ya que así se evita realizar doble trabajo. El resultado se aprecia en la Figura 6.

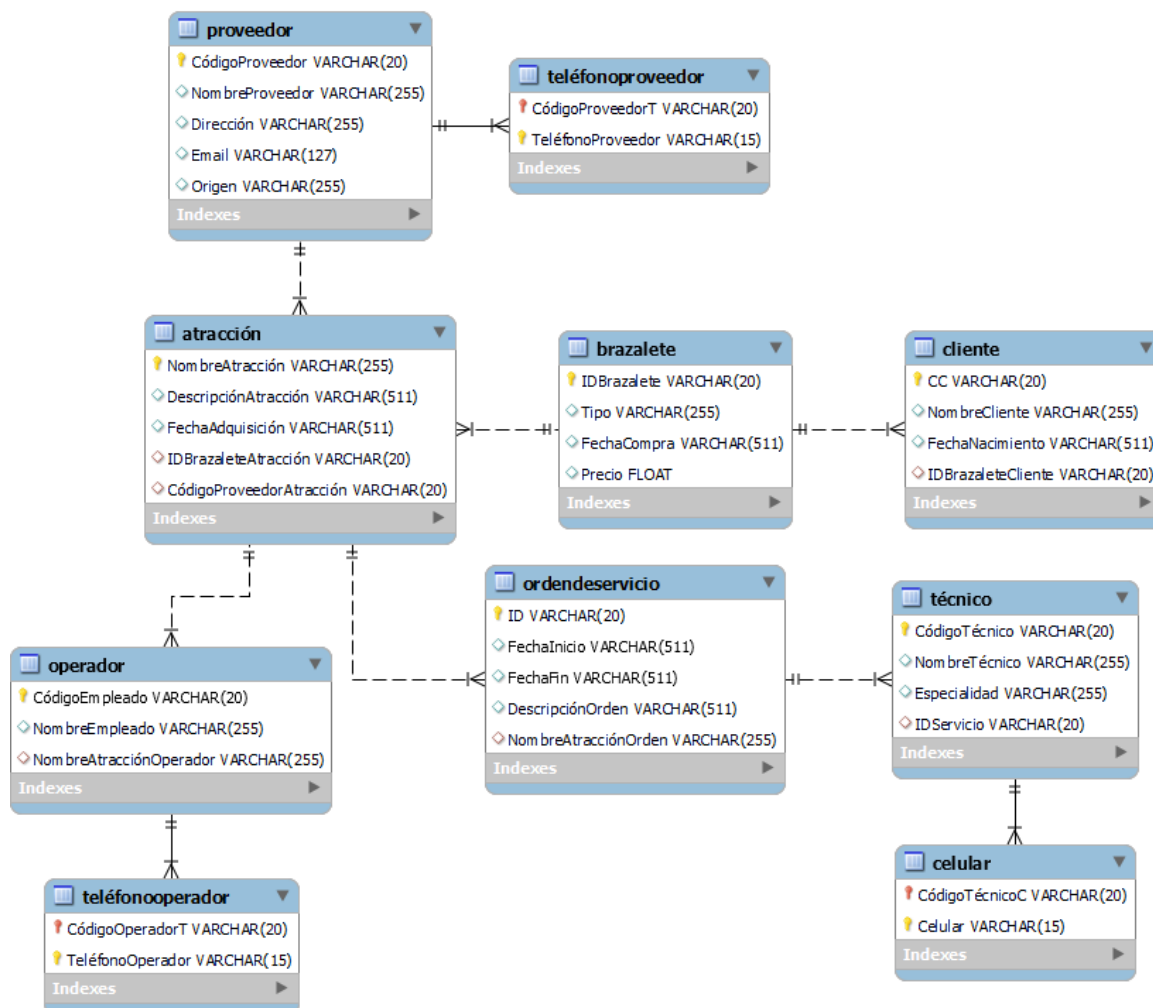


Figura 6 MR Compañero corregido.

Finalmente, se realiza la traducción de este modelo a sentencias SQL, de la siguiente manera.

```

CREATE SCHEMA `segundaActividad`;
CREATE TABLE `segundaActividad`.`Proveedor` (
  CódigoProveedor varchar(20) primary key,
  NombreProveedor varchar(255),
  Dirección varchar(255),
  Email varchar(127),
  Origen varchar(255)
);
CREATE TABLE `segundaActividad`.`TeléfonoProveedor` (
  CódigoProveedorT varchar(20),
  TeléfonoProveedor varchar(15),
  primary key(CódigoProveedorT, TeléfonoProveedor),
  foreign key(CódigoProveedorT) references Proveedor(CódigoProveedor)
);
CREATE TABLE `segundaActividad`.`Brazalete` (
  IDBrazalete varchar(20) primary key,
  Tipo varchar(255),
  FechaCompra varchar(511),
  Precio float(15)
);
CREATE TABLE `segundaActividad`.`Cliente` (
  CC varchar(20) primary key,
  NombreCliente varchar(255),
  FechaNacimiento varchar(511),
  IDBrazaleteCliente varchar(20),
  foreign key(IDBrazaleteCliente) references Brazalete(IDBrazalete)
);
CREATE TABLE `segundaActividad`.`Atracción` (
  NombreAtracción varchar(255) primary key,
  DescripciónAtracción varchar(511),
  FechaAdquisición varchar(511),
  IDBrazaleteAtracción varchar(20),
  foreign key(IDBrazaleteAtracción) references Brazalete(IDBrazalete),
  CódigoProveedorAtracción varchar(20),
  foreign key(CódigoProveedorAtracción) references Proveedor(CódigoProveedor)
);
CREATE TABLE `segundaActividad`.`Operador` (
  CódigoEmpleado varchar(20) primary key,
  NombreEmpleado varchar(255),
  NombreAtracciónOperador varchar(255),
  foreign key(NombreAtracciónOperador) references Atracción(NombreAtracción)
);
CREATE TABLE `segundaActividad`.`TeléfonoOperador` (
  CódigoOperadorT varchar(20),

```

```

    TeléfonoOperador varchar(15),
    primary key(CódigoOperadorT, TeléfonoOperador),
    foreign key(CódigoOperadorT) references Operador(CódigoEmpleado)
);
CREATE TABLE `segundaActividad`.`OrdenDeServicio` (
    ID varchar(20) primary key,
    FechaInicio varchar(511),
    FechaFin varchar(511),
    DescripciónOrden varchar(511),
    NombreAtracciónOrden varchar(255),
    foreign key(NombreAtracciónOrden) references Atracción(NombreAtracción)
);
CREATE TABLE `segundaActividad`.`Técnico` (
    CódigoTécnico varchar(20) primary key,
    NombreTécnico varchar(255),
    Especialidad varchar(255),
    IDServicio varchar(20),
    foreign key(IDServicio) references OrdenDeServicio(ID)
);
CREATE TABLE `segundaActividad`.`Celular` (
    CódigoTécnicoC varchar(20),
    Celular varchar(15),
    primary key(CódigoTécnicoC, Celular),
    foreign key(CódigoTécnicoC) references Técnico(CódigoTécnico)
);

```

Tercera Actividad

TODO