

Documentación correcciones:

Como punto de partida, para la realización de este trabajo tenemos los modelos realizados por el compañero Jonathan Sepúlveda.

El ejercicio propuesto era este:

MR Parque Norte

Se sabe que:

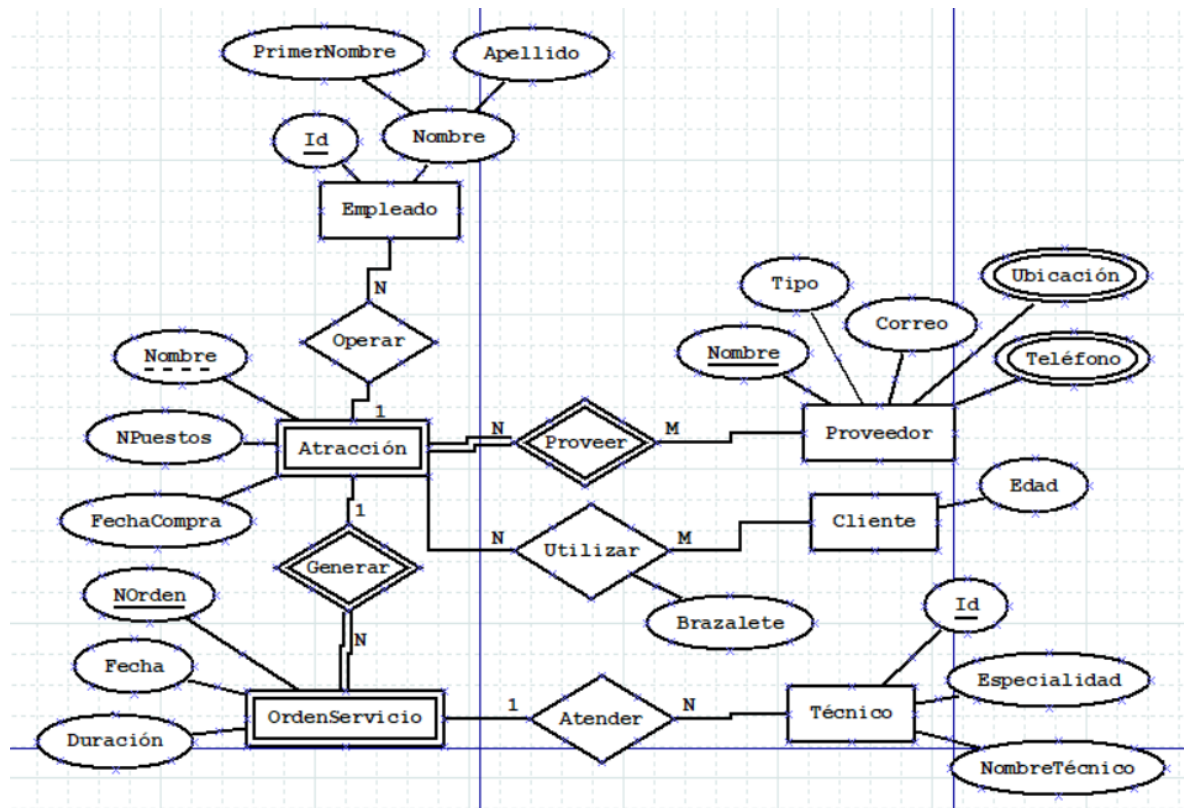
- Las atracciones que están en el parque fueron adquiridas a un proveedor (nacional o internacional) el cual debe tener una información de contacto y son manipuladas por un empleado (operador).
- Las atracciones que hay en el parque pueden necesitar un mantenimiento que será realizado por un técnico (nombre, especialidad y demás información que considere relevante) por medio de una orden de servicio (fecha, duración).
- Para poder ingresar a las atracciones es necesario comprar un brazalete el cual es adquirido por los clientes.

Se sabe que:

- Verificar el modelo entidad relación realizado en la actividad anterior. <https://github.com/training-practice-sofkau/C1-2023-QA-BD-01>
- Transformar el modelo E-R en un M-R.
- Normalizar el modelo.
- Elaborar un documento PDF con la imagen del modelo ER, capturas de pantalla con la transformación del modelo y por lo menos un registro completo de cada una de sus tablas.

Adjunto link de la solución propuesta por el compañero: [Jonathan Andrés Vargas Sepúlveda by Ochocho · Pull Request #2 · training-practice-sofkau/C1-2023-QA-BD-02 \(github.com\)](#)

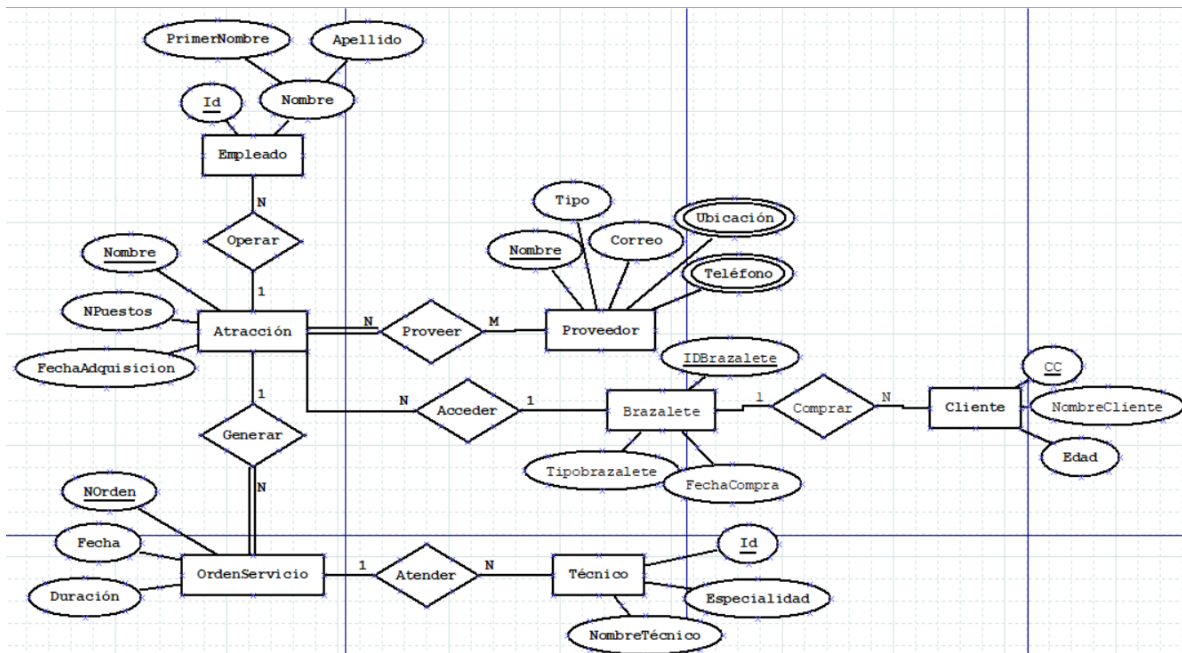
Para empezar, revisamos el diagrama ER de Jonathan:



Este diagrama contaba con dos errores, los cuales son los siguientes:

- La clave primaria de la entidad atracción no estaba debidamente marcada como tal.
- No creo una entidad brazalete que permitiera la entrada a las atracciones

Por ello realice la corrección del diagrama entidad relación, solventando los errores, así:



Como podemos ver se marcó correctamente la llave primaria de atracciones, y se creó la entidad Brazalete.

Brazalete tiene los atributos:

- IDBrazalete (PK)
- Fecha de compra
- Tipo de brazalete

Y esta relacionado de esta manera:

- Un brazalete permite la entrada a una o muchas atracciones, y una atracción puede ser accedida con un brazalete.
- Un cliente puede comprar uno o mas brazaletes, y un brazalete puede ser comprado por un cliente.



Concluyendo el modelo realizamos el proceso de normalización:

Normalización:

N1:

Normas	Estado
Todos los atributos tienen valores atómicos	Cumple
No hay atributos multivaluados	Cumple
No existen registros duplicados	Cumple
Se eliminaron todas las columnas repetidas	Cumple
Definir clave principal	Cumple

N2:

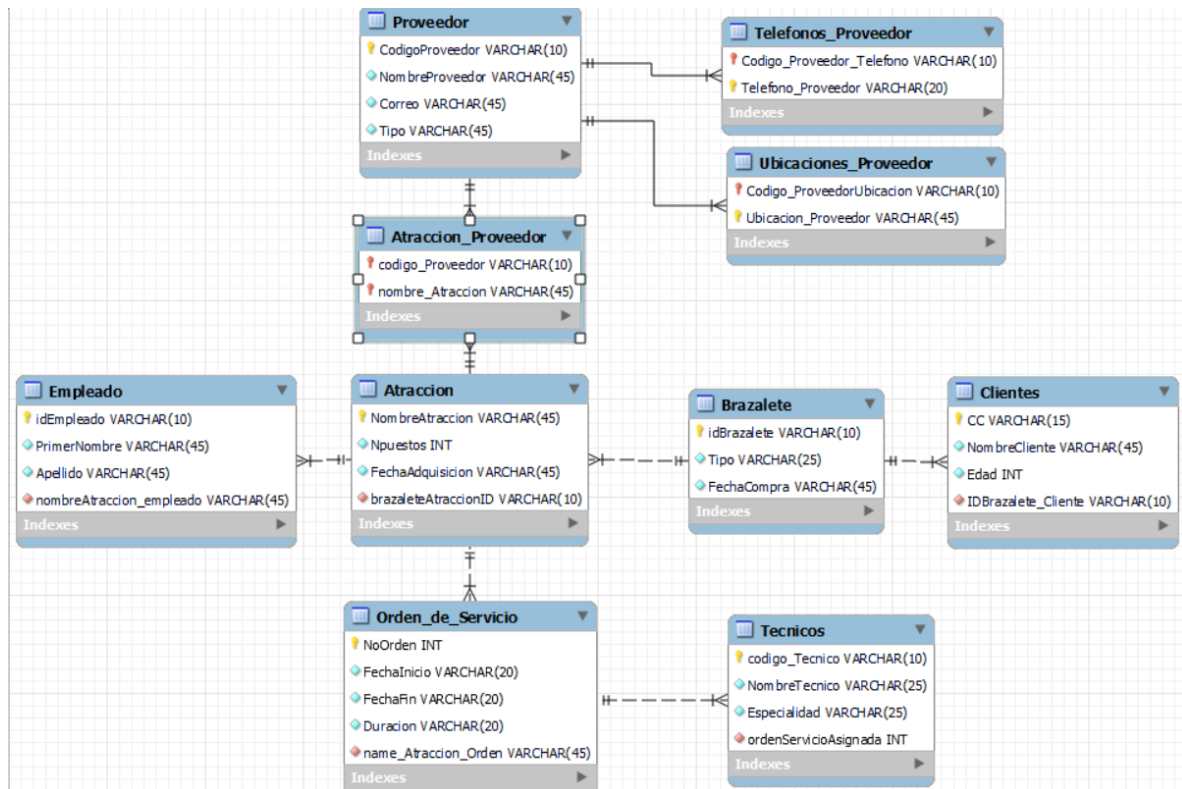
Normas	Estado
Cumple con la norma 1	Cumple
Los valores de las columnas dependen solo de la llave primaria	Cumple
Las tablas tienen una única llave primaria que las identifique	Cumple

N3:

Normas	Estado
Cumple la norma 2	Cumple
Los atributos no incluidos en la clave primaria no dependen transitivamente la clave primaria	Cumple

Para completar la actividad procedimos a crear el modelo con MySQL Workbench, y realizar la creación de la base de datos.

Modelo de base de datos en Workbench:



Para terminar, se creo la base de datos en SQL, adjuntare pantallazos del código SQL usado y ejemplos de los registros:

Creación de la base de datos parquenorte y la tabla Brazalete:

```

CREATE SCHEMA IF NOT EXISTS `parquenorte` DEFAULT CHARACTER SET utf8 ;
USE `parquenorte` ;

CREATE TABLE IF NOT EXISTS `parquenorte`.`Brazalete` (
  `idBrazalete` VARCHAR(10) NOT NULL,
  `Tipo` VARCHAR(25) NOT NULL,
  `FechaCompra` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idBrazalete`));
  
```






Ejemplo registros:

idBrazalete	Tipo	FechaCompra
ADV	Adventure	9/02/2023
EXT	Extremo	9/02/2023

Creación de la tabla atracción:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Atraccion` (  
  `NombreAtraccion` VARCHAR(45) NOT NULL,  
  `Npuestos` INT NOT NULL,  
  `FechaAdquisicion` VARCHAR(45) NOT NULL,  
  `brazaleteAtraccionID` VARCHAR(10) NOT NULL,  
  PRIMARY KEY (`NombreAtraccion`),  
  INDEX `brazaleteAtraccionID_idx` (`brazaleteAtraccionID` ASC) VISIBLE,  
  CONSTRAINT `brazaleteAtraccionID`  
    FOREIGN KEY (`brazaleteAtraccionID`)  
      REFERENCES `parquenorte`.`Brazalete` (`idBrazalete`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

Ejemplo registros atracción:

Result Grid				Filter Rows:		Edit:			
	NombreAtraccion	Npuestos	FechaAdquisicion	brazaleteAtraccionID					
	Carros chocones	30	03/20/2023	EXT					
	Montaña extrema	25	03/20/2023	ADV					

Creación de la tabla Proveedor:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Proveedor` (  
  `CodigoProveedor` VARCHAR(10) NOT NULL,  
  `NombreProveedor` VARCHAR(45) NOT NULL,  
  `Correo` VARCHAR(45) NOT NULL,  
  `Tipo` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`CodigoProveedor`));
```



Ejemplo registros proveedor:

Result Grid	Filter Rows:	Edit:	Export
CodigoProveedor	NombreProveedor	Correo	Tipo
1	Heros	atencion@heros.com	Nacional
2	Matel	atencion@matel.com	Internacional
* NULL	NULL	NULL	NULL

Creación de la tabla atracción-proveedor:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Atraccion_Proveedor` (  
  `codigo_Proveedor` VARCHAR(10) NOT NULL,  
  `nombre_Atraccion` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`codigo_Proveedor`, `nombre_Atraccion`),  
  INDEX `nombre_Atraccion_idx` (`nombre_Atraccion` ASC) VISIBLE,  
  CONSTRAINT `codigo_Proveedor`  
    FOREIGN KEY (`codigo_Proveedor`)  
      REFERENCES `parquenorte`.`Proveedor` (`CodigoProveedor`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION,  
  CONSTRAINT `nombre_Atraccion`  
    FOREIGN KEY (`nombre_Atraccion`)  
      REFERENCES `parquenorte`.`Atraccion` (`NombreAtraccion`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION);
```

Ejemplo registros atracción-proveedor:

Result Grid   Filter Rows: <input type="text"/>		
	codigo_Proveedor	nombre_Atraccion
▶	2	Carros chocones
	1	Montaña extrema

Creación de la tabla empleado:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Empleado` (  
  `idEmpleado` VARCHAR(10) NOT NULL,  
  `PrimerNombre` VARCHAR(45) NOT NULL,  
  `Apellido` VARCHAR(45) NOT NULL,  
  `nombreAtraccion_empleado` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`idEmpleado`),  
  INDEX `Atraccion_Empleado_idx` (`nombreAtraccion_empleado` ASC) VISIBLE,  
  CONSTRAINT `nombreAtraccion_empleado`  
    FOREIGN KEY (`nombreAtraccion_empleado`)  
      REFERENCES `parquenorte`.`Atraccion` (`NombreAtraccion`)  
      ON DELETE NO ACTION  
      ON UPDATE NO ACTION);
```


Ejemplo registros empleado:

Result Grid		Filter Rows:		Edit:
	idEmpleado	PrimerNombre	Apellido	nombreAtraccion_empleado
▶	12N	David	Mantilla	Carros chocones
	13N	Carlos	Cuesta	Montaña extrema

Creación de la tabla clientes:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Clientes` (  
  `CC` VARCHAR(15) NOT NULL,  
  `NombreCliente` VARCHAR(45) NOT NULL,  
  `Edad` INT NOT NULL,  
  `IDBrazalete_Cliente` VARCHAR(10) NOT NULL,  
  PRIMARY KEY (`CC`),  
  INDEX `IDBrazalete_Cliente_idx` (`IDBrazalete_Cliente` ASC) VISIBLE,  
  CONSTRAINT `IDBrazalete_Cliente`  
    FOREIGN KEY (`IDBrazalete_Cliente`)  
    REFERENCES `parquenorte`.`Brazalete` (`idBrazalete`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

Ejemplo registro tabla clientes:

Result Grid

Filter Rows:

Edit:

	CC	NombreCliente	Edad	IDBrazalete_Cliente
▶	123456789	Claudia Sanches	20	EXT
	987654312	Andres Gomez	26	ADV
*	NULL	NULL	NULL	NULL

Creación de la tabla orden de servicio:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Orden_de_Servicio` (  
  `NoOrden` INT NOT NULL,  
  `FechaInicio` VARCHAR(20) NOT NULL,  
  `FechaFin` VARCHAR(20) NOT NULL,  
  `Duracion` VARCHAR(20) NOT NULL,  
  `name_Atraccion_Orden` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`NoOrden`),  
  INDEX `name_Atraccion_Orden_idx` (`name_Atraccion_Orden` ASC) VISIBLE,  
  CONSTRAINT `name_Atraccion_Orden`  
    FOREIGN KEY (`name_Atraccion_Orden`)  
    REFERENCES `parquenorte`.`Atraccion` (`NombreAtraccion`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

Ejemplo registros tabla orden de servicio:

	NoOrden	FechaInicio	FechaFin	Duracion	name_Atraccion_Orden
▶	36	10/02/2023	11/02/2023	1 Dia	Montaña extrema
	37	11/02/2023	12/02/2023	1 Dia	Carros chocones
*	NULL	NULL	NULL	NULL	NULL

Creación de la tabla técnicos:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Tecnicos` (  
  `codigo_Tecnico` VARCHAR(10) NOT NULL,  
  `NombreTecnico` VARCHAR(25) NOT NULL,  
  `Especialidad` VARCHAR(25) NOT NULL,  
  `ordenServicioAsignada` INT NOT NULL,  
  PRIMARY KEY (`codigo_Tecnico`),  
  INDEX `ordenServicioAsignada_idx` (`ordenServicioAsignada` ASC) VISIBLE,  
  CONSTRAINT `ordenServicioAsignada`  
    FOREIGN KEY (`ordenServicioAsignada`)  
    REFERENCES `parquenorte`.`Orden_de_Servicio` (`NoOrden`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION);
```

Ejemplo registros técnicos:

	codigo_Tecnico	NombreTecnico	Especialidad	ordenServicioAsignada
▶	105	Raul hernandez	Mecanica	36
	106	Camila Arevalo	Fisica	37

Creación de la tabla teléfonos-proveedor:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Telefonos_Proveedor` (  
  `Codigo_Proveedor_Telefono` VARCHAR(10) NOT NULL,  
  `Telefono_Proveedor` VARCHAR(20) NOT NULL,  
  PRIMARY KEY (`Codigo_Proveedor_Telefono`, `Telefono_Proveedor`)  
  CONSTRAINT `Codigo_Proveedor_Telefono`  
    FOREIGN KEY (`Codigo_Proveedor_Telefono`)  
    REFERENCES `parquenorte`.`Proveedor` (`CodigoProveedor`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION);
```

Ejemplo registros teléfonos-proveedor:

	Codigo_Proveedor_Telefono	Telefono_Proveedor
▶	1	3205606633
	2	5656633
*	NULL	NULL

Creación de la tabla ubicaciones-proveedor:

```
CREATE TABLE IF NOT EXISTS `parquenorte`.`Ubicaciones_Proveedor` (  
  `Codigo_ProveedorUbicacion` VARCHAR(10) NOT NULL,  
  `Ubicacion_Proveedor` VARCHAR(45) NOT NULL,  
  PRIMARY KEY (`Codigo_ProveedorUbicacion`, `Ubicacion_Proveedor`),  
  CONSTRAINT `Codigo_ProveedorUbicacion`  
    FOREIGN KEY (`Codigo_ProveedorUbicacion`)  
    REFERENCES `parquenorte`.`Proveedor` (`CodigoProveedor`)  
  ON DELETE NO ACTION  
  ON UPDATE NO ACTION);
```

Ejemplos registros ubicaciones-proveedor:

	Codigo_ProveedorUbicacion	Ubicacion_Proveedor
▶	1	Medellin
	2	Houston
*	NULL	NULL

Conclusiones:

- Logre familiarizarme más con los procesos de modelado de las bases de datos, y logre aprender de la lógica de mi compañero Jonathan
- Comprendí el funcionamiento de la aplicación MySQL Workbench para realizar el modelado de bases de datos.