



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA



Departamento de Computación,
Facultad de Ciencias Exactas y Naturales,
Universidad de Buenos Aires

Trabajo Práctico 1

Bases de Datos

Primer Cuatrimestre de 2016

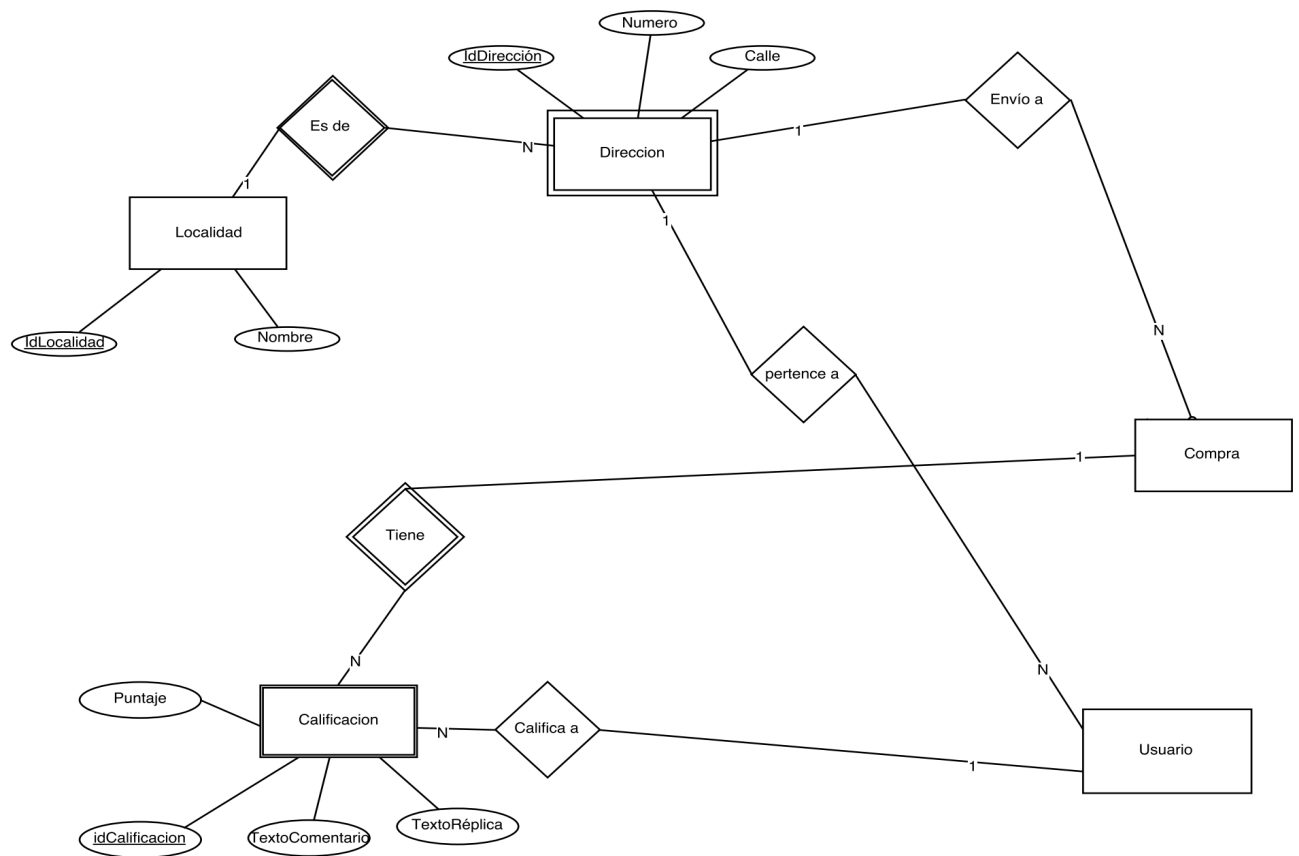
Apellido y Nombre	LU	E-mail
Russo, Christian	679/10	christian.russo8@gmail.com
Russo, Christian	679/10	christian.russo8@gmail.com
Russo, Christian	679/10	christian.russo8@gmail.com
Russo, Christian	679/10	christian.russo8@gmail.com

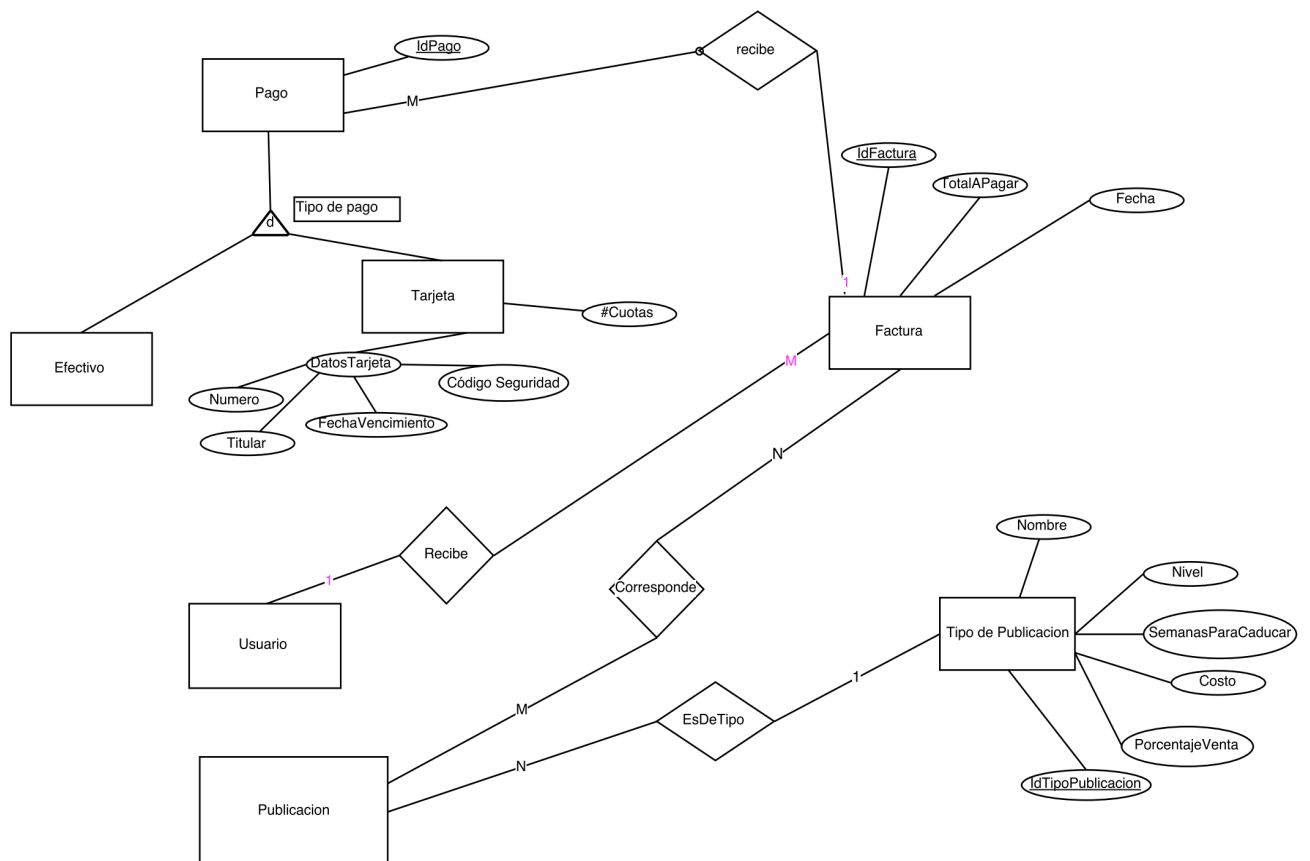
1. Introducción

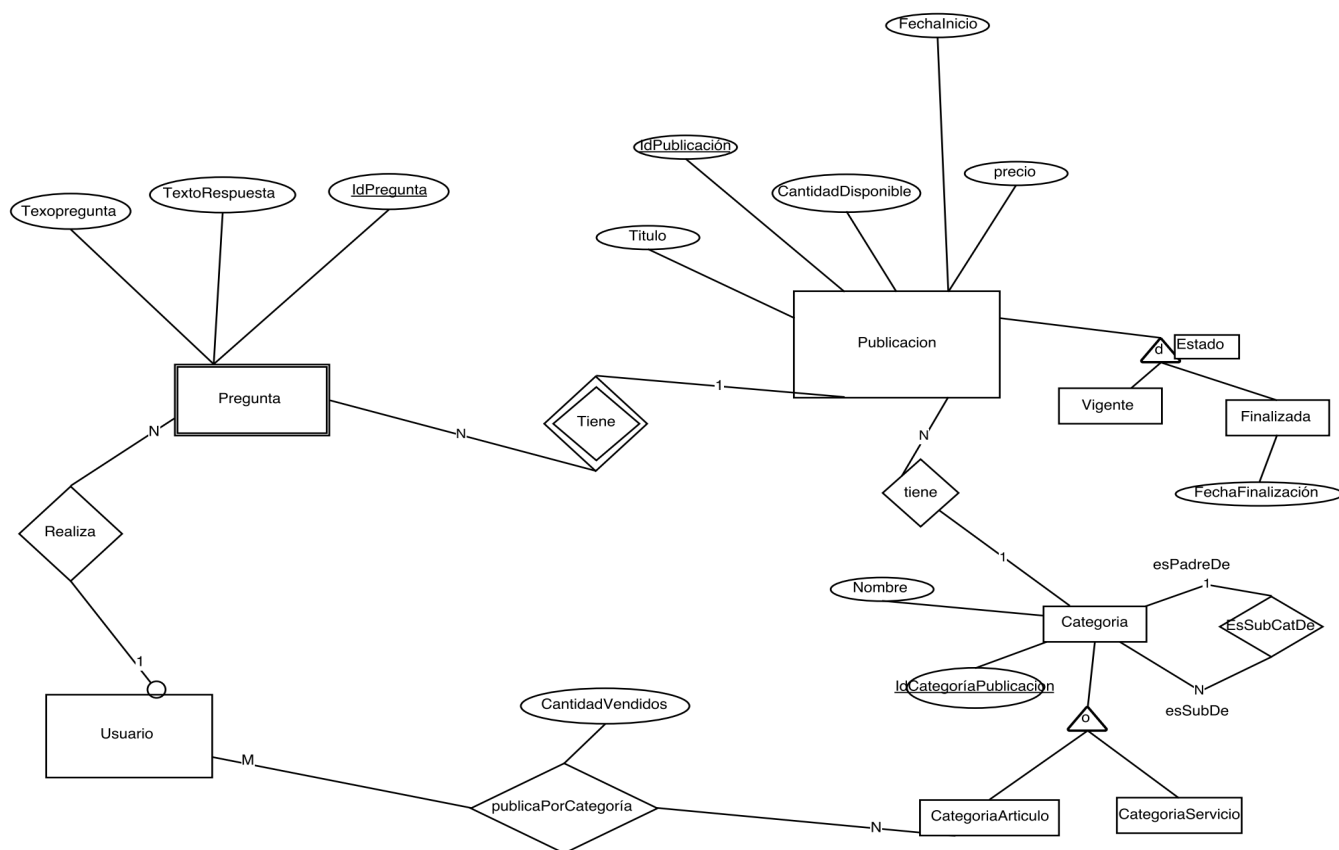
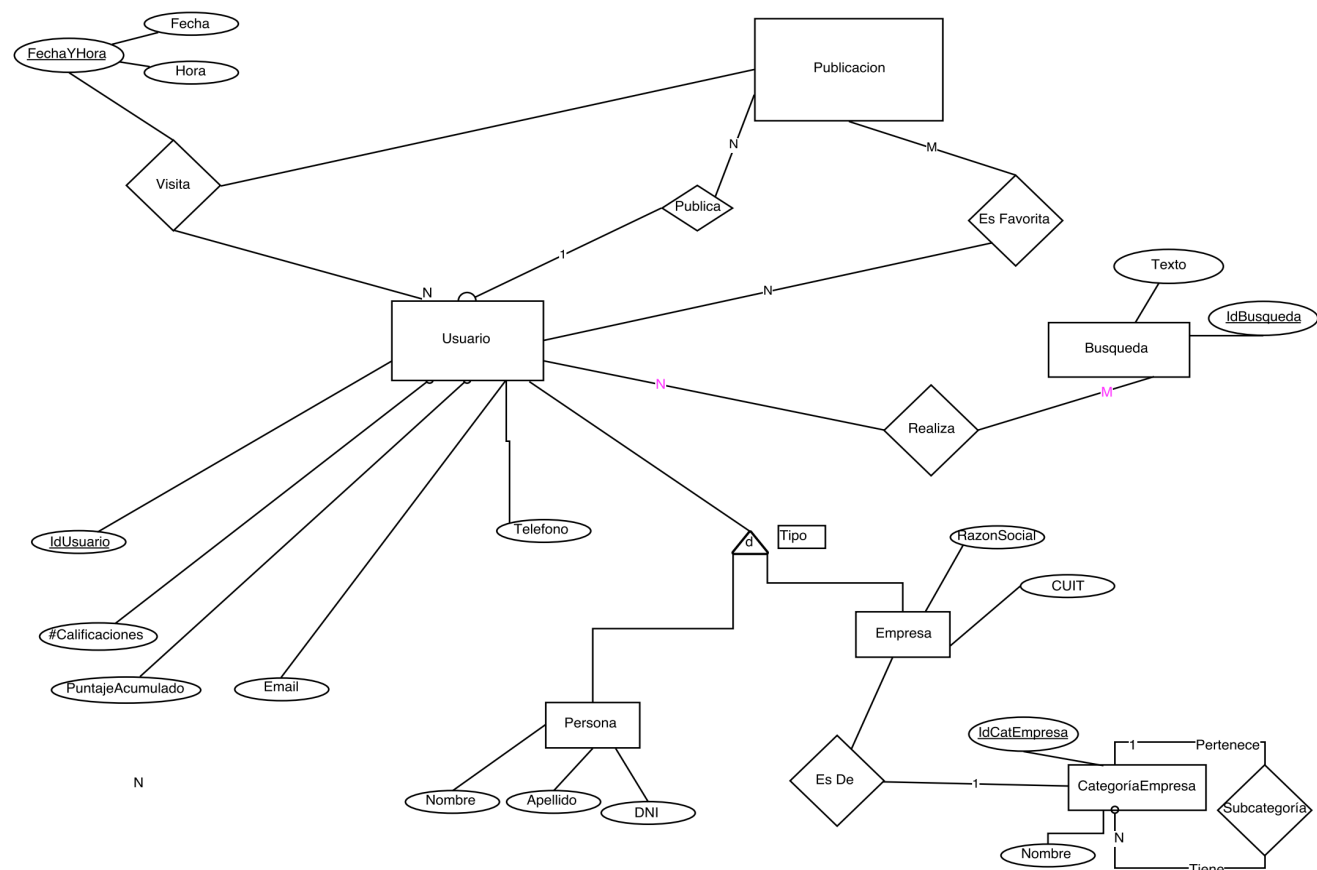
Presentaremos una solución para el problema de un **Mercado Virtual** tomando como guía el **Khan El-Khalili** ubicado en El Cairo, Egipto. El problema en cuestión contempla una serie de restricciones sobre como se realizan la compra y venta de productos por internet. Cada publicación puede tener distintos tipos y ser de distintas formas, haciendo que esto impacte en la facturación del usuario que publica. Por otro lado se cuenta con un sistema de comentarios y calificaciones.

Utilizaremos las herramientas vistas en la materia, el modelado basado en el Diagrama de Entidad Relación, su MR resultante y la base de datos final que presentaremos en MySQL.

2. Diagrama de Entidad Relación







2.1. Restricciones

1. No puede haber dos DNI iguales.
2. La publicación del tipo Libre! tiene costo 0.
3. Para cada compraventa existen a lo sumo 2 calificaciones, cada una correspondiente al comprador y vendedor.
4. El nivel de la publicación es distinto para cada tipo de publicación.
5. El usuario que compra una publicación no puede ser el usuario que publica.
6. Si un usuario realiza una respuesta a un comentario, entonces dicho comentario es de una calificación y la calificación fue hecha por un usuario que hizo dicha compra
7. Las publicaciones del tipo RubíDeOriente aparecen primeras en las búsquedas. Luego aparecen en orden de mayor a menor costo de comisión y por último las de la categoría Libre!.
8. El costo por mes de RubíDeOriente es Fijo y se pueden hacer 3 publicaciones por mes de este tipo usuario.
9. El costo de las de Oro cobran más porcentaje de la venta como comisión que las de Plata
10. El costo de las de Plata cobran más porcentaje de la venta como comisión que las de Bronce
11. El monto de una oferta en una subasta debe ser superior en al menos 1 peso a la oferta actual, e inferior al doble de la oferta actual
12. Una calificación tiene completado el atributo textoRéplica, entonces tiene completado el atributo textoComentario
13. El atributo puntaje de calificación está entre [1, 10].
14. El atributo nombre de la entidad Tipo sólo puede ser uno de los siguientes: RubíDeOriente, Oro, Plata, Bronce o Libre!
15. No se puede realizar una pregunta a una publicación que está finalizada.
16. El atributo cantidad de la entidad compra siempre es menor o igual al atributo cantidadDisponible de la publicacion.
17. Si el atributo cantidadDisponible de la entidad Publicación vale 0, entonces la publicación está finalizada.
18. Dada una oferta, el monto de dicha oferta debe ser mayor en al menos 1 peso a todas las ofertas anteriores (en fecha), y al precioBase de la subasta. Además dicha oferta no puede superar el doble del precio base.
19. El precio de una publicación que es de tipo subasta es igual a: el precioBase de la subasta si no hay una oferta realizada.
20. Un usuario no puede ofertar en una publicación finalizada.
21. Si un usuario compró una publicación de tipo subasta, entonces dicho usuario tiene que tener la oferta más alta en dicha publicación.
22. El TotalAPagar de la Factura es lo que adeuda el Usuario al sistema en concepto de abonos a RubíDeOriente y/o comisión por las ventas de sus programador.
23. Un pago o se relaciona con una Publicación, o se relaciona con una Compra. Nunca con los dos.
24. El atributo CantidadVendidos de la relación publicaPorCategoria es igual a todas las ventas que realizó ese usuario en esa categoría.
25. El tipo dentro de la tabla Artículo puede ser o venta o subaste

3. Modelo Relacional

Persona(idUsuario, nombre, apellido, DNI)

PK = CK = {idUsuario}

Empresa(idUsuario, RazonSocial, CUIT, idCatEmpresa)

PK = CK = {idUsuario}

FK = {idUsuario, idCatEmpresa}

CategoriaEmpresa(idCaEmpresa, Nombre, IdCategoriaPadre)

PK = CK = {idCaEmpresa}

FK = {IdCategoriaPadre}

Articulo(IdPublicacion, tipo)

PK = CK = {IdPublicacion}

FK = {IdPublicacion}

Subasta(IdPublicacion, PrecioBase)

PK = CK = {IdPublicacion}

FK = {IdPublicacion}

Oferta(IdPublicacion, idUsuario, FechaYHora, Monto)

PK = CK = {IdPublicacion, idUsuario, FechaYHora}

FK = {IdPublicacion, idUsuario}

Venta(IdPublicacion,)

PK = CK = {IdPublicacion}

FK = {IdPublicacion}

Servicio(IdPublicacion, Comision)

PK = CK = {IdPublicacion}

FK = {IdPublicacion}

Vigente(IdPublicacion)

PK = CK = {IdPublicacion}

FK = {IdPublicacion}

Finalizada(IdPublicacion, FechaFinalizacion)

PK = CK = {IdPublicacion}

FK = {IdPublicacion}

Categoria(IdCategoriaPublicacion, Nombre, IdCategoriaPadre)

PK = CK = {IdCategoriaPublicacion}

FK = {IdCategoriaPadre}

CategoriaArticulo(IdCategoriaPublicacion)

PK = CK = {IdCategoriaPublicacion}

FK = {IdCategoriaPublicacion}

publicaPorCategoria(IdCategoriaPublicacion, IdUsuario, cantidadVendidos)

PK = CK = {IdCategoriaPublicacion, IdUsuario}

FK = {IdCategoriaPublicacion, IdUsuario}

CategoriaServicio(idCategoriaPublicacion)

PK = CK = {idCategoriaPublicacion}

FK = {idCategoriaPublicacion}

Usuario(idUsuario, tipo, cantCalificaciones, puntajeAcumulado, Email, Telefono, idDireccion, idLocalidad)

PK = CK = {idUsuario}

FK = {idDireccion, idLocalidad}

Localidad(idLocalidad, nombre)

PK = CK = {IdPublicacion}

Direccion(idDireccion, idLocalidad, Numero, Calle)

PK = CK = {(idDireccion, idLocalidad)}

FK = {idLocalidad}

Calificacion(idCalificacion, idCompra, idUsuario, puntaje, TextoComentario, TextoReplica)

PK = CK = {(idCalificacion, idCompra)}

FK = {idCompra, idUsuario}

TipoDePublicacion(idTipoPublicacion, Nombre, Nivel, SemanasParaCaducar, Costo, PorcentajeVenta)

PK = CK = {(idCalificacion, idCompra)}

Factura(idFactura, TotalAPagar, Fecha, idUsuario,)

PK = CK = {idFactura}

FK = {idUsuario}

Corresponde(idFactura, idPublicacion)

PK = CK = {(idFactura, idPublicacion)}

FK = {idFactura, idPublicacion}

Pago(idPago, idFactura, TipoDePago)

PK = CK = {idPago}

FK = {idFactura}

Efectivo(idPago,)

PK = CK = {idPago}

FK = {idPago}

Tarjeta(idPago, NumCuotas, Numero, Titular, FechaVencimiento, CodSeguridad)

PK = CK = {idPago}

FK = {idPago}

Pregunta(idPregunta, idPublicacion, TextoPregunta, TextoRespuesta, idUsuario)

PK = CK = {idPregunta, idPublicacion}

FK = {idPublicacion, idUsuario}

Compra(idCompra, idPublicacion, idPago, idDireccion, Fecha, Cantidad idUsuario)

PK = CK = {idCompra}

FK = {idUsuario, idPublicacion, idPago, idDireccion}

Busqueda(idBusqueda, Texto)

PK = CK = {idBusqueda}

Realiza(idBusqueda, idUsuario)

PK = CK = {(idBusqueda, idUsuario)}

FK = {idBusqueda, idUsuario}

EsFavorita(idPublicacion, idUsuario)

PK = CK = {(idPublicacion, idUsuario)}

FK = {idPublicacion, idUsuario}

Visita(idPublicacion, Fecha, Hora, idUsuario)

PK = CK = {(idPublicacion, idUsuario, Fecha, Hora)}

FK = {idPublicacion, idUsuario}

Publicacion(idPublicacion, idCategoria, idUsuario, IdTipoDePublicacion, Estado, FechaInicio, Titulo, CantidadDisponible, Precio)

PK = CK = {idPublicacion}

FK = {idCategoria, idUsuario, idTipoDePublicacion}

4. Suposiciones

Detalle de los supuestos asumidos para la resolución del problema.

5. Diseño Físico

5.1. Creación de tablas

5.1.1. Visita

```
CREATE TABLE visita(
  idUsuario int NOT NULL,
    idPublicacion int NOT NULL,
    fecha date not null,
    hora time not null,
  PRIMARY KEY (idUsuario, idPublicacion, fecha, hora),
  FOREIGN KEY (idUsuario) REFERENCES usuario(idUsuario),
  FOREIGN KEY (idPublicacion) REFERENCES publicacion(idPublicacion)
)
```

5.1.2. Vigente

```
CREATE TABLE Vigente(
  idPublicacion int NOT NULL,
  PRIMARY KEY (idPublicacion),
    FOREIGN KEY (idPublicacion) REFERENCES Publicacion(idPublicacion)
);
```

5.1.3. Venta

```
CREATE TABLE Venta(
  idPublicacion int NOT NULL,
  PRIMARY KEY (idPublicacion),
    FOREIGN KEY (idPublicacion) REFERENCES Publicacion(idPublicacion)
);
```

5.1.4. Usuario

```
CREATE TABLE usuario(
  idUsuario INT NOT NULL AUTO_INCREMENT,
  tipo varchar(255) NOT NULL,
    cantCalificaciones int NOT NULL,
    puntajeAcumulado int NOT NULL,
    email VARCHAR(255) NOT NULL,
```



```
telefono VARCHAR(255) NOT NULL,  
idDireccion INT NOT NULL,  
idLocalidad INT NOT NULL,  
PRIMARY KEY(idUsuario),  
FOREIGN KEY (idDireccion) REFERENCES direccion(idDireccion),  
FOREIGN KEY (idLocalidad) REFERENCES localidad(idLocalidad)  
)
```

5.1.5. Tipo de publicacion

```
CREATE TABLE Tipo_de_Publicacion(  
idTipoPublicacion int NOT NULL AUTO_INCREMENT,  
nombre varchar(255) NOT NULL,  
nivel int Not null,  
semanasParaCaducar int,  
costo int not null,  
porcentajeVenta int not null,  
PRIMARY KEY (idTipoPublicacion)  
);
```

5.1.6. Tarjeta

```
CREATE TABLE tarjeta(  
idPago int NOT NULL,  
numCuotas int not null,  
numero int(16) not null,  
titular varchar(255) not null,  
fechaVencimiento date not null,  
codSeguridad int(3) not null,  
PRIMARY KEY (idPago),  
FOREIGN KEY (idPago) REFERENCES Pago(idPago)  
);
```

5.1.7. Subasta

```
CREATE TABLE Subasta(  
idPublicacion int NOT NULL,  
precioBase int NOT NULL,  
PRIMARY KEY (idPublicacion),  
FOREIGN KEY (idPublicacion) REFERENCES Publicacion(idPublicacion)  
);
```

5.1.8. Servicio

```
CREATE TABLE servicio(  
idPublicacion int Not null,  
comision int not null,  
PRIMARY KEY (idPublicacion),  
FOREIGN KEY (idPublicacion) REFERENCES publicacion(idPublicacion)  
)
```

5.1.9. Realiza

```
CREATE TABLE realiza(  

```

```
idBusqueda int NOT NULL AUTO_INCREMENT,  
idUsuario int NOT NULL,  
PRIMARY KEY (idBusqueda, idUsuario),  
    FOREIGN KEY (idBusqueda) REFERENCES busqueda(idBusqueda),  
    FOREIGN KEY (idUsuario) REFERENCES usuario(idUsuario)  
);
```

5.1.10. Publicacion

```
CREATE TABLE publicacion(  
idPublicacion INT NOT NULL AUTO_INCREMENT,  
idCategoria int NOT NULL,  
    idUsuario int not null,  
    idTipoDePublicacion int not null,  
    estado varchar(255) not null,  
    fechaInicio date not null,  
    titulo varchar(255) not null,  
    cantidadDisponible int not null,  
    precio int not null,  
PRIMARY KEY(idPublicacion),  
FOREIGN KEY (idCategoria) REFERENCES categoria(idCategoriaPublicacion),  
FOREIGN KEY (idUsuario) REFERENCES usuario(idUsuario),  
FOREIGN KEY (idTipoDePublicacion) REFERENCES tipo_de_publicacion(idTipoPublicacion)  
)
```

5.1.11. Publica por categoria

```
CREATE TABLE publica_por_categoria(  
idCategoriaPublicacion int NOT NULL,  
cantidadVendidos int not null,  
idUsuario int not null,  
    PRIMARY KEY (idCategoriaPublicacion, idUsuario),  
    FOREIGN KEY (idCategoriaPublicacion) REFERENCES categoria(idCategoriaPublicacion),  
    FOREIGN KEY (idUsuario) REFERENCES usuario(idUsuario)  
);
```

5.1.12. Pregunta

```
CREATE TABLE pregunta(  
idPregunta int Not null AUTO_INCREMENT,  
    idPublicacion int not null,  
    texto_pregunta text(1000),  
    texto_respuesta text(1000),  
idUsuario int not null,  
    PRIMARY KEY (idPregunta, idPublicacion),  
    FOREIGN KEY (idPublicacion) REFERENCES publicacion(idpublicacion),  
    FOREIGN KEY (idUsuario) REFERENCES usuario(idusuario)  
)
```

5.1.13. Persona

```
CREATE TABLE Persona(  
idUsuario int NOT NULL,  
nombre varchar(255) NOT NULL,  
apellido varchar(255) NOT NULL,
```

```
DNI varchar(255) NOT NULL,  
    PRIMARY KEY (idUserario),  
    FOREIGN KEY (idUserario) REFERENCES Usuario(idUsuario)  
);
```

5.1.14. Pago

```
CREATE TABLE pago(  
    idPago int not null AUTO_INCREMENT,  
    idFactura int,  
        tipoDePago varchar(255) not null, # check constraint (es efectivo o tarjeta)  
    PRIMARY KEY (idPago),  
    FOREIGN KEY (idFactura) REFERENCES factura(idFactura)  
)
```

5.1.15. Oferta

```
CREATE TABLE Oferta(  
    idPublicacion int NOT NULL,  
    idUsuario int NOT NULL,  
    fecha date,  
        hora TIME NOT NULL,  
    monto int NOT NULL,  
        PRIMARY KEY (idPublicacion,idUsuario,fecha, hora),  
        FOREIGN KEY (idPublicacion) REFERENCES Publicacion(idPublicacion),  
    FOREIGN KEY (idUserario) REFERENCES Usuario(idUsuario)  
);
```

5.1.16. Localidad

```
CREATE TABLE localidad(  
    idLocalidad INT NOT NULL AUTO_INCREMENT,  
        nombre varchar(255) NOT NULL,  
    PRIMARY KEY(idLocalidad)  
)
```

5.1.17. Finalizada

```
CREATE TABLE Finalizada(  
    idPublicacion int NOT NULL,  
    fechaFinalizacion DATE NOT NULL,  
    PRIMARY KEY (idPublicacion),  
        FOREIGN KEY (idPublicacion) REFERENCES Publicacion(idPublicacion)  
);
```

5.1.18. Factura

```
CREATE TABLE factura(  
    idFactura INT NOT NULL AUTO_INCREMENT,  
    totalAPagar int NOT NULL,  
        fecha date NOT NULL,
```

```
    idUsuario int not null,  
    PRIMARY KEY(idFactura),  
    FOREIGN KEY (idUsuario) REFERENCES usuario(idUsuario)  
)
```

5.1.19. es favorita

```
CREATE TABLE es_favorita(  
    idUsuario int NOT NULL,  
    idPublicacion int NOT NULL,  
    PRIMARY KEY (idUsuario, idPublicacion),  
    FOREIGN KEY (idUsuario) REFERENCES usuario(idUsuario),  
    FOREIGN KEY (idPublicacion) REFERENCES publicacion(idPublicacion)  
)
```

5.1.20. Empresa

```
CREATE TABLE empresa(  
    idUsuario int NOT NULL,  
    RazonSocial varchar(255) not null,  
    cuit int not null,  
    idCatEmpresa int not null,  
    PRIMARY KEY (idUsuario),  
    FOREIGN KEY (idUsuario) REFERENCES usuario(idUsuario),  
    FOREIGN KEY (idCatEmpresa) REFERENCES categoria_empresa(idCatEmpresa)  
)
```

5.1.21. Efectivo

```
CREATE TABLE efectivo(  
    idPago int NOT NULL,  
    PRIMARY KEY (idPago),  
    FOREIGN KEY (idPago) REFERENCES pago(idPago)  
);
```

5.1.22. Direccion

```
CREATE TABLE direccion(  
    idDireccion INT NOT NULL AUTO_INCREMENT,  
    idLocalidad INT NOT NULL,  
    numero int not null,  
    calle varchar(255) not null,  
    PRIMARY KEY(idDireccion, idLocalidad),  
    FOREIGN KEY (idLocalidad) REFERENCES localidad(idLocalidad)  
)
```

5.1.23. Corresponde

```
CREATE TABLE corresponde(  
    idFactura int not null,  
    idPublicacion int not null,  
    PRIMARY KEY (idFactura, idPublicacion),  
    FOREIGN KEY (idFactura) REFERENCES factura(idFactura),  
    FOREIGN KEY (idPublicacion) REFERENCES publicacion(idPublicacion)  
)
```

5.1.24. Compra

```
CREATE TABLE compra(  
  idCompra int not null AUTO_INCREMENT,  
    idUsuario int not null,  
    idPublicacion int not null,  
    idPago int not null,  
    idDireccion int,  
    fecha date not null,  
    cantidad int not null,  
  PRIMARY KEY (idCompra),  
  FOREIGN KEY (idUsuario) REFERENCES usuario(idusuario),  
  FOREIGN KEY (idPublicacion) REFERENCES publicacion(idPublicacion),  
  FOREIGN KEY (idPago) REFERENCES pago(idPago),  
  FOREIGN KEY (idDireccion) REFERENCES direccion(idDireccion)  
)
```

5.1.25. Categoria

```
CREATE TABLE categoria(  
  idCategoriaPublicacion INT NOT NULL AUTO_INCREMENT,  
  nombre varchar(255) NOT NULL,  
    idCategoriaPadre int,  
  PRIMARY KEY(idCategoriaPublicacion),  
  FOREIGN KEY (idCategoriaPadre) REFERENCES categoria(idCategoriaPublicacion)  
)
```

5.1.26. Categoria Servicio

```
CREATE TABLE Categoria_Servicio(  
  idCategoriaPublicacion int NOT NULL,  
    PRIMARY KEY (idCategoriaPublicacion),  
    FOREIGN KEY (idCategoriaPublicacion) REFERENCES categoria(idCategoriaPublicacion)  
)
```

5.1.27. Categoria Empresa

```
CREATE TABLE categoria_empresa(  
  idCatEmpresa INT NOT NULL AUTO_INCREMENT,  
  nombre varchar(255) NOT NULL,  
    idCategoriaPadre int,  
  PRIMARY KEY(idCatEmpresa),  
  FOREIGN KEY (idCategoriaPadre) REFERENCES categoria_empresa(idCatEmpresa)  
)
```

5.1.28. Categoria Articulo

```
CREATE TABLE Categoria_Articulo(  
  idCategoriaPublicacion int NOT NULL,  
    PRIMARY KEY (idCategoriaPublicacion),  
    FOREIGN KEY (idCategoriaPublicacion) REFERENCES categoria(idCategoriaPublicacion)  
)
```

5.1.29. Calificacion

```
CREATE TABLE calificacion(  
  idCalificacion int Not null AUTO_INCREMENT,  
    idCompra int not null,  
    idUsuario int not null,  
    puntaje int not null,  
  textoComentario text(1000),  
    textoReplica text(1000),  
  
  PRIMARY KEY (idCalificacion, idCompra),  
  FOREIGN KEY (idCompra) REFERENCES compra(idCompra),  
  FOREIGN KEY (idUsuario) REFERENCES usuario(idusuario)  
)
```

5.1.30. Busqueda

```
CREATE TABLE busqueda(  
  idBusqueda int NOT NULL AUTO_INCREMENT,  
  texto text,  
  PRIMARY KEY (idBusqueda)  
);
```

5.1.31. Artículo

```
CREATE TABLE Artículo(  
  idPublicacion int NOT NULL,  
  tipo varchar(255),  
  CHECK (tipo in ('Subasta','Venta')),  
  PRIMARY KEY (idPublicacion),  
  FOREIGN KEY (idPublicacion) REFERENCES Publicacion(idPublicacion)  
);
```

5.1.32. Triggers

```
USE 'tp1';
```

```
DELIMITER $$
```

```
DROP TRIGGER IF EXISTS tp1.oferta_BEFORE_INSERT$$
```

```
USE 'tp1'$$
```

```
CREATE DEFINER = CURRENT_USER TRIGGER 'tp1'.'ofertaValida' BEFORE INSERT ON 'oferta' FOR EACH ROW  
BEGIN
```

```
IF NEW.monto > (select precio from publicacion where idPublicacion = new.idPublicacion) and NEW.monto < 1000000  
update publicacion set precio = new.monto where idPublicacion = new.idPublicacion;
```

```
END IF;
```

```
--
```

```
END$$
```

```
DELIMITER ;
```

5.2. Inserccion de datos de prueba

5.2.1. Visita

```
INSERT INTO 'tp1'.'visita' ('idUsuario', 'idPublicacion', 'fecha', 'hora') VALUES ('1', '1', '2008-0
```

```
INSERT INTO 'tp1'.'visita' ('idUserario', 'idPublicacion', 'fecha', 'hora') VALUES ('1', '4', '2008-0
```

5.2.2. Vigente

```
INSERT INTO 'tp1'.'vigente' ('idPublicacion') VALUES ('1');  
INSERT INTO 'tp1'.'vigente' ('idPublicacion') VALUES ('2');
```

5.2.3. Venta

```
INSERT INTO 'tp1'.'venta' ('idPublicacion') VALUES ('1');  
INSERT INTO 'tp1'.'venta' ('idPublicacion') VALUES ('2');
```

5.2.4. Usuario

```
INSERT INTO usuario (tipo, cantCalificaciones, puntajeAcumulado, email, telefono, idDireccion, idLoc  
INSERT INTO usuario (tipo, cantCalificaciones, puntajeAcumulado, email, telefono, idDireccion, idLoc
```

5.2.5. Tipo de publicacion

```
INSERT INTO 'tp1'.'tipo_de_publicacion' ('idTipoPublicacion', 'nombre', 'nivel', 'semanasParaCaducar  
INSERT INTO 'tp1'.'tipo_de_publicacion' ('idTipoPublicacion', 'nombre', 'nivel', 'semanasParaCaducar
```

5.2.6. Tarjeta

```
INSERT INTO 'tp1'.'tarjeta' ('idPago', 'numCuotas', 'numero', 'titular', 'fechaVencimiento', 'codSeg
```

5.2.7. Subasta

```
INSERT INTO 'tp1'.'subasta' ('idPublicacion', 'precioBase') VALUES ('1', '100');
```

5.2.8. Servicio

```
INSERT INTO 'tp1'.'servicio' ('idPublicacion', 'comision') VALUES ('5', '10');
```

5.2.9. Realiza

```
INSERT INTO 'tp1'.'realiza' ('idBusqueda', 'idUserario') VALUES ('1', '1');  
INSERT INTO 'tp1'.'realiza' ('idBusqueda', 'idUserario') VALUES ('2', '3');
```

5.2.10. Publicacion

```
INSERT INTO 'tp1'.'publicacion' ('idPublicacion', 'idCategoria', 'idUserario', 'idTipoDePublicacion',  
INSERT INTO 'tp1'.'publicacion' ('idPublicacion', 'idCategoria', 'idUserario', 'idTipoDePublicacion',
```

5.2.11. Publica por categoria

```
INSERT INTO 'tp1'.'publica_por_categoria' ('idCategoriaPublicacion', 'cantidadVendidos', 'idUserario')  
INSERT INTO 'tp1'.'publica_por_categoria' ('idCategoriaPublicacion', 'cantidadVendidos', 'idUserario')
```

5.2.12. Pregunta

```
INSERT INTO 'tp1'.'pregunta' ('idPregunta', 'idPublicacion', 'texto_pregunta', 'texto_respuesta', 'idUserario')  
INSERT INTO 'tp1'.'pregunta' ('idPregunta', 'idPublicacion', 'texto_pregunta', 'idUserario') VALUES ('1', '1', 'Pregunta', 'Respuesta')
```

5.2.13. Persona

```
INSERT INTO 'tp1'.'persona' ('idUserario', 'nombre', 'apellido', 'DNI') VALUES ('1', 'Christian', 'Ruiz', '12345678')  
INSERT INTO 'tp1'.'persona' ('idUserario', 'nombre', 'apellido', 'DNI') VALUES ('3', 'Guido', 'Kaska', '87654321')
```

5.2.14. Pago

```
INSERT INTO 'tp1'.'pago' ('idPago', 'idFactura', 'tipoDePago') VALUES ('1', '1', 'Efectivo');  
INSERT INTO 'tp1'.'pago' ('idPago', 'idFactura', 'tipoDePago') VALUES ('2', '2', 'Tarjeta');
```

5.2.15. Oferta

```
INSERT INTO 'tp1'.'oferta' ('idPublicacion', 'idUserario', 'fecha', 'hora', 'monto') VALUES ('1', '1', '2008-02-02', '14:00', '1000');
```

5.2.16. Localidad

```
INSERT INTO localidad (nombre) VALUES ('25 de Mayo');  
INSERT INTO localidad (nombre) VALUES ('9 de Julio');
```

5.2.17. Finalizada

```
INSERT INTO 'tp1'.'finalizada' ('idPublicacion', 'fechaFinalizacion') VALUES ('4', '2008-02-02');
```

5.2.18. Factura

```
INSERT INTO 'tp1'.'factura' ('idFactura', 'totalAPagar', 'fecha', 'idUserario') VALUES ('1', '1000', '2008-02-02', '1');  
INSERT INTO 'tp1'.'factura' ('idFactura', 'totalAPagar', 'fecha', 'idUserario') VALUES ('2', '574', '2008-02-02', '3');
```

5.2.19. es favorita

```
INSERT INTO 'tp1'.'es_favorita' ('idUserario', 'idPublicacion') VALUES ('1', '1');  
INSERT INTO 'tp1'.'es_favorita' ('idUserario', 'idPublicacion') VALUES ('3', '3');
```


5.2.20. Empresa

```
INSERT INTO 'tp1'.'empresa' ('idUsuario', 'RazonSocial', 'cuit', 'idCatEmpresa') VALUES ('1', 'Empre
INSERT INTO 'tp1'.'empresa' ('idUsuario', 'RazonSocial', 'cuit', 'idCatEmpresa') VALUES ('2', 'Empre
```

5.2.21. Efectivo

```
INSERT INTO 'tp1'.'efectivo' ('idPago') VALUES ('1');
INSERT INTO 'tp1'.'efectivo' ('idPago') VALUES ('3');
```

5.2.22. Direccion

```
INSERT INTO direccion (idLocalidad,numero,calle) VALUES ('6','6101',' BACACAY');
INSERT INTO direccion (idLocalidad,numero,calle) VALUES ('24','489','BACON');
```

5.2.23. Corresponde**5.2.24. Compra**

```
INSERT INTO 'tp1'.'corresponde' ('idFactura', 'idPublicacion') VALUES ('1', '1');
INSERT INTO 'tp1'.'corresponde' ('idFactura', 'idPublicacion') VALUES ('2', '2');
```

5.2.25. Categoria

```
INSERT INTO categoria (nombre, idCategoriaPadre) VALUES ('Accesorios para Vehiculos',NULL);
INSERT INTO categoria (nombre, idCategoriaPadre) VALUES ('Animales y Mascotas',NULL);
```

5.2.26. Categoria Servicio

```
INSERT INTO 'tp1'.'categoria_servicio' ('idCategoriaPublicacion') VALUES ('2');
INSERT INTO 'tp1'.'categoria_servicio' ('idCategoriaPublicacion') VALUES ('3');
```

5.2.27. Categoria Empresa

```
INSERT INTO 'tp1'.'categoria_empresa' ('idCatEmpresa', 'nombre') VALUES ('1', 'Construccion');
INSERT INTO 'tp1'.'categoria_empresa' ('idCatEmpresa', 'nombre', 'idCategoriaPadre') VALUES ('2', 'C
```

5.2.28. Categoria Articulo

```
INSERT INTO categoria_articulo(idCategoriaPublicacion) VALUES ('3');
INSERT INTO categoria_articulo (idCategoriaPublicacion) VALUES ('5');
```

5.2.29. Calificacion

```
INSERT INTO 'tp1'.'calificacion' ('idCalificacion', 'idCompra', 'idUsuario', 'puntaje', 'textoComent
INSERT INTO 'tp1'.'calificacion' ('idCalificacion', 'idCompra', 'idUsuario', 'puntaje', 'textoComent
```

5.2.30. Búsqueda

```
INSERT INTO 'tp1'.'busqueda' ('idBusqueda', 'texto') VALUES ('1', 'Conejos baratos');  
INSERT INTO 'tp1'.'busqueda' ('idBusqueda', 'texto') VALUES ('2', 'Alfombras para autos');
```

5.2.31. Artículo

```
INSERT INTO 'tp1'.'articulo' ('idPublicacion', 'tipo') VALUES ('1', 'Venta'); INSERT INTO 'tp1'.'articulo'  
('idPublicacion', 'tipo') VALUES ('2', 'Venta');
```

6. Código

Código correspondiente al sistema y a las consultas/stored procedures/triggers que se piden en el punto Otras Funcionalidades a Implementar

7. Conclusiones