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

Trabajo Práctico 1

Bases de Datos

Primer Cuatrimestre de 2016

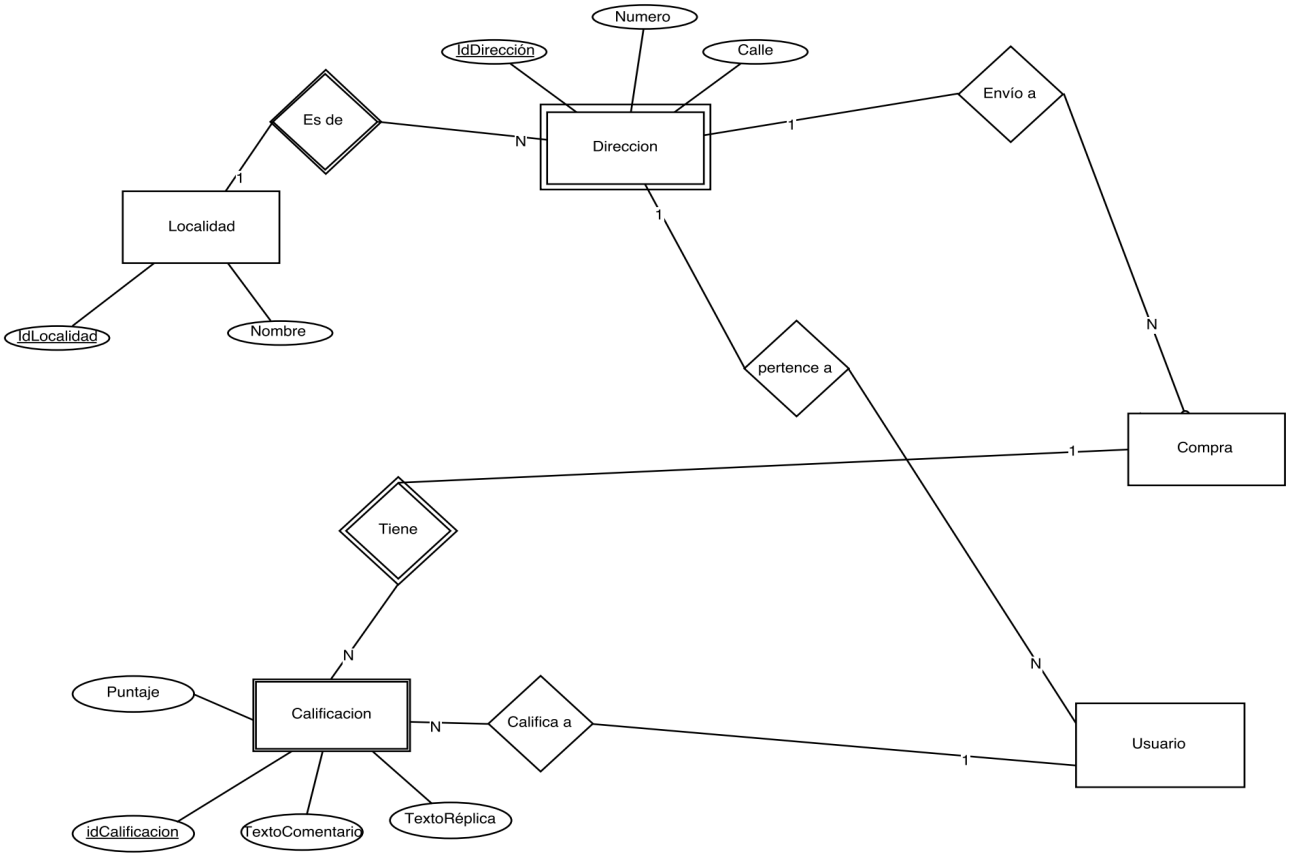
Apellido y Nombre	LU	E-mail
Federico Hosen	825/12	federico.hosen@gmail.com
Martin Caravario	470/12	martin.caravario@gmail.com
Guido Rajngewerc	379/12	guido.raj@gmail.com
Christian Russo	679/10	christian.russo8@gmail.com

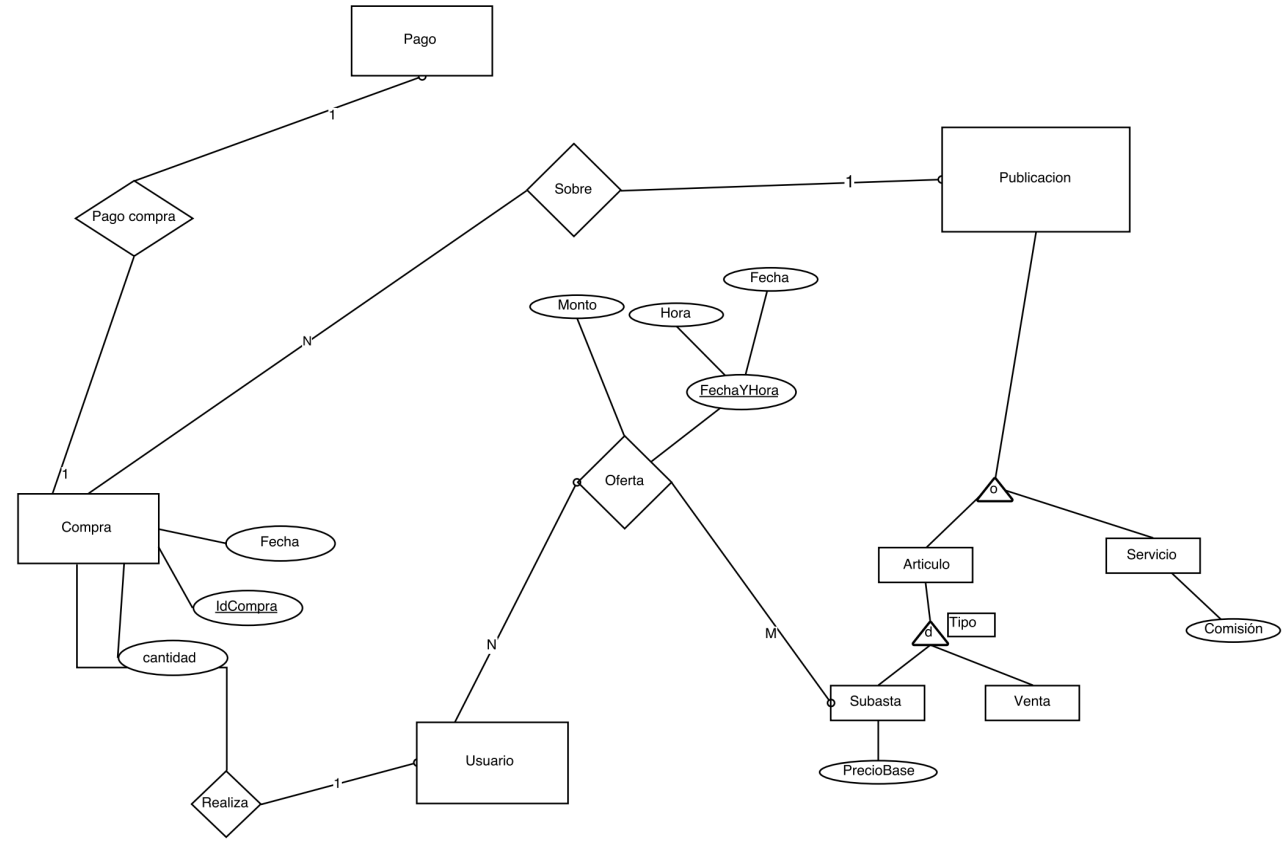
1. Introducción

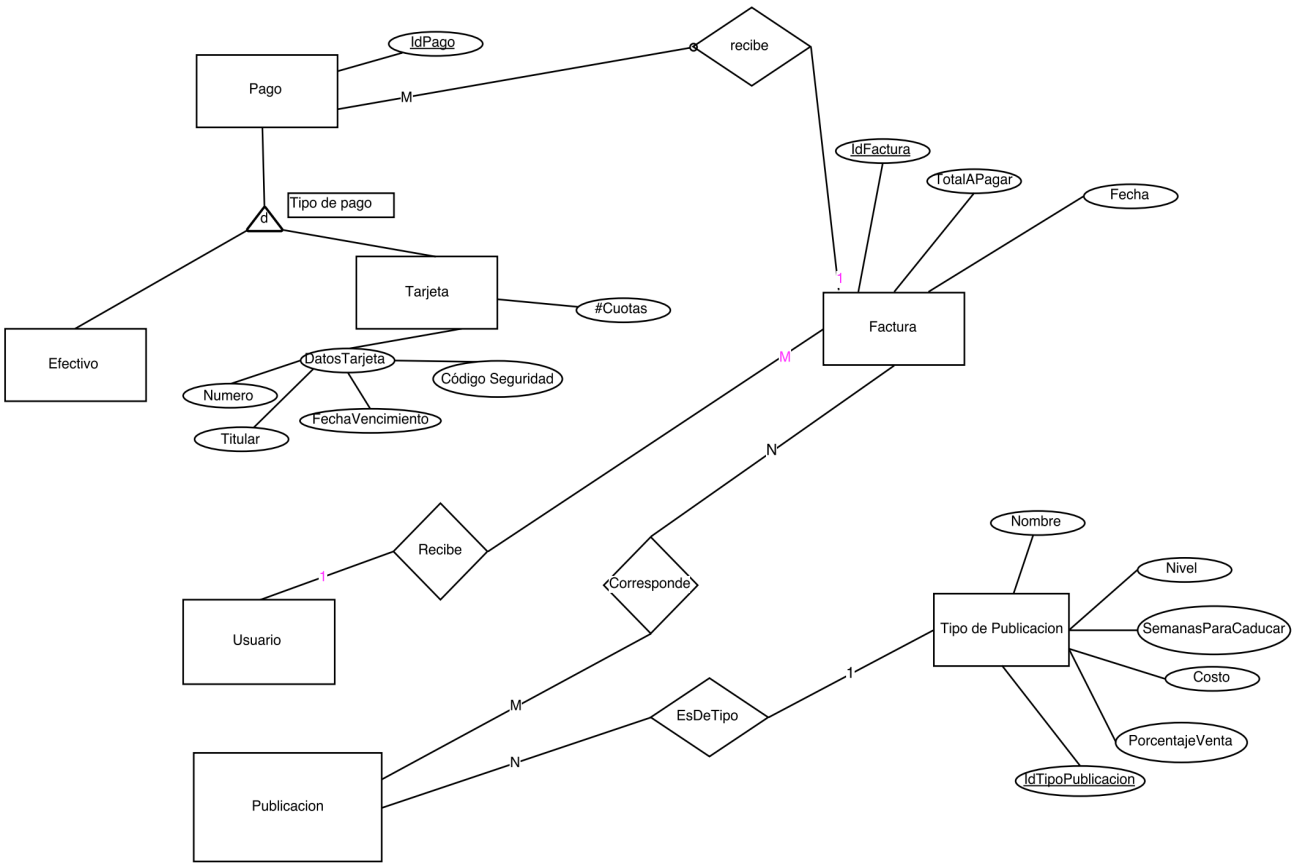
Presentaremos una solución para el problema de un **Mercado Virtual** tomando como guía el **Khan El-Khalili** ubicado en El Cairo, Egipto. El problema en cuestión contempla una serie de restricciones sobre como se realizan la compra y venta de productos por internet. Cada publicación puede tener distintos 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.

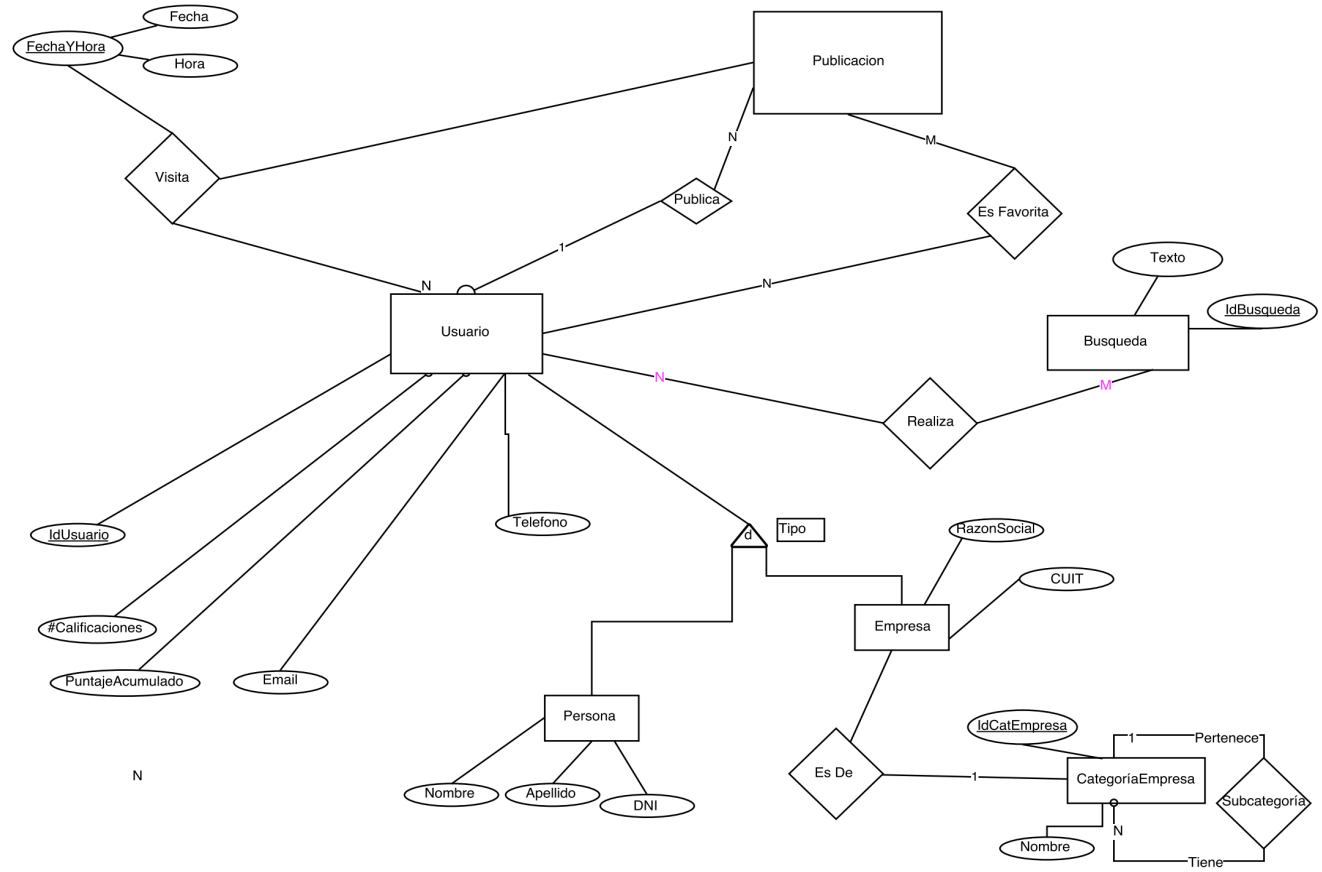
Utilizaremos las herramientas vistas en la materia, el modelado basado en el Diagrama de Entidad Relación, su MR resultante y la base de datos final que presentaremos en MySQL.

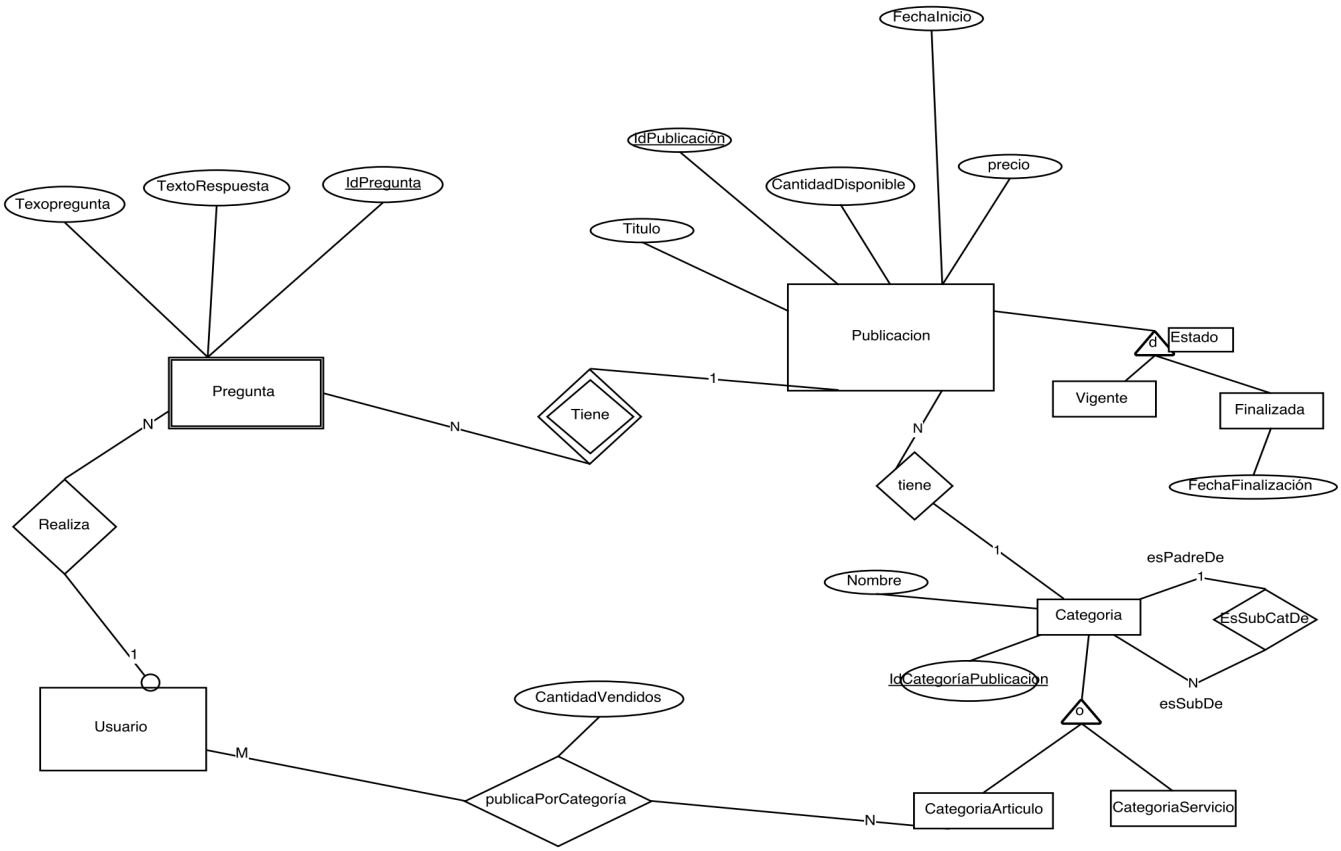
2. Diagrama de Entidad Relación




Pago
1
Pago compra
Sobre
1
Publicacion
N
Fecha
Monto
Hora
FechaYHora
1
Oferta
o
Compra
Fecha
IdCompra
cantidad
N
M
Articulo
Servicio
d
Tipo
Comisión
Subasta
Venta
PrecioBase
Usuario
Realiza
1


IdPago
recibe
Pago
M
IdFactura
TotalAPagar
Fecha
Tipo de pago
d
Tarjeta
#Cuotas
1
Factura
Efectivo
DatosTarjeta
Código Seguridad
Numero
FechaVencimiento
Titular
M
N
Recibe
Corresponde
Nombre
1
Usuario
Tipo de Publicacion
Nivel
SemanasParaCaducar
Costo
PorcentajeVenta
IdTipoPublicacion
M
N
EsDeTipo
1
Publicacion


Fecha
FechaYHora
Hora
Publicacion
Visita
N
M
Publica
Es Favorita
1
N
N
Usuario
Texto
IdBusqueda
Busqueda
N
M
Realiza
IdUsuario
Telefono
d
Tipo
RazonSocial
CUIT
Empresa
#Calificaciones
PuntajeAcumulado
Email
Persona
Nombre
Apellido
DNI
N
Es De
1
IdCatEmpresa
1
Pertenece
CategoríaEmpresa
Subcategoría
N
Nombre
Tiene


FechaInicio
IdPublicación
precio
CantidadDisponible
Texopregunta
TextoRespuesta
IdPregunta
Titulo
Publicacion
d
Estado
Vigente
Finalizada
FechaFinalización
Pregunta
1
Tiene
N
N
N
tiene
Realiza
1
1
esPadreDe
1
Nombre
Categoria
EsSubCatDe
IdCategoríaPublicacion
N
esSubDe
o
Usuario
M
CantidadVendidos
publicaPorCategoría
N
CategoriaArticulo
CategoriaServicio

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 RubíDelOriente 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 al 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 subasta.

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

1. Para calcular el ganador de un año específico, se calcula qué calificación tuvo ese año y aquel que más pago al sitio. Esos datos alcanzan para que el departamento de marketing haga una ponderación y calcule el ganador

2. Si se realiza una oferta invalida, es decir que no cumple con lo necesario de la subasta, esa oferta se guarda en la tabla oferta pero el monto de la publicación no se modifica. De esta forma la tabla oferta contendrá datos que son inválidos, pero la oferta siempre estará con el valor correcto.

3. Los datos de prueba cargados en la tablas no son consistentes en su totalidad, es decir puede existir un Usuario que publica una oferta y a la vez este usuario comprar la misma.

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 publicación

```
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. Publicación

```
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 categoría

```
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 (idUsuario),
    FOREIGN KEY (idUsuario) REFERENCES Usuario(idUsuario)
);
```

5.1.14. Pago

```
CREATE TABLE pago(
idPago int not null AUTO_INCREMENT,
idFactura int,
    tipoDePago varchar(255) not null, # check contraint (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 (idUsuario) 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. Dirección

```
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. Categoría

```
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. Categoría Servicio

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

5.1.27. Categoría 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. Categoría Artículo

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

5.1.29. Calificación

```
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. Búsqueda

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

5.1.31. Artículo

```
CREATE TABLE Articulo(
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

A continuacion se muestran los triggers utilizados en las tablas.

```
USE ‘tp1‘;

DELIMITER $$

DROP TRIGGER IF EXISTS tp1.oferta_BEFORE_INSERT$$
USE ‘tp1‘$$
CREATE DEFINER = CURRENT_USER TRIGGER ofertaValida BEFORE INSERT ON ‘oferta‘ FOR EACH ROW
BEGIN
IF NEW.monto > (select precio from publicacion where idPublicacion = new.idPublicacion) and
NEW.monto <= (select 2*precio from publicacion where idPublicacion = new.idPublicacion) THEN
update publicacion set precio = new.monto where idPublicacion = new.idPublicacion;
END IF;
 --
END$$
DELIMITER ;
```

5.2. Inserción de datos de prueba

A continuación se muestran las queries ejecutadas para insertar los datos. Observaciones:

1. Para cada tabla, utilizamos más queries de las mostradas, pero para simplicidad del informe mostramos un subconjunto de ellas.
2. Los datos ingresados en las tablas no son totalmente consistentes. Es decir, pueden existir usuarios que realizaron pagos pero no tienen compras. Es importante notar que, aunque existan inconsistencias en las tablas, las FK y CK se cumplen, sino sería imposible insertar datos.

5.2.1. Visita

```
INSERT INTO visita (‘idUsuario‘, ‘idPublicacion‘, ‘fecha‘, ‘hora‘)
VALUES (’1’, ’1’, ’2008-01-02’, ’20’);

INSERT INTO visita (‘idUsuario‘, ‘idPublicacion‘, ‘fecha‘, ‘hora‘)
VALUES (’1’, ’4’, ’2008-01-02’, ’21’);
```

5.2.2. Vigente

```
INSERT INTO vigente (‘idPublicacion‘)
VALUES (’1’);

INSERT INTO vigente (‘idPublicacion‘)
VALUES (’2’);
```

5.2.3. Venta

```
INSERT INTO venta (‘idPublicacion‘)
VALUES (’1’);

INSERT INTO venta (‘idPublicacion‘)
VALUES (’2’);
```

5.2.4. Usuario

```
INSERT INTO usuario (tipo, cantCalificaciones, puntajeAcumulado, email,
 telefono, idDireccion, idLocalidad)
VALUES ('Persona',0,0,'christian.russo8@gmail.com', '48444479', 2,4);

INSERT INTO usuario (tipo, cantCalificaciones, puntajeAcumulado, email,
telefono, idDireccion, idLocalidad)
VALUES ('Persona',0,0,'guido.raj@gmail.com', '154444444', 3,9);
```

5.2.5. Tipo de publicación

```
INSERT INTO tipo_de_publicacion` (`idTipoPublicacion`, `nombre`, `nivel`,
 `semanasParaCaducar`, `costo`, `porcentajeVenta`)
VALUES ('1', 'RubiDeOriente', '1', '10', '100', '10');

INSERT INTO tipo_de_publicacion` (`idTipoPublicacion`, `nombre`, `nivel`,
`semanasParaCaducar`, `costo`, `porcentajeVenta`)
VALUES ('2', 'Oro', '2', '12', '80', '8');
```

5.2.6. Tarjeta

```
INSERT INTO tarjeta (`idPago`, `numCuotas`, `numero`, `titular`, `fechaVencimiento`, `codSeguridad`)
VALUES ('2', '6', '1234567812345678', 'Christian Russo', '2008-01-02', '123');
```

5.2.7. Subasta

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

5.2.8. Servicio

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

5.2.9. Realiza

```
INSERT INTO realiza` (`idBusqueda`, `idUsuario`)
VALUES ('1', '1');

INSERT INTO realiza (`idBusqueda`, `idUsuario`)
VALUES ('2', '3');
```

5.2.10. Publicación

```
INSERT INTO publicacion`(`idPublicacion`, `idCategoria`, `idUsuario`, `idTipoDePublicacion`,
 `estado`, `fechaInicio`, `titulo`, `cantidadDisponible`, `precio`)
VALUES ('1', '1', '1', '1', 'Vigente', '2008-01-02', 'Alfombra para auto', '2', '1000');

INSERT INTO publicacion(`idPublicacion`, `idCategoria`, `idUsuario`, `idTipoDePublicacion`,
`estado`, `fechaInicio`, `titulo`, `cantidadDisponible`, `precio`)
VALUES ('2', '8', '1', '1', 'Vigente', '2008-01-02', 'Poni recien nacido', '1', '10000');
```

5.2.11. Publica por categoría

```
INSERT INTO publica_por_categoria(‘idCategoriaPublicacion‘, ‘cantidadVendidos‘, ‘idUsuario‘)
VALUES (’1’, ’0’, ’1’);

INSERT INTO publica_por_categoria (‘idCategoriaPublicacion‘, ‘cantidadVendidos‘, ‘idUsuario‘)
VALUES (’2’, ’2’, ’3’);
```

5.2.12. Pregunta

```
INSERT INTO pregunta (‘idPregunta‘, ‘idPublicacion‘, ‘texto_pregunta‘, ‘texto_respuesta‘,
 ‘idUsuario‘)
VALUES (’1’, ’1’, ’Hola, queria saber si la alfombra esta nueva’, ’No, esta usada’, ’4’);

INSERT INTO pregunta (‘idPregunta‘, ‘idPublicacion‘, ‘texto_pregunta‘, ‘idUsuario‘)
VALUES (’2’, ’1’, ’Hola, las alfombras son para un Fiat?’, ’4’);
```

5.2.13. Persona

```
INSERT INTO persona (‘idUsuario‘, ‘nombre‘, ‘apellido‘, ‘DNI‘)
VALUES (’1’, ’Christian’, ’Russo’, ’35561654’);

INSERT INTO persona (‘idUsuario‘, ‘nombre‘, ‘apellido‘, ‘DNI‘)
VALUES (’3’, ’Guido’, ’Kaska’, ’35561666’);
```

5.2.14. Pago

```
INSERT INTO pago (‘idPago‘, ‘idFactura‘, ‘tipoDePago‘)
VALUES (’1’, ’1’, ’Efectivo’);

INSERT INTO pago (‘idPago‘, ‘idFactura‘, ‘tipoDePago‘)
VALUES (’2’, ’2’, ’Tarjeta’);
```

5.2.15. Oferta

```
INSERT INTO oferta (‘idPublicacion‘, ‘idUsuario‘, ‘fecha‘, ‘hora‘, ‘monto‘)
VALUES (’1’, ’1’, ’2008-01-02’, ’10’, ’100’);
```

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 finalizada (‘idPublicacion‘, ‘fechaFinalizacion‘)
VALUES (’4’, ’2008-02-02’);
```

5.2.18. Factura

```
INSERT INTO factura (‘idFactura‘, ‘totalAPagar‘, ‘fecha‘, ‘idUsuario‘)
VALUES (’1’, ’1000’, ’2008-01-02’, ’1’);

INSERT INTO factura (‘idFactura‘, ‘totalAPagar‘, ‘fecha‘, ‘idUsuario‘)
VALUES (’2’, ’574’, ’2008-01-02’, ’3’);
```

5.2.19. Es Favorita

```
INSERT INTO es_favorita (‘idUsuario‘, ‘idPublicacion‘)
VALUES (’1’, ’1’);

INSERT INTO es_favorita (‘idUsuario‘, ‘idPublicacion‘)
VALUES (’3’, ’3’);
```

5.2.20. Empresa

```
INSERT INTO empresa(‘idUsuario‘, ‘RazonSocial‘, ‘cuit‘, ‘idCatEmpresa‘)
VALUES (’1’, ’Empresa1’, ’20355616542’, ’1’);

INSERT INTO empresa (‘idUsuario‘, ‘RazonSocial‘, ‘cuit‘, ‘idCatEmpresa‘)
VALUES (’2’, ’Empresa2’, ’20355616542’, ’1’);
```

5.2.21. Efectivo

```
INSERT INTO efectivo (‘idPago‘)
VALUES (’1’);

INSERT INTO efectivo (‘idPago‘)
VALUES (’3’);
```

5.2.22. Dirección

```
INSERT INTO direccion (idLocalidad,numero,calle)
VALUES (’6’,’6101’,’ BACACAY’);

INSERT INTO direccion (idLocalidad,numero,calle)
VALUES (’24’,’489’,’BACON’);
```

5.2.23. Corresponde

```
INSERT INTO corresponde (‘idFactura‘, ‘idPublicacion‘)
VALUES (’1’, ’1’);

INSERT INTO corresponde (‘idFactura‘, ‘idPublicacion‘)
VALUES (’2’, ’2’);
```

5.2.24. Compra

```
INSERT INTO compra(‘idCompra‘, ‘idUsuario‘, ‘idPublicacion‘, ‘idPago‘,
‘idDireccion‘, ‘fecha‘, ‘cantidad‘)
VALUES (’1’, ’1’, ’2’, ’1’, ’1’, ’2008-01-02’, ’1’);
```

5.2.25. Categoría

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

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

5.2.26. Categoría Servicio

```
INSERT INTO categoria_servicio (`idCategoriaPublicacion`)
VALUES ('2');

INSERT INTO categoria_servicio (`idCategoriaPublicacion`)
VALUES ('3');
```

5.2.27. Categoría Empresa

```
INSERT INTO categoria_empresa (`idCatEmpresa`, `nombre`)
VALUES ('1', 'Construccion');

INSERT INTO categoria_empresa (`idCatEmpresa`, `nombre`, `idCategoriaPadre`)
VALUES ('2', 'Carpinteria', '1');
```

5.2.28. Categoría Artículo

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

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

5.2.29. Calificación

```
INSERT INTO calificacion (`idCalificacion`, `idCompra`, `idUsuario`, `puntaje`,
 `textoComentario`, `textoReplica`)
VALUES ('1', '3', '4', '10', 'Un capo!', 'Gracias');

INSERT INTO calificacion (`idCalificacion`, `idCompra`, `idUsuario`, `puntaje`,
`textoComentario`, `textoReplica`)
VALUES ('2', '4', '4', '4', 'Me cago', 'Si');
```

5.2.30. Búsqueda

```
INSERT INTO busqueda (`idBusqueda`, `texto`)
VALUES ('1', 'Conejos baratos');

INSERT INTO busqueda (`idBusqueda`, `texto`)
VALUES ('2', 'Alfombras para autos');
```

5.2.31. Artículo

```
INSERT INTO articulo (‘idPublicacion‘, ‘tipo‘)
VALUES (’1’, ’Venta’);

INSERT INTO articulo (‘idPublicacion‘, ‘tipo‘)
VALUES (’2’, ’Venta’);
```

6. Código

6.1. Consultas por Usuario

6.1.1. Obtener, para un usuario específico, información sobre los artículos que ha comprado y vendido

```
select titulo from publicacion p join
compra c on p.idPublicacion = c.idPublicacion join articulo a
on p.idPublicacion = a.idPublicacion
where p.idUsuario = 1 or c.idUsuario = 1
```

6.1.2. Dado un usuario, los artículos que visitó con su fecha

```
select titulo,email, fecha from visita v
join usuario u on v.idUsuario = u.idUsuario
join articulo a on a.idPublicacion = v.idPublicacion
join publicacion p on p.idPublicacion = v.idPublicacion
where  v.idUsuario = 3
```

6.1.3. Dado un usuario, los artículos que tiene en su lista de favoritos.

```
select titulo, email from es_favorita ef
join publicacion p on ef.idPublicacion = p.idPublicacion
join usuario u on ef.idusuario = u.idUsuario
where p.idUsuario = 1
```

6.1.4. Dado un usuario, las primeras tres categorías más visitadas en el ultimo año

```
select idCategoria, u.idUsuario, count(*) as cantidad, nombre from visita v
join publicacion p on v.idPublicacion = p.idPublicacion
join usuario u on u.idUsuario = v.idUsuario and u.idUsuario = 1
join categoria c on c.idCategoriaPublicacion = p.idCategoria
where datediff(CURDATE(), fecha) <= 365
group by idCategoria, u.idUsuario
order by cantidad desc
limit 3
```

6.2. Consultas por categoría de producto

6.2.1. Obtener dado una categoría de producto un listado de los vendedores que han publicado artículos de dicha categoría y la cantidad de ventas que hizo cada uno.

```
select u.idUsuario,u.email,  sum(c.cantidad) as CantidadVendido from publicacion p
join usuario u on u.idUsuario = p.idUsuario
    join compra c on c.idPublicacion = p.idPublicacion
where p.idCategoria = 1
    group by u.idUsuario, u.email
```

6.3. Función Ofertar

6.3.1. Debe permitir al usuario ofertar una suma en una subasta, un peso mayor a la oferta actual y menor al doble.

```
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 <= (select 2*precio from publicacion where idPublicacion = new.idPublicacion) THEN
update publicacion set precio = new.monto where idPublicacion = new.idPublicacion;
END IF;
 --
END$$
DELIMITER ;
```

6.4. Consultas por Usuario y preguntas

6.4.1. Obtener para un usuario la lista de preguntas que hizo y su respuesta.

```
select email, texto_pregunta, texto_respuesta from pregunta p join usuario u
on p.Idusuario = u.idUsuario
where u.idUsuario = 4
```

6.5. Consultas por keyword

6.5.1. Obtener para un cierto keyword la lista de publicaciones vigentes que tengan en el título dicho keyword.

```
select * from publicacion p
join vigente v on  p.idPublicacion = v.idPublicacion
where p.titulo LIKE "%alfombra%" and p.idCategoria = 2
```

6.6. Consultas por ganadores

6.6.1. obtener, para un año específico, el ganador/es.

Primero calculamos los usuarios con mayor calificación anual:

```
select * from (Select ca.idUsuario, sum(puntaje)/count(*) as reputacionAnual from calificacion ca
join compra c on ca.idCompra = c.idCompra
where fecha like "%2008%"
group by ca.idUsuario) as reps
where reps.reputacionAnual in (select max(reputacionAnual) from (
Select sum(puntaje)/count(*) as reputacionAnual from calificacion ca
join compra c on ca.idCompra = c.idCompra
where fecha like "%2008%"
group by ca.idUsuario) as maximo)
```

Luego calculamos los usuarios que más pagaron al sitio

```
select * from (Select ca.idUsuario, sum(puntaje)/count(*) as reputacionAnual from calificacion ca
join compra c on ca.idCompra = c.idCompra
where fecha like "%2008%"
group by ca.idUsuario) as reps
where reps.reputacionAnual in (select max(reputacionAnual) from (
Select sum(puntaje)/count(*) as reputacionAnual from calificacion ca
join compra c on ca.idCompra = c.idCompra
where fecha like "%2008%"
group by ca.idUsuario) as maximo)
```

Con estos dos resultados se los enviamos al departamento del marketing para que evalúe al ganador.

7. Conclusiones

Luego de realizar este trabajo práctico concluimos que a la hora de necesitar realizar las tablas de una base de datos de un sistema de gran tamaño, es conveniente armar diagramas como el DER y MR. Estos diagramas nos ayudaron enormemente a resolver las queries pedidas y poder hacer las tablas sin errores. Más allá que hacer el MR a partir de un DER es un trabajo tedioso, es muy recomendable hacerlo dado que nosotros para realizar las queries podíamos ver los campos y la relaciones entre tablas fácilmente así también como las FK y CK de cada una.