



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 2

## Bases de Datos

Primer Cuatrimestre de 2016

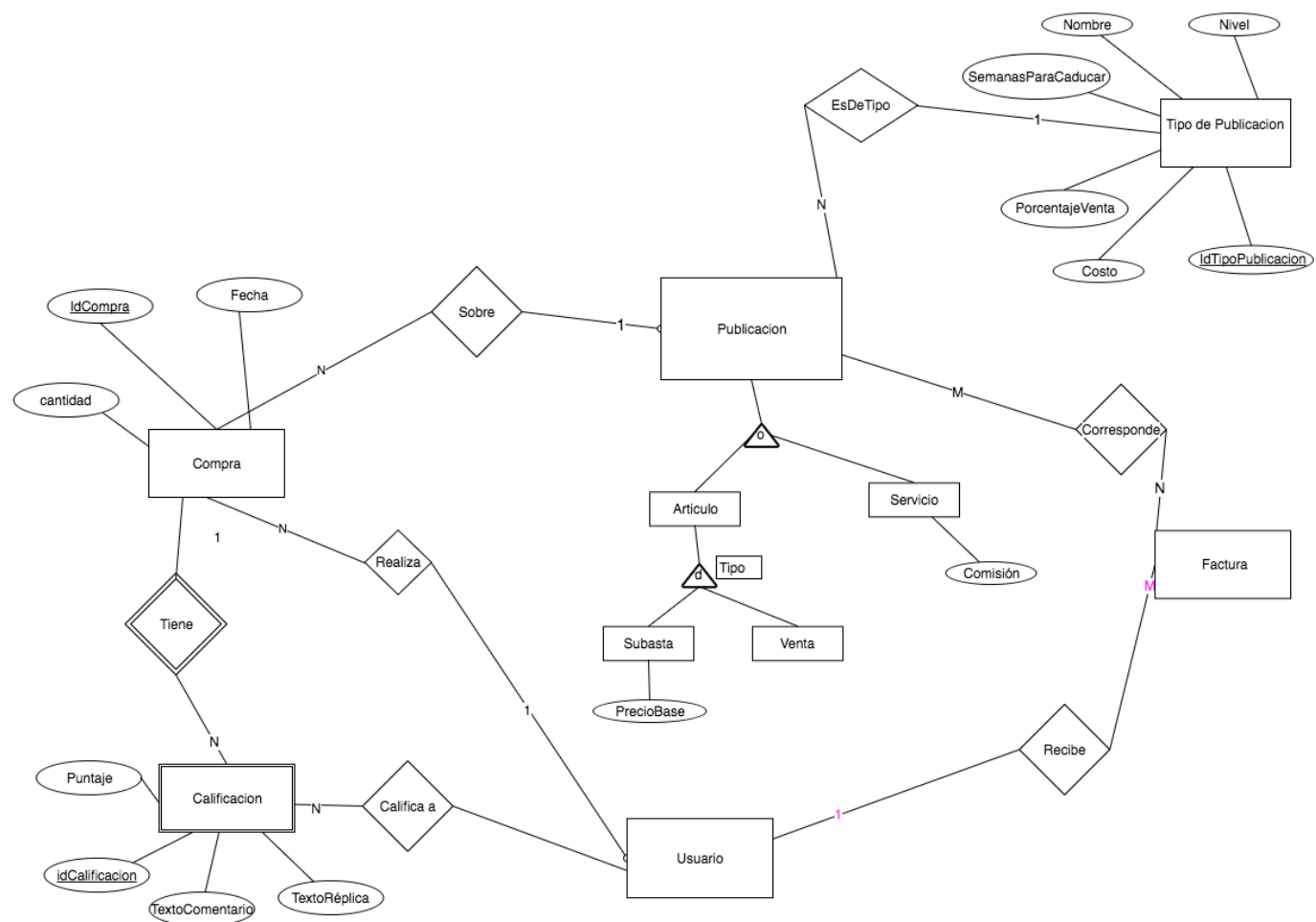
Apellido y Nombre	LU	E-mail
Federico Hosen	825/12	federico.hosen@gmail.com
Martin Caravario	470/12	martin.caravario@gmail.com
Guido Rajngewerc	379/12	guido.raj@gmail.com
Christian Russo	679/10	christian.russo8@gmail.com

## 1. Introducción

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

## 2. Parte 1 - Diseño

### 2.1. Diagrama Entidad relacion



Explicacion de porque quedo ese der.

## 2.2. Colecciones

A continuacion mostramos las colecciones que utilizamos para resolver las consultas propuestas en el enunciado

Coleccion de publicaciones

```
1 {
2   "Id": 65,
3   "TipoPublicacion": "Servicio"
4 }
```

Coleccion de compra

```
1 {
2 {
3   "idCompra": 656897,
4   "idUsuarioComprador": 65468,
5   "Fecha": "2-06-2014",
6   "Cantidad": 2,
7   "Publicacion": {
8     "idPublicacion": 2365,
9     "idUsuario": 123,
10    "Precio": 3655,
11    "PorcentajeVenta": 20 //entre 0 y 100
12  },
13   "CalificacionDelVendedor": 8,
14   "CalificacionDelComprador": 6
15 }}
```

Coleccion de Factura:

```
1 {
2   "idFactura": 687468,
3   "IdUsuario": 654,
4   "Fecha": "16-9-2015",
5   "TotalAPagar": 5624,
6   "estoyPagando": [
7     {
8       "idPublicacion": 54,
9       "TipoSuscripcion": "Rubi"
10    },
11    {
12      "idPublicacion": 54,
13      "TipoSuscripcion": "Libre"
14    }
15  ]
16 }
```

## 2.3. Migracion datos SQL a Colecciones

Para migrar los datos de SQL (MySQLWorkwench) al formato de JSON que se necesita en MongoDB lo que hicimos fue exportar los datos directo desde MySQLWorkwench, pero estos se exportan en JSON plano, es decir no quedaba formateado de la forma que nosotros queriamos. Para solucionar este problema realizamos unos algoritmos en python para formatearlos a mano. Estos algoritmos reciben como input el JSON plano y devuelven un JSON formateado para MongoDB

Parser para el JSON de la compra:

```
import json
from pprint import pprint

with open(raw_input()) as data_file:
    data = json.load(data_file)
for a in data:
    print "{"
    print '"idCompra": ', a['idCompra'], ","
    print '"idUsuario": ', a['idUsuario'], ","
    print '"fecha": ', "'", a['fecha'], "'", ","
    print '"cantidad": ', a['cantidad'], ","
    print '"Publicacion": {'
    print '"idPublicacion": ', a['idPublicacion'], ","
    print '"idUsuario": ', a['idUsuario'], ","
    print '"Precio": ', a['precio'], ","
    print '"PorcentajeVenta": ', a['porcentajeVenta']
    print "}" , ","
    print '"nombre": ', "'", a['nombre'], "'"
    print '}'
```

Parser para el JSON de la Facutra:

```
import json
from pprint import pprint

with open('factura_sql.json') as data_file:
    data = json.load(data_file)

c = {}
for a in data:
    c.setdefault(a["idFactura"], []).append(a)

for a in c:
    print '{'
    print '"idFactura": ', c[a][0]['idFactura'], ","
    print '"IdUsuario": ', c[a][0]['idUsuario'], ","
    print '"Fecha": ', "'", c[a][0]['fecha'], "'", ","
    total = 0
    for x in c[a]:
        total = total + c[a][c[a].index(x)]['totalAPagar']
    print '"TotalAPagar": ', total, ","
    print '"estoyPagando": ['
    i = 1
    for x in c[a]:
        print "{"
        print '"idPublicacion": ', c[a][c[a].index(x)]['idPublicacion'], ","
        print '"TipoSuscripcion": ', "'", c[a][c[a].index(x)]['nombre'], "'"
        if i == len(c[a]):
            print "}"
        else:
            print "},"
        i = i+1
```

```
print ']'
print '}'
```

Nota: para la coleccion de publicaciones no tuvimos que hacer ningun algoritmo.

## 2.4. Consultas SQL

A continuacion listamos las consultas realizadas para conseguir los datos necesarios para nuestras colecciones:

Para la coleccion de Publicaciones:

```
select p.idPublicacion, t.Nombre
from publicaciones p
inner join TipoDePublicacion t
on t.idTipoPublicacion = p.idTipoPublicacion
```

Para la coleccion de Compras:

```
select c.idCompra, c.idUsuario, c.fecha, c.cantidad, c.idPublicacion, p.precio, tp.cost, tp.porcentaje
from compra c
join publicacion p
on c.idPublicacion = p.idPublicacion
join tipo_de_publicacion tp
on p.idTipoDePublicacion = tp.idTipoPublicacion
```

Para la coleccion de Facturas:

```
select f.idFactura, p.idUsuario, f.fecha, f.totalAPagar, p.idPublicacion, tp.nombre
from factura f
inner join corresponde c
on c.idFactura = f.idFactura
inner join publicacion p
on c.idPublicacion = p.idPublicacion
inner join tipo_de_publicacion tp
on tp.idTipoPublicacion = p.idTipoDePublicacion
```

## 2.5. Implementacion MongoDB

Implementar en MongoDB la base de datos de documentos para almacenar la informacion y cargarla con datos de prueba

### **3. Parte 2 - Map Reduce**

### **4. Parte 3 - Sharding**

### **5. Conclusiones**