



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

## Trabajo Práctico 2

[Primera entrega]

14 de noviembre de 2016

Bases de Datos

Integrante	LU	Correo electrónico
Fernandez, Esteban	691/12	esteban.pmf@gmail.com
Marta, Cristian G.	079/12	cristiangmarta@gmail.com
Wright, Carolina Rocío	876/12	wright.carolina@gmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Diseño NoSql</b>	<b>2</b>
2.1. JSON schemas . . . . .	2
2.1.1. Información de personas . . . . .	2
2.1.2. Casos criminales . . . . .	2
2.2. Personas culpables . . . . .	3
2.3. Personas involucradas . . . . .	3
2.4. Categorías . . . . .	3
<b>3. Parte 2 - Map Reduce</b>	<b>4</b>
3.1. Número promedio de crímenes cometidos por personas que ya han sido encontradas culpables de algún crimen . . . . .	4
3.2. Personas involucradas como testigos en la mayor cantidad de casos . . . . .	4
3.3. Casos en los que se han visto involucradas el mayor número de personas . . . . .	4
3.4. Cantidad de crímenes por localidad y por año . . . . .	4
3.5. Mayor número de crímenes cometido por alguna persona . . . . .	5
3.6. Cantidad total de evidencias por caso . . . . .	5
3.7. Las 10 ciudades con mayor número de crímenes . . . . .	5

## 1. Introducción

Continuaremos el desarrollo del Sistema de seguimiento de casos policiales. Agregaremos una base de datos NoSQL basada en documentos (MongoDB) para guardar el historial de los casos. De esta forma podremos hacer consultas de manera más eficiente.

## 2. Diseño NoSql

Basándonos en el diseño entregado en el Trabajo Práctico 1, hemos creado el siguiente diagrama que refleja el contenido de los documentos que utilizaremos para realizar las consultas.

### 2.1. JSON schemas

#### 2.1.1. Información de personas

```
{ "type": "object ",
  "properties": {
    "dni_persona": { "type": "integer" },
    "caso_y_rol": { "type": "array",
      "items": { "type": "object ",
        "properties": {
          "id": { "type": "integer" },
          "fecha_ingreso": { "type": "string", "format": "date" },
          "fecha": { "type": "string", "format": "date" },
          "lugar": { "type": "string" },
          "descripcion": { "type": "string" },
          "nombre_categoria": { "type": "string" },
          "estado": { "type": "string" },
          "nombre_rol": { "type": "string" }
        }
      }
    }
  }
}
```

Este documento se creó para resolver las siguientes consultas: personas involucradas como testigos en la mayor cantidad de casos y mayor número de crímenes cometido por alguna persona. En ambas consultas buscamos información de las personas que participaron en casos criminales. Pudimos unificarlas en un mismo documento, agregando al mismo el rol en el caso (necesario para la primer consulta) y el nombre de la categoría del crimen (necesario para la segunda).

#### 2.1.2. Casos criminales

```
{ "type": "object ",
  "properties": {
    "id": { "type": "integer" },
    "fecha_ingreso": { "type": "string", "format": "date" },
    "fecha": { "type": "string", "format": "date" },
    "lugar": { "type": "string" },
    "descripcion": { "type": "string" },
    "categoria": { "type": "string" },
    "evidencias": { "type": "array",
      "items": { "type": "object ",
        "properties": {
          "idEvidencia": { "type": "integer" }
        }
      }
    }
  }
}
```

Los casos criminales nos ayudaron a resolver: cantidad de crímenes por localidad y por año, y cantidad total de evidencias por caso. Para ambas consultas necesitamos información de los casos. Para la primera la localidad y año, que son atributos de la entidad Caso criminal. Para la segunda, embebimos las evidencias en este documento que provienen de la relación Caso criminal - recolecta - Evidencia.

## 2.2. Personas culpables

```
{ "type": "object ",
  "properties": {
    "id_caso_criminal": { "type": "integer" },
    "personas_involucradas": { "type": "array",
      "items": { "type": "object ",
        "properties": {
          "dni_persona": { "type": "integer" }
        }
      }
    }
  }
}
```

Decidimos crear este documento para poder resolver la primera consulta: número promedio de crímenes cometidos por personas que ya han sido encontradas culpables de algún crimen. De esta forma tenemos agrupadas las personas involucradas por caso criminal. Esto nos permitirá poder filtrar dentro de las personas involucradas a las culpables.

## 2.3. Personas involucradas

```
{ "type": "object ",
  "properties": {
    "id_caso_criminal": { "type": "integer" },
    "personas_involucradas": { "type": "array",
      "items": { "type": "object ",
        "properties": {
          "dni_persona": { "type": "integer" }
        }
      }
    }
  }
}
```

En este caso, el documento fue pensado para resolver: casos en los que se han visto involucradas el mayor número de personas. Agrupamos las personas involucradas en los casos criminales por caso. Tomando la cantidad de los involucrados ya podemos resolver la consulta.

## 2.4. Categorías

```
{ "type": "object ",
  "properties": {
    "nombre": { "type": "string" },
    "casos_criminales": { "type": "array",
      "items": { "type": "object ",
        "properties": {
          "id": { "type": "integer" },
          "lugar": { "type": "string" }
        }
      }
    }
  }
}
```

Este documento surge para poder resolver: las 10 ciudades con mayor número de crímenes. De esta forma tenemos agrupados los casos por categoría, en este caso nos concierne la categoría Crimenz de ella las ciudades. Por este motivo embebimos en Categoría los atributos id y lugar de Caso

criminal.

### 3. Parte 2 - Map Reduce

A continuación se presentarán los algoritmos utilizados para resolver las consultas pedidas a partir de la utilización de Map-Reduce.

#### 3.1. Número promedio de crímenes cometidos por personas que ya han sido encontradas culpables de algún crimen

```
function map(){
    emit("acum", this.personas_culpables.length);
}

function reduce(acum_key, criminals){
    var amount_of_criminals = 0;
    var amount_of_criminal_cases = criminals.length;

    for (var i = 0; i < amount_of_criminal_cases; i++) {
        amount_of_criminals += criminals[i];
    }

    return amount_of_criminals / amount_of_criminal_cases;
}

db.personasCulpables.mapReduce(map, reduce, {out: {inline: 1}})
```

En esta consulta, nos interesaba calcular el promedio de los crímenes. Es por ello que fuimos acumulando estos valores, todos juntos, para luego poder calcular el promedio sobre ellos.

#### 3.2. Personas involucradas como testigos en la mayor cantidad de casos

#### 3.3. Casos en los que se han visto involucradas el mayor número de personas

```
function map(){
    emit("acum", {id: this.id_caso_criminal, count: this.
        personas_involucradas.length});
}

function reduce(acum_key, cases) {
    var result_cases = [];
    var max_people = 0;

    cases.forEach( function(c) {
        if (c.count > max_people) {
            max_people = c.count;
            result_cases = [];
            result_cases.push(c.id);
        }
        else if (c.count == max_people) {
            result_cases.push(c.id);
        }
    });
    return {"result": result_cases};
}

db.personasInvolucradas.mapReduce(map, reduce, {out: {inline: 1}})
```

Como buscamos de entre todos los casos, los que hayan tenido involucrados mayor cantidad de personas, nos pareció conveniente acumular estos valores y luego filtrar para quedarnos con los que efectivamente cumplen lo buscado. El map-reduce se hace sobre la colección de personas involucradas que contiene para cada caso, una lista de las personas involucradas.

#### 3.4. Cantidad de crímenes por localidad y por año

```

function map(){
    var year = new Date(Date.parse(this.fecha)).getFullYear();
    emit(this.lugar + year, 1);
}

function reduce(cityAndYear, count){
    return count.length;
}

db.casosCriminales.mapReduce(map, reduce, {out: {inline: 1}})

```

En este caso, la función map agrega por cada caso que sucede en un lugar-año un elemento a la lista relacionada con ese lugar-año. Luego en la función reduce solo resta tomar el tamaño de esa lista para obtener la cantidad de crímenes asociados a esa localidad y año (pues por cada caso criminal se agrego un elemento a la lista).

### 3.5. Mayor número de crímenes cometido por alguna persona

### 3.6. Cantidad total de evidencias por caso

```

function map(){
    emit(this.id, this.evidencias);
}

function reduce(case_id, evidences){
    return evidences.length;
}

db.casosCriminales.mapReduce(map, reduce, {out: {inline: 1}})

```

En esta consulta agrupamos por cada caso criminal, sus evidencias.

### 3.7. Las 10 ciudades con mayor número de crímenes

