

## **PROYECTO PRÁCTICO INTEGRADOR**

### **MÓDULO 2. FUNDAMENTOS**

**Arley Fuentes Arenales**

**Enlace GitHub:**

**<https://github.com/ArleyF/proyectomodulodos>**

## CONTENIDO

Parte 1 – Diagramas de Representación .....	3
Miniferreteria SA PSInt .....	3
Diagrama de flujo.....	4
Diagrama NS .....	4
Diagrama UML.....	5
Parte 2 – Programación Orientada a Objetos.....	6
Clase Personas .....	6
Clase Animales.....	7
Clase Veterinaria .....	8
Diagrama de base de datos relacional .....	9
Diagrama de base de datos no relacional .....	9
Creación de una BD No relacional con Mongo DB: .....	10
Capturas MongoDB.....	11

## PARTE 1 – DIAGRAMAS DE REPRESENTACIÓN

Crear un archivo que te permita calcular los ingresos de un mes de la compañía Miniferretería SA con las siguientes condiciones:

- Ingresar el valor de los artículos “martillos” y “puntillas”.
- Si supera el \$1'000,000 de pesos en ventas, agregar al resultado final (ingresos) \$200.000 de bonificación que dan los proveedores.
- ¡Ten presente! Las ventas no pueden ser inferiores a 200,000 mil pesos, si es el caso indicar por medio de un mensaje, “No cumpliste con las ventas mínimas del mes”.

### Miniferretería SA PSInt

#### Proceso Miniferreteria

```
Definir valorMartillos, valorPuntillas, ventas como Real;
```

```
Escribir "Ingrese el valor de los martillos: ";
```

```
Leer valorMartillos;
```

```
Escribir "Ingrese el valor de las puntillas: ";
```

```
Leer valorPuntillas;
```

```
ventas <- valorMartillos + valorPuntillas;
```

```
si ventas > 1000000 Entonces
```

```
    ventas <- ventas + 200000;
```

```
FinSi
```

```
si ventas < 200000 Entonces
```

```
    Escribir "No cumpliste con las ventas mínimas del mes";
```

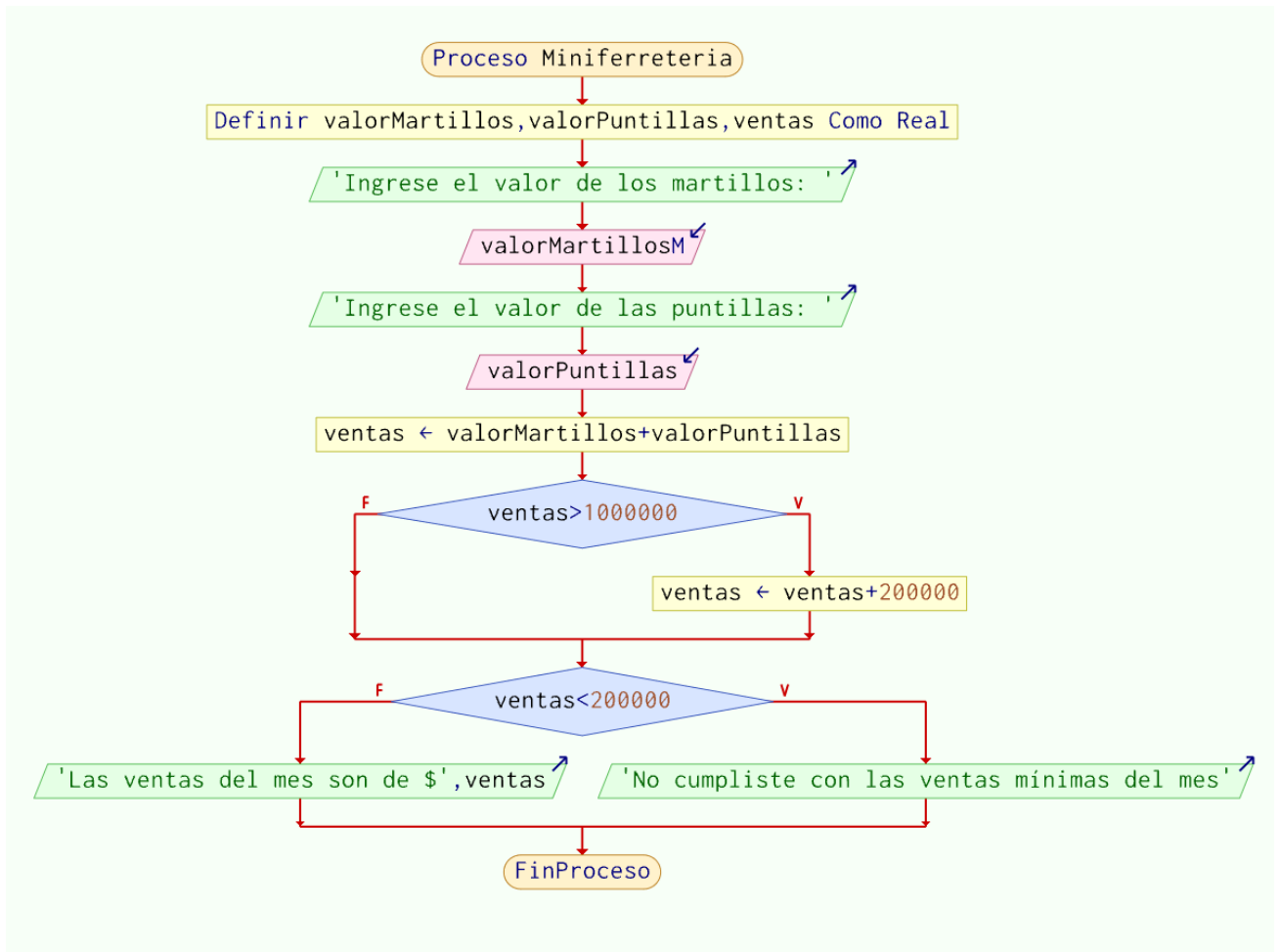
```
SiNo
```

```
    Escribir "Las ventas del mes son de $", ventas;
```

```
FinSi
```

```
FinProceso
```

## Diagrama de flujo



## Diagrama NS

Proceso Miniferreteria	
Definir valorMartillos, valorPuntillas, ventas Como Real	
Escribir 'Ingrese el valor de los martillos: '	
Leer valorMartillos	
Escribir 'Ingrese el valor de las puntillas: '	
Leer valorPuntillas	
ventas ← valorMartillos+valorPuntillas	
ventas > 1000000	
Si	No
ventas ← ventas+200000	
ventas < 200000	
Si	No
Escribir 'No cumpliste con las ventas minimas del mes'	Escribir 'Las ventas del mes son de \$', ventas
FinProceso	

## Diagrama UML



## PARTE 2 – PROGRAMACIÓN ORIENTADA A OBJETOS

En JavaScript

### Clase Personas

```
class Personas {  
    constructor(nombre, apellido, edad, cedula) {  
        this.nombre = nombre;  
        this.apellido = apellido;  
        this.edad = edad;  
        this.cedula = cedula;  
    }  
  
    getNombre() {  
        return this.nombre;  
    }  
  
    setNombre(nombre) {  
        this.nombre = nombre;  
    }  
  
    getApellido() {  
        return this.apellido;  
    }  
  
    setApellido(apellido) {  
        this.apellido = apellido;  
    }  
  
    getEdad() {  
        return this.edad;  
    }  
  
    setEdad(edad) {  
        this.edad = edad;  
    }  
  
    getCedula() {  
        return this.cedula;  
    }  
  
    setCedula(cedula) {  
        this.cedula = cedula;  
    }  
}
```

## Clase Animales

```
class Animales {  
    constructor(nombre, genero, raza) {  
        this.nombre = nombre;  
        this.genero = genero;  
        this.raza = raza;  
    }  
  
    getNombre() {  
        return this.nombre;  
    }  
  
    setNombre(nombre) {  
        this.nombre = nombre;  
    }  
  
    getGenero() {  
        return this.genero;  
    }  
  
    setGenero(genero) {  
        this.genero = genero;  
    }  
  
    getRaza() {  
        return this.raza;  
    }  
  
    setRaza(raza) {  
        this.raza = raza;  
    }  
}
```

## Clase Veterinaria

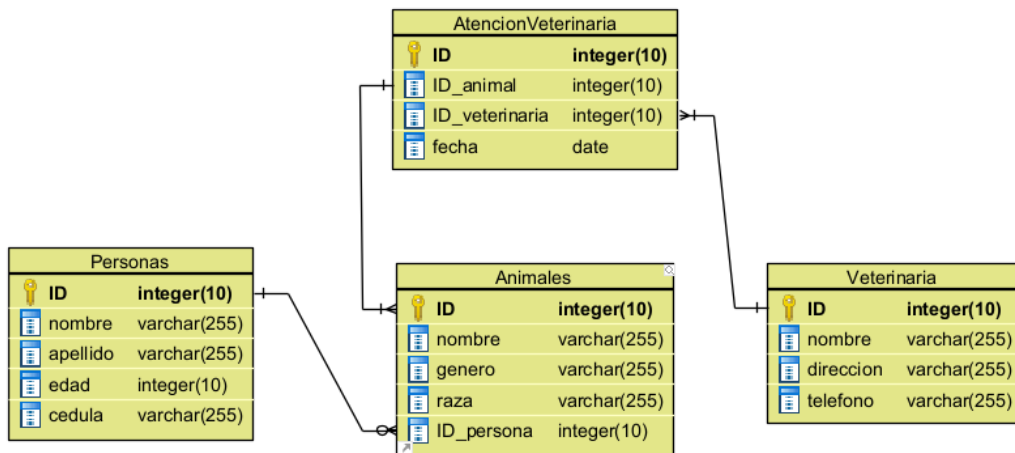
```
class Veterinaria {  
    constructor(nombre, direccion, telefono) {  
        this.nombre = nombre;  
        this.direccion = direccion;  
        this.telefono = telefono;  
    }  
  
    getNombre() {  
        return this.nombre;  
    }  
  
    setNombre(nombre) {  
        this.nombre = nombre;  
    }  
  
    getDireccion() {  
        return this.direccion;  
    }  
  
    setDireccion(direccion) {  
        this.direccion = direccion;  
    }  
  
    getTelefono() {  
        return this.telefono;  
    }  
  
    setTelefono(telefono) {  
        this.telefono = telefono;  
    }  
}
```



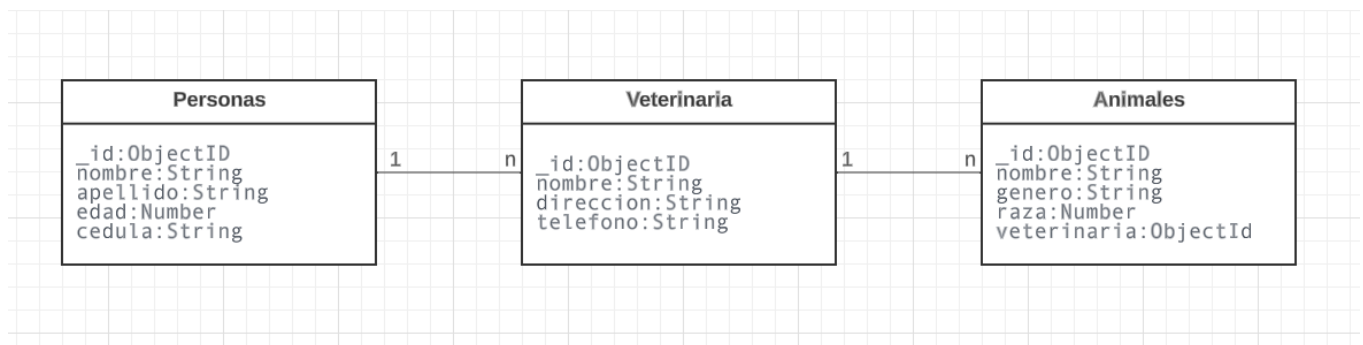
## Diagrama de base de datos relacional

Se asume que un animal puede estar asociado a una sola persona, pero una persona puede tener varios animales y una veterinaria puede atender a varios animales y personas. En consecuencia, se crearán las siguientes tablas:

La tabla de relación "AtenciónVeterinaria" se crea para representar la relación entre las tablas "Animales" y "Veterinaria", ya que un animal puede ser atendido por una Veterinaria y una Veterinaria puede atender a varios animales.



## Diagrama de base de datos no relacional



## Creación de una BD No relacional con Mongo DB:

```
use veterinaria

'switched to db veterinaria'
db.createCollection('Personas')

{ ok: 1 }

db.Personas.insert({
  nombre: "Arley",
  apellido: "Fuentes",
  edad: 33,
  cedula: "1038439006"
})

'DeprecationWarning: Collection.insert() is deprecated. Use insertOne,
insertMany, or bulkWrite.'

{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64504adcbde58b9e029bbac5")
  }
}

db.createCollection('Veterinaria')

{ ok: 1 }

db.Veterinaria.insert({
  nombre: "VeterinaFa",
  direccion: "Calle 28 #33-31",
  telefono: "3117409297"
})

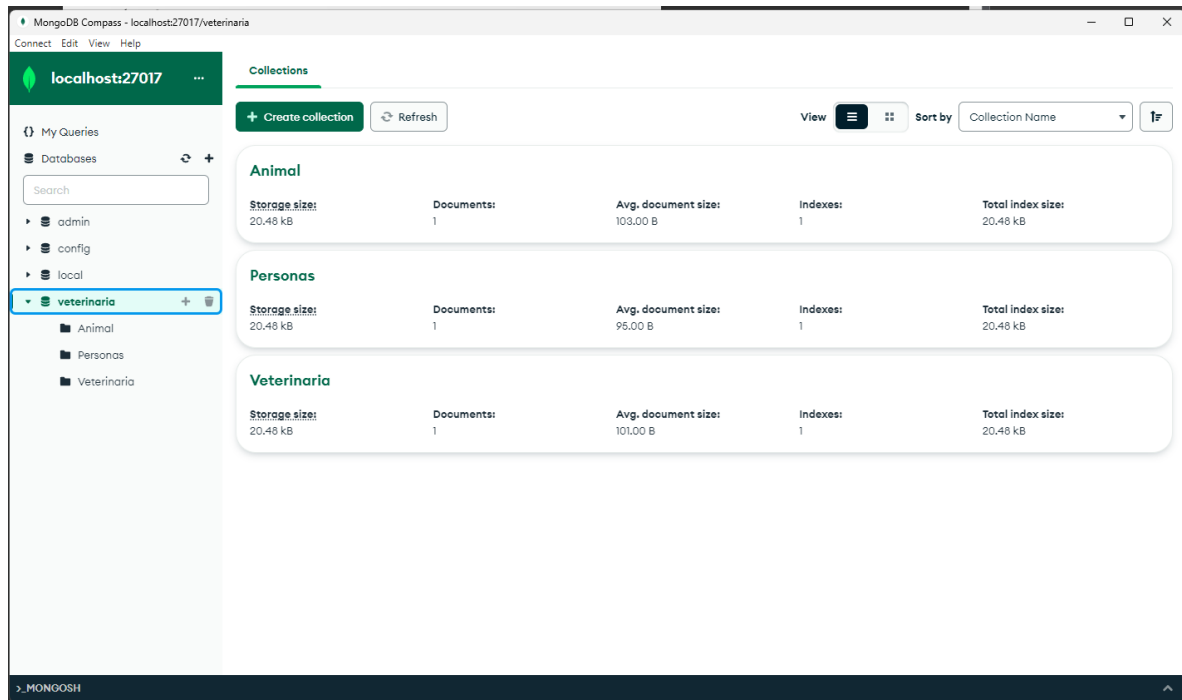
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64504b31bde58b9e029bbac6")
  }
}

db.createCollection('Animal')

{ ok: 1 }

db.Animal.insert({
  nombre: "Putin",
  genero: "Macho",
  raza: "Azul Ruso",
  veterinaria: ObjectId("64504b31bde58b9e029bbac6")
})
```

## Capturas MongoDB



MongoDB Compass - localhost:27017/veterinaria

Connect Edit View Help

localhost:27017

My Queries

Databases

Search

admin

config

local

**veterinaria**

Animal

Personas

Veterinaria

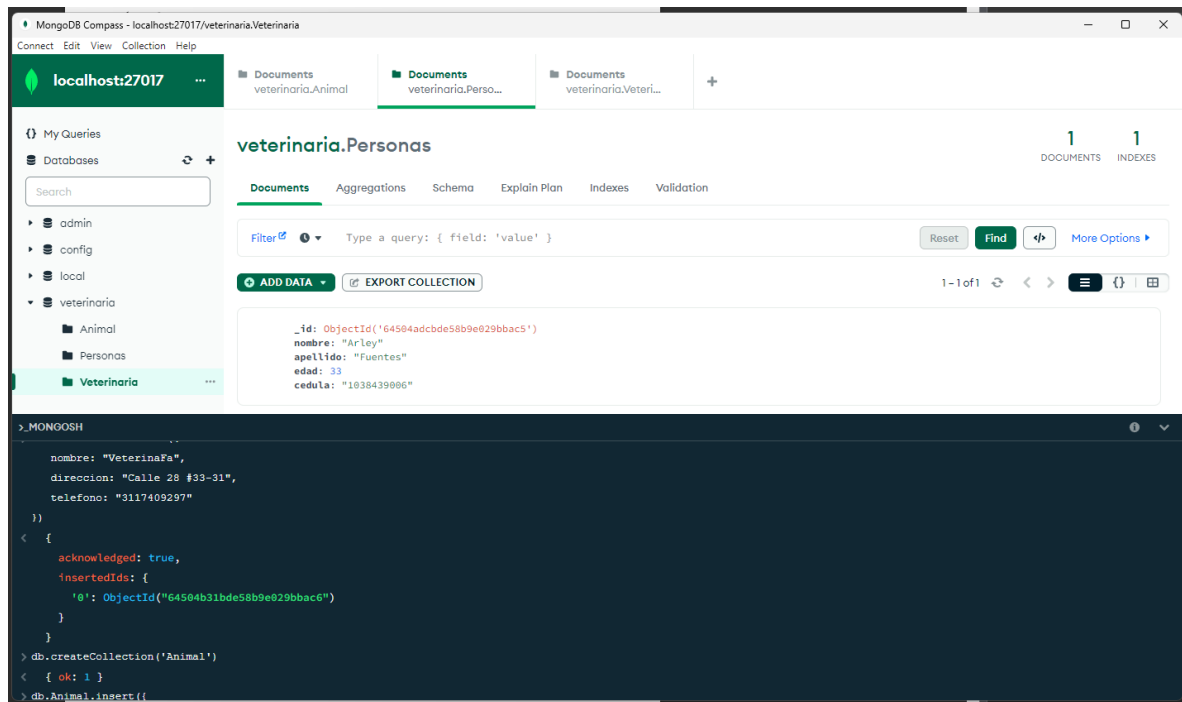
Collections

+ Create collection Refresh

View Sort by Collection Name

Collection	Storage size	Documents	Avg. document size	Indexes	Total index size
Animal	20.48 kB	1	103.00 B	1	20.48 kB
Personas	20.48 kB	1	95.00 B	1	20.48 kB
Veterinaria	20.48 kB	1	101.00 B	1	20.48 kB

> MONGODB



MongoDB Compass - localhost:27017/veterinaria.Veterinaria

Connect Edit View Collection Help

localhost:27017

My Queries

Databases

Search

admin

config

local

veterinaria

Animal

Personas

**Veterinaria**

Documents

Documents

Documents

+

veterinaria.Personas

1 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Explain Plan Indexes Validation

Filter Type a query: { field: 'value' } Reset Find More Options

ADD DATA EXPORT COLLECTION

1 - 1 of 1

```
{
  "_id": ObjectId("64504adcbe58b9e029bbac5"),
  "nombre": "Arley",
  "apellido": "Fuentes",
  "edad": 33,
  "cedula": "1838439086"
}
```

> MONGODB

```
nombre: "VeterinaFa",
direccion: "Calle 28 #33-31",
telefono: "3117409297"
})
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("64504b31bde58b9e029bbac6")
  }
}
> db.createCollection('Animal')
< { ok: 1 }
> db.Animal.insert({
```