



JAVA™
AVA™

EXAMEN DEL CURSO

Instancia Final

Contenido	
1. Introducción: Contexto	¡Error! Marcador no definido.
2. Ejercicio 1: Implementar desde el diagrama de clases	2
3. Ejercicio 2: Implementar un método a partir de un enunciado.....	5
4. Ejercicio 3: Interpretación de código	5
5. Ejercicio 4: Interpretación de Diagrama de Entidades y Relaciones (DER).....	6
6. Ejercicio 5: Consultas SQL	7

1. Introducción: Contexto.

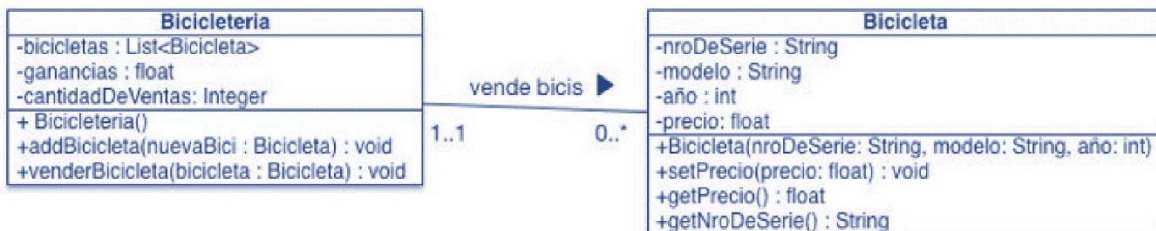
Ludmila tiene un negocio especializado en la venta de bicicletas. Hasta el momento registra todas sus operaciones en un cuaderno. Sin embargo, dado el éxito de su empresa, cada día le es más difícil poder tener control sobre las ventas de sus bicicletas y las ganancias que registra. Por esta razón, Ludmila necesita desarrollar un sistema que la ayude en su negocio.

Como sabe que nos capacitamos en el curso en Programación Java se puso en contacto con nosotros para que la ayudemos a construir el sistema, que en esta primera etapa del proyecto deberá registrar las bicicletas en venta y las ventas registradas.

2. Ejercicio 1: Implementar desde el diagrama de clases. 20%

Dado que Marisa es experta en modelado de software, ha creado un diagrama de clases UML para indicarnos lo que desea Ludmila. Otros programadores comenzaron con la codificación, pero nos piden que completemos el código para:

- Declarar las variables bicicletas, ganancias y cantidadDeVentas de la clase Bicicleteria.
- Implementar el método addBicicleta de la clase Bicicleteria. El mismo deberá agregar una nueva bicicleta en la lista bicicletas.
- Implementar los métodos getPrecio y setPrecio de la clase Bicicleta.




```
public class Bicicleteria {

    //a. DECLARAR LAS VARIABLES bicicletas, ganancias y cantidadDeVentas

    public Bicicleteria() {
        bicicletas = new ArrayList<Bicicleta>();
        cantidadDeVentas = 0;
        ganancias = 0;
    }
    public void venderBicicleta(Bicicleta bicicleta){
        bicicletas.remove(bicicleta);
        cantidadDeVentas++;
        ganancias = ganancias + bicicleta.getPrecio();
    }

    //b. IMPLEMENTAR EL METODO addBicicleta

}

public class Bicicleta {
    private String nroDeSerie;
    private String modelo;
    private int año;
    private float precio;
    public Bicicleta(String nroDeSerie, String modelo, int año) {
        this.nroDeSerie = nroDeSerie;
        this.modelo = modelo;
        this.año = año;
    }
    public String getNroDeSerie() {
        return nroDeSerie;
    }

    //c. IMPLEMENTAR LOS METODOS getPrecio y setPrecio

}
```

```

public void setPrecio(float precio) {
    this.precio = precio;
}

public String getNroSerie() {
    return nroSerie;
}

```

3. Ejercicio 2: Implementar un método a partir de un enunciado. 20%

Programar en Java el código del método buscarBicicleta en la clase Bicicleria. El mismo recibirá por parámetros una variable llamada nroDeSerie de tipo String y retornará la bicicleta con dicho nroDeSerie. En caso de no encontrarlo deberá retornar null.

```

public Bicicleta bucarBiciocleta(String nroSerie){
    for (Bicicleta bicicleta : bicicletas) {
        if(bicicleta.getNroSerie().equals(nroSerie))
            return bicicleta;
    }
    return null;
}

```

4. Ejercicio 3: Interpretación de código. 20%

Un desarrollador implementó el siguiente método en la clase Bicicleria pero no usó nombres representativos. Indique cuál de las siguientes opciones explica lo que hace el código:

```

public float xxxx(){
    float ret = 0;
    for (int i = 0; i < bicicletas.size(); i++) {
        ret = ret + bicicletas.get(i).getPrecio();
    }
    return ret;
}

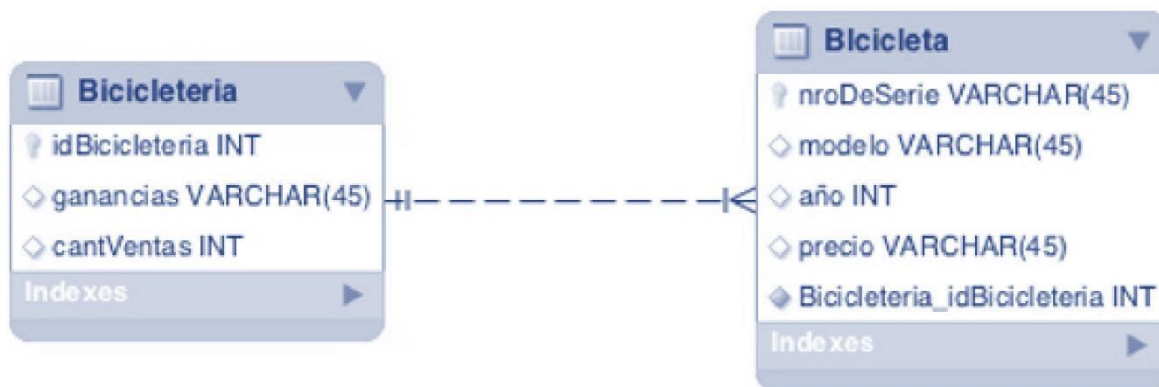
```

- a) Calcula el total de bicicletas de la bicicletería.
- b) Calcula el precio total de las bicicletas de la bicicletería.
- c) Calcula el promedio del precio de las bicicletas de la bicicletería.
- d) Ninguna de las anteriores.

Respuesta: (b) Calcula el precio total de las bicicletas de la bicicletería.

5. Ejercicio 4: Interpretación de Diagrama de Entidades y Relaciones (DER). 20%

A partir del DER, responda Verdadero (V) o Falso (F) a las siguientes afirmaciones (asignar a cada afirmación un valor V o F). Todas las afirmaciones tienen que tener un valor asignado. Puede haber varios V y varios F:



AFIRMACIÓN	VERDADERO / FALSO
a) Entre las tablas Bicicleteria y Bicicleta existe una relación 1 a 1.	f
b) Entre las tablas Bicicleteria y Bicicleta existe una relación 1 a N.	v
c) Entre las tablas Bicicleteria y Bicicleta existe una relación N a N.	f
d) La clave primaria de la tabla Bicicleta es nroDeSerie.	v
e) La clave foránea de la tabla Bicicleta es nroDeSerie.	f
f) La tabla Bicicleteria no tiene clave primaria.	f
g) La tabla Bicicleteria no tiene clave foránea.	v

6. Ejercicio 5: Consultas SQL. 20%

Dado el diagrama de entidad-relación presentado en el ejercicio anterior, escriba una consulta SQL que liste: la cantidad de ventas, ganancias y la cantidad de bicicletas en venta, de la bicicletería cuyo idBicicleteria = 2.

Entidad-Relación: Ejercicio 5

```
SELECT cantVentas, ganancias, COUNT(bic.nroDeSerie) AS "cantBici Venta" FROM bicicleterias
INNER JOIN bicicletas bic ON bic.bicicleteriaId = bicicleterias.id
WHERE bicicleterias.id = 2;
```

¡Éxitos!