

# Programación

---



Proveedores  
Ejercicio 8  
Relación 1

# Proveedores.

## Enunciado del Problema

- ☐ Una Empresa desea mantener la información de sus proveedores por lo que nos solicita la realización de un programa. Se nos pide:
  - Cada proveedor debe quedar identificado por un número único de 5 cifras (desde el 10.000 en adelante - la cantidad de proveedores es mucho menor que 10.000)
  - Se desea mantener la siguiente información:
    - ☐ Nombre del proveedor
    - ☐ Dirección
    - ☐ Teléfono
- ☐ Debe ser posible saber el total de lo que se le ha comprado a cada proveedor desde el inicio de la relación comercial con el mismo.

# Proveedores.

## Enunciado del Problema

- Se pide
  - Crear una clase proveedor que permita cumplir con lo anterior.
    - Al momento de ingresar un nuevo proveedor se conoce en algunos casos su nombre y en otros se tiene además los datos de dirección y teléfono.
    - El número de identificación y el nombre no deben poder modificarse una vez creado un nuevo proveedor.
  - El siguiente código pertenece a otra clase que utiliza la clase Proveedor.

```
Proveedor p = new Proveedor("La Divina Tuerca","Esclava del Señor",  
                             "954059322");  
System.out.print(p);
```

- y provoca la siguiente salida a pantalla:

```
Proveedor Nº 13425 - La Divina Tuerca  
Esclava del Señor  
Tel. 954059322
```

- Agregue en Proveedor el método que falta para que esto sea posible

# Clase **Proveedor**.

## Resolución del Problema

---

- Esta clase describe a un **único** proveedor, por lo tanto tenemos que pensar en:
  - Los atributos
  - Los métodos constructores
  - Demás métodos que tiene un proveedor

# Clase **Proveedor**.

## Resolución del Problema

---

### □ Atributos

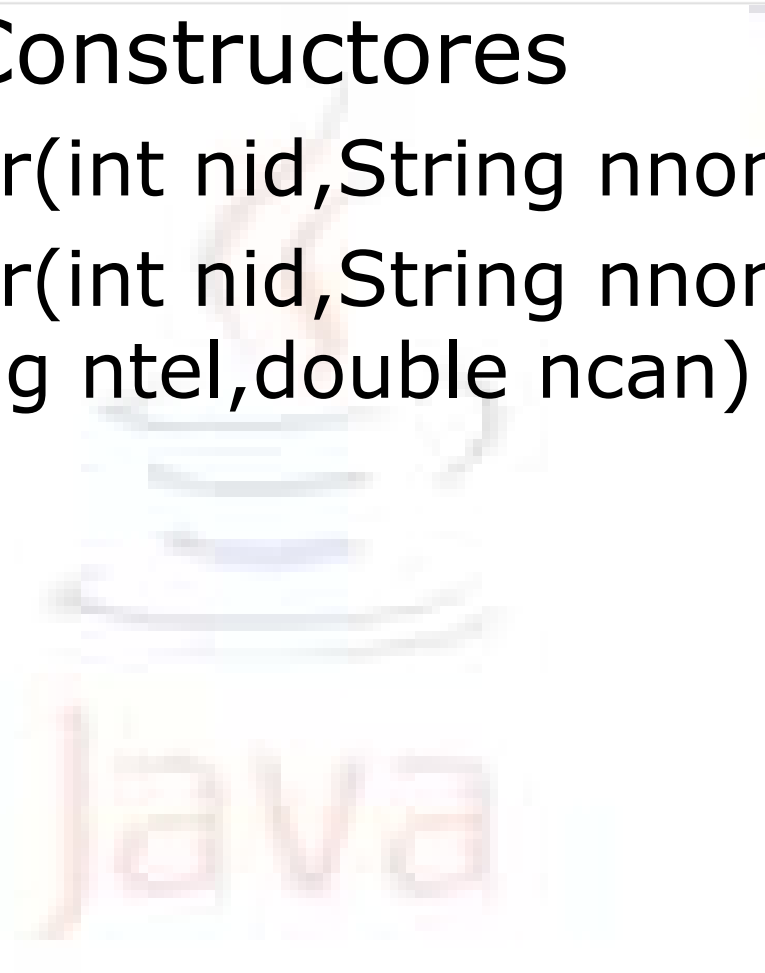
- `private int id;`
- `private String nombre;`
- `private String direccion;`
- `private String telefono;`
- `private double cantidad;`

# Clase **Proveedor**.

## Resolución del Problema

---

- Métodos Constructores
  - Proveedor(int nid,String nnom)
  - Proveedor(int nid,String nnom,String ndir,String ntel,double ncan)



# Clase **Proveedor**.

## Resolución del Problema

---

### □ Métodos

- `public void Modificar(String ndir,String ntel,double ncan)`
- `public int getID()`
- `public String getNOMBRE()`
- `public void setDIRECCION(String ndir)`
- `public String getDIRECCION()`

# Clase **Proveedor**.

## Resolución del Problema

---

### □ Métodos

- `public void setTELEFONO(String ntel)`
- `public String getTELEFONO()`
- `public void setCANTIDAD(double ncan)`
- `public double getCANTIDAD()`
- `public void Indicar_Compra(double ncan)`
- `public String toString()`



# Clase Principal **Proveedores.**

## Resolución del Problema

- Ahora tenemos que pensar que esta clase representa una lista de proveedores. Por lo tanto los atributos y métodos que pongamos estarán relacionado con esta lista.

- Atributos

- Vector proveedor;

- Método Constructor

```
Proveedores()  
{  
    proveedor=new Vector();  
}
```

# Clase Principal **Proveedores.**

## Resolución del Problema

---

- ❑ Otros métodos
  - `public void Añadir_Proveedor()`
  - `public void Borrar_Proveedor()`
  - `public void Modificar_Proveedor()`
  - `public void Incrementar_Cantidad()`
  - `public void Calcular_Total()`
  - `public void Listar_Proveedor()`
  - `public void Menu()`

# Clase Principal **Proveedores**.

## Resolución del Problema

---

### □ Método Main

- En el programa principal vamos a usar un objeto de la clase Proveedores.

```
Proveedores lista_proveedores =  
                                new Proveedores();
```

- Después simplemente llamaremos al método menu()