

# Enunciado

➤ Realizar una nueva versión del programa para la gestión de una agenda de contactos. La funcionalidad será la misma que la del anterior, pero en este caso, los contactos se almacenarán en un fichero de texto, cada contacto en una línea siguiendo el formato:

nombre|edad|email



# Lección 3 I. Ejercicio práctico 12



# Enunciado

➤ **Realizar un programa que muestre un menú con las siguientes opciones:**

- 1.- Anadir contacto
- 2.- Buscar contacto
- 3.- Mostrar todos
- 4.- Salir

▪ **Opción 1. Si hay espacio libre (se podrán guardar un máximo de 10 contactos), se solicitará al usuario la introducción de los datos del contacto (nombre, edad e email) y el contacto quedará guardado.**

▪ **Opción 2. Se solicitará el nombre del contacto y si se encuentra, se mostrarán sus datos. Si hay más de un contacto con ese nombre se mostrará el primero que se encuentre.**

▪ **Opción 3. Se mostrarán los datos de todos los contactos almacenados.**

# Lección 30. Javabeans



# Características

- Clase que encapsula un conjunto de datos asociados a una entidad, como una persona, empleado, libro,...
- Permiten tener los datos agrupados en un único objeto, en lugar de que estén separados

nombre

email

telefono

objeto

Persona

A blue rounded rectangle representing an object. It is divided into three vertical sections by two black lines. The sections are labeled 'nombre', 'email', and 'telefono' from left to right.

nombre

email

telefono



# Estructura

```
public class Persona{
    //atributos
    private String nombre;
    private String email;
    private int telefono;
    //constructores
    public Persona(){..}
    public Persona(String nombre,String email,int telefono){
        this.nombre=nombre;
        this.email=email;
        this.telefono=telefono;
    }
    //setter y getter
    public void setNombre(String nombre){
        this.nombre=nombre;
    }
    public String getNombre(){
        return nombre;
    }

    public void setEmail(String email){
        this.email = email;
    }
    :
}
```

# Setter y getter

➤ **Métodos que dan acceso externo a los atributos para escritura (setter) y lectura (getter)**

➤ **Convenio:**

- **Escritura: setNombreAtributo**

- **Lectura: getNombreAtributo**

# Utilización

- Se emplean en aquellos contextos donde se necesite tener agrupados los datos asociados a una entidad.
- A nivel de código se utiliza como una clase normal. Se crea el objeto, se asignan los valores a través del constructor y/o métodos set, y se recuperan con get:

```
Persona p=new Persona("Jose",jose@gmail.com,33);  
System.out.println("Te llamas "+p.getNombre());
```

- Pueden almacenarse en arrays:

```
Persona [] pers=new Persona[5];  
pers[0]=p;
```