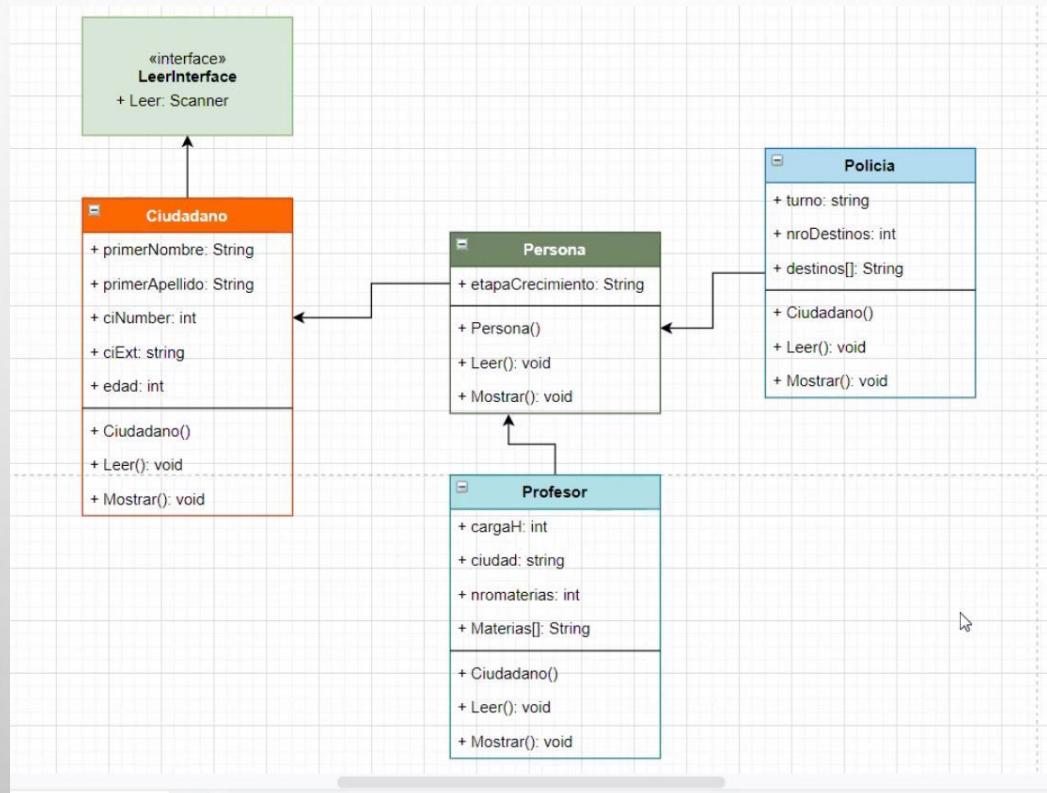


RESOLUCIÓN HITO 2

DIEGO EMILIANO RIVERA TAPIA

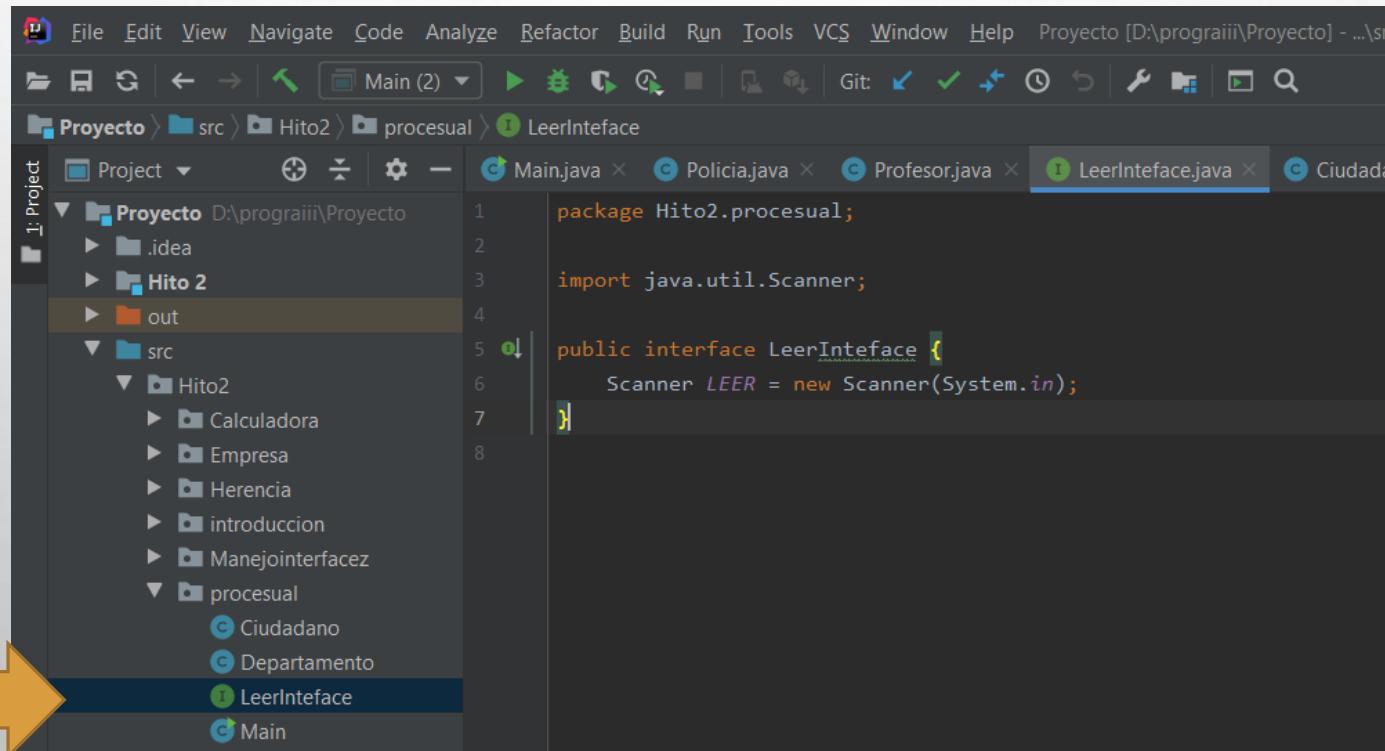
PREGUNTA 1:

- SE TIENE QUE CREAR EL SIGUIENTE DIAGRAMA:



INTERFACE

- SE EMPIEZA CREANDO LA INTERFACE DONDE ÚNICAMENTE EXISTA LA VARIABLE LEER:



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help Proyecto [D:\prograii\Proyecto] - ...\\src\\Hito2\\procesual\\LeerInterface.java
```

```
package Hito2.procesual;

import java.util.Scanner;

public interface LeerInterface {
    Scanner LEER = new Scanner(System.in);
}
```

The screenshot shows a Java code editor in an IDE. The project structure on the left includes a .idea folder, a Hito 2 folder, and a src folder containing several packages: Hito2, Calculadora, Empresa, Herencia, introduccion, Manejointerfacez, and procesual. Inside the procesual package, there are four files: Ciudadano, Departamento, LeerInterface (which is currently selected and highlighted in blue), and Main. The main window displays the code for the LeerInterface.java file. An orange arrow at the bottom left points to the word "Interface" in the code, indicating the focus of the slide.

CIUDADANO:

- SE CREA LA CLASE CIUDADANO LA CUAL CONTENDRÁ LAS VARIABLES BÁSICAS DE UNA PERSONA MAS EL CONSTRUCTOR CIUDADANO, VOID LEER Y MOSTRAR.

The screenshot shows an IDE interface with the following details:

- Project Structure:** The project is named "Proyecto" and contains a package "Hito2.procesual". Inside "Hito2.procesual", there are several files: Main.java, Policia.java, Profesor.java, LeerInterface.java, Ciudadano.java, Departamento.java, Persona.java, and Policia.java. The "Ciudadano.java" file is currently selected.
- Code Editor:** The code for the "Ciudadano" class is displayed:

```
package Hito2.procesual;
public class Ciudadano implements LeerInterface {
    private String primerNombre;
    private String primerApellido;
    private int ciNumber;
    private String ciExt;
    private int edad;
    public Ciudadano() {
        this.primerNombre = "";
        this.primerApellido = "";
        this.ciNumber = 0;
        this.ciExt = "";
        this.edad = 0;
    }
    public void Leer() {
    }
}
```
- Annotations:**
 - A yellow box labeled "Clase" with an arrow points to the "Ciudadano" class in the project tree.
 - A yellow box labeled "Se implementa la interface" with a bracket above the "implements" keyword indicates that the class implements the "LeerInterface".
 - A yellow box labeled "Variables" with a bracket on the right side of the class body groups the five instance variables: primerNombre, primerApellido, ciNumber, ciExt, and edad.
 - A yellow box labeled "Constructor" with a bracket on the right side of the constructor groups the code within the constructor's braces.

CIUDADANO: VOID LEER Y MOSTRAR

```
14         this.ciExt = "";
15         this.edad = 0;
16     }
17
18     public void Leer(){
19         System.out.printf("Ingresar Primer nombre: ");
20         this.primerNombre = LEER.next();
21         System.out.printf("Ingresar Primer apellido: ");
22         this.primerApellido = LEER.next();
23         System.out.printf("Ingresar Numero CI: ");
24         this.ciNumber = LEER.nextInt();
25         System.out.printf("Ingresar Extension CI: ");
26         this.ciExt = LEER.next();
27         System.out.printf("Ingresar Edad: ");
28         this.edad = LEER.nextInt();
29
30     }
31
32     public void Mostrar(){
33         System.out.printf("Nombre : %s, Apellido : %s, Numero CI: %d, Extension CI: %s, Edad: %d %s",
34             this.primerNombre, this.primerApellido, this.ciNumber, this.ciExt, this.edad, "\n");
35     }
36
37
38 }
```

Void Leer

Void Mostrar

PERSONA:

- LA CLASE PERSONA HEREDARA LA INFORMACIÓN BÁSICA DE CIUDADANO TENIENDO SU PROPIO CONSTRUCTOR, VVOIDS Y VARIABLE.

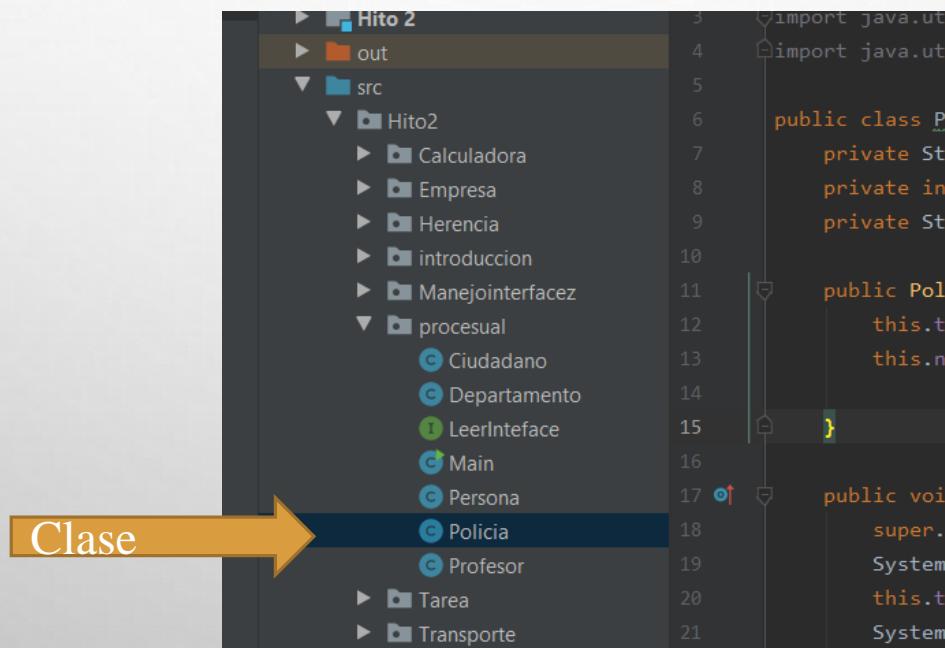
The screenshot shows an IDE interface with the following details:

- Project:** D:\prograiii\Proyecto
- File:** Person.java
- Code Content:**

```
1 package Hito2.procesual;
2
3 public class Persona extends Ciudadano {
4     private String etapaCrecimiento;
5
6     public Persona() {
7         this.etapaCrecimiento = "";
8     }
9
10    public void Leer(){
11        super.Leer();
12        System.out.printf("Ingrese etapa de crecimiento: ");
13        this.etapaCrecimiento = LEER.next();
14    }
15
16    public void Mostrar(){
17        super.Mostrar();
18        System.out.printf("etapa crecimiento : %s %s", this.etapaCrecimiento, "\n");
19    }
20}
```
- Annotations:**
 - A yellow box labeled "Constructor y variable" points to the constructor and the private variable declaration.
 - A yellow box labeled "Hereda de ciudadano" points to the extends Ciudadano line.
 - A yellow box labeled "Clase" has an arrow pointing to the class definition.
 - A yellow box labeled "Voids Leer y Mostrar" points to the two void methods.

POLICÍA:

- SE CREA UNA CLASE POLICÍA LA CUAL TENDRÁ 3 VARIABLES DONDE LA VARIABLE DESTINOS TIENE QUE SER UN ARRAY ADICIONALES Y HEREDARA DE PERSONA, TENDRÁ UN CONSTRUCTOR POLICÍA Y LOS VVOIDS LEER Y MOSTRAR.



The image shows a screenshot of an IDE (Integrated Development Environment) displaying a Java project named "Hito 2". The project structure is as follows:

- Project: Hito 2
- Folder: out
- Folder: src
 - Folder: Hito2
 - File: Calculadora
 - File: Empresa
 - File: Herencia
 - File: introduccion
 - File: Manejointerfacez
 - Folder: procesual
 - File: Ciudadano
 - File: Departamento
 - File: LeerInteface
 - File: Main
 - File: Persona
 - File: Policia
 - File: Profesor
 - File: Tarea
 - File: Transporte

The code editor on the right contains the following Java code:

```
import java.util.*;  
import java.util.ArrayList;  
  
public class Policia extends Persona {  
    private String nombre;  
    private int edad;  
    private String[] destinos;  
  
    public Policia(String nombre, int edad, String[] destinos) {  
        super(nombre, edad);  
        this.nombre = nombre;  
        this.edad = edad;  
        this.destinos = destinos;  
    }  
  
    public void leer() {  
        super.leer();  
        System.out.println("Nombre: " + nombre);  
        System.out.println("Edad: " + edad);  
        System.out.println("Destinos: " + Arrays.toString(destinos));  
    }  
  
    public void mostrar() {  
        super.mostrar();  
        System.out.println("Nombre: " + nombre);  
        System.out.println("Edad: " + edad);  
        System.out.println("Destinos: " + Arrays.toString(destinos));  
    }  
}
```

A yellow arrow points to the word "Clase" (Class) in the code editor, highlighting the class definition.

POLICÍA: CONSTRUCTOR, HERENCIA Y VARIABLES

```
1 package Hito2.procesual;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class Policia extends Persona {
7     private String turno;
8     private int nroDestinos;
9     private String[] destinos;
10
11    public Policia() {
12        this.turno = "";
13        this.nroDestinos = 0;
14    }
15
16 }
```

Hereda de persona

Variables

Constructor

POLICÍA: VOIDS Y GETDESTINOS

```
↑ ↴    public void Leer(){
      ↴    super.Leer();
      ↴    System.out.printf("Ingrese turno: ");
      ↴    this.turno = LEER.nextInt();
      ↴    System.out.printf("Ingrese nro Destinos: ");
      ↴    this.nroDestinos = LEER.nextInt();
      ↴    this.destinos = new String[this.nroDestinos];
      ↴
      ↴    for(int i=0; i<this.nroDestinos; i++){
      ↴        System.out.printf("Ingrese destino: ");
      ↴        this.destinos[i] = LEER.next();
      ↴        System.out.println();
      ↴    }
      ↴}
      ↴
      ↴    public void Mostrar(){
      ↴        super.Mostrar();
      ↴        System.out.printf("Turno: %s,Nro Destinos: %d %s", this.turno,this.nroDestinos, "\n");
      ↴        for(int j=0; j<this.nroDestinos; j++){
      ↴            System.out.printf("Destino: %d: %s",j,this.destinos[j]);//Destinos: 0 Cochabamba
      ↴        }
      ↴    }
      ↴    public String[] getDestinos(){
      ↴        return this.destinos;
      ↴    }
      ↴}
```

Voids Leer
y Mostrar

Obtener el array
Destinos

PROFESOR:

- LA CLASE PROFESOR HEREDA DE PERSONA TIENE SUS PROPIAS VARIABLES, CONSTRUCTOR, VODS Y VARIABLES DONDE LA VARIABLE MATERIAS ES UN ARRAY.

Clase

Hereda de persona

Variables

Constructor

```
import java.util.Arrays;  
public class Profesor extends Persona {  
    private int cargaH;  
    private String ciudad;  
    private int nroMaterias;  
    private String[] MATERIAS;  
  
    public Profesor() {  
        this.cargaH = 0;  
        this.ciudad = "";  
        this.nroMaterias = 0;  
    }  
  
    public void Leer(){  
        super.Leer();  
        System.out.printf("CargaH: ");  
        this.cargaH = LEER.nextInt();  
        System.out.printf("Ciudad: ");  
        this.ciudad = LEER.next();  
        System.out.printf("Nro de MATERIAS: ");  
        this.nroMaterias = LEER.nextInt();  
        this.MATERIAS = new String[this.nroMaterias];  
  
        for(int i=0; i<this.nroMaterias; i++){  
            System.out.printf("Ingrese MATERIA: ");  
            this.MATERIAS[i] = LEER.next();  
            System.out.println();  
        }  
    }  
}
```

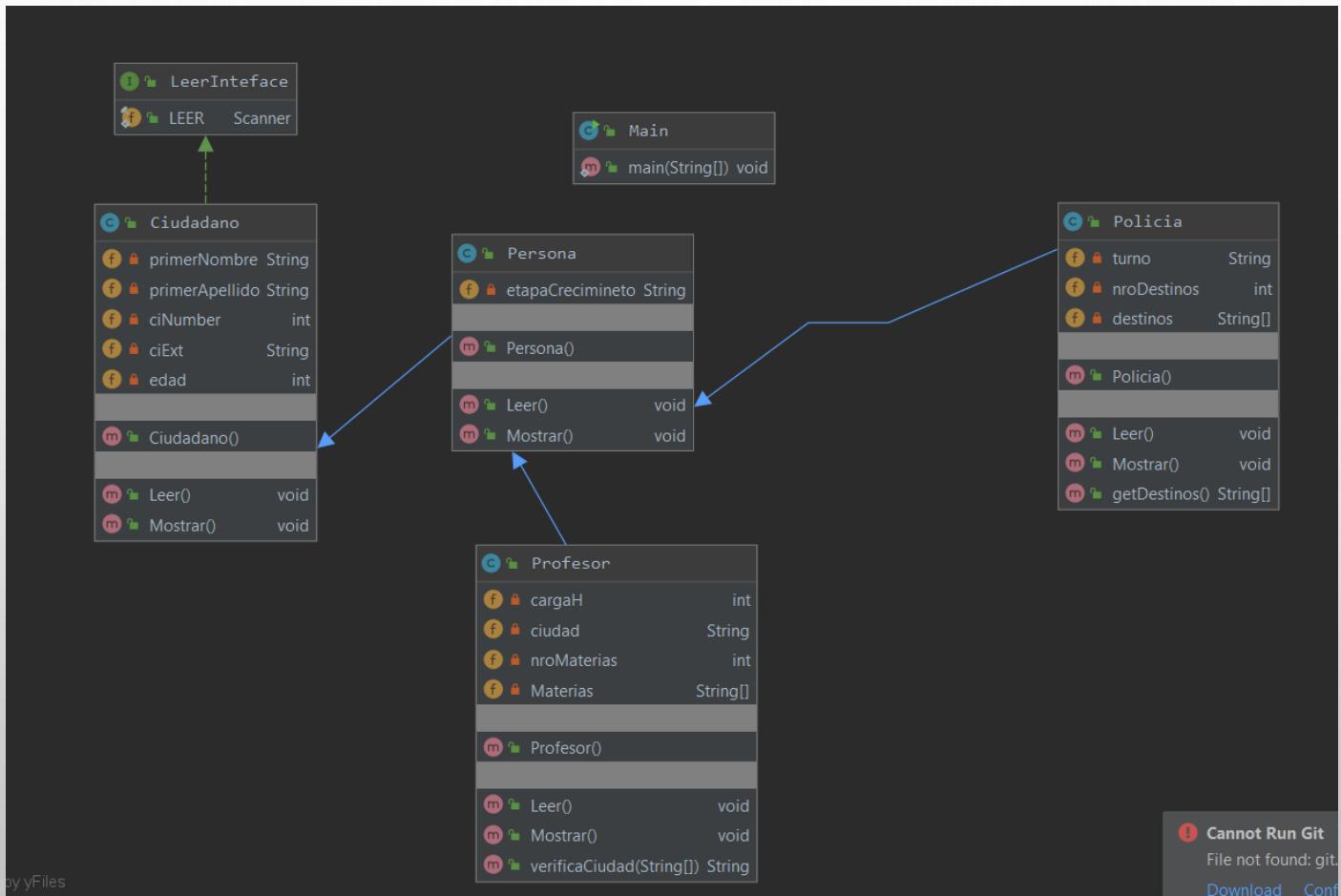
PROFESOR: VOID

```
17  
18  ⏷ public void Leer(){  
19      super.Leer();  
20      System.out.printf("CargaH: ");  
21      this.cargaH = LEER.nextInt();  
22      System.out.printf("Ciudad: ");  
23      this.ciudad = LEER.next();  
24      System.out.printf("Nro de materias: ");  
25      this.nroMaterias = LEER.nextInt();  
26      this.Materias = new String[this.nroMaterias];  
27  
28      for(int i=0; i<this.nroMaterias; i++){  
29          System.out.printf("Ingrese materia: ");  
30          this.Materias[i] = LEER.next();  
31          System.out.println();  
32      }  
33  }  
34  
35  ⏷ public void Mostrar(){  
36      super.Mostrar();  
37      System.out.printf("cargaH: %d,Ciudad: %s,Nro Destinos: %d %s", this.cargaH,this.ciudad,this.nroMaterias, "\n");  
38      for(int j=0; j<this.nroMaterias; j++){  
39          System.out.printf("Materia: %d: %s", (j+1),this.Materias[j]);  
40      }  
41  }  
42 }
```

Void: Leer

Void: Mostrar

DIAGRAMA:



PREGUNTA 2

- INSTANCIAR 2 OBJETOS DE LA CLASE POLICÍA Y DOS INSTANCIAS DE LA CLASE PROFESOR:

Main

```
3 import java.util.Scanner;
4 import java.util.function.Predicate;
5
6 public class Main {
7     public static void main(String[] args) {
8         Policia po1 = new Policia();
9         Policia po2 = new Policia();
10        po1.Leer();
11        po2.Leer();
12
13        Profesor pf1 = new Profesor();
14        Profesor pf2 = new Profesor();
15        pf1.Leer();
16        pf2.Leer();
17        pf2.Mostrar();
18    }
19 }
20 }
```

Instanciar dos policías

Instanciar dos profesores

EJEMPLO DE LA EJECUCIÓN:

```
"C:\Program Files\JetBrains\IntelliJ IDEA 2019.3.2\jbr\bin\j
Ingrese Primer nombre: Diego
Ingrese Primer apellido: Rivera
Ingrese Numero CI: 5900277
Ingrese Extension CI: cbba
Ingrese Edad: 20
Ingrese etapa de crecimiento: 15
Ingrese turno: Tarde
Ingrese nro Destinos: 2
Ingrese destino: Cochabamba

Ingrese destino: La paz

Ingrese Primer nombre: Laura
Ingrese Primer apellido: Canedo
Ingrese Numero CI: 5900278
Ingrese Extension CI: cbba
Ingrese Edad: 22
Ingrese etapa de crecimiento: 20
Ingrese turno: Tarde
Ingrese nro Destinos: 1
Ingrese destino: La paz
```

```
Nombre : Diego, Apellido : Rivera, Numero CI: 5900277, Extension CI: cbba,Edad: 20
etapa crecimiento : 12
Turno: Tarde,Nro Destinos: 2
Destino: 0: CochabambaDestino: 1: La
Process finished with exit code 0
```

PREGUNTA 3:

- INSERTAR DOS PROFESORES Y VERIFICAR SI TIENEN ASIGNADO UNA MISMA MATERIA:

Main

Llamamos al método

```
11 Profesor pf2 = new Profesor();
12 pf1.Leer();
13 pf1.Mostrar();
14 pf2.Leer();
15 pf2.Mostrar();
System.out.println(pf1.verificaCiudad(po1.getDestinos()));
pf1.verificarMateria();

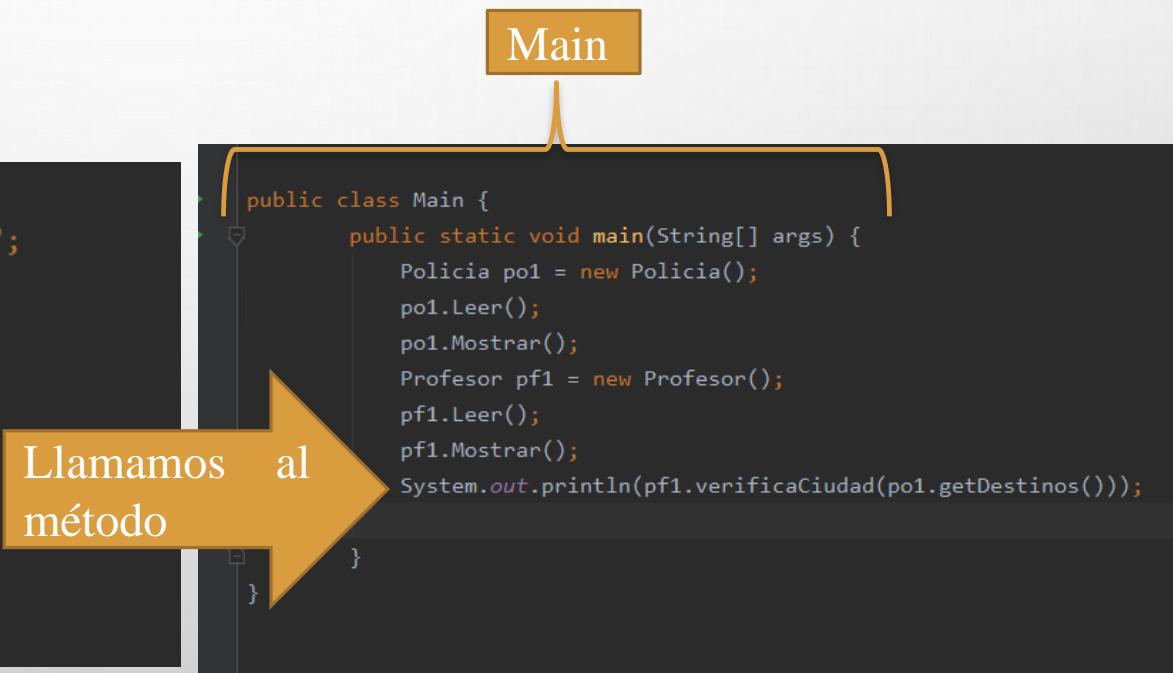
19 }
20 }
```

```
1 }
2
3 public void verificarMateria() {
4     for (int i = 0; i < Materias.length; i++) {
5         for (int j = 0; j < Materias.length; j++) {
6             if (Materias[i].equals(Materias[j])) {
7                 System.out.printf("Asignada misma materia");
8             } else {
9                 System.out.printf("No tienen la misma materia");
10            }
11        }
12    }
13 }
14
15 }
```

PREGUNTA 4:

- VERIFICAR SI LA CIUDAD DE UN PROFESOR PERTENECE A UNO DE LOS DESTINOS DE LA POLICÍA .

```
public String verificaCiudad(String[] destinos) {  
    String respuesta = "No trabajaron en la misma ciudad";  
  
    for (int i = 0; i < destinos.length; i++) {  
        if (this.ciudad.equals(destinos[i])) {  
            respuesta = "Si trabajaron en esa ciudad";  
        }  
    }  
    return respuesta;  
}
```



FIN