DEFENSA HITO 2

ESTRUCTURA DE DATOS

PRESENTA:
ILIA ARACELI SARZO LAURA
3do SEMESTRE INGENIERIA DE SISTEMAS

UNIVERSIDAD PRIVADA
UNIFRANZ
FRANZ TAMAYO

1¿A que se refiere cuando se habla de POO?

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos.

2¿Cuales son los 4 componen que componen POO?

SON LOS SIGUIENTES:
CLASE
PROPIDAD
METODOS
OBJETOS

3.-¿CUALES SON LOS PILARES DE POO?

1. Abstracción
2. Encapsulamiento: .
3. Herencia: .
4. Polimorfismo:

4 ¿Que es Encapsulamiento y muestre un ejemplo?

Es el proceso de almacenar en una misma sección los elementos de una abstracción que constituyen su estructura y su comportamiento; sirve para separar el interfaz contractual de una abstracción y su implantación.

```
public class MiClase{

    private int tipo;

        public void setTipo(int t) {

            tipo = t;

        }

    public int getTipo() {

        return tipo;

    }

}

class AccesoIndirecto {

    public static void main(String[] args) {

        MiClase mc = new MiClase();
        mc.setTipo(5);
        System.out.println("El tipo es:" + mc.getTipo());

    }

}
```

Content Here

You can simply impress your audience and add a unique zing.

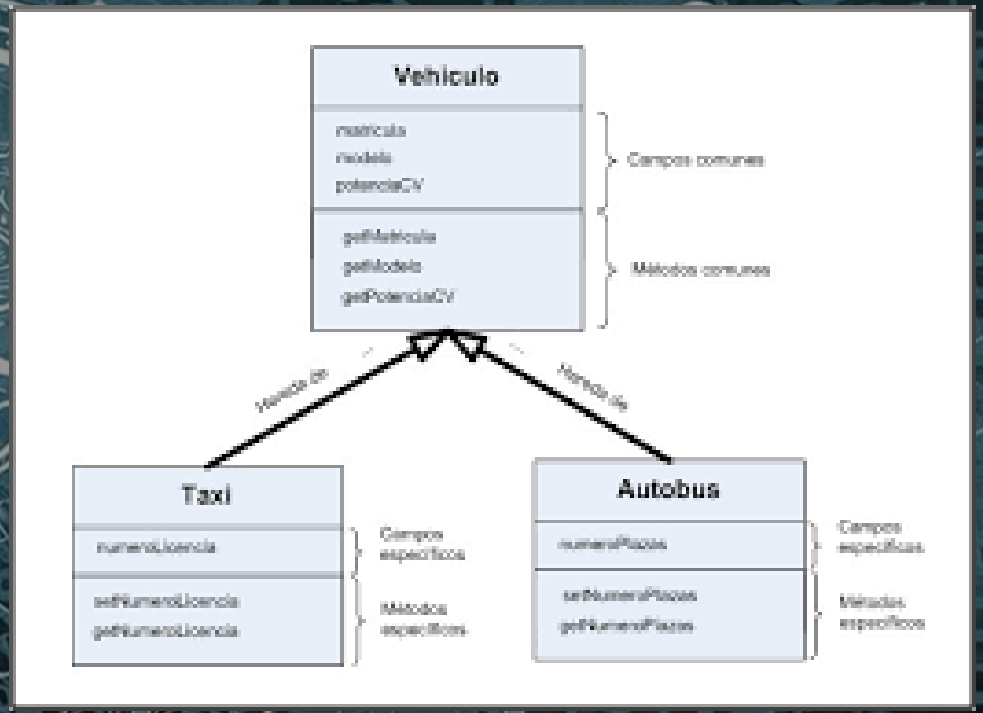
¿Que es Abstraction y muestra un ejemplo?

```java
public class UniversityStudent {
    int id;
    String name;
    String gender;
    String university;
    String career;
    int numSubjects;
    public UniversityStudent(int id, String name, String gender,
        String university, String  career, int numSubjects) {
        this.id = id;
        this.name = name;
        this.gender = gender;
        this.university = university;
        this.career = career;
        this.numSubjects = numSubjects;
    }
    void inscribeSubjects() {
      // TODO: implement
    }
    void cancelSubjects() {
       // TODO: implement
    }
    void consultRatings() {
       // TODO: implement
    }
}
```

La **abstracción** consiste en seleccionar datos de un conjunto más grande para mostrar solo los detalles relevantes del objeto. Ayuda a reducir la complejidad y el esfuerzo de **programación**. En Java, la **abstracción** se logra usando clases e interfaces abstractas. Es uno de los conceptos más importantes .

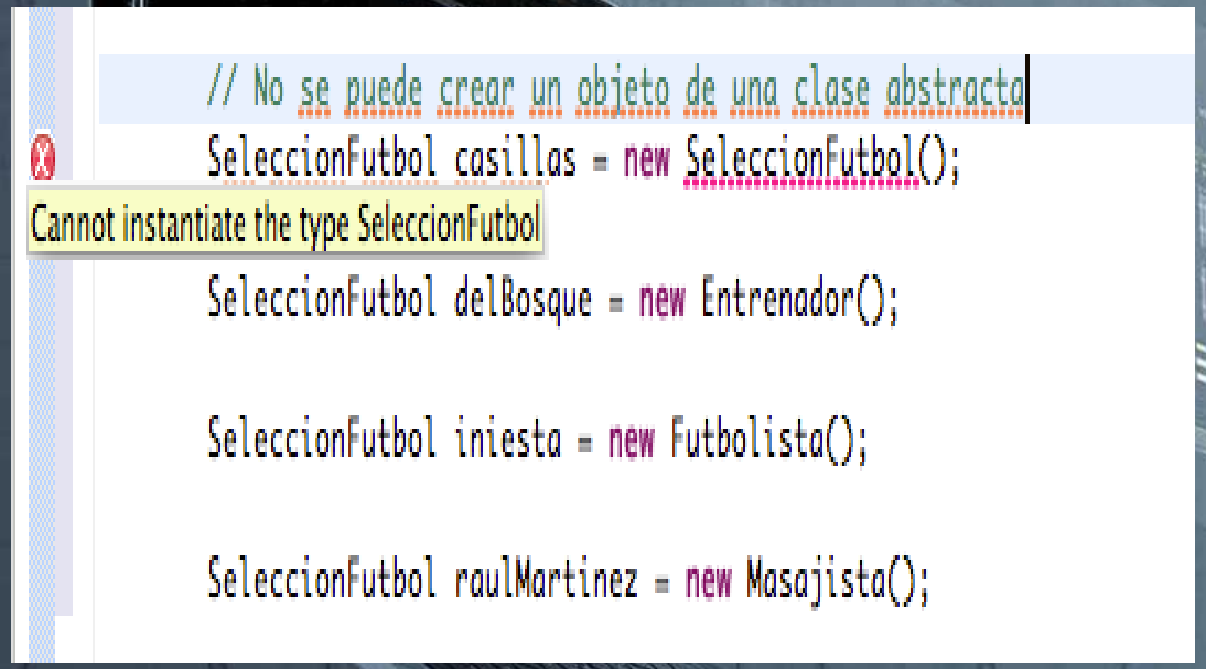
¿Que es Herencia y muestre un ejemplo?

La **herencia permite** que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.



Content Here

You can simply impress your audience and add a unique zing.

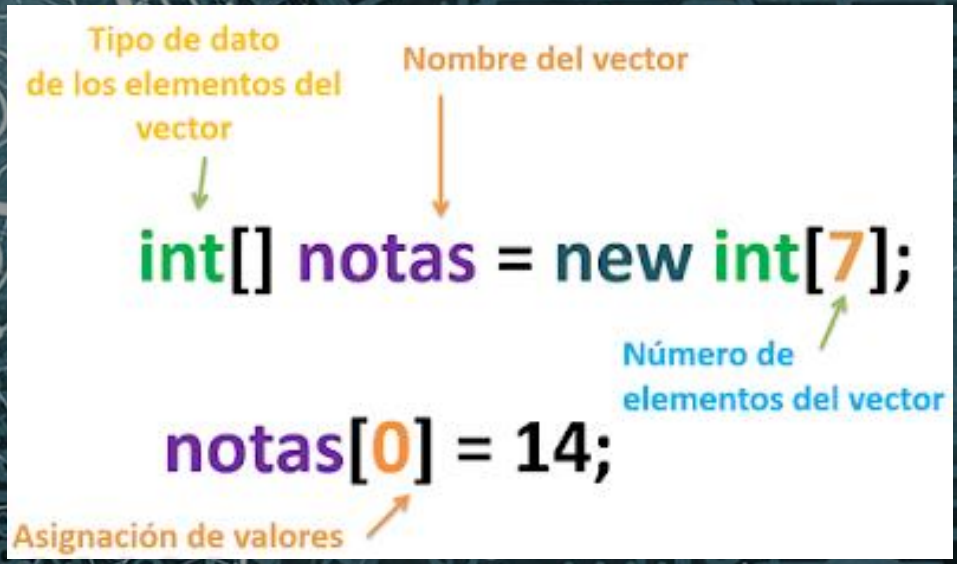


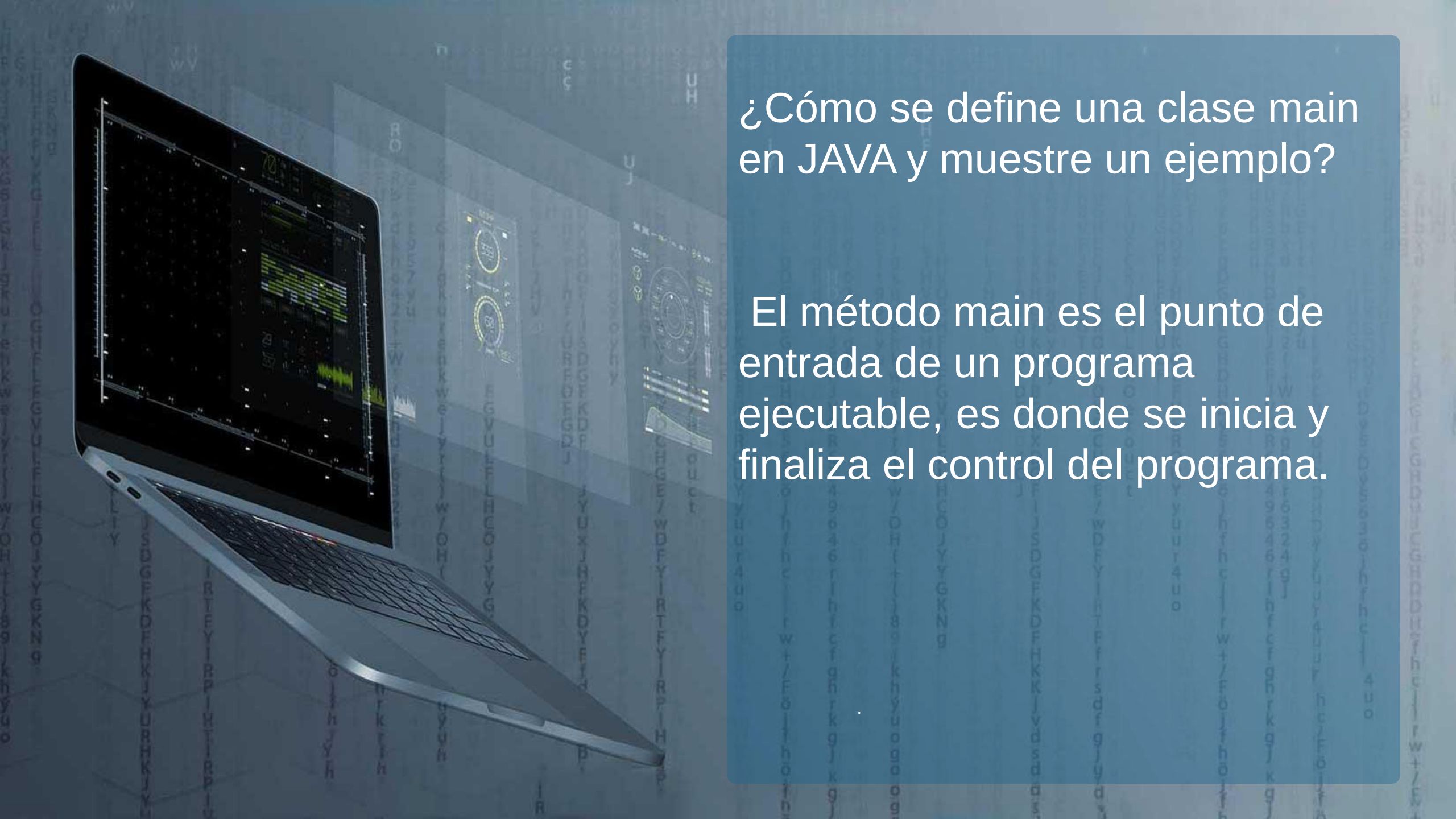
¿Que es Polimorfismo y muestra un ejemplo?

polimorfismo (en POO) es la capacidad que tienen ciertos lenguajes para hacer que, al enviar el mismo mensaje (o, en otras palabras, invocar al mismo método) desde distintos objetos, cada uno de esos objetos pueda responder a ese mensaje (o a esa invocación) de forma distinta.

¿Que es Array?

Cuando hablamos de programación orientada a objetos, **una array se considera un objeto**. Eso quiere decir que si la declaramos tal y como hemos hecho, no estamos creando el objeto, sino que crea una referencia para poder utilizarlo. Para inicializar una array solemos utilizar la palabra reservada new .





¿Cómo se define una clase main en JAVA y muestre un ejemplo?

El método main es el punto de entrada de un programa ejecutable, es donde se inicia y finaliza el control del programa.

¿Cómo se define una clase main en JAVA y muestre un ejemplo?

El método main es el punto de entrada de un programa ejecutable, es donde se inicia y finaliza el control del programa

```
{
    static int main(string[] args)
    {
        // Test if input arguments were supplied.
        if (args.Length == 0)
        {
            Console.WriteLine("Please enter a numeric argument.");
            Console.WriteLine("Usage: Factorial <num>");
            return 1;
        }

        // Try to convert the input arguments to numbers. This will throw
        // an exception if the argument is not a number.
        // num = int.Parse(args[0]);
        int num;
        bool test = int.TryParse(args[0], out num);
        if (!test)
        {
            Console.WriteLine("Please enter a numeric argument.");
            Console.WriteLine("Usage: Factorial <num>");
            return 1;
        }

        // Calculate factorial.
        long result = Functions.Factorial(num);

        // Print result.
        if (result == -1)
            Console.WriteLine("Input must be >= 0 and <= 20.");
        else
            Console.WriteLine($"The Factorial of {num} is {result}.");

        return 0;
    }
}
// If 3 is entered on command line, the
// output reads: The factorial of 3 is 6.
```

Content Here

You can simply impress your audience and add a unique zing.

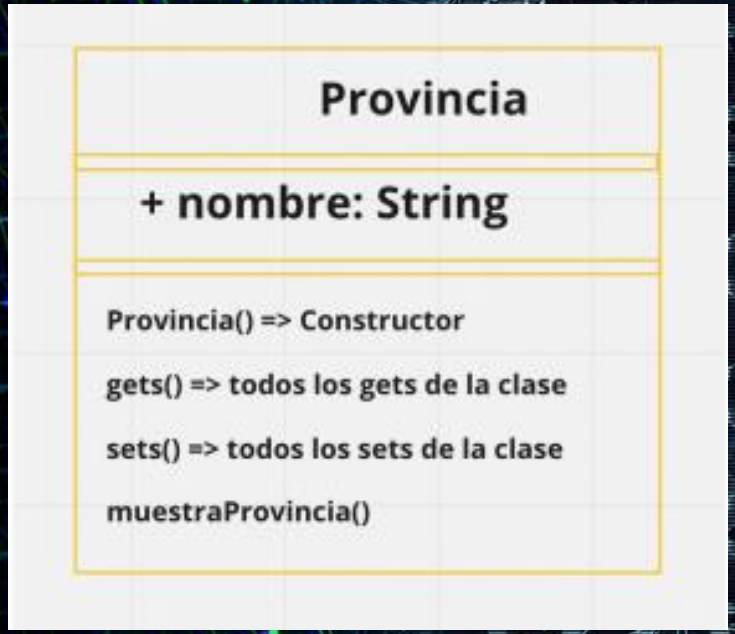
¿Que es Abstraction y muestra un ejemplo?

```java
public class UniversityStudent {
    int id;
    String name;
    String gender;
    String university;
    String career;
    int numSubjects;
    public UniversityStudent(int id, String name, String gender,
        String university, String  career, int numSubjects) {
        this.id = id;
        this.name = name;
        this.gender = gender;
        this.university = university;
        this.career = career;
        this.numSubjects = numSubjects;
    }
    void inscribeSubjects() {
      // TODO: implement
    }
    void cancelSubjects() {
       // TODO: implement
    }
    void consultRatings() {
       // TODO: implement
    }
}
```

La **abstracción** consiste en seleccionar datos de un conjunto más grande para mostrar solo los detalles relevantes del objeto. Ayuda a reducir la complejidad y el esfuerzo de **programación**. En Java, la **abstracción** se logra usando clases e interfaces abstractas. Es uno de los conceptos más importantes .

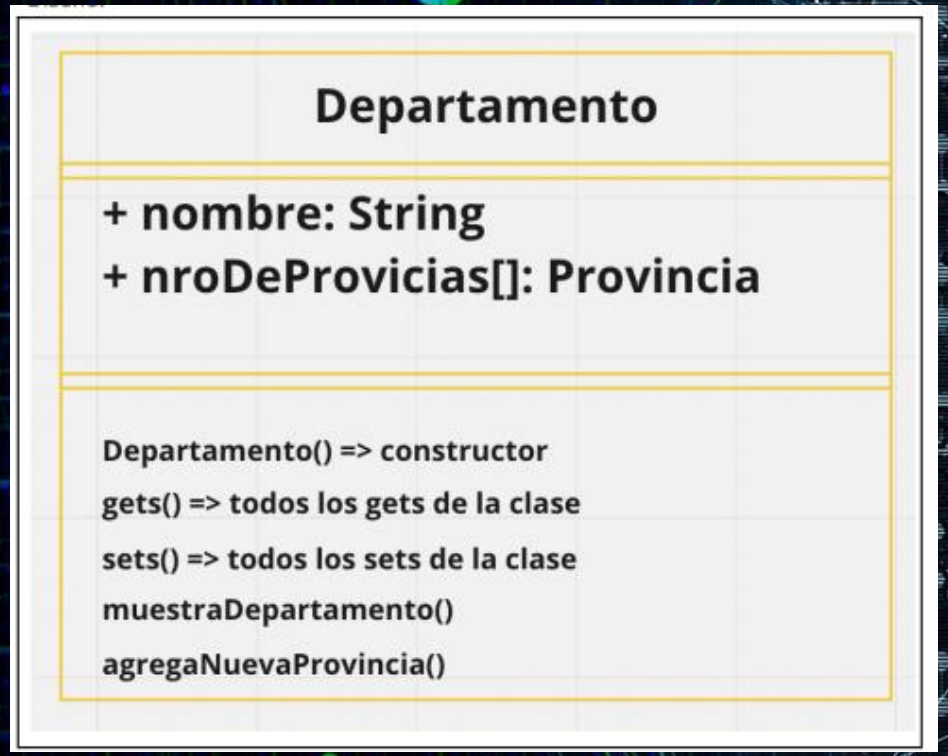
Practica

Generación la clase Provincia




```java
package Pais;

public class Provincia {
    4 usages
    private String nombreProvincia;

    public Provincia(String nombreProvincia){
        this.nombreProvincia = nombreProvincia;
    }

    1 usage
    public Provincia(){
        this.nombreProvincia = "";
    }

    1 usage
    public String getNombreProvincia() {
        return nombreProvincia;
    }

    1 usage
    public void setNombreProvincia(String nombreProvincia) {
        this.nombreProvincia = nombreProvincia;
    }
    1 usage
    public void mostrarProvincia(){
        System.out.println("Mostrando Nombre de la Provincia");
        System.out.println("nombreProvincia: " + this.getNombreProvincia());
    }
```

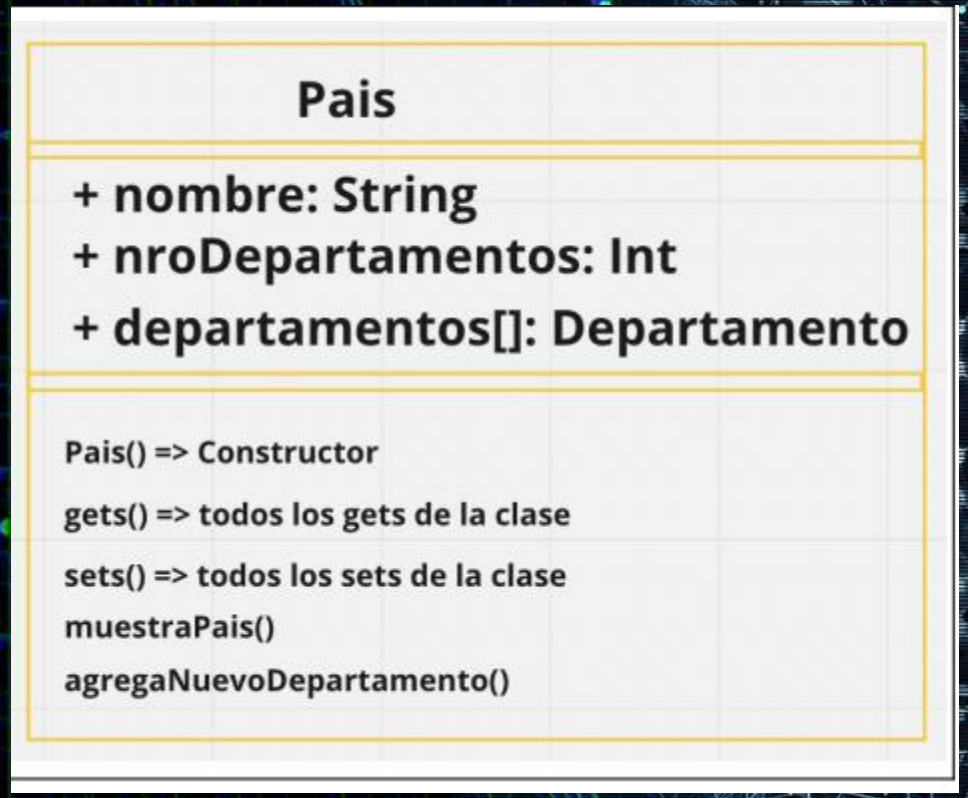
General la clase Departamento




```java
package Pais;

public class Departamento {
    3 usages
    private String nombreDepartamento;
    3 usages
    private Provincia[] nroProvincias;


    public Departamento(String nombreDepartamento, Provincia[] nroProvincias) {
        this.nroProvincias = nroProvincias;
        this.nombreDepartamento = nombreDepartamento;
    }

    1 usage
    public Departamento(){

    }

    1 usage
    public String getNombreDepartamento() { return this.nombreDepartamento; }

    2 usages
    public Provincia[] getNroDeProvincias() { return this.nroProvincias; }

    1 usage
    public void setNombreDepartamento(String nombreDepartamento) { this.nombreDepartamento = nombreDepartamento; }

    1 usage
    public void setNroDeProvincias(Provincia[] nroDeProvincias) { this.nroProvincias = nroDeProvincias; }
```

```java
    }

    1 usage
    public String getNombreDepartamento() { return this.nombreDepartamento; }

    2 usages
    public Provincia[] getNroDeProvincias() { return this.nroProvincias; }

    1 usage
    public void setNombreDepartamento(String nombreDepartamento) { this.nombreDepartamento = nombreDepartamento; }

    1 usage
    public void setNroDeProvincias(Provincia[] nroDeProvincias) { this.nroProvincias = nroDeProvincias; }

        1 usage
        public void mostrarDepartamento () {
            System.out.println("Mostrando Nombre del Departamento");
            System.out.println("nombreDepartamento: " + this.getNombreDepartamento());

            for (int i = 0; i < this.getNroDeProvincias().length; i++) {
                this.getNroDeProvincias()[i].mostrarProvincia();


        }
    }
}
```

Generar la clase País




```java
package Pais;

public class Pais {

    3 usages
    private String nombrePais;
    4 usages
    private int nroDepartamentos;
    3 usages
    private Departamento[] departamentos;

    1 usage
    public Pais(String nombrePais , Departamento[] departamentos){
        this.nombrePais = nombrePais;
        this.nroDepartamentos = nroDepartamentos;
        this.departamentos = departamentos;
    }


    1 usage
    public String getNombrePais() { return this.nombrePais; }

    public int getNroDepartamentos() {
        return nroDepartamentos;
    }

    2 usages
    public Departamento[] getDepartamentos() { return this.departamentos; }
```

Generar la clase País

```
    1 usage
    public String getNombrePais() { return this.nombrePais; }

    public int getNroDepartamentos() {
        return nroDepartamentos;
    }

    2 usages
    public Departamento[] getDepartamentos() { return this.departamentos; }

    public void setNombrePais(String nombrePais) { this.nombrePais = nombrePais; }

    public void setNroDepartamentos(int nroDepartamentos) { this.nroDepartamentos = nroDepartamentos; }

    public void setDepartamentos(Departamento[] departamentos) { this.departamentos = departamentos; }
    1 usage
    public void mostrarPais() {
        System.out.println("Mostrando Nombre del Pais");
        System.out.println("nombrePais: " + this.getNombrePais());

        for (int i=0; i<this.getDepartamentos().length; i++){
            this.getDepartamentos()[i].mostrarDepartamento();
        }
    }
}
```

Crear el diseño completo de las clases

```java
package Pais;

import java.util.Scanner;

public class Main1 {
    public static void main(String [] args) {
        Scanner leer = new Scanner(System.in);

        //Pregunta 1

        String nombreProvincia;
        int i, nProvincias;
        nProvincias = 2;

        String nombreDepartamento;
        int j, nDepartamentos = 2;

        Departamento[] departamentos = new Departamento[100];

        for (j = 0; j < nDepartamentos; j = j + 1) {
            System.out.println("Ingrese el nombre del departamento " + (j + 1) + ": ");
            nombreDepartamento = leer.next();
            Provincia[] provincias = new Provincia[100];

            for (i = 0; i < nProvincias; i = i + 1) {
                System.out.println("Ingrese el nombre de la provincia " + (i + 1) + ": ");
                nombreProvincia = leer.next();

                Provincia prov = new Provincia();
                prov.setNombreProvincia(nombreProvincia);
                System.out.println("Ingrese el nombre del departamento " + (j + 1) + ": ");
                nombreDepartamento = leer.next();
                Provincia[] provincias = new Provincia[100];

                for (i = 0; i < nProvincias; i = i + 1) {
                    System.out.println("Ingrese el nombre de la provincia " + (i + 1) + ": ");
                    nombreProvincia = leer.next();

                    Provincia prov = new Provincia();
                    prov.setNombreProvincia(nombreProvincia);

                    provincias[i] = prov;
                };

                Departamento dep = new Departamento();
                dep.setNombreDepartamento(nombreDepartamento);
                dep.setNroDeProvincias(provincias);
                departamentos[j] = dep;
            }

            Pais pais = new Pais( nombrePais: "BOLIVIA", departamentos);
            pais.mostrarPais();
        }
    }
```

GRACIAS POR SU ATENCION

eate.iliaaraceli.sarzo.la@unifranz.edu.bo