ESTRUCTURA DE DATOS

DEFENSA HITO 2 ESTRUCTURA DE DATOS

ESTUDIANTE

JHONATAN DAVID
ALANOCA BLANCO

MANEJO DE CONCEPTOS

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

Habla de programación orientada a objetos se enfoca en la creación de objetos que tienen ciertas propiedades y métodos.

2. ¿Cuáles son los 4 componentes que componen POO?

1.Clases: Las clases pueden ser definidas como un molde que contendrá todas las características.
2.Propiedades: Las propiedades son las características de una clase.
3.Métodos: Los métodos son las acciones que una clase puede realizar.
4.Objetos: Son aquellos que tienen propiedades y comportamientos, estos pueden ser físicos o conceptuales.

3. ¿Cuáles son los pilares de POO?.

son cuatro : El encapsulamiento, la herencia, el polimorfismo y la abstracción

4. ¿Qué es Encapsulamiento y muestre un ejemplo?

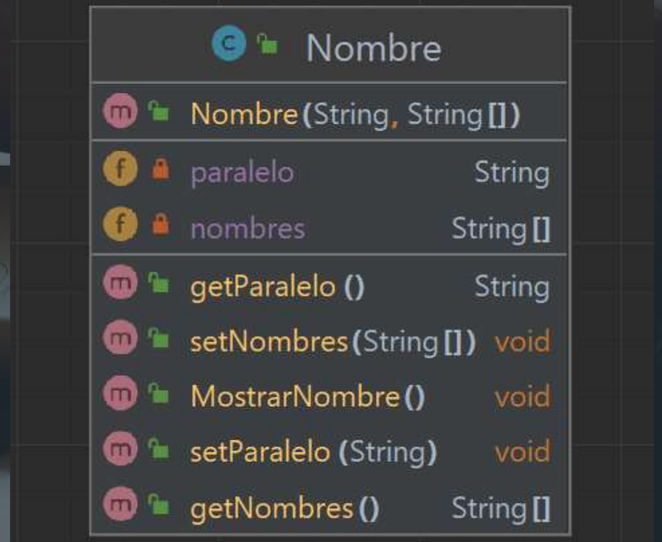
El encapsulamiento establece que los datos y el comportamiento de un objeto deben estar protegidos de la manipulación no autorizada

```
public class Nombre{
    3 usages
    private String paralelo;
    4 usages
    private String[] nombres;

    1 usage
    public Nombre(String paralelo, String[] nombres)
    {
        this.nombres=nombres;
        this.paralelo=paralelo;
    }

    1 usage
    public String getParalelo() {
        return paralelo;
    }

    no usages
    public void setParalelo(String paralelo) {
        this.paralelo = paralelo;
    }

    6 usages
    public String[] getNombres() {
        return nombres;
    }

    3 usages
    public void setNombres(String[] nombres) {
        this.nombres = nombres;
    }
```

5. ¿Qué es Abstracción y muestre un ejemplo?

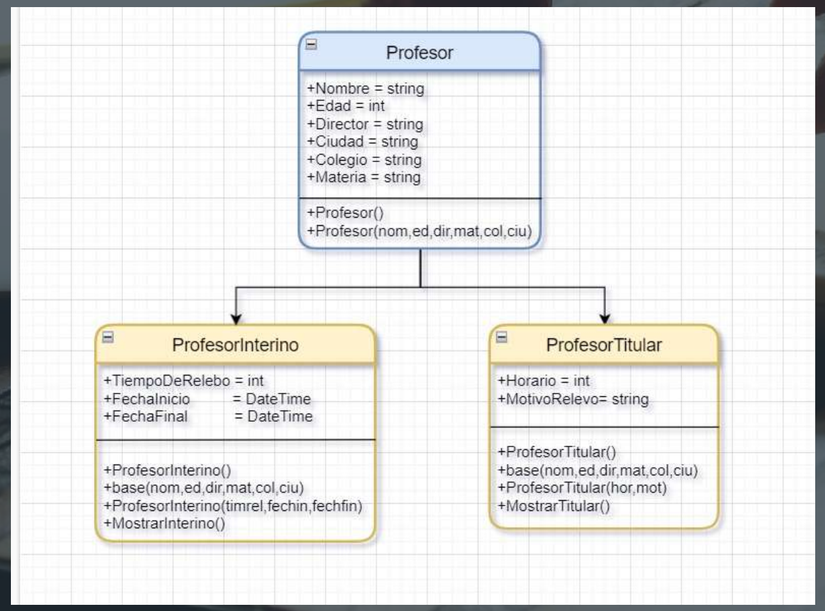
Es el principio que establece que un objeto debe ser representado en forma de un modelo conceptual simplificado y generalizado, que identifica sus características esenciales y relevantes.

```
public class Nombre{
    3 usages
    private String paralelo;
    4 usages
    private String[] nombres;
```



6. ¿Que es Herencia y muestre un ejemplo?

es el principio que establece que una clase puede heredar propiedades y métodos de una clase base existente, y agregar o modificar su comportamiento para adaptarlo a sus necesidades específicas.




```java
class Animal {
    no usages
    public void comer() {
        System.out.println("Estoy comiendo");
    }
}

// Clase hija que hereda de Animal
no usages
class Perro extends Animal {
    no usages
    public void ladrar() {
        System.out.println("Estoy ladrando");
    }
}
```

7. ¿Qué es Polimorfismo y muestre un ejemplo?

es el principio que establece que los objetos de diferentes clases pueden responder al mismo mensaje o método de forma diferente, según su propia implementación. El polimorfismo permite la flexibilidad y la extensibilidad del código.

Una clase puede adquirir diferentes valores un ejemplo una clase vehículo puede adquirir diferentes valores, puede ser una moto o un auto

```
public class Vehiculo {
 // atributos

    public void moverse() {
          System.out.println(
         "El vehículo se está moviendo con energia");
    }
}

public class Camion extends Vehiculo {
 // atributos private int cargaMaxima;

    @Override
    public void moverse() {
           System.out.println(
           "El camión se está moviendo con gasolina");
    }
}

public class Moto extends Vehiculo {
 // atributos private boolean esElectrica;

    @Override
    public void moverse() {
          System.out.println(
          "La moto se está moviendo con electricidad");
    }
}
```

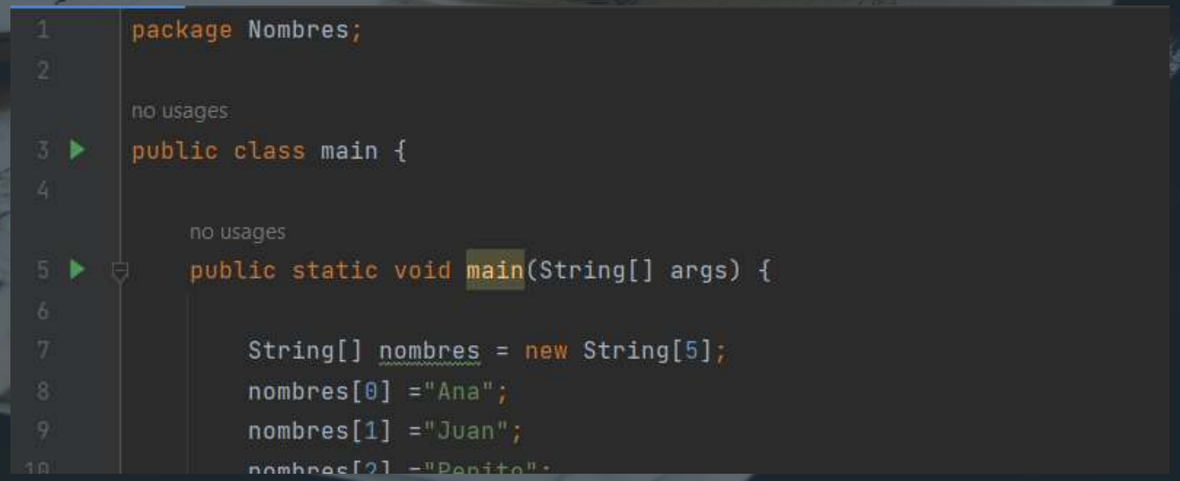
8. Que es un ARRAY?

Es una estructura de datos que permite almacenar una colección de elementos del mismo tipo en una sola variable.

9. ¿Qué son los paquetes en JAVA?

Es una forma de organizar clases relacionadas, se utilizan para evitar colisiones de nombres entre clases.

10.¿Cómo se define una clase main en JAVA y muestra un ejemplo?

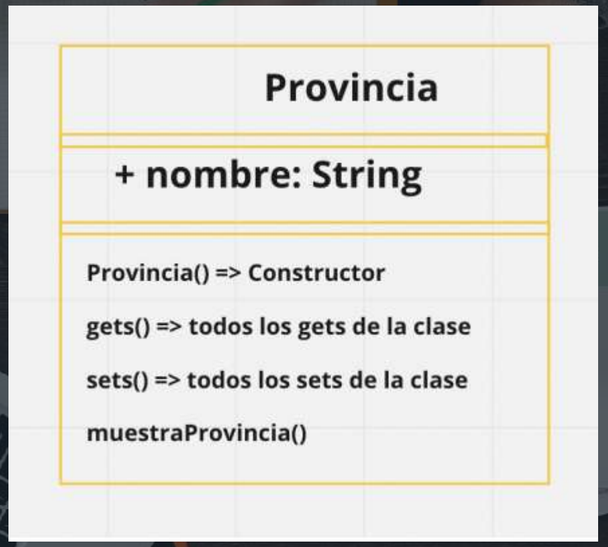


```
1   package Nombres;
2
    no usages
3   public class main {
4
        no usages
5       public static void main(String[] args) {
6
7           String[] nombres = new String[5];
8           nombres[0] ="Ana";
9           nombres[1] ="Juan";
10          nombres[2] ="Benito";
```

PARTE PRACTICA

11. Generar la clase Provincia.

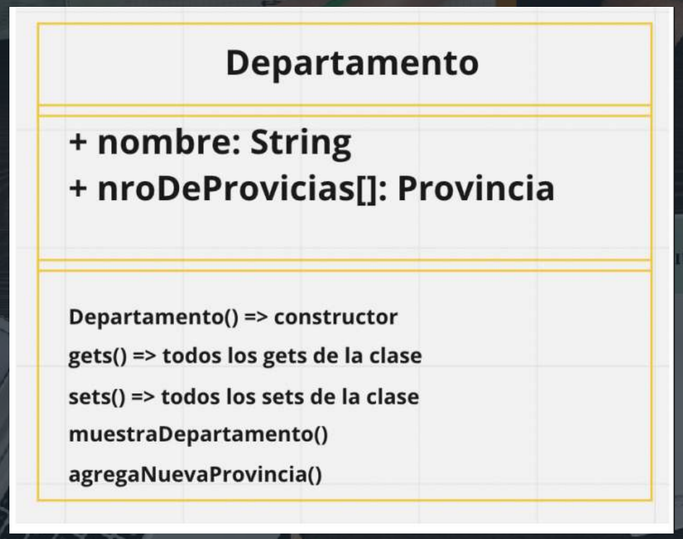
- Diseno.



- Crear una clase MAIN
  - Crear todos los gets y sets de la clase.
  - El constructor no recibe parámetros.
  - Crear una instancia de la clase Provincia
  - Mostrar los datos de una provincia 4
- Adjuntar el código JAVA generado.

12.Generar la clase Departamento.

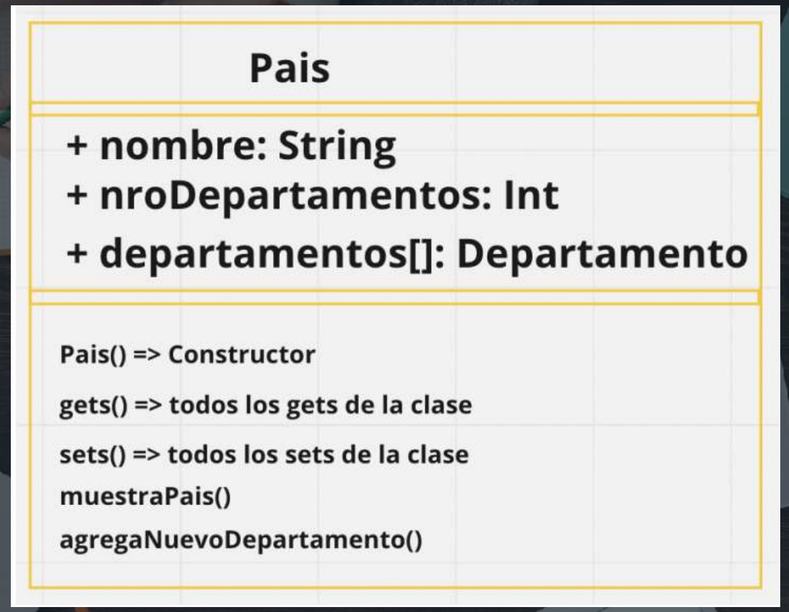
- Diseno.



- Crear una clase MAIN (Utilizar el MAIN del anterior ejercicio)
  - Crear todos los gets y sets de la clase.
  - El constructor no recibe parámetros.
  - Crear una instancia de la clase Departamento.
  - Omitir el método agregaNuevaProvincia()
  - Mostrar los datos de los departamentos.
- Adjuntar el código JAVA generado.

13.Generar la clase País.

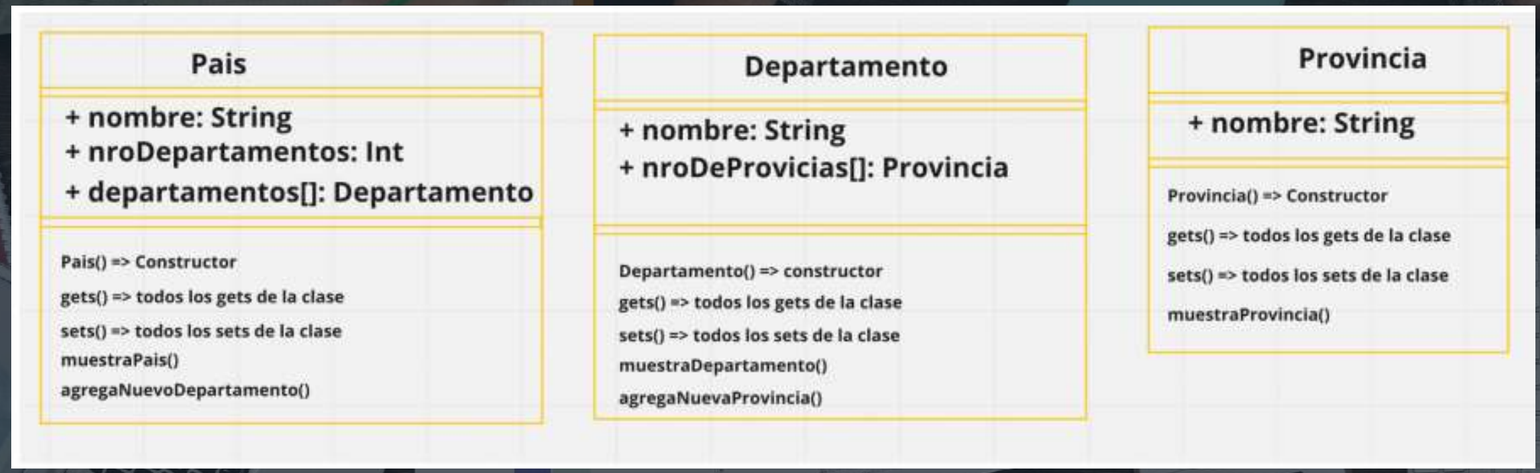
- Diseno.



- Crear una clase MAIN (Utilizar el MAIN del anterior ejercicio)
  - Crear una instancia de la clase País
  - El constructor no recibe parámetros.
  - Crear una instancia de la clase Departamento.
  - Omitir el método agregaNuevoDepartamento()
  - Mostrar los datos del País. 6
- Adjuntar el código JAVA generado.

14.Crear el diseño completo de las clases.

- Diseno.



- Crear todos gets y sets de cada clase.
- Implementar los métodos agregarNuevoDepartamento(), agregarNuevaProvincia(), es decir todos los métodos.
- El método agregarNuevoDepartamento permite ingresar un nuevo departamento a un país.
- El método agregarNuevaProvincia permite ingresar una nueva provincia a un departamento. ○ La clase Main debe mostrar lo siguiente:
  - Crear el PAÍS Bolivia
  - Al país Bolivia agregarle 3 departamentos.
  - Cada departamento deberá tener 2 provincias.
- Adjuntar el código JAVA generado.