
ESTRUCTURA DE DATOS
HITO 2
POR:MARCO ANTONIO CALLE VAQUIATA

ÍNDICE

INTRODUCCION

Crear un sistema para gestionar todos los departamentos y provincias que tiene un país (Ejemplo Bolivia). Un país tiene departamentos y un departamento tiene provincias.

Objetivos

1. Manejo de Clases en JAVA
2. Manejo del paradigma POO.
3. Manejo de estructuras ARRAY

Esta tarea se divide en dos partes:

- La primera parte corresponde a la parte TEÓRICA necesaria, en donde se encuentra un conglomerado de preguntas relacionadas a ESTRUCTURA DE DATOS.
- La segunda parte corresponde a la parte PRÁCTICA necesaria en donde deberá realizar y crear CLASES que muestran POO.

Manejo de conceptos

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. ¿Cuáles son los 4 componentes que componen POO?

Abstracción, Encapsulamiento, Modularidad, Jerarquía.

3. ¿Cuáles son los pilares de POO?.

- **Pilar** 1: Encapsulación. El concepto de encapsulación es el más evidente de todos. ...
- **Pilar** 2: Abstracción. Este concepto está muy relacionado con el anterior. ...
- **Pilar** 3: Herencia. ...
- **Pilar** 4: Polimorfismo.

4. ¿Qué es Encapsulamiento y muestre un ejemplo?

- El elemento más común de encapsulamiento son las clases, donde encapsulamos y englobamos tanto métodos como propiedades. Otro ejemplo muy común de encapsulamiento son los getters y setters de las propiedades dentro de una clase. Por defecto nos dan el valor “normal” pero podemos modificarlos para que cambie.

5. ¿Qué es Abstracción y muestre un ejemplo?

Se trata de una actividad intelectual que se produce en todas las etapas del desarrollo cognitivo y que es fundamental para el aprendizaje. También es un concepto presente en áreas específicas como el arte, las ciencias naturales, la filosofía, entre otras.

Elaboración de definiciones. Las personas retienen los aspectos fundamentales del elemento que describen al elaborar este tipo de enunciados explicativos y breves. Para elaborarlos es necesario descartar las cuestiones menos relevantes que no reflejan la esencia del concepto.

6. ¿Que es Herencia y muestre un ejemplo?

Al principio cuesta un poco entender estos conceptos característicos del paradigma de la POO porque solemos venir de otro paradigma de programación como el paradigma de la programación estructurada , pero se ha de decir que la complejidad está en entender este nuevo paradigma y no en otra cosa.

```
public class Futbolista
{

        private int id;
        private String Nombre;
        private String Apellidos;
        private int Edad;
        private int dorsal;
        private String demarcacion;

        // constructor, getter y setter

        public void Concentrarse() {
                ...
        }
```

```
        public void Viajar() {
                ...
        }

        public void jugarPartido() {
                ...
        }

        public void entrenar() {
                ...
        }
}
```

7. ¿Qué es Polimorfismo y muestre un ejemplo?

Un **ejemplo** clásico de polimorfismo es el siguiente. Podemos crear dos clases distintas: Gato y Perro, **que** heredan de la superclase Animal. La clase Animal tiene el método abstracto makesound() **que** se implementa de forma distinta en cada una de las subclases (gatos y perros suenan de forma distinta)

8. ¿Que es un ARRAY?

Un array (unidimensional, también denominado vector) es una **variable estructurada formada de un número "n" de variables simples del mismo tipo que son denominadas los componentes o elementos del array**. El número de componentes "n" es, entonces, la dimensión del array

9. ¿Qué son los paquetes en JAVA?

Un **Paquete en Java** es un contenedor de clases **que** permite agrupar las distintas partes de un programa y **que** por lo general tiene una funcionalidad y elementos comunes, definiendo la ubicación de dichas clases en un directorio de estructura jerárquica

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

En Java, el método main () es el método de entrada de una aplicación Java, es decir, cuando el programa está en ejecución, el primer método a ejecutar es el método main (), este método es muy diferente a otros métodos. Por ejemplo, el nombre del método debe ser main, el método debe ser de tipo public static void, el método debe recibir una matriz de cadenas de parámetros, etc.

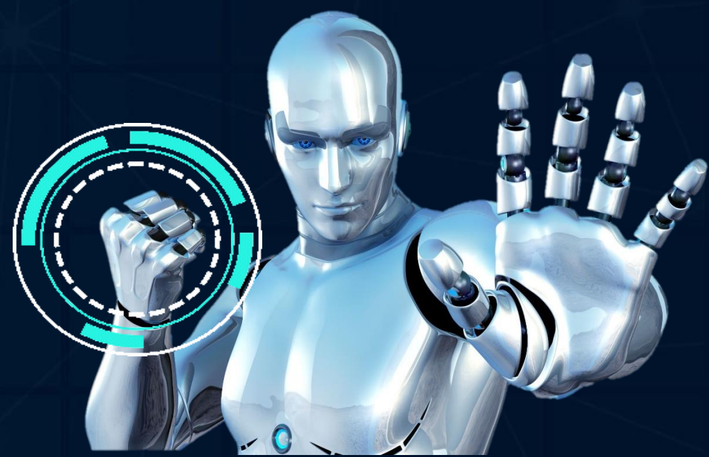
Parte practica

11. Generar la clase Provincia.

Provincia
+ nombre: String
Provincia() => Constructor gets() => todos los gets de la clase sets() => todos los sets de la clase muestraProvincia()

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**

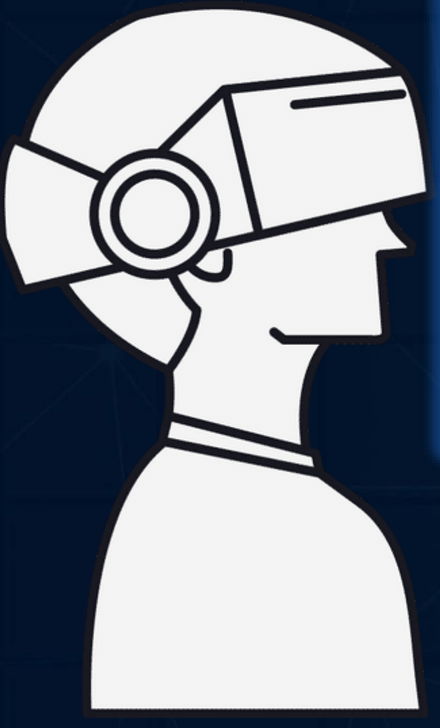


RESULTADO

```java
package HITO2;

public class PROVINCIA {
    3 usages
    private String Nombre;

    public PROVINCIA(String Nombre)
    {
        this.Nombre="";
    }

    1 usage
    public String getNombre() {
        return this.Nombre;
    }

    public void setNombre(String nombre) {
        this.Nombre = nombre;
    }
    public void muestraProvencia(){
System.out.println("\nEL NOMBRE DE LAS PROVINCIAS ES:");
System.out.println("NOMBRE DE PROVINCIA:"+this.getNombre());
    }
}
```

```java
package HITO2;

public class MAIN {
    public static void main(String[] args) {
        PROVINCIA P1= new PROVINCIA( Nombre: "CARANAVI");
        P1.muestraProvencia();

    }

}
```

```
EL NOMBRE DE LAS PROVINCIAS ES:
NOMBRE DE PROVINCIA:CARANAVI

Process finished with exit code 0
```



12.Generar la clase Departamento.

Departamento

+ nombre: String
+ nroDeProvicias[]: Provincia

Departamento() => constructor
gets() => todos los gets de la clase
sets() => todos los sets de la clase
muestraDepartamento()
agregaNuevaProvincia()

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.**

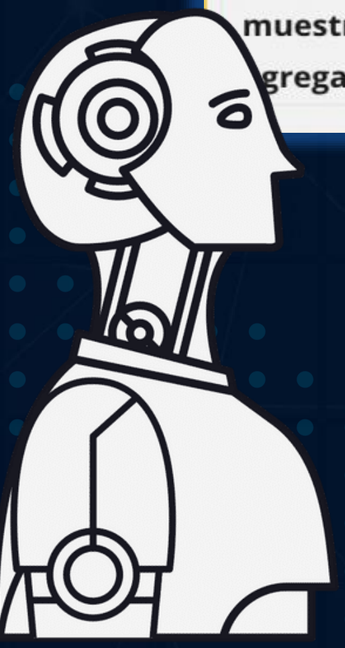
RESULTADO

```java
package HITO2;

public class DEPARTAMENTO {
    3 usages
    private String Nombre;
    3 usages
    private PROVINCIA[] provencias;

    1 usage
    public DEPARTAMENTO(String Nombre,PROVINCIA[] provencias)
    {
        this.Nombre=Nombre;
        this.provencias=provencias;
    }

    1 usage
    public String getNombre() {
        return this.Nombre;
    }

    2 usages
    public PROVINCIA[] getProvencias() {
        return provencias;
    }
```

```java
public PROVINCIA[] getProvencias() {
    return provencias;
}

public void setProvencias(PROVINCIA[] provencias) {
    this.provencias = provencias;
}

public void setNombre(String nombre) {
    Nombre = nombre;
}
1 usage
public void muestraDepartamento (){
    System.out.println("\nMOSTRANDO NOMBRE DE LOS DEPARTAMENTOS");
    System.out.println("Nombre De Departamento: " + this.getNombre());
    for (int i = 0; i < getProvencias().length; i++) {
        this.getProvencias()[i].muestraProvencia();
    }
```

```java
package HITO2;

public class MAIN {
    public static void main(String[] args) {
        PROVINCIA P1= new PROVINCIA( Nombre: "CARANAVI");

        PROVINCIA P2= new PROVINCIA( Nombre: "PALOS BLANCOS");

        PROVINCIA[] PROVINCIAS=new PROVINCIA[2];
        PROVINCIAS[0]=P1;
        PROVINCIAS[1]=P2;
        DEPARTAMENTO D1=new DEPARTAMENTO( Nombre: "LAS LOMAS",PROVINCIAS);
        D1.muestraDepartamento();
```

13.Generar la clase País



Pais

+ nombre: String
+ nroDepartamentos: Int
+ departamentos[]: Departamento

Pais() => Constructor

gets() => todos los gets de la clase

sets() => todos los sets de la clase

muestraPais()

gregaNuevoDepartamento()

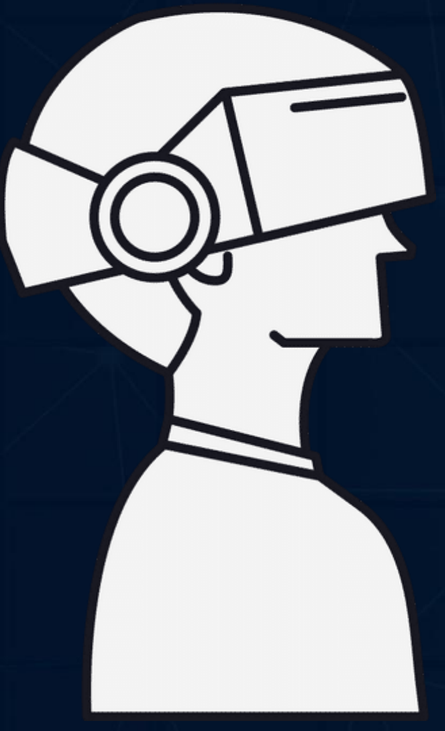
Crear una clase MAIN (Utilizar el MAIN del anterior ejercicio)

- **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.**

Tanta tarea...

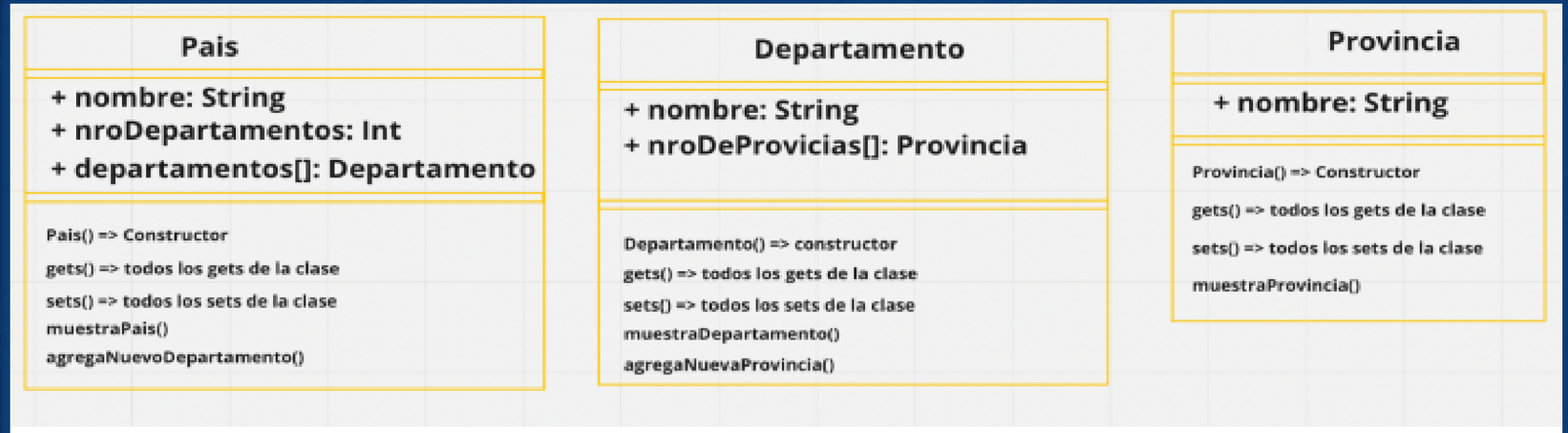
RESULTADO

```java
package HIT02;

public class PAIS {
    3 usages
    private String nombrePais;
    2 usages
    private int noDepartamentos;
    3 usages
    private DEPARTAMENTO[] departamentos;

    1 usage
    public PAIS(String nombrePais,int noDepartamentos, DEPARTAMENTO[] departamentos) {
        this.nombrePais = nombrePais;
        this.noDepartamentos = noDepartamentos;
        this.departamentos=departamentos;
    }
    public PAIS () {}
    1 usage
    public String getNombrePais() {
        return this.nombrePais;
    }
    1 usage
    public DEPARTAMENTO[] getDepartamentos() {
```

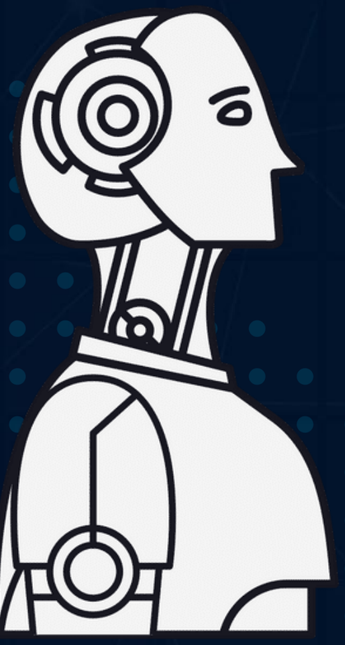


```java
public void setNombrePais(String nombrePais) {
    this.nombrePais = nombrePais;
}

public void setDepartamentos(DEPARTAMENTO[] departamentos) {
    this.departamentos = departamentos;
}
1 usage
public void mostrarPais() {
    System.out.println("\nMOSTRANDO DATOS DEL PAIS");
    System.out.println("Nombre Pais: " + this.getNombrePais());
    for (int i = 0; i < this.noDepartamentos; i++) {
        this.getDepartamentos()[i].muestraDepartamento();
    }
}
```

14.Crear el diseño completo de las clases.



- 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.



RESULTADO

```java
package HITO2;

public class DEPARTAMENTO {
    3 usages
    private String Nombre;
    4 usages
    private int noProvincias;
    3 usages
    private PROVINCIA[] provencias;

    public DEPARTAMENTO(String Nombre,PROVINCIA[] provencias,int noProvincias)
    {
        this.Nombre=Nombre;
        this.provencias=provencias;
        this.noProvincias=noProvincias;
    }
```

```java
package HITO2;

public class PAIS {
    3 usages
    private String nombrePais;
    2 usages
    private int noDepartamentos;
    3 usages
    private DEPARTAMENTO[] departamentos;

    1 usage
    public PAIS(String nombrePais,int noDepartamentos, DEPARTAMENTO[] departamentos) {
        this.nombrePais = nombrePais;
        this.noDepartamentos = noDepartamentos;
        this.departamentos=departamentos;
    }
    public PAIS () {}
```

RESULTADO

```java
package HIT02;

import java.util.Scanner;

public class MAIN2 {
    public static void main(String[] args) {
        Scanner leer = new Scanner(System.in);

        System.out.println("INGRESE DATOS DE PROVINCIAS");
        String nombreProvincia;
        int i, nProvincias;
        nProvincias = 2;

        System.out.println("INGRESE DATOS DE DEPARTAMENTOS");
        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();
            System.out.println("Ingrese el numero de provincias");
```

```java
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.setNombre(nombreProvincia);

    provincias[i] = prov;
}
System.out.println("Ingrese el nombre de la nueva provincia: ");
nombreProvincia = leer.next();

PROVINCIA prov = new PROVINCIA();
prov.setNombre(nombreProvincia);

provincias[nProvincias] = prov;

DEPARTAMENTO dep = new DEPARTAMENTO();
dep.setNombre(nombreDepartamento);
dep.setProvencias(provincias);
dep.setNoProvincias(nProvincias + 1);
departamentos[j] = dep;
```

```java
System.out.println("Ingrese el nombre del nuevo departamento: ");
nombreDepartamento = leer.next();

DEPARTAMENTO dep = new DEPARTAMENTO();
dep.setNombre(nombreDepartamento);

departamentos[nDepartamentos] = dep;
PAIS pais = new PAIS( nombrePais: "BOLIVIA",  noDepartamentos: nDepartamentos + 1, departamentos);
pais.mostrarPais();
```

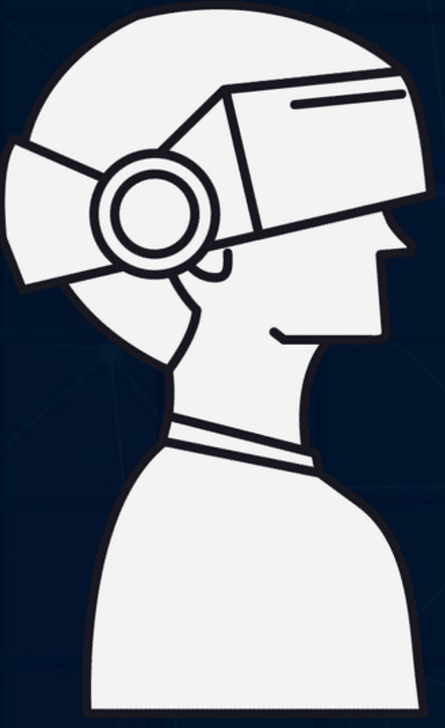
CONCLUSION

En este tema nuestros objetivos fueron:

1. Manejo de Clases en JAVA,
2. Manejo del paradigma POO.
3. Manejo de estructuras ARRAY.

Se concluyo que alcanzamos exitosamente con los objetivos

Pudiendo alcanzar nuevos temas y aprendes mas cada día.



GRACIAS