



Nombre Completo: Univ. Ricardo David La Madrid Alarcón

Asignatura: PRIGRAMACION III

Carrera: INGENIERÍA DE SISTEMAS

Paralelo: BDA (1)

Docente: Lic. William R. Barra Paredes

fecha: 30/03/2020

Preguntas.

Responda de manera breve y clara posible.

1.-Defina y muestre ejemplos de la clase Scanner.

La clase Scanner de Java provee métodos para leer valores de entrada de varios tipos y está localizada en el paquete java.util. Los valores de entrada pueden venir de varias fuentes, incluyendo valores que se entren por el teclado o datos almacenados en un archivo.

Ejemplos:

```
int n;
```

```
System.out.print("Introduzca un número entero: ");
```

```
n = sc.nextInt()
```

```
double x;
```

```
System.out.print("Introduzca número de tipo double: ");
```

```
x = sc.nextDouble()
```

2.- ¿Que es la programación orientada a objetos(POO)?

Es una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.

3.-¿Cuál es la diferencia entre interfaz y herencia?

La diferencia principal entre interface y herencia es que la herencia tiene una relacion mas estrecha con las clases derivadas que una interface. Una interface tan solo define los metodos que deben tener las clases que la implementan, no puede definir que cuerpo tendran ni variables (salvo static y final).Sin embargo, una herencia puede definir un cuerpo, variables y como seran los metodos de las clases que extiendan de ella

4.- ¿Qué elementos crees que definen a un objeto?

Su cardinalidad y su tipo

Sus atributos y sus métodos

Su interface y los eventos asociados

5.- ¿Que es una clases abstracta y muestre un ejemplo?

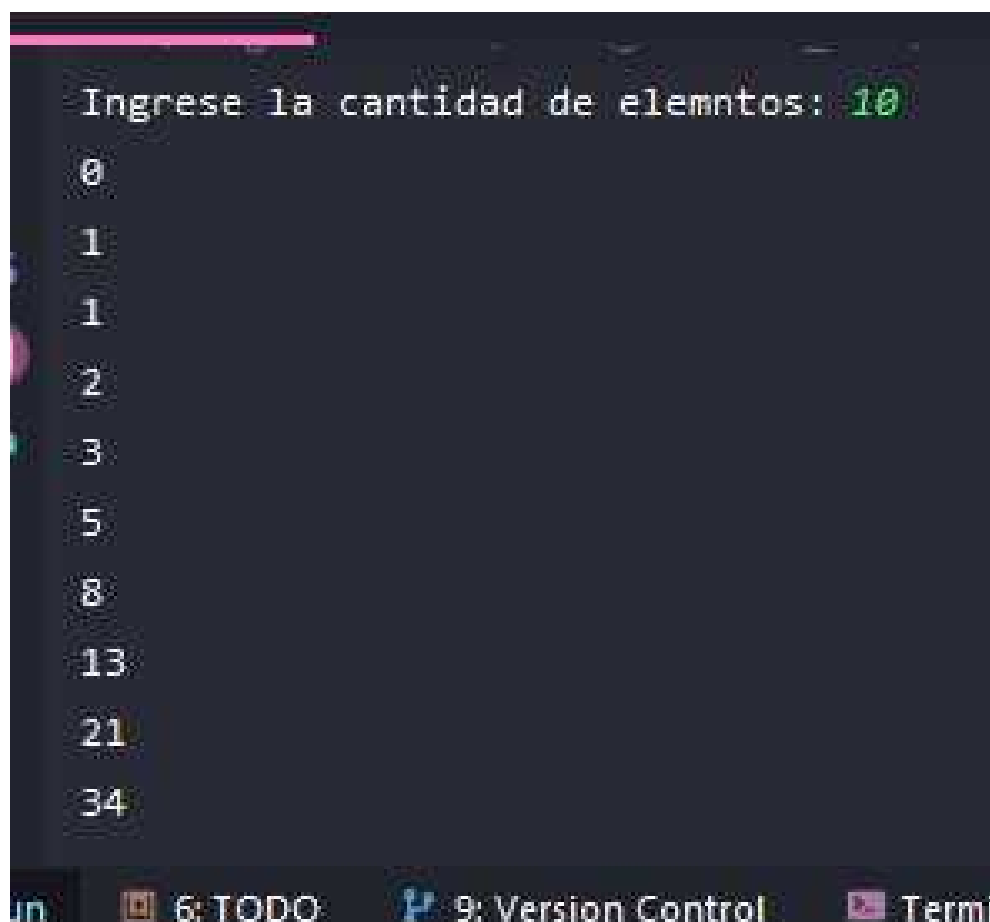
Las clases abstractas, como su nombre lo indica, son algo abstracto, no representan algo específico y las podemos usar para crear otras clases. No pueden ser instanciadas, por lo que no podemos crear nuevos objetos con ellas.

```
public abstract class Animal {  
  
    public Animal(String value) {  
        // Constructor  
        this.value = value;  
    }  
  
    public abstract void sound()  
  
}
```

Ejercicios.

Generar la serie Fibonacci hasta un valor n leído por teclado

Captura



A screenshot of a terminal window with a dark background. The prompt 'Ingrese la cantidad de elemntos:' is followed by the input '10' in green. Below this, the Fibonacci sequence is displayed: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34. The numbers 21 and 34 are highlighted in yellow. At the bottom of the terminal, there is a status bar with icons and text: '6: TODO', '9: Version Control', and 'Termi'.

```
Ingrese la cantidad de elemntos: 10  
0  
1  
1  
2  
3  
5  
8  
13  
21  
34
```

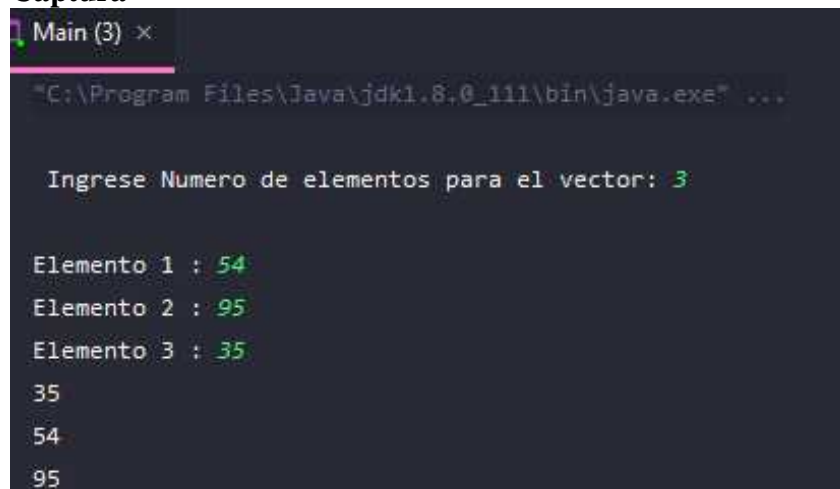
Código

```
public void Fibonacci()
{
    int a=0,b=1,x,y;
    System.out.printf("Ingrese la cantidad de elemntos: ");
    x = scn.nextInt();
    for(int i=0; i<x;i++)
    {
        System.out.println(a);
        y = a+b;
        a = b;
        b = y;
    }
}
```

Mostrar 2 métodos de ordenación de vectores.
Puede ser burbuja el método por selección.

Burbuja

Captura



```
Main (3) x
"C:\Program Files\Java\jdk1.8.0_111\bin\java.exe" ...

Ingrese Numero de elementos para el vector: 3

Elemento 1 : 54
Elemento 2 : 95
Elemento 3 : 35
35
54
95
```

Código

```
public void Burbujas() throws IOException {
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    System.out.print("\n Ingrese Numero de elementos para el vector: ");
    int tam = Integer.parseInt(in.readLine());
    int arr[] = new int[tam];
    System.out.println();
    int j = 0;
    for (int i = 0 ; i < arr.length;i++)
    {
        j+=1;
        System.out.print("Elemento " + j + " : ");
        arr[i] = Integer.parseInt(in.readLine());
    }
    burbuja(arr);
}
static void burbuja(int arreglo[])
{
    for(int i = 0; i < arreglo.length - 1; i++)
```

```

{
    for(int j = 0; j < arreglo.length - 1; j++)
    {
        if (arreglo[j] > arreglo[j + 1])
        {
            int tmp = arreglo[j+1];
            arreglo[j+1] = arreglo[j];
            arreglo[j] = tmp;
        }
    }
}
for(int i = 0; i < arreglo.length; i++)
{
    System.out.print(arreglo[i]+"\\n");
}
}

```

Selección

Captura

```

Ingrese el numero de elementos para el vector: 3
Elemento1:56
Elemento2:95
Elemento3:85
56, 85, 95, }
Process finished with exit code 0

```

Código

```

public void Seleccion() {

    Scanner numero=new Scanner(System.in);

    int num;

    System.out.print("Ingrese el numero de elementos para el vector: ");

    num=numero.nextInt();

    int vector[] = new int[30];

    int aux;

    for (int i=0;i<num;i++)

    {
        System.out.print("Elemento"+(i+1)+":");
        vector[i]=numero.nextInt();
    }
    for (int i = 0; i < num - 1; i++)
    {
        int min = i;
        for (int j = i + 1; j < num; j++)
        {
            if (vector[j] < vector[min])
            {
                min = j;
            }
        }
        if (i != min)
        {
            aux= vector[i];
            vector[i] = vector[min];

```

```

        vector[min] = aux;
    }
}

for (int i=0;i<num;i++)
{
    System.out.print(vector[i]+ ", ");
}
System.out.print("{}");
}
}

```

Usando while y el método hasNext() de la clase Scanner, leer N cadenas hasta encontrar una cadena que tenga una cantidad de caracteres igual a 10. Si la cadena ingresada tiene un número igual a 10 caracteres mostrar un mensaje indicando “Cadena Encontrada” y salir del while.

Crear un array con 10 elementos enteros.
Determinar cuántos elementos de ese array son pares.

I. Manejo de Interfaces: LeerInteface.java

Esta **interface** tiene declarado una instancia del objeto **Scanner(System.in)**. Esta **interfaz** debe ser implementada en la **clase Empleado**.

El nombre de esta variable es LEER.

```

    Scanner LEER = new Scanner(System.in);

```

II. Manejo de Herencia: Empleado.java

Esta es la clase **padre** del cual heredan otras subclases. Esta clase implementa la interfaz **LeerInterface.java**.

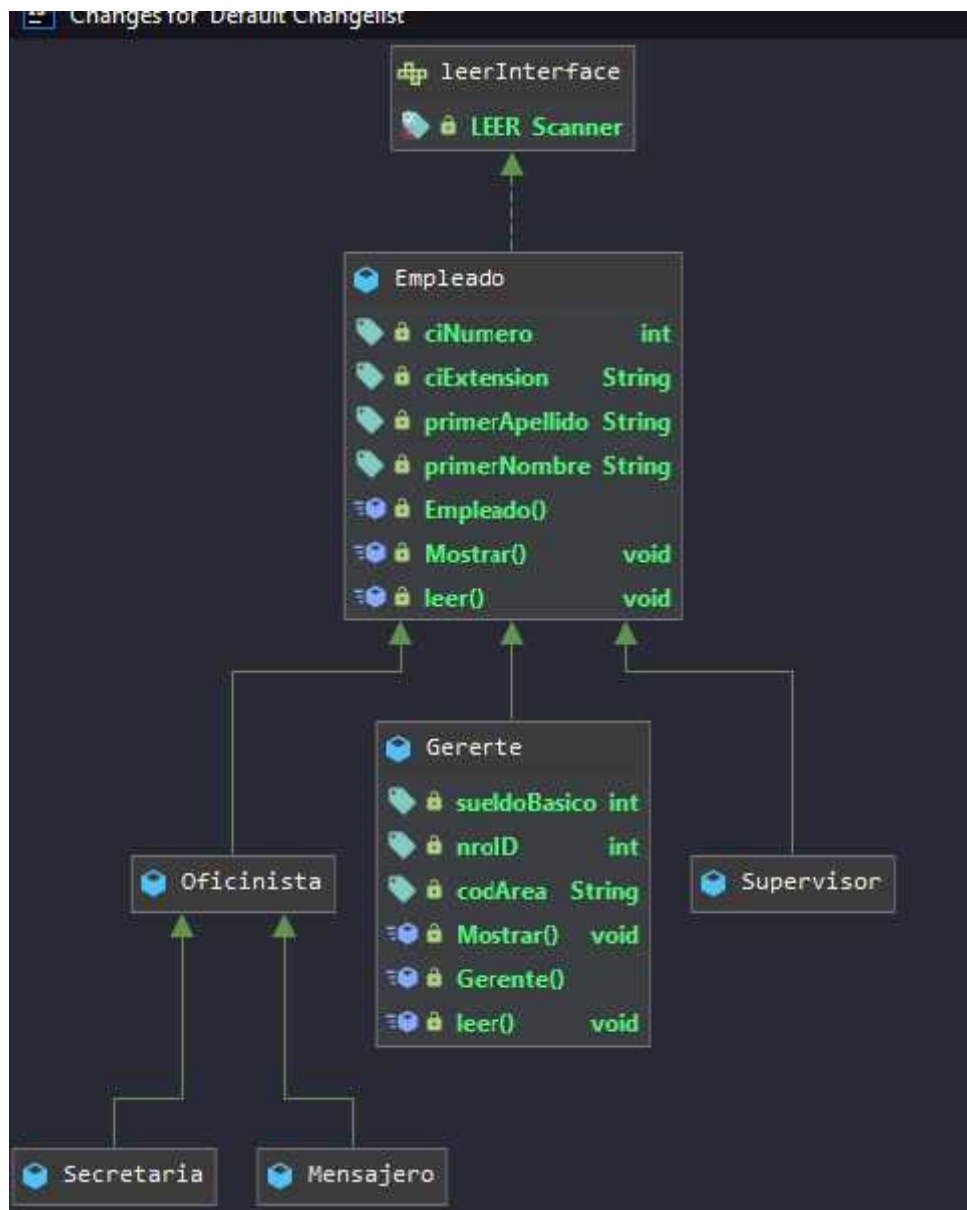
III. Manejo de Clases: Gerente.java, Supervisor.java y Oficinista.java

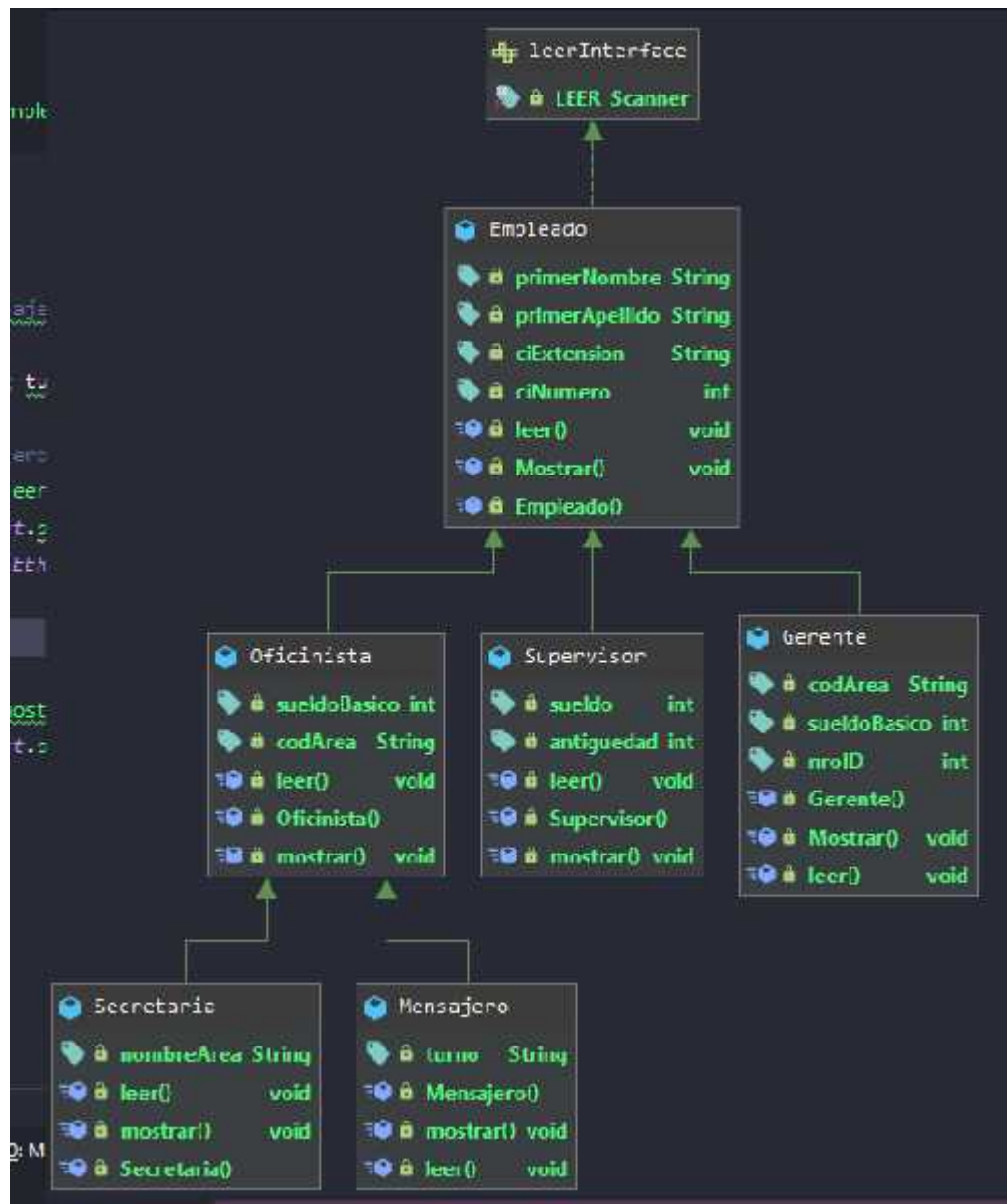
Todas estas clases heredan de la clase **Empleado.java**

IV. Manejo de Interfaces, Herencia y Clases: Secretaria.java y Mensajero.java

Estas clases heredan de la clases **Oficinista.java**.

Capturas





```

Main (2) %
TC: [2] Ingrese el File y el Tipo del ROL: [1] Ingrese los datos:
Ingrese Primer Nombre: [1] Ingrese
Ingrese Primer Apellido: [1] Ingrese
Ingrese Numero CI: [0] 9517843
Ingrese Direccion CI: [0] 10300
Ingrese Numero de LU: [0] 123
IngreseCodigo de Area: [0] 100-215
Ingrese Sueldo Base: [0] 2500
Nombre: Ricardo, Apellido: Lohndred, NumeroCI: 9517843, ExtensionCI: 1000, NoLU: 123, CodArea: 100-215, SueldoBase: 2500
Presione el tecla enter para salir

```


Código

Interface

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

Empleado

```
public class Empleado implements LeerInterface {  
    public String primerNombre;  
    public String primerApellido;  
    public int ciNumero;  
    public String ciExtension;  
  
    public Empleado() {  
        super();  
    }  
  
    public void leer() {  
        System.out.printf("Ingrese Primer Nombre :");  
        primerNombre = LEER.next();  
        System.out.printf("Ingrese Primer Apellido :");  
        primerApellido = LEER.next();  
        System.out.printf("Ingrese Numero CI :");  
        ciNumero = LEER.nextInt();  
        System.out.printf("Ingrese Extencio CI :");  
        ciExtension = LEER.next();  
    }  
  
    public void Mostrar ()  
    {  
        System.out.printf("Nombre: %s, Apellido: %s, NumeroCI: %d, ExtensionCI: %s",primerNombre,primerApellido,ciNumero,ciExtension);  
    }  
}
```

Gerente

```
public class Gerente extends Empleado {  
  
    public int nroID;  
    public String codArea;  
    public int sueldoBasico;  
  
    public Gerente() {  
        super();  
    }  
  
    public void leer() {  
        System.out.printf("Ingrese Primer Nombre :");  
        primerNombre = LEER.next();  
        System.out.printf("Ingrese Primer Apellido :");  
        primerApellido = LEER.next();  
        System.out.printf("Ingrese Numero CI :");  
        ciNumero = LEER.nextInt();  
        System.out.printf("Ingrese Extencio CI :");  
        ciExtension = LEER.next();  
        System.out.printf("Ingrese Numero de ID :");  
        nroID = LEER.nextInt();  
    }  
}
```

```

        System.out.printf("IngreseCodigo de Area :");
        codArea = LEER.next();
        System.out.printf("Ingrese Sueldo Basico :");
        sueldoBasico = LEER.nextInt();

    }

    public void Mostrar ()
    {
        System.out.printf("Nombre: %s, Apellido: %s, NumeroCI: %d, ExtensionCI: %s,
NroID: %d, CodArea: %s, SueldoBasico:
%d",primerNombre,primerApellido,ciNumero,ciExtension,nroID,codArea,sueldoBasico);
    }

}

```

Supervisor

```

public class Supervisor extends Empleado {

    public int sueldo;
    public int antiguedad ;

    public Supervisor() {
        super();
    }

    public void leer() {
        System.out.printf("Ingresar sueldo:");
        sueldo = LEER.nextInt();
        System.out.printf("Ingresar antiguedad:");
        antiguedad = LEER.nextInt();

    }

    public void mostrar() {
        System.out.printf("Sueldo:%d, Antiguedad:%d",sueldo,antiguedad);

    }

}

```

Oficinista

```

public class Oficinista extends Empleado {

    public String codArea;
    public int sueldoBasico;

    public Oficinista() {
        super();
    }

    public void leer() {
        System.out.printf("Ingresar codArea:");
        codArea = LEER.next();
        System.out.printf("Ingresar sueldoBasico:");
        sueldoBasico = LEER.nextInt();

    }

    public void mostrar() {
        System.out.printf("codArea:%s, sueldoBasico:%d",codArea,sueldoBasico);

    }

}

```

Secretaria

```
public class Secretaria extends Oficinista {  
  
    public String nombreArea;  
  
    public Secretaria() {  
        super();  
    }  
    public void leer() {  
        System.out.printf("Ingresar nombreArea:");  
        nombreArea = LEER.next();  
    }  
    public void mostrar() {  
        System.out.printf("nombreArea:%s", nombreArea);  
    }  
}
```

Mensajero

```
public class Mensajero extends Oficinista {  
  
    public String turno;  
  
    public Mensajero() {  
        super();  
    }  
    public void leer() {  
        System.out.printf("Ingresar turno:");  
        turno = LEER.next();  
    }  
    public void mostrar() {  
        System.out.printf("turno:%s", turno);  
    }  
}
```

Main

```
public class Main {  
    public static void main(String[] args) {  
  
        Gerente g = new Gerente();  
  
        g.leer();  
        g.Mostrar();  
    }  
}
```