



DEFENSA HITO 2 - TAREA FINAL

Nombre Completo: **Univ. Juan Elían Álvarez Vallejos**

Asignatura: **PROGRAMACIÓN III**

Carrera: **INGENIERÍA DE SISTEMAS**

Paralelo: **PROG (1)**

Docente: **Lic. William R. Barra Paredes**

fecha: **30/03/2020**

github: **LINK de la carpeta hito2 de GIT**

PARTE TEORICA

Defina y muestre ejemplos de la clase Scanner.

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

```
Scanner leer = new Scanner(System.in);
//Para Leer cadenas
System.out.println("Ingrese nombre");
String miNombre = leer.nextLine();
System.out.println("Ingrese apellido");
String miApellido = leer.nextLine();

System.out.printf("Su nombre completo es: %s
%s",miNombre,miApellido);System.out.println();

System.out.println("Ingrese su edad");
int miEdad = leer.nextInt();
System.out.printf("Su edad es: %d",miEdad);
//sumar Las primeros 5 edades
int sumaEdades=0;
double promedio=0;
int cont=0;
System.out.println("Ingrese su edad");
while (cont <5)
{
    int miEdad2 = leer.nextInt();
    sumaEdades= sumaEdades + miEdad2;
    System.out.println("Ingrese su edad");
    cont ++;
}
```

Que es la programación orientada a objetos (POO).

R. La programación orientada a objetos (POO) es un paradigma de programación que usa objetos para crear aplicaciones. Está basada en tres pilares fundamentales: herencia, polimorfismo, encapsulación.

Cuál es la diferencia entre interfaz y herencia.

R. La herencia es un *mecanismo* para extender las funcionalidades y atributos de una clase. Una interfaz es un tipo especial de clase (*que también puede tener herencia*) que *no posee funcionalidad implementada*, solo define un conjunto de funcionalidades para las clases que la *implementen* (un conjunto de métodos con parámetros pero sin código).

Qué elementos crees que definen a un objeto.

R. Sus usos, las formas y las propiedades que los caracterizan.

Que es una clase abstracta y muestre un ejemplo.

R. Las clases abstractas 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 Empleado(String nombre1, String ci1, int edad1){  
    this.nombre = nombre1;  
    this.ci = ci1;  
    this.edad = edad1;  
}
```

EJERCICIO 1

```
public String SerieFibonacci(int n)
{

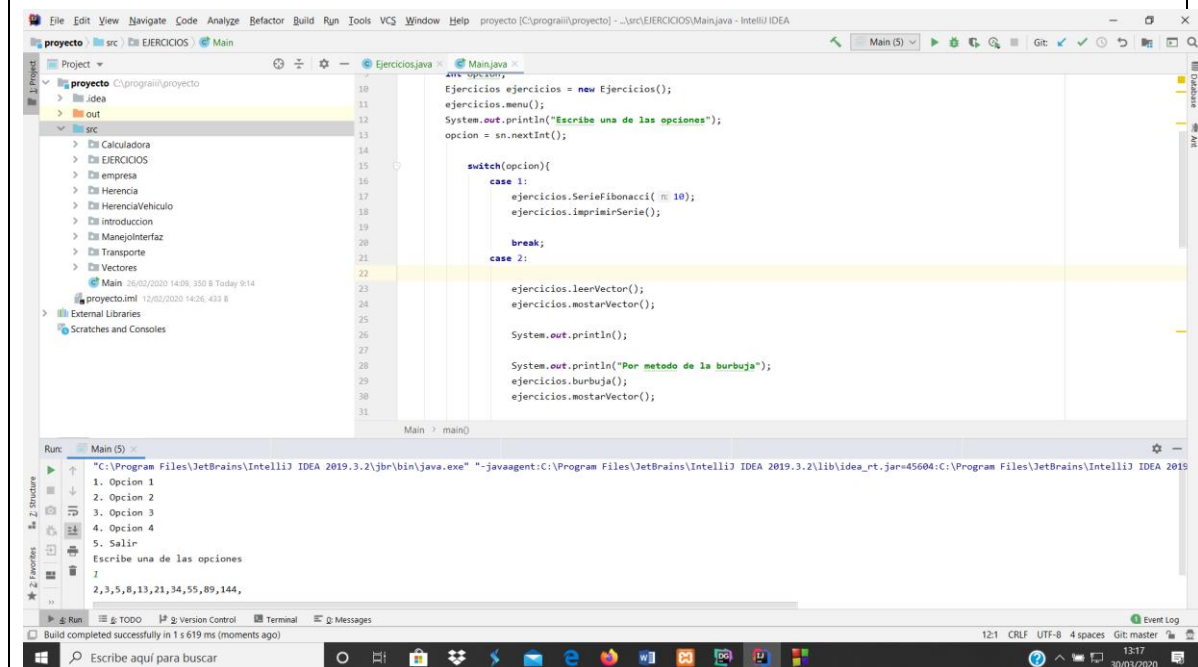
    int conta = 0;
    int a=1,b=1;
    int auxi = 0;

    while (conta < n)
    {
        auxi = a + b;
        resultadoSerie = resultadoSerie + String.valueOf(auxi) + ",";
        a = b;
        b = auxi;
        conta ++;
    }
    return resultadoSerie;
}

public void imprimirSerie(){
    System.out.println(resultadoSerie);
}

case 1:
    ejercicios.SerieFibonacci(10);
    ejercicios.imprimirSerie();

    break;
```



EJERCICIO 2

```
public void seleccion(){
    int i, j, menor, pos, tmp;
    for (i = 0; i < vec.length - 1; i++) { // tomamos como menor el primero
        menor = vec[i]; // de los elementos que quedan por ordenar
        pos = i; // y guardamos su posición
        for (j = i + 1; j < vec.length; j++){ // buscamos en el resto
            if (vec[j] < menor) { // del array algún elemento
                menor = vec[j]; // menor que el actual
                pos = j;
            }
        }
        if (pos != i){ // si hay alguno menor se intercambia
            tmp = vec[i];
            vec[i] = vec[pos];
            vec[pos] = tmp;
        }
    }
}
```

case 2:

```
ejercicios.leerVector();
ejercicios.mostrarVector();

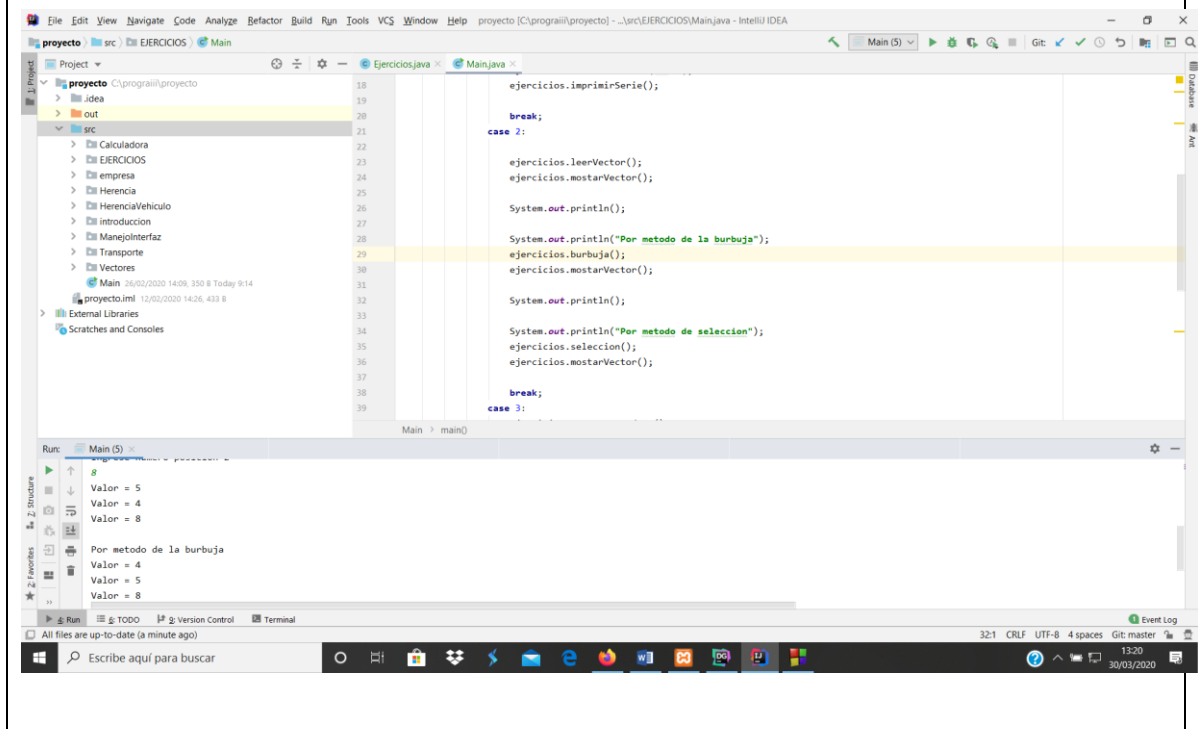
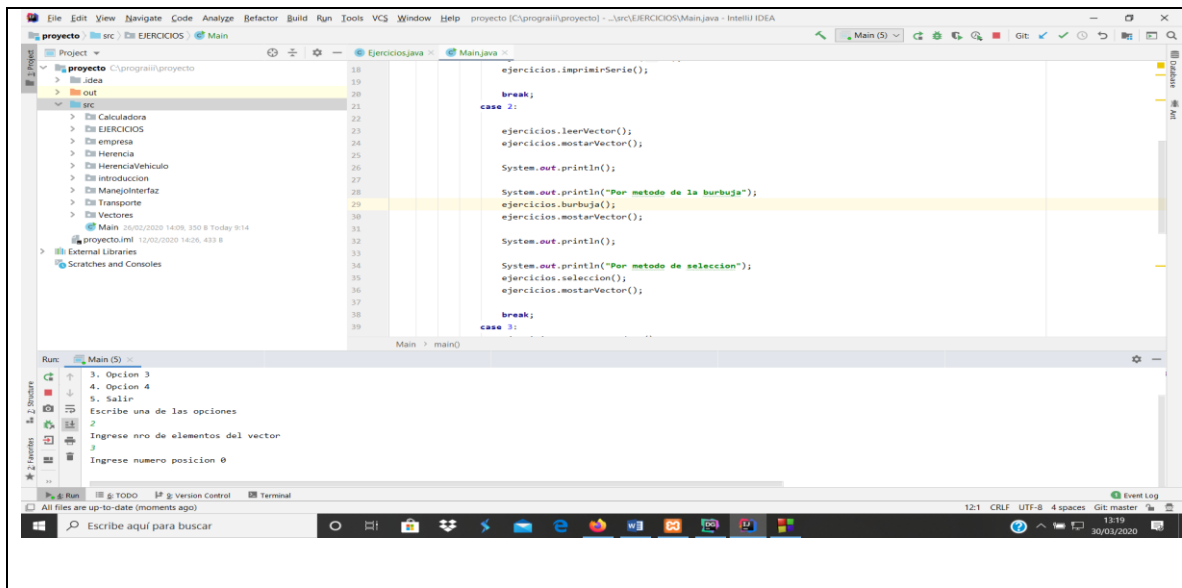
System.out.println();

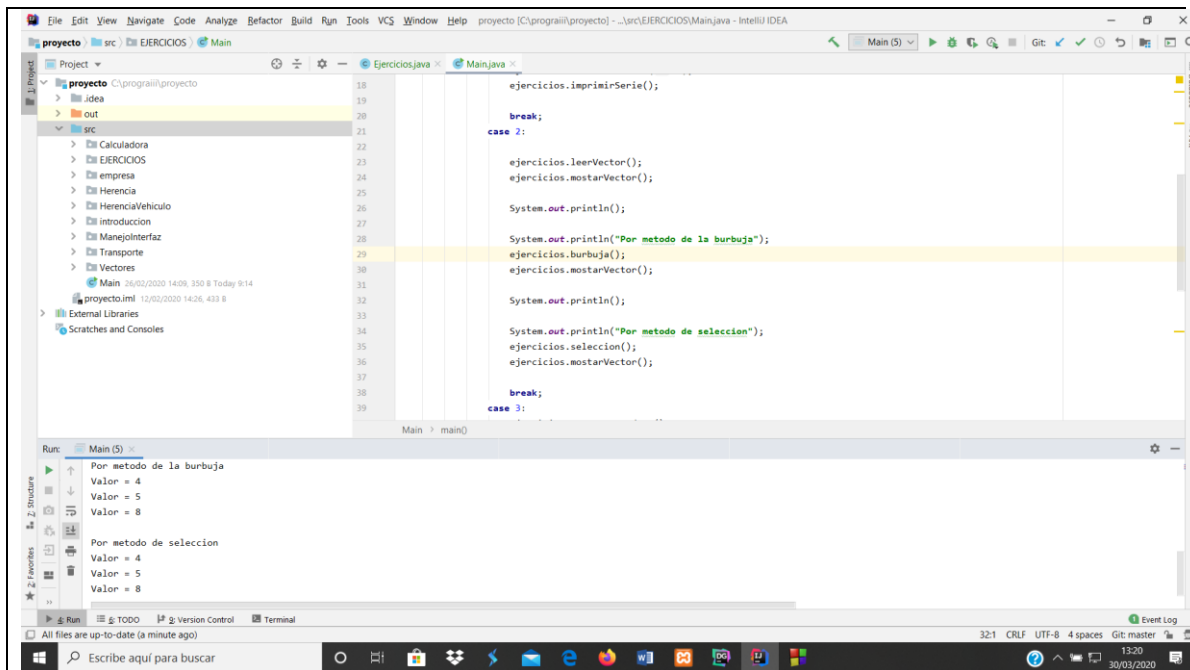
System.out.println("Por metodo de la burbuja");
ejercicios.burbuja();
ejercicios.mostrarVector();

System.out.println();

System.out.println("Por metodo de seleccion");
ejercicios.seleccion();
ejercicios.mostrarVector();

break;
```



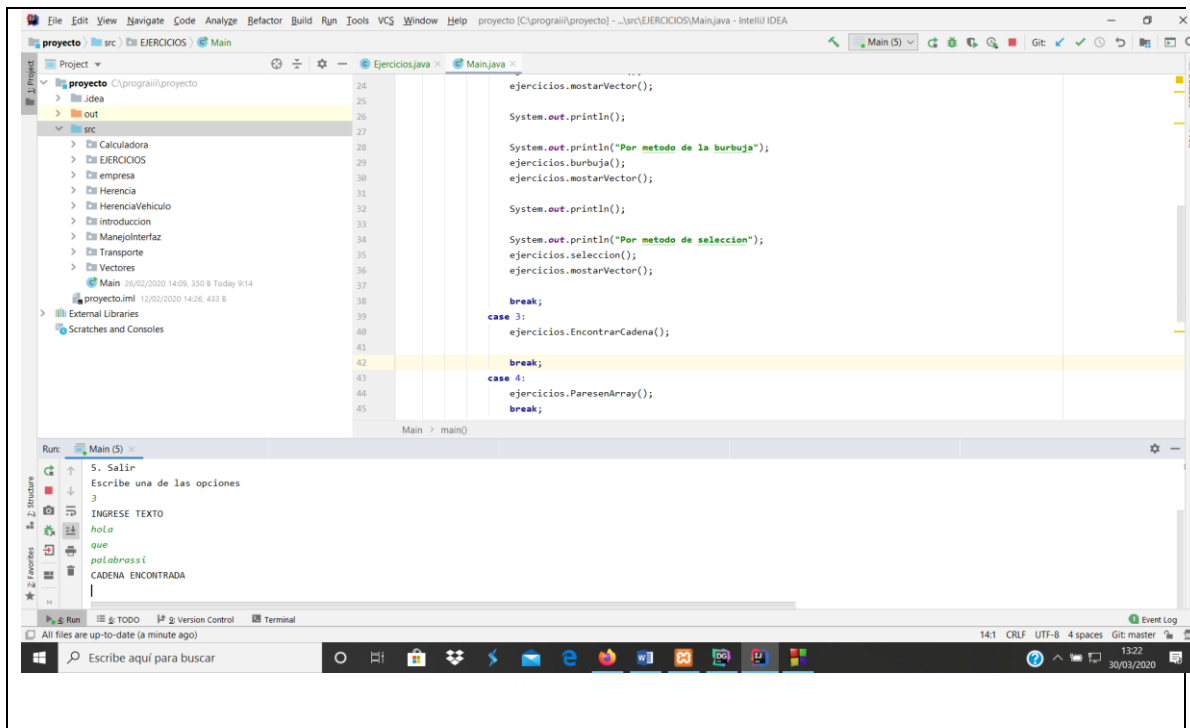


EJERCICIO 3

```
public void EncontrarCadena(){
    System.out.println("INGRESE TEXTO");
    while(leer.hasNext()){
        String cadena = leer.nextLine();
        if(cadena.length() == 10){
            System.out.println("CADENA ENCONTRADA");
        }
    }
}

case 3:
    ejercicios.EncontrarCadena();

    break;
```



EJERCICIO 4

```
private Integer array[];
public void ParesenArray(){

    System.out.println("ingrese el tamaño para el array");
    int tamaño = leer.nextInt();
    array = new Integer[tamaño];

    for(int i=0;i<tamaño;i++){
        System.out.printf("Ingrese los valores del array %d\n",i);
        int numeroLeido = leer.nextInt();
        array[i] = numeroLeido;
    }
    for (int i = 0; i < 10; i++) {
        if( array[i] % 2 == 0){
            System.out.println("LOS NUMEROS PARES DE LA MATRIZ SON: " +
array[i]);

        }
    }
}

case 4:
    ejercicios.ParesenArray();
    break;
```