|  |
| --- |
| **UNIVERSIDAD PRIVADA FRANZ TAMAYO**  **DEFENSA HITO 2 - TAREA FINAL**  Nombre Completo: **José Antonio Secko Beltrán**  Asignatura: **PROGRAMACIÓN III**  Carrera: **INGENIERÍA DE SISTEMAS**  Paralelo: **PROG (3)**  Docente: **Lic. William R. Barra Paredes**  fecha: **30/03/2020**  GitHub: **https://github.com/jhorts689/prograiii.git** |

1. Preguntas.

Responda de manera breve y clara posible.

* Defina y muestre ejemplos de la clase Scanner.

Scanner de java es una librería que permite ingresar datos por el usuario.

import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.println("Ingrese numero");  
 int num = scanner.nextInt();  
 System.*out*.println(num);  
 }  
}

* Que es la programación orientada a objetos(POO).

POO es una forma más de programar que ayuda a organizar la forma de programar.

* Cuál es la diferencia entre interfaz y herencia.

En una interfaz se crean métodos, que llegaran a implementarse en una clase donde se podrá realizar las operaciones que hará dicho método mientras que la herencia permite heredar los métodos de otras clases.

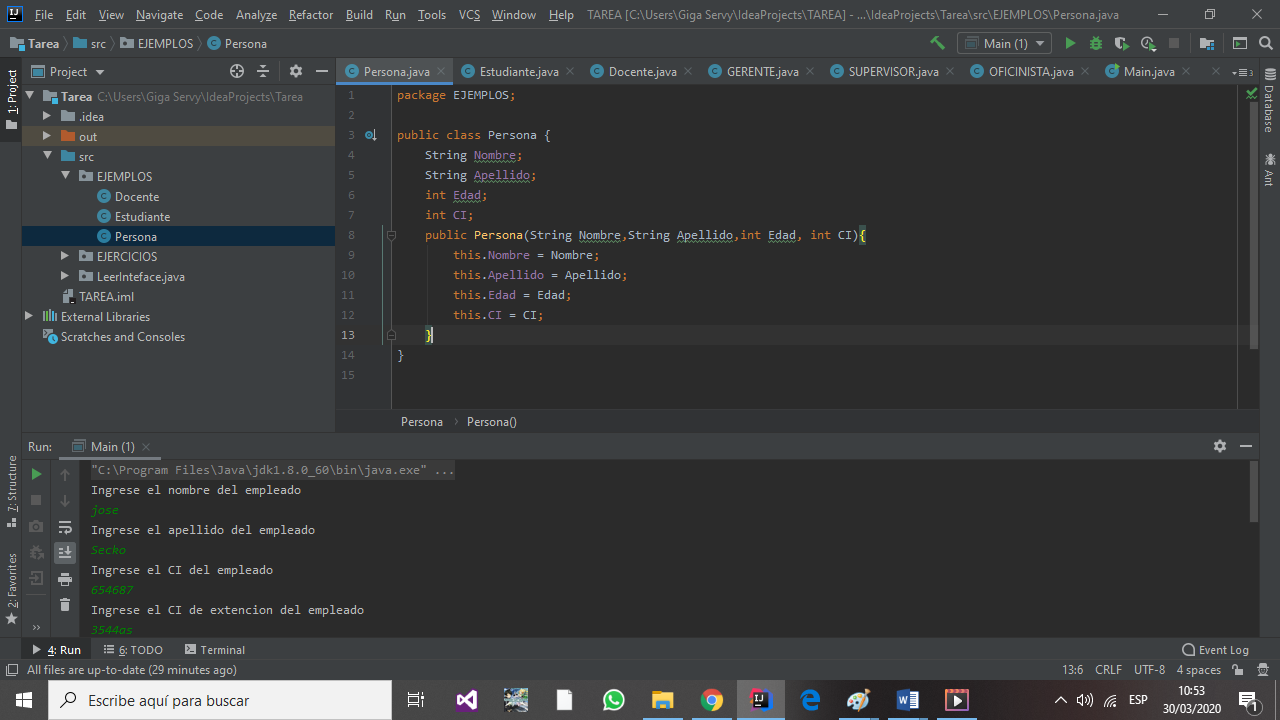
* Qué elementos crees que definen a un objeto.

Los atributos y métodos.

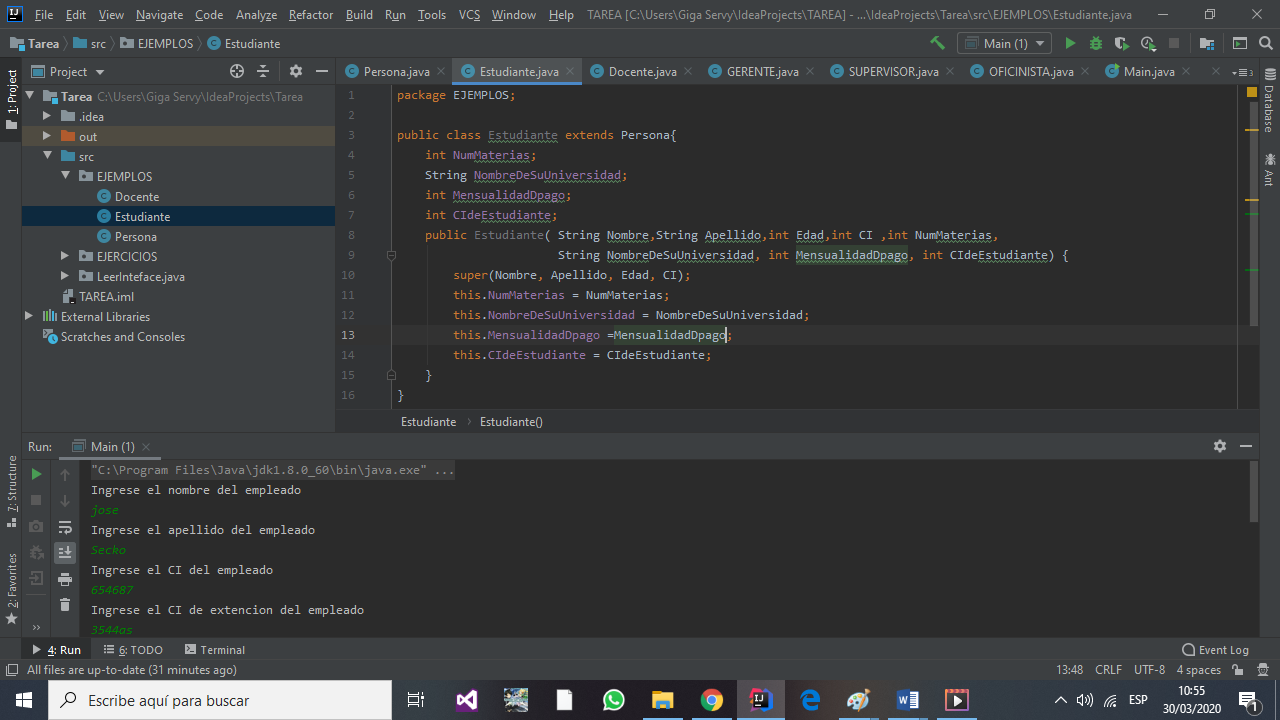
* Que es una clase abstracta y muestre un ejemplo.

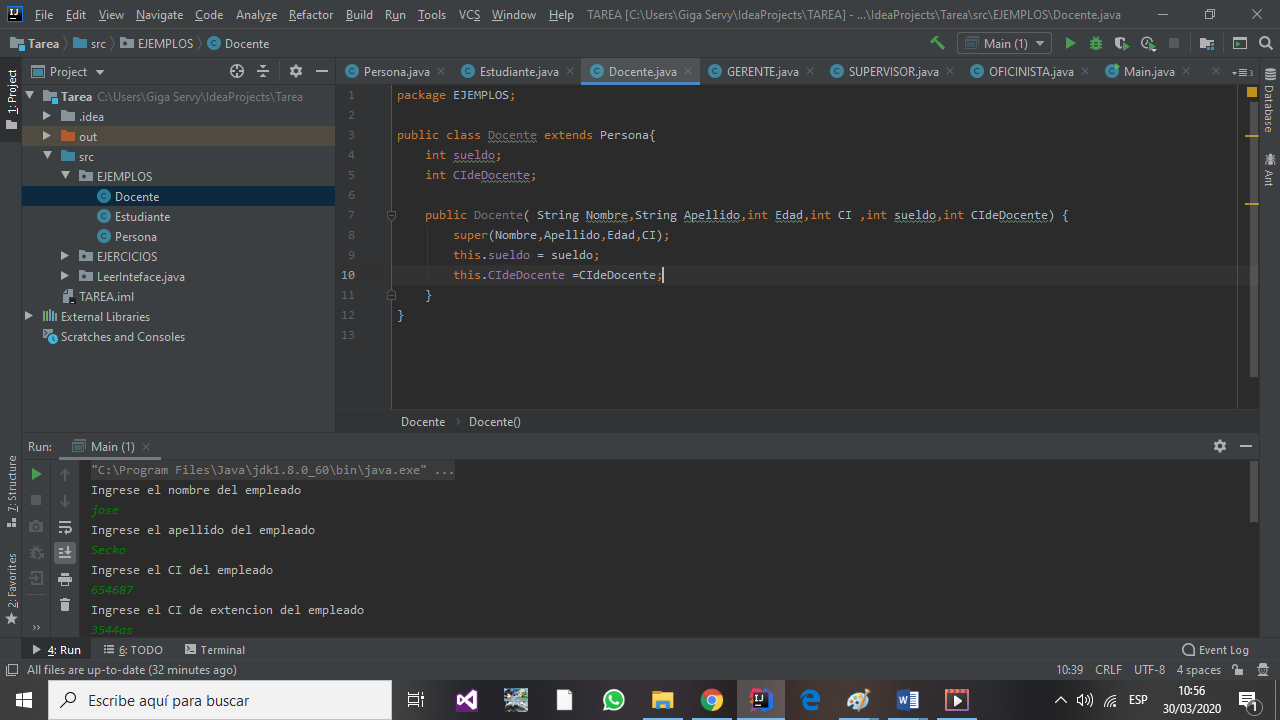
Una clase abstracta es aquella de la que podemos utilizar para crear otras clases y heredar sus métodos, pero no nos permite crear objetos de una clase abstracta.

//CLASE PADRE 0 CLASE ABSTRACTA



//CLASE HIJA O HEREDA





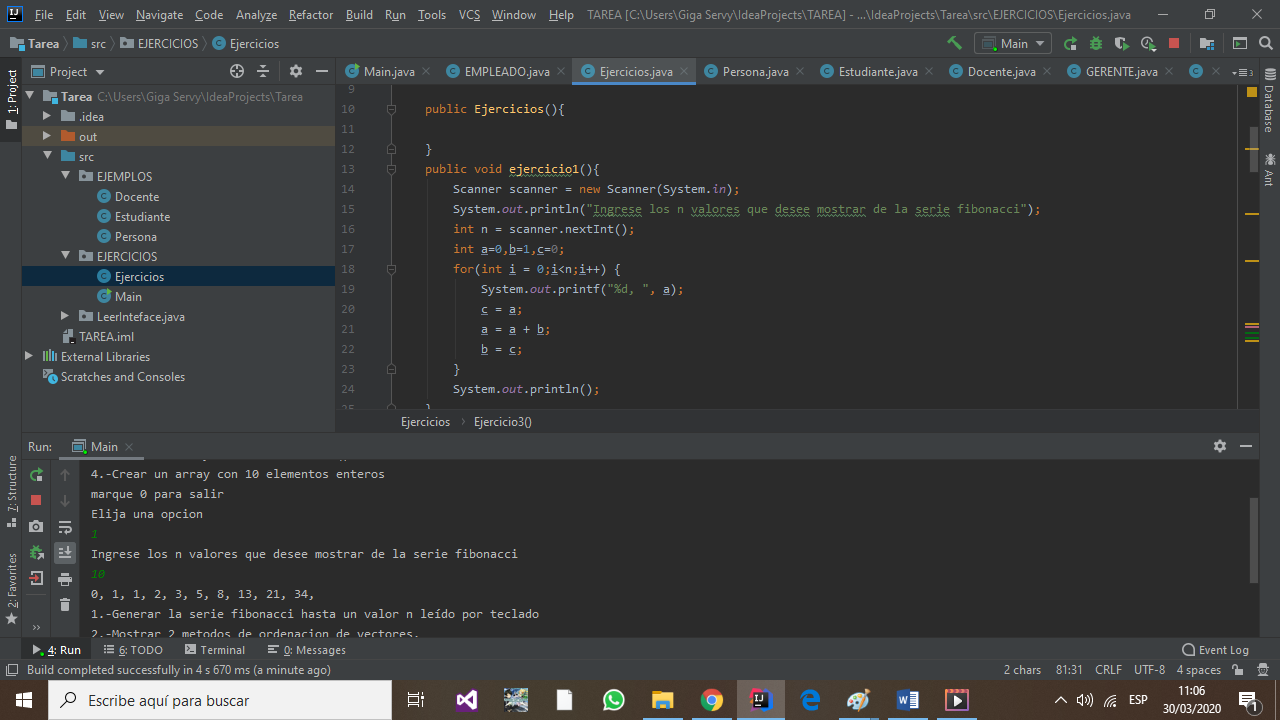
## Ejercicios.

Debe de trabajarlos en el package EJERCICIOS.

* Crear un PACKAGE de nombre EJERCICIOS.
* Crear una clase de nombre Ejercicios.java.
* Crear la clase MAIN para mostrar las soluciones.

## Preguntas ejercicios.

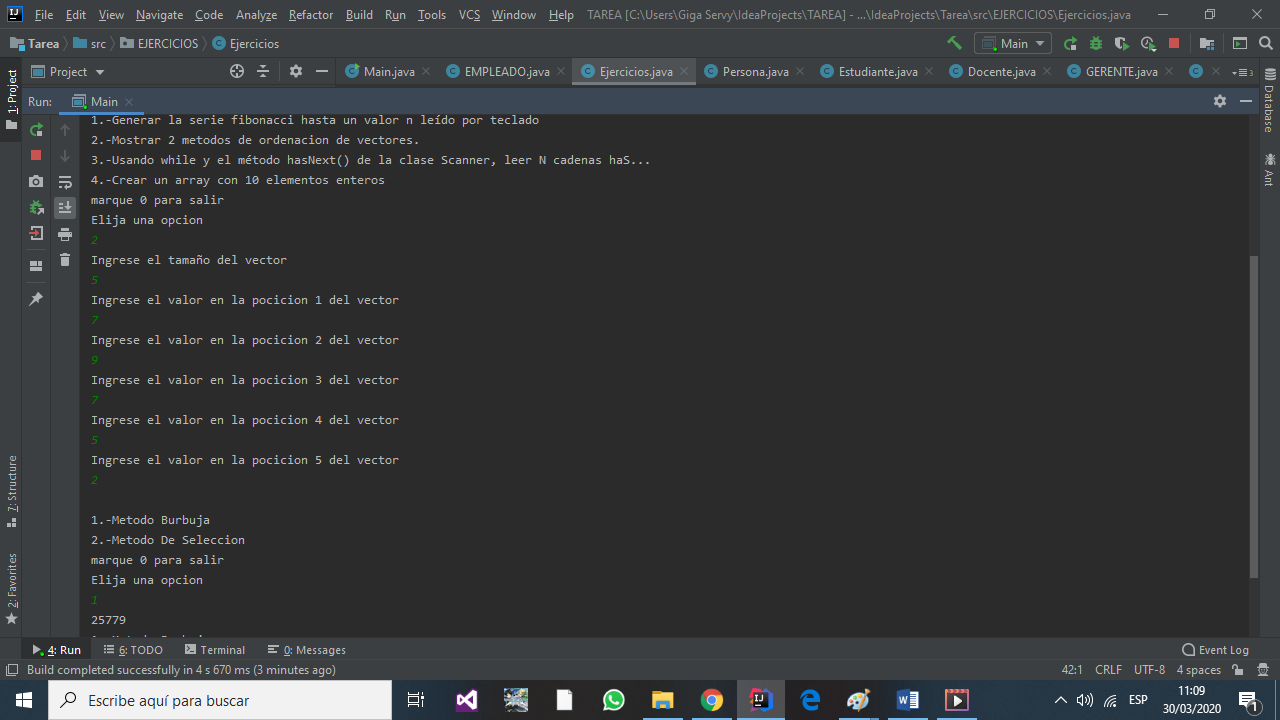
* + Generar la serie **fibonacci** hasta un valor n leído por teclado



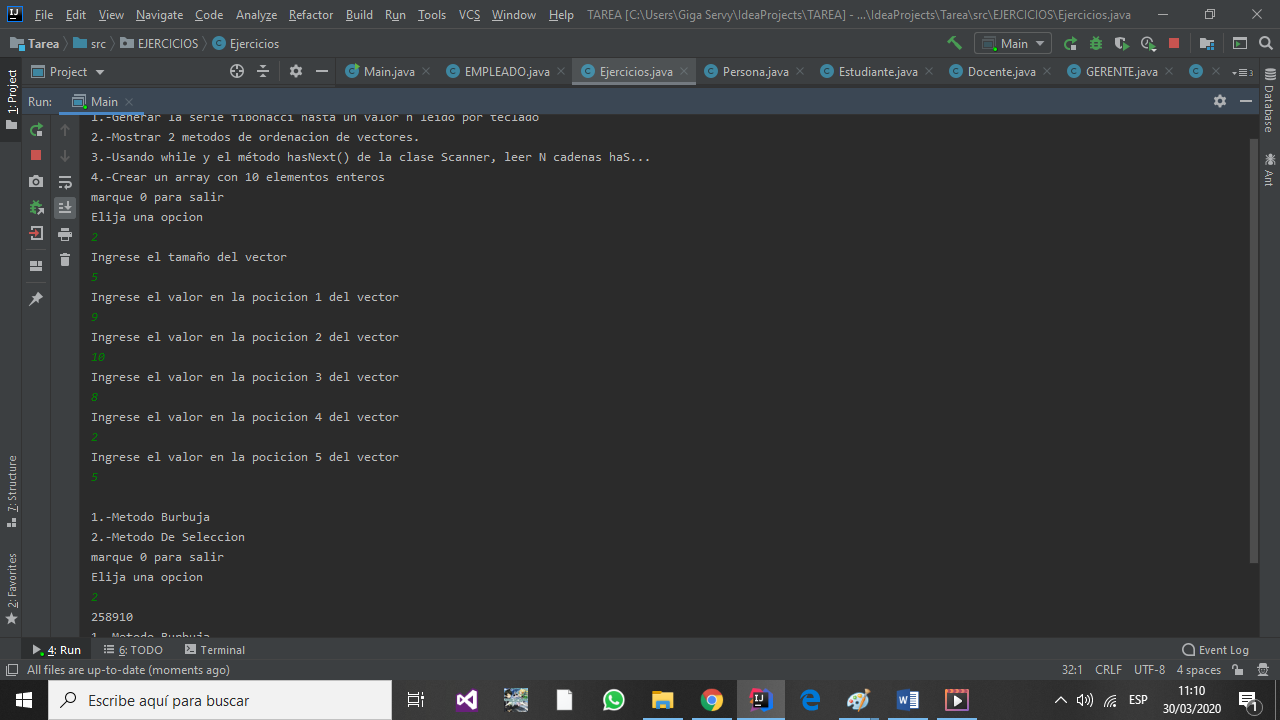
Scanner scanner = new Scanner(System.*in*);  
System.*out*.println("Ingrese los n valores que desee mostrar de la serie fibonacci");  
int n = scanner.nextInt();  
int a=0,b=1,c=0;  
for(int i = 0;i<n;i++) {  
 System.*out*.printf("%d, ", a);  
 c = a;  
 a = a + b;  
 b = c;  
}  
System.*out*.println();

* + Mostrar 2 metodos de ordenacion de vectores.
    - Puede ser burbuja el método por selección.

//método Burbuja

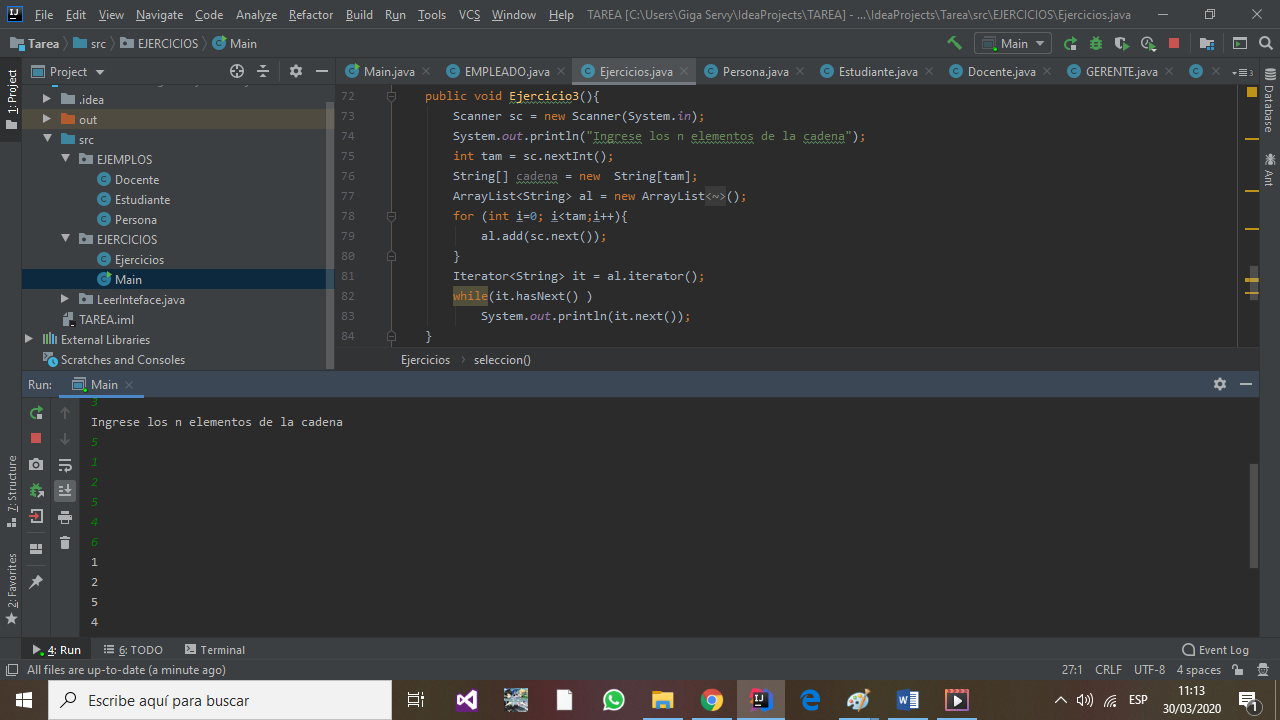


//método de selección



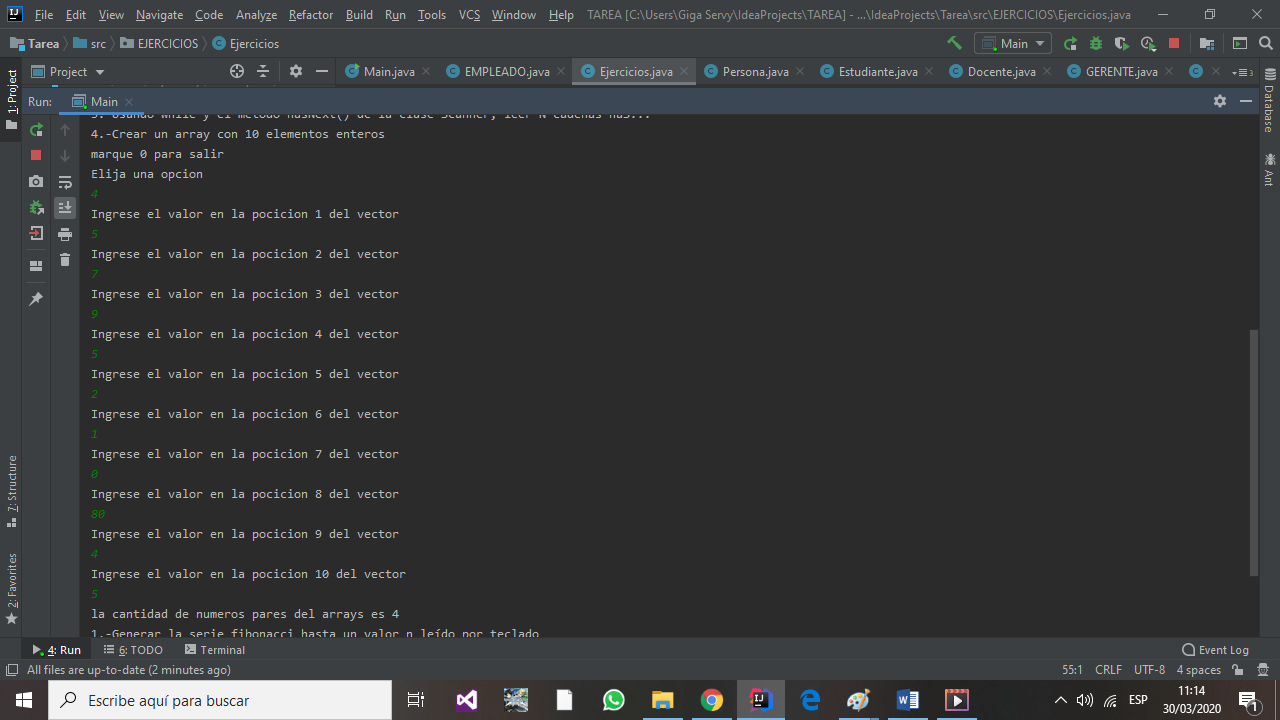
public void Ingresar(){  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Ingrese el tamaño del vector");  
 tam = sc.nextInt();  
 vector = new int[tam];  
 for (int i=0; i< tam;i++){  
 System.*out*.println("Ingrese el valor en la pocicion " + (i+1)+" del vector");  
 vector[i]= sc.nextInt();  
 }  
 System.*out*.println();  
}  
public void Burbuja(){  
  
 int aux=0;  
 for(int i=0;i<=tam;i++){  
 for(int j=0;j<tam-1;j++){  
 if(vector[j]>vector[j+1]){  
 aux=vector[j];  
 vector[j]=vector[j+1];  
 vector[j+1]=aux;  
 }  
 }  
 }  
 for(int i=0;i<tam;i++){  
 System.*out*.printf("%d",vector[i]);  
 }  
 System.*out*.println();  
}  
public void seleccion(){  
 int aux=0,minimo=0;  
 for(int i=0;i<tam;i++){  
 minimo=i;  
 for(int j=i+1;j<tam;j++){  
 if(vector[minimo] > vector[j]) {  
 aux = vector[minimo];  
 vector[i] = vector[j];  
 vector[j] = aux;  
 }  
 }  
 }  
  
 for(int i=0;i<tam;i++){  
 System.*out*.printf("%d",vector[i]);  
 }  
 System.*out*.println();  
}

* + 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.



public void Ejercicio3(){  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Ingrese los n elementos de la cadena");  
 int tam = sc.nextInt();  
 String[] cadena = new String[tam];  
 ArrayList<String> al = new ArrayList<String>();  
 for (int i=0; i<tam;i++){  
 al.add(sc.next());  
 }  
 Iterator<String> it = al.iterator();  
 while(it.hasNext() )  
 System.*out*.println(it.next());  
}

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



public void ejercicio4(){  
 int[] vector = new int[10];  
 int cont=0;  
 for(int i=0; i<10; i++){  
 Scanner sc = new Scanner(System.*in*);  
 System.*out*.println("Ingrese el valor en la pocicion " + (i+1)+" del vector");  
 vector[i] = sc.nextInt();  
 if ((vector[i]%2==0)){  
 cont++;  
 }  
 }  
 System.*out*.println("la cantidad de numeros pares del arrays es " + (cont));  
}

# Parte Práctica.

## 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);**

## Manejo de Herencia: Empleado.java

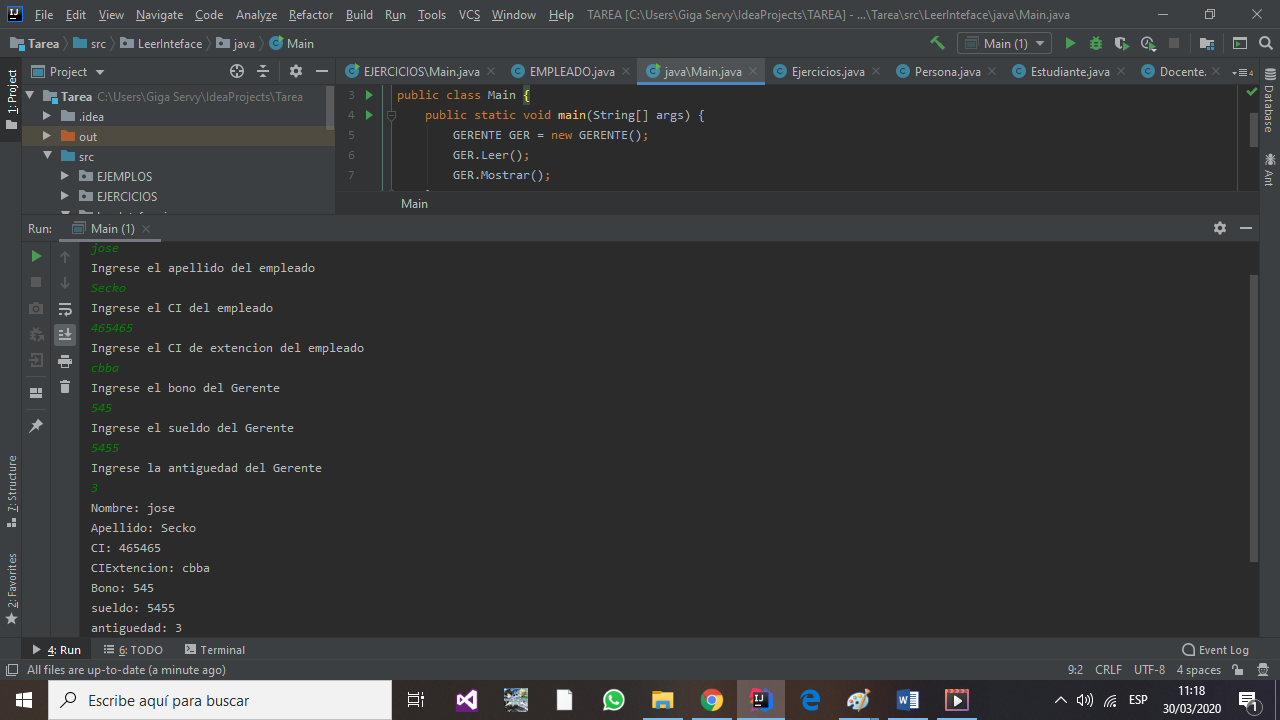
Esta es la clase **padre** del cual heredan otras subclases. Esta clase implementa la interfaz **LeerInterface.java.**

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

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

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

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



//interfaz:

package LeerInteface.java;  
  
import java.util.Scanner;  
  
public interface LeerInteface {  
 Scanner *LEER* = new Scanner(System.*in*);  
 EMPLEADO *Empleado* = new EMPLEADO();  
}

//Clase Abstracta : EMPLEADO

public class EMPLEADO implements LeerInteface{  
 String PrimerNombre;  
 String PrimerApellido;  
 int CINumero;  
 String CIExtencion;  
 public EMPLEADO(){  
 }  
 public void Leer() {  
 System.*out*.println("Ingrese el nombre del empleado");  
 PrimerNombre = *LEER*.next();  
 System.*out*.println("Ingrese el apellido del empleado");  
 PrimerApellido = *LEER*.next();  
 System.*out*.println("Ingrese el CI del empleado");  
 CINumero = *LEER*.nextInt();  
 System.*out*.println("Ingrese el CI de extencion del empleado");  
 CIExtencion = *LEER*.next();  
 }  
 public void Mostrar(){  
 System.*out*.println("Nombre: "+ PrimerNombre);  
 System.*out*.println("Apellido: "+ PrimerApellido);  
 System.*out*.println("CI: "+ CINumero);  
 System.*out*.println("CIExtencion: "+ CIExtencion);  
 }  
}

//CLASE gerente

public class GERENTE extends EMPLEADO{  
 int bono,sueldo,antiguedad;  
  
 public GERENTE(){  
  
 }  
 public void Leer(){  
  
 *Empleado*.Leer();  
 System.*out*.println("Ingrese el bono del Gerente");  
 bono = *LEER*.nextInt();  
 System.*out*.println("Ingrese el sueldo del Gerente");  
 sueldo = *LEER*.nextInt();  
 System.*out*.println("Ingrese la antiguedad del Gerente");  
 antiguedad = *LEER*.nextInt();  
 }  
 public void Mostrar(){  
 *Empleado*.Mostrar();  
 System.*out*.println("Bono: "+ bono);  
 System.*out*.println("sueldo: "+ sueldo);  
 System.*out*.println("antiguedad: "+ antiguedad);  
 }

//CLASE SUPERVISOR

public class SUPERVISOR extends EMPLEADO{  
 int sueldo,antiguedad;  
 public SUPERVISOR(){  
  
 }  
 public void Leer(){  
 *Empleado*.Leer();  
 System.*out*.println("Ingrese el sueldo del Supervisor");  
 sueldo = *LEER*.nextInt();  
 System.*out*.println("Ingrese la antiguedad del Supervisor");  
 antiguedad = *LEER*.nextInt();  
 }  
 public void Mostrar(){  
 *Empleado*.Mostrar();  
 System.*out*.println("sueldo: "+ sueldo);  
 System.*out*.println("antiguedad: "+ antiguedad);  
 }  
}

//CLASE OFICINISTA

public class OFICINISTA extends EMPLEADO{  
 String codArea;  
 int SueldoBasico;  
 OFICINISTA oficinista = new OFICINISTA();  
 public OFICINISTA(){  
  
 }  
 public void Leer(){  
 *Empleado*.Leer();  
 System.*out*.println("Ingrese el Codigo de Area del Oficinista");  
 codArea = *LEER*.next();  
 System.*out*.println("Ingrese el sueldo basico del Oficinista");  
 SueldoBasico = *LEER*.nextInt();  
 }  
 public void Mostrar(){  
 *Empleado*.Mostrar();  
 System.*out*.println("Cod. de Area: "+ codArea);  
 System.*out*.println("Sueldo Basico: "+ SueldoBasico);  
 }  
}

//CLASE SECRETARIA HIJA DE OFICINISTA

public class SECRETARIA extends OFICINISTA {  
 String NombreArea;  
 public SECRETARIA(){  
  
 }  
 public void Leer(){  
 oficinista.Leer();  
 System.*out*.println("Ingrese el Nombre de Area del Oficinista");  
 NombreArea = *LEER*.next();  
 }  
 public void Mostrar(){  
 oficinista.Mostrar();  
 System.*out*.println("Nombre Del Area: "+ NombreArea);  
 }  
}

//CLASE MENSAJERO HIJA DE OFICINISTA

public class MENSAJERO extends OFICINISTA{  
 String turno;  
 public MENSAJERO(){  
  
 }  
 public void Leer(){  
 oficinista.Leer();  
 System.*out*.println("Ingrese el turno en el que trabaja el Oficinista");  
 turno = *LEER*.next();  
 }  
 public void Mostrar(){  
 oficinista.Mostrar();  
 System.*out*.println("Turno: "+ turno);  
 }  
}

//MAIN

public class Main {  
 public static void main(String[] args) {  
 GERENTE GER = new GERENTE();  
 GER.Leer();  
 GER.Mostrar();  
 }  
}