



Servicio Nacional de Aprendizaje – SENA

Centro de Diseño y Metrología

InformeProgramasRealizados

Aprendiz:

Juan Esteban Castro Escamilla

No Ficha 2848530-A

Tabla de contenido

Tabla de contenido	2
Estructurado y Orientado a objetos	3
Programación Estructurada:	3
Informe Programas Estructurados	4
Programa n-1 : NotasEstructurada - PROGRAMA REALIZADO EN CLASE	4
Programa n-2 : EmpresaEstructurada - PROGRAMA REALIZADO EN CLASE	5
Programa n-3 : EstudianteEstructurada - PROGRAMA TRABAJO EN CASA	8
Programa n-4 : EmpleadoEstructurada - PROGRAMA TRABAJO EN CASA	10
Informe Programas POO	12
Programa n-1 : NotasObjetos - PROGRAMA REALIZADO EN CLASE	12
Programa n-2 : EmpresaObjetos - PROGRAMA REALIZADO EN CLASE	14
Programa n-3 : Biblioteca - PROGRAMA REALIZADO EN CLASE (NO SE TERMINÓ, FUE UN EJEMPLO)	18
Programa n-4 : EstudiantePOO - PROGRAMA REALIZADO EN CASA	20
Programa n-5 : EmpleadosPOO - PROGRAMA REALIZADO EN CASA	23

Estructurado y Orientado a objetos

Programación Estructurada:

- Se basa en el uso de funciones o procedimientos, que son bloques de código organizados secuencialmente.
- El flujo de control sigue una estructura lineal o jerárquica, con instrucciones que se ejecutan de forma secuencial, con bifurcaciones (condicionales) y repeticiones (bucles).
- El objetivo principal es dividir el programa en subrutinas (funciones o procedimientos) que realicen tareas específicas.

Programación Orientada a Objetos (POO):

- Se organiza en objetos y clases. Una clase es una plantilla o molde que define las propiedades y comportamientos comunes de los objetos, mientras que los objetos son instancias de esas clases.
- El enfoque principal es modelar el mundo real mediante entidades (objetos) que tienen propiedades (atributos) y comportamientos (métodos).
- La idea es que los objetos interactúan entre sí mediante métodos, y el programa se estructura en torno a estos objetos y sus relaciones.

Informe Programas Estructurados

Programa n-1 : NotasEstructurada - **PROGRAMA REALIZADO EN CLASE**

(Antes de iniciar la clase en donde trabajaremos, vamos a importar la utilidad **JOptionPane** la cual nos servirá para poder solicitar por pantalla al usuario que ingrese los diferentes datos que se pedirán mas adelante, cabe recalcar que esto aplicará para todos los ejercicios que se verán despues que importe esta utilidad)

```
import javax.swing.JOptionPane;
```

// Usamos la palabra reservada import para utilizar la herramienta

(Creamos la clase pública con su respectivo nombre, luego se creará una estructura repetitiva FOR la cual será controlada por una variable llamada est que se le asigna el número 1 y se repetirá hasta que sea igual o menor a 2 es decir, 1 vez mas en este caso) // el ++ permite que se itere de 1 en 1)

(Se crean allí las variables de texto nombre y de decimal (Float) nota1, nota2 y notaFinal)
(Gracias a la utilidad OptionPane podremos mostrar un diálogo de entrada para asignar los valores que allí nos dice tales como nombre del estudiante y sus notas)

(Finalizando la asignación de los datos va a realizar una función (promedio de las notas 2 obtenidas) en la variable notaFinal y se guardará en esta)

(Por último gracias al estructura de un IF básico, nos permitirá evaluar si la nota final es mayor o igual a 3 y de esta manera si se cumple dicha función mostrará un mensaje de aprobado para el alumno y si no, mostrara un mensaje que el estudiante reprobó)

```
public class NotasEstructurada {  
    Run main | Debug main | Run | Debug  
    public static void main(String[] args) throws Exception {  
        for(int est = 1; est <= 2; est++) {  
            String nombre = JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese nombre del estudiante");  
            float nota1 = Float.parseFloat(JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese la nota 1 del estudiante"));  
            float nota2 = Float.parseFloat(JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese la nota 2 del estudiante"));  
            float notaFinal = (nota1+nota2)/2;  
            JOptionPane.showMessageDialog(parentComponent:null, "Informacion del estudiante/nNombre : " + nombre + " Nota Final : " + notaFinal);  
            if(notaFinal >=3) {  
                JOptionPane.showMessageDialog(parentComponent:null, message:"Aprobo");  
            }else {  
                JOptionPane.showMessageDialog(parentComponent:null, message:"Reprobo");  
            }  
        }  
    }  
}
```

Programa n-2 : EmpresaEstructurada - **PROGRAMA REALIZADO EN CLASE**

(Antes de iniciar el programa como se comentó anteriormente, se utilizara la utilidad JOptionPane y JPanel de la misma manera, en este caso en concreto, se agregó la utilidad ArrayList para poder crear un arreglo que se podrá observar después de esta imagen)

```
import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
```

(Utilidad ArrayList permite crear diferentes arreglos para lo que necesitemos, en este caso, se creó 3 diferentes arreglos para nombres, cargos y salarios de los empleados con sus respectivos tipo de dato)

```
ArrayList<String> nombres = new ArrayList<>();
ArrayList<String> cargos = new ArrayList<>();
ArrayList<Double> salarios = new ArrayList<>();
```

(Creamos una variable de tipo int que se llamó cantidad de la misma forma con la utilidad JOptionPane nos permitirá asignar el valor de la cantidad en este caso de los empleados)

(Luego se creo la estructura repetitiva FOR con su variable emp iniciada en 0 y que esta se cumplira hasta que sea menor a la cantidad es decir, al valor que ya se asignó anteriormente)

(Mismo proceso de pedir la asignación de los datos de las variables nombre, cargo y salario, con la diferencia que al esto ser un programa con Arreglos, al final del FOR se agregara dichos valores en sus respectivos arreglos y posiciones=

```
int cantidad = Integer.parseInt(JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese la cantidad de
empelados:"));

for(int emp = 0; emp < cantidad; emp++) {
    String nombre = JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese nombre del empleado:");
    String cargo = JOptionPane.showInputDialog(parentComponent:null, message:"Ingrese el cargo del empleado:");
    Double salario = Double.parseDouble(JOptionPane.showInputDialog (parentComponent:null, message:"Ingrese el salario
del empleado:"));

    nombres.add(nombre);
    cargos.add(cargo);
    salarios.add(salario);
}
```

// UN ARREGLO SE CUENTA DESDE EL NUMERO 0, TENER MUY EN CUENTA ESO

(Se muestra por pantalla con el println los nombres y salarios de cada empleado, gracias a una estructura FOR la cual recorrerá los arreglos de nombres y salarios)

(Luego vamos a mostrar por pantalla y/o consola el total de los salarios pagados, esto se nos ha permitido gracias a la función FOR que almacenara en la variable total que se ha creado cada uno de los salarios y estos se sumarán entre si, gracias a la auxiliar total que se suma así misma cargando cada salario y sumandolo)

```
System.out.println(x:"Los nombre y los salarios de los empleados son: ");
for(int emp = 0; emp < cantidad; emp++)
{
    System.out.println("Nombre: " + nombres.get(emp) + " Salario: " + salarios.get(emp));
}

double total = 0;

for(int emp = 0; emp < cantidad; emp++) {
    total = total + salarios.get(emp);
}

System.out.println("El total de salarios pagados a los empleados es: " + total);
```

(Finalmente mostraremos quien fue el empleado que ganó más dinero, es decir el de mayor salario y también el que gano menos, es decir el de menor salario)

(Creamos una variable mayorSalario que nos permite cargar la posición del primer salario es decir el índice 0 del arreglo salarios y luego otra variable auxiliar llamada posición que nos permitirá dentro del for saber si dicho salario guardado es mayor a esta sucesivamente hasta encontrar el mayor de todo y guardarlo finalmente en el variable mayorSalario)

(Asi mismo para el de menor salario pero en este caso, evaluará el menor de estos)

```
System.out.println("El total de salarios pagados a los empleados es: " + total);

System.out.println(x:"El empleado que mas dinero gana es: " );
double mayorSalario = salarios.get(index:0);
int posicion = 0;
for(int emp = 1; emp < cantidad; emp++) {
    if(salarios.get(emp) > mayorSalario) {
        mayorSalario = salarios.get(emp);
        posicion = emp;
    }
}
System.out.println("Nombre: " + nombres.get(posicion) + " Cargo: " + cargos.get(posicion) + " Salario:" + salarios.get(posicion));

System.out.println(x:"El empleado que menos dinero gana es: " );
double menorSalario = salarios.get(index:0);
posicion = 0;
for(int emp = 1; emp < cantidad; emp++) {
    if(salarios.get(emp) < menorSalario) {
        menorSalario = salarios.get(emp);
        posicion = emp;
    }
}
System.out.println("Nombre: " + nombres.get(posicion) + " Cargo: " + cargos.get(posicion) + " Salario:" + salarios.get(posicion));

}
```

Programa n-3 : EstudianteEstructurada - **PROGRAMA TRABAJO EN CASA**

(Mismo programa mostrado en NotasEstructurada con la diferencia única que se utilizó la herramienta Scanner para asignar los valores a las variables)

```
import java.util.Scanner;
```

// Se declaro las variables a parte para usar dicha utilidad Scanner

// Scanner permite asignar los valores desde la propia terminal que nos brinda el IDE y no desde una interfaz, permitiendo realizar el trabajo de manera más eficaz

// El scanner tendrá su propias variables las cuales guardaran la asignación de las variables principales

```
// declaramos variables primero
String name;
Double noteParcial1;
Double noteParcial2;
Double noteFinal;

// Se utiliza la herramienta Scanner brindada por java para almacenar los datos

Scanner nombre = new Scanner(System.in);
Scanner nota1 = new Scanner(System.in);
Scanner nota2 = new Scanner(System.in);
```

// A diferencia del JOptionPane, aquí utilizamos println y luego abajo de este la variable principal y esto será igual al scanner nombre y el next que nos permite indicar qué tipo de datos se asigna y siempre debe ponerse para evitar errores o que el programa no funcione)

```
System.out.println("Ingrese el nombre del " + i + " Estudiante: ");
name = nombre.next();
System.out.println("Ingrese la nota 1 del parcial del " + i + " Estudiante: ");
noteParcial1 = nota1.nextDouble();
System.out.println("Ingrese la nota 2 del parcial del " + i + " Estudiante: ");
noteParcial2 = nota2.nextDouble();
```



```
for (int i = 1 ; i < 4 ; i++)
{
    // Solicitamos que ingrese por terminal los datos del estudiante y sus 2 notas
    System.out.println("Ingrese el nombre del " + i + " Estudiante: ");
    name = nombre.next();
    System.out.println("Ingrese la nota 1 del parcial del " + i + " Estudiante: ");
    noteParcial1 = nota1.nextDouble();
    System.out.println("Ingrese la nota 2 del parcial del " + i + " Estudiante: ");
    noteParcial2 = nota2.nextDouble();

    // Se hace una funcion para que calcule el promedio de las 2 notas
    noteFinal = (noteParcial1 + noteParcial2) / 2;

    // Se muestra por pantalla la informacion con sus 2 notas y el promedio
    System.out.println("Informacion del Estudiante: " + name + " Nota 1: " + noteParcial1 + " Nota 2: " + noteParcial2 + " Resultado del promedio de las 2 notas: " + noteFinal );

    // Finalizando la iteracion, se hace una condicion IF para saber si aprobo o no
    if (noteFinal < 3) {
        System.out.println("Usted: " + name + " Reprobo");
    }
    else {
        System.out.println("Usted: " + name + " Aprobo");
    }
}
```

Programa n-4 : EmpleadoEstructurada - PROGRAMA TRABAJO EN CASA

```
// Declaramos variables primero
String nombre, cargo;

double salario, total = 0, minSalario, maxSalario;

// Variable INT cantidad para asignar cuantas veces se hara la iteracion FOR
int cantidad, posicion = 0; // La posicion del Array

// Creamos 3 arrays (Listas para almacenar cada uno de los datos del empleado)

ArrayList<String> nombres = new ArrayList<String>(); // para los nombres
ArrayList<Double> salarios = new ArrayList<Double>(); // para los salarios
ArrayList<String> cargos = new ArrayList<String>(); // para los cargos

// Usamos util java Scanner para almacenar los datos que se le pidiran luego en el FOR al usuario
Scanner cantidades = new Scanner(System.in);
Scanner names = new Scanner(System.in);
Scanner salari= new Scanner(System.in);
Scanner carg = new Scanner(System.in);
```

```
// Usamos util java Scanner para almacenar los datos que se le pidiran luego en el FOR al usuario
Scanner cantidades = new Scanner(System.in);
Scanner names = new Scanner(System.in);
Scanner salari= new Scanner(System.in);
Scanner carg = new Scanner(System.in);

// Pedir que ingrese cantidad de empleados que se agregaran
System.out.println(x:"Ingresar cantidad de empleados");
cantidad = cantidades.nextInt();

for (int i =1 ; i <= cantidad ; i++) {

    // Solicitar los datos x terminal del empleado
    System.out.println(x:"Ingrese el nombre del empleado");
    nombre = names.next();
    System.out.println(x:"Ingrese el cargo del empleado");
    cargo = carg.next();
    System.out.println(x:"Ingrese el salario del empleado");
    salario = salari.nextDouble();

    // Agregar los datos que se solicitaron en el arreglo (Fila) correspondiente

    nombres.add(nombre);
    salarios.add(salario);
    cargos.add(cargo);

}
```

```

// Mostrar total de empleados
System.out.println("El total de empleados es: " + cantidad);

// Mostrar los nombres y salarios de los empleados
System.out.println(x:"Los nombres y salarios son: ");

// Se hace una iteracion for para ir mostrando cada empleado junto con su salario correspondiente
for (int i =0; i < cantidad ; i++) {
    System.out.println("Nombres: " + nombres.get(i) + " Salarios: " + salarios.get(i));
}

// Una iteracion FOR para ir sumando cada uno de los salarios asignados
for (int i =0; i < cantidad ; i++) {
    total = total + salarios.get(i);
}

// Mostrar por terminarl el total de salarios de los empleados

System.out.println("El total de salarios pagados es de: " +total);

// Asignar el empleado que mas dinero gana
System.out.println(x:"El empleado que mas dinero gana es: ");
// el max salario va ser igual a lo que contenga el arreglo salarios desde la primera posicion
// es decir la 1 (los arreglos empiezan desde 0 no desde 1)
maxSalario = salarios.get(index:0);

```

```

// Se realiza iteracion con if para ir comparando con es el mayor salario de todos los que hay
// en el arreglo, desde la primera posicion que fue desde donde es inicio originalmente
for (int i =1; i < cantidad ; i++) {
    // Comparacion con if todos los salarios y saber cual es el mayor y segun ello
    // asignar max salario el salario mayor que este en x posicion
    if(salarios.get(i) > maxSalario) {
        maxSalario = salarios.get(i);
        posicion = i;
    }
}

// Mostrar la persona con mayor salario en la posicion que esta (EJM: Juanpilaguna posicion = 2, cargo = x, salario = x)

System.out.println("Nombre: " + nombres.get(posicion) + " Cargo: " + cargos.get(posicion) + " Salario: " + salarios.get(posicion));

// Mismo proceso que el del arriba pero esta vez comparando cual es el menor y no el mayor

System.out.println(x:"El empleado que menos dinero gana es: ");
minSalario = salarios.get(index:0);

for (int i =1; i < cantidad ; i++) {
    if(salarios.get(i) < minSalario) {
        minSalario = salarios.get(i);
        posicion = i;
    }
}

System.out.println("Nombre: " + nombres.get(posicion) + " Cargo: " + cargos.get(posicion) + " Salario: " + salarios.get(posicion));

```

Informe Programas POO

Programa n-1 : NotasObjetos - **PROGRAMA REALIZADO EN CLASE**

(En este caso el primer paso fue crear una clase la cual tuvo de nombre Estudiante)

(Dicha clase tuvo los siguientes atributos: nombre, notaParcial1, notaParcial2, notaFinal)

```
public class Estudiante {  
    String nombre;  
    float notaParcial1, notaParcial2, notaFinal;  
}
```

(Se crea los respectivos constructores y se indica cuáles datos o atributos se van a utilizar específicamente con el this)

```
public Estudiante (String nombre) {  
    this.nombre = nombre;  
}  
  
public Estudiante (String nombre, float np1, float np2) {  
    this.nombre = nombre;  
    this.notaParcial1 = np1;  
    this.notaParcial2 = np2;  
}
```

(Creamos los respectivos GET y SET en este caso con la palabra en clave obtener y asigna para cada uno de los atributos que hemos definido desde el principio luego, creamos una función para calcular el promedio es decir la notaFinal del estudiante y dicha función se guardara en la variable notaFinal y del mismo modo se obtiene dicho resultado y finalmente se creó el obtenerMensaje que nos permite saber por medio de una condición quien aprobo o no)

```
public String obtenerNombre() {  
    return nombre;  
}  
  
public void asignarNotaParcial1(float np1) {  
    this.notaParcial1 = np1;  
}  
  
public void asignarNotaParcial2(float np2) {  
    this.notaParcial2 = np2;  
}  
  
public float obtenerNotaParcial1() {  
    return notaParcial1;  
}  
  
public float obtenerNotaParcial2() {  
    return notaParcial2;  
}  
  
public void calcularNotaFinal () {  
    notaFinal = (notaParcial1+notaParcial2)/2;  
}  
  
public float obtenerNotaFinal () {  
    return notaFinal;  
}  
  
public String obtenerMensaje(){  
    if (notaFinal <= 3) {  
        return "Reprobado";  
    }else {  
        return "Aprobado";  
    }  
}  
}
```

Programa n-2 : EmpresaObjetos - **PROGRAMA REALIZADO EN CLASE**

(Se creó principalmente la clase Empleado la cual contiene los siguientes atributos: nombre, cargo, salario)

```
public class Empleado {  
    private String nombre;  
    private String cargo;  
    private Double salario;
```

(Creación del respectivo constructor de empleado y sus SETTERS y GETTERS los cuales nos permite obtener y asignar dichos valores)

```
public Empleado(String nombre, String cargo, Double salario) {  
    this.nombre = nombre;  
    this.cargo = cargo;  
    this.salario = salario;  
}  
  
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
  
public String getNombre() {  
    return nombre;  
}  
  
public void setCargo(String cargo) {  
    this.cargo = cargo;  
}  
  
public String getCargo() {  
    return cargo;  
}  
  
public void setSalario(Double salario) {  
    this.salario = salario;  
}  
  
public Double getSalario() {  
    return salario;  
}
```

(Luego se crea la siguiente clase para la contratación de dicho empleado a la empresa, dicha clase no contiene ningún atributo y principalmente se creó un ArrayList de tipo empleados para guardar dicha información en un Array de tipo ArrayList)

```
public class Empresa {  
    ArrayList<Empleado> empleados;  
  
    public Empresa() {  
        empleados = new ArrayList<>();  
    }  
}
```

(Luego de declarar el ArrayList se crea un método que nos permite agregar a los empleados la cual se abrevia con la letra e de empleado y se agregaran en el array empleados)

```
// metodo que agrega empleados a la enterprise  
  
public void contratarEmpleados(Empleado e){  
    empleados.add(e);  
}
```

(Creamos los metodos getTotalEmpleados la cual nos permite conocer el tamaño total del array es decir la cantidad de lo empleados, nombreSalario para mostrar los nombres y salarios de cada empleado digitado y el getTotalSalarios que nos permite conocer por medio de una variable auxiliar llamada total que comienza en cero y se va sumando simultáneamente con cada uno de lo salarios y se guarda allí en total y se retornal)

```
//Metodo que obtiene el total de empleados

public int getTotalEmpleados() {
    return empleados.size();
}

public void nombreSalario() {
    for(Empleado e:empleados) {
        System.out.println("Nombre: " + e.getNombre() + " Salario: " + e.getSalario());
    }
}

//Metodo que muestra el nombre y el salario de todos los empleados
public double getTotalSalarios(){
    double total = 0;
    for(Empleado e:empleados) {
        total += e.getSalario();
    }
    return total;
}
```


(Misma función mostrada en los ejercicios de estructurada, básicamente se asigna al empMayorSalario la posición del empleado luego la variable auxiliar maxSalario guardará la posición y el salario de este, luego en la condición if evalúa si ese salario en dicha posición es mayor a maxSalario y así se recorre hasta que encuentre el mayor de todos y lo deje asignado en la variable empMayorSalario)

```
public void empleadoMayorSalario() {  
  
    Empleado empMayorSalario = empleados.get(index:0);  
  
    double maxSalario = empleados.get(index:0).getSalario();  
  
    for (Empleado e:empleados) {  
        if(e.getSalario() > maxSalario)  
        {  
            maxSalario = e.getSalario();  
            empMayorSalario = e;  
        }  
    }  
}
```

(Misma función que la anterior, solamente cambia las variables y que se evalúa si es menor)

```
public void empleadoMenorSalario() {  
    Empleado empMenorSalario = empleados.get(index:0);  
    double menSalario = empleados.get(index:0).getSalario();  
    for(Empleado e:empleados) {  
        if(e.getSalario() < menSalario ) {  
            menSalario = e.getSalario();  
            empMenorSalario = e;  
        }  
    }  
  
    System.out.println(x:"El empleado que menos plata gana es: ");  
    System.out.println("Nombre: " + empMenorSalario.getNombre());  
    System.out.println("Cargo: " + empMenorSalario.getCargo());  
    System.out.println("Salario: " + empMenorSalario.getSalario());  
}  
}
```

Programa n-3 : Biblioteca - **PROGRAMA REALIZADO EN CLASE (NO SE TERMINÓ, FUE UN EJEMPLO)**

(En este ejemplo se creó las clases Libro y Prestatario con los siguientes métodos y atributos)

```
public class Libro extends general{
    private int edicion;
    private int añoPublicacion;
    private Autor Autor;

    public Libro(){
    }
    public Libro(int edicion, int añoPublicacion, Autor autor) {
        this.edicion = edicion;
        this.añoPublicacion = añoPublicacion;
        this.Autor = autor;
    }
}
```

```
public class prestatario extends general {
    private String direccion;
    private String telefono;
    private String ruc;
    private Ciudad ciudad;

    public prestatario(){
    }
    public prestatario(int codigo, String nombre, String direccion, String telefono, String ruc, Ciudad ciudad) {
        this.direccion = direccion;
        this.telefono = telefono;
        this.ruc = ruc;
        this.ciudad = ciudad;
    }
}
```

```

}

    public void setDireccion(String direccion){
        this.direccion = direccion;
    }
    public String getDireccion(){
        return direccion;
    }

    public void setTelefono(String telefono){
        this.telefono = telefono;
    }
    public String getTelefono(){
        return telefono;
    }

    public void setRuc(String ruc){
        this.ruc = ruc;
    }
    public String getRuc(){
        return ruc;
    }
    public void setCiudad(Ciudad ciudad){
        this.ciudad = ciudad;
    }
    public Ciudad getCiudad(){
        return ciudad;
    }

    public String toString(){
        return "Prestatario (" + super.toString() + " Direccion: " + direccion + ", telefono: " + telefono + ", ruc: " + ruc + ", ciudad: " + ciudad + ')';
    }
}

```

Programa n-4 : EstudiantePOO - **PROGRAMA REALIZADO EN CASA**

(Mismo ejercicio realizado en clase solo con algunos diferencias, pero básicamente hace la misma función)

```
public class Estudiante {  
  
    // Creamos los atributos que posee la clase Estudiante  
    // o en otras palabras las variables la misma joda  
    public String nombre;  
    public double nota1;  
    public double nota2;  
    public double notaFinal;  
  
  
    // Asignamos los constructores es decir  
    // Aquellos que no sirven para identificar  
    // Los atributos que se usaran  
  
    public Estudiante (String nombre)  
    {  
        this.nombre = nombre;  
    }  
  
    public Estudiante (String nombre, double np1, double np2)  
    {  
        this.nombre = nombre;  
        nota1 = np1;  
        nota2 = np2;  
    }  
}
```

```

// No devolveran ningún valor obviamente
public void asignarNota1(double np1){
    nota1 = np1;
}

public void asignarNota2(double np2){
    nota2 = np2;
}

// Las funciones con su tipo si devolveran o retornaran
// Ejemplo la nota1 del estudiante etc..
public double obtenerNota1(){
    return nota1;
}

public double obtenerNota2(){
    return nota2;
}

public void calcularNotaFinal()
{
    notaFinal = (nota1 + nota2) / 2;
}

public double obtenerNotaFinal(){
    return notaFinal;
}

```

```

public String obtenerMensaje(){
    if(notaFinal < 3)
        return " ha reprobado lo lamento";
    else
        return " ha aprobado felicidades";
}

```

```

Estudiante est;

// Declaramos variables a utilizar, estas deben ser EXACTAMENTE iguales
// a las que se puso en la clase Estudiante
String nombre;
double nota1, nota2;

// Utilidad scanner me da pija explicar xdddd almacena datos y eso
Scanner name = new Scanner(System.in);
Scanner not1 = new Scanner(System.in);
Scanner not2 = new Scanner(System.in);

// Iteracion for que en este caso se repetira hasta llegar al 3

for (int i = 1; i < 4; i++) {
    // Solicitar los datos
    System.out.println("Ingrese el nombre del " + i + " Estudiante: ");
    nombre = name.next();
    System.out.println("Ingrese la nota 1 del parcial del " + i + " Estudiante: ");
    nota1 = not1.nextDouble();
    System.out.println("Ingrese la nota 2 del parcial del " + i + " Estudiante: ");
    nota2 = not2.nextDouble();

    // Dichos datos que se solicitaron al principio se iran guardando en los metodos
    // previamente hechos en la clase Estudiante

    // Usaremos el est que creamos es decir el objeto de Estudiante

    est = new Estudiante(nombre);
    est.asignarNota1(nota1);
    est.asignarNota2(nota2);
    est.calcularNotaFinal();

    // Mostrar por pantalla la informacion y notas y promedio del estudiante y la funcion IF
    // previamente creada en Estudiante EASY papa

    System.out.println("Querido estudiante: " + est.nombre + " Su nota 1 fue de: " + est.obtenerNota1() + " Su nota 2 fue de: " + est.obtenerNota2() + " El promedio de sus 2 notas parciales es
de: " + est.obtenerNotaFinal() + " Usted" + est.obtenerMensaje());
}

```

Programa n-5 : EmpleadosPOO - **PROGRAMA REALIZADO EN CASA**

(Mismo ejercicio que el de empresa, con algunos cambios mínimos. Pero proporciona la misma función y ejercicio)

```
public class Empresa {

    // EMPRESA NO TENDRA ATRIBUTOS ES DECIR, aquí lo que nos interesa
    // es contratar y almacenar dichos empleados en una array (fila) para ellos y crear funciones

    // Creamos el array (fila) para la clase empleado que se llamara empleados
    public ArrayList <Empleado> empleados; // empleados es el nombre de array no confundir

    //constructor que tomara los datos del array list empleados
    // o algo asi xd

    public Empresa()
    {
        empleados = new ArrayList<Empleado>();
    }

    //método que agrega empleados a la empresa
    // el metodo sera para clase Empleado y lo llamaremos como e para no hacerlo tan largo

    public void contratarEmpleado(Empleado e)
    {
        empleados.add(e); // Se agregara todos los datos del arreglo empleados es decir de la fila a la variable e
    }

    //método que obtiene el total de empleados
    public int getTotalEmpleados()
    {
        return empleados.size();
    }

    //método que muestra el nombre y salario de todos los empleados
    public void nombreSalario()
    {
        for(Empleado e:empleados) // por cada empleado que haya esto se repetira, esto es un foreach no un FOR
        {
            System.out.println("Nombre: "+ e.getNombre()+ " Salario: " +e.getSalario());
        }
    }
}
```

```

//método que retorna el total de dinero pagado a todos los empleados
public double getTotalSalarios()
{
    double total=0;
    for(Empleado e:empleados)

    {
        total=total+e.getSalario();
    }
    return total;
}

public void empleadoMayorSalario()
{
    Empleado empMayorSalario=empleados.get(0);
    double maxSalario=empleados.get(0).getSalario();
    for(Empleado e:empleados)
    {
        if(e.getSalario()>maxSalario)
        {
            maxSalario=e.getSalario();
            empMayorSalario=e;
        }
    }
    System.out.println("El empleado que mas dinero gana es: ");
    System.out.print("Nombre: " +empMayorSalario.getNombre());
    System.out.print(" Cargo: " +empMayorSalario.getCargo());
    System.out.println(" Salario: " +empMayorSalario.getSalario());
}

//método que muestra el nombre, cargo y salario del empleado que menos dinero

```



```
public void empleadoMenorSalario()  
{  
    Empleado empMenorSalario=empleados.get(0);  
    double menSalario=empleados.get(0).getSalario();  
  
    for(Empleado e:empleados)  
    {  
        if(e.getSalario()<menSalario)  
        {  
            menSalario=e.getSalario();  
            empMenorSalario=e;  
        }  
    }  
    System.out.println("El empleado que menos dinero gana es: ");  
    System.out.print("Nombre: "+empMenorSalario.getNombre());  
    System.out.print(" Cargo: "+empMenorSalario.getCargo());  
    System.out.println(" Salario: " +empMenorSalario.getSalario());  
}  
  
}
```