**PROGRAMA DE ESTUDIO DE ARQUITECTURA DE PLATAFORMAS Y SERVICIOS DE TECNOLOGÍAS DE LA** **INFORMACIÓN**

**FUNDAMENTOS DE PROGRAMACIÓN**

**ACTIVIDAD FINAL(PROBLEMA N°3)**

**INTEGRANTES**

Montalvan Pintado Edilsa

Nanfuñay Carrion Javier

Neyra Quesquen Renzo

**Chiclayo , Abril del 2023**

**Análisis del Problema:**

Realizar un programa que solicite ingresar el nombre del estudiante, programa de estudio (Adm, Ati, Con, Eli, Elo, Ete, Mau, Mep), así como su respectivo turno (M, N), las notas siguientes: examen 1 y exámen 2, y las notas de prácticas (en total 4).

El programa debe de realizar:

1. La validación de las notas (0 a 20)
2. Calcular el promedio de cada alumno (P= 70% de exámenes + 30% de prácticas)
3. La condición del estudiante (Aprobado si es >=12.5/Recuperación <12.5 y > a 10/Desaprobado <10)
4. Mostrar los datos del estudiante, promedio y condición
5. El total de estudiantes por programa de estudios
6. El % de aprobados por programa de estudios
7. Listar los estudiantes que ingresan a recuperación por programa de estudio

Antes de realizar el código primero se realiza un análisis del problema esto nos ayudará visualizar, entender y comprender el problema, por esa parte empezamos y eso es lo que hicimos en este problema que se nos planteó en esta actividad final.

Nosotros al leer lo siguiente : Realizar un programa que solicite ingresar el nombre del estudiante, programa de estudio (Adm, Ati, Con, Eli, Elo, Ete, Mau, Mep), así como su respectivo turno (M, N), las notas siguientes: examen 1 y exámen 2, y las notas de prácticas (en total 4).

Entendimos que el problema ya nos da una idea de cómo quiere que sea el programa (solicitar al usuario y pedirle nombre, programa de estudio turno y las notas respectivas en este caso son 6), aparte de eso el programa también debe realizar los siguientes requisitos funcionales y no funcionales… (Los funcionales son los que son visibles e interactúan con el usuario y los no funcionales son cosas que están en el código y eso no lo puede ver el usuario).

En este caso hemos separado los funcionales con los no funcionales:

Requisitos Funcionales:

* Mostrar los datos del estudiante, promedio y condición
* El total de estudiantes por programa de estudios
* El % de aprobados por programa de estudios
* Listar los estudiantes que ingresan a recuperación por programa de estudio

Requisitos no funcionales:

* La validación de las notas (0 a 20)
* Calcular el promedio de cada alumno (P= 70% de exámenes + 30% de prácticas)
* La condición del estudiante (Aprobado si es >=12.5/Recuperación <12.5 y > a 10/Desaprobado <10)

Pero si nos damos cuenta que en requisitos funcionales aunque no nos pidan, debemos registrar al estudiante y quedaría de esta manera:

* Registrar Estudiante
* Mostrar los datos del estudiante, promedio y condición
* El total de estudiantes por programa de estudios
* El % de aprobados por programa de estudios
* Listar los estudiantes que ingresan a recuperación por programa de estudio

Y mediante ese análisis ya tendríamos un posible menú:

Por otra parte los requisitos no funcionales son algo indispensable en este menú ya que sin ello no conociéramos el promedio y condición del estudiante, y de esa forma las otras funciones no tendrían sentido.

Así que nosotros hicimos un bosquejo de cómo sería el menu

INTRANET

1. Registrar Estudiante
2. Mostrar los datos del estudiante, promedio y condición
3. El total de estudiantes por programa de estudios
4. El % de aprobados por programa de estudios
5. Listar los estudiantes que ingresan a recuperación por programa de estudio
6. Salir

El menú quedaria asi de titulo le pusimos intranet y agregamos una opción de salida para que de esa manera el que esté interactuando con el programa pueda salir, lo demás queda igual.

Al ingresar a la primera opción “Registrar Estudiantes” se le mostrará lo siguiente:

* Ingrese nombre(Aquí se ingresa el nombre)
* Ingrese programa(Aquí el programa de estudios )
* Ingresar Turno(M/N)
* Ingrese notas (6)

(De esta manera se irán guardando los datos del estudiante que nosotros coloquemos)

Segunda opción “Mostrar estudiante, promedio y condición”:

* Se ingresa el (?)(Se buscará mediante el identificador al usuario y se mostrará)
* Se muestra el nombre
* Se muestra el programa
* Se muestra el promedio (P= 70% de exámenes + 30% de prácticas)
* Se muestra la condición (Aprobado si es >=12.5/Recuperación <12.5 y > a 10/Desaprobado <10)

(?) En este punto nos dimos cuenta que necesitamos un identificación en cada estudiante para que con ese identificador se muestra los datos que están pidiendo en la segunda opción, porque si colocamos que ingrese el nombre habrían muchos con ese nombre o el apellido igualmente pasaría lo mismo, asi que lo que hicimos es establecer como identificador el número de DNI del estudiante de esta manera pues como el DNI es único no habrá interrupciones, se pedirá el DNI del estudiante .

Teniendo esto en cuenta se puede continuar con lo demas opciones del menu:

Tercera opción ”Total de estudiantes por programa de estudio”

* Pedir al usuario que ingrese el programa de estudios
* Pedir al usuario que ingrese el turno
* Mostrar los estudiantes por programa de estudio en el turno elegido

Cuarta opción “El % de aprobados por programa de estudios”:

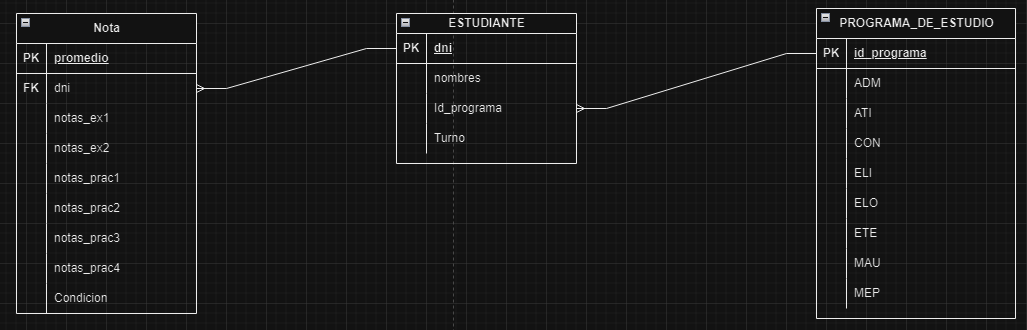
* Pedir al usuario que ingrese el programa de estudios
* Mostrar los estudiantes por programa de estudio que están aprobado

Quinta opción ”Listar los estudiantes que ingresan a recuperación por programa de estudio”

* Pedir al usuario que ingrese el programa de estudios
* Mostrar los estudiantes por programa de estudio que van a recuperación

En este punto ya sabíamos que cosas va hacer cada opción del menú. Ahora tuvimos que realizar el modelo lógico y físico para poder entender mejor al momento de codificar el código.

**MODELO LÓGICO**

****

**MODELO FÍSICO**

| estudiantes | | | |
| --- | --- | --- | --- |
| dni | nombre del estudiante | id\_prog | turno |
| 12345789 | Pedro Flores | 1 | M |
| 45645645 | Santos Santos | 2 | N |
| 15446452 | Perez Ulloa | 3 | M |
| 15478210 | Tenorio Sanchez | 4 | N |
| 45783650 | Delgado Mejia | 5 | M |

| Programa de estudios | |
| --- | --- |
| id\_prog | Programa |
| 1 | ADM |
| 2 | ATI |
| 3 | CON |
| 4 | ELI |
| 5 | ELO |
| 6 | ETE |
| 7 | MAU |
| 8 | MEP |

| notas\_ex\_pract\_prom | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| dni | examen1 | examen2 | not\_prac1 | not\_prac2 | not\_prac3 | not\_prac4 | Promedio | Condicion |
| 12345789 | 10 | 14 | 15 | 16 | 14 | 15 | 14 |  |
| 45645645 | 14 | 13 | 18 | 15 | 15 | 16 | 14 |  |
| 15446452 | 18 | 12 | 13 | 17 | 14 | 18 | 16 |  |
| 15478210 | 17 | 15 | 13 | 12 | 13 | 16 | 14 |  |
| 45783650 | 12 | 15 | 17 | 15 | 16 | 15 | 14 |  |

**CÓDIGO EN PYTHON:**

**estudiantes=[]**

**notas\_ex\_pract\_prom=[]**

**def registrar\_estudiantes():**

**sw = True**

**while sw:**

**print("\nRegistro de estudiantes")**

**print("======================\n")**

**dni = int(input("Ingrese DNI de estudiante: "))**

**# Verificar si el DNI ya existe en la lista de estudiantes**

**for estudiante in estudiantes:**

**if estudiante[0] == dni:**

**print("El DNI ingresado ya existe. Intente nuevamente.")**

**break**

**else:**

**nombre = input("Ingrese Nombres del estudiante: ")**

**while sw:**

**programa = int(input("""**

**1.- ADM**

**2.- ATI**

**3.- CON**

**4.- ELI**

**5.- ELO**

**6.- ETE**

**7.- MAU**

**8.- MEP**

**INGRESE PROGRAMA DE ESTUDIOS (1-8): """))**

**if programa <= 8 and programa >= 1:**

**programa\_nom = programa\_estu(programa)**

**while sw:**

**turno = input("Ingrese Turno: (M/N)")**

**turno = turno.upper()**

**if turno == "M" or turno == "N":**

**estudiante = [dni, nombre, programa\_nom, turno]**

**estudiantes.append(estudiante)**

**print("\nEstudiante registrado correctamente\n")**

**sw = False**

**break**

**else:**

**print("""**

**ERROR: OPCION INVALIDA**

**INTENTE NUEVAMENTE""")**

**else:**

**print("""**

**ERROR: OPCION INVALIDA**

**INTENTE NUEVAMENTE""")**

**p = input("Desea agregar otro estudiante (SI/NO)")**

**if p.lower() == "no":**

**sw = False; break**

**elif p.lower() == "si":**

**registrar\_estudiantes()**

**else:**

**print("Opcion Invalida")**

**def registrar\_notas():**

**dni = int(input("Ingrese DNI del estudiante: "))**

**# Verificar si el estudiante ya tiene notas registradas**

**for notas in notas\_ex\_pract\_prom:**

**if notas[0] == dni:**

**print("El estudiante ya tiene notas registradas. Ingrese otro DNI.")**

**break**

**else:**

**estudiante\_encontrado = None**

**for estudiante in estudiantes:**

**if estudiante[0] == dni:**

**estudiante\_encontrado = estudiante**

**break**

**if estudiante\_encontrado is not None:**

**ex1 = obtener\_nota("primer examen")**

**ex2 = obtener\_nota("segundo examen")**

**pr1 = obtener\_nota("primera práctica")**

**pr2 = obtener\_nota("segunda práctica")**

**pr3 = obtener\_nota("tercera práctica")**

**pr4 = obtener\_nota("cuarta práctica")**

**notas = [ex1, ex2, pr1, pr2, pr3, pr4]**

**promedio = promedio\_estu(notas)**

**condicion = condicion\_estu(promedio)**

**notas = [dni, ex1, ex2, pr1, pr2, pr3, pr4, promedio, condicion]**

**notas\_ex\_pract\_prom.append(notas)**

**print("\nNotas del Estudiante registrado correctamente\n")**

**p = input("Desea agregar otro estudiante (SI/NO)")**

**if p.lower() == "no":**

**print(" ")**

**elif p.lower() == "si":**

**registrar\_notas()**

**else:**

**print("Opcion Invalida")**

**else:**

**print("Estudiante no registrado\n")**

**def programa\_estu(programa):**

**if programa == 1:**

**return "ADM"**

**elif programa == 2:**

**return "ATI"**

**elif programa == 3:**

**return "CON"**

**elif programa == 4:**

**return "ELI"**

**elif programa == 5:**

**return "ELO"**

**elif programa == 6:**

**return "ETE"**

**elif programa == 7:**

**return "MAU"**

**elif programa == 8:**

**return "MEP"**

**def obtener\_nota(nombre\_evaluacion):**

**while True:**

**nota = int(input(f"Ingrese la nota de {nombre\_evaluacion}: "))**

**if nota >= 0 and nota <= 20:**

**return nota**

**else:**

**print("Error: La nota debe estar en el rango de 0 a 20. Intente nuevamente.")**

**def promedio\_estu(notas):**

**promedio = 0.7 \* ((notas[0] + notas[1]) / 2) + 0.3 \* ((notas[2] + notas[3] + notas[4] + notas[5]) / 4)**

**return promedio**

**def condicion\_estu(promedio):**

**if ( promedio >= 12.5):**

**return "Aprobado"**

**elif (promedio <= 12 and promedio >= 10):**

**return "Recuperacion"**

**else:**

**return "Desaprobado"**

**def ver\_datos():**

**dni = int(input("Ingrese DNI del estudiante: "))**

**estudiante\_encontrado = None**

**notas\_estudiante = None**

**for estudiante in estudiantes:**

**if estudiante[0] == dni:**

**estudiante\_encontrado = estudiante**

**break**

**for notas in notas\_ex\_pract\_prom:**

**if notas[0] == dni:**

**notas\_estudiante = notas**

**break**

**if estudiante\_encontrado is not None and notas\_estudiante is not None:**

**print("Datos del estudiante:")**

**print(f"DNI: {estudiante\_encontrado[0]}")**

**print(f"Nombres: {estudiante\_encontrado[1]}")**

**print(f"Programa de estudios: {estudiante\_encontrado[2]}")**

**print(f"Turno: {estudiante\_encontrado[3]}")**

**print(f"Promedio: {notas\_estudiante[7]:.2f}")**

**print(f"Condición: {notas\_estudiante[8]}")**

**else:**

**print("Estudiante no encontrado.")**

**def listar\_estudiantes():**

**sw = True**

**while sw:**

**programa = int(input("""**

**1.- ADM**

**2.- ATI**

**3.- CON**

**4.- ELI**

**5.- ELO**

**6.- ETE**

**7.- MAU**

**8.- MEP**

**INGRESE PROGRAMA DE ESTUDIOS (1-8): """))**

**if programa <= 8 and programa >= 1:**

**programa\_nombre = programa\_estu(programa)**

**turno = input("Ingrese Turno (M/N): ").upper()**

**pro = 0**

**for estudiante in estudiantes:**

**if estudiante[2] == programa\_nombre and estudiante[3] == turno:**

**pro += 1**

**print(f"Programa de estudios {programa\_nombre} - Turno {turno}: {pro}")**

**break**

**else:**

**print("Opción inválida")**

**listar\_estudiantes()**

**def ver\_aprobados():**

**sw=True**

**while (sw):**

**programa=int(input("""**

**1.- ADM**

**2.- ATI**

**3.- CON**

**4.- ELI**

**5.- ELO**

**6.- ETE**

**7.- MAU**

**8.- MEP**

**INGRESE PROGRAMA DE ESTUDIOS (1-8): """))**

**if programa <= 8 and programa >= 1:**

**programa\_nombre = programa\_estu(programa)**

**count\_total = 0**

**count\_aprobados = 0**

**for notas in notas\_ex\_pract\_prom:**

**dni = notas[0]**

**estudiante\_encontrado = None**

**for estudiante in estudiantes:**

**if estudiante[0] == dni:**

**estudiante\_encontrado = estudiante**

**break**

**if estudiante\_encontrado is not None and estudiante\_encontrado[2] == programa\_nombre:**

**count\_total += 1**

**if condicion\_estu(notas[7]) == "Aprobado":**

**count\_aprobados += 1**

**if count\_total > 0:**

**porcentaje\_aprobados = (count\_aprobados / count\_total) \* 100**

**print(f"Porcentaje de alumnos aprobados en el programa de estudios {programa\_nombre}: {porcentaje\_aprobados:.2f}%")**

**else:**

**print(f"No se encontraron estudiantes en el programa de estudios {programa\_nombre}")**

**break**

**else:**

**print("Opción inválida")**

**def listar\_recuperacion():**

**sw=True**

**while (sw):**

**programa=int(input("""**

**1.- ADM**

**2.- ATI**

**3.- CON**

**4.- ELI**

**5.- ELO**

**6.- ETE**

**7.- MAU**

**8.- MEP**

**INGRESE PROGRAMA DE ESTUDIOS (1-8): """))**

**if programa <= 8 and programa >= 1:**

**programa\_nombre = programa\_estu(programa)**

**estudiantes\_desaprobados = [] # Lista para almacenar estudiantes desaprobados**

**count\_total = 0**

**count\_desaprobados = 0**

**for notas in notas\_ex\_pract\_prom:**

**dni = notas[0]**

**estudiante\_encontrado = None**

**for estudiante in estudiantes:**

**if estudiante[0] == dni:**

**estudiante\_encontrado = estudiante**

**break**

**if estudiante\_encontrado is not None and estudiante\_encontrado[2] == programa\_nombre:**

**count\_total += 1**

**if condicion\_estu(notas[7]) == "Recuperacion":**

**count\_desaprobados += 1**

**estudiantes\_desaprobados.append(estudiante\_encontrado) # Agregar estudiante desaprobado a la lista**

**print(f"Estudiantes que entrar a recuperacion en el programa {programa\_nombre}:")**

**for estudiante in estudiantes\_desaprobados:**

**print(f"DNI: {estudiante[0]}")**

**print(f"Nombres: {estudiante[1]}")**

**print(f"Total de estudiantes que van a recuperacion: {count\_desaprobados} de {count\_total}\n")**

**sw=False; break**

**else:**

**print("Opción inválida")**

**sw=True**

**while (sw):**

**print("---INTRANET---")**

**print("===============\n")**

**print("1.- Registrar Estudiantes")**

**print("2.- Registrar Notas de Estudiantes")**

**print("3.- Ver datos del estudiante")**

**print("4.- Listar estudiantes por programa")**

**print("5.- Ver porcentaje de estudiantes aprobados por programa")**

**print("6.- Lista de estudiantes que van recuperacion por programa")**

**print("7.- Salir")**

**print("================\n")**

**n=int(input("\nIngresa una opcion (1-7): "))**

**if (n == 1):**

**registrar\_estudiantes()**

**elif (n == 2):**

**registrar\_notas()**

**elif (n == 3):**

**ver\_datos()**

**elif (n == 4):**

**listar\_estudiantes()**

**elif (n == 5):**

**ver\_aprobados()**

**elif (n == 6):**

**listar\_recuperacion()**

**elif (n == 7):**

**print("\n!Gracias!")**

**break**

**else:**

**print("Error: No es una opcion valida")**

**CÓDIGO EN JAVA:**

import java.util.Scanner;

public class Main {

public static int[][] *estudiantes* = new int[1][100];

public static String[][] *datosEstudiantes* = new String[3][100];

public static int[][] *notasExPractProm* = new int[9][100];

public static String[] *condicionesEstudiantes* = new String[100];

public static int *numEstudiantes* = 0;

public static int *numNotas* = 0;

public static void registrarEstudiantes() {

boolean sw = true;

Scanner sc = new Scanner(System.*in*);

while (sw) {

System.*out*.println("\nRegistro de estudiantes");

System.*out*.println("======================\n");

System.*out*.println("Ingrese DNI del estudiante: ");

int dni = sc.nextInt();

// Verificar si el DNI ya existe en la matriz de estudiantes

boolean dniExiste = false;

for (int i = 0; i < *numEstudiantes*; i++) {

if (*estudiantes*[0][i] == dni) {

dniExiste = true;

System.*out*.println("El DNI ingresado ya existe. Intente nuevamente.");

break;

}

}

if (!dniExiste) {

System.*out*.println("Ingrese Nombre del Estudiante");

String nombre = sc.next();

System.*out*.println("""

1.- ADM

2.- ATI

3.- CON

4.- ELI

5.- ELO

6.- ETE

7.- MAU

8.- MEP

INGRESE PROGRAMA DE ESTUDIOS (1-8): """);

int programa = sc.nextInt();

if (programa <= 8 && programa >=1) {

System.*out*.println("Ingrese Turno (M/N)");

String turno = sc.next();

if (turno.equals("m") || turno.equals("n") || turno.equals("M") || turno.equals("N")) {

System.*out*.println("\nEstudiante registrado correctamente\n");

*estudiantes*[0][*numEstudiantes*] = dni;

*datosEstudiantes*[0][*numEstudiantes*] = nombre;

*datosEstudiantes*[1][*numEstudiantes*] = *programaEstu*(programa);

*datosEstudiantes*[2][*numEstudiantes*] = turno.toUpperCase();

*numEstudiantes*++;

}

else{

System.*out*.println("""

ERROR: OPCION INVALIDA

INTENTE NUEVAMENTE""");

}

}

else {

System.*out*.println("""

ERROR: OPCION INVALIDA

INTENTE NUEVAMENTE""");

}

}

System.*out*.println("Desea agregar otro estudiante (SI/NO)");

String respuesta = sc.next();

if (respuesta.equalsIgnoreCase("no")) {

sw = false;

}

}

}

public static void registrarNotas() {

boolean sw = true;

Scanner sc = new Scanner(System.*in*);

while (sw) {

System.*out*.println("Ingrese DNI del estudiante: ");

int dni = sc.nextInt();

// Verificar si el estudiante ya tiene notas registradas

boolean notasRegistradas = false;

for (int i = 0; i < *numNotas*; i++) {

if (*notasExPractProm*[0][i] == dni) {

notasRegistradas = true;

System.*out*.println("El estudiante ya tiene notas registradas. Ingrese otro DNI.");

break;

}

}

if (!notasRegistradas) {

int ex1 = *obtener\_nota*("Primer Examen");

int ex2 = *obtener\_nota*("Segundo Examen");

int pr1 = *obtener\_nota*("Primera Práctica");

int pr2 = *obtener\_nota*("Segunda Práctica");

int pr3 = *obtener\_nota*("Tercera Practica");

int pr4 = *obtener\_nota*("Cuarta Practica");

int promedio = *promedioEstu*(ex1, ex2, pr1, pr2, pr3, pr4);

String condicion = *condicionEstu*(promedio);

*notasExPractProm*[0][*numNotas*] = dni;

*notasExPractProm*[1][*numNotas*] = ex1;

*notasExPractProm*[2][*numNotas*] = ex2;

*notasExPractProm*[3][*numNotas*] = pr1;

*notasExPractProm*[4][*numNotas*] = pr2;

*notasExPractProm*[5][*numNotas*] = pr3;

*notasExPractProm*[6][*numNotas*] = pr4;

*notasExPractProm*[7][*numNotas*] = promedio;

*condicionesEstudiantes*[*numNotas*] = condicion;

*numNotas*++;

System.*out*.println("\nNotas del Estudiante registrado correctamente\n");

}

else{

System.*out*.println("Estudiante no registrado\n");

*registrarNotas*();

}

System.*out*.println("Desea agregar otro estudiante (SI/NO)");

String respuesta = sc.next();

if (respuesta.equalsIgnoreCase("no")) {

sw = false;

}

}

}

public static String programaEstu(int programa) {

switch (programa) {

case 1:

return "ADM";

case 2:

return "ATI";

case 3:

return "CON";

case 4:

return "ELI";

case 5:

return "ELO";

case 6:

return "ETE";

case 7:

return "MAU";

case 8:

return "MEP";

default:

return "";

}

}

public static int obtener\_nota(String nombreEvaluacion) {

Scanner sc = new Scanner(System.*in*);

while (true) {

System.*out*.print("Ingrese la nota de " + nombreEvaluacion + ": ");

int nota = sc.nextInt();

if (nota >= 0 && nota <= 20) {

return nota;

} else {

System.*out*.println("Error: La nota debe estar en el rango de 0 a 20. Intente nuevamente.");

}

}

}

public static int promedioEstu(int ex1, int ex2, int pr1, int pr2, int pr3, int pr4) {

int promedio = (int) (0.7 \* ((ex1 + ex2) / 2) + 0.3 \* ((pr1 + pr2 + pr3 + pr4) / 4));

return promedio;

}

public static String condicionEstu(int promedio) {

if (promedio >= 12.5) {

return "Aprobado";

} else if (promedio < 12.5 && promedio > 10) {

return "Recuperacion";

} else {

return "Desaprobado";

}

}

public static void verDatos() {

Scanner sc = new Scanner(System.*in*);

System.*out*.println("Ingrese DNI del estudiante: ");

int dni = sc.nextInt();

String estudianteEncontrado = null;

String notasEstudiante = null;

for (int i = 0; i < *numEstudiantes*; i++) {

if (*estudiantes*[0][i] == dni) {

estudianteEncontrado = *datosEstudiantes*[0][i];

for (int j = 0; j < *numNotas*; j++) {

if (*notasExPractProm*[0][j] == dni) {

notasEstudiante = *condicionesEstudiantes*[j];

if (estudianteEncontrado != null && notasEstudiante != null) {

System.*out*.println("Datos del estudiante:");

System.*out*.println("DNI: " + dni);

System.*out*.println("Nombres: " + estudianteEncontrado);

System.*out*.println("Programa de estudios: " + *datosEstudiantes*[1][i]);

System.*out*.println("Turno: " + *datosEstudiantes*[2][i]);

}

}

}

for (int k = 0; k < *numNotas*; k++) {

if (*notasExPractProm*[0][k] == dni) {

System.*out*.println("Promedio: " + *notasExPractProm*[7][k]);

System.*out*.println("Condición: " + notasEstudiante);

System.*out*.println(" ");

break;

}

}

} else {

System.*out*.println("Estudiante no encontrado.");

}

}

}

public static void listar\_Estudiantes() {

Scanner sc = new Scanner(System.*in*);

System.*out*.println("Ingrese el número del programa de estudios (1-8): ");

int programa = sc.nextInt();

String programaNombre = *programaEstu*(programa);

System.*out*.println("Ingrese el turno (M/N): ");

String turno = sc.next().toUpperCase();

int cantidadEstudiantes = 0;

for (int i = 0; i < *numEstudiantes*; i++) {

String programaEstudio = *datosEstudiantes*[1][i];

String turnoEstudiante = *datosEstudiantes*[2][i];

if (programaEstudio.equals(programaNombre) && turnoEstudiante.equals(turno)) {

cantidadEstudiantes++;

}

}

System.*out*.println("Cantidad de estudiantes en el programa de estudios " + programaNombre +

" - Turno " + turno + ": " + cantidadEstudiantes);

}

public static void ver\_aprobados() {

Scanner sc = new Scanner(System.*in*);

System.*out*.println("""

1.- ADM

2.- ATI

3.- CON

4.- ELI

5.- ELO

6.- ETE

7.- MAU

8.- MEP

INGRESE PROGRAMA DE ESTUDIOS (1-8): """);

int programa = sc.nextInt();

if (programa <= 8 && programa >= 1) {

String programa\_nom = *programaEstu*(programa);

int totalEstudiantes = 0;

int aprobados = 0;

for (int i = 0; i < *numNotas*; i++) {

if (*datosEstudiantes*[1][i] == programa\_nom) {

totalEstudiantes++;

if (*condicionesEstudiantes*[i].equals("Aprobado")) {

aprobados++;

}

}

}

if (totalEstudiantes > 0) {

double porcentajeAprobados = (double) aprobados / totalEstudiantes \* 100;

System.*out*.println("Porcentaje de estudiantes aprobados en el programa de estudios " + *programaEstu*(programa) + ": " + porcentajeAprobados + "%");

} else {

System.*out*.println("No se encontraron estudiantes en el programa de estudios especificado.");

}

}

}

public static void lista\_recuperacion() {

Scanner sc = new Scanner(System.*in*);

System.*out*.println("Ingrese programa de estudios (1-8): ");

int programa = sc.nextInt();

String programa\_nom = *programaEstu*(programa);

System.*out*.println("Estudiantes en recuperación en el programa " + *programaEstu*(programa) + ":");

boolean estudiantesEnRecuperacion = false;

for (int i = 0; i < *numNotas*; i++) {

if (*datosEstudiantes*[1][i] == programa\_nom && *condicionesEstudiantes*[i].equals("Recuperacion")) {

System.*out*.println("DNI: " + *estudiantes*[0][i]);

System.*out*.println("Nombres: " + *datosEstudiantes*[0][i]);

estudiantesEnRecuperacion = true;

}

}

if (!estudiantesEnRecuperacion) {

System.*out*.println("No se encontraron estudiantes en recuperación en el programa de estudios especificado.");

}

}

//Menu del Programa ("INTRANET")

public static void main(String[] args) {

Scanner sc = new Scanner(System.*in*);

boolean sw = true;

while (sw) {

System.*out*.println("\n---INTRANET---");

System.*out*.println("===============\n");

System.*out*.println("1.- Registrar Estudiantes");

System.*out*.println("2.- Registrar Notas de Estudiantes");

System.*out*.println("3.- Ver datos del estudiante");

System.*out*.println("4.- Listar estudiantes por programa");

System.*out*.println("5.- Ver porcentaje de estudiantes aprobados por programa");

System.*out*.println("6.- Lista de estudiantes que van a recuperación por programa");

System.*out*.println("7.- Salir");

System.*out*.println("================\n");

System.*out*.print("\nIngresa una opcion (1-7): ");

int n = sc.nextInt();

switch (n) {

case 1:

*registrarEstudiantes*();

break;

case 2:

*registrarNotas*();

break;

case 3:

*verDatos*();

break;

case 4:

*listar\_Estudiantes*();

break;

case 5:

*ver\_aprobados*();

break;

case 6:

*lista\_recuperacion*();

break;

case 7:

sw = false;

System.*out*.println("\n¡Gracias!");

break;

default:

System.*out*.println("Error: Opción inválida");

break;

}

}

}

}

**ANÁLISIS GENERAL DEL CÓDIGO:**

1. El programa comienza definiendo dos listas vacías: "estudiantes" y "notas\_ex\_pract\_prom", que se utilizarán para almacenar los datos de los estudiantes y sus notas respectivamente.
2. La función "registrar\_estudiantes()" permite registrar nuevos estudiantes en la lista "estudiantes". Solicita al usuario ingresar el DNI, verifica si ya existe en la lista, luego pide el nombre, el programa de estudios y el turno del estudiante. Si todos los datos son válidos, se agrega el estudiante a la lista y se muestra un mensaje de confirmación. El proceso se repite si el usuario desea agregar más estudiantes.
3. La función "registrar\_notas()" permite registrar las notas de un estudiante. Solicita el DNI del estudiante y verifica si ya se han registrado sus notas. Si no se han registrado, se busca al estudiante en la lista "estudiantes" y se solicitan las notas del examen 1, examen 2 y las 4 notas de prácticas. Luego se calcula el promedio del estudiante utilizando la fórmula proporcionada y se determina su condición. Finalmente, se agrega el registro de notas a la lista "notas\_ex\_pract\_prom".
4. Las funciones "programa\_estu()" y "obtener\_nota()" son funciones auxiliares utilizadas para convertir el número de programa de estudios a su correspondiente nombre y para obtener una nota válida, respectivamente.
5. La función "promedio\_estu()" calcula el promedio del estudiante utilizando la fórmula proporcionada.
6. La función "condicion\_estu()" determina la condición del estudiante (Aprobado, Recuperación o Desaprobado) según su promedio.
7. La función "ver\_datos()" permite ver los datos de un estudiante dado su DNI. Busca al estudiante en la lista "estudiantes" y busca sus notas correspondientes en la lista "notas\_ex\_pract\_prom". Si se encuentran ambas coincidencias, muestra los datos del estudiante, su promedio y su condición.
8. La función "listar\_estudiantes()" permite listar el total de estudiantes en un programa de estudios y turno específico. Solicita al usuario ingresar el programa de estudios y turno, luego realiza el conteo de estudiantes correspondientes y muestra el resultado.
9. La función "ver\_aprobados()" muestra el porcentaje de estudiantes aprobados en un programa de estudios específico. Solicita al usuario ingresar el programa de estudios y realiza el cálculo contando el número de estudiantes aprobados y el número total de estudiantes en ese programa. Muestra el resultado en porcentaje.
10. La función "listar\_recuperacion()" muestra la lista de estudiantes que necesitan recuperación en un programa de estudios específico. Solicita al usuario ingresar el programa de estudios y muestra los estudiantes desaprobados que necesitan recuperación.
11. El programa principal se encuentra dentro de un bucle while. Muestra un menú de opciones al usuario y permite ejecutar las funciones correspondientes según la opción elegida. El bucle se repite hasta que el usuario elija la opción "Salir".