

Universidad de Panamá
Facultad de Informática, Electrónica y Comunicación
Centro Regional Universitario de Veraguas

Proyecto No. 2
ENCAPSULACIÓN

Autor
Faustino Aguilar Quirós
2-732-727

Facilitador
Prof. Abundio Mendoza

II° Semestre de 2015

DESARROLLO DE LA ESTRATEGIA

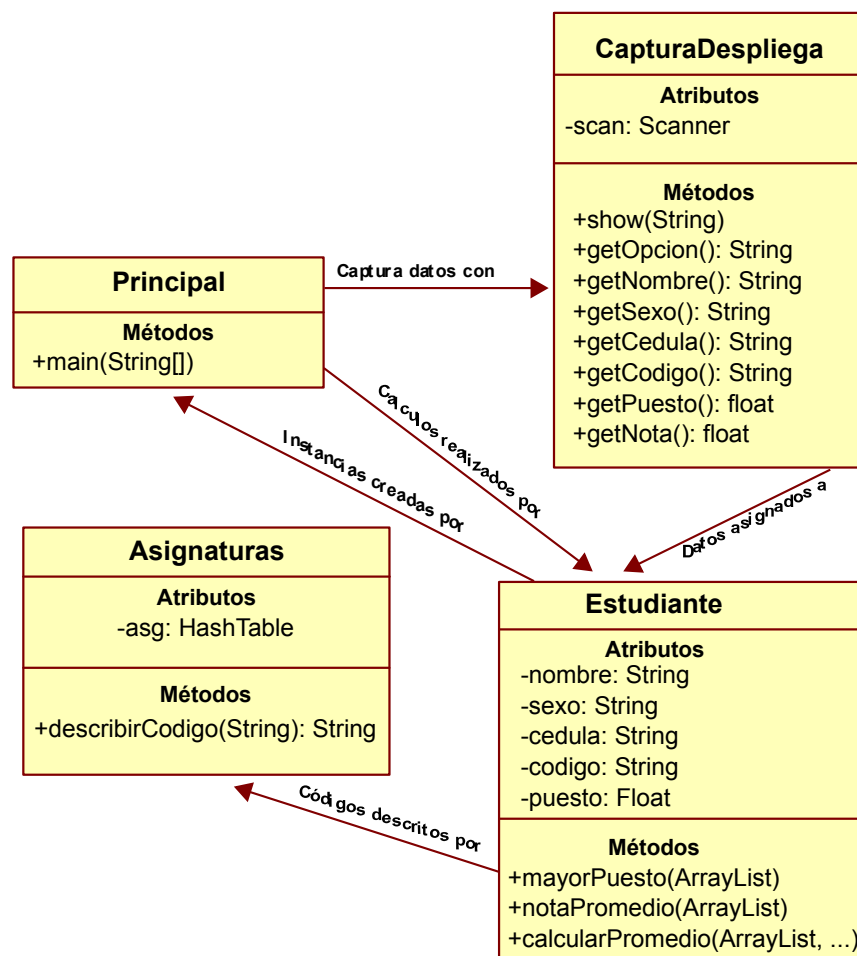
Se desarrolla un programa en Java estrictamente orientado a objetos que captura el nombre, sexo, cédula, el puesto académico que ocupa y el código de la asignatura de mayor calificación para cada uno de los estudiantes de un grupo. El programa debe determinar el nombre y género del estudiante con mayor puesto y desplegar la descripción de la asignatura con mayor calificación. Luego debe obtener las notas de cada estudiante y se presentan sus promedios.

En la clase *Principal* de nuestro programa crearemos objetos de las demás clases. La clase *CapturaDespliega* es utilizada para capturar datos que se asignan a variables en la clase *Principal*. Se crea un *ArrayList* donde cada elemento es un objeto clase *Estudiante* inicializados con las variables antes asignadas.

Cuando ya se tienen los datos de los estudiantes en el arreglo se utiliza el método *mayorPuesto* de la clase *Estudiante* que crea un objeto de la clase *Asignatura* para obtener la descripción del código de asignatura y calcula el estudiante de mayor puesto y muestra sus datos. Entonces se utiliza el método *notaPromedio* de la clase *Estudiante* para obtener las notas de cada estudiante y luego se llama al método *calcularPromedio* que pregunta por una cédula de un estudiante y calcula su promedio correspondiente, según las notas capturadas por *notaPromedio*.

DIAGRAMA UML

A continuación el diagrama UML del problema:



CÓDIGO FUENTE

Estudiante.java

```
import java.util.*;

class Estudiante {

    // Contiene los atributos del estudiante
    private String nombre, sexo, cedula, codigo;
    private float puesto;
    CapturaDespliega input = new CapturaDespliega();

    Estudiante(){}
    Estudiante(String nombre, String sexo, String cedula, String codigo, float
puesto) {
        // EL constructor asigna datos al estudiante
        this.nombre = nombre;
        this.sexo = sexo;
        this.cedula = cedula;
        this.codigo = codigo;
        this.puesto = puesto;
    }

    public void mayorPuesto(ArrayList<Estudiante> alumnos) {
        Estudiante maximo; // Estudiante de mayor puesto
        Asignaturas asg = new Asignaturas();
        String materia;
        maximo = alumnos.get(0);
        // Se calcula el máximo puesto
        for(Estudiante e: alumnos) {
            if(e.puesto > maximo.puesto){
                maximo = e;
            }
        }

        input.show("\nMayor Puesto\n=====\n");
        input.show("Nombre: " + maximo.nombre + "\n");
        input.show("Género: " + maximo.sexo + "\n");
        input.show("Código de asignatura: " + maximo.codigo + "\n");
        // Si el código de asg. no corresponde
        // a ninguno en la clase Asignaturas
        // se muestra un mensaje de error.
        materia = asg.describirCodigo(maximo.codigo);
        if (materia != null) {
            input.show("Descripción de la Asignatura: " + materia + "\n");
        } else {
            input.show("Error: Código de asignatura no relacionado\n");
        }
    }
}
```

```

public void notaPromedio(ArrayList<Estudiante> alumnos){
    // Calcula el promedio de las notas por cada
    // estudiante.
    int i;
    String opcion = "no";
    float suma;
    ArrayList<String> cedulas = new ArrayList<String>();
    ArrayList<Float> sumas = new ArrayList<Float>();
    ArrayList<Integer> notas = new ArrayList<Integer>();
    input.show("\n\tNotas de los estudiantes\n");
    for(Estudiante e: alumnos) {
        i=0;
        suma=0;
        input.show("\nEstudiante: " + e.nombre + "\n");
        do {
            suma += input.getNota(i);
            opcion = input.getOpcion();
            i++;
        } while (opcion.equalsIgnoreCase("si"));
        cedulas.add(e.cedula);
        sumas.add(suma);
        notas.add(i);
    }
    calcularPromedio(cedulas, sumas, notas);
}

public void calcularPromedio(ArrayList<String> cedulas,
                             ArrayList<Float> sumas,
                             ArrayList<Integer> notas) {
    String cedula, opcion;
    float promedio;
    int indice;
    input.show("\n\tPromedio de los estudiantes\n");
    do {
        cedula = input.getCedula();
        indice = cedulas.indexOf(cedula);
        if (indice >= 0) {
            promedio = sumas.get(indice)/notas.get(indice);
            input.show("\nPromedio de " + cedula + ": " + promedio + "\n");
        } else {
            input.show("Error: Cédula no encontrada\n");
        }
        opcion = input.getOpcion();
    } while (opcion.equalsIgnoreCase("si"));
}
}

```

Asignatura.java

```
// Contiene clase Hashtable
import java.util.Hashtable;

public class Asignaturas {
    // Clase que relaciona las asignaturas y sus codigos
    private Hashtable<String, String> asg = new Hashtable<String, String>();

    Asignaturas() {
        // Asignaturas que cubre el programa
        this.asg.put("MAT101", "Matemáticas");
        this.asg.put("ESP100", "Español");
        this.asg.put("ING121", "Inglés");
        this.asg.put("FIS112", "Física");
        this.asg.put("BI0122", "Biología");
        this.asg.put("QUIM210", "Química");
    }

    public String describirCodigo(String code){
        /*
        Método para obtener la descripción del
        código de la asignatura.
        No es necesario un método para agregar
        asignaturas ya que no es el objetivo del
        problema:
        */
        return this.asg.get(code);
    }
}
```

CapturaDespliega.java

```
// util contiene Scanner
import java.util.Scanner;

class CapturaDespliega {

    private Scanner scan = new Scanner(System.in);

    public void show(String data) {
        System.out.print(data);
    }

    public String getOpcion(){
        // Captura la opción para continuar capturando datos
        System.out.print("¿Desea introducir más datos? si/no: ");
        return this.scan.nextLine();
    }

    public String getNombre(){
        // Captura el Nombre
        System.out.print("Escriba el nombre: ");
        return this.scan.nextLine();
    }
}
```

```

public String getSexo(){
    // Captura el sexo
    System.out.print("Escriba el sexo: ");
    return this.scan.nextLine();
}

public String getCedula(){
    // Captura la cedula
    System.out.print("Escriba una cedula: ");
    return this.scan.nextLine();
}

public String getCodigo(){
    // Captura el código de la asignatura
    System.out.print("Código de asignatura de mayor calificación: ");
    return this.scan.nextLine();
}

public float getPuesto(){
    /*
    Captura el puntaje del puesto del estudiante
    Como es un numero flotante, no se captura
    el salto de linea y se tiene que utilizar
    nextLine para recogerlo.
    Si el puesto es negativo se responde con cero
    ya que es inválido, lo mismo pasa en getNota
    */
    float puesto;
    System.out.print("Número de puesto académico: ");
    puesto = this.scan.nextFloat();
    this.scan.nextLine();
    if (puesto >= 0) {
        return puesto;
    } else {
        System.out.println("Puesto minimo es 0");
        return 0;
    }
}

public float getNota(int i){
    // Captura la nota de un estudiante
    float nota;
    i++;
    System.out.print("Nota " + i + ": ");
    nota = this.scan.nextFloat();
    this.scan.nextLine();
    if (nota >= 0) {
        return nota;
    } else {
        System.out.println("nota minima es 0");
        return 0;
    }
}
}

```

Principal.java

```
/*
En el compilador javac utilizado
la librería io es incluida por defecto
util Contiene ArrayList */
import java.util.ArrayList;

class Principal {
    // Clase Principal que invoca a las demás clases

    public static void main(String[] args) {
        // Variables a utilizar
        int i=0;
        float puesto;
        String nombre, sexo, cedula, codigo, opcion;
        ArrayList<Estudiante> alumnos = new ArrayList<Estudiante>();
        CapturaDespliega input = new CapturaDespliega();
        Estudiante E = new Estudiante();
        /* Se presentan las asignaturas y se comienzan a capturar datos de los
        estudiantes hasta que el usuario detenga la captura respondiendo "no" */
        input.show("\tPlataforma de organización Estudiantil\n");
        input.show("\nAsignaturas\n");
        input.show("MAT101: Matemáticas\n");
        input.show("ESP100: Español\n");
        input.show("ING121: Inglés\n");
        input.show("FIS112: Física\n");
        input.show("BI0122: Biología\n");
        input.show("QUI210: Química\n");
        try {
            do {
                i++;
                input.show("\nEstudiante " + i + "\n");
                // Lee los datos del estudiante
                nombre = input.getNombre();
                sexo = input.getSexo();
                cedula = input.getCedula();
                codigo = input.getCodigo();
                puesto = input.getPuesto();
                Estudiante est = new Estudiante(nombre, sexo, cedula, codigo, puesto);
                alumnos.add(est);
                opcion = input.getOpcion();
            } while (opcion.equalsIgnoreCase("si"));
            // Se utilizan los métodos de la clase Estudiante para calcular
            // el mayorPuesto y el promedio de cada estudiante.
            E.mayorPuesto(alumnos);
            E.notaPromedio(alumnos);
            input.show(";Gracias por utilizar la Plataforma!" + "\n");
        } catch (Exception e) {
            input.show("Ocurrió un error: " + e + "\n");
        }
    }
}
```

EVIDENCIA

Plataforma de organización Estudiantil

Asignaturas

MAT101: Matemáticas
ESP100: Español
ING121: Inglés
FIS112: Física
BIO122: Biología
QUI210: Química

Estudiante 1

Escriba el nombre: Faustino Aguilar
Escriba el sexo: Masculino
Escriba la cedula: 2-732-727
Código de asignatura de mayor calificación: MAT101
Número de puesto académico: 2.1
¿Desea introducir más datos? si/no: si

Estudiante 2

Escriba el nombre: Carlos Álvarez
Escriba el sexo: Masculino
Escriba la cedula: 9-122-1251
Código de asignatura de mayor calificación: FIS112
Número de puesto académico: 2.9
¿Desea introducir más datos? si/no: si

Estudiante 3

Escriba el nombre: María Jiménez
Escriba el sexo: Femenino
Escriba la cedula: 9-123-2143
Código de asignatura de mayor calificación: ING121
Número de puesto académico: 1.7
¿Desea introducir más datos? si/no: no

Mayor Puesto

=====
Nombre: Carlos Álvarez
Género: Masculino
Código de asignatura: FIS112
Descripción de la Asignatura: Física

Descripción de la Captura de Pantalla:

En la captura de pantalla podemos ver que muestra las asignaturas con sus respectivos códigos para luego pedir los datos del estudiante 1 y así sucesivamente hasta que el usuario no desee introducir más datos, calculando el mayor puesto y mostrando los datos de este, así como la descripción de la asignatura en que obtuvo la mayor calificación.

La siguiente sección del programa nos pide calificaciones de cada estudiante hasta que el usuario decida, para luego calcular su promedio y mostrarlo en pantalla.

EVIDENCIA

Notas de los estudiantes

Estudiante: Faustino Aguilar
Nota 1: 90
¿Desea introducir más datos? si/no: si
Nota 2: 77
¿Desea introducir más datos? si/no: no

Estudiante: Carlos Álvarez
Nota 1: 98
¿Desea introducir más datos? si/no: si
Nota 2: 82
¿Desea introducir más datos? si/no: si
Nota 3: 100
¿Desea introducir más datos? si/no: no

Estudiante: María Jiménez
Nota 1: 64
¿Desea introducir más datos? si/no: no

Promedio de los estudiantes

Escriba una cedula: 2-732-727

Promedio de 2-732-727: 83.5
¿Desea introducir más datos? si/no: si
Escriba una cedula: 9-123-4567

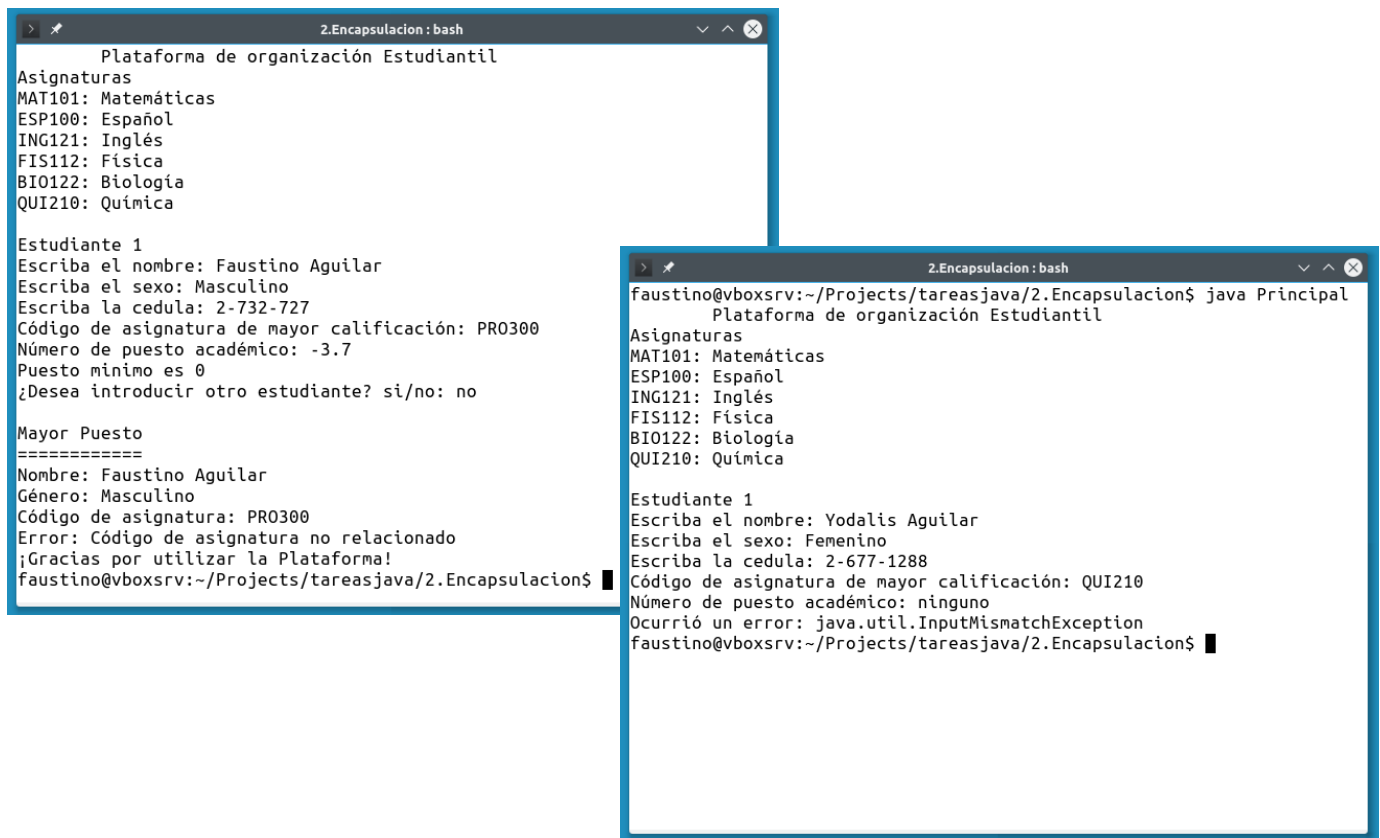
Promedio de 9-123-4567: 93.333336
¿Desea introducir más datos? si/no: si
Escriba una cedula: 9-987-6543

Promedio de 9-987-6543: 64.0
¿Desea introducir más datos? si/no: si
Escriba una cedula: 1-231-1232
Error: Cédula no encontrada
¿Desea introducir más datos? si/no: no
¡Gracias por utilizar la Plataforma!
faustino@vboxsrv:~/Projects/tareasjava/2.Encapsulacion\$

En esta captura se muestra como se capturan las calificaciones, luego se pide una cédula y se calcula el promedio para el estudiante, luego se muestra el resultado en pantalla y se pregunta si se desea continuar.

Finalmente mostrando un mensaje de agradecimiento.

EVIDENCIA



```
2.Encapsulacion : bash
Plataforma de organización Estudiantil
Asignaturas
MAT101: Matemáticas
ESP100: Español
ING121: Inglés
FIS112: Física
BIO122: Biología
QUI210: Química

Estudiante 1
Escriba el nombre: Faustino Aguilar
Escriba el sexo: Masculino
Escriba la cedula: 2-732-727
Código de asignatura de mayor calificación: PRO300
Número de puesto académico: -3.7
Puesto mínimo es 0
¿Desea introducir otro estudiante? si/no: no

Mayor Puesto
=====
Nombre: Faustino Aguilar
Género: Masculino
Código de asignatura: PRO300
Error: Código de asignatura no relacionado
¡Gracias por utilizar la Plataforma!
faustino@vboxsrv:~/Projects/tareasjava/2.Encapsulacion$

2.Encapsulacion : bash
faustino@vboxsrv:~/Projects/tareasjava/2.Encapsulacion$ java Principal
Plataforma de organización Estudiantil
Asignaturas
MAT101: Matemáticas
ESP100: Español
ING121: Inglés
FIS112: Física
BIO122: Biología
QUI210: Química

Estudiante 1
Escriba el nombre: Yodalis Aguilar
Escriba el sexo: Femenino
Escriba la cedula: 2-677-1288
Código de asignatura de mayor calificación: QUI210
Número de puesto académico: ninguno
Ocurrió un error: java.util.InputMismatchException
faustino@vboxsrv:~/Projects/tareasjava/2.Encapsulacion$
```

Algunos puntos extra a tomar en cuenta son cuando el usuario introduce un número negativo en el puesto, dando se por entendido que el mínimo puesto es cero. También si no introduce números; osea, caracteres alfanuméricos en el puesto, se causa una excepción y muestra un mensaje de error.

COMENTARIOS

La encapsulación en la programación orientada a objetos es un recurso importante a la hora de realizar proyectos, dándole la característica de que el código puede crecer y ser actualizado sin mayores problemas, ya que todo está dividido en partes y cada uno tiene un propósito.

Algo Parecido a esto y que ya habíamos visto antes en Programación II de Estructura de Datos es el uso de funciones en C y librerías que permitían dividir el trabajo. Ahora con Java muchas cosas son parecidas solo que las clases son un concepto mayor que puede contener en su interior atributos y métodos que se encapsulan para ser utilizados dependiendo de sus permisos.