Actividad Experimental N.° 1

Andrés Villacis, Cristina Guamán, Pablo Jiménez, William Villasis, Robert Celi

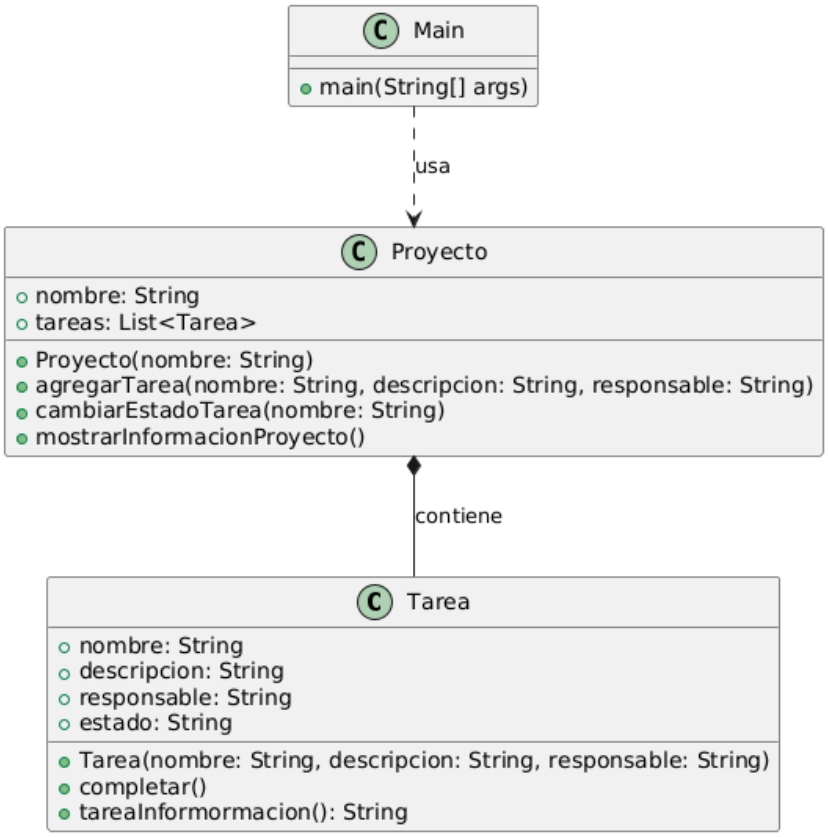
Universidad de las Fuerzas Armadas ESPE

NRC 1323: Programación Orientada a Objetos

08 de diciembre de 2024

Control de Lectura 2

Primero establecemos un diagrama UML, donde vamos a crear los objetos necesarios, además de cada uno de sus atributos, al igual que cada uno de los métodos a usar, dándonos como resultado el siguiente gráfico:



A partir del UML declaramos cada clase y también declaramos cada uno de los atributos al igual cada una de las acciones o métodos a realizar cada uno de ellos.

```java
package Proyecto;

class Tarea {  5 usages
    public String nombre;  3 usages
    public String descripcion;  2 usages
    public String responsable;  2 usages
    public String estado;  3 usages

    public Tarea(String nombre,String descripcion, String responsable) {  1 usage
        this.nombre = nombre;
        this.descripcion = descripcion;
        this.responsable = responsable;
        this.estado = "Pendiente";
    }

    public String obtenerNombre() {  1 usage
        return nombre;
    }

    public void completar() {  1 usage
        this.estado = "Completada";
    }

    public void tareaInformormacion() {  1 usage
        System.out.println("Tarea: "+ nombre + ", Descripcion: " + descripcion + ", Responsable: " + responsable
                + ", Estado: " + estado);
    }
}
```

```java
package Proyecto;

import java.util.ArrayList;
import java.util.List;

class Proyecto {  7 usages
    public String nombre;  3 usages
    public List<Tarea> tareas;  4 usages

    public Proyecto(String nombre) {  1 usage
        this.nombre = nombre;
        this.tareas = new ArrayList<>();
    }

    public String obtenerNombreProyecto() {  2 usages
        return nombre;
    }

    public void agregarTarea(String nombre, String descripcion, String responsable) {  1 usage
        Tarea nuevaTarea = new Tarea(nombre,descripcion, responsable);
        tareas.add(nuevaTarea);
    }

    public void cambiarEstadoTarea(String nombre) {  1 usage
        for (Tarea tarea : tareas) {
            if (tarea.obtenerNombre().equalsIgnoreCase(nombre)) {
                tarea.completar();
                System.out.println("Tarea marcada como completada.");
                return;
            }
        }
        System.out.println("Tarea no encontrada.");
```

En esta imagen se puede observar como esta estructurado el código de los objetos los cuales tienen sus atributos además de sus propios métodos. A continuación, vamos a explicar cada uno de ellos comenzando con Tarea

Primero tenemos “package Proyecto”, esto indica que está dentro de una carpeta llmada proyecto. Luego instanciamos la clase “Tarea”, de la siguiente forma “class Tarea”, ahora le damos sus atributos mediante “public (Tipo de varibale) <Variable>”, donde tenemos un modificador de acceso público, también definimos el tipo de variable y le damos un nombre.

Para el primer caso tenemos: “public String nombre”. Luego continuamos con la siguiente línea la cual tiene un modificador de acceso y también la clase, para de esta manera declarar que valor van a obtener las variables antes usadas. En este caso sería de esta manera “public Tarea(String nombre,String descripcion, String responsable)”

Luego debemos de declarar que valor van a tener cada una de estas variables mediante los siguiente, en el caso del primer atributo quedaría de la siguiente manera:0 “this.nombre = nombre” Con esto se la da un valor a cada variable además de darle un valor especifico a una variable siendo esto: “this.estado = "Pendiente"”

Luego de esto tenemos un método el cual tiene un modificador de acceso y también se le define un tipo y se le da un nombre a su variable, en este caso: public String obtenerNombre()”,

está declarado de esta manera porque necesitamos verificar un String pero más tarde en el código. Obteniendo el resultado del nombre de “Tarea”, mediante lo siguiente “return nombre”.

Luego tenemos otro método con lo siguiente: “public void completar()”, dentro de este método le cambiamos el valor a una variable antes declarada mediante la siguiente línea: “this.estado = "Completada"”, donde la variable estado con valor “Pendiente” , ahora tiene el valor “Completada”

Por ultimo cerramos con el siguiente método: “public void tareaInformormacion()” el cual contiene lo siguiente: “System.*out*.println("Tarea: "+ nombre + ", Descripcion: " + descripcion + ", Responsable: " + responsable + ", Estado: " + estado)”, el cual mediante una línea en la consola nos da la información de las variables de “Tarea”

Terminamos con la clase “Tarea”, ahora continuamos con la clase “Proyecto”

Primero tenemos “package Proyecto”, esto indica que esta dentro de una carpeta llmada proyecto, luego tenemos “import” en dos líneas, estos comandos son para llamar librerías de Java, siendo la primera una parte que se incluye dentro de la librería “java.util.List”, solo que igual necesita se declara para su uso, en esta caso hablo de “java.util.ArrayList”.

Siendo “List” como su nombre lo indica una lista que permite valores duplicados y llamarlos mediante índices, en cambio “ArrayList”, es una lista redimensionable. Luego de esto tenemos la clase “Proyecto”, declara como “class Proyecto”.

Una vez declara la clase le damos atributos, los cuales tienen un modificador de acceso público, la manera en la que se le declararía seria la siguiente “public (Tipo de varibale) <Variable>”, en este caso quedando de la siguiente manera: “public String nombre”.

La línea declara la primera lista de la siguiente manera “public List<Clase a llamar> <Variable>” En este caso se llamada a la clase “Tarea” de la siguiente manera “public List<Tarea> tareas”.

En la siguiente línea se declara nuevamente un modificador de acceso y también la clase, para de esta manera cuando poderle dar atributos y declarar que valor van a obtener las variables antes usadas. En este caso sería de esta manera “public Proyecto(String nombre)” para darle un valor a la variable nombre.

Ahora continuamos con lo que esta dentro de esta sección “this.nombre = nombre”, de esta manera le damos un valor a la variable. “this.tareas = new ArrayList <> ()”, en este caso es una ArrayList, es decir que la lista “tareas” va ser un arreglo redimensionable.

Luego de esto tenemos un método el cual tiene un modificador de acceso y también se le define un tipo y se le da un nombre a su variable, en este caso: “public String obtenerNombreProyecto()”, esta declarado de esta manera porque necesitamos verificar un String pero mas tade en el código.

Esta línea nos va a retornar el nombre que se la ha asignado al proyecto, mediante la siguiente línea “return nombre”.

En la siguiente línea tenemos un método el cual tiene un modificador de acceso, un tipo de retorno que no devuelve ningún valor llamado “void”, luego una variable, acompañado de los atributos que pertenecen a la clase “Tarea”, dándonos como resultado lo siguiente: “public void agregarTarea(String nombre, String descripcion, String responsable)”

Siendo que en este clase es donde recién declaramos el valor que le vamos a dar a cada uno de los atributos o variables que corresponden a “Tarea”. Continuamos con “Tarea

nuevaTarea = new Tarea(nombre,descripcion, responsable), donde se declara el nombre donde se van a guardar los atributos de “Tarea”.

Como última parte de esta sección tenemos: “tareas.add(nuevaTarea)”, la cual llama al arreglo redimensionable, para guardar dentro de este la información de los atributos de “Tarea”, de manera que podemos tener varios de estos.

Luego tenemos “public void cambiarEstadoTarea(String nombre)” es decir otro declaración de método, donde esta vez solo vamos a usar en este caso el nombre de “Tarea”, para lo que esta dentro de esta sección.

Dentro de esta sección tenemos un bucle “for (Tarea tarea : tareas)”, donde el for realizar una búsqueda dentro de “Tarea”, dándoles el valor “tarea” que es temporal, para el valor del arreglo “tareas”, esto con el objetivo de realizar lo siguiente.

Se va a verificar si un método de “Tarea”, llamado como “tarea.obtenerNombre” es igual al nombre de que esta dentro del arreglo mediante lo siguiente, si es verdadero se va a realizar el llamado de un método de “Tarea” llamado “tarea.completar()”. Esto mediante lo siguiente:

“if (tarea.obtenerNombre().equalsIgnoreCase(nombre))

{tarea.completar()”

Donde el “if” verifica si el nombre seleccionado corresponde a alguno dentro del arreglo, en el caso que de se cierto también nos da lo siguiente: “System.*out*.println("Tarea marcada como completada.")” el cual es una línea de texto que nos indica que, si se realizó el llamado, además de un “return” para terminar el bucle en caso de ser verdadero.

En caso que no sea verdadero es decir que no se encontró el nombre dentro del arreglo tenemos lo siguiente: “System.*out*.println("Tarea no encontrada.")”, el cual es una línea de la consola la cual nos indica que no se encontró la tarea.

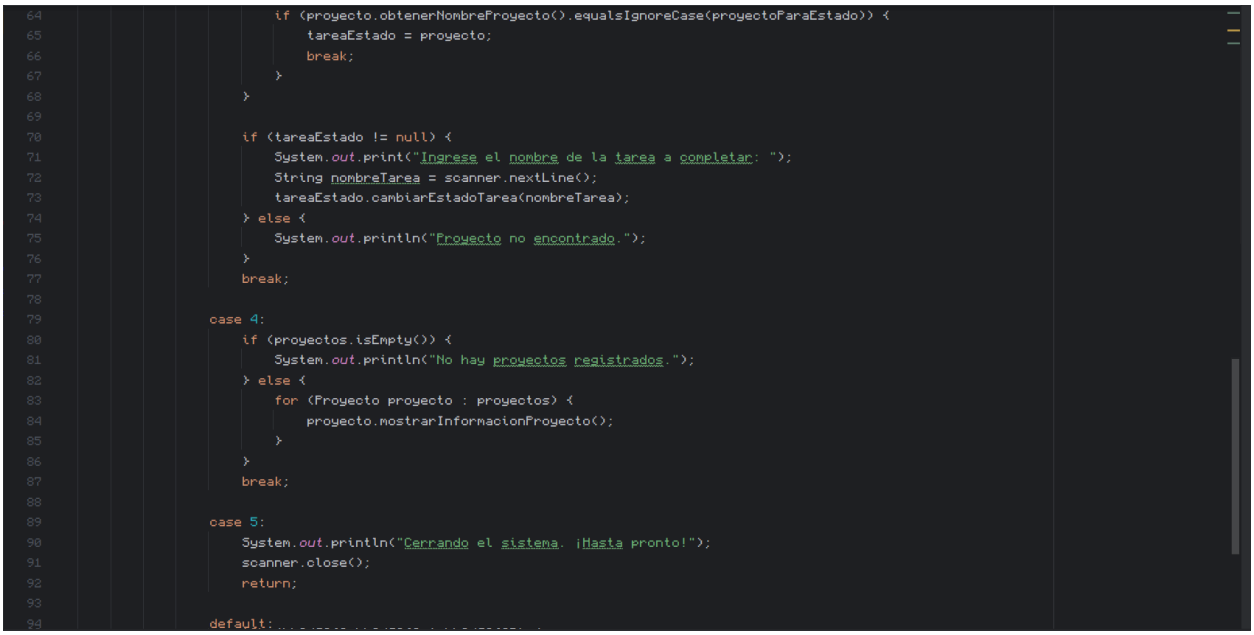
Tenemos como ultimo método: “public void mostrarInformacionProyecto()” el cual contiene lo siguiente “System.*out*.println("Proyecto: " + nombre)” una línea de la consola la cual indica el nombre del proyecto.

Continuamos con un bucle: “for (Tarea tarea : tareas) {

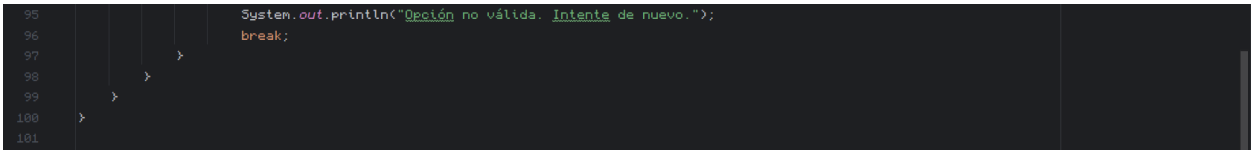
tarea.tareaInformormacion();” el cual es otro bucle que recorre cada nombre dentro del arreglo, para después llamar a un método dentro de “Tarea” llamado: “tarea.tareaInformormacion()”

Una vez explicada cada clase expliquemos que hace el código principal:

```java
package Proyecto;

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Gestion {
    public static void main(String[] args) {
        List<Proyecto> proyectos = new ArrayList<>();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nMenú de Gestión de Proyectos:");
            System.out.println("1. Añadir Proyecto");
            System.out.println("2. Asignar Tarea");
            System.out.println("3. Cambiar Estado de Tarea");
            System.out.println("4. Ver Proyectos");
            System.out.println("5. Cerrar");
            System.out.print("Seleccione una opción: ");

            int opcion = scanner.nextInt();
            scanner.nextLine();

            switch (opcion) {
                case 1:
                    System.out.print("Ingrese el nombre del proyecto: ");
                    String nombreProyecto = scanner.nextLine();
                    proyectos.add(new Proyecto(nombreProyecto));
                    System.out.println("Proyecto añadido correctamente.");
                    break;

                case 2:
                    System.out.print("Ingrese el nombre del proyecto donde asignar la tarea: ");
                    String proyectoParaTarea = scanner.nextLine();
                    Proyecto proyectoEncontrado = null;

                    for (Proyecto proyecto : proyectos) {
                        if (proyecto.obtenerNombreProyecto().equalsIgnoreCase(proyectoParaTarea)) {
                            proyectoEncontrado = proyecto;
                            break;
                        }
                    }

                    if (proyectoEncontrado != null) {
                        System.out.print("Ingrese el nombre de la tarea: ");
                        String nombreTarea = scanner.nextLine();
                        System.out.print("Ingrese la descripción de la tarea: ");
                        String descripcionTarea = scanner.nextLine();
                        System.out.print("Ingrese el responsable de la tarea: ");
                        String responsable = scanner.nextLine();
                        proyectoEncontrado.agregarTarea(nombreTarea,descripcionTarea, responsable);
                        System.out.println("Tarea asignada correctamente.");
                    } else {
                        System.out.println("Proyecto no encontrado.");
                    }
                    break;

                case 3:
                    System.out.print("Ingrese el nombre del proyecto donde está la tarea: ");
                    String proyectoParaEstado = scanner.nextLine();
                    Proyecto tareaEstado = null;

                    for (Proyecto proyecto : proyectos) {
```



```
                    if (proyecto.obtenerNombreProyecto().equalsIgnoreCase(proyectoParaEstado)) {
                        tareaEstado = proyecto;
                        break;
                    }
                }

                if (tareaEstado != null) {
                    System.out.print("Ingrese el nombre de la tarea a completar: ");
                    String nombreTarea = scanner.nextLine();
                    tareaEstado.cambiarEstadoTarea(nombreTarea);
                } else {
                    System.out.println("Proyecto no encontrado.");
                }
                break;

            case 4:
                if (proyectos.isEmpty()) {
                    System.out.println("No hay proyectos registrados.");
                } else {
                    for (Proyecto proyecto : proyectos) {
                        proyecto.mostrarInformacionProyecto();
                    }
                }
                break;

            case 5:
                System.out.println("Cerrando el sistema. ¡Hasta pronto!");
                scanner.close();
                return;

            default:
```



```
                System.out.println("Opción no válida. Intente de nuevo.");
                break;
        }
    }
}
```

Primero tenemos “package Proyecto”, que nos indica que esta dentro de una carpeta, luego de eso tenemos de nuevo los “import” y el nombre de la librería que se usan dentro de “Proyecto” e incluimos la siguiente libreia “java.util.Scanner” para usarla después.

Declaramos la clase con el modificador de acceso en este caso como: “public class Gestion”, luego inicializamos al aplicación con la siguiente línea: “public static void main(String[] args)”. Continuamos declarando un nuevo arreglo redimensionable de esta manera: “List<Proyecto> proyectos = new ArrayList<>()”, el cual es un arreglo de la clase “Proyecto”.

Continuamos creando un scanner, mediante la siguiente línea: “Scanner scanner = new Scanner(System.*in*)” para poder leer las entradas que vamos a introducir en la consola. Luego un bucle que se ejecuta todo el tiempo ya que no tiene condición falsa, mediante la siguiente línea “while (true)”.

Dentro de este bucle tenemos un menú de texto que se muestra mediante la consola, la cual dependiendo del numero que escribamos se va a leer y dar una respuesta, la línea que lee el numero necesario es: “int opcion = scanner.nextInt()”, donde se le da el dato y lo almacena para lo siguiente.

Para cambiar de opciones se lee la información antes dada y mediante la siguiente línea se van dando diferentes opciones: “switch (opcion)”, donde dependiendo del numero que pusimos en la anterior línea se da un “case” u otro.

Siendo este el menú:

"1. Añadir Proyecto"

"2. Asignar Tarea"

“3. Cambiar Estado de Tarea"

"4. Ver Proyectos"

"5. Cerrar"

Comenzamos con el “case 1” el cual presenta una línea de texto en la consola que nos pide darle un nombre al proyecto: “System.*out*.print("Ingrese el nombre del proyecto: ")”, luego tenemos un “scanner” que lee esta información y la almacena dentro de otra variable la cual va a pertenecer al arreglo de “Proyecto”: “String nombreProyecto = scanner.nextLine()”

En la siguiente línea guardamos ese dato dentro del arreglo mediante: “proyectos.add(new Proyecto(nombreProyecto))” para introducirlo dentro del arreglo. Luego obtenemos un mensaje de confirmación mediante la consola: “System.*out*.println("Proyecto añadido correctamente.")”

Terminamos con un “break” para cerrar el case y devolvernos al menú principal.

Continuamos con el “case 2” el cual nos dice mediante una línea en la consola ingresar el nombre del proyecto: “System.*out*.print("Ingrese el nombre del proyecto donde asignar la tarea: ")”, luego tenemos un “scanner” que lee esta información y la almacena dentro de otra variable: “String proyectoParaTarea = scanner.nextLine()”.

También declaramos otra variable: “Proyecto proyectoEncontrado = null” la cual pertenece a “Proyecto” y esta variable teniendo el valor nulo. Luego tenemos: “for (Proyecto proyecto : proyectos)” el cual es un bucle que recorre cada nombre dentro del arreglo para luego mediante un “if” verificar lo siguiente:

“if (proyecto.obtenerNombreProyecto().equalsIgnoreCase(proyectoParaTarea)) {

proyectoEncontrado = proyecto” en este caso se llamada al método proyecto.obtenerNombreProyecto, para obtener el nombre del proyecto que está dentro del arreglo, comparándolo con “proyectoParaTarea”, en el caso de ser cierto se le da un valor a “proyectoEncontrado”

Luego un “brake” para salir del “if”, en el caso de que “proyectoEncontrado” sea diferente a “null”: “if (proyectoEncontrado != null)”, se pide darle un nombre a la tarea, luego una descripción y también un responsable, dándole los valores puestos a los atributos de “Tarea”: “proyectoEncontrado.agregarTarea(nombreTarea,descripcionTarea, responsable)”.

Terminando con un mensaje de confirmación, en el caso de que el valor de “proyectoEncontrado”, no se encuentre dentro del arreglo nos muestra un mensaje de proyecto no encontrado: “System.*out*.println("Proyecto no encontrado.")”. Cerrar el “case” con un “break”.

“case 3”, en este se nos pide ingresar el nombre de la tarea, el cual se lee mediante un “scanner” y se guarda en una variable: “String proyectoParaEstado = scanner.nextLine()”, ahora también se crea otra variable con el valor de “null”: “Proyecto tareaEstado = null”

Luego otro “for” para poder comprobar si la línea leida esta dentro del arreglo de “Proyecto”, luego se verifica si el método “proyecto.obtenerNombreProyecto” es igual a “proyectoParaEstado”, para darle un valor a: “tareaEstado”: “if (proyecto.obtenerNombreProyecto().equalsIgnoreCase(proyectoParaEstado)) {

tareaEstado = proyecto;” Cerramos con un “Break”

Si se verifica que “tareaEstado” es diferente a “null”: “if (tareaEstado != null)” se le pide el nombre de la tarea para poder realizar el cambio de estado. Se lee mediante un “scanner”: “String nombreTarea = scanner.nextLine()”, luego se llama al método “cambiarEstadoTarea”, para darle un valor diferente al atributo estado.

En el caso de no encontrar el proyecto se da una línea en la consola indicando el error, para luego terminar con un “break”.

En el “case 4” se verifica si el arreglo de “Proyecto” tiene información: “if (proyectos.isEmpty())”, en el caso de que no se indica esto en la consola. En el caso de ser verdadero se realiza otro bucle para encontrar el nombre del proyecto dentro del arreglo: “for (Proyecto proyecto : proyectos)”.

Luego se llama un método para poder representar la información de ese proyecto incluido las tareas: “proyecto.mostrarInformacionProyecto()”. Por ultimo se cierra con un “break”.

En el “case 5” se usa para cerrar el sistema, cerrando el “scanner” scanner.close(); y regresar un falso “return”.

En el caso de que no se cumpla ningún “case”, se manda un mensaje en la consola indicando esto y se devuelve al menú principal. Se cierra este último con un “break”. Ahora continuamos con la representación en consola de todo esto:

```
Menú de Gestión de Proyectos:
1. Añadir Proyecto
2. Asignar Tarea
3. Cambiar Estado de Tarea
4. Ver Proyectos
5. Cerrar
Seleccione una opción: 1
Ingrese el nombre del proyecto: Deber1
Proyecto añadido correctamente.
```

```
Menú de Gestión de Proyectos:
1. Añadir Proyecto
2. Asignar Tarea
3. Cambiar Estado de Tarea
4. Ver Proyectos
5. Cerrar
Seleccione una opción: 2
Ingrese el nombre del proyecto donde asignar la tarea: Deber1
Ingrese el nombre de la tarea: Hacer deber
Ingrese la descripción de la tarea: Realizar el deber de POO
Ingrese el responsable de la tarea: Cristina
Tarea asignada correctamente.
```

```
Menú de Gestión de Proyectos:
1. Añadir Proyecto
2. Asignar Tarea
3. Cambiar Estado de Tarea
4. Ver Proyectos
5. Cerrar
Seleccione una opción: 3
Ingrese el nombre del proyecto donde está la tarea: Deber1
Ingrese el nombre de la tarea a completar: Hacer deber
Tarea marcada como completada.
```

```
Menú de Gestión de Proyectos:
1. Añadir Proyecto
2. Asignar Tarea
3. Cambiar Estado de Tarea
4. Ver Proyectos
5. Cerrar
Seleccione una opción: 4
Proyecto: Deber1
Tarea: Hacer deber, Descripcion: Realizar el deber de POO, Responsable: Cristina, Estado: Completada
```

```
Menú de Gestión de Proyectos:
1. Añadir Proyecto
2. Asignar Tarea
3. Cambiar Estado de Tarea
4. Ver Proyectos
5. Cerrar
Seleccione una opción: 5
Cerrando el sistema. ¡Hasta pronto!

Process finished with exit code 0
```

```
Menú de Gestión de Proyectos:
1. Añadir Proyecto
2. Asignar Tarea
3. Cambiar Estado de Tarea
4. Ver Proyectos
5. Cerrar
Seleccione una opción: 4
No hay proyectos registrados.
```

```
Menú de Gestión de Proyectos:
1. Añadir Proyecto
2. Asignar Tarea
3. Cambiar Estado de Tarea
4. Ver Proyectos
5. Cerrar
Seleccione una opción: 2
Ingrese el nombre del proyecto donde asignar la tarea: Deber2
Proyecto no encontrado.
```

Comprobando de esta manera cada “case” y cada condición.