
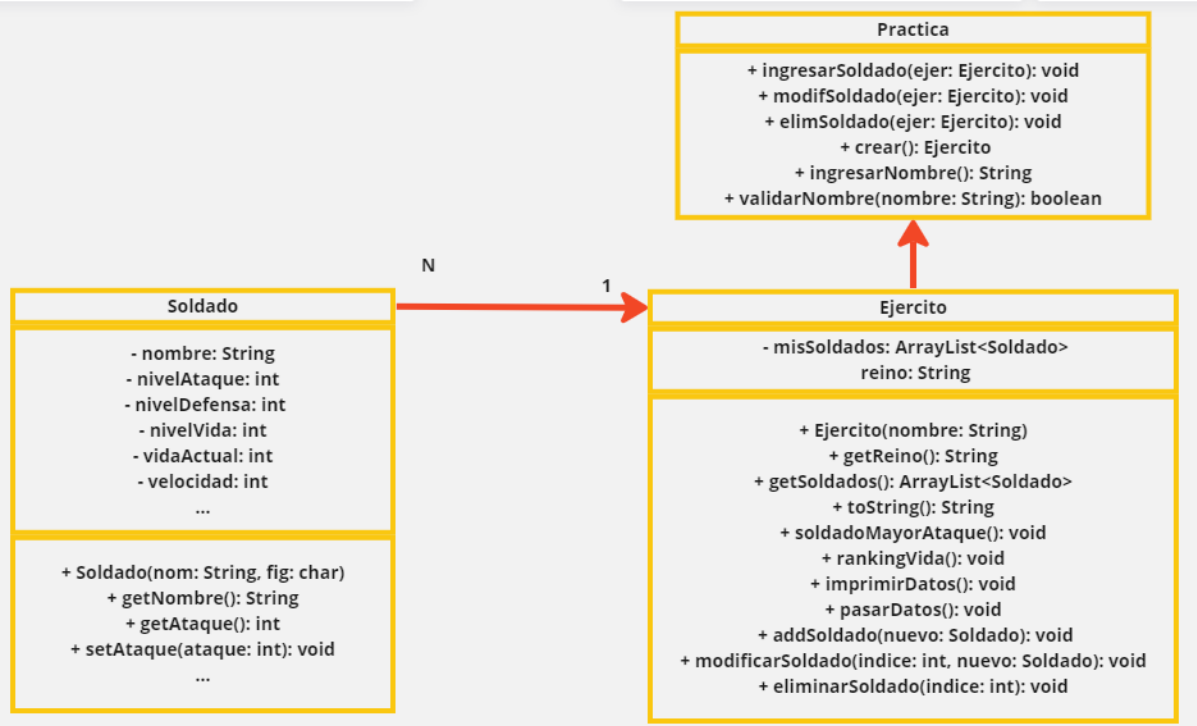


	<p align="center">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		
Aprobación: 2022/03/01	Código: GUIA-PRLE-001	Página: 1

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA					
ASIGNATURA:	Fundamentos de la Programacion 2				
TÍTULO DE LA PRÁCTICA:	<i>Laboratorio 15</i>				
NÚMERO DE PRÁCTICA:	<i>15</i>	AÑO LECTIVO:	<i>1</i>	NRO. SEMESTRE:	<i>2</i>
FECHA DE PRESENTACIÓN	<i>27/12/2023</i>	HORA DE PRESENTACIÓN	<i>5:00 pm</i>		
INTEGRANTE (s): Cuno Cahuari Armando Steven					NOTA:

SOLUCIÓN Y RESULTADOS	
I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS	
 <pre> classDiagram class Soldado { - nombre: String - nivelAtaque: int - nivelDefensa: int - nivelVida: int - vidaActual: int - velocidad: int + Soldado(nom: String, fig: char) + getNombre(): String + getAtaque(): int + setAtaque(ataque: int): void } class Ejercito { - misSoldados: ArrayList<Soldado> - reino: String + Ejercito(nombre: String) + getReino(): String + getSoldados(): ArrayList<Soldado> + toString(): String + soldadoMayorAtaque(): void + rankingVida(): void + imprimirDatos(): void + pasarDatos(): void + addSoldado(nuevo: Soldado): void + modificarSoldado(indice: int, nuevo: Soldado): void + eliminarSoldado(indice: int): void } class Practica { + ingresarSoldado(ejer: Ejercito): void + modifSoldado(ejer: Ejercito): void + elimSoldado(ejer: Ejercito): void + crear(): Ejercito + ingresarNombre(): String + validarNombre(nombre: String): boolean } Soldado "N" --> "1" Ejercito Ejercito --> Practica </pre> <p>The diagram illustrates the structure of the laboratory solution. It features three classes: Soldado, Ejercito, and Practica. Soldado is a class with attributes like <code>nombre</code>, <code>nivelAtaque</code>, <code>nivelDefensa</code>, <code>nivelVida</code>, <code>vidaActual</code>, and <code>velocidad</code>. It has methods for creating, getting, and setting attributes. Ejercito is a class that contains a collection of Soldado objects (<code>misSoldados: ArrayList<Soldado></code>) and a <code>reino</code> attribute. It includes methods for creating, getting, and modifying the collection of soldiers, as well as other game-related methods like <code>soldadoMayorAtaque</code> and <code>rankingVida</code>. Practica is a class that provides methods to interact with the Ejercito class, such as <code>ingresarSoldado</code>, <code>modifSoldado</code>, and <code>elimSoldado</code>. The relationships are defined as follows: Soldado has an association with Ejercito (N to 1), and Ejercito has an association with Practica.</p>	
<p>El laboratorio nos dice que creamos una clase Ejercito y este almacene como atributo un arraylist de soldados para ello</p>	

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 2</p>

```
public class Ejercito {
    private ArrayList<Soldado> misSoldados;
    private String reino;
    public Ejercito(String nombre){
        reino = nombre;
        misSoldados = new ArrayList<Soldado>();
    }
}
```

Mediante la instanciación de Ejercito también se instancia el arreglo de Soldado con identificador llamado misSoldados, asimismo es necesario asignarle un nombre a este reino por lo cual solo hay un conjunto de nombres los cuales son aceptables desde el método main nosotros creamos un método para que se ingrese el nombre correcto instanciando desde el mismo objeto Ejercito.

```
public static Ejercito crear(){
    Scanner sc = new Scanner(System.in);
    String nombre = ingresarNombre();
    return new Ejercito(nombre);
}
public static String ingresarNombre(){
    Scanner sc = new Scanner(System.in);
    System.out.println("Ingrese el nombre del ejercito solo validos :");
    System.out.println("Inglaterra - Francia - Castilla o Aragon - Sacro - Moros");
    String nombre = sc.next().toUpperCase();
    while(!validarNombre(nombre)){
        System.out.println("Reino incorrecto ingrese otro");
        nombre = sc.next().toUpperCase();
    }
    return nombre;
}
```

Este nombre debe ser ingresado y este pasa por un método booleano llamado validarNombre encargado de verificar si el nombre es correcto.

```
public static boolean validarNombre(String nombre){
    return nombre.equals("INGLATERRA") || nombre.equals("FRANCIA") || nombre.equals("CASTILLA") || nombre.equals("ARAGON") || nombre.equals("SACRO") || nombre.equals("MOROS");
}
```

Asimismo al crear este objeto Ejercito nosotros podemos acceder a él mediante 3 opciones: uno para agregar, otro para modificar y el último para eliminar. Estas opciones las colocamos mediante un menú el cual el usuario podrá ingresar una opción y se realiza diversa tarea.

```
System.out.println("Hay " + ejercito.getSoldados().size() + " soldados");
System.out.println("Ingrese una opcion:");
opcion = sc.nextInt();
switch(opcion){
    case 1:
        System.out.println("Ingresar un nuevo soldado: ");
        ingresarSoldado(ejercito);
        break;
    case 2:
        System.out.println("Modificar un soldado: ");
        modifSoldado(ejercito);
        break;
    case 3:
        System.out.println("Eliminar un soldado: ");
        elimSoldado(ejercito);
        break;
```

Los métodos como bien son ingresarSoldado, modifSoldado y elimSoldado

```
public static void ingresarSoldado(Ejercito ejer){
    Scanner sc = new Scanner(System.in);
    if(ejer.getSoldados().size() <= 10 && ejer.getSoldados().size() >= 0){
        System.out.println("Desea ingresar un soldado autogenerado o personalizado\nAutogenerado - 1\nPersonalizado - 2");
        int opcion = sc.nextInt();
        if(opcion == 1){
            ejer.addSoldado(new Soldado(ejer.getReino(), '*'));
        } else {
            System.out.println("Ingrese el nivel de ataque");
            int ataque = sc.nextInt();
            System.out.println("Ingresa el nivel de defensa");
            int defensa = sc.nextInt();
            ejer.addSoldado(new Soldado(ejer.getReino(), ataque, defensa, '*'));
        }
    } else {
        System.out.println("Sobrepasa la capacidad de limite de soldados");
    }
}
```

```
public void addSoldado(Soldado nuevo){
    misSoldados.add(nuevo);
}
```

En el método ingresarSoldado tenemos 2 opciones uno que es aleatorio los valores y otro el cual el usuario puede personalizar mediante el nivel de ataque y defensa.

```
public static void modifSoldado(Ejercito ejer){
    Scanner sc = new Scanner(System.in);
    System.out.println("Ingrese el indice del soldado a modificar");
    int indice = sc.nextInt();
    Soldado ref = ejer.getSoldados().get(indice);
    System.out.println("Ingrese el nivel de ataque");
    int ataque = sc.nextInt();
    System.out.println("Ingrese el nivel de defensa");
    int defensa = sc.nextInt();
    ref.setAtaque(ataque);
    ref.setDefensa(defensa);
    ejer.modificarSoldado(indice, ref);
}
```

```
public void modificarSoldado(int indice, Soldado nuevo){
    misSoldados.set(indice, nuevo);
}
```

El método modifSoldado se encarga de solicitar el índice el cual desea cambiar, los atributos que puede ser cambiados vienen a ser el nivel de ataque y defensa.

```
public static void elimSoldado(Ejercito ejer){
    Scanner sc = new Scanner(System.in);
    System.out.println("Ingrese el indice del soldado a eliminar");
    int indice = sc.nextInt();
    System.out.println("Soldado eliminado: ");
    ejer.getSoldados().get(indice);
    ejer.eliminarSoldado(indice);
}
```

```
public void eliminarSoldado(int indice){
    misSoldados.remove(indice);
}
```

El método elimSoldado pedirá el índice del soldado a eliminar, antes que este se elimine imprimirá todos los datos del soldado a eliminar.

De ahí quedaría el método del soldado con mayor nivel de ataque para ello se hizo una búsqueda lineal y de ahí se imprimiré todos los valores iguales que tuvo esta búsqueda.

```
public void soldadoMayorAtaque(){
    int mayor = 0;
    for(int i = 1; i < misSoldados.size(); i++){
        if(misSoldados.get(mayor).getAtaque() < misSoldados.get(i).getAtaque()){
            mayor = i;
        }
    }
    System.out.println("Los soldados con mayor ataque son: ");
    for(Soldado may: misSoldados){
        if(may.getAtaque() == misSoldados.get(mayor).getAtaque()){
            System.out.println(may);
        }
    }
}
```

El método de ranking de vida ya trabajado anteriormente.



```
public void rankingVida(){
    ArrayList<Soldado> copia = new ArrayList<Soldado>();
    pasarDatos(misSoldados, copia);
    for(int i = 0; i < copia.size() - 1; i++){
        for(int j = i + 1; j < copia.size(); j++){
            Soldado may = copia.get(i);
            Soldado men = copia.get(j);
            if(men.getVida() > may.getVida()){
                Soldado mayor = new Soldado(men.getNombre(), men.getAtaque(), men.getDefensa(), men.getVida(), men.getN);
                Soldado menor = new Soldado(may.getNombre(), may.getAtaque(), may.getDefensa(), may.getVida(), may.getN);
                copia.set(i, mayor);
                copia.set(j, menor);
            }
        }
    }
    System.out.println("Reino " + reino);
    for(Soldado m: copia){
        System.out.println("\n" + m);
    }
}
```

También el método que imprime los datos de los soldados de un ejército

```
public void imprimirDatos(){
    System.out.println("Reino " + reino);
    for(Soldado m: misSoldados){
        System.out.println("\n" + m);
    }
}
```

El menú para que se lleven acabo estas diversas tareas fue realizado mediante un booleano, una variable de opción y el uso de switch.



```
opcion = sc.nextInt();
switch(opcion){
    case 1:
        System.out.println("Ingresar un nuevo soldado: ");
        ingresarSoldado(ejercito);
        break;
    case 2:
        System.out.println("Modificar un soldado: ");
        modifSoldado(ejercito);
        break;
    case 3:
        System.out.println("Eliminar un soldado: ");
        elimSoldado(ejercito);
        break;
    case 4:
        System.out.println("Mayor nivel de ataque");
        ejercito.soldadoMayorAtaque();
        break;
    case 5:
        System.out.println("Ranking de vida");
        ejercito.rankingVida();
        break;
    case 6:
        System.out.println("Datos del ejercito");
        ejercito.imprimirDatos();
        break;
    case 7:
        System.out.println("Saliendo...");
        validez = false;
        break;
}
```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 7</p>



```

Creacion del ejercito
Ingrese el nombre del ejercito solo validos :
Inglaterra - Francia - Castilla o Aragon - Sacro - Moros
inglaterra
*****
Hay 0 soldados
Ingrese una opcion:
1
Ingresar un nuevo soldado:
Desea ingresar un soldado autogenerado o personalizado
Autogenerado - 1
Personalizado - 2
1
*****
*****
Hay 1 soldados
Ingrese una opcion:
1
Ingresar un nuevo soldado:
Desea ingresar un soldado autogenerado o personalizado
Autogenerado - 1
Personalizado - 2
1
*****
*****
Hay 2 soldados
Ingrese una opcion:
1
Ingresar un nuevo soldado:
Desea ingresar un soldado autogenerado o personalizado
Autogenerado - 1
Personalizado - 2
1
*****

```

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 8</p>

<pre> ***** Hay 3 soldados Ingrese una opcion: 1 Ingresar un nuevo soldado: Desea ingresar un soldado autogenerado o personalizado Autogenerado - 1 Personalizado - 2 2 Ingrese el nivel de ataque 3 Ingres el nivel de defensa 5 ***** ***** Hay 4 soldados Ingrese una opcion: 2 Modificar un soldado: Ingrese el indice del soldado a modificar 2 Ingrese el nivel de ataque 3 Ingrese el nivel de defensa 1 ***** ***** Hay 4 soldados Ingrese una opcion: 3 Eliminar un soldado: Ingrese el indice del soldado a eliminar 1 Soldado eliminado: ***** </pre>	
--	--

	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p style="text-align: center;">Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 9</p>

Hay 3 soldados

Ingrese una opcion:

4

Mayor nivel de ataque

Los soldados con mayor ataque son:

Nombre: INGLATERRA Vida:5 Fila:8 Ataque:4 Defensa:4 Columna:H Actitud:Defensiva

Hay 3 soldados

Ingrese una opcion:

5

Ranking de vida

Reino INGLATERRA

Nombre: INGLATERRA Vida:5 Fila:8 Ataque:4 Defensa:4 Columna:H Actitud:Defensiva

Nombre: INGLATERRA Vida:5 Fila:1 Ataque:3 Defensa:1 Columna:C Actitud:Defensiva

Nombre: INGLATERRA Vida:4 Fila:7 Ataque:3 Defensa:5 Columna:I Actitud:Defensiva

Hay 3 soldados

Ingrese una opcion:

6

Datos del ejercito

Reino INGLATERRA

Nombre: INGLATERRA Vida:5 Fila:8 Ataque:4 Defensa:4 Columna:H Actitud:Defensiva

Nombre: INGLATERRA Vida:5 Fila:1 Ataque:3 Defensa:1 Columna:C Actitud:Defensiva

Nombre: INGLATERRA Vida:4 Fila:7 Ataque:3 Defensa:5 Columna:I Actitud:Defensiva

Hay 3 soldados

Ingrese una opcion:

7



Saliendo...

II. CONCLUSIONES

Para cuando se indique la opción o la tarea que se quiere realizar es necesario una interfaz o algo que ayude para orientarnos mejor que opción hace cada una asimismo el no uso de copiar código empleando el uso mediante métodos necesarios

RETROALIMENTACIÓN GENERAL

Este laboratorio nos ayuda mucho ya que con este podemos hacer uso de una clase dentro de otra asimismo facilita la lógica que nosotros podemos hacer mediante este tipo de códigos asimismo enfrentar diversos problemas dados en la informática para resolver este.

	<p>UNIVERSIDAD NACIONAL DE SAN AGUSTIN FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMA</p>	
<p>Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación</p>		
<p>Aprobación: 2022/03/01</p>	<p>Código: GUIA-PRLE-001</p>	<p>Página: 10</p>

REFERENCIAS Y BIBLIOGRAFÍA

<https://docs.oracle.com/javase/8/docs/api/>