

Informe de Laboratorio 22

Tema: Laboratorio 22

Nota

Estudiante	Escuela	Asignatura
Victor Mamani Anahua vmamanian@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Fundamentos de la Programación II Semestre: II Código: 20230489

Laboratorio	Tema	Duración
22	Laboratorio 22	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - B	Del 15 Enero 2024	Al 22 Enero 2024

1. Tarea

- Cree una versión del videojuego de estrategia usando componentes básicos GUI: Etiquetas, botones, cuadros de texto, JOptionPane, Color.
- Además, utilizar componentes avanzados GUI: Layouts, JPanel, áreas de texto, checkbox, botones de radio y combobox.
- Considerar nivel estratégico y táctico.
- Considerar hasta las unidades especiales de los reinos.
- Hacerlo iterativo.

2. Equipos, materiales y temas utilizados

- Sistema Operativo Ubuntu GNU Linux 23 lunar 64 bits Kernell 6.2.v
- Visual Studio Code.
- VIM 9.0.
- OpenJDK 64-Bits 19.0.7.
- Git 2.39.2.
- Cuenta en GitHub con el correo institucional.
- Programación Orientada a Objetos.
- Actividades del Laboratorio 22.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/VictorMA18/fp2-23b.git>
- URL para el laboratorio 22 en el Repositorio GitHub.
- <https://github.com/VictorMA18/fp2-23b/tree/main/Fase03/Lab22>

4. Actividades del Laboratorio 22

4.1. Ejercicio Soldado(Herencias)

- En esta seccion solo reutilizamos las Herencias de la clase Soldado.
- El codigo y el commit seria el siguiente:

Listing 1: Commit

```
$ git commit -m "Agregando la clase Soldado y Espadachin para poder hacer el juego  
bueno solo en la clase espadachin usamos la herencia que nos deja la clase Soldado  
y tambien creamos la funcion muroEscudo() la cual devuelve como mensaje el uso de  
esta habilidad defensiva y los getters y setters"
```

Listing 2: Las lineas de codigos de la clase Espadachin creada:

```
public class Espadachin extends Soldado{  
    private int swordlth;  
    public Espadachin(String name , int attacklevel, int defenselevel, int lifelevel,  
        int speed, String attitude ,boolean lives, int row, String column, int swordlth){  
        super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,  
            column);  
        this.swordlth = swordlth;  
    }  
    public void muroEscudo(){  
        System.out.println("Usted uso la habilidad muro de Escudos");  
    }  
    public int getSwordlth(){  
        return swordlth;  
    }  
    public void setSwordlth(int n){  
        this.swordlth = n;  
    }  
}
```

Listing 3: Las lineas de codigos de la clase Caballero creada:

```
public class Caballero extends Soldado{  
    private boolean montar;  
    private String arma;  
    public Caballero(){  
    }  
    public Caballero(String name , int attacklevel, int defenselevel, int lifelevel, int  
        speed, String attitude ,boolean lives, int row, String column,boolean montar){
```

```
        super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,
              column);
        this.montar = montar;
    }
    public void montar(){
        if(!this.montar){
            this.arma = "Lanza";
            this.embestir();
        }
    }
    public void desmontar(){
        if (this.montar) {
            this.arma = "Espada";
        }
    }
    public void embestir(){
        if(!montar){
            this.atacar();
            this.atacar();
        }else{
            this.atacar();
            this.atacar();
            this.atacar();
        }
    }
    public String getArma(){
        return arma;
    }
}
```

Listing 4: Las líneas de códigos de la clase Arquero creada:

```
public class Arquero extends Soldado{
    private int flechas;
    public Arquero(){

    }
    public Arquero(String name , int attacklevel, int defenselevel, int lifelevel, int
        speed, String attitude ,boolean lives, int row, String column, int flechas){
        super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,
            column);
        this.flechas = flechas;
    }
    public void disparar(){
        if(this.flechas == 0){
            System.out.println("El arquero ya tiene flechas para poder disparar");
        }else{
            this.flechas = flechas - 1;
            this.atacar();
        }
    }
    public void setFlechas(int n){
        this.flechas = n;
    }
    public int getFlechas(){
        return flechas;
    }
}
```

```
}  
}
```

Listing 5: Las líneas de códigos de la clase Lancero creada:

```
public class Lancero extends Soldado{  
    private int lancelth;  
    public Lancero(){  
    }  
    public Lancero(String name , int attacklevel, int defenselevel, int lifelevel, int  
        speed, String attitude ,boolean lives, int row, String column, int lancelth){  
        super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,  
            column);  
        this.lancelth = lancelth;  
    }  
    public void schiltrom(){  
        this.setDefenseLevel(this.getDefenseLevel() + 1);  
        System.out.println("El lancero uso el schiltrom su nivel de defensa subio 1  
            punto");  
    }  
    public void setLancelth(int n){  
        this.lancelth = n;  
    }  
    public int getLancelth(){  
        return lancelth;  
    }  
}
```

Listing 6: Las líneas de códigos de la clase Soldado creada:

```
// Laboratorio Nro 22 - Ejercicio Soldado  
// Autor: Mamani Anahua Victor Narciso  
// Colaboro:  
// Tiempo:  
import java.util.*;  
public class Soldado { //CREAMOS LA CLASE SOLDADO PARA PODER USAR UN ARREGLO  
    BIDIMENSIONAL DONDE NECESITAMOS LA VIDA , EL NOMBRE DEL SOLDADO Y TAMBIEN SU  
    POSICION COMO LA FILA Y LA COLUMNA  
  
    private String name;  
    private int lifeactual;  
    private int row;  
    private String column;  
    private int attacklevel;  
    private int defenselevel;  
    private int lifelevel;  
    private int speed;  
    private String attitude;  
    private boolean lives;  
  
    Random rdm = new Random();  
  
    //Anadiendo metodo que nos permita que un arreglo tenga datos nulos si este esta  
    vacio  
    public Soldado(){
```

```
this.name = "";
this.row = 0;
this.column = "";
this.attacklevel = 0;
this.defenselevel = 0;
this.lifelevel = 0;
this.lifeactual = 0;
this.speed = 0;
this.attitude = "";
this.lives = false;
}

//Constructor
public Soldado(String name, int health, int row, String column){
    this.name = name;
    this.lifeactual = health;
    this.lifelevel = health;
    this.lifeactual = health;
    this.row = row;
    this.column = column;
    this.lives = true;

    //YA QUE ESTOS DATOS SERIAN ALEATORIOS YA QUE SE ESTARIA CREANDO EL SOLDADO
    TENDRIAMOS DATOS QUE SERIAN COMO ATTACKLEVEL DEFENSELEVEL EL CUAL TENDRIAN
    QUE SER ALEATORIOS
    this.attacklevel = rdm.nextInt(5) + 1;
    this.defenselevel = rdm.nextInt(5) + 1;
}

//Constructor para los diferentes niveles como de vida defensa ataque velocidad
public Soldado(String name , int attacklevel, int defenselevel, int lifelevel, int
    speed, String attitude ,boolean lives, int row, String column) {
    this.name = name;
    this.attacklevel = attacklevel;
    this.defenselevel = defenselevel;
    this.lifeactual = lifelevel;
    this.lifelevel = lifelevel;
    this.speed = speed;
    this.lives = lives;
    this.row = row;
    this.column = column;
    this.attitude = attitude;
}

//Metodos necesarios como avanzar defender huir al ser atacado al retroceder
public void advance(){
    this.speed = getSpeed() + 1;
    System.out.println("El soldado " + this.name + "avanzo");
}
public void defense(){
    this.speed = 0;
    this.attitude = "DEFENSIVA";
    System.out.println("El soldado " + this.name + "esta defendiendo");
}
public void flee(){
```

```
        this.speed = getSpeed() + 2;
        this.attitude = "HUYE";
        System.out.println("El soldado " + this.name + "esta huyendo");
    }
    public void back(){
        System.out.println("El soldado " + this.name + "esta retrocediendo");
        if(this.speed == 0){
            this.speed = rdm.nextInt(5) - 5;
        }else{
            if(this.speed > 0){
                this.speed = 0;
                this.attitude = "DEFENSIVA";
            }
        }
    }
    public void atacar(){
        this.speed += 1;
        this.attitude = "Atacar";
        this.lifeactual += 1;
        if(this.lifeactual == 0){
            morir();
        }
    }
    public void attack(Soldado soldier){
        if(this.getLifeActual() > soldier.getLifeActual()){
            int life = this.getLifeActual() - soldier.getLifeActual();
            this.setLifeActual(life);
            this.setLifeLevel(life);
            soldier.lives = false;
            soldier.morir();
            System.out.println(this.name + " asesino al soldado " + soldier.name);
        }else if(soldier.getLifeActual() > this.getLifeActual()){
            int life = soldier.getLifeActual() - this.getLifeActual();
            this.lives = false;
            this.morir();
            soldier.setLifeActual(life);
            soldier.setLifeLevel(life);
            System.out.println(soldier.name + " asesino al soldado " + this.name);
        }else{
            this.lives = false;
            this.morir();
            soldier.lives = false;
            soldier.morir();
            System.out.println("los 2 soldados se asesinaron");
        }
    }
    public void morir(){
        this.lives = false;
        this.attitude = "SOLDADO MUERTO";
    }
    // Metodos mutadores
    public void setName(String n){
        name = n;
    }
    public void setLifeActual(int p){
```

```
        lifeactual = p;
    }
    public void setRow(int b){
        row = b;
    }
    public void setColumn(String c){
        column = c;
    }
    public void setAttackLevel(int attacklevel) {
        this.attacklevel = attacklevel;
    }
    public void setDefenseLevel(int defenselevel) {
        this.defenselevel = defenselevel;
    }
    public void setLifeLevel(int lifelevel){
        this.lifelevel = lifelevel;
    }
    public void setSpeed(int speed) {
        this.speed = speed;
    }
    public void setAttitude(String attitude) {
        this.attitude = attitude;
    }
    public void setLives(boolean lives) {
        this.lives = lives;
    }

    // Metodos accesores
    public String getName(){
        return name;
    }
    public int getLifeActual(){
        return lifeactual;
    }
    public int getRow(){
        return row;
    }
    public String getColumn(){
        return column;
    }
    public int getAttackLevel() {
        return attacklevel;
    }
    public int getDefenseLevel() {
        return defenselevel;
    }
    public int getLifeLevel(){
        return lifelevel;
    }
    public int getSpeed() {
        return speed;
    }
    public String getAttitude() {
        return attitude;
    }
    public boolean getLives() {
```

```
        return lives;
    }

    // Completar con otros metodos necesarios
    public String toString(){ //CREAMOS ESTE METODO PARA IMPRIMIR LOS DATOS DEL OBJETO
        String join = "\nNombre: " + getName() + "\nVida: " + getLifeActual() + "\nFila: "
            + getRow() + "\nColumna: " + getColumn() + "\nNivel de ataque: " +
            getAttackLevel() + "\nNivel de Defensa: " + getDefenseLevel() + "\nNivel de
            vida: " + getLifeLevel() + "\nVelocidad: " + getSpeed() + "\nActitud: " +
            getAttitude() + "\nEstado: " + getLives(); //Agregamos un espaciador para
            poder separar
        return join;
    }
}
```

4.2. Ejercicio Ejercito y Mapa

- En esta seccion En la clase Ejercito creamos un constructor null y constructor el cual reciba parametros para reconocer a este le agregamos los metodos viewsoldiers() , longerlife() , averagelife(), rankingInsercionLife() y tambien sus getters y setters esto con el fin que nos ayude a simplificar algunas funciones las cuales sean representadas mejor con una clase Ejercito.
- Por parte de clase Mapa integramos 2 objetos de la clase Ejercito las cuales nos ayudan con sus metodos para ver la informacion de la batalla entre los 2 ejercitos.
- El codigo y el commit seria el siguiente:

Listing 7: Commit

```
$ git commit -m "Creamos una clase Ejercito la cual nos va a poder ayudar al momento de
querer datos de este como los soldados , el soldado de mayor vida promedio y
tambien tenemos en la clase Mapa la cual creamos objetos de esta clase Ejercito
las cuales vamos a poder usar sus metodos"
```

Listing 8: Las lineas de codigos de la clase Ejercito creada:

```
import java.util.*;
class Ejercito {
    private ArrayList<ArrayList<Soldado>> army;
    private String kingdom;
    public Ejercito(){
    }
    public Ejercito(ArrayList<ArrayList<Soldado>> army, String kingdom){
        this.army = army;
        this.kingdom = kingdom;
    }
    public void viewSoldiers(String armyespe, int num, ArrayList<ArrayList<Soldado>>
        army){
        int numbersoldiers = 0;
        System.out.println("El Ejercito " + armyespe + " del " + num + " ejercito sus
            soldados son :");
        for(int i = 0; i < 10; i++){ //ITERACION
            for(int j = 0; j < 10 ; j++){//ITERACION
```



```
        if (army.get(i).get(j) != null){
            System.out.println("\n*****");
            System.out.println("El " + (numbersoldiers + 1) + " soldado es: ");
            System.out.println(army.get(i).get(j).toString());
            numbersoldiers++;
        }
    }
}

public void longerLife(ArrayList<ArrayList<Soldado>> army, String kingdom){
    int mayor = 0; //METODO CREADO PARA PODER PERMITIRNOS A CONOCER EL SOLDADO CON
    MAYOR VIDA DE CADA EJERCITO
    Soldado soldier = null;
    for (int i = 0; i < army.size(); i++){
        for (int j = 0; j < army.get(i).size(); j++){
            if (army.get(i).get(j) != null){ //COMPROBACION QUE HACEMOS PARA PODER DECIR
                QUE EL CASILLERO DONDE ESTAMOS ES UN SOLDADO QUE EXISTE
                if (army.get(i).get(j).getLifeActual() > mayor){ //COMPARAMOS PUNTOS DE
                    VIDA DE CADA SOLDADO PARA VER QUIEN ES EL MAYOR
                        mayor = army.get(i).get(j).getLifeActual();
                        soldier = army.get(i).get(j);
                    }
                }
            }
        }
    }
    System.out.println("El soldado con mayor vida del Ejercito " + kingdom + " es: ");
    System.out.println(soldier.toString()); //IMPRIMIMOS SUS DATOS PARA PODER VER DE
    QUE SOLDADO SE TRATA
    System.out.println("*****");
}

public double averageLife(ArrayList<ArrayList<Soldado>> army, String kingdom){
    int sum = 0;
    int count = 0;
    System.out.println("El promedio de puntos de vida del Ejercito " + kingdom + " es:
    ");
    for (int i = 0; i < 10; i++){
        for (int j = 0; j < 10; j++){ //ITERACION LA CUAL NOS AYUDA A PASAR POR TODOS
            LOS SOLDADOS DE CADA EJERCITO
                if (army.get(i).get(j) != null){
                    sum += army.get(i).get(j).getLifeActual();
                    count++;
                }
            }
        }
    }
    if (sum != 0){
        double avg = sum / (count * 1.0);
        System.out.println(avg); // DAMOS A CONOCER EL PROMEDIO DE VIDA DE CADA EJERCITO
        System.out.println("*****");
        return avg;
    } else {
        double avg = 0;
        System.out.println(avg); // DAMOS A CONOCER EL PROMEDIO DE VIDA DE CADA EJERCITO
        System.out.println("*****");
        return avg;
    }
}
```

```
public void rankingInsercionLife(ArrayList<ArrayList<Soldado>> army , String
kingdom){
    System.out.println("\nEl Ejercito " + kingdom + " ordenando por metodo insercion:
    ");
    int count = 0;
    for(int i = 0; i < 10; i++){
        for(int j = 0; j < 10; j++){ //ITERACION LA CUAL NOS AYUDA A PASAR POR TODOS
            LOS SOLDADOS DE CADA EJERCITO
            if(army.get(i).get(j) != null){
                count++;
            }
        }
    }
    System.out.println("-----");
    System.out.println("Mostrando Ranking del Ejercito " + kingdom + " ..... /////
    --->");
    Soldado[] soldados = new Soldado[count];
    int x = 0;
    for(int i = 0; i < 10; i++){
        for(int j = 0; j < 10; j++){ //ITERACION LA CUAL NOS AYUDA A PASAR POR TODOS
            LOS SOLDADOS AL ARRAY SOLDADO PARA PODER USAR EL USO DEL METODO DE
            ORDENACION INSERCIÓN
            if(army.get(i).get(j) != null){
                if(count - count + x == count){
                    break;
                }else{
                    soldados[count - count + x] = army.get(i).get(j); //LA MISMA LOGICA QUE
                    EL ANTERIOR METODO SOLO QUE EN ESTE LO USARIAMOS DE MANERA
                    DIFERENTE YA QUE ESTE SERIA DE FORMA DE INSERCIÓN
                }
                x++;
            }
        }
    }
    int n = soldados.length;
    for (int i = 1; i < n; i++) {
        Soldado actual = soldados[i];
        int j = i - 1;
        while (j >= 0 && soldados[j].getLifeActual() < actual.getLifeActual()) {
            //ORDENAMOS EL EJERCITO RESPECTIVAMENTE MEDIANTE EL METODO QUE NOS OFRECE
            INSERCIÓN EL CUAL ES ESTE CODIGO
            soldados[j + 1] = soldados[j];
            j--;
        }
        soldados[j + 1] = actual;
    }
    for(int i = 0; i < soldados.length; i++){
        System.out.print("\n" + "Puesto " + (i + 1));
        System.out.println(soldados[i].toString()); //PUBLICAMOS RESULTADOS
        System.out.println("-----");
    }
    System.out.println("*****");
}
public void setArmy(ArrayList<ArrayList<Soldado>> army){
    this.army = army;
}
```

```
public void setKingdom(String kingdom){
    this.kingdom = kingdom;
}
public String getKingdom(){
    return kingdom;
}
}
```

Listing 9: Las líneas de códigos de la clase Mapa creada:

```
import java.text.DecimalFormat;
import java.util.*;

public class Mapa {
    Scanner sc = new Scanner(System.in);
    Random rdm = new Random();
    private String territory;
    private ArrayList<ArrayList<Soldado>> board;
    private Ejercito army1e;
    private Ejercito army2e;
    private String[] typesteritory = {"bosque", "campo abierto", "montana", "desierto",
        "playa"};
    private String[] kingdoms = {"Inglaterra", "Francia", "Sacro", "Castilla", "Aragon",
        "Moros"};
    public Mapa(){
        this.board = fillboard();
    }
    public void iniciarJuego() {
        menuBatalla();
        int n = sc.nextInt();
        while (n == 1) {
            String kingdom1 = kingdoms[rdm.nextInt(6)];
            String kingdom2 = kingdoms[rdm.nextInt(6)];
            ArrayList<ArrayList<Soldado>> army1 = fillarray(kingdom1, 1);
            ArrayList<ArrayList<Soldado>> army2 = fillarray(kingdom2, 2);
            army1e = new Ejercito(army1, kingdom1);
            army2e = new Ejercito(army2, kingdom2);
            territory = typesteritory[rdm.nextInt(5)];
            System.out.println("\n*****");
            System.out.println("El tipo de territorio es: " + territory);
            System.out.println("\n*****");
            bonificacion(army1, territory, kingdom1);
            bonificacion(army2, territory, kingdom2);
            army1e.viewSoldiers(kingdom1, 1, army1);
            army2e.viewSoldiers(kingdom2, 2, army2);
            viewBoard(army1, army2);
            army1e.longerLife(army1, kingdom1);
            army2e.longerLife(army2, kingdom2);
            double avg1 = army1e.averageLife(army1, kingdom1);
            double avg2 = army2e.averageLife(army2, kingdom2);
            army1e.rankingInsercionLife(army1, kingdom1);
            army2e.rankingInsercionLife(army2, kingdom2);
            viewBoard(army1, army2);
            resultBattleInfo(army1, kingdom1, 1);
            resultBattleInfo(army2, kingdom2, 2);
            int sum1 = resultBattleSum(army1, kingdom1, 1);
```

```
int sum2 = resultBattleSum(army2, kingdom2, 2);
double sumtotal = (sum1 * 1.0) + (sum2 * 1.0);
resbattle(sumtotal, sum1, sum2, 1, 2, kingdom1, kingdom2);
volveraJugar();
n = sc.nextInt();
}
}
public static ArrayList<ArrayList<Soldado>> fillboard(){
    ArrayList<ArrayList<Soldado>> army = new ArrayList<ArrayList<Soldado>>();
    for(int i = 0; i < 10; i++){ //ITERACION
        army.add(new ArrayList<Soldado>()); //LLENAMOS NUESTROS ARRAYLIST BIDIMENSIONAL
        CON CADA FILA PARA QUE CUMPLAN CON ESTRUCTURA DEL TABLERO
        for(int j = 0; j < 10; j++){//ITERACION
            army.get(i).add(null); // LLENAMOS CADA FILA DEL ARRAYLIST CON UN OBJETO
            SOLDADO CON TAL QUE ESTE SEA NULL PARA QUE SEPA QUE ESTE TIENE UNA
            CASILLA PERO NO HAY NADIE TODAVIA SE PUEDE LLENAR
        }
    }
    return army;
}
public static ArrayList<ArrayList<Soldado>> fillarray(String armyespe, int num){
    Random rdm = new Random();
    ArrayList<ArrayList<Soldado>> army = new ArrayList<ArrayList<Soldado>>();
    int numbersoldiers = rdm.nextInt(10) + 1; //NUMERO DE SOLDADOS ALEATORIOS ENTRE 1
    A 10 SOLDADOS
    for(int i = 0; i < 10; i++){ //ITERACION
        army.add(new ArrayList<Soldado>()); //LLENAMOS NUESTROS ARRAYLIST BIDIMENSIONAL
        CON CADA FILA PARA QUE CUMPLAN CON ESTRUCTURA DEL TABLERO
        for(int j = 0; j < 10; j++){//ITERACION
            army.get(i).add(null); // LLENAMOS CADA FILA DEL ARRAYLIST CON UN OBJETO
            SOLDADO CON TAL QUE ESTE SEA NULL PARA QUE SEPA QUE ESTE TIENE UNA
            CASILLA PERO NO HAY NADIE TODAVIA SE PUEDE LLENAR
        }
    }
    System.out.println("El Ejercito " + armyespe + " tiene " + numbersoldiers + "
        soldados : " );
    System.out.println("");
    for(int i = 0; i < numbersoldiers; i++){ //LLENAMOS CASILLAS CON CADA SOLDADO
        CREADO ALEATORIAMENTE
        Soldado soldado = getRandomSoldado();
        String name = "Soldado" + i + "X" + num;
        //System.out.println(name); PRUEBA QUE SE HIZO PARA VER LOS NOMBRES
        int health = rdm.nextInt(5) + 1;
        int row = rdm.nextInt(10) + 1;
        int speed = rdm.nextInt(5) + 1;
        String column = String.valueOf((char)(rdm.nextInt(10) + 65)); //REUTILIZAMOS
        CODIGO DEL ANTERIOR ARCHIVO VIDEOJUEGO2.JAVA YA QUE TENDRIAN LA MISMA
        FUNCIONALIDAD
        //System.out.println(army.get(row - 1).get((int)column.charAt(0) - 65)); PRUEBA
        QUE SE HIZO PARA COMPROBAR SI EL OBJETO SE ESTABA DANDO O NO CAPAZ NI
        EXISTIA
        int lifelevel, defenselevel = 0;
        int attacklevel = 0;
        if (soldado instanceof Espadachin) {
            name = "Espadachin" + i + "X" + num;
            lifelevel = rdm.nextInt(3) + 8;
```

```
        attacklevel = 10;
        defenselevel = 8;
        soldado.setName(name);
        soldado.setAttackLevel(attacklevel);
        soldado.setDefenseLevel(defenselevel);
        soldado.setLifeLevel(lifelevel);
        soldado.setRow(row);
        soldado.setColumn(column);
        if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
            System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito "
                + armyespe + "");
            army.get(row - 1).set((int)column.charAt(0) - 65, new Espadachin(name,
                attacklevel, defenselevel, lifelevel, speed, "Espadachin", true, row,
                column, attacklevel));
            army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
            System.out.println(army.get(row - 1).get((int)column.charAt(0) -
                65).toString());
            System.out.println("-----");
        } else {
            i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO
                CASILLERO CON TAL QUE NO DEBERIA CONTAR
        }
    } else if (soldado instanceof Arquero) {
        name = "Arquero" + i + "X" + num;
        attacklevel = 7;
        defenselevel = 3;
        lifelevel = rdm.nextInt(3) + 3;
        soldado.setName(name);
        soldado.setAttackLevel(attacklevel);
        soldado.setDefenseLevel(defenselevel);
        soldado.setLifeLevel(lifelevel);
        soldado.setRow(row);
        soldado.setColumn(column);
        if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
            System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito "
                + armyespe + "");
            army.get(row - 1).set((int)column.charAt(0) - 65, new Arquero(name,
                attacklevel, defenselevel, lifelevel, speed, "Arquero", true, row,
                column, attacklevel));
            army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
            System.out.println(army.get(row - 1).get((int)column.charAt(0) -
                65).toString());
            System.out.println("-----");
        } else {
            i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO
                CASILLERO CON TAL QUE NO DEBERIA CONTAR
        }
    } else if (soldado instanceof Caballero) {
        name = "Caballero" + i + "X" + num;
        attacklevel = 13;
        defenselevel = 7;
        lifelevel = rdm.nextInt(3) + 10;
        soldado.setName(name);
        soldado.setAttackLevel(attacklevel);
        soldado.setDefenseLevel(defenselevel);
        soldado.setLifeLevel(lifelevel);
```

```
soldado.setRow(row);
soldado.setColumn(column);
if (army.get(row - 1).get((int)column.charAt(0) - 65) == null) {
    System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito "
        + armyespe + "");
    army.get(row - 1).set((int)column.charAt(0) - 65, new Caballero(name,
        attacklevel, defenselevel, lifelevel, speed, "Caballero", true, row,
        column, false));
    army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
    System.out.println(army.get(row - 1).get((int)column.charAt(0) -
        65).toString());
    System.out.println("-----");
} else {
    i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO
        CASILLERO CON TAL QUE NO DEBERIA CONTAR
}
} else if (soldado instanceof Lancero) {
    name = "Lancero" + i + "X" + num;
    attacklevel = 5;
    defenselevel = 10;
    lifelevel = rdm.nextInt(3) + 5;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
    soldado.setDefenseLevel(defenselevel);
    soldado.setLifeLevel(lifelevel);
    soldado.setRow(row);
    soldado.setColumn(column);
    if (army.get(row - 1).get((int)column.charAt(0) - 65) == null) {
        System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito "
            + armyespe + "");
        army.get(row - 1).set((int)column.charAt(0) - 65, new Lancero(name,
            attacklevel, defenselevel, lifelevel, speed, "Lancero", true, row,
            column, attacklevel));
        army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
        System.out.println(army.get(row - 1).get((int)column.charAt(0) -
            65).toString());
        System.out.println("-----");
    } else {
        i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO
            CASILLERO CON TAL QUE NO DEBERIA CONTAR
    }
}
}
}
System.out.println("*****");
return army;
}

public static void menuBatalla() {
    System.out.println("-----");
    System.out.println("--          MENU          --");
    System.out.println("-----");
    System.out.println(" SELECCIONE UN NUMERO PARA PODER EMPEZAR O TERMINAR");
    System.out.println(" 1 : JUGAR");
    System.out.println(" 2 : NO JUGAR");
}

public static void viewBoard(ArrayList<ArrayList<Soldado>> army1,
    ArrayList<ArrayList<Soldado>> army2) { //EN ESTE METODO DEMOSTRAREMOS LA TABLA
```

Página 15

```
        case 2:
            return new Lancero();
        case 3:
            return new Caballero();
        default:
            return new Espadachin();
    }
}

public static String obtenerInicial(Soldado soldado) {
    if (soldado instanceof Espadachin) {
        return "E";
    } else if (soldado instanceof Arquero) {
        return "A";
    } else if (soldado instanceof Caballero) {
        return "C";
    } else if (soldado instanceof Lancero) {
        return "L";
    } else {
        return "S";
    }
}

public void bonificacion(ArrayList<ArrayList<Soldado>> army, String territory,
    String kingdom) {
    for(int i = 0; i < 10; i++){ //ITERACION
        for(int j = 0; j < 10 ; j++){//ITERACION
            if(army.get(i).get(j) != null){
                if(kingdom.equals("Inglaterra") && territory.equals("bosque")){
                    army.get(i).get(j).setLifeLevel(army.get(i).get(j).getLifeLevel() + 1);
                }else if(kingdom.equals("Francia") && territory.equals("campo abierto")){
                    army.get(i).get(j).setLifeLevel(army.get(i).get(j).getLifeLevel() + 1);
                }else if((kingdom.equals("Castilla") || kingdom.equals("Aragon")) &&
                    territory.equals("montana")){
                    army.get(i).get(j).setLifeLevel(army.get(i).get(j).getLifeLevel() + 1);
                }else if(kingdom.equals("Moros") && territory.equals("desierto")){
                    army.get(i).get(j).setLifeLevel(army.get(i).get(j).getLifeLevel() + 1);
                }else if(kingdom.equals("Sacro") && (territory.equals("desierto") ||
                    territory.equals("playa") || territory.equals("campo abierto"))){
                    army.get(i).get(j).setLifeLevel(army.get(i).get(j).getLifeLevel() + 1);
                }
            }
        }
    }
}

public static void resultBattleInfo(ArrayList<ArrayList<Soldado>> army, String
    kingdom, int n){
    System.out.println("Ejercito " + n + " : " + kingdom);
    int numbersoldiers = 0;
    int numberespadachines = 0;
    int numbercaballeros = 0;
    int numberlanceros = 0;
    int numberarqueros = 0;
    for(int i = 0; i < 10; i++){ //ITERACION
        for(int j = 0; j < 10 ; j++){//ITERACION
            if(army.get(i).get(j) != null){
                numbersoldiers++;
                if(army.get(i).get(j) instanceof Espadachin){
```



```
        numberespadachines++;
    }else if(array.get(i).get(j) instanceof Caballero){
        numbercaballeros++;
    }else if(array.get(i).get(j) instanceof Lancero){
        numberlanceros++;
    }else if(array.get(i).get(j) instanceof Arquero){
        numberarqueros++;
    }
    }
}
}
System.out.println("Cantidad total de soldados creados: " + numbersoldiers);
System.out.println("Espadachines: " + numberespadachines);
System.out.println("Arqueros: " + numberarqueros);
System.out.println("Caballeros: " + numbercaballeros);
System.out.println("Lanceros: " + numberlanceros + "\n");
}
public static int resultBattleSum(ArrayList<ArrayList<Soldado>> army, String
    kingdom, int n){
    int sum = 0;
    for(int i = 0; i < 10; i++){ //ITERACION
        for(int j = 0; j < 10 ; j++){//ITERACION
            if(array.get(i).get(j) != null){
                sum += array.get(i).get(j).getLifeLevel();
            }
        }
    }
    System.out.println("Ejercito " + n + ": " + kingdom + ": " + sum);
    return sum;
}
public static void resbattle(double sumtotal , int sum1, int sum2 , int n1, int n2 ,
    String kingdom1, String kingdom2){
    DecimalFormat df = new DecimalFormat("#.##");
    String numero1 = df.format(((sum1 / sumtotal) * 100));
    String numero2 = df.format(((sum2 / sumtotal) * 100));
    if(sum1 > sum2){
        System.out.println("El ganador es el ejercito " + n1 + " de: " + kingdom1 + ".
        Ya que al generar los porcentajes de probabilidad de victoria basada en los
        niveles de vida de sus soldados y aplicando un experimento aleatorio salio
        vencedor. (Aleatorio generado : "+ numero1 + ")");
    }else if (sum2 > sum1){
        System.out.println("El ganador es el ejercito " + n2 + " de: " + kingdom2 + ".
        Ya que al generar los porcentajes de probabilidad de victoria basada en los
        niveles de vida de sus soldados y aplicando un experimento aleatorio salio
        vencedor. (Aleatorio generado : "+ numero2 + ")");
    }else{
        System.out.println("El resultado de la batalla es Empate");
    }
}
public static void volveraJugar(){
    System.out.println("\n*****");
    System.out.println(" DESEA VOLVER A JUGAR");
    System.out.println(" 1 : JUGAR");
    System.out.println(" 2 : NO JUGAR");
}
}
```

4.3. Ejercicio Mapa

- En esta sección solo mejoramos cuando se imprime el tablero con la información del soldado el cual está en esta cuadrilla.
- El código y el commit sería el siguiente:

Listing 10: Commit

```
$ git commit -m "Mejorando el tablero su información con el ejército donde pertenecen ,  
el tipo de soldado y su nivel de vida"
```

Listing 11: Las líneas de código de la clase Mapa:

```
public static void viewBoard(ArrayList<ArrayList<Soldado>> army1,  
    ArrayList<ArrayList<Soldado>> army2){ //EN ESTE METODO DEMOSTRAREMOS LA TABLA  
    REUTILIZAREMOS CODIGOS DE ANTERIORES LABORATORIOS PARA PODER HACER LA BASE DE ESTE  
    TABLERO  
    System.out.println("\nMostrando tabla de posición ... --");  
    System.out.println("Leyenda: Ejército1 --> 1#- | Ejército2 --> 2#-");  
    //RECONOCIMIENTO PARA LOS EJERCITOS Y POSICIÓN DE SUS SOLDADOS  
    System.out.println("\n \t A\t B\t C\t D\t E\t F\t G\t H\t I\t J"); //  
    RECONOCIMIENTO PARA CADA UBICACIÓN DE CADA SOLDADO EN EL TABLERO POR PARTE DE LAS  
    COLUMNAS  
    System.out.println("\t-----");  
    for(int i = 0; i < 10; i++){  
        System.out.print((i + 1) + "\t"); // RECONOCIMIENTO PARA CADA UBICACIÓN DE CADA  
        SOLDADO EN EL TABLERO POR PARTE DE LAS FILAS  
        for(int j = 0; j < 10; j++){  
            if(army1.get(i).get(j) != null && army2.get(i).get(j) != null){  
                //CREAMOS UN IF PARA QUE ESTE NOS AYUDE A SABER QUIEN DE ESTOS  
                SOLDADOS SE OCUPA DEL CASILLERO EL CUAL DONDE ESTAN PELEANDO  
                if(army1.get(i).get(j).getLifeActual() >  
                    army2.get(i).get(j).getLifeActual()){  
                    army1.get(i).get(j).setLifeActual(army1.get(i).get(j).getLifeActual()  
                        - army2.get(i).get(j).getLifeActual()); //Cambiamos  
                    army2.get(i).set(j, null);  
                    System.out.printf("| 1%s%d ",  
                        obtenerInicial(army1.get(i).get(j)),  
                        army1.get(i).get(j).getLifeActual());  
                }else if(army2.get(i).get(j).getLifeActual() >  
                    army1.get(i).get(j).getLifeActual()){  
                    army2.get(i).get(j).setLifeActual(army2.get(i).get(j).getLifeActual()  
                        - army1.get(i).get(j).getLifeActual());  
                    army1.get(i).set(j, null);  
                    System.out.printf("| 2%s%d ",  
                        obtenerInicial(army2.get(i).get(j)),  
                        army2.get(i).get(j).getLifeActual());  
                }else{  
                    army2.get(i).set(j, null);  
                    army1.get(i).set(j, null);  
                    System.out.print("| " + " " + " " + " ");  
                }  
            }else if(army1.get(i).get(j) != null){  
                System.out.printf("| 1%s%d ", obtenerInicial(army1.get(i).get(j)),  
                    army1.get(i).get(j).getLifeActual());  
            }else if(army2.get(i).get(j) != null){  
                System.out.printf("| 2%s%d ", obtenerInicial(army2.get(i).get(j)),  
                    army2.get(i).get(j).getLifeActual());  
            }else{  
                System.out.print("| " + " " + " " + " ");  
            }  
        }  
        System.out.println();  
    }  
}
```

Listing 12: Ejecucion:

Página 19

```
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 1
Actitud: Lancero
Estado: true
```

Registrando al 4 soldado del Ejercito Inglaterra

```
Nombre: Caballero3X1
Vida: 11
Fila: 10
Columna: J
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 3
Actitud: Caballero
Estado: true
```

Registrando al 5 soldado del Ejercito Inglaterra

```
Nombre: Espadachin4X1
Vida: 10
Fila: 7
Columna: D
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 1
Actitud: Espadachin
Estado: true
```

Registrando al 6 soldado del Ejercito Inglaterra

```
Nombre: Espadachin5X1
Vida: 8
Fila: 6
Columna: J
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 8
Velocidad: 4
Actitud: Espadachin
Estado: true
```

Registrando al 7 soldado del Ejercito Inglaterra

```
Nombre: Lancero6X1
Vida: 7
Fila: 8
Columna: F
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 3
```

```
Actitud: Lancero
Estado: true
-----
Registrando al 8 soldado del Ejercito Inglaterra
```

```
Nombre: Espadachin7X1
Vida: 8
Fila: 10
Columna: E
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 8
Velocidad: 2
Actitud: Espadachin
Estado: true
-----
```

```
Registrando al 9 soldado del Ejercito Inglaterra
```

```
Nombre: Lancero8X1
Vida: 6
Fila: 1
Columna: D
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 6
Velocidad: 2
Actitud: Lancero
Estado: true
-----
```

```
*****
El Ejercito Sacro tiene 3 soldados :
```

```
Registrando al 1 soldado del Ejercito Sacro
```

```
Nombre: Espadachin0X2
Vida: 10
Fila: 4
Columna: I
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 4
Actitud: Espadachin
Estado: true
-----
```

```
Registrando al 2 soldado del Ejercito Sacro
```

```
Nombre: Lancero1X2
Vida: 5
Fila: 3
Columna: G
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 5
Velocidad: 1
Actitud: Lancero
```

```
Estado: true
-----
Registrando al 3 soldado del Ejercito Sacro

Nombre: Lancero2X2
Vida: 7
Fila: 7
Columna: B
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 3
Actitud: Lancero
Estado: true
-----
*****

*****
El tipo de territorio es: campo abierto

*****
El Ejercito Inglaterra del 1 ejercito sus soldados son :

*****
El 1 soldado es:

Nombre: Lancero8X1
Vida: 6
Fila: 1
Columna: D
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 6
Velocidad: 2
Actitud: Lancero
Estado: true

*****
El 2 soldado es:

Nombre: Espadachin0X1
Vida: 10
Fila: 2
Columna: E
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 1
Actitud: Espadachin
Estado: true

*****
El 3 soldado es:

Nombre: Lancero2X1
Vida: 7
```

```
Fila: 2
Columna: G
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 1
Actitud: Lancero
Estado: true

*****
El 4 soldado es:

Nombre: Espadachin5X1
Vida: 8
Fila: 6
Columna: J
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 8
Velocidad: 4
Actitud: Espadachin
Estado: true

*****
El 5 soldado es:

Nombre: Espadachin4X1
Vida: 10
Fila: 7
Columna: D
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 1
Actitud: Espadachin
Estado: true

*****
El 6 soldado es:

Nombre: Lancero6X1
Vida: 7
Fila: 8
Columna: F
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 3
Actitud: Lancero
Estado: true

*****
El 7 soldado es:

Nombre: Caballero1X1
Vida: 12
```

```
Fila: 9
Columna: E
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 5
Actitud: Caballero
Estado: true

*****
El 8 soldado es:

Nombre: Espadachin7X1
Vida: 8
Fila: 10
Columna: E
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 8
Velocidad: 2
Actitud: Espadachin
Estado: true

*****
El 9 soldado es:

Nombre: Caballero3X1
Vida: 11
Fila: 10
Columna: J
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 3
Actitud: Caballero
Estado: true
El Ejercito Sacro del 2 ejercito sus soldados son :

*****
El 1 soldado es:

Nombre: Lancero1X2
Vida: 5
Fila: 3
Columna: G
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 6
Velocidad: 1
Actitud: Lancero
Estado: true

*****
El 2 soldado es:

Nombre: Espadachin0X2
```



```
Vida: 10
Fila: 4
Columna: I
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 11
Velocidad: 4
Actitud: Espadachin
Estado: true
```

```
*****
El 3 soldado es:
```

```
Nombre: Lancero2X2
Vida: 7
Fila: 7
Columna: B
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 8
Velocidad: 3
Actitud: Lancero
Estado: true
```

Mostrando tabla de posicion ...

Legenda: Ejercito1 --> 1#- | Ejercito2 --> 2#-

Tpx	A	B	C	D	E	F	G	H	I	J
1	Arquero.java			1L 6						
2	Caballero.java				1E10					
3	Ejercito.java									
4	Espadachin.java								2E10	
5	Juegoprincipal.java									
6	Lancero.java									1E 8
7	Lancero2X2.java	2L 7		1E10						
8	Mapa.java									
9	Soldado.java				1C12					
10					1E 8					1C11

\end{lstlisting}

4.4. Ejercicio Soldado(Herencia)

- En esta seccion publicamos los otros tipos de soldados especiales por cada reino los cuales son CaballeroFranco, CaballeroMoro , EspadachinReal , EspadachinConquistador, EspadachinTeutonico.

- El código y el commit sería el siguiente:

Listing 13: Commit

```
$ git commit -m "Agregando la clase EspadachinReal el cual es una unidad especial para  
el reino de Inglaterra creamos sus metodos y sus atributos necesarios"  
$ git commit -m "Agregando la clase CaballeroFranco la cual tiene sus metodos , getters  
y setters, y sus debidos atributos"  
$ git commit -m "Agregando la clase EspadachinConquistador la cual tiene sus  
metodos, getters y setters, y sus debidos atributos"  
$ git commit -m "Agregando la clase EspadachinTeutonico la cual tiene sus metodos como  
el modoTortuga() y otros, getters y setters, y sus debidos atributos"  
$ git commit -m "Agregando la clase CaballeroMoro la cual tiene sus metodos, getters y  
setters, y sus debidos atributos"
```

Listing 14: Las líneas de código de la clase CaballeroFranco:

```
public class CaballeroFranco extends Soldado{  
    private int numspears;  
    private int longspear;  
    private int evolve = 0;  
    public CaballeroFranco(){  
    }  
    public CaballeroFranco(String name , int attacklevel, int defenselevel, int  
        lifelevel, int speed, String attitude ,boolean lives, int row, String column,int  
        numspears, int longspear){  
        super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,  
            column);  
        this.numspears = numspears;  
        this.longspear = longspear;  
    }  
    public void throwSpears(){  
        if(numspears == 0){  
            System.out.println("Ya no quedan lanzas");  
        }else{  
            numspears--;  
        }  
    }  
    public void evolveSoldier(){  
        if(evolve < 4){  
            evolve++;  
            numspears += evolve;  
            longspear += evolve;  
            System.out.println("Caballero Franco evoluciono a nivel " + evolve);  
        }else{  
            System.out.println("Caballero Franco ya esta en su nivel maximo de evolucion.");  
        }  
    }  
    public int getnumSpears(){  
        return numspears;  
    }  
    public int getlongSpears(){  
        return longspear;  
    }  
    public void setnumSpears(int n){  
        this.numspears = n;  
    }  
}
```

```
}  
}
```

Listing 15: Las lineas de codigos de la clase CaballeroMoro:

```
public class CaballeroMoro extends Soldado{  
    private int flechas;  
    private int longflechas;  
    private int evolve = 0;  
    public CaballeroMoro(){  
    }  
    public CaballeroMoro(String name , int attacklevel, int defenselevel, int lifelevel,  
        int speed, String attitude ,boolean lives, int row, String column, int flechas,  
        int longflechas){  
        super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,  
            column);  
        this.flechas = flechas;  
        this.longflechas = longflechas;  
    }  
    public void disparar(){  
        if(this.flechas == 0){  
            System.out.println("El arquero ya no tiene flechas para poder disparar");  
        }else{  
            this.flechas = flechas - 1;  
            this.atacar();  
        }  
    }  
    public void evolveSoldier(){  
        if(evolve < 4){  
            evolve++;  
            flechas += evolve;  
            longflechas += evolve;  
            System.out.println("Espadachin Conquistador evoluciono a nivel " + evolve);  
        }else{  
            System.out.println("Espadachin Conquistador ya esta en su nivel maximo de  
                evolucion.");  
        }  
    }  
    public void setFlechas(int n){  
        this.flechas = n;  
    }  
    public int getFlechas(){  
        return flechas;  
    }  
    public int getLongFlechas(){  
        return longflechas;  
    }  
}
```

Listing 16: Las lineas de codigos de la clase EspadachinReal:

```
public class EspadachinReal extends Soldado{  
    private int numknives;  
    private int longknife;  
    private int evolve = 0;
```

```
public EspadachinReal(){
}
public EspadachinReal(String name , int attacklevel, int defenselevel, int
    lifelevel, int speed, String attitude ,boolean lives, int row, String column,
    int numknives, int longknife){
    super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,
        column);
    this.numknives = numknives;
    this.longknife = longknife;
}
public int getnumKnives(){
    return numknives;
}
public int getlongKnives(){
    return longknife;
}
public void throwKnives(){
    if(numknives == 0){
        System.out.println("Ya no quedan cuchillos");
    }else{
        numknives--;
    }
}
public void evolveSoldier(){
    if(evolve < 4){
        evolve++;
        numknives += evolve;
        longknife += evolve;
        System.out.println("Espadachin Real evoluciona a nivel " + evolve);
    }else{
        System.out.println("Espadachin Real ya esta en su nivel maximo de evolucion.");
    }
}
public void setnumKnives(int n){
    this.numknives = n;
}
}
```

Listing 17: Las lineas de codigos de la clase EspadachinConquistador:

```
public class EspadachinConquistador extends Soldado{
    private int numaxes;
    private int longaxes;
    private int evolve;
    public EspadachinConquistador(){
    }
    public EspadachinConquistador(String name , int attacklevel, int defenselevel, int
        lifelevel, int speed, String attitude ,boolean lives, int row, String column,
        int numaxes, int longaxes){
        super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,
            column);
        this.numaxes = numaxes;
        this.longaxes = longaxes;
    }
    public void throwAxes(){
        if(numaxes == 0){
```

```
        System.out.println("No es posible que el espadachin lance hachas");
    } else {
        numaxes--;
    }
}

public void evolveSoldier(){
    if(evolve < 4){
        evolve++;
        numaxes += evolve;
        longaxes += evolve;
        System.out.println("Espadachin Conquistador evoluciona a nivel " + evolve);
    }else{
        System.out.println("Espadachin Conquistador ya esta en su nivel maximo de
            evolucion.");
    }
}

public void setnumAxes(int n){
    this.numaxes = n;
}

public int getnumAxes(){
    return numaxes;
}

public int getlongAxes(){
    return longaxes;
}
}
```

Listing 18: Las líneas de códigos de la clase EspadachinTeutonico:

```
public class EspadachinTeutonico extends Soldado{
    private int numjavelin;
    private int longjavelin;
    private int evolve;
    public EspadachinTeutonico(){
    }

    public EspadachinTeutonico(String name , int attacklevel, int defenselevel, int
        lifelevel, int speed, String attitude ,boolean lives, int row, String column,
        int numjavelin, int longjavelin){
        super(name, attacklevel, defenselevel, lifelevel, speed, attitude, lives, row,
            column);
        this.numjavelin = numjavelin;
        this.longjavelin = longjavelin;
    }

    public void throwJavelin(){
        if(numjavelin == 0){
            System.out.println("No es posible que el espadachin lance jabalinas");
        } else {
            numjavelin--;
        }
    }

    public void modoTortuga(){
        System.out.println("Usted uso la habilidad modo Tortuga, Defensa Especial");
    }

    public void evolveSoldier(){
        if(evolve < 4){
            evolve++;
        }
    }
}
```

```
        numjavelin += evolve;
        longjavelin += evolve;
        System.out.println("Espadachin Teutonico evoluciona a nivel " + evolve);
    }else{
        System.out.println("Espadachin Teutonico ya esta en su nivel maximo de
            evolucion.");
    }
}
}
public void setnumJavelin(int n){
    this.numjavelin = n;
}
public int getnumJavelin(){
    return numjavelin;
}
public int getlongJavelin(){
    return longjavelin;
}
}
```

4.5. Ejercicio Mapa

- En esta seccion publicamos los otros tipos de soldados especiales por cada reino los cuales son CaballeroFranco, CaballeroMoro , EspadachinReal , EspadachinConquistador, EspadachinTeutonico.
- El codigo y el commit seria el siguiente:

Listing 19: Commit

```
$ git commit -m "Agregando opciones en getRandomSoldado() el cual serian estas clases
heredadas anteriormente respetando si estas pertenecen a cierto reino especifico
tambien modificamos el metodo obtenerIncial() el cual tambien agregamos las
iniciales de cada tipo de Soldado y tambien en el metodo fillarray() el cual creara
a cada soldado dependiendo el tipo que sea con un constructor seria solo agregar
mas clases heredadas en resumen "
```

Listing 20: Las lineas de codigos de la clase Mapa

```
public static Soldado getRandomSoldado(String armyespe) {
    Random rdm = new Random();
    int tipoSoldado = rdm.nextInt(5);
    switch (tipoSoldado) {
        case 0:
            return new Espadachin();
        case 1:
            return new Arquero();
        case 2:
            return new Lancero();
        case 3:
            return new Caballero();
        case 4:
            if(armyespe == "Inglaterra"){
                return new EspadachinReal();
            }else if(armyespe == "Francia"){
```

```
        return new CaballeroFranco();
    }else if(armyespe == "Sacro"){
        return new EspadachinTeutonico();
    }else if(armyespe == "Aragon" || armyespe == "Castilla"){
        return new EspadachinConquistador();
    }else if(armyespe == "Moros"){
        return new CaballeroMoro();
    }
    default:
        return new Espadachin();
    }
}

public static String obtenerInicial(Soldado soldado) {
    if (soldado instanceof Espadachin) {
        return "E";
    } else if (soldado instanceof Arquero) {
        return "A";
    } else if (soldado instanceof Caballero) {
        return "C";
    } else if (soldado instanceof Lancero) {
        return "L";
    } else if (soldado instanceof EspadachinReal){
        return "ER";
    } else if (soldado instanceof CaballeroFranco){
        return "CF";
    } else if (soldado instanceof EspadachinTeutonico){
        return "ET";
    } else if (soldado instanceof EspadachinConquistador){
        return "EC";
    } else if (soldado instanceof CaballeroMoro){
        return "CM";
    }else{
        return " ";
    }
}

public static ArrayList<ArrayList<Soldado>> fillarray(String armyespe, int num){
    Random rdm = new Random();
    ArrayList<ArrayList<Soldado>> army = new ArrayList<ArrayList<Soldado>>();
    int numbersoldiers = rdm.nextInt(10) + 1; //NUMERO DE SOLDADOS ALEATORIOS ENTRE 1 A
    10 SOLDADOS
    for(int i = 0; i < 10; i++){ //ITERACION
        army.add(new ArrayList<Soldado>()); //LLENAMOS NUESTROS ARRAYLIST BIDIMENSIONAL
        CON CADA FILA PARA QUE CUMPLAN CON ESTRUCTURA DEL TABLERO
        for(int j = 0; j < 10 ; j++){//ITERACION
            army.get(i).add(null); // LLENAMOS CADA FILA DEL ARRAYLIST CON UN OBJETO
            SOLDADO CON TAL QUE ESTE SEA NULL PARA QUE SEPA QUE ESTE TIENE UNA CASILLA
            PERO NO HAY NADIE TODAVIA SE PUEDE LLENAR
        }
    }
    System.out.println("El Ejercito " + armyespe + " tiene " + numbersoldiers + "
        soldados : " );
    System.out.println("");
    for(int i = 0; i < numbersoldiers; i++){ //LLENAMOS CASILLAS CON CADA SOLDADO CREADO
        ALEATORIAMENTE
        Soldado soldado = getRandomSoldado(armyespe);
        String name = "Soldado" + i + "X" + num;
```

```
//System.out.println(name); PRUEBA QUE SE HIZO PARA VER LOS NOMBRES
int health = rdm.nextInt(5) + 1;
int row = rdm.nextInt(10) + 1;
int speed = rdm.nextInt(5) + 1;
String column = String.valueOf((char)(rdm.nextInt(10) + 65)); //REUTILIZAMOS
CODIGO DEL ANTERIOR ARCHIVO VIDEOJUEGO2.JAVA YA QUE TENDRIAN LA MISMA
FUNCIONALIDAD
//System.out.println(array.get(row - 1).get((int)column.charAt(0) - 65)); PRUEBA
QUE SE HIZO PARA COMPROBAR SI EL OBJETO SE ESTABA DANDO O NO CAPAZ NI EXISTIA
int lifelevel, defenselevel = 0;
int attacklevel = 0;
if (soldado instanceof Espadachin) {
    name = "Espadachin" + i + "X" + num;
    lifelevel = rdm.nextInt(3) + 8;
    attacklevel = 10;
    defenselevel = 8;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
    soldado.setDefenseLevel(defenselevel);
    soldado.setLifeLevel(lifelevel);
    soldado.setRow(row);
    soldado.setColumn(column);
    if(array.get(row - 1).get((int)column.charAt(0) - 65) == null){
        System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
            armyespe + "");
        army.get(row - 1).set((int)column.charAt(0) - 65, new Espadachin(name,
            attacklevel, defenselevel, lifelevel, speed, "Espadachin", true, row,
            column, attacklevel));
        army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
        System.out.println(array.get(row - 1).get((int)column.charAt(0) -
            65).toString());
        System.out.println("-----");
    }else{
        i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
        CON TAL QUE NO DEBERIA CONTAR
    }
} else if (soldado instanceof Arquero) {
    name = "Arquero" + i + "X" + num;
    attacklevel = 7;
    defenselevel = 3;
    lifelevel = rdm.nextInt(3) + 3;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
    soldado.setDefenseLevel(defenselevel);
    soldado.setLifeLevel(lifelevel);
    soldado.setRow(row);
    soldado.setColumn(column);
    if(array.get(row - 1).get((int)column.charAt(0) - 65) == null){
        System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
            armyespe + "");
        army.get(row - 1).set((int)column.charAt(0) - 65, new Arquero(name,
            attacklevel, defenselevel, lifelevel, speed, "Arquero", true, row,
            column, attacklevel));
        army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
        System.out.println(array.get(row - 1).get((int)column.charAt(0) -
            65).toString());
    }
```



```
        System.out.println("-----");
    }else{
        i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
                CON TAL QUE NO DEBERIA CONTAR
    }
} else if (soldado instanceof Caballero) {
    name = "Caballero" + i + "X" + num;
    attacklevel = 13;
    defenselevel = 7;
    lifelevel = rdm.nextInt(3) + 10;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
    soldado.setDefenseLevel(defenselevel);
    soldado.setLifeLevel(lifelevel);
    soldado.setRow(row);
    soldado.setColumn(column);
    if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
        System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
            armyespe + "");
        army.get(row - 1).set((int)column.charAt(0) - 65, new Caballero(name,
            attacklevel, defenselevel, lifelevel, speed, "Caballero", true, row,
            column, false));
        army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
        System.out.println(army.get(row - 1).get((int)column.charAt(0) -
            65).toString());
        System.out.println("-----");
    }else{
        i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
                CON TAL QUE NO DEBERIA CONTAR
    }
} else if (soldado instanceof Lancero) {
    name = "Lancero" + i + "X" + num;
    attacklevel = 5;
    defenselevel = 10;
    lifelevel = rdm.nextInt(3) + 5;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
    soldado.setDefenseLevel(defenselevel);
    soldado.setLifeLevel(lifelevel);
    soldado.setRow(row);
    soldado.setColumn(column);
    if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
        System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
            armyespe + "");
        army.get(row - 1).set((int)column.charAt(0) - 65, new Lancero(name,
            attacklevel, defenselevel, lifelevel, speed, "Lancero", true, row,
            column, attacklevel));
        army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
        System.out.println(army.get(row - 1).get((int)column.charAt(0) -
            65).toString());
        System.out.println("-----");
    }else{
        i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
                CON TAL QUE NO DEBERIA CONTAR
    }
} else if (soldado instanceof EspadachinReal){
```

```
name = "Espadachin Real" + i + "X" + num;
attacklevel = 10;
defenselevel = 8;
lifelevel = 12;
soldado.setName(name);
soldado.setAttackLevel(attacklevel);
soldado.setDefenseLevel(defenselevel);
soldado.setLifeLevel(lifelevel);
soldado.setRow(row);
soldado.setColumn(column);
if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
    System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
        armyespe + "");
    army.get(row - 1).set((int)column.charAt(0) - 65, new EspadachinReal(name,
        attacklevel, defenselevel, lifelevel, speed, "Espadachin Real", true,
        row, column, attacklevel, attacklevel));
    army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
    System.out.println(army.get(row - 1).get((int)column.charAt(0) -
        65).toString());
    System.out.println("-----");
}else{
    i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
        CON TAL QUE NO DEBERIA CONTAR
}
} else if (soldado instanceof CaballeroFranco){
    name = "Caballero Franco" + i + "X" + num;
    attacklevel = 13;
    defenselevel = 7;
    lifelevel = 15;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
    soldado.setDefenseLevel(defenselevel);
    soldado.setLifeLevel(lifelevel);
    soldado.setRow(row);
    soldado.setColumn(column);
    if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
        System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
            armyespe + "");
        army.get(row - 1).set((int)column.charAt(0) - 65, new CaballeroFranco(name,
            attacklevel, defenselevel, lifelevel, speed, "Caballero Franco", true,
            row, column, attacklevel, attacklevel));
        army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
        System.out.println(army.get(row - 1).get((int)column.charAt(0) -
            65).toString());
        System.out.println("-----");
    }else{
        i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
            CON TAL QUE NO DEBERIA CONTAR
    }
} else if (soldado instanceof EspadachinTeutonico){
    name = "Espadachin Teutonico" + i + "X" + num;
    attacklevel = 10;
    defenselevel = 8;
    lifelevel = 13;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
```

```
soldado.setDefenseLevel(defenselevel);
soldado.setLifeLevel(lifelevel);
soldado.setRow(row);
soldado.setColumn(column);
if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
    System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
        armyespe + "");
    army.get(row - 1).set((int)column.charAt(0) - 65, new
        EspadachinTeutonico(name, attacklevel, defenselevel, lifelevel, speed,
            "Espadachin Teutonico", true, row, column, attacklevel, attacklevel));
    army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
    System.out.println(army.get(row - 1).get((int)column.charAt(0) -
        65).toString());
    System.out.println("-----");
}else{
    i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
        CON TAL QUE NO DEBERIA CONTAR
}
} else if (soldado instanceof EspadachinConquistador){
    name = "Espadachin Conquistador" + i + "X" + num;
    attacklevel = 10;
    defenselevel = 8;
    lifelevel = 14;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
    soldado.setDefenseLevel(defenselevel);
    soldado.setLifeLevel(lifelevel);
    soldado.setRow(row);
    soldado.setColumn(column);
    if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
        System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
            armyespe + "");
        army.get(row - 1).set((int)column.charAt(0) - 65, new
            EspadachinConquistador(name, attacklevel, defenselevel, lifelevel,
                speed, "Espadachin Conquistador", true, row, column, attacklevel,
                    attacklevel));
        army.get(row - 1).get((int)column.charAt(0) - 65).setSpeed(speed);
        System.out.println(army.get(row - 1).get((int)column.charAt(0) -
            65).toString());
        System.out.println("-----");
    }else{
        i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
            CON TAL QUE NO DEBERIA CONTAR
    }
} else if (soldado instanceof CaballeroMoro){
    name = "Caballero Moro" + i + "X" + num;
    attacklevel = 13;
    defenselevel = 7;
    lifelevel = 13;
    soldado.setName(name);
    soldado.setAttackLevel(attacklevel);
    soldado.setDefenseLevel(defenselevel);
    soldado.setLifeLevel(lifelevel);
    soldado.setRow(row);
    soldado.setColumn(column);
    if (army.get(row - 1).get((int)column.charAt(0) - 65) == null){
```

```
        System.out.println("Registrando al " + (i + 1) + " soldado del Ejercito " +
            armyespe + "");
        army.get(row - 1).set(((int)column.charAt(0) - 65, new CaballeroMoro(name,
            attacklevel, defenselevel, lifelevel, speed, "Caballero Moro", true,
            row, column, attacklevel, attacklevel));
        army.get(row - 1).get(((int)column.charAt(0) - 65).setSpeed(speed);
        System.out.println(army.get(row - 1).get(((int)column.charAt(0) -
            65).toString());
        System.out.println("-----");
    }else{
        i -= 1; //NOS AYUDARIA CON LOS SOLDADOS QUE SE REPITEN EN EL MISMO CASILLERO
            CON TAL QUE NO DEBERIA CONTAR
    }
}
}
System.out.println("*****");
return army;
}
```

Listing 21: Ejecucion:

```
-----
--          MENU          --
-----

SELECCIONE UN NUMERO PARA PODER EMPEZAR O TERMINAR
1 : JUGAR
2 : NO JUGAR
1
El Ejercito Francia tiene 3 soldados :

Registrando al 1 soldado del Ejercito Francia

Nombre: Espadachin0X1
Vida: 8
Fila: 6
Columna: H
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 8
Velocidad: 2
Actitud: Espadachin
Estado: true
-----

Registrando al 2 soldado del Ejercito Francia

Nombre: Espadachin1X1
Vida: 10
Fila: 2
Columna: A
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 5
Actitud: Espadachin
Estado: true
-----
```

Registrando al 3 soldado del Ejercito Francia

Nombre: Arquero2X1
Vida: 3
Fila: 4
Columna: F
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 3
Velocidad: 2
Actitud: Arquero
Estado: true

El Ejercito Aragon tiene 3 soldados :

Registrando al 1 soldado del Ejercito Aragon

Nombre: Espadachin0X2
Vida: 10
Fila: 7
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 5
Actitud: Espadachin
Estado: true

Registrando al 2 soldado del Ejercito Aragon

Nombre: Arquero1X2
Vida: 3
Fila: 5
Columna: C
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 3
Velocidad: 2
Actitud: Arquero
Estado: true

Registrando al 3 soldado del Ejercito Aragon

Nombre: Lancero2X2
Vida: 6
Fila: 8
Columna: E
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 6
Velocidad: 5
Actitud: Lancero
Estado: true

```
*****
El tipo de territorio es: campo abierto

*****
El Ejercito Francia del 1 ejercito sus soldados son :

*****
El 1 soldado es:

Nombre: Espadachin1X1
Vida: 10
Fila: 2
Columna: A
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 11
Velocidad: 5
Actitud: Espadachin
Estado: true

*****
El 2 soldado es:

Nombre: Arquero2X1
Vida: 3
Fila: 4
Columna: F
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 4
Velocidad: 2
Actitud: Arquero
Estado: true

*****
El 3 soldado es:

Nombre: Espadachin0X1
Vida: 8
Fila: 6
Columna: H
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 9
Velocidad: 2
Actitud: Espadachin
Estado: true
El Ejercito Aragon del 2 ejercito sus soldados son :

*****
El 1 soldado es:

Nombre: Arquero1X2
Vida: 3
Fila: 5
```

```
Columna: C
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 3
Velocidad: 2
Actitud: Arquero
Estado: true

*****
El 2 soldado es:

Nombre: Espadachin0X2
Vida: 10
Fila: 7
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 5
Actitud: Espadachin
Estado: true

*****
El 3 soldado es:

Nombre: Lancero2X2
Vida: 6
Fila: 8
Columna: E
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 6
Velocidad: 5
Actitud: Lancero
Estado: true

*****
El tipo de territorio es: campo abierto

*****
```

Mostrando tabla de posicion ... --
Leyenda: Ejercito1 --> 1#- | Ejercito2 --> 2#-

	A	B	C	D	E	F	G	H	I	J
1	Arguero.java					El tipo de territorio es: campo abierto				
2	1 E10									
3	CaballeroFronco.class					El Ejército Francia del 1 ejército sus soldados				
4	CaballeroFronco.java					1 A 3				
5	CaballeroFronco.class					El 1 soldado es:				
6	Ejercito1.java		2 A 3			Nombre: Espadachin1X1				
7	Espadachin.class					Vida: 10	1 E 8			
8	Espadachin.java					Fila: 2	2 E10			
9	EspadachinConquistador.class				2 L 6	Columna: A				
10	EspadachinConquistador.java					Nivel de ataque: 10				
	EspadachinReal.class					Nivel de Defensa: 8				
	EspadachinReal.java					Nivel de vida: 11				
	EspadachinTeutonico.class					Velocidad: 5				
	EspadachinTeutonico.java					Actitud: Espadachin				
						Estado: true				

Listing 22: Ejecucion:

```
*****
El soldado con mayor vida del Ejercito Francia es:

Nombre: Espadachin1X1
Vida: 10
Fila: 2
Columna: A
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 11
Velocidad: 5
Actitud: Espadachin
Estado: true
*****
El soldado con mayor vida del Ejercito Aragon es:

Nombre: Espadachin0X2
Vida: 10
Fila: 7
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 5
Actitud: Espadachin
Estado: true
*****
El promedio de puntos de vida del Ejercito Francia es:
7.0
```



```
*****
El promedio de puntos de vida del Ejercito Aragon es:
6.333333333333333
*****

El Ejercito Francia ordenando por metodo insercion:
-----
Mostrando Ranking del Ejercito Francia ..... //---->

Puesto 1
Nombre: Espadachin1X1
Vida: 10
Fila: 2
Columna: A
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 11
Velocidad: 5
Actitud: Espadachin
Estado: true
-----

Puesto 2
Nombre: Espadachin0X1
Vida: 8
Fila: 6
Columna: H
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 9
Velocidad: 2
Actitud: Espadachin
Estado: true
-----

Puesto 3
Nombre: Arquero2X1
Vida: 3
Fila: 4
Columna: F
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 4
Velocidad: 2
Actitud: Arquero
Estado: true
-----
*****

El Ejercito Aragon ordenando por metodo insercion:
-----
Mostrando Ranking del Ejercito Aragon ..... //---->

Puesto 1
Nombre: Espadachin0X2
Vida: 10
```

```
Fila: 7
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 5
Actitud: Espadachin
Estado: true
-----
```

```
Puesto 2
Nombre: Lancero2X2
Vida: 6
Fila: 8
Columna: E
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 6
Velocidad: 5
Actitud: Lancero
Estado: true
-----
```

```
Puesto 3
Nombre: Arquero1X2
Vida: 3
Fila: 5
Columna: C
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 3
Velocidad: 2
Actitud: Arquero
Estado: true
-----
```

El tipo de territorio es: campo abierto

Listing 23: Ejecucion:

Página 43

4.6. Ejercicio Mapa

- En esta sección mi función `resultBattleInfo`, proporcione información detallada sobre la composición del ejército representado por la matriz de objetos `ArrayList` bidimensional `army`. La función recorre la matriz y cuenta la cantidad de soldados totales, así como la distribución por tipo, que incluye Espadachines, Arqueros, Caballeros, Lanceros, Espadachines Reales, Caballeros Francos, Caballeros Moros, Espadachines Teutónicos y Espadachines Conquistadores. La información se presenta de manera organizada, especificando la cantidad de cada tipo de soldado en el ejército del reino correspondiente (identificado por el parámetro `kingdom`). Los resultados se imprimen en la consola, proporcionando una visión general clara de la fuerza militar de cada reino en términos de composición de tropas.
- El código y el commit sería el siguiente:

Listing 24: Commit

```
$ git commit -m "Agregando opciones en getRandomSoldado() el cual serian estas clases heredadas anteriormente respetando si estas pertenecen a cierto reino especifico tambien modificamos el metodo obtenerIncial() el cual tambien agregamos las iniciales de cada tipo de Soldado y tambien en el metodo fillarray() el cual creara a cada soldado dependiendo el tipo que sea con un constructor seria solo agregar mas clases heredadas en resumen "
```

Listing 25: Las líneas de código de la clase Mapa

```
public static void resultBattleInfo(ArrayList<ArrayList<Soldado>> army, String kingdom,
int n){
    System.out.println("Ejercito " + n + " : " + kingdom);
    int numbersoldiers = 0;
    int numberespadachines = 0;
    int numbercaballeros = 0;
    int numberlanceros = 0;
    int numberarqueros = 0;
    int numberespadachinesreales = 0;
    int numbercaballerosfrancos = 0;
    int numbercaballerosmoros = 0;
    int numberespadachinesteutonicos = 0;
    int numberespadachinesconquistadores = 0;
    for(int i = 0; i < 10; i++){ //ITERACION
        for(int j = 0; j < 10 ; j++){//ITERACION
            if(army.get(i).get(j) != null){
                numbersoldiers++;
                if(army.get(i).get(j) instanceof Espadachin){
                    numberespadachines++;
                }else if(army.get(i).get(j) instanceof Caballero){
                    numbercaballeros++;
                }else if(army.get(i).get(j) instanceof Lancero){
                    numberlanceros++;
                }else if(army.get(i).get(j) instanceof Arquero){
                    numberarqueros++;
                }else if(army.get(i).get(j) instanceof EspadachinReal){
                    numberespadachinesreales++;
                }else if(army.get(i).get(j) instanceof CaballeroFranco){
                    numbercaballerosfrancos++;
                }else if(army.get(i).get(j) instanceof CaballeroMoro){
                    numbercaballerosmoros++;
                }
            }
        }
    }
}
```

```
        }else if (army.get(i).get(j) instanceof EspadachinTeutonico){
            numberespadachinesteutonicos++;
        }else if (army.get(i).get(j) instanceof EspadachinConquistador){
            numberespadachinesconquistadores++;
        }
    }
}

System.out.println("Cantidad total de soldados creados: " + numbersoldiers);
System.out.println("Espadachines: " + numberespadachines);
System.out.println("Arqueros: " + numberarqueros);
System.out.println("Caballeros: " + numbercaballeros);
System.out.println("Lanceros: " + numberlanceros);
System.out.println("Espadachin Real: " + numberespadachinesreales);
System.out.println("Caballero Moro: " + numbercaballerosmoros);
System.out.println("Caballero Franco: " + numbercaballerosfrancos);
System.out.println("Espadachin Teutonico: " + numberespadachinesteutonicos);
System.out.println("Espadachin Conquistador: " + numberespadachinesconquistadores +
    "\n");
}
```

Listing 26: Ejecucion:

```
-----
--          MENU          --
-----

SELECCIONE UN NUMERO PARA PODER EMPEZAR O TERMINAR
1 : JUGAR
2 : NO JUGAR
1
El Ejercito Castilla tiene 6 soldados :

Registrando al 1 soldado del Ejercito Castilla

Nombre: Lancero0X1
Vida: 6
Fila: 6
Columna: G
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 6
Velocidad: 1
Actitud: Lancero
Estado: true
-----

Registrando al 2 soldado del Ejercito Castilla

Nombre: Lancero1X1
Vida: 6
Fila: 9
Columna: D
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 6
Velocidad: 5
```

```
Actitud: Lancero
Estado: true
-----
Registrando al 3 soldado del Ejercito Castilla
```

```
Nombre: Espadachin Conquistador2X1
Vida: 14
Fila: 9
Columna: J
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true
-----
```

```
Registrando al 4 soldado del Ejercito Castilla
```

```
Nombre: Espadachin Conquistador3X1
Vida: 14
Fila: 2
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true
-----
```

```
Registrando al 5 soldado del Ejercito Castilla
```

```
Nombre: Caballero4X1
Vida: 10
Fila: 10
Columna: F
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 10
Velocidad: 2
Actitud: Caballero
Estado: true
-----
```

```
Registrando al 6 soldado del Ejercito Castilla
```

```
Nombre: Espadachin Conquistador5X1
Vida: 14
Fila: 9
Columna: F
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 1
Actitud: Espadachin Conquistador
Estado: true
-----
```

```
*****
```

El Ejercito Aragon tiene 8 soldados :

Registrando al 1 soldado del Ejercito Aragon

Nombre: Espadachin0X2
Vida: 8
Fila: 1
Columna: F
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 8
Velocidad: 4
Actitud: Espadachin
Estado: **true**

Registrando al 2 soldado del Ejercito Aragon

Nombre: Espadachin Conquistador1X2
Vida: 14
Fila: 4
Columna: I
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 1
Actitud: Espadachin Conquistador
Estado: **true**

Registrando al 3 soldado del Ejercito Aragon

Nombre: Espadachin Conquistador2X2
Vida: 14
Fila: 6
Columna: B
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 2
Actitud: Espadachin Conquistador
Estado: **true**

Registrando al 4 soldado del Ejercito Aragon

Nombre: Espadachin Conquistador3X2
Vida: 14
Fila: 4
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 4
Actitud: Espadachin Conquistador
Estado: **true**

Registrando al 5 soldado del Ejercito Aragon

```
Nombre: Espadachin Conquistador4X2
Vida: 14
Fila: 6
Columna: D
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true
-----
```

Registrando al 6 soldado del Ejercito Aragon

```
Nombre: Espadachin5X2
Vida: 9
Fila: 8
Columna: A
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 9
Velocidad: 5
Actitud: Espadachin
Estado: true
-----
```

Registrando al 7 soldado del Ejercito Aragon

```
Nombre: Caballero6X2
Vida: 10
Fila: 10
Columna: E
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 10
Velocidad: 2
Actitud: Caballero
Estado: true
-----
```

Registrando al 8 soldado del Ejercito Aragon

```
Nombre: Caballero7X2
Vida: 10
Fila: 5
Columna: D
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 10
Velocidad: 5
Actitud: Caballero
Estado: true
-----
```

El tipo de territorio es: montana

El Ejercito Castilla del 1 ejercito sus soldados son :

El 1 soldado es:

Nombre: Espadachin Conquistador3X1

Vida: 14

Fila: 2

Columna: G

Nivel de ataque: 10

Nivel de Defensa: 8

Nivel de vida: 15

Velocidad: 3

Actitud: Espadachin Conquistador

Estado: **true**

El 2 soldado es:

Nombre: Lancero0X1

Vida: 6

Fila: 6

Columna: G

Nivel de ataque: 5

Nivel de Defensa: 10

Nivel de vida: 7

Velocidad: 1

Actitud: Lancero

Estado: **true**

El 3 soldado es:

Nombre: Lancero1X1

Vida: 6

Fila: 9

Columna: D

Nivel de ataque: 5

Nivel de Defensa: 10

Nivel de vida: 7

Velocidad: 5

Actitud: Lancero

Estado: **true**

El 4 soldado es:

Nombre: Espadachin Conquistador5X1

Vida: 14

Fila: 9

Columna: F

Nivel de ataque: 10

Nivel de Defensa: 8

Nivel de vida: 15

Velocidad: 1

Actitud: Espadachin Conquistador

```
Estado: true

*****
El 5 soldado es:

Nombre: Espadachin Conquistador2X1
Vida: 14
Fila: 9
Columna: J
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true

*****
El 6 soldado es:

Nombre: Caballero4X1
Vida: 10
Fila: 10
Columna: F
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 2
Actitud: Caballero
Estado: true
El Ejercito Aragon del 2 ejercito sus soldados son :

*****
El 1 soldado es:

Nombre: Espadachin0X2
Vida: 8
Fila: 1
Columna: F
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 9
Velocidad: 4
Actitud: Espadachin
Estado: true

*****
El 2 soldado es:

Nombre: Espadachin Conquistador3X2
Vida: 14
Fila: 4
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 4
```

```
Actitud: Espadachin Conquistador
Estado: true

*****

El 3 soldado es:

Nombre: Espadachin Conquistador1X2
Vida: 14
Fila: 4
Columna: I
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 1
Actitud: Espadachin Conquistador
Estado: true
```

```
*****

El 4 soldado es:

Nombre: Caballero7X2
Vida: 10
Fila: 5
Columna: D
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 5
Actitud: Caballero
Estado: true
```

```
*****

El 5 soldado es:

Nombre: Espadachin Conquistador2X2
Vida: 14
Fila: 6
Columna: B
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 2
Actitud: Espadachin Conquistador
Estado: true
```

```
*****

El 6 soldado es:

Nombre: Espadachin Conquistador4X2
Vida: 14
Fila: 6
Columna: D
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 3
```

Actitud: Espadachin Conquistador

Estado: **true**

El 7 soldado es:

Nombre: Espadachin5X2

Vida: 9

Fila: 8

Columna: A

Nivel de ataque: 10

Nivel de Defensa: 8

Nivel de vida: 10

Velocidad: 5

Actitud: Espadachin

Estado: **true**

El 8 soldado es:

Nombre: Caballero6X2

Vida: 10

Fila: 10

Columna: E

Nivel de ataque: 13

Nivel de Defensa: 7

Nivel de vida: 11

Velocidad: 2

Actitud: Caballero

Estado: **true**

El tipo de territorio es: montana

Mostrando tabla de posicion ... --
Leyenda: Ejercito1 --> 1#- | Ejercito2 --> 2#-

	A	B	C	D	E	F	G	H	I	J
1	Arquero.c					2 E 8				
2	Arquero.java						1EC14			
3	Caballero.c									
4	Caballero.java									
5	CaballeroFranco.c									
6	Ejercito.c	2EC14		2EC14			1 L 6			
7	Espadachin.c									
8	Espadachin.java									
9	EspadachinConquistador.c			1 L 6		1EC14				1EC14
10	EspadachinReal.c				2 C10	1 C10				

Listing 27: Ejecucion:

El soldado con mayor vida del Ejercito Castilla es:

Nombre: Espadachin Conquistador3X1
Vida: 14
Fila: 2
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: **true**

El soldado con mayor vida del Ejercito Aragon es:

Nombre: Espadachin Conquistador3X2
Vida: 14
Fila: 4
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 4
Actitud: Espadachin Conquistador
Estado: **true**

El promedio de puntos de vida del Ejercito Castilla es:
10.666666666666666

```
El promedio de puntos de vida del Ejercito Aragon es:
11.625
*****

El Ejercito Castilla ordenando por metodo insercion:
-----
Mostrando Ranking del Ejercito Castilla ..... /////// --->

Puesto 1
Nombre: Espadachin Conquistador3X1
Vida: 14
Fila: 2
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true
-----

Puesto 2
Nombre: Espadachin Conquistador5X1
Vida: 14
Fila: 9
Columna: F
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 1
Actitud: Espadachin Conquistador
Estado: true
-----

Puesto 3
Nombre: Espadachin Conquistador2X1
Vida: 14
Fila: 9
Columna: J
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true
-----

Puesto 4
Nombre: Caballero4X1
Vida: 10
Fila: 10
Columna: F
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 2
```

```
Actitud: Caballero
Estado: true
-----
```

```
Puesto 5
Nombre: Lancero0X1
Vida: 6
Fila: 6
Columna: G
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 1
Actitud: Lancero
Estado: true
-----
```

```
Puesto 6
Nombre: Lancero1X1
Vida: 6
Fila: 9
Columna: D
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 5
Actitud: Lancero
Estado: true
-----
```

```
El Ejercito Aragon ordenando por metodo insercion:
-----
```

```
Mostrando Ranking del Ejercito Aragon ..... /////  
-->
```

```
Puesto 1
Nombre: Espadachin Conquistador3X2
Vida: 14
Fila: 4
Columna: G
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 4
Actitud: Espadachin Conquistador
Estado: true
-----
```

```
Puesto 2
Nombre: Espadachin Conquistador1X2
Vida: 14
Fila: 4
Columna: I
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
```

```
Velocidad: 1
Actitud: Espadachin Conquistador
Estado: true
-----
```

```
Puesto 3
Nombre: Espadachin Conquistador2X2
Vida: 14
Fila: 6
Columna: B
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 2
Actitud: Espadachin Conquistador
Estado: true
-----
```

```
Puesto 4
Nombre: Espadachin Conquistador4X2
Vida: 14
Fila: 6
Columna: D
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 15
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true
-----
```

```
Puesto 5
Nombre: Caballero7X2
Vida: 10
Fila: 5
Columna: D
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 5
Actitud: Caballero
Estado: true
-----
```

```
Puesto 6
Nombre: Caballero6X2
Vida: 10
Fila: 10
Columna: E
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 2
Actitud: Caballero
Estado: true
-----
```



```
Puesto 7
Nombre: Espadachin5X2
Vida: 9
Fila: 8
Columna: A
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 5
Actitud: Espadachin
Estado: true
-----
```

```
Puesto 8
Nombre: Espadachin0X2
Vida: 8
Fila: 1
Columna: F
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 9
Velocidad: 4
Actitud: Espadachin
Estado: true
-----
```

```
*****

*****
El tipo de territorio es: montana

*****
```

Mostrando tabla de posicion ... --
Leyenda: Ejercito1 --> 1#- | Ejercito2 --> 2#-

	A	B	C	D	E	F	G	H	I	J
1	Arquero.c					2 E 8				
2	Arquero.java						1EC14			
3	Caballero.c									
4	Caballero.java						2EC14		2EC14	
5	CaballeroMoro.c			2 C10						
6	Ejercito.c	2EC14		2EC14			1 L 6			
7	Espadachin.c									
8	Espadachin.java									
9	EspadachinConquistador.c			1 L 6		1EC14				1EC14
10	EspadachinReal.c				2 C10	1 C10				

Listing 28: Ejecucion:

```
*****
Ejercito 1 : Castilla
Cantidad total de soldados creados: 6
Espadachines: 0
Arqueros: 0
Caballeros: 1
Lanceros: 2
Espadachin Real: 0
Caballero Moro: 0
Caballero Franco: 0
Espadachin Teutonico: 0
Espadachin Conquistador: 3

Ejercito 2 : Aragon
Cantidad total de soldados creados: 8
Espadachines: 2
Arqueros: 0
Caballeros: 2
Lanceros: 0
Espadachin Real: 0
Caballero Moro: 0
Caballero Franco: 0
Espadachin Teutonico: 0
Espadachin Conquistador: 4

Ejercito 1: Castilla: 70
Ejercito 2: Aragon: 101
El ganador es el ejercito 2 de: Aragon. Ya que al generar los porcentajes de
probabilidad de victoria basada en los niveles de vida de sus soldados y aplicando
```

```
un experimento aleatorio salio vencedor. (Aleatorio generado : 59.06)
```

```
*****
```

```
DESEA VOLVER A JUGAR
```

```
1 : JUGAR
```

```
2 : NO JUGAR
```

4.7. Ejercicio JuegoPrincipal

- En esta seccion en el método main, he creado una interfaz gráfica de usuario (GUI) utilizando la biblioteca Swing de Java. La interfaz consta de un JFrame llamado ventanaBienvenida que representa el menú de inicio del juego. Este marco tiene un tamaño predeterminado de 350x145 píxeles y se cerrará cuando el usuario haga clic en el botón de cierre. El diseño del contenido se gestiona mediante un JPanel llamado panelPrincipal, que utiliza un diseño de flujo (FlowLayout) para centrar y espaciar los elementos.
- Dentro del panel, he añadido un JLabel llamado mensajeLabel con el texto "¿Desea iniciar el juego?¿ dos botones (JButton), botonSi y botonNo, con los textos "Sí" "No" respectivamente. Los botones tienen colores de fondo distintivos (verde para "Sí" rojo para "No"). Se ha agregado un ActionListener a cada botón para manejar los eventos de clic.
- Si el usuario hace clic en el botón "Sí", se cierra la ventana de bienvenida (ventanaBienvenida.dispose()) y se crea un objeto Mapa para iniciar el juego llamando al método iniciarJuego() de la clase Mapa. Si el usuario hace clic en el botón "No", se cierra la ventana de bienvenida y se muestra un cuadro de diálogo (JOptionPane) con el mensaje "Programa finalizado".
- Finalmente, se configuran propiedades adicionales de la ventana, como su posición (ventanaBienvenida.setLocationRelativeTo(null)) para que aparezca en el centro de la pantalla, y se establece la visibilidad de la ventana en true para mostrarla al usuario.
- El código y el commit seria el siguiente:

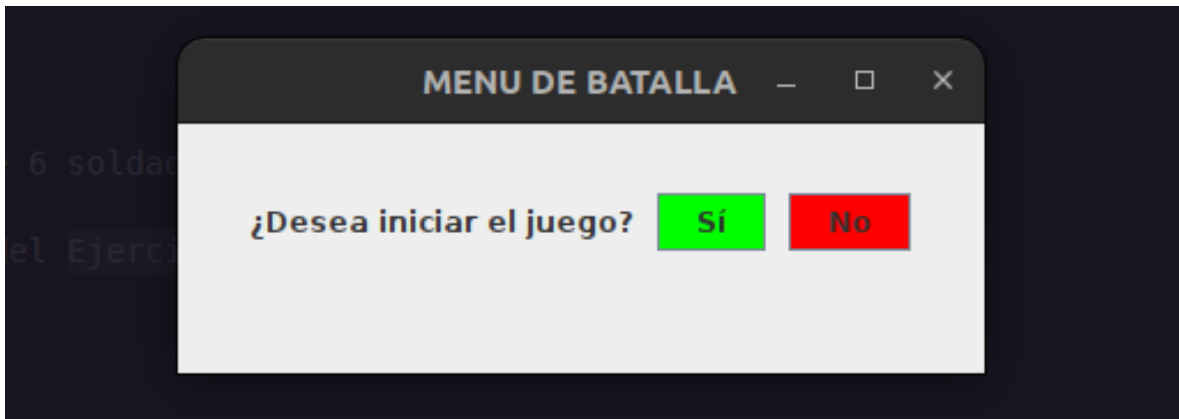
Listing 29: Commit

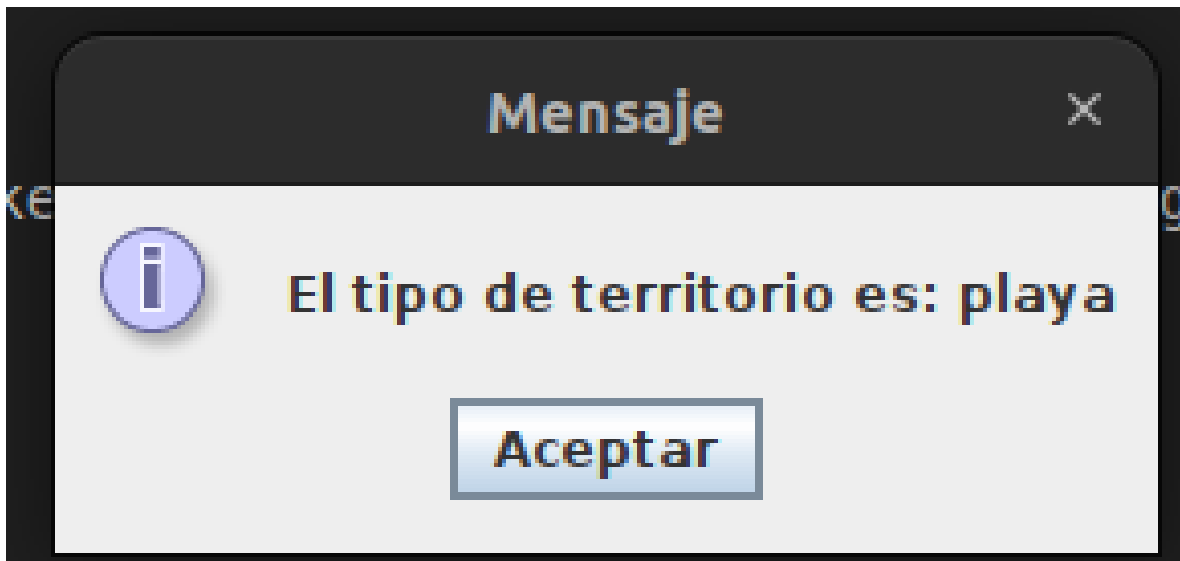
```
$ git commit -m "Agregandole GUI al main el cual importamos java swing y java awt para  
el menu de inicio del juego y los diferentes metodos que aplicamos para su buena  
disposicion graficamente"
```

Listing 30: Las lineas de codigos de la clase JuegoPrincipal

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
public class Juegoprincipal{  
    public static void main(String[] args) {  
        JFrame ventanaBienvenida = new JFrame("MENU DE BATALLA");  
        ventanaBienvenida.setSize(350, 145);  
        ventanaBienvenida.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        JPanel panelPrincipal = new JPanel(new FlowLayout(FlowLayout.CENTER, 10, 30));  
        JLabel mensajeLabel = new JLabel("-Desea iniciar el juego-");  
        JButton botonSi = new JButton("Si");  
        JButton botonNo = new JButton("No");  
        botonSi.setBackground(Color.GREEN);
```

```
        botonNo.setBackground(Color.RED);
        panelPrincipal.add(mensajeLabel);
        panelPrincipal.add(botonSi);
        panelPrincipal.add(botonNo);
        botonSi.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                ventanaBienvenida.dispose();
                Mapa mapa = new Mapa();
                mapa.iniciarJuego();
            }
        });
        botonNo.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                ventanaBienvenida.dispose();
                JOptionPane.showMessageDialog(null, "Programa finalizado.");
            }
        });
        ventanaBienvenida.add(panelPrincipal);
        ventanaBienvenida.setLocationRelativeTo(null);
        ventanaBienvenida.setVisible(true);
    }
}
```





Listing 31: Ejecucion:

```
El Ejercito Castilla tiene 9 soldados :

Registrando al 1 soldado del Ejercito Castilla

Nombre: Arquero0X1
Vida: 4
Fila: 1
Columna: J
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 4
Velocidad: 5
Actitud: Arquero
Estado: true
-----
Registrando al 2 soldado del Ejercito Castilla

Nombre: Caballero1X1
Vida: 11
Fila: 5
Columna: J
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 4
Actitud: Caballero
Estado: true
-----
Registrando al 3 soldado del Ejercito Castilla

Nombre: Arquero2X1
Vida: 5
Fila: 10
Columna: J
```

Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 5
Velocidad: 5
Actitud: Arquero
Estado: **true**

Registrando al 4 soldado del Ejercito Castilla

Nombre: Caballero3X1
Vida: 11
Fila: 3
Columna: C
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 2
Actitud: Caballero
Estado: **true**

Registrando al 5 soldado del Ejercito Castilla

Nombre: Arquero4X1
Vida: 4
Fila: 5
Columna: B
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 4
Velocidad: 3
Actitud: Arquero
Estado: **true**

Registrando al 6 soldado del Ejercito Castilla

Nombre: Caballero5X1
Vida: 10
Fila: 9
Columna: D
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 10
Velocidad: 4
Actitud: Caballero
Estado: **true**

Registrando al 7 soldado del Ejercito Castilla

Nombre: Caballero6X1
Vida: 12
Fila: 5
Columna: H
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 2

```
Actitud: Caballero
Estado: true
-----
Registrando al 8 soldado del Ejercito Castilla
```

```
Nombre: Caballero7X1
Vida: 10
Fila: 2
Columna: I
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 10
Velocidad: 3
Actitud: Caballero
Estado: true
-----
```

```
Registrando al 9 soldado del Ejercito Castilla
```

```
Nombre: Espadachin Conquistador8X1
Vida: 14
Fila: 9
Columna: H
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true
-----
```

```
*****
El Ejercito Moros tiene 8 soldados :
```

```
Registrando al 1 soldado del Ejercito Moros
```

```
Nombre: Caballero0X2
Vida: 12
Fila: 9
Columna: D
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 1
Actitud: Caballero
Estado: true
-----
```

```
Registrando al 2 soldado del Ejercito Moros
```

```
Nombre: Caballero1X2
Vida: 12
Fila: 10
Columna: A
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 3
Actitud: Caballero
```

Estado: true

Registrando al 3 soldado del Ejercito Moros

Nombre: Espadachin2X2
Vida: 10
Fila: 6
Columna: B
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 2
Actitud: Espadachin
Estado: true

Registrando al 4 soldado del Ejercito Moros

Nombre: Espadachin3X2
Vida: 8
Fila: 4
Columna: J
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 8
Velocidad: 3
Actitud: Espadachin
Estado: true

Registrando al 5 soldado del Ejercito Moros

Nombre: Caballero Moro4X2
Vida: 13
Fila: 8
Columna: A
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 13
Velocidad: 5
Actitud: Caballero Moro
Estado: true

Registrando al 6 soldado del Ejercito Moros

Nombre: Caballero Moro5X2
Vida: 13
Fila: 1
Columna: A
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 13
Velocidad: 2
Actitud: Caballero Moro
Estado: true

Registrando al 7 soldado del Ejercito Moros


```
Nombre: Lancero6X2
Vida: 7
Fila: 9
Columna: H
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 2
Actitud: Lancero
Estado: true
```

Registrando al 8 soldado del Ejercito Moros

```
Nombre: Arquero7X2
Vida: 5
Fila: 1
Columna: H
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 5
Velocidad: 5
Actitud: Arquero
Estado: true
```

El tipo de territorio es: playa

El Ejercito Castilla del 1 ejercito sus soldados son :

El 1 soldado es:

```
Nombre: Arquero0X1
Vida: 4
Fila: 1
Columna: J
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 4
Velocidad: 5
Actitud: Arquero
Estado: true
```

El 2 soldado es:

```
Nombre: Caballero7X1
Vida: 10
Fila: 2
Columna: I
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 10
```

```
Velocidad: 3
Actitud: Caballero
Estado: true

*****
El 3 soldado es:

Nombre: Caballero3X1
Vida: 11
Fila: 3
Columna: C
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 2
Actitud: Caballero
Estado: true

*****
El 4 soldado es:

Nombre: Arquero4X1
Vida: 4
Fila: 5
Columna: B
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 4
Velocidad: 3
Actitud: Arquero
Estado: true

*****
El 5 soldado es:

Nombre: Caballero6X1
Vida: 12
Fila: 5
Columna: H
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 2
Actitud: Caballero
Estado: true

*****
El 6 soldado es:

Nombre: Caballero1X1
Vida: 11
Fila: 5
Columna: J
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
```

```
Velocidad: 4
Actitud: Caballero
Estado: true

*****
El 7 soldado es:

Nombre: Caballero5X1
Vida: 10
Fila: 9
Columna: D
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 10
Velocidad: 4
Actitud: Caballero
Estado: true

*****
El 8 soldado es:

Nombre: Espadachin Conquistador8X1
Vida: 14
Fila: 9
Columna: H
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true

*****
El 9 soldado es:

Nombre: Arquero2X1
Vida: 5
Fila: 10
Columna: J
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 5
Velocidad: 5
Actitud: Arquero
Estado: true
El Ejercito Moros del 2 ejercito sus soldados son :

*****
El 1 soldado es:

Nombre: Caballero Moro5X2
Vida: 13
Fila: 1
Columna: A
Nivel de ataque: 13
Nivel de Defensa: 7
```

```
Nivel de vida: 13
Velocidad: 2
Actitud: Caballero Moro
Estado: true

*****
El 2 soldado es:

Nombre: Arquero7X2
Vida: 5
Fila: 1
Columna: H
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 5
Velocidad: 5
Actitud: Arquero
Estado: true

*****
El 3 soldado es:

Nombre: Espadachin3X2
Vida: 8
Fila: 4
Columna: J
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 8
Velocidad: 3
Actitud: Espadachin
Estado: true

*****
El 4 soldado es:

Nombre: Espadachin2X2
Vida: 10
Fila: 6
Columna: B
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 10
Velocidad: 2
Actitud: Espadachin
Estado: true

*****
El 5 soldado es:

Nombre: Caballero Moro4X2
Vida: 13
Fila: 8
Columna: A
Nivel de ataque: 13
Nivel de Defensa: 7
```

```
Nivel de vida: 13
Velocidad: 5
Actitud: Caballero Moro
Estado: true

*****
El 6 soldado es:

Nombre: Caballero0X2
Vida: 12
Fila: 9
Columna: D
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 1
Actitud: Caballero
Estado: true

*****
El 7 soldado es:

Nombre: Lancero6X2
Vida: 7
Fila: 9
Columna: H
Nivel de ataque: 5
Nivel de Defensa: 10
Nivel de vida: 7
Velocidad: 2
Actitud: Lancero
Estado: true

*****
El 8 soldado es:

Nombre: Caballero1X2
Vida: 12
Fila: 10
Columna: A
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 3
Actitud: Caballero
Estado: true

*****
El tipo de territorio es: playa

*****
```

Mostrando tabla de posicion ... --
Leyenda: Ejercito1 --> 1#- | Ejercito2 --> 2#-

	A	B	C	D	E	F	G	H	I	J
1	2CM13							2 A 5		1 A 4
2									1 C10	
3			1 C11							
4										2 E 8
5		1 A 4						1 C12		1 C11
6		2 E10								
7										
8	2CM13									
9				2 C 2				1EC 7		
10	2 C12									1 A 5

Listing 32: Ejecucion:

```
*****
El soldado con mayor vida del Ejercito Castilla es:

Nombre: Caballero6X1
Vida: 12
Fila: 5
Columna: H
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 2
Actitud: Caballero
Estado: true
*****
El soldado con mayor vida del Ejercito Moros es:

Nombre: Caballero Moro5X2
Vida: 13
Fila: 1
Columna: A
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 13
Velocidad: 2
Actitud: Caballero Moro
Estado: true
*****
El promedio de puntos de vida del Ejercito Castilla es:
8.0
```

```
*****
El promedio de puntos de vida del Ejercito Moros es:
9.0
*****

El Ejercito Castilla ordenando por metodo insercion:
-----
Mostrando Ranking del Ejercito Castilla ..... //---->

Puesto 1
Nombre: Caballero6X1
Vida: 12
Fila: 5
Columna: H
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 12
Velocidad: 2
Actitud: Caballero
Estado: true
-----

Puesto 2
Nombre: Caballero3X1
Vida: 11
Fila: 3
Columna: C
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 2
Actitud: Caballero
Estado: true
-----

Puesto 3
Nombre: Caballero1X1
Vida: 11
Fila: 5
Columna: J
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 11
Velocidad: 4
Actitud: Caballero
Estado: true
-----

Puesto 4
Nombre: Caballero7X1
Vida: 10
Fila: 2
Columna: I
Nivel de ataque: 13
Nivel de Defensa: 7
Nivel de vida: 10
```

```
Velocidad: 3
Actitud: Caballero
Estado: true
-----

Puesto 5
Nombre: Espadachin Conquistador8X1
Vida: 7
Fila: 9
Columna: H
Nivel de ataque: 10
Nivel de Defensa: 8
Nivel de vida: 14
Velocidad: 3
Actitud: Espadachin Conquistador
Estado: true
-----

Puesto 6
Nombre: Arquero2X1
Vida: 5
Fila: 10
Columna: J
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 5
Velocidad: 5
Actitud: Arquero
Estado: true
-----

Puesto 7
Nombre: Arquero0X1
Vida: 4
Fila: 1
Columna: J
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 4
Velocidad: 5
Actitud: Arquero
Estado: true
-----

Puesto 8
Nombre: Arquero4X1
Vida: 4
Fila: 5
Columna: B
Nivel de ataque: 7
Nivel de Defensa: 3
Nivel de vida: 4
Velocidad: 3
Actitud: Arquero
Estado: true
-----
```

El Ejercito Moros ordenando por metodo insercion:

Mostrando Ranking del Ejercito Moros /////

Puesto 1

Nombre: Caballero Moro5X2

Vida: 13

Fila: 1

Columna: A

Nivel de ataque: 13

Nivel de Defensa: 7

Nivel de vida: 13

Velocidad: 2

Actitud: Caballero Moro

Estado: **true**

Puesto 2

Nombre: Caballero Moro4X2

Vida: 13

Fila: 8

Columna: A

Nivel de ataque: 13

Nivel de Defensa: 7

Nivel de vida: 13

Velocidad: 5

Actitud: Caballero Moro

Estado: **true**

Puesto 3

Nombre: Caballero1X2

Vida: 12

Fila: 10

Columna: A

Nivel de ataque: 13

Nivel de Defensa: 7

Nivel de vida: 12

Velocidad: 3

Actitud: Caballero

Estado: **true**

Puesto 4

Nombre: Espadachin2X2

Vida: 10

Fila: 6

Columna: B

Nivel de ataque: 10

Nivel de Defensa: 8

Nivel de vida: 10

Velocidad: 2

Actitud: Espadachin

Estado: **true**

```
-----  
  
Puesto 5  
Nombre: Espadachin3X2  
Vida: 8  
Fila: 4  
Columna: J  
Nivel de ataque: 10  
Nivel de Defensa: 8  
Nivel de vida: 8  
Velocidad: 3  
Actitud: Espadachin  
Estado: true  
-----
```

```
  
Puesto 6  
Nombre: Arquero7X2  
Vida: 5  
Fila: 1  
Columna: H  
Nivel de ataque: 7  
Nivel de Defensa: 3  
Nivel de vida: 5  
Velocidad: 5  
Actitud: Arquero  
Estado: true  
-----
```

```
  
Puesto 7  
Nombre: Caballero0X2  
Vida: 2  
Fila: 9  
Columna: D  
Nivel de ataque: 13  
Nivel de Defensa: 7  
Nivel de vida: 12  
Velocidad: 1  
Actitud: Caballero  
Estado: true  
-----
```

```
*****
```

```
*****
```

```
El tipo de territorio es: playa
```

```
*****
```

Mostrando tabla de posicion ... --
Leyenda: Ejercito1 --> 1#- | Ejercito2 --> 2#-

	A	B	C	D	E	F	G	H	I	J
1	2CM13							2 A 5		1 A 4
2									1 C10	
3			1 C11							
4										2 E 8
5		1 A 4								
6		2 E10								
7										
8	2CM13									
9				2 C 2				1EC 7		
10	2 C12									1 A 5

Listing 33: Ejecucion:

```
*****
Ejercito 1 : Castilla
Cantidad total de soldados creados: 8
Espadachines: 0
Arqueros: 3
Caballeros: 4
Lanceros: 0
Espadachin Real: 0
Caballero Moro: 0
Caballero Franco: 0
Espadachin Teutonico: 0
Espadachin Conquistador: 1

Ejercito 2 : Moros
Cantidad total de soldados creados: 7
Espadachines: 2
Arqueros: 1
Caballeros: 2
Lanceros: 0
Espadachin Real: 0
Caballero Moro: 2
Caballero Franco: 0
Espadachin Teutonico: 0
Espadachin Conquistador: 0

Ejercito 1: Castilla: 71
Ejercito 2: Moros: 73
El ganador es el ejercito 2 de: Moros. Ya que al generar los porcentajes de
probabilidad de victoria basada en los niveles de vida de sus soldados y aplicando
```

un experimento aleatorio salio vencedor. (Aleatorio generado : 50.69)

DESEA VOLVER A JUGAR

1 : JUGAR

2 : NO JUGAR

4.8. Estructura de laboratorio 22

- El contenido que se entrega en este laboratorio22 es el siguiente:

\Lab22

5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos los ítems.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	4	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente están dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	2	
Total		20		18	

6. Referencias

- https://drive.google.com/drive/folders/1_y046U0axs7uKVK7nrrkcNwybnk_ZJXJ