

Manual tecnico

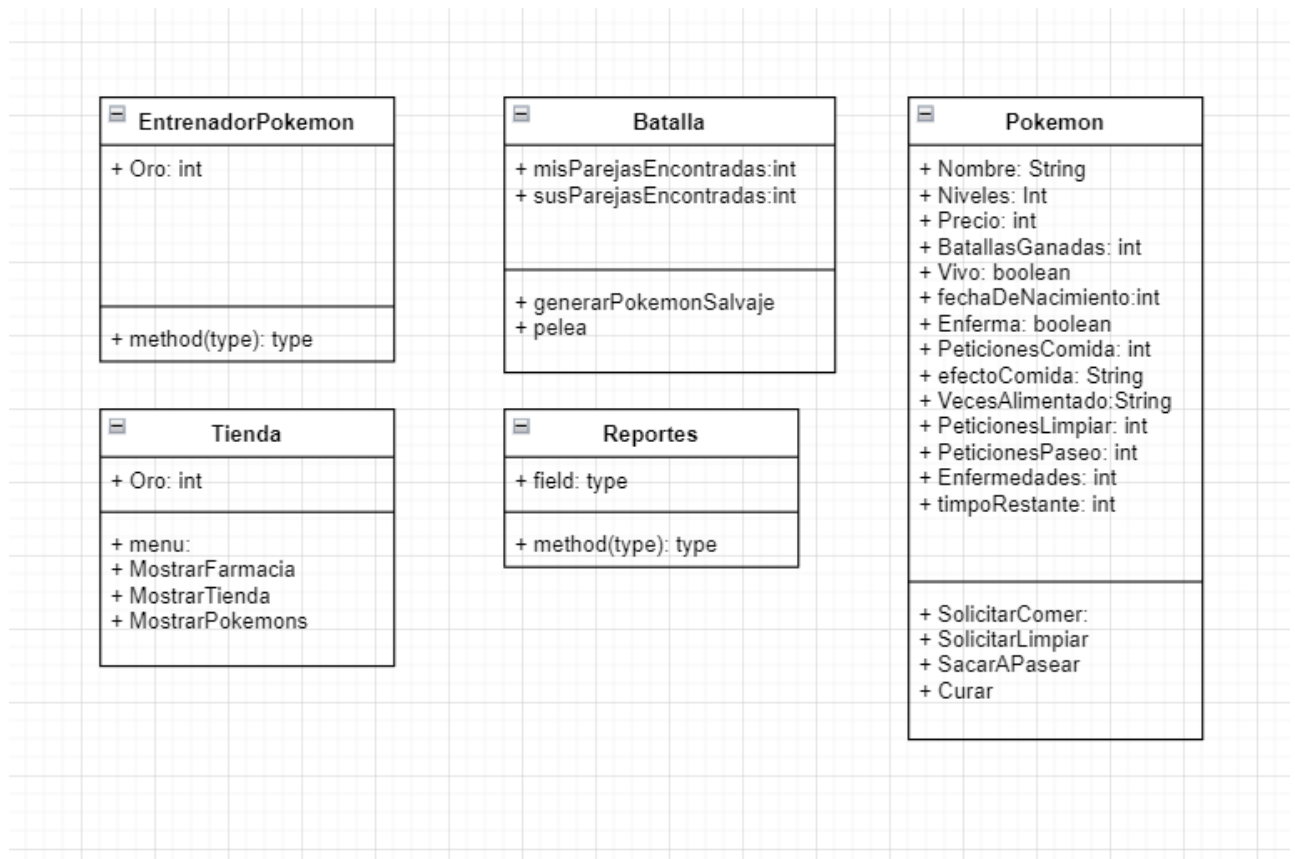
Sistema operativo: Windows 10

IDE: Apache Netbeans 12.6

Lenguaje: Java

JDK: 15

Diagrama UML inicial



InterfazDelJuego

(no se incluyen los initComponents que genera el IDE automáticamente)

```
private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {  
    Tienda tienda = new Tienda();  
    tienda.setVisible(true);  
    tienda.getLabelOro().setText(tienda.getDineroString());  
    tienda.getLabelOro2().setText(tienda.getDineroString());  
    tienda.getLabelOro1().setText(tienda.getDineroString());  
}
```

```
private void BotonMascota1ActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        NombreMascota1.setText(tienda.getPokemons(0).getNombre());
```

```

        mostrarPokemon(0);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

```

```

private void BotonMascota2ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota2.setText(tienda.getPokemones(1).getNombre());
        mostrarPokemon(1);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

```

```

private void BotonMascota3ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota3.setText(tienda.getPokemones(2).getNombre());
        mostrarPokemon(2);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

```

```

private void BotonMascota4ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota4.setText(tienda.getPokemones(3).getNombre());
        mostrarPokemon(3);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

```

```

private void BotonMascota5ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota5.setText(tienda.getPokemones(4).getNombre());
        mostrarPokemon(4);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

```

```

private void BotonMascota9ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota6.setText(tienda.getPokemones(5).getNombre());
        mostrarPokemon(5);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

```

```

    }
}

private void BotonMascota6ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota7.setText(tienda.getPokemones(6).getNombre());
        mostrarPokemon(6);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

private void BotonMascota7ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota8.setText(tienda.getPokemones(7).getNombre());
        mostrarPokemon(7);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

private void BotonMascota8ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota9.setText(tienda.getPokemones(8).getNombre());
        mostrarPokemon(8);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

private void BotonMascota10ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        NombreMascota10.setText(tienda.getPokemones(9).getNombre());
        mostrarPokemon(9);
    } catch (NullPointerException e) {
        JOptionPane.showMessageDialog(this, "No hay un pokemon en ese lugar");
    }
}

private void BotonRevivirActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        int pokemonARevivir = Integer.parseInt(labelAnimalARevivir.getText());
        if (!tienda.getPokemones(pokemonARevivir).isVivo()) {
            int precio = (5 * tienda.getPokemones(pokemonARevivir).getNivel()) + 10;
            tienda.setDinero(precio);
            tienda.getPokemones(pokemonARevivir).cambiarEstado(true);
            JOptionPane.showMessageDialog(this,
tienda.getPokemones(pokemonARevivir).getNombre() + " ha revivido");
        }
    }
}

```

```

        tienda.getPokemones(pokemonARevivir).resume();
        tienda.getPokemones(pokemonARevivir).reiniciarHambre();
    } else {
        JOptionPane.showMessageDialog(this, "Este pokemon esta vivo");
    }
} catch (NullPointerException e) {
    JOptionPane.showMessageDialog(this, "No hay ningun pokemon en esta posicion");
}
}

private void BotonLimpiarActionPerformed(java.awt.event.ActionEvent evt) {
    tienda.getPokemones(this.posicionActual).limpiar();
    JOptionPane.showMessageDialog(this, "Haz limpiado a " +
tienda.getPokemones(this.posicionActual).getNombre());
}

private void BotonPasearActionPerformed(java.awt.event.ActionEvent evt) {
    menuinicio juego = new menuinicio();
    juego.setVisible(true);
    tienda.aumentarDinero();
}

private void mostrarPokemon(int noMascota) {
    if (tienda.pokemones[noMascota].isVivo()) {
        LabelNombre.setText(tienda.getPokemones(noMascota).getNombre());
        LabelNivel.setText(tienda.getPokemones(noMascota).getNivelString());
        LabelTipo.setText(tienda.getPokemones(noMascota).getTipo());
        imagenPokemon.setIcon(new
ImagenIcon(tienda.getPokemones(noMascota).getImagen()));
        labelMostrarHambre.setText(tienda.getPokemones(noMascota).getHambreString());
        labelEfecto.setText(tienda.getPokemones(noMascota).getEfectoComida());

labelEnfermedades.setText(tienda.getPokemones(noMascota).getEnfermedadString());
        labelLimpiar.setText(tienda.getPokemones(noMascota).getLimpiar());
        labelPasear.setText(tienda.getPokemones(noMascota).getPasear());
        this.posicionActual = noMascota;
    } else {
        imagenPokemon.setIcon(new ImagenIcon("C:/Users/julio/Documents/Tareas
universidad tercer semestre/intro a la
progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/muerto.png"));
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */

```

```

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
/* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and
feel.
   * For details see
   http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
   */
try {
    for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(InterfazJuego.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(InterfazJuego.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(InterfazJuego.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(InterfazJuego.class.getName()).log(java.util.logging.Level
.SEVERE, null, ex);
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new InterfazJuego().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton BotonAlimentar1;
private javax.swing.JButton BotonLimpiar;
private javax.swing.JButton BotonMascota1;
private javax.swing.JButton BotonMascota10;
private javax.swing.JButton BotonMascota2;
private javax.swing.JButton BotonMascota3;

```

```
private javax.swing.JButton BotonMascota4;
private javax.swing.JButton BotonMascota5;
private javax.swing.JButton BotonMascota6;
private javax.swing.JButton BotonMascota7;
private javax.swing.JButton BotonMascota8;
private javax.swing.JButton BotonMascota9;
private javax.swing.JButton BotonPasear;
private javax.swing.JButton BotonRevivir;
private javax.swing.JLabel LabelNivel;
private javax.swing.JLabel LabelNombre;
private javax.swing.JLabel LabelTipo;
private javax.swing.JMenu MenuAcercaDe;
private javax.swing.JMenu MenuAyuda;
private javax.swing.JMenu MenuReportes;
private javax.swing.JMenu MenuTienda;
private javax.swing.JLabel NombreMascota1;
private javax.swing.JLabel NombreMascota10;
private javax.swing.JLabel NombreMascota2;
private javax.swing.JLabel NombreMascota3;
private javax.swing.JLabel NombreMascota4;
private javax.swing.JLabel NombreMascota5;
private javax.swing.JLabel NombreMascota6;
private javax.swing.JLabel NombreMascota7;
private javax.swing.JLabel NombreMascota8;
private javax.swing.JLabel NombreMascota9;
private javax.swing.JLabel imagenPokemon;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel9;
private javax.swing.JMenuBar jMenuBar1;
private javax.swing.JMenuItem jMenuItem1;
private javax.swing.JPanel jPanel1;
private javax.swing.JTextField labelAnimalARevivir;
private javax.swing.JLabel labelEfecto;
private javax.swing.JLabel labelEnfermedades;
private javax.swing.JLabel labelLimpiar;
private javax.swing.JLabel labelMostrarHambre;
private javax.swing.JLabel labelPasear;
```

Tienda

(no se incluyen los initComponents que genera el IDE automáticamente)

```
public class Tienda extends javax.swing.JFrame {

    static Pokemon[] pokemones = new Pokemon[10];
    private static int posicion;
    private static int dinero=100;
    /**
     * Creates new form Tienda
     */

    public Tienda() {
        initComponents();

        this.setLocationRelativeTo(null);
        this.setTitle("Tienda");
        this.getContentPane().setBackground(getBackground());
    }

    public void aumentarDinero(){
        this.dinero+=50;
    }

    public String getDineroString(){
        String dinero= String.valueOf(this.dinero);
        return dinero;
    }

    public Pokemon getPokemones(int posicion) {
        return pokemones[posicion];
    }

    public int getPosicion() {
        return posicion;
    }

    public void setDinero(int dinero) {
        Tienda.dinero -= dinero;
    }

    public JLabel getLabelOro() {
        return labelOro;
    }

    public JLabel getLabelOro1() {
        return labelOro1;
    }
}
```

```

}

public JLabel getLabelOro2() {
    return labelOro2;
}

private void comprarComida(String comida, int precio, int aumento){
    try{
        if(this.dinero>=precio){
            int posicion=Integer.parseInt(TextFieldMascotaADarComida.getText());
            pokemones[posicion].restarHambre();
            pokemones[posicion].setEfectoComida(comida);
            pokemones[posicion].aumentarHambreMaxima(aumento);
            this.dinero-=precio;
            String dinero= String.valueOf(this.dinero);
            labelOro.setText(dinero);

        }else{
            JOptionPane.showMessageDialog(this,"No tienes dinero suficiente");
        }
    }catch(NullPointerException e){
        JOptionPane.showMessageDialog(this,"No tienes ningun pokemon en esa
posicion");
    }
}

private void curarEnfermedad(int medicamento, int precio){
    try{
        if(this.dinero>=precio){
            int posicion = Integer.parseInt(TextFielMascotaADarMedicamento.getText());
            pokemones[posicion].restarEnfermedades(medicamento);
            this.dinero-=precio;
            String dinero= String.valueOf(this.dinero);
            labelOro1.setText(dinero);
        }

    }catch(NullPointerException e){
        JOptionPane.showMessageDialog(this,"No tienes ningun pokemon en esa posicion");
    }
}

private void BotonWafflesActionPerformed(java.awt.event.ActionEvent evt) {
    comprarComida("Waffles",50,10);
}

```



```

private void BotonManzana1ActionPerformed(java.awt.event.ActionEvent evt) {
    comprarComida("Manzana",10,5);
}

private void BotonCereal1ActionPerformed(java.awt.event.ActionEvent evt) {
    comprarComida("Cereal",30,7);
}

private void TextFieldMascotaADarComidaActionPerformed(java.awt.event.ActionEvent
evt) {
    // TODO add your handling code here:
}

private void
TextFielMascotaADarMedicamentoActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void BotonVitaminaActionPerformed(java.awt.event.ActionEvent evt) {
    curarEnfermedad(2,50);
}

private void BotonVitamina1ActionPerformed(java.awt.event.ActionEvent evt) {
    curarEnfermedad(1,20);
}

private void BotonComprarActionPerformed(java.awt.event.ActionEvent evt) {

    int comprar = JOptionPane.showConfirmDialog(this,"Seguro que quieres comprar este
pokemon");
    if(comprar==0){
        if(this.dinero>=50){
            pokemones[this.posicion]= new Pokemon();
            pokemones[this.posicion].setNombre(NombrePokemon.getText());
            int tipo=Integer.parseInt(NoPokemonAComprar.getText());
            pokemones[this.posicion].setTipo(tipo);
            pokemones[this.posicion].setImagen(tipo);
            pokemones[this.posicion].start();
            this.posicion=this.posicion+1;
            this.dinero-=50;
            String dinero= String.valueOf(this.dinero);
            labelOro2.setText(dinero);
        }else{
            JOptionPane.showMessageDialog(this,"No tienes dinero suficiente para
comprar");
        }
    }
}

```

```

    }

}

private void NombrePokemonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void NoPokemonAComprarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void BotonVitamina2ActionPerformed(java.awt.event.ActionEvent evt) {
    curarEnfermedad(3,80);
}


// Variables declaration - do not modify
private javax.swing.JButton BotonCereal1;
private javax.swing.JButton BotonComprar;
private javax.swing.JButton BotonManzana1;
private javax.swing.JButton BotonVitamina;
private javax.swing.JButton BotonVitamina1;
private javax.swing.JButton BotonVitamina2;
private javax.swing.JButton BotonWaffles;
private javax.swing.JTextField NoPokemonAComprar;
private javax.swing.JTextField NombrePokemon;
private javax.swing.JTextField TextFielMascotaADarMedicamento;
private javax.swing.JTextField TextFieldMascotaADarComida;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JTabbedPane jTabbedPane1;
private javax.swing.JLabel labelOro;
private javax.swing.JLabel labelOro1;
private javax.swing.JLabel labelOro2;

```

Pokemon

```
import static java.lang.Thread.sleep;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;

/**
 *
 * @author julio
 */
public class Pokemon extends Thread {

    private String nombre;
    private String efectoComida = "Ninguno";
    private String tipo;
    private String imagen;
    private boolean vivo = true;
    private int nivel = 1;
    private int batallasGanadas;
    private int enfermedades = 0;
    private int alimentado = 0;
    private int hambre = 0;
    private int hambreMaxima = 5;
    private int limpiar = 0;
    private int pasear = 0;
    private int paseoMaximo=4;
    private int tiempoDeVida;

    //getters y setters
    public String getLimpiar() {
        String limpiar = String.valueOf(this.limpiar);
        return limpiar;
    }

    public String getPasear() {
        String pasear = String.valueOf(this.pasear);
        return pasear;
    }

    public String getEnfermedadString() {
        String enfermedades = String.valueOf(this.enfermedades);
        return enfermedades;
    }

    public void setEfectoComida(String efecto) {
        this.efectoComida = efecto;
    }
}
```

```
public String getEfectoComida() {  
    return this.efectoComida;  
}
```

```
public String getTipo() {  
    return tipo;  
}
```

```
public void setTipo(int tipo) {  
    switch (tipo) {  
        case 1 -> {  
            this.tipo = "Bulbasaur";  
        }  
        case 2 -> {  
            this.tipo = "Ivysaur";  
        }  
        case 3 -> {  
            this.tipo = "Venusaur";  
        }  
        case 4 -> {  
            this.tipo = "Charmander";  
        }  
        case 5 -> {  
            this.tipo = "Charmeleon";  
        }  
        case 6 -> {  
            this.tipo = "Charizard";  
        }  
        case 7 -> {  
            this.tipo = "Squirtle";  
        }  
        case 8 -> {  
            this.tipo = "Wartortle";  
        }  
        case 9 -> {  
            this.tipo = "Blastoise";  
        }  
        case 10 -> {  
            this.tipo = "Caterpie";  
        }  
    }  
}
```

```
public String getImagen() {  
    return this.imagen;  
}
```

```

public void setImagen(int tipo) {
    switch (tipo) {
        case 1 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/bulbasaur.png";
        }
        case 2 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/lvysaur.png";
        }
        case 3 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/Venusaur.png";
        }
        case 4 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/Pokemon.png";
        }
        case 5 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/Charmeleon.png";
        }
        case 6 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/Charizard.png";
        }
        case 7 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/Squirtle.png";
        }
        case 8 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/Wartortle.png";
        }
        case 9 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/Blastoise.png";
        }
        case 10 -> {
            this.imagen = "C:/Users/julio/Documents/Tareas universidad tercer semestre/intro
a la progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/Caterpie.png";
        }
    }
}

public String getNombre() {
    return nombre;
}

```

```

}

public String getHambreString() {
    String hambre = String.valueOf(this.hambre);
    return hambre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public int getNivel() {
    return nivel;
}

public String getNivelString() {
    String nivel = String.valueOf(this.nivel);
    return nivel;
}

public boolean isVivo() {
    return vivo;
}

public void cambiarEstado(boolean estado) {
    this.vivo = estado;
}

public void reiniciarHambre() {
    this.hambre = 0;
}

//procesos

public void aumentarHambreMaxima(int aumento) {
    this.hambreMaxima = aumento;
}

public void restarHambre() {
    if (this.hambre >= 1) {
        this.hambre = this.hambre - 1;
    }
    this.alimentado++;
    pedirLimpieza();
}

public void restarEnfermedades(int medicamento) {

```

```

        if (this.enfermedades >= 1) {
            this.enfermedades -= medicamento;
        }
    }

    public void pedirComida() {
        if (vivo) {
            try {
                sleep(vecesPedirComida());
            } catch (InterruptedException ex) {
                Logger.getLogger(Pokemon.class.getName()).log(Level.SEVERE, null, ex);
            }
            JOptionPane.showMessageDialog(null, this.nombre + " tiene hambre");
            this.hambre++;
            if (this.hambre == this.hambreMaxima) {
                JOptionPane.showMessageDialog(null, this.nombre + " ha muerto de hambre :(");
                this.vivo = false;
                suspend();
            }
        }
    }

    public void pedirLimpieza() {
        if (this.alimentado == 2) {
            JOptionPane.showMessageDialog(null, this.nombre + " necesita que lo limpies");
            this.limpiar++;
        }
    }

    public void limpiar() {
        if (this.limpiar > 0) {
            this.limpiar--;
        } else {
            JOptionPane.showMessageDialog(null, this.nombre + " ya esta limpio");
        }
    }

    public void pedirCurar() {
        System.out.println(this.nombre + " esta enfermo");
        this.enfermedades++;
    }

    public void pedirPaseo() {
        try {

```

```

        sleep(vecesPedirPaseo());
    } catch (InterruptedException ex) {
        Logger.getLogger(Pokemon.class.getName()).log(Level.SEVERE, null, ex);
    }
    JOptionPane.showMessageDialog(null, this.nombre + " quiere pasear");
    this.pasear++;
    if (this.pasear == this.paseoMaximo) {
        JOptionPane.showMessageDialog(null, this.nombre + " se ha enfermado");
        this.enfermedades++;
    }
}

//tiempos

public int vecesPedirComida() {
    if (this.nivel == 1) {
        return 60000;
    } else if (this.nivel == 2) {
        return 50000;
    } else if (this.nivel == 3) {
        return 30000;
    } else if (this.nivel == 4) {
        return 20000;
    } else {
        return 10000;
    }
}

public int vecesPedirPaseo() {
    if (this.nivel == 1) {
        return 60000;
    } else if (this.nivel == 2) {
        return 40000;
    } else if (this.nivel == 3) {
        return 30000;
    } else if (this.nivel == 4) {
        return 20000;
    } else {
        return 10000;
    }
}

//run
@Override
public void run() {
    while (true) {

```



```

        pedirComida();
        pedirPaseo();
    }

}

}

```

ConfirmarCompra

```

public class ConfirmarCompra extends javax.swing.JFrame {

    /**
     * Creates new form ConfirmarCompra
     */
    public ConfirmarCompra() {
        initComponents();
        this.setLocationRelativeTo(null);
        this.setTitle("Tienda");
        this.getContentPane().setBackground(getBackground());
    }
}

```

JuegoDeMemoria

```

public class JuegoMemoria extends JFrame implements ActionListener{
    JPanel panel;
    JLabel matriz [][],etiqueta,etique,nombreju,cronometro,lblfecha,lblhora;
    int mat [][] = new int[4][5];
    int mat2 [][] = new int[4][5];
    Random ran;
    int contador,ban,ban1,annum,anposx,anposy,acnum,acposx,acposy;
    Timer espera, espera2,tiempo;
    int consegund,seg,min;
    int hora,minutos,segundos;
    JButton reiniciar;
    //Thread hilo;

    public JuegoMemoria(){
        this.setTitle("Juego de Memoria");
        this.setSize(1010, 720);
        this.setLocationRelativeTo(null);
        this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        this.setResizable(false);

        //tiempo.start();

        //se coloca un panel a la ventana
        panel = new JPanel();
        this.getContentPane().add(panel);
    }
}

```

```

panel.setLayout(null);

//proposito de mat2, que las cartas aparescan volteadas
ran = new Random();
this.numaleatorios();

//matriz de imagenes
//este hara una matriz de 4filas por 5columnas donde mostrara imagenes
matriz = new JLabel[4][5];
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 5; j++) {
        matriz[i][j] = new JLabel();
        matriz[i][j].setSize(matriz[i][j].getWidth(), matriz[i][j].getHeight());
        //aqui la demas espacios a la imagenes para que no salgan pegadas
        matriz[i][j].setBounds(350+(j*125),30+(i*156), 125, 156);
        //declaramos la imagenes que tiene el los nombres de 1 a 10
        matriz[i][j].setIcon(new ImageIcon("C:/Users/julio/Documents/Tareas universidad
tercer semestre/intro a la
progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/"+mat2[i][j]+\".png"));
        //colocamos que la matriz se muestre en pantalla
        matriz[i][j].setVisible(true);
        //aqui se añaden junto con la 0
        panel.add(matriz[i][j],0);

    }

}

seg = 0;
min = 0;

//este lo colocamos para podermo mostrar el tiempo que
//transcurre durante el juego
tiempo = new Timer (1000, new ActionListener()
{

    public void actionPerformed(ActionEvent e)
    {

        seg++;
        if(seg == 60){
            min++;
            seg=0;
        }

    }

});

//declaramos en una variable cronometro el tiempo que transcurre

```

```
});
```

```
//declaramos en la variable espera la cual es otro tiempo el cual lo utilizamos
```

```
//para colocar un tiempo a la hora de que las cartas se voltean
```

```
consegund = 0;
```

```
espera = new Timer (1000, new ActionListener()
```

```
{
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        consegund++;
```

```
    });
```

```
espera.start();
```

```
espera.stop();
```

```
consegund = 0;
```

```
ban=0;
```

```
ban1=0;
```

```
//evento de clic en la cartas
```

```
contador = 0;
```

```
for (int i = 0; i < 4; i++) {
```

```
    for (int j = 0; j < 5; j++) {
```

```
        matriz[i][j].addMouseListener(new MouseAdapter(){
```

```
            public void mousePressed(MouseEvent e){
```

```
                for (int k = 0; k < 4; k++) {
```

```
                    for (int l = 0; l < 5; l++) {
```

```
                        if(e.getSource() == matriz[k][l]){
```

```
                            //cuando se da click a la carta esta se volteara
```

```
                            if(mat2[k][l] == 0 && contador !=2){
```

```
                                mat2[k][l] = mat[k][l];
```

```
                                matriz[k][l].setIcon(new
```

```
ImageIcon("C:/Users/julio/Documents/Tareas universidad tercer semestre/intro a la  
progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/"+mat2[k][l]+".png"));
```

```
                                contador++;
```

```
                                acnum = mat[k][l];
```

```
                                acposx = k;
```

```
                                acposy = l;
```

```
                                if(contador == 1){
```

```
                                    annum = mat[k][l];
```

```
                                    anposx = k;
```

```
                                    anposy = l;
```

```
                                }
```

```
                            //tiempo que se tarda en dar vuelta
```

```
                            espera2 = new Timer (500, new ActionListener()
```

```

{

    public void actionPerformed(ActionEvent e) {

        if(contador == 2 && ban1 == 0){
            espera.restart();
            ban1=1;
        }
        if(contador == 2 && consegund == 2){
            espera.stop();
            consegund = 0;

            //Desaparecen las cartas que son iguales y deja las que aun
no se encuentran

            if(mat2[acposx][acposy]==mat2[anposx][anposy]){

                mat2[acposx][acposy] = -1;
                mat2[anposx][anposy] = -1;
                matriz[acposx][acposy].setIcon(new
Imagelcon("C:/Users/julio/Documents/Tareas universidad tercer semestre/intro a la
progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/"+mat2[acposx][acposy
]+".png"));

                matriz[anposx][anposy].setIcon(new
Imagelcon("C:/Users/julio/Documents/Tareas universidad tercer semestre/intro a la
progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/"+mat2[anposx][anposy
]+".png"));

                contador=0;
                //gano si toda la mat2 es -1
                int acum = 0;
                for (int m = 0; m < 4; m++) {
                    for (int n = 0; n < 5; n++) {
                        if (mat2[m][n] == -1)
                            acum++;
                    }
                }
                //cuando no se encuentre ninguna para entonces aparecera
                //un mensaje diciendo que gano y ejecutando

automaticamente

                //la ventana Recordjugador
                if(acum == 20){
                    JOptionPane.showMessageDialog(panel,
"FELICIDADES GANASTE");

                    RecordJugador ventana = new RecordJugador();
                    ventana.setVisible(true);
                    tiempo.stop();
                    ventana.lbltiempoju.setText(min+": "+seg);
                    ventana.lblnombrejuga.setText(nombreju.getText());

```

```

        ventana.lblhorainicio.setText(lblhora.getText());
        ventana.lblfechaju.setText(lblfecha.getText());

    }

}

for (int m = 0; m < 4; m++) {
    for (int n = 0; n < 5; n++) {
        //se coloca el valor -1 a las cartas pares
        if(mat2[m][n]!=0 && mat2[m][n]!=-1){
            mat2[m][n] = 0;
            matriz[m][n].setIcon(new
Imagelcon("C:/Users/julio/Documents/Tareas universidad tercer semestre/intro a la
progra/Practica2/src/main/java/com/mycompany/practica2/imagenes/"+mat2[m][n]+".png"));
            contador=0;

        }
        System.out.println("salio");

    }

}

}
espera2.stop();
ban1=0;
}
});
if(ban == 0)
    espera2.start();
    ban = 1;
if(contador == 2)
    espera2.restart();
}

}

}

}

});

}

}

componentes();

```

```
    //hilo = new Thread(this);  
}
```

```
//este metodo se hace para obtener aleatoriamente las cartas  
//cada vez que se inicie el juego las cartas apareceran en  
//diferentes lugares.
```

```
//tambien para obtener dos veces la misma carta
```

```
private void numaleatorios(){
```

```
    int acumulador = 0;
```

```
    for (int i = 0; i < 4; i++){
```

```
        for (int j = 0; j < 5; j++){
```

```
            mat[i][j] = 0;
```

```
            // mat2[i][j] = 0;
```

```
        }
```

```
    for (int i = 0; i < 4; i++) {
```

```
        for (int j = 0; j < 5; j++) {
```

```
            mat[i][j] = ran.nextInt(10)+1;
```

```
            do{
```

```
                acumulador = 0;
```

```
                for (int k = 0; k < 4; k++) {
```

```
                    for (int l = 0; l < 5; l++) {
```

```
                        if(mat[i][j]== mat[k][l]){
```

```
                            acumulador +=1;
```

```
                    }
```

```
                }
```

```
            }
```

```
            //esto se coloca para que solo se duplique dos veces la carta
```

```
            if(acumulador == 3){
```

```
                mat[i][j] = ran.nextInt(10)+1;
```

```
            }
```

```
            }while(acumulador == 3);
```

```
        }
```

```
    }
```

```
}
```

```

//estos son los componentes del programa
private void componentes(){
    etiqueta = new JLabel("Pokemon: ");
    etiqueta.setBounds(40,40,150,40);
    etiqueta.setFont(new Font("Lucida Sans Unicode", Font.PLAIN, 15));
    panel.add(etiqueta);

    nombreju = new JLabel();
    nombreju.setBounds(135,40,150,40);
    nombreju.setFont(new Font("Times New Roman", Font.PLAIN, 20));
    panel.add(nombreju);

    //este boton reiniciara el juego
    reiniciar = new JButton("Volver a intentar");
    reiniciar.setBounds(115,560,120,40);
    reiniciar.setFont(new Font("Lucida Sans Unicode", Font.PLAIN, 15));
    reiniciar.addActionListener(this);
    panel.add(reiniciar);

}

//estos son los eventos de accion
@Override
public void actionPerformed(ActionEvent e) {

    //se coloca que el boton reinicia mostrara un mensaje preguntando si esta seguro de
    querea hacerlo
    //si es si, se reiniciara desde el inicio y sino pues seguira en la partida
    if(e.getSource() == reiniciar){
        if (JOptionPane.showConfirmDialog(rootPane, "¿Estas seguro de querer reiniciar el
        juego?\n Al hacer esto lo mandara a menu inico y su trayectoria abra desaparecido.",
        "Reinicio de Juego", JOptionPane.YES_NO_OPTION,
        JOptionPane.ERROR_MESSAGE) == JOptionPane.YES_OPTION)
        {
            menuinicio ventana = new menuinicio();
            ventana.setVisible(true);
            this.setVisible(false);
        }
        else{
            setDefaultCloseOperation(0);
        }
    }
}
}

```

MenúInicio

```
package com.mycompany.practica2;

import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JTextField;

/**
 *
 * @author julio
 */
public class menuinicio extends JFrame implements ActionListener{
    JLabel nomjudador,etiqueta;
    JButton creditos,iniciarjuego,salir;
    JTextField txtnomjugador;

    public menuinicio(){
        this.setTitle("Menu");
        this.setSize(1010, 720);
        this.setLocationRelativeTo(null);
        this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        this.setResizable(false);

        componentes();
    }

    //metodo que contiene todos los compoenentes de la ventana

    public void componentes(){
        JPanel panel = new JPanel();
        panel.setLayout(null);
        this.getContentPane().add(panel);

        etiqueta = new JLabel("Juego de Memoria ");
        etiqueta.setBounds(380,60,250,60);
        etiqueta.setFont(new Font("Berlin Sans FB", Font.PLAIN, 30));
        panel.add(etiqueta);

        etiqueta = new JLabel("Nombre del pokemon ");
        etiqueta.setBounds(400,250,250,40);
```



```
etiqueta.setFont(new Font("Lucida Sans Unicode", Font.PLAIN, 20));
panel.add(etiqueta);
```

```
//texto donde se coloca el nombre de jugador
txtnomjugador = new JTextField();
txtnomjugador.setBounds(300,300,400,40);
txtnomjugador.setHorizontalAlignment(JTextField.CENTER);
txtnomjugador.setFont(new Font("Lucida Sans Unicode", Font.PLAIN, 20));
panel.add(txtnomjugador);
```

```
//boton iniciar juego, este mandara directamente a iniciar el juego
iniciarjuego = new JButton("Iniciar Juego");
iniciarjuego.setBounds(370,400,250,40);
iniciarjuego.setFont(new Font("Lucida Sans Unicode", Font.PLAIN, 15));
iniciarjuego.addActionListener(this);
panel.add(iniciarjuego);
```

```
}
```

```
//eventos de acción
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    if(e.getSource() == iniciarjuego){
```

```
        if(txtnomjugador.getText().equals("")){
```

```
            JOptionPane.showMessageDialog(null, "Escriba el nombre del pokemon que
peleara");
```

```
        }else{
```

```
            JuegoMemoria ventana = new JuegoMemoria();
```

```
            ventana.nombreju.setText(txtnomjugador.getText());
```

```
            ventana.tiempo.start();
```

```
            ventana.setVisible(true);
```

```
this.setVisible(false);
```

```
    }  
}
```

```
    }  
}
```