INDICE

```java
package org.pmoo.packproyecto;

public abstract class Pokemon {
    private String nombre;
    private String tipo;
    private int vida;
    private ListaAtaque listaAtaques=new ListaAtaque();

    public Pokemon(String pNombre,String pTipo, int pVida) {
        this.nombre=pNombre;
        this.tipo=pTipo;
        this.vida=pVida;
        this.listaAtaques=crearListaAtaques();
    }

    public void imprimir() {
        System.out.println("\nNOMBRE : " + this.nombre + ", VIDA : " + this.vida +", TIPO : " + this.tipo);
        this.listaAtaques.imprimir();
    }

    private int usarAtaque(int pPosAtaqueEnLista) {
        return this.listaAtaques.usarAtaque(pPosAtaqueEnLista);
    }

    public void cambiarVida(int pDaño) {
        this.vida=this.vida-pDaño;
    }

    public boolean muerto() {
        boolean muerto=false;
        if(this.vida<=0) {
            muerto=true;
        }
        return muerto;
    }

    public void atacar(Pokemon pPK) throws CambiarDePokemonException,UsarCuraException{
        int num=0;
        boolean repetir=true;
        int daño=0;
        while(num==0||repetir) {
            num=Teclado.getTeclado().leerNumero(6);
            if(num==5) {
                throw(new CambiarDePokemonException());
            }
            else if(num==6) {
                throw(new UsarCuraException());
            }
            daño=this.usarAtaque(num-1);
            if((num<1&&num>4)||daño==-100000) {
                System.out.println("ERROR: Su respuesta no es la esperada.");
                System.out.println("¡VUELVE A INTENTARLO!");
            }
```

```
54              else{
55                  repetir=false;
56              }
57          }
58          if(this.supereficaz(pPK)) {
59              daño=daño+(daño/5);
60              System.out.println("Tu ataque es SUPEREFICAZ");
61          }
62          else if(this.pocoeficaz(pPK)) {
63              daño=daño-(daño/5);
64              System.out.println("Tu ataque es POCO EFICAZ");
65          }
66          pPK.cambiarVida(daño);
67      }
68
69      protected abstract ListaAtaque crearListaAtaques();
70
71      protected abstract boolean supereficaz(Pokemon pPK);
72
73      protected abstract boolean pocoeficaz(Pokemon pPK);
74
75      public void imprimirSoloInfoPK() {
76          System.out.println("                          NOMBRE : " + this.nombre + ", VIDA : " + this.vida +", TIPO : " + this.tipo);
77      }
78
79      public int getVida() {
80          return this.vida;
81      }
82  }
```

```java
package org.pmoo.packproyecto;

import static org.junit.Assert.*;

public class PokemonTest {

    private Agua pk1;
    private Fuego pk2;
    private Planta pk3;
    @Before
    public void setUp() throws Exception {
        pk1 = new Agua("Pk1", "Agua", 10);
        pk2 = new Fuego("Pk2", "Fuego", 3);
        pk3 = new Planta("Pk3","Planta", 11);
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void testPokemon() {
        assertNotNull(pk1);
    }

    @Test
    public void testCambiarVida() {
        pk1.cambiarVida(5);
        assertFalse(pk1.muerto());
    }

    @Test
    public void testMuerto() {
        pk1.cambiarVida(10);
        assertTrue(pk1.muerto());
    }
}
```

```java
package org.pmoo.packproyecto;

public class Agua extends Pokemon{
    public Agua(String pNombre,String pTipo, int pVida) {
        super(pNombre,pTipo,pVida);
    }

    protected ListaAtaque crearListaAtaques() {
        ListaAtaque lista=new ListaAtaque();
        Ataque atq1=new Ataque("Tsunami", "Agua", 25, 3);
        lista.añadirAtaque(atq1);
        Ataque atq2=new Ataque("Manguerazo", "Agua", 5, 20);
        lista.añadirAtaque(atq2);
        Ataque atq3=new Ataque("Remolino", "Agua", 10, 15);
        lista.añadirAtaque(atq3);
        Ataque atq4=new Ataque("Maremoto", "Agua", 20, 8);
        lista.añadirAtaque(atq4);
        return lista;
    }

    @Override
    protected boolean supereficaz(Pokemon pPK) {
        boolean supereficaz=false;
        if(pPK instanceof Fuego) {
            supereficaz=true;
        }
        return supereficaz;
    }

    @Override
    protected boolean pocoeficaz(Pokemon pPK) {
        boolean pocoeficaz=false;
        if(pPK instanceof Planta) {
            pocoeficaz=true;
        }
        return pocoeficaz;
    }
}
```

```java
package org.pmoo.packproyecto;

import static org.junit.Assert.*;

public class AguaTest {
    Agua p1;
    Fuego p2;
    Planta p3;

    @Before
    public void setUp() throws Exception {
        p1= new Agua("Squirtle","Agua",100);
        p2= new Fuego ("Charmander","Fuego",100);
        p3= new Planta ("Bulbasur","Planta",100);
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void testAgua() {
        assertNotNull(p1);
    }

    @Test
    public void superEficazTest() {
        assertTrue(p1.supereficaz(p2));
        assertFalse(p1.supereficaz(p3));
    }

    @Test
    public void pocoEficazTest() {
        assertFalse(p1.pocoeficaz(p2));
        assertTrue(p1.pocoeficaz(p3));
    }
}
```

```java
package org.pmoo.packproyecto;

public class Fuego extends Pokemon{
    public Fuego(String pNombre,String pTipo, int pVida) {
        super(pNombre,pTipo,pVida);
    }

    protected ListaAtaque crearListaAtaques() {
        ListaAtaque lista=new ListaAtaque();
        Ataque atq1=new Ataque("Ascuas", "Fuego", 10, 15);
        lista.añadirAtaque(atq1);
        Ataque atq2=new Ataque("Llamarada", "Fuego", 20, 8);
        lista.añadirAtaque(atq2);
        Ataque atq3=new Ataque("Bomba termica", "Fuego", 25, 3);
        lista.añadirAtaque(atq3);
        Ataque atq4=new Ataque("Antibomberos", "Fuego", 5, 20);
        lista.añadirAtaque(atq4);
        return lista;
    }

    @Override
    protected boolean supereficaz(Pokemon pPK) {
        boolean supereficaz=false;
        if(pPK instanceof Planta) {
            supereficaz=true;
        }
        return supereficaz;
    }

    @Override
    protected boolean pocoeficaz(Pokemon pPK) {
        boolean pocoeficaz=false;
        if(pPK instanceof Agua) {
            pocoeficaz=true;
        }
        return pocoeficaz;
    }
}
```

```java
package org.pmoo.packproyecto;

import static org.junit.Assert.*;

public class FuegoTest {
    Fuego p1;
    Agua p2;
    Planta p3;

    @Before
    public void setUp() throws Exception {
        p1= new Fuego ("Charmander","Fuego",100);
        p2= new Agua("Squirtle","Agua",100);
        p3 = new Planta ("Bulbasur","Planta",100);
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void testFuego() {
        assertNotNull(p1);
    }

    @Test
    public void superEficazTest() {
        assertTrue(p1.supereficaz(p3));
        assertFalse(p2.supereficaz(p2));
    }

    @Test
    public void pocoEficazTest() {
        assertFalse(p1.pocoeficaz(p3));
        assertTrue(p1.pocoeficaz(p2));
    }
}
```

```java
package org.pmoo.packproyecto;

public class Planta extends Pokemon{
    public Planta(String pNombre,String pTipo, int pVida) {
        super(pNombre,pTipo,pVida);
    }

    protected ListaAtaque crearListaAtaques() {
        ListaAtaque lista=new ListaAtaque();
        Ataque atq1=new Ataque("Ortiga Salvaje", "Planta", 20, 8);
        lista.añadirAtaque(atq1);
        Ataque atq2=new Ataque("Fotosintesis", "Planta", 10, 15);
        lista.añadirAtaque(atq2);
        Ataque atq3=new Ataque("Espora", "Planta", 5, 20);
        lista.añadirAtaque(atq3);
        Ataque atq4=new Ataque("Abrazo de Cactus", "Planta", 25, 3);
        lista.añadirAtaque(atq4);
        return lista;
    }

    @Override
    protected boolean supereficaz(Pokemon pPK) {
        boolean supereficaz=false;
        if(pPK instanceof Agua) {
            supereficaz=true;
        }
        return supereficaz;
    }

    @Override
    protected boolean pocoeficaz(Pokemon pPK) {
        boolean pocoeficaz=false;
        if(pPK instanceof Fuego) {
            pocoeficaz=true;
        }
        return pocoeficaz;
    }
}
```

```java
package org.pmoo.packproyecto;

import static org.junit.Assert.*;

public class PlantaTest {
    Planta p1;
    Fuego p2;
    Agua p3;

    @Before
    public void setUp() throws Exception {
        p1 = new Planta ("Bulbasur","Planta",100);
        p2=new Fuego ("Charmander","Fuego",100);
        p3=new Agua("Squirtle","Agua",100);
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void testPlanta() {
        assertNotNull(p1);
    }

    @Test
    public void superEficazTest() {
        assertTrue(p1.supereficaz(p3));
        assertFalse(p1.supereficaz(p2));
    }

    @Test
    public void pocoEficazTest() {
        assertFalse(p1.pocoeficaz(p3));
        assertTrue(p1.pocoeficaz(p2));
    }
}
```

```java
package org.pmoo.packproyecto;

import java.util.ArrayList;

public class ListaPokemon {
    private ArrayList<Pokemon> listaPokemon;

    public ListaPokemon() {
        this.listaPokemon=new ArrayList<Pokemon>();
    }

    private Iterator<Pokemon> getIterador(){
        return this.listaPokemon.iterator();
    }

    public void imprimir() {
        Iterator<Pokemon> itr=this.getIterador();
        Pokemon pokemon;
        int pos=1;
        while(itr.hasNext()) {
            pokemon=itr.next();
            System.out.println("\n"+pos+". ");
            pokemon.imprimir();

            pos++;
        }
    }

    public Pokemon sacarPokemon(int pPosPK) throws Exception{
        Pokemon pk=null;
        pk=this.listaPokemon.get(pPosPK);
        return pk;
    }

    public boolean listaVacia() {
        boolean vacia=false;
        if(this.listaPokemon.size()==0) {
            vacia=true;
        }
        return vacia;
    }

    public void eliminarPokemon(Pokemon pPK) {
        this.listaPokemon.remove(pPK);
    }

    public void añadirPokemon(Pokemon pPK) {
        this.listaPokemon.add(pPK);
    }
}
```

```java
import static org.junit.Assert.*;

public class ListaPokemonTest {
    private Agua pk1;
    private Fuego pk2;
    private Planta pk3;
    private ListaPokemon lp;
    private ListaPokemon lp2;
    @Before
    public void setUp() throws Exception {
        lp = new ListaPokemon();
        lp2 = new ListaPokemon();
        pk1 = new Agua("Pk1", "Agua", 10);
        pk2 = new Fuego("Pk2", "Fuego", 3);
        pk3 = new Planta("Pk3","Planta", 11);
        lp.añadirPokemon(pk1);
        lp.añadirPokemon(pk2);
        lp.añadirPokemon(pk3);
    }

    @After
    public void tearDown() throws Exception {
    }

    @Test
    public void testListaPokemon() {
        assertNotNull(lp);
    }

    @Test
    public void testSacarPokemon() {
        try {
            Pokemon p = lp.sacarPokemon(3);
            assertEquals(p,pk1);
        } catch (Exception e) {}
    }

    @Test
    public void testListaVacia() {
        assertFalse(lp.listaVacia());
        assertTrue(lp2.listaVacia());
    }

    @Test
    public void testEliminarPokemon() {
        lp.eliminarPokemon(pk2);
        Pokemon p;
        try {
            p = lp.sacarPokemon(4);
            assertNotEquals(p,pk2);
        } catch (Exception e) {}
    }

    @Test
    public void testAñadirPokemon() {
        Agua pk4 = new Agua("PK4", "Agua", 8);
        lp.añadirPokemon(pk4);
        Pokemon p;
        try {
            p = lp.sacarPokemon(6);
            assertEquals(p,pk4);
        } catch (Exception e) {}
    }
}
```

```java
package org.pmoo.packproyecto;

public class ListaPokemonAElegir {

    private ListaPokemon listaPAE;
    private static ListaPokemonAElegir miLPAE=new ListaPokemonAElegir();

    private ListaPokemonAElegir() {
        this.listaPAE = new ListaPokemon();
        this.crearListaPokemons();
    }

    public static ListaPokemonAElegir getListaPokemonAElegir() {
        return miLPAE;
    }

    private void crearListaPokemons() {
        Fuego Pokemon1 = new Fuego ("Typhlosion", "Fuego", 110);
        this.listaPAE.añadirPokemon(Pokemon1);
        Fuego Pokemon2 = new Fuego ("Infernia", "Fuego", 110);
        this.listaPAE.añadirPokemon(Pokemon2);
        Planta Pokemon3 = new Planta ("Venusaur", "Planta", 90);
        this.listaPAE.añadirPokemon(Pokemon3);
        Planta Pokemon4 = new Planta ("Sylvanis", "Planta", 90);
        this.listaPAE.añadirPokemon(Pokemon4);
        Agua Pokemon5 = new Agua ("Greninja", "Agua", 100);
        this.listaPAE.añadirPokemon(Pokemon5);
        Agua Pokemon6 = new Agua ("Marinna", "Agua", 100);
        this.listaPAE.añadirPokemon(Pokemon6);
        Fuego Pokemon7 = new Fuego ("Charizard", "Fuego", 100);
        this.listaPAE.añadirPokemon(Pokemon7);
        Fuego Pokemon8 = new Fuego ("Flamaris", "Fuego", 100);
        this.listaPAE.añadirPokemon(Pokemon8);
        Planta Pokemon9 = new Planta ("Sceptile", "Planta", 110);
        this.listaPAE.añadirPokemon(Pokemon9);
        Planta Pokemon10 = new Planta ("Petalina", "Planta", 110);
        this.listaPAE.añadirPokemon(Pokemon10);
        Agua Pokemon11 = new Agua ("Gyarados", "Agua", 90);
        this.listaPAE.añadirPokemon(Pokemon11);
        Agua Pokemon12 = new Agua ("Nereidra", "Agua", 90);
        this.listaPAE.añadirPokemon(Pokemon12);
    }

    public void imprimir() {
        this.listaPAE.imprimir();
    }

    public void eliminarPokemon(Pokemon pPK) {
        this.listaPAE.eliminarPokemon(pPK);
    }

    public Pokemon sacarPokemon(int pPosPK) throws Exception {
        return this.listaPAE.sacarPokemon(pPosPK-1);
    }
}
```

```java
package org.pmoo.packproyecto;

public class Jugador {
    private String nombre;
    private int numCuras=3;
    private ListaPokemon listaPokemon;

    public Jugador(String pNombre) {
        this.nombre=pNombre;
        this.listaPokemon=new ListaPokemon();
    }

    public Pokemon sacarPokemon(int pPosPK) throws Exception{
        return this.listaPokemon.sacarPokemon(pPosPK-1);
    }

    public boolean listaVacia() {
        return this.listaPokemon.listaVacia();
    }

    public void imprimirLista() {
        this.listaPokemon.imprimir();
    }

    public String getNombre() {
        return this.nombre;
    }

    public boolean usarCura() {
        boolean agotadas=false;
        if(this.numCuras<=0) {
            agotadas=true;
            System.out.println("No te quedan curas.");
        }
        else {
            this.numCuras--;
        }
        return agotadas;
    }

    public void imprimirNumCuras() {
        System.out.println("\n6.   USAR CURA (Te quedan "+this.numCuras+" curas).");
    }

    public void eliminarPokemon(Pokemon pPK) {
        this.listaPokemon.eliminarPokemon(pPK);
    }

    public void añadirPokemon(Pokemon pPK) {
        this.listaPokemon.añadirPokemon(pPK);
    }

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

```java
import static org.junit.Assert.*;

public class JugadorTest {

    private Jugador j1;
    private Agua pk1;
    private Fuego pk2;
    private Planta pk3;

    @Before
    public void setUp() throws Exception {
        j1 = new Jugador("Jugador1");
        pk1 = new Agua("PK1", "Agua", 10);
        pk2 = new Fuego("PK2", "fuego", 3);
        pk3 = new Planta("PK3", "Planta", 11);
        j1.añadirPokemon(pk1);
        j1.añadirPokemon(pk2);
        j1.añadirPokemon(pk3);}
    @After
    public void tearDown() throws Exception {}

    @Test
    public void testSacarPokemon() {
        Pokemon p;
        try {
            p = j1.sacarPokemon(5);
            assertEquals(p,pk2);
        } catch (Exception e) {}
        assertThrows(Exception.class, ()->{j1.sacarPokemon(-1);});}

    @Test
    public void testListaVacia() {
        assertFalse(j1.listaVacia());}

    @Test
    public void testGetNombre() {
        assertEquals("Jugador1", j1.getNombre());}

    @Test
    public void testEliminarPokemon() {
        j1.eliminarPokemon(pk2);
        Pokemon p;
        try {
            p = j1.sacarPokemon(5);
            assertNotEquals(p,pk2);
        } catch (Exception e) {}}

    @Test
    public void testAñadirPokemon() {
        Agua pk4 = new Agua("PK4", "Agua", 8);
        j1.añadirPokemon(pk4);
        Pokemon p;
        try {
            p = j1.sacarPokemon(7);
            assertEquals(p,pk4);
        }
        catch (Exception e) {}}

    @Test
    public void testSetNombre() {
        assertEquals("Jugador1", j1.getNombre());
        j1.setNombre("J1");
        assertEquals("J1", j1.getNombre());
    }
}
```

```java
package org.pmoo.packproyecto;

public class ListaJugador {
    private Jugador jugador1;
    private Jugador jugador2;
    private static ListaJugador miLJ=new ListaJugador();

    private ListaJugador() {
        this.jugador1=new Jugador(null);
        this.jugador2=new Jugador(null);
    }

    public static ListaJugador getListaJugador() {
        return miLJ;
    }

    public void imprimir(int pNumJugador) {
        if(pNumJugador==1) {
            this.jugador1.imprimirLista();
        }
        else if(pNumJugador==2) {
            this.jugador2.imprimirLista();
        }
    }

    public boolean listaVacia(int pPosJugador) {
        boolean vacia=false;
        if(pPosJugador==1&&this.jugador1.listaVacia()) {
            vacia=true;
        }
        else if(pPosJugador==2&&this.jugador2.listaVacia()) {
            vacia=true;
        }
        return vacia;
    }

    public String getNombre(int pNumJugador) {
        String nombre=null;
        if(pNumJugador==1) {
            nombre=this.jugador1.getNombre();
        }
        else if(pNumJugador==2) {
            nombre=this.jugador2.getNombre();
        }
        return nombre;
    }

    public Pokemon sacarPokemon(int pNumJugador, int pPosPK)throws Exception{
        Pokemon pokemon=null;
        if(pNumJugador==1) {
            pokemon=this.jugador1.sacarPokemon(pPosPK);
        }
        else if(pNumJugador==2) {
            pokemon=this.jugador2.sacarPokemon(pPosPK);
        }
        return pokemon;
    }

    public void añadirPokemon(int pNumJugador, Pokemon pPK) {
        if(pNumJugador==1) {
            this.jugador1.añadirPokemon(pPK);
        }
        else if(pNumJugador==2) {
            this.jugador2.añadirPokemon(pPK);
        }
```

```
66      }
67
68      public void eliminarPokemon(int pNumJugador, Pokemon pPK) {
69          if(pNumJugador==1) {
70              this.jugador1.eliminarPokemon(pPK);
71          }
72          else if(pNumJugador==2) {
73              this.jugador2.eliminarPokemon(pPK);
74          }
75      }
76
77      public void setNombres(String pNomJugador1,String pNomJugador2) {
78          this.jugador1.setNombre(pNomJugador1);
79          this.jugador2.setNombre(pNomJugador2);
80      }
81
82      public boolean usarCura(int pNumJugador) {
83          boolean agotada=false;
84          if(pNumJugador==1) {
85              agotada=this.jugador1.usarCura();
86          }
87          else if(pNumJugador==2) {
88              agotada=this.jugador2.usarCura();
89          }
90          return agotada;
91      }
92
93      public void imprimirNumCuras(int pNumJugador) {
94          if(pNumJugador==1) {
95              this.jugador1.imprimirNumCuras();
96          }
97          else if(pNumJugador==2) {
98              this.jugador2.imprimirNumCuras();
99          }
100     }
101 }
```

```java
public class ListaJugadorTest {
    Agua pk1,pk4;
    Fuego pk2,pk5;
    Planta pk3,pk6;
    @Before
    public void setUp() throws Exception {
         ListaJugador.getListaJugador().setNombres("Jugador1","Jugador2");
         pk1=new Agua("PK1", "Agua", 10);
         pk2=new Fuego("PK2", "Fuego", 3);
         pk3=new Planta("PK3", "Planta", 11);
         ListaJugador.getListaJugador().añadirPokemon(1, pk1);
         ListaJugador.getListaJugador().añadirPokemon(1, pk2);
         ListaJugador.getListaJugador().añadirPokemon(1, pk3);
         pk4=new Agua("PK4", "Agua", 10);
         pk5=new Fuego("PK5", "Fuego", 3);
         pk6=new Planta("PK6", "Planta", 11);
         ListaJugador.getListaJugador().añadirPokemon(2, pk4);
         ListaJugador.getListaJugador().añadirPokemon(2, pk5);
         ListaJugador.getListaJugador().añadirPokemon(2, pk6);
    }
    @After
    public void tearDown() throws Exception {}
    @Test
    public void testListaVacia() {
        assertFalse(ListaJugador.getListaJugador().listaVacia(1));
        assertFalse(ListaJugador.getListaJugador().listaVacia(2));}

    @Test
    public void testAñadirPokemon() {
        try {
            assertEquals(ListaJugador.getListaJugador().sacarPokemon(1,1),pk1);
        } catch (Exception e) {}}

    @Test
    public void testGetNombre() {
        assertEquals("Jugador1", ListaJugador.getListaJugador().getNombre(1));
        assertEquals("Jugador2", ListaJugador.getListaJugador().getNombre(2));}

    @Test
    public void testSacarPokemon() {
        try {
            assertNotNull(ListaJugador.getListaJugador().sacarPokemon(1, 1));
        } catch (Exception e) {}
        try {
            assertNotNull(ListaJugador.getListaJugador().sacarPokemon(2, 1));
        } catch (Exception e) {}}

    @Test
    public void testEliminarPokemon() {
        ListaJugador.getListaJugador().eliminarPokemon(1, pk2);
        Pokemon p;
        try {
            p = ListaJugador.getListaJugador().sacarPokemon(1, 2);
            assertNotEquals(p,pk2);
        } catch (Exception e) {}}

    @Test
    public void testSetNombres() {
        assertEquals("Jugador1", ListaJugador.getListaJugador().getNombre(1));
        assertEquals("Jugador2", ListaJugador.getListaJugador().getNombre(2));
        assertNotEquals("Jug1", ListaJugador.getListaJugador().getNombre(1));
        assertNotEquals("Jug2", ListaJugador.getListaJugador().getNombre(2));
        }
}
```

```java
package org.pmoo.packproyecto;

public class Ataque {
    private String nombre;
    private String tipo;
    private int daño;
    private int usos;

    public Ataque(String pNombre,String pTipo,int pDaño,int pUsos) {
        this.nombre=pNombre;
        this.tipo=pTipo;
        this.daño=pDaño;
        this.usos=pUsos;
    }

    public void imprimir(int pPosAtaqueEnLista) {
        System.out.println(pPosAtaqueEnLista + ".   NOMBRE : " + this.nombre + ", TIPO : " + this.tipo + ", DAÑO : " + this.daño + ", USOS : " + this.usos);
    }

    public int getDaño() {
        return this.daño;
    }

    public void disminuirUso() {
        this.usos--;
    }

    public boolean ataqueGastado() {
        boolean gastado=false;
        if(this.usos<=0) {
            gastado=true;
        }
        return gastado;
    }
}
```

```java
package org.pmoo.packproyecto;

import static org.junit.Assert.*;

public class AtaqueTest {

    Ataque a1,a2,a3;

    @Before
    public void setUp() throws Exception
    {
         a1= new Ataque("Llama", "Fuego", 50, 4);
         a2= new Ataque("Follaje", "Planta", 50, 1);
         a3= new Ataque("Chorro", "Agua", 50, 0);
    }

    @After
    public void tearDown() throws Exception {
        a1 = null;
        a2 = null;
        a3 = null;
    }

    @Test
    public void testAtaque() {
        assertNotNull(a1);
    }

    @Test
    public void testGetDaño() {
        a1= new Ataque("Llama", "Fuego", 50, 4);
        assertEquals(50,a1.getDaño());
    }

    @Test
    public void testDisminuirUso() {
        a2.disminuirUso();
        assertTrue(a2.ataqueGastado());
    }

    @Test
    public void testAtaqueGastado() {
        assertFalse(a1.ataqueGastado());
        assertTrue(a3.ataqueGastado());
    }

}
```

```java
package org.pmoo.packproyecto;

import java.util.ArrayList;

public class ListaAtaque {
    private ArrayList<Ataque> listaAtaque;

    public ListaAtaque() {
        this.listaAtaque=new ArrayList<Ataque>();
    }

    private Iterator<Ataque> getIterador(){
        return this.listaAtaque.iterator();
    }

    public void imprimir() {
        Iterator<Ataque> itr=this.getIterador();
        Ataque ataque;
        int posEnLista=1;
        System.out.println("\nLos ataques de tu Pokemon son :");
        while(itr.hasNext()) {
            ataque=itr.next();
            ataque.imprimir(posEnLista);
            posEnLista++;
        }
    }

    public int usarAtaque(int pPosAtaqueEnLista) {
        Ataque ataque=this.listaAtaque.get(pPosAtaqueEnLista);
        int daño=-100000;
        if(!ataque.ataqueGastado()&&ataque!=null) {
            daño=ataque.getDaño();
            ataque.disminuirUso();
        }
        else{
            System.out.println("\nSe han agotado los usos de este ataque.");
        }
        return daño;
    }

    public void añadirAtaque(Ataque pAtaque) {
        this.listaAtaque.add(pAtaque);
    }
}
```

```java
package org.pmoo.packproyecto;

import static org.junit.Assert.*;

public class ListaAtaqueTest {

    private ListaAtaque l1;

    @Before
    public void setUp() {
        l1 = new ListaAtaque();
    }

    @After
    public void tearDown() {
        l1 = null;
    }
    @Test
    public void testListaAtaque() {
        assertNotNull(l1);
    }

    @Test
    public void testUsarAtaque() {
        Ataque a = new Ataque("PRUEBA", "TIPO1", 100, 2);
        l1.añadirAtaque(a);
        assertEquals(100, l1.usarAtaque(0));
        assertEquals(100, l1.usarAtaque(0));
        assertEquals(-100000, l1.usarAtaque(0));
    }
    public void testAñadirAtaque(Ataque pAtaque) {
        Ataque a1 = new Ataque("Pepe", "Agua", 50, 1);
        l1.añadirAtaque(a1);
        assertEquals(50, l1.usarAtaque(0));

    }
}
```

```java
package org.pmoo.packproyecto;
import java.util.Scanner;

public class Teclado {
    private static Teclado miTEC=new Teclado();
    private Scanner sc;
    private Teclado() {
        sc=new Scanner(System.in);}
    public static Teclado getTeclado() {
        return miTEC;
    }
    public String leerString() {
        return sc.next();
    }
    public int leerNumero() {
        int num=0;
        boolean seguir=true;
        do {
            try {
                seguir=false;
                num=sc.nextInt();
            }
            catch(Exception e) {
                sc.next();
                seguir=true;
            }}while(seguir);
        return num;}

    public int leerNumero(int pMax) {
        int num=0;
        boolean seguir=true;
        do {
            try {
                seguir=false;
                num=sc.nextInt();
                while(num>pMax||num<0) {
                    System.out.println("ERROR: Su respuesta no es el esperada.");
                    num=sc.nextInt();
                }}
            catch(Exception e) {
                sc.next();
                seguir=true;
            }} while(seguir);
        return num;
    }
    public boolean leerSiNo() {
        boolean sino=false;
        boolean hecho=false;
        String respuesta;
        while(!hecho) {
            respuesta=sc.next().toLowerCase();
            if(respuesta.equals("sí") || respuesta.equals("si")) {
                sino=true;
                hecho=true;}
            else if(respuesta.equals("no")){
                sino=false;
                hecho=true;}
            else {
                System.out.println("ERROR: Su respuesta no es la esperada");
            }}
        return sino;
    }
}
```

```java
package org.pmoo.packproyecto;

import java.util.Random;

public class Tablero {
    private Pokemon pokemon1;
    private Pokemon pokemon2;
    private static Tablero miTAB=new Tablero();

    private Tablero() {}

    public static Tablero getTablero() {
        return miTAB;
    }

    public void empezarPartida() {
        System.out.println("¿Quieres empezar una partida? (Escribe 'Si' o 'No')");
        boolean respuesta=Teclado.getTeclado().leerSiNo();
        if(respuesta) {
            partida();
        }
        else {
            System.out.println("¡¡Esperamos que vuelvas pronto!!");
        }
    }

    private void partida() {
        System.out.println("COMIENZA LA PARTIDA");
        crearJugadores();
        try {
            this.pokemon1=ListaJugador.getListaJugador().sacarPokemon(1,this.generarNumRandom(3));
            this.pokemon2=ListaJugador.getListaJugador().sacarPokemon(2,this.generarNumRandom(3));
        }
        catch(Exception e) {}
        while(!ListaJugador.getListaJugador().listaVacia(1)&&!ListaJugador.getListaJugador().listaVacia(2)) {
            this.atacar();
        }
        if(ListaJugador.getListaJugador().listaVacia(1)) {
            System.out.println("EL GANADOR ES "+ListaJugador.getListaJugador().getNombre(2));
        }
        else if(ListaJugador.getListaJugador().listaVacia(2)) {
            System.out.println("EL GANADOR ES "+ListaJugador.getListaJugador().getNombre(1));
        }
    }

    private void crearJugadores() {
        System.out.println("\nEscribe el nombre del Jugador1");
        String nombre1=Teclado.getTeclado().leerString();
        System.out.println("Escribe el nombre del Jugador2");
        String nombre2=Teclado.getTeclado().leerString();
        ListaJugador.getListaJugador().setNombres(nombre1, nombre2);
        this.elegirListaPokemon();
    }

    private void elegirListaPokemon() {
        System.out.println("ELIGE LOS TRES POKEMON QUE QUIERAS UTILIZAR.");
        int terminar=0;
        while(terminar<3) {
            ListaPokemonAElegir.getListaPokemonAElegir().imprimir();
            this.elegirPokemon(1);
            ListaPokemonAElegir.getListaPokemonAElegir().imprimir();
            this.elegirPokemon(2);
            terminar++;
        }
    }
```

```
    private void elegirPokemon(int pPosJug) {
        System.out.println("\nTE TOCA ELEGIR "+ListaJugador.getListaJugador().getNombre(pPosJug));
        int num=Teclado.getTeclado().leerNumero(12);
        try {
            Pokemon pk=ListaPokemonAElegir.getListaPokemonAElegir().sacarPokemon(num);
            ListaPokemonAElegir.getListaPokemonAElegir().eliminarPokemon(pk);
            ListaJugador.getListaJugador().añadirPokemon(pPosJug, pk);
        }
        catch(Exception e) {
            System.out.println("ERROR: Su repuesta no es la esperada.");
            this.elegirPokemon(pPosJug);
        }
    }

    private int generarNumRandom(int pMax) {
        Random r=new Random();
        return r.nextInt(pMax)+1;
    }
```

```
private void atacar() {
    try {
        System.out.println("\n¿Qué ataque quieres usar "+ListaJugador.getListaJugador().getNombre(1)+"?");
        System.out.println("\n                                                CONTRINCANTE");
        this.pokemon2.imprimirSoloInfoPK();
        this.pokemon1.imprimir();
        System.out.println("\n5.   CAMBIAR DE POKEMON");
        ListaJugador.getListaJugador().imprimirNumCuras(1);
        this.pokemon1.atacar(pokemon2);
        if(this.pokemon2.muerto()) {
            ListaJugador.getListaJugador().eliminarPokemon(2, this.pokemon2);
            this.pokemon2=null;
            while(this.pokemon2==null&&!ListaJugador.getListaJugador().listaVacia(1)&&!ListaJugador.getListaJugador().listaVacia(2)) {
                this.muerto(2);
            }
        }
    }
    catch(CambiarDePokemonException cdpe){
        this.cambiarPokemon(1);
    }
    catch(UsarCuraException uce) {
        if(!ListaJugador.getListaJugador().usarCura(1)) {
            this.pokemon1.cambiarVida(-15);
        }
    }
    try {
        if(!ListaJugador.getListaJugador().listaVacia(1)&&!ListaJugador.getListaJugador().listaVacia(2)) {
            System.out.println("\n¿Qué ataque quieres usar "+ListaJugador.getListaJugador().getNombre(2)+"?");
            System.out.println("\n                                                CONTRINCANTE");
            this.pokemon1.imprimirSoloInfoPK();
            this.pokemon2.imprimir();
            System.out.println("\n5.   CAMBIAR DE POKEMON");
            ListaJugador.getListaJugador().imprimirNumCuras(2);
            this.pokemon2.atacar(pokemon1);
            if(this.pokemon1.muerto()&&!ListaJugador.getListaJugador().listaVacia(1)&&!ListaJugador.getListaJugador().listaVacia(2)) {
                ListaJugador.getListaJugador().eliminarPokemon(1, this.pokemon1);
                this.pokemon1=null;
                while(this.pokemon1==null&&!ListaJugador.getListaJugador().listaVacia(1)&&!ListaJugador.getListaJugador().listaVacia(2)) {
                    this.muerto(1);
                }
            }
        }
    }
    catch(CambiarDePokemonException cdpe) {
        this.cambiarPokemon(2);
    }
    catch(UsarCuraException uce) {
        if(!ListaJugador.getListaJugador().usarCura(2)) {
            this.pokemon2.cambiarVida(-15);
        }
    }
}
```

```
139    private void muerto(int pPosJugador) {
140        System.out.println("Ha muerto tu pokemon " + ListaJugador.getListaJugador().getNombre(pPosJugador) + " ¿Que pokemon quieres sacar ahora?");
141        ListaJugador.getListaJugador().imprimir(pPosJugador);
142        boolean parar=false;
143        while(!parar) {
144            int num=Teclado.getTeclado().leerNumero();
145            if(pPosJugador==1) {
146                try {
147                    this.pokemon1=ListaJugador.getListaJugador().sacarPokemon(pPosJugador,num);
148                    parar=true;
149                }
150                catch(Exception e) {
151                    System.out.println("ERROR: Su repuesta no es la esperada.");
152                }
153            }
154            else if(pPosJugador==2) {
155                try {
156                    this.pokemon2=ListaJugador.getListaJugador().sacarPokemon(pPosJugador,num);
157                    parar=true;
158                }
159                catch(Exception e) {
160                    System.out.println("ERROR: Su respuesta no es la esperada.");
161                }
162            }
163        }
164    }
165
166     private void cambiarPokemon(int pPosJugador) {
167         System.out.println("¿Qué pokemon de tu lista quieres sacar?");
168         ListaJugador.getListaJugador().imprimir(pPosJugador);
169         boolean parar=false;
170         while(!parar){
171             try {
172                 int num=Teclado.getTeclado().leerNumero(3);
173                 if(pPosJugador==1){
174                     this.pokemon1=ListaJugador.getListaJugador().sacarPokemon(1,num);
175                     parar=true;
176                 }
177                 else if(pPosJugador==2) {
178                     this.pokemon2=ListaJugador.getListaJugador().sacarPokemon(2,num);
179                     parar=true;
180                 }
181             }
182             catch(Exception e) {
183                 System.out.println("ERROR: Su respuesta no es la esperada.");
184             }
185         }
186     }
187
188     public static void main(String[]args) {
189         miTAB.empezarPartida();
190     }
191 }
```

```java
package org.pmoo.packproyecto;

public class CambiarDePokemonException extends Exception{

    public CambiarDePokemonException() {
        super();
    }
}
```

```java
package org.pmoo.packproyecto;

public class UsarCuraException extends Exception{
    public UsarCuraException() {
        super();
    }
}
```