

Evaluación de Aprendizaje - Kek Iván - Tecnicatura en desarrollo web

```
package ar.edu.unlam.pb2.ea1;
```

```
import java.util.ArrayList;
```

```
public class Diabetico extends Paciente {
```

```
    private Integer azucarEnSangre;
```

```
    public Diabetico(String nombre, String apellido, Integer dni, Integer azucarEnSangre) {
```

```
        super(nombre, apellido, dni);
```

```
        this.azucarEnSangre = azucarEnSangre;
```

```
    }
```

```
    public Integer getAzucarEnSangre() {
```

```
        return azucarEnSangre;
```

```
    }
```

```
    public void setAzucarEnSangre(Integer azucarEnSangre) {
```

```
        this.azucarEnSangre = azucarEnSangre;
```

```
    }
```

```
    @Override
```

```
    public void agregarDietaDiaria(DietaDiaria dieta) {
```

```
        if (dieta.aptaDiabetico() == true) {
```

```
            getIngesta().add(dieta);
```

```
        }
```

```
    }  
}
```

```
package ar.edu.unlam.pb2.ea1;
```

```
public class Hipertenso extends Paciente {
```

```
    private Integer presionMinima;
```

```
    private Integer presionMaxima;
```

```
    public Hipertenso(String nombre, String apellido, Integer dni, Integer presionMinima,  
Integer presionMaxima) {
```

```
        super(nombre, apellido, dni);
```

```
        this.presionMinima = presionMinima;
```

```
        this.presionMaxima = presionMaxima;
```

```
    }
```

```
    public Integer getPresionMinima() {
```

```
        return presionMinima;
```

```
    }
```

```
    public void setPresionMinima(Integer presionMinima) {
```

```
        this.presionMinima = presionMinima;
```

```
    }
```

```
    public Integer getPresionMaxima() {
```

```
        return presionMaxima;
```

```
    }
```

```

        public void setPresionMaxima(Integer presionMaxima) {

            this.presionMaxima = presionMaxima;

        }

        @Override

        public void agregarDietaDiaria(DietaDiaria dieta) {

            if (dieta.aptaHipertenso() == true) {

                getIngesta().add(dieta);

            }

        }

    }

}

package ar.edu.unlam.pb2.ea1;

import java.util.ArrayList;

public abstract class Paciente {

    private String nombre;

    private String apellido;

    private Integer dni;

    private ArrayList<DietaDiaria> ingesta;

    public Paciente(String nombre, String apellido, Integer dni) {

```

```
        this.nombre = nombre;

        this.apellido = apellido;

        this.dni = dni;

        this.ingesta = new ArrayList<DietaDiaria>();
    }
}
```

```
public String getNombre() {
    return nombre;
}
```

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

```
public String getApellido() {
    return apellido;
}
```

```
public void setApellido(String apellido) {
    this.apellido = apellido;
}
```

```
public Integer getDni() {
    return dni;
}
```

```
public void setDni(Integer dni) {
```

```

        this.dni = dni;
    }

    public ArrayList<DietaDiaria> getIngesta() {
        return ingesta;
    }

    public void setIngesta(ArrayList<DietaDiaria> ingesta) {
        this.ingesta = ingesta;
    }

    public abstract void agregarDietaDiaria(DietaDiaria dieta);
}

package ar.edu.unlam.pb2.ea1;

public class Ingrediente{

    private String nombre;

    private Integer cantidadSal;

    private Integer cantidadAzucar;

    public Ingrediente(String nombre, Integer cantidadSal, Integer cantidadAzucar) {
        this.nombre = nombre;

        this.cantidadSal = cantidadSal;

        this.cantidadAzucar = cantidadAzucar;
    }

```

```
}
```

```
public String getNombre() {
```

```
    return nombre;
```

```
}
```

```
public void setNombre(String nombre) {
```

```
    this.nombre = nombre;
```

```
}
```

```
public Integer getCantidadSal() {
```

```
    return cantidadSal;
```

```
}
```

```
public void setCantidadSal(Integer cantidadSal) {
```

```
    this.cantidadSal = cantidadSal;
```

```
}
```

```
public Integer getCantidadAzucar() {
```

```
    return cantidadAzucar;
```

```
}
```

```
public void setCantidadAzucar(Integer cantidadAzucar) {
```

```
    this.cantidadAzucar = cantidadAzucar;
```

```
}
```

```
}
```

```
package ar.edu.unlam.pb2.ea1;
```

```
import java.util.ArrayList;
```

```
public class Plato {
```

```
    /**
```

```
     * Por ejemplo, si para un Plato se necesitan DOS (2) tomates, se deberá agregar
```

```
     * DOS(2) unidades de tomate.
```

```
    */
```

```
    private String nombreDePlato;
```

```
    private Integer cantidadIngrediente;
```

```
    private ArrayList<Ingrediente> ingredientes = new ArrayList<Ingrediente>();
```

```
    public Plato(String nombreDePlato) {
```

```
        this.nombreDePlato = nombreDePlato;
```

```
    }
```

```
    public String getNombreDePlato() {
```

```
        return nombreDePlato;
```

```
    }
```

```
    public void setNombreDePlato(String nombreDePlato) {
```

```
        this.nombreDePlato = nombreDePlato;
```

```
    }
```

```
public ArrayList<Ingrediente> getIngredientes() {  
    return ingredientes;  
}
```

```
public void setIngredientes(ArrayList<Ingrediente> ingredientes) {  
    this.ingredientes = ingredientes;  
}
```

```
public boolean agregarIngredienteAPlato(Ingrediente ingrediente, Integer  
cantidadIngrediente) {  
    for (int i = 0; i < cantidadIngrediente; i++) {  
        ingredientes.add(ingrediente);  
    }  
    return true;  
}
```

```
public Integer obtenerCantidadDeSal() {  
    Integer cantidadDeSalEnUnPlato = 0;  
    for (Ingrediente ingrediente : ingredientes) {  
        cantidadDeSalEnUnPlato += ingrediente.getCantidadSal();  
    }  
    return cantidadDeSalEnUnPlato;  
}
```



```

public Integer obtenerCantidadDeAzucar() {

    Integer cantidadDeAzucarEnUnPlato = 0;

    for (Ingrediente ingrediente : ingredientes) {

        cantidadDeAzucarEnUnPlato += ingrediente.getCantidadAzucar();

    }

    return cantidadDeAzucarEnUnPlato;

}

}

```

```

package ar.edu.unlam.pb2.ea1;

```

```

import java.util.HashSet;

```

```

public class DietaDiaria {

```

```

    /**

```

```

     * No se permite repetir platos durante todo el día.

```

```

     */

```

```

    private String dia;

```

```

    private HashSet<Plato> dietaDiaria = new HashSet<Plato>();

```

```

    public DietaDiaria(String dia) {

```

```
        this.dia = dia;
    }
}
```

```
public boolean agregarPlatoADieta(Plato plato) {

    return dietaDiaria.add(plato);
}
```

```
public Plato buscarPlatoEnDieta(String nombre) {

    Plato platoEncontrado = null;

    for (Plato plato : dietaDiaria) {

        if (plato.getNombreDePlato().equals(nombre)) {

            platoEncontrado = plato;

        }

    }

    return platoEncontrado;

}
```

```
private Integer totalDeSalEnDieta() {

    Integer cantSalEnDieta = 0;

    for (Plato plato : dietaDiaria) {

        cantSalEnDieta += plato.obtenerCantidadDeSal();

    }

}
```

```
}
```

```
return cantSalEnDieta;
```

```
}
```

```
private Integer totalDeAzucarEnDieta() {
```

```
    Integer cantAzucarEnDieta = 0;
```

```
    for (Plato plato : dietaDiaria) {
```

```
        cantAzucarEnDieta += plato.obtenerCantidadDeAzucar();
```

```
    }
```

```
    return cantAzucarEnDieta;
```

```
}
```

```
public Boolean aptaHipertenso() {
```

```
    if (totalDeSalEnDieta() < 4) {
```

```
        return true;
```

```
    }
```

```
    return false;
```

```
}
```

```
public Boolean aptaDiabetico() {
```

```
    if (totalDeAzucarEnDieta() <= 25) {
```

```
        return true;
```

```

        }

        return false;

    }

    public HashSet<Plato> getDietaDiaria() {

        return dietaDiaria;

    }

    public void setDietaDiaria(HashSet<Plato> dietaDiaria) {

        this.dietaDiaria = dietaDiaria;

    }

    @Override

    public int hashCode() {

        final int prime = 31;

        int result = 1;

        result = prime * result + ((dietaDiaria == null) ? 0 : dietaDiaria.hashCode());

        return result;

    }

    @Override

    public boolean equals(Object obj) {

        if (this == obj)

            return true;

        if (obj == null)

            return false;

```

```

        if (getClass() != obj.getClass())
            return false;

        DietaDiaria other = (DietaDiaria) obj;

        if (dietaDiaria == null) {
            if (other.dietaDiaria != null)
                return false;
        } else if (!dietaDiaria.equals(other.dietaDiaria))
            return false;

        return true;
    }

}

package ar.edu.unlam.pb2.ea1;

import static org.junit.Assert.*;

import org.junit.Test;

public class DietaDiariaTest {

    @Test
    public void testQueNosePuedanAgregarPlatoRepetidos() {
        Ingrediente i1 = new Ingrediente("Puré", 1, 2);
        Ingrediente i2 = new Ingrediente("Milanesas", 1, 0);
        Plato p1 = new Plato("Milanesas con puré");
        DietaDiaria d1 = new DietaDiaria("Dia1");
    }
}

```

```

        p1.agregarIngredienteAPlato(i1, 1);
        p1.agregarIngredienteAPlato(i2, 2);
        d1.agregarPlatoADieta(p1);
        d1.agregarPlatoADieta(p1);
        assertEquals(1, d1.getDietaDiaria().size());
    }

```

@Test

```

public void testQueDietaDiariaSeaAptaHipertenso() {

    Ingrediente i1 = new Ingrediente("Puré", 1, 2);
    Ingrediente i2 = new Ingrediente("Milanesas", 1, 0);
    Plato p1 = new Plato("Milanesas con puré");
    DietaDiaria d1 = new DietaDiaria("Dia1");

    p1.agregarIngredienteAPlato(i1, 1);
    p1.agregarIngredienteAPlato(i2, 2);
    d1.agregarPlatoADieta(p1);
    assertTrue(d1.aptaHipertenso());
}

```

@Test

```

public void testQueDietaDiariaNoSeaAptaHipertenso() {

    Ingrediente i1 = new Ingrediente("Puré", 1, 2);
    Ingrediente i2 = new Ingrediente("Milanesas", 4, 2);

```

```
        Plato p1 = new Plato("Milanesas con puré");

        DietaDiaria d1 = new DietaDiaria("Dia1");


        p1.agregarIngredienteAPlato(i1, 1);

        p1.agregarIngredienteAPlato(i2, 2);

        d1.agregarPlatoADieta(p1);

        assertFalse(d1.aptaHipertenso());

    }
}
```

@Test

```
public void testQueDietaDiariaSeaAptaDiabetico() {

    Ingrediente i1 = new Ingrediente("Puré", 1, 20);

    Ingrediente i2 = new Ingrediente("Milanesas", 4, 2);

    Plato p1 = new Plato("Milanesas con puré");

    DietaDiaria d1 = new DietaDiaria("Dia1");


    p1.agregarIngredienteAPlato(i1, 1);

    p1.agregarIngredienteAPlato(i2, 2);

    d1.agregarPlatoADieta(p1);

    assertTrue(d1.aptaDiabetico());

}
}
```

@Test

```
public void testQueDietaDiariaNoSeaAptaDiabetico() {
```

```

        Ingrediente i1 = new Ingrediente("Puré", 1, 22);

        Ingrediente i2 = new Ingrediente("Milanesas", 4, 2);

        Plato p1 = new Plato("Milanesas con puré");

        DietaDiaria d1 = new DietaDiaria("Dia1");


        p1.agregarIngredienteAPlato(i1, 1);

        p1.agregarIngredienteAPlato(i2, 2);

        d1.agregarPlatoADieta(p1);

        assertFalse(d1.aptaDiabetico());

    }

}

package ar.edu.unlam.pb2.ea1;

import static org.junit.Assert.*;

import org.junit.Test;

public class PacienteTest {

    @Test

    public void testQueUnDiabeticoIngesteDietaAptaParaDiabetico() {

        Ingrediente i1 = new Ingrediente("Puré", 1, 2);

        Ingrediente i2 = new Ingrediente("Milanesas", 1, 2);

```



```

        Plato p1 = new Plato("Milanesas con puré");

        DietaDiaria d1 = new DietaDiaria("Dia1");

        Diabetico diabetico1 = new Diabetico("Josemir", "Lujambio", 42225291, 5);

        p1.agregarIngredienteAPlato(i1, 1);

        p1.agregarIngredienteAPlato(i2, 2);

        d1.agregarPlatoADieta(p1);

        d1.agregarPlatoADieta(p1);

        diabetico1.agregarDietaDiaria(d1);

        assertEquals(diabetico1.getIngesta().size(), 1);

    }

```

@Test

```

public void testQueUnDiabeticoNoIngesteDietaAptaParaDiabetico() {

    Ingrediente i1 = new Ingrediente("Puré", 1, 25);

    Ingrediente i2 = new Ingrediente("Milanesas", 1, 2);

    Plato p1 = new Plato("Milanesas con puré");

    DietaDiaria d1 = new DietaDiaria("Dia1");

    Diabetico diabetico1 = new Diabetico("Josemir", "Lujambio", 42225291, 5);

    p1.agregarIngredienteAPlato(i1, 1);

    p1.agregarIngredienteAPlato(i2, 2);

    d1.agregarPlatoADieta(p1);

    d1.agregarPlatoADieta(p1);

    diabetico1.agregarDietaDiaria(d1);

    assertEquals(diabetico1.getIngesta().size(), 0);

}

```

@Test

public void testQueUnHipertendoIngesteDietaAptaParaHipertenso() {

Ingrediente i1 = new Ingrediente("Puré", 1, 2);

Ingrediente i2 = new Ingrediente("Milanesas", 1, 2);

Plato p1 = new Plato("Milanesas con puré");

DietaDiaria d1 = new DietaDiaria("Dia1");

Hipertenso hipertenso1 = new Hipertenso("Josemir", "Lujambio", 42225291,
10, 29);

p1.agregarIngredienteAPlato(i1, 1);

p1.agregarIngredienteAPlato(i2, 2);

d1.agregarPlatoADieta(p1);

d1.agregarPlatoADieta(p1);

hipertenso1.agregarDietaDiaria(d1);

assertEquals(hipertenso1.getIngesta().size(), 1);

}

@Test

public void testQueUnHipertendoNoIngesteDietaAptaParaHipertenso() {

Ingrediente i1 = new Ingrediente("Puré", 4, 2);

Ingrediente i2 = new Ingrediente("Milanesas", 2, 2);

Plato p1 = new Plato("Milanesas con puré");

DietaDiaria d1 = new DietaDiaria("Dia1");

Hipertenso hipertenso1 = new Hipertenso("Josemir", "Lujambio", 42225291,
10, 29);

```

        p1.agregarIngredienteAPlato(i1, 1);

        p1.agregarIngredienteAPlato(i2, 2);

        d1.agregarPlatoADieta(p1);

        d1.agregarPlatoADieta(p1);

        hipertenso1.agregarDietaDiaria(d1);

        assertEquals(hipertenso1.getIngesta().size(), 0);

    }

}

```

```

package ar.edu.unlam.pb2.ea1;

```

```

import static org.junit.Assert.assertEquals;

```

```

import org.junit.Test;

```

```

public class PlatoTest {

```

```

    @Test

```

```

    public void testQueSePuedanAgregarIngredientesIguales(){

```

```

        Ingrediente i2 = new Ingrediente("Milanesas", 4, 2);

```

```

        Plato p1 = new Plato("Milanesas con puré");

```

```

        p1.agregarIngredienteAPlato(i2, 2);

```

```

        assertEquals(p1.getIngresos().size(), 2);

```

```

    }

```

@Test

```
public void testQueSeOBTengaLaCantidadDeSalCorrecta(){  
    Ingrediente i2 = new Ingrediente("Milanesas", 4, 2);  
    Plato p1 = new Plato("Milanesas con puré");  
    p1.agregarIngredienteAPlato(i2, 2);  
  
    assertEquals(p1.obtenerCantidadDeSal(), 8, 0);  
  
}
```

@Test

```
public void testQueSeObtenegaLaCanidadDeAzucarCorrecta() {  
  
    Ingrediente i2 = new Ingrediente("Milanesas", 4, 2);  
    Plato p1 = new Plato("Milanesas con puré");  
    p1.agregarIngredienteAPlato(i2, 2);  
  
    assertEquals(p1.obtenerCantidadDeAzucar(), 4, 0);  
  
}
```

}