

Actividad 1. Estructuras de control

Lenguajes de Programación IV

Ingeniería en Desarrollo de Software

Tutor: Aarón Iván Salazar Macías

Alumno: José Luis Pacheco González

Fecha: 20 de marzo 2024

Índice

Introducción	3
Descripción	4
Justificación	5
Desarrollo	6
• Interfaz	6
• Codificación	9
Conclusión	22
Referencias	23

Introducción

Java es un lenguaje de programación que se utiliza para el desarrollo de aplicaciones de software debido a su versatilidad y portabilidad. NetBeans, por otro lado, es un entorno de desarrollo integrado (IDE) que proporciona herramientas robustas para escribir, compilar y depurar código Java de manera eficiente.

Esta actividad consiste en desarrollar una aplicación común para medir el índice de masa corporal IMC utilizando Java y NetBeans. Esta aplicación permite al usuario ingresar su peso y estatura, luego calcula y muestra su IMC, una medida que evalúa la relación entre el peso y la estatura de una persona para determinar si se encuentra en un rango saludable.

El proceso de desarrollo de esta aplicación implica la creación de una interfaz de usuario intuitiva utilizando componentes gráficos proporcionados por NetBeans, así como la implementación de la lógica de cálculo del IMC en Java. NetBeans facilita la creación de aplicaciones Java mediante su diseño visual y sus capacidades de depuración, lo que permite a los desarrolladores concentrarse en la lógica de la aplicación en lugar de preocuparse por detalles de implementación más tediosos.

Descripción

El índice de masa corporal (IMC) es una medida que relaciona la masa y la altura de una persona, permitiendo evaluar su estado de peso. Dado el alto índice de masa corporal en la población mexicana, un hospital en la Ciudad de México necesita un programa para calcular el IMC de sus pacientes. Para abordar esto, se propone un sistema en Java 8.

El sistema solicitará al usuario ingresar su peso en kilogramos y su estatura en metros. Luego, calculará el IMC utilizando la fórmula establecida ($IMC = \text{peso} / (\text{altura} * \text{altura})$). Según el resultado, se mostrará un mensaje que clasificará el estado de peso del individuo, utilizando las categorías establecidas: bajo peso, peso normal, sobrepeso, obesidad grado I, obesidad grado II y obesidad grado III.

La implementación requerirá una interfaz gráfica para capturar los datos del paciente y presentar el resultado. Además, se integrará lógica de programación para realizar los cálculos del IMC y clasificar el estado de peso correctamente, siguiendo los parámetros establecidos por la tabla de referencia.

Justificación

La creación de aplicaciones utilizando Java y NetBeans posee una importancia significativa debido a varios factores clave. En primer lugar, Java es un lenguaje de programación ampliamente utilizado y respaldado por una gran comunidad de desarrolladores y empresas. Su portabilidad y robustez lo convierten en una opción ideal para desarrollar aplicaciones que funcionen en una amplia variedad de dispositivos y sistemas operativos.

NetBeans, como entorno de desarrollo integrado, proporciona herramientas poderosas que simplifican y agilizan el proceso de desarrollo de software. Su interfaz intuitiva y su conjunto de características, como la edición de código, la depuración y la administración de proyectos, hacen que sea más eficiente para los desarrolladores crear y mantener aplicaciones Java.

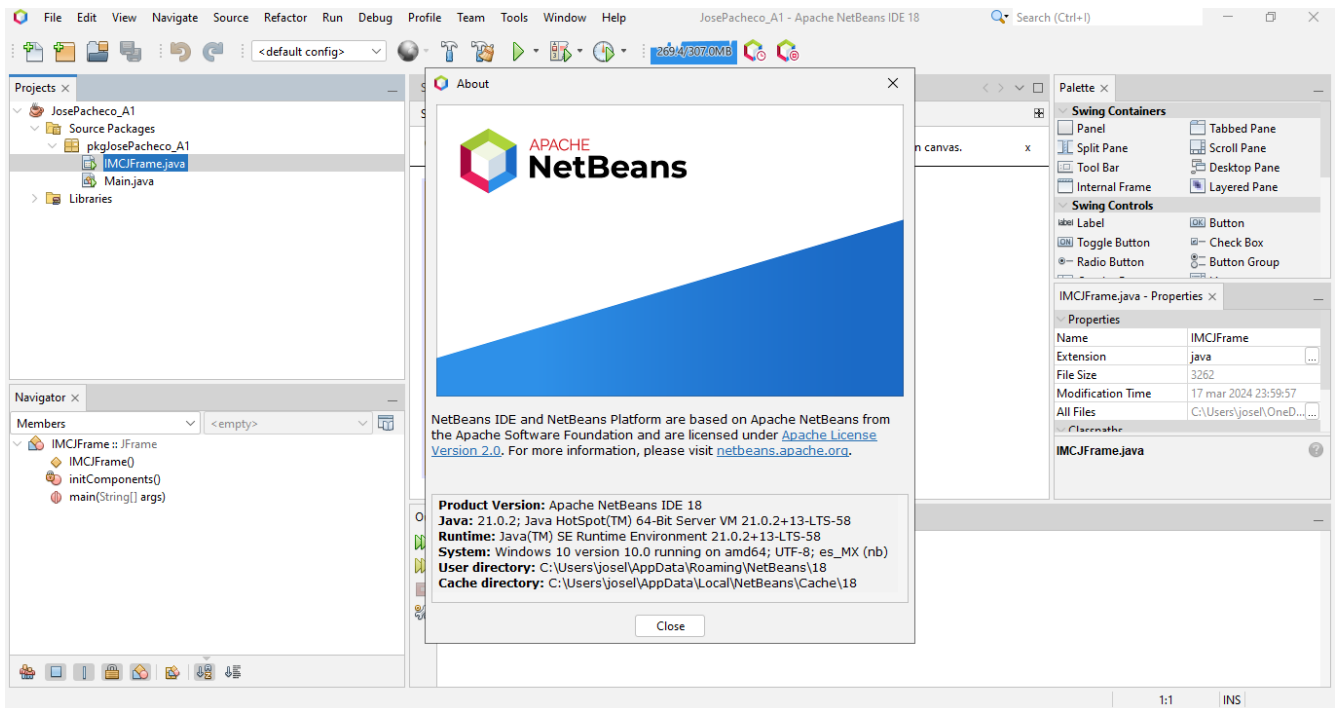
La combinación de Java y NetBeans permite a los desarrolladores crear aplicaciones complejas y funcionales de manera más rápida y efectiva. Además, estas herramientas son ideales para desarrollar aplicaciones con interfaces de usuario atractivas y amigables, lo que mejora la experiencia del usuario final.

La importancia de crear aplicaciones utilizando Java y NetBeans radica en su capacidad para ofrecer soluciones robustas, portables y eficientes que satisfacen las necesidades de los usuarios en una amplia variedad de contextos, desde aplicaciones empresariales hasta herramientas de productividad personal.

Desarrollo

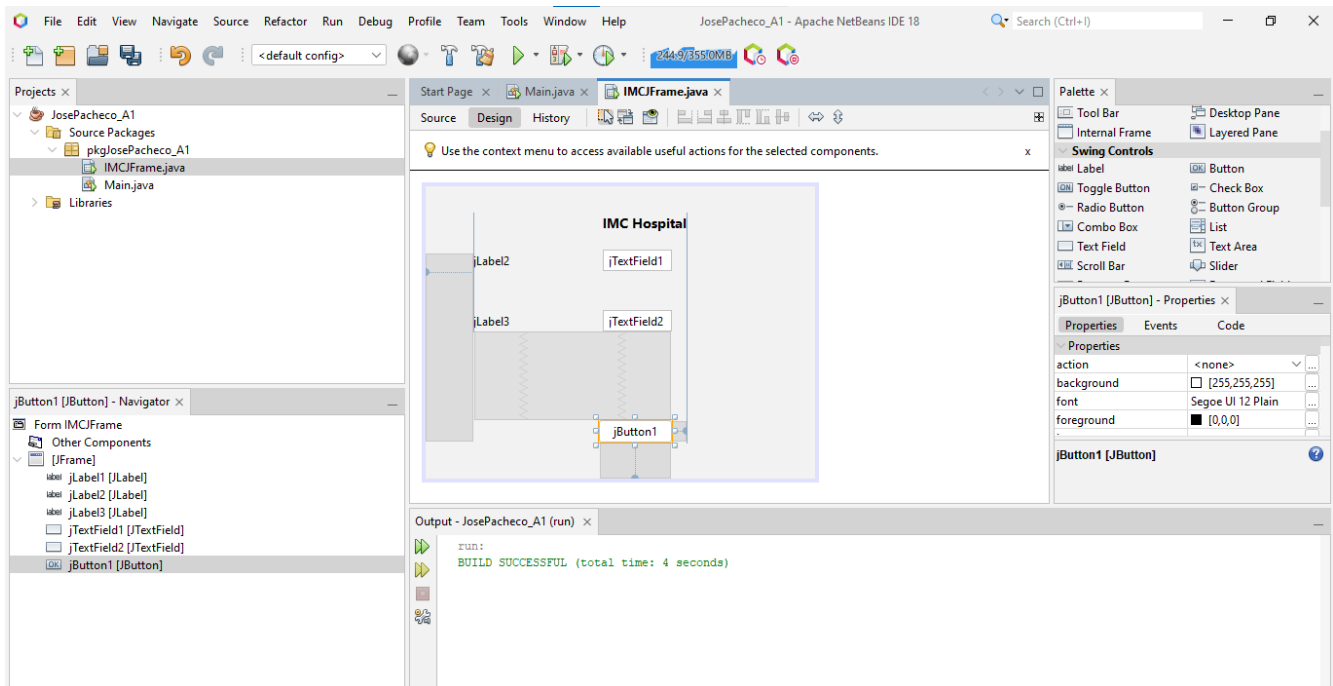
- Interfaz

Instalación de netbeans

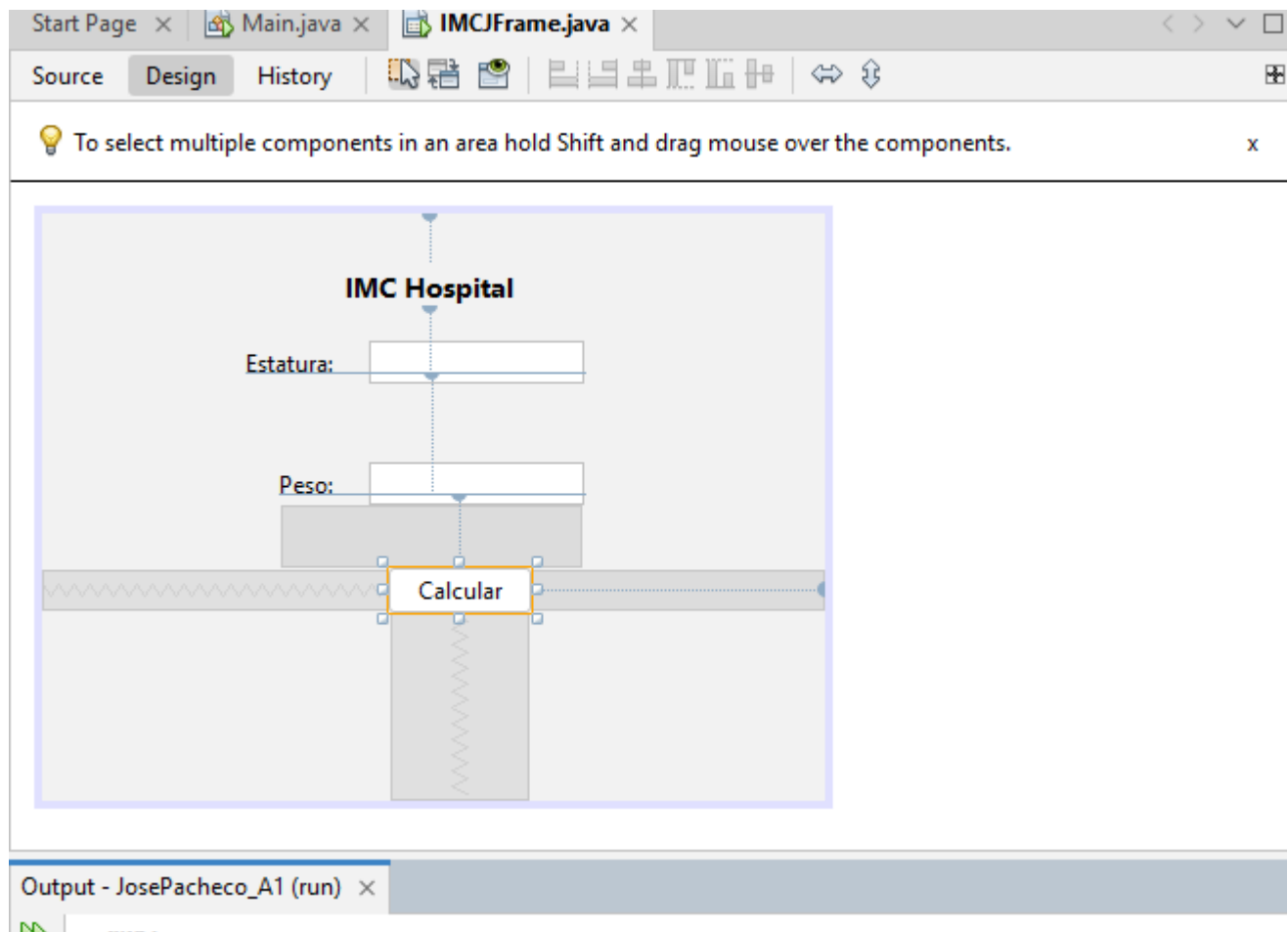


Proceso de diseño

De inicio se debe crear un form que incluya las etiquetas de estatura y peso, así como las cajas de texto para poder ingresar datos y sobre todo el botón que servirá para poder ejecutar el cálculo del IMC.

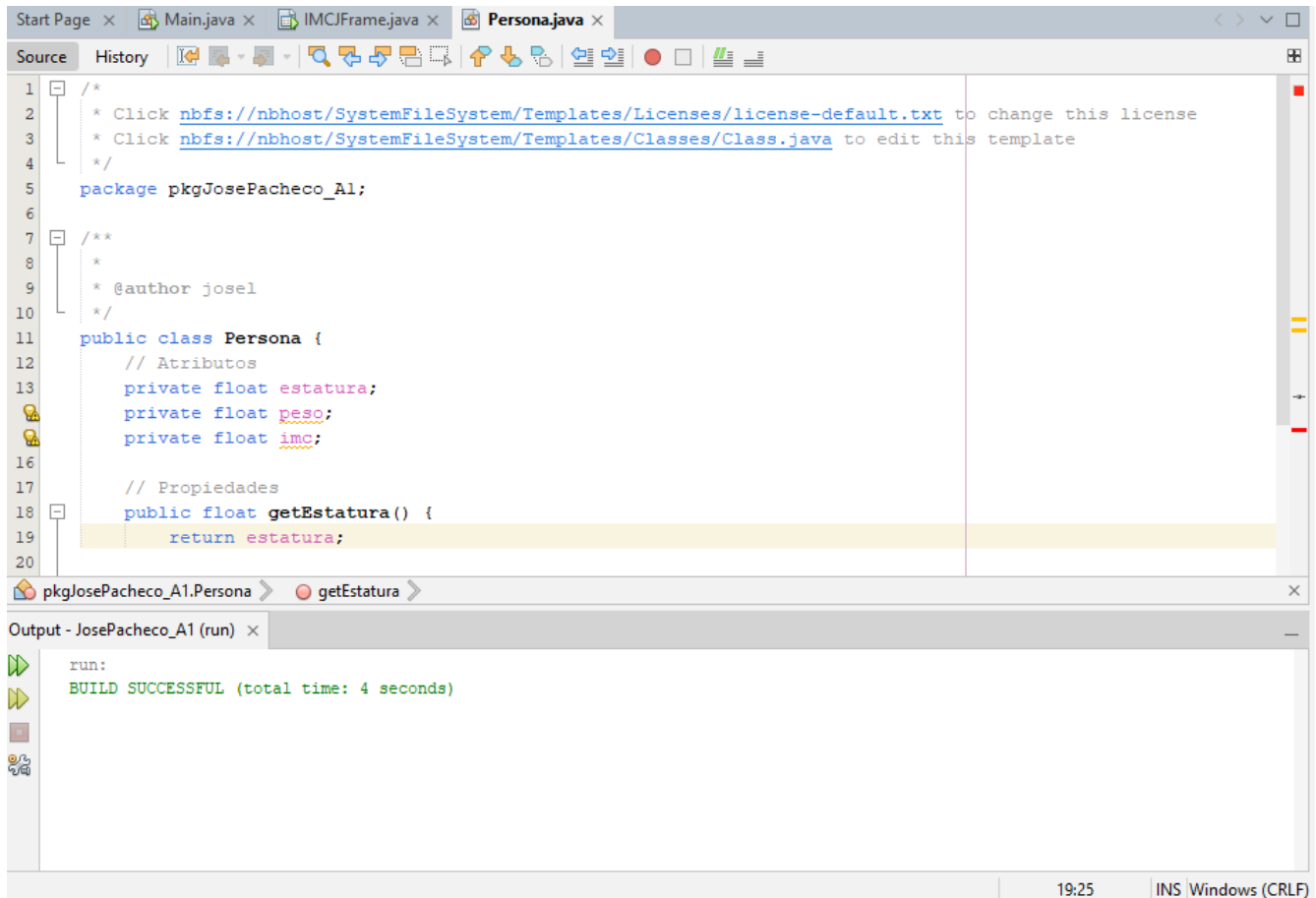


Aquí se muestra el diseño final .



- **Codificación**

Proceso de creación de la clase persona, primero se agregan los atributos necesarios, como lo es la estatura, peso y el IMC.



The screenshot displays an IDE window with the file 'Persona.java' open. The code defines a package 'pkgJosePacheco_A1' and a public class 'Persona'. Inside the class, there are three private float attributes: 'estatura', 'peso', and 'imc'. A public method 'getEstatura()' is also defined, which returns the value of 'estatura'. The IDE's output window at the bottom shows a successful build message: 'BUILD SUCCESSFUL (total time: 4 seconds)'. The status bar at the bottom right indicates the time is 19:25 and the system is INS Windows (CRLF).

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package pkgJosePacheco_A1;
6
7  /**
8   *
9   * @author josel
10  */
11  public class Persona {
12      // Atributos
13      private float estatura;
14      private float peso;
15      private float imc;
16
17      // Propiedades
18      public float getEstatura() {
19          return estatura;
20      }
```

Output - JosePacheco_A1 (run) ×

```
run:
BUILD SUCCESSFUL (total time: 4 seconds)
```

19:25 | INS | Windows (CRLF)

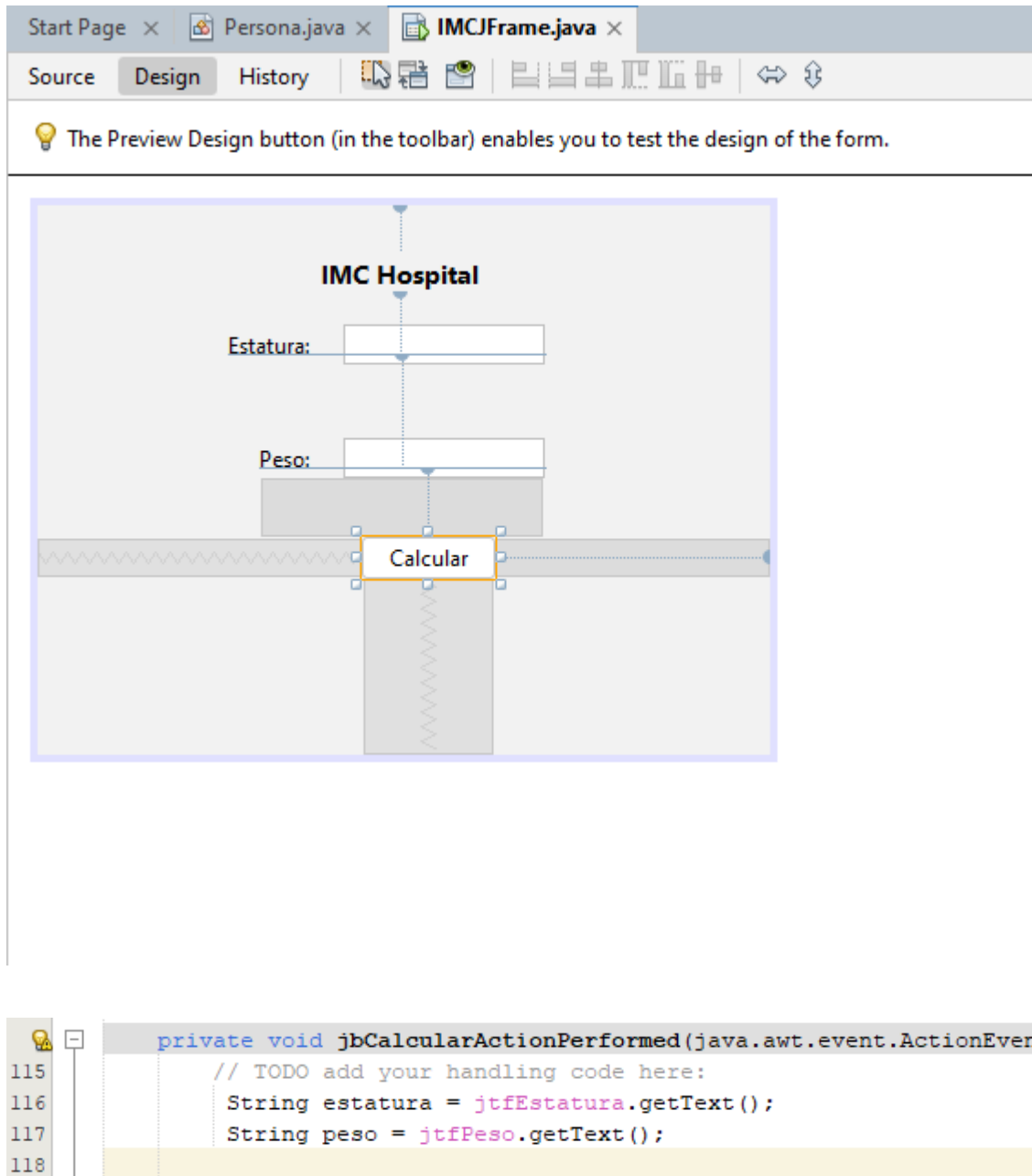
En esta captura se muestran las propiedades necesarias y se agrega la fórmula para obtener el cálculo del IMC, para ello se debe dividir el peso entre la estatura y multiplicarlo por la estatura, quedando de la siguiente manera.

$$\text{imc} = \text{getPeso()} / (\text{getEstatura()} * \text{getEstatura()})$$

En la parte inferior de la captura se muestra la línea de código que generará el mensaje con el resultado del cálculo del IMC.

```
19 // Propiedades
20 public float getEstatura() {
21     return estatura;
22 }
23
24 public void setEstatura(float e) {
25     estatura = e;
26 }
27
28 public float getPeso() {
29     return peso;
30 }
31
32 public void setPeso(float p) {
33     peso = p;
34 }
35
36 public float getIMC() {
37     return imc;
38 }
39
40 public void setIMC() {
41     imc = getPeso() / (getEstatura() * getEstatura());
42 }
43
44 public void mostrarIMC() {
45     JOptionPane.showMessageDialog(parentComponent:null, "Su IMC es: " + getIMC());
46 }
```

En el paso siguiente debe abrir el IMCJFrame para ingresar al código del botón calcular y declarar 2 variables de tipo string.



The screenshot shows an IDE with three tabs: 'Start Page', 'Persona.java', and 'IMCJFrame.java'. The 'Design' tab is selected. The design view shows a window titled 'IMC Hospital' with two text input fields labeled 'Estatura:' and 'Peso:', and a 'Calcular' button. A tooltip message says: 'The Preview Design button (in the toolbar) enables you to test the design of the form.'

```
115 private void jbCalcularActionPerformed(java.awt.event.ActionEvent evt) {  
116     // TODO add your handling code here:  
117     String estatura = jTextFieldEstatura.getText();  
118     String peso = jTextFieldPeso.getText();
```

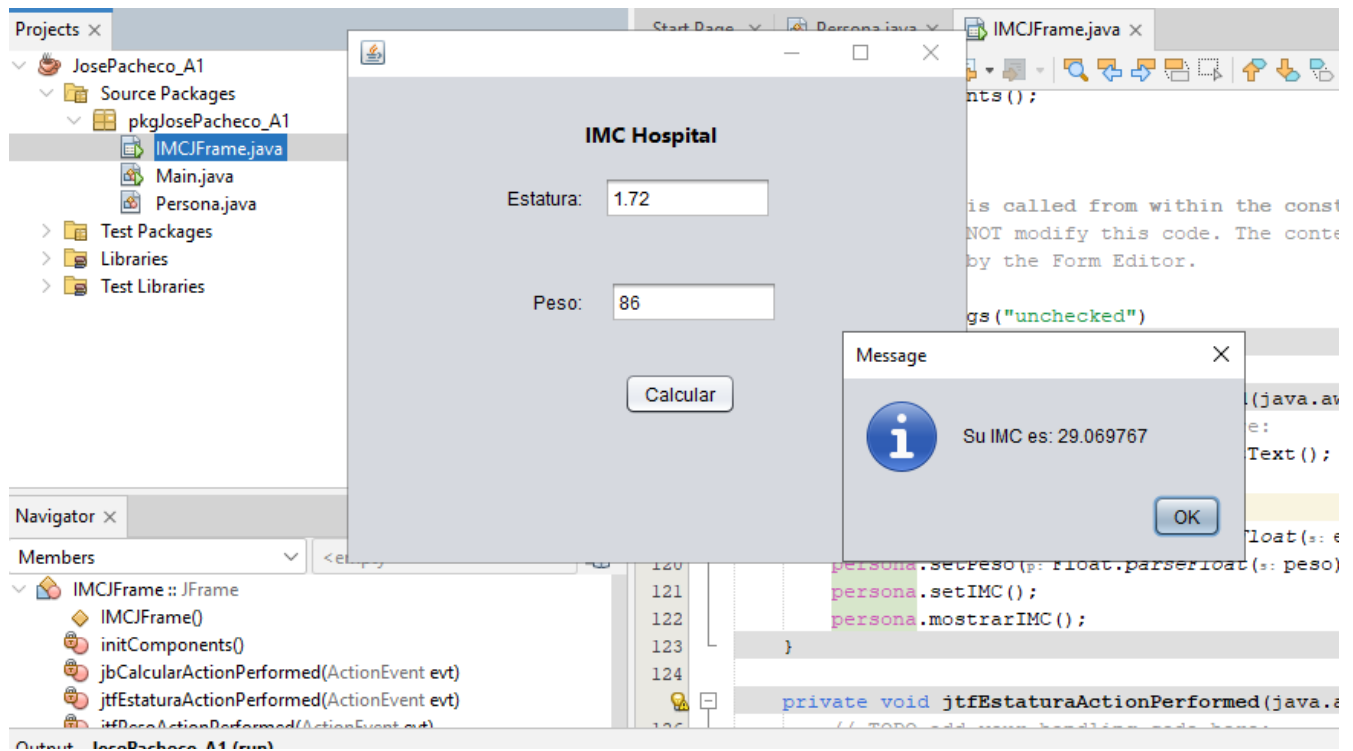
También se debe crear la instancia de la clase.

```
11 public class IMCJFrame extends javax.swing.JFrame {  
12  
13     Persona persona = new Persona ();  
14  
15 }
```

Paso seguido, delante de las 2 variables que se declararon anteriormente, se manda llamar a persona, pero se le debe cambiar el valor a tiempo float, como se muestra en la captura.

```
15 private void jbCalcularActionPerformed(java.awt.event.ActionEvent evt) {  
16     // TODO add your handling code here:  
17     String estatura = jtEstatura.getText();  
18     String peso = jtPeso.getText();  
19     persona.setEstatura(e: Float.parseFloat(s: estatura));  
20     persona.setPeso(p: Float.parseFloat(s: peso));  
21     persona.setIMC();  
22     persona.mostrarIMC();  
23 }
```

Llegado a este punto se realiza una prueba de funcionamiento agregando datos y como se puede observar en la captura, en el funcionamiento de la aplicación funciona de forma correcta.



Para poder definir los niveles de IMC se debe ingresar la clase persona y e ingresar las siguientes líneas de código que definen las mediciones como lo solicita la actividad.



Para poder codificar esta función se deben agregar if anidados donde se definen si el resultado es menor, mayor o igual acorde cada nivel y agregando el mensaje correspondiente a cada uno.

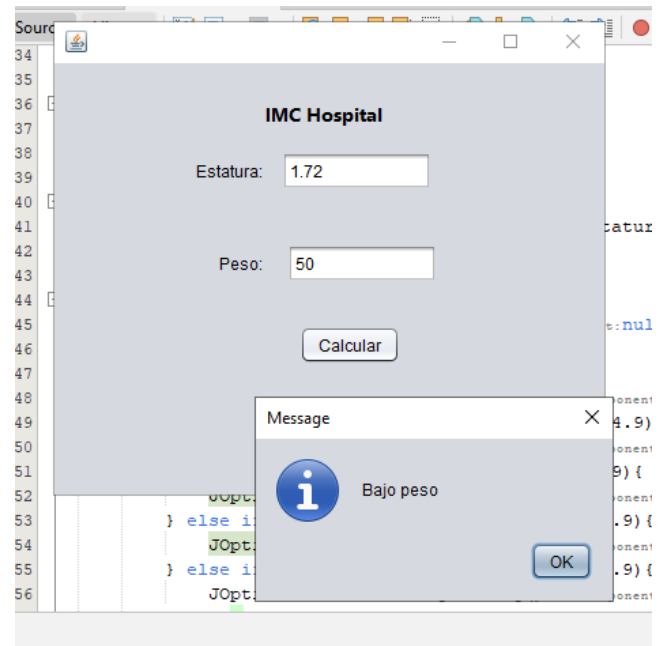
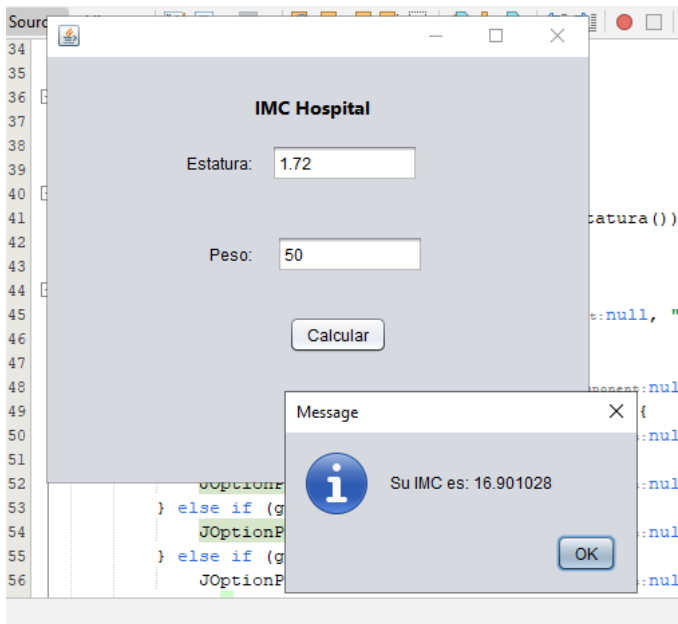
Quedando de la siguiente manera.

```
46
47     if(getIMC()<18.5){
48         JOptionPane.showMessageDialog(parentComponent:null, message:"Bajo peso");
49     } else if (getIMC()>= 18.5 && getIMC()<=24.9){
50         JOptionPane.showMessageDialog(parentComponent:null, message:"Normal");
51     } else if (getIMC()>= 25 && getIMC()<=29.9){
52         JOptionPane.showMessageDialog(parentComponent:null, message:"Sobrepeso");
53     } else if (getIMC()>= 30 && getIMC()<= 34.9){
54         JOptionPane.showMessageDialog(parentComponent:null, message:"Obesidad I");
55     } else if (getIMC()>= 35 && getIMC()<= 39.9){
56         JOptionPane.showMessageDialog(parentComponent:null, message:"Obesidad II");
57     } else{
58         JOptionPane.showMessageDialog(parentComponent:null, message:"Obesidad III");
59     }
60 }
61
62
```

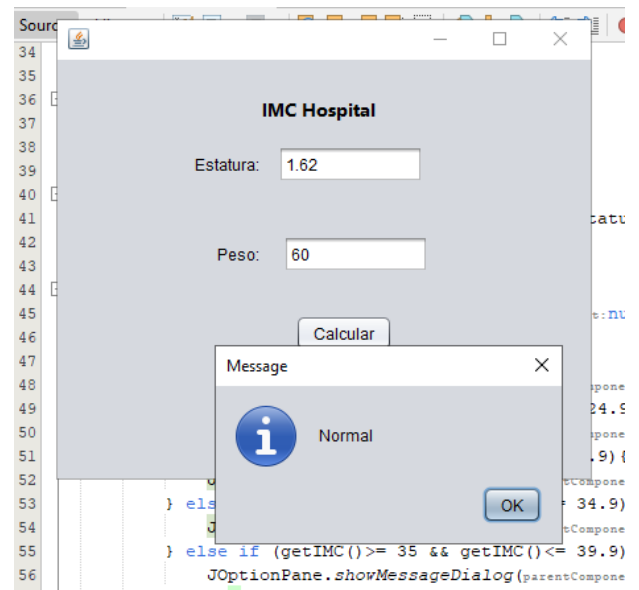
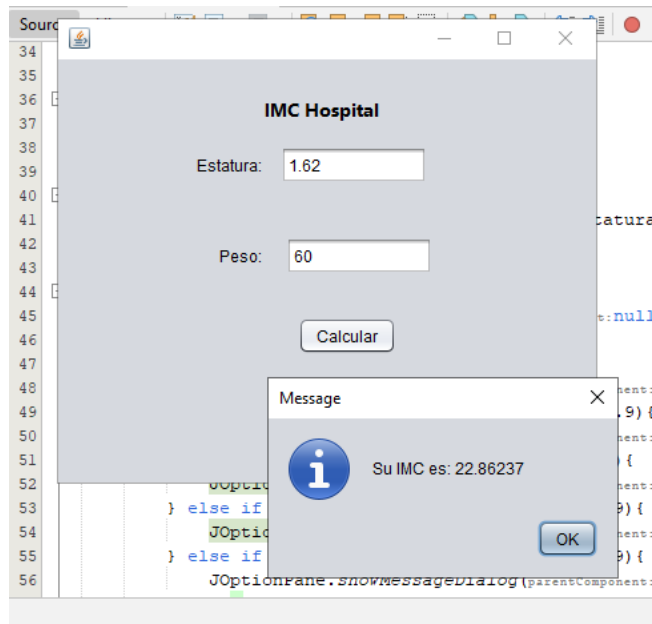
pkgJosePacheco_A1.Persona > mostrarIMC > if (getIMC() < 18.5) else if (getIMC() >= 18.5 && getIMC() <= 24.9) else if (getIMC()

A continuación, se muestra el funcionamiento correcto de la aplicación, mostrando cada uno de los rangos definidos.

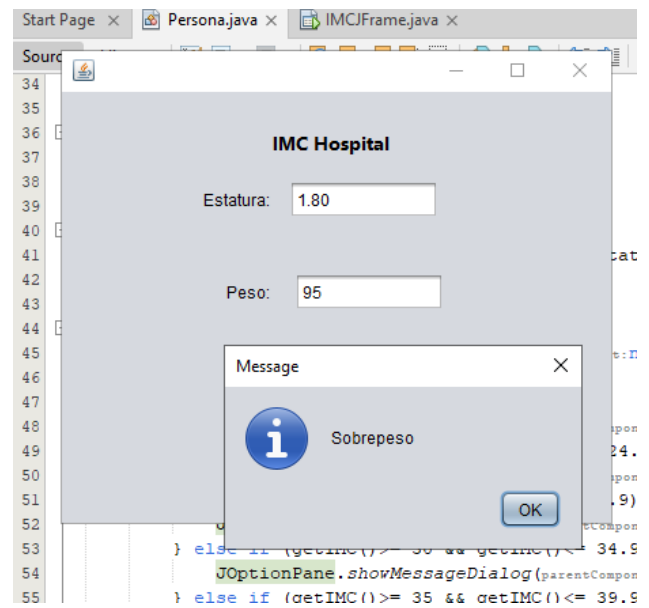
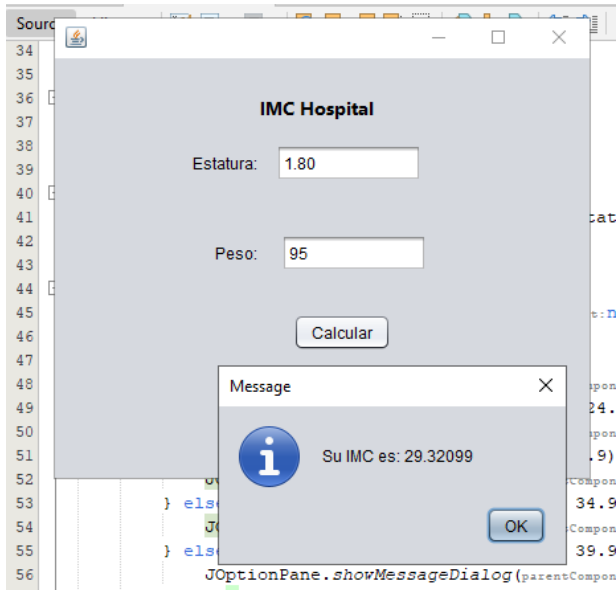
Bajo peso:



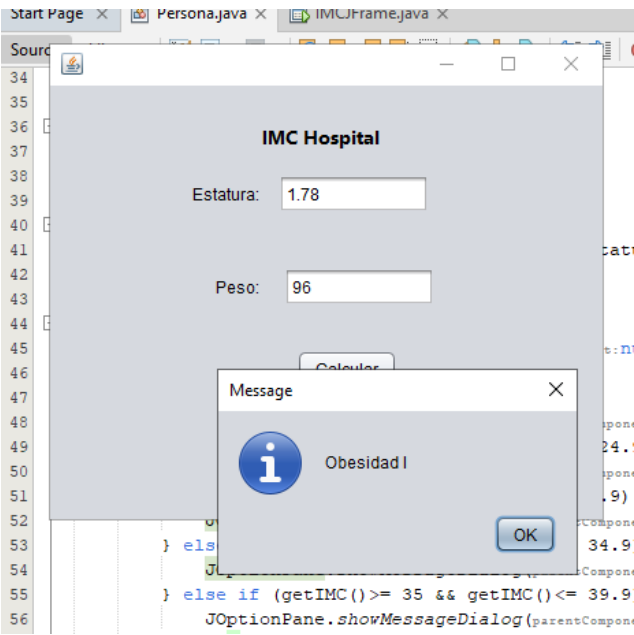
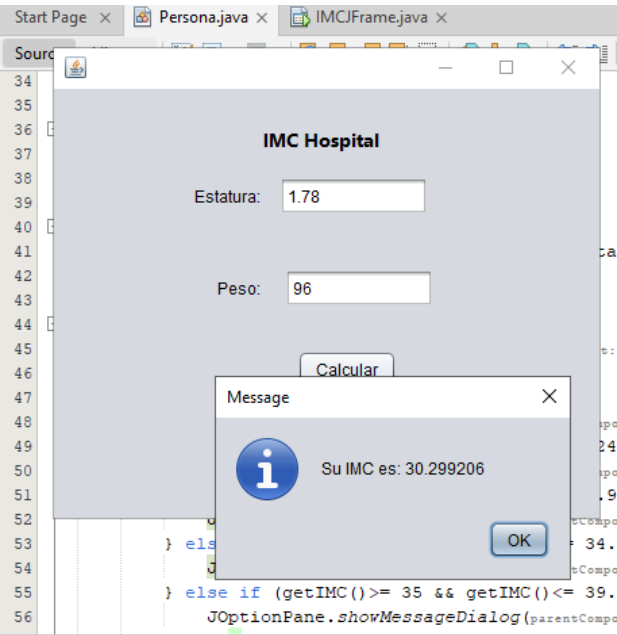
Normal:



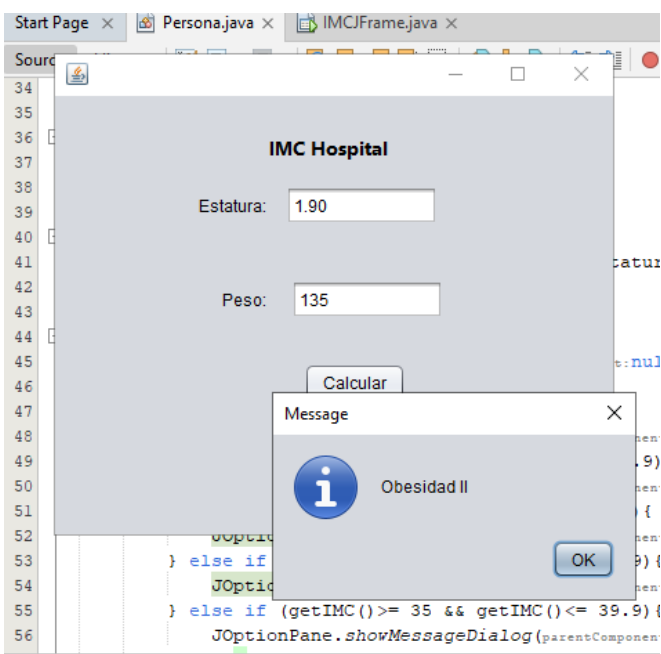
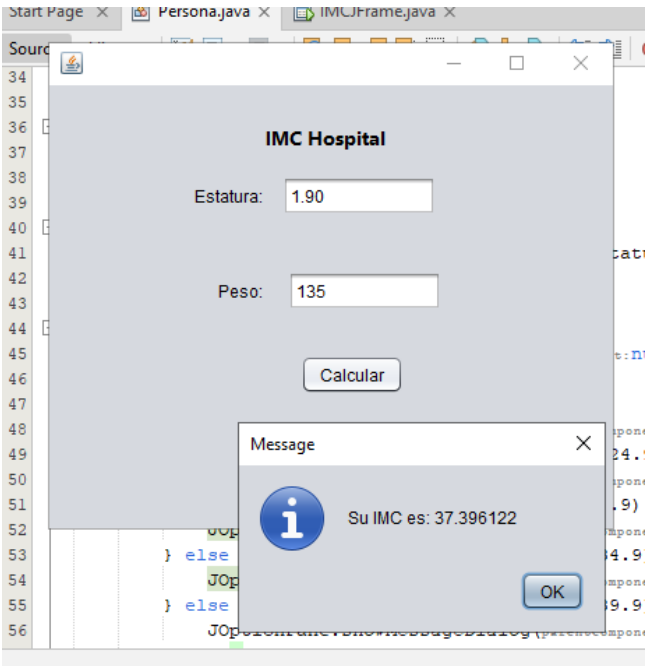
Sobrepeso:



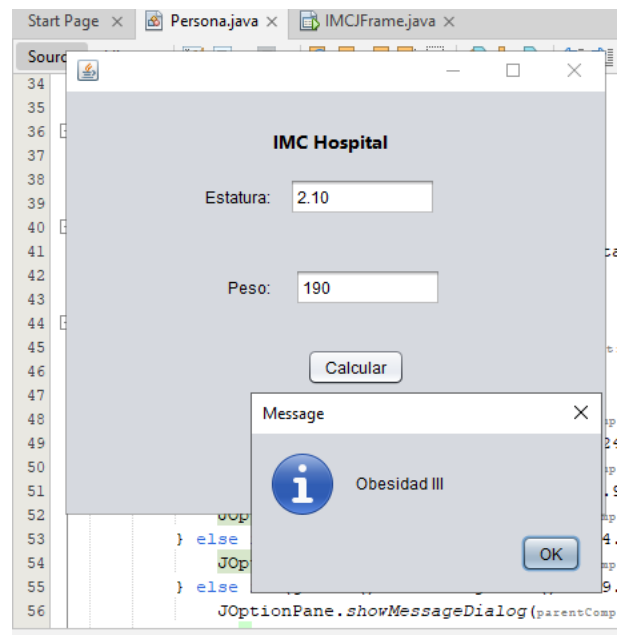
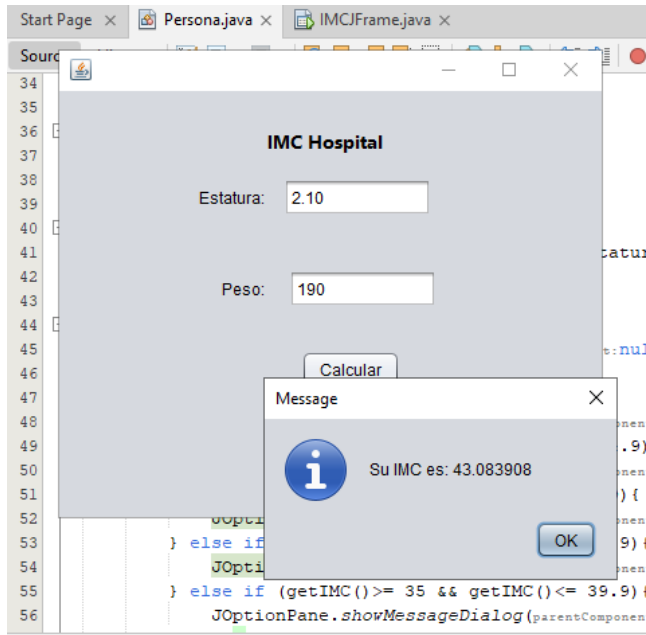
Obesidad I



Obesidad II



Obesidad III



Conclusión

El desarrollo de aplicaciones en Java y NetBeans emerge como una solución versátil y eficaz tanto en el ámbito laboral como en la vida cotidiana. La combinación de Java, un lenguaje de programación robusto y altamente portátil, con NetBeans, un entorno de desarrollo integrado (IDE) poderoso y amigable, ofrece una plataforma ideal para crear una amplia gama de aplicaciones.

En el ámbito laboral, estas herramientas permiten a los desarrolladores crear aplicaciones empresariales complejas con interfaces de usuario intuitivas y funcionales. La capacidad de Java para ofrecer portabilidad y su sólida estructura hacen que sea una opción ideal para desarrollar soluciones que aborden problemas específicos en diversas industrias.

Por otro lado, en la vida cotidiana, el desarrollo de aplicaciones en Java y NetBeans también juega un papel importante. Desde aplicaciones de salud como el calculador de Índice de Masa Corporal (IMC) hasta herramientas de productividad personal, estas tecnologías permiten a los usuarios acceder a aplicaciones útiles y funcionales que mejoran su calidad de vida.

El desarrollo de aplicaciones en Java y NetBeans ofrece beneficios tangibles tanto en entornos profesionales como en situaciones cotidianas. Su versatilidad, portabilidad y eficiencia hacen que sean herramientas indispensables para aquellos que buscan crear soluciones innovadoras y funcionales en el mundo digital actual.

Referencias

Jiménez, S. (2023, August 21). Guía para crear aplicaciones Java eficientes y fáciles con NetBeans.

Aprender Java. <https://aprenderjava.net/base/guia-para-crear-aplicaciones-java-eficientes-y-faciles-con-netbeans/>



Enlace Github