



**TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA**

TÓPICOS AVANZADOS DE PROGRAMACIÓN

**PRACTICA 1 CREACIÓN DE UNA INTERFAZ GRÁFICA QUE
MANEJA EVENTOS BÁSICOS.**

**PRESENTA ESTUDIANTE DE LA CARRERA INGENIERÍA EN SISTEMAS
COMPUTACIONALES:**

ZARATE CARREÑO JOSÉ VALENTÍN

PROFESOR: HERNANDEZ ABREGO ANAYANSI

GRUPO: 4SA

HORA: 12:00- 1:00

Oaxaca de Juárez, Oax, 21 de Febrero de 2020.

COMPETENCIA A DESARROLLAR

Desarrolla programas para interactuar con el usuario de una manera amigable, utilizando GUI (Interfaz Gráfica de Usuario) manipuladas a través de eventos.

INTRODUCCIÓN

El desarrollo de programas con interfaz gráfica de usuario (GUI) comprende dos grandes temas, el diseño de la interfaz gráfica en sí y el control de los elementos que lo conforman a través de eventos. Esta práctica al ser la primera utiliza una pequeña pero útil aplicación para involucrar a los estudiantes en el ambiente de la GUI y la forma de controlar la funcionalidad mediante eventos básicos de tipo acción, que son los más utilizados.

El aspecto de diseño abarca la distribución de los elementos y sus propiedades. Para la distribución se hace uso de uno de los administradores de esquemas más utilizados denominado “FlowLayout” que realiza una distribución relativa, y se hace una segunda versión de la misma aplicación utilizando una distribución absoluta para establecer una comparación entre estas formas de distribuir los elementos. En ambas versiones se hace uso de los eventos de acción para implementar la funcionalidad de la aplicación.

MATERIAL Y EQUIPO NECESARIO

- Acceso al api de java (internet requerido) Apuntes del curso
- Una imagen que haga alusión al peso y talla de una persona.
- Equipo de cómputo: Laptop o PC
- Software: cualquier “IDE” de java, versión del jdk 1.8.

METODOLOGÍA

Se parte de una aplicación que sirve como base para que el estudiante experimente los aspectos del software relacionados con el diseño de una GUI y manejo de eventos. Primero se describe la aplicación a realizar y luego durante su desarrollo se construye el software haciendo que el estudiante complemente parte del código para verificar si está consiguiendo la competencia referida en la práctica.

DESCRIPCIÓN

La aplicación GUI propuesta es un programa que sirve para calcular el índice de masa corporal (IMC) y el índice de cintura-cadera (ICC), los cuales se consideran como indicadores básicos para conocer si la salud de una persona está en riesgo de padecer algún problema debido a su peso o medidas de su cintura. Estos indicadores se calculan de la siguiente manera:

$$\text{IMC} = \text{peso} / \text{altura}^2$$

Donde el peso de una persona es dado en Kilogramos y la altura (estatura de una persona) es dada en metros.

Los rangos del IMC a considerar para personas adultas entre 20 y 63 años son:

Menor de 18.5 => Peso bajo entre 18.5-24.9=> Peso normal entre 25-29.9 => Sobre
Peso de 30 o más => Obesidad

Fuente: Organización Mundial de la Salud (OMS), 2007.

ICC = cintura / cadera.

Las medidas son dadas en centímetros. Rango de valores del ICC a considerar:

Hombres	Mujeres	Riesgo cardiovascular
De 0.95 o menos	De .80 o menos	Bajo
Mayor a 0.95 y menor a 1.0	Mayor a 80 y menor a 0.85	Medio
De 1.0 o más	De 0.85 o más	Alto

DESARROLLO DE LA PRÁCTICA

Preparativos

- I. Dentro de tu IDE, crea un proyecto de java simple y llámalo tap_practica_01, dentro de Source Packages, crea dos paquetes, uno puedes nombrarlo como “indicadoresSalud” y otro como “imagenes”.
 - a) Dentro del paquete “indicadoresSalud” crea una clase java con nombre IndicadoresSalud.
 - b) Abre el explorador de archivos y ubícate en la carpeta donde tengas tu imagen, selecciónala, activa el menú contextual y selecciona Copiar. Ubícate en el IDE donde tengas el paquete “imagenes”, activa el menú contextual y selecciona Pegar.
- II. Creación de los elementos GUI de la clase IndicadoresSalud.
- III. Prueba de ejecución para comprobar cómo se distribuyen los elementos en “IndicadoresSalud”
 - a. Se crea una clase Prueba y dentro de ella agrega el método main dentro de cual crea un objeto de la clase “IndicadoresSalud”
- IV. Agregando funcionalidad: Cálculo de IMC y al ICC.
 - a) Se implementa la interface ActionListener en la clase IndicadoresSalud.
 - b) Implementa su método de esta interface: public void actionPerformed(ActionEvent ae){}.
- V. Finalmente se agregan las validaciones del programa.

CÓDIGO

```
1  package indicadoresSalud;
2
3  import java.awt.Container;
4  import java.awt.FlowLayout;
5  import java.awt.event.ActionEvent;
6  import java.awt.event.ActionListener;
7  import java.net.URL;
8  import java.text.DecimalFormat;
9  import javax.swing.*.*;
10
11 public class IndicadoresSalud extends JFrame {
12
13     private JButton ciMC;
14     private JButton ciCC;
15     private JButton borrar;
16     private JLabel resultadoIMC;
17     private JLabel resultadoICC;
18     private JTextField edad;
19     private JTextField estatura;
20     private JTextField peso;
21     private JTextField cadera;
22     private JTextField cintura;
23     private JRadioButton hombre;
24     private JRadioButton mujer;
25     private Container panel;
26
27     public IndicadoresSalud() {
28         super("IMC-ICC");
29         panel = getContentPane();
30         panel.setLayout(new FlowLayout());
31
32         String patch = "/imagenes/IMC2.png";
33         URL url = this.getClass().getResource(patch);
34         ImageIcon icon = new ImageIcon(url);
35
36         JLabel presentacion = new JLabel("INDICADORES BASICOS DE RIESGOS A LA SALUD",
37             icon, SwingConstants.CENTER);
38         this.ciMC = new JButton("Calcular IMC");
39         this.ciCC = new JButton("Calcular ICC");
40         this.edad = new JTextField(4);
41         this.estatura = new JTextField(4);
42         this.peso = new JTextField(4);
43         this.resultadoIMC = new JLabel("_____");
44         this.cintura = new JTextField(4);
45         this.cadera = new JTextField(4);
46         this.hombre = new JRadioButton("Hombre");
47         this.mujer = new JRadioButton("Mujer");
48         this.resultadoICC = new JLabel("_____");
49         this.borrar = new JButton("Borrar Datos");
50         JLabel tEdad = new JLabel(" Proporciona tu edad (>19):");
51         JLabel tIMC = new JLabel("CÁLCULO DEL ÍNDICE DE MASA CORPORAL (IMC)");
52         JLabel testatura = new JLabel("Estura (cms) : ");
53         JLabel tpeso = new JLabel(" Peso(kgs.) : ");
54         JLabel tiCC = new JLabel("CÁLCULO DEL INDICE DE CINTURA-CADERA (I-C-C)");
55         JLabel tcintura = new JLabel(" Cintura (cms) : ");
56         JLabel tcadera = new JLabel(" Cadera (cms) : ");
57         JLabel sexo = new JLabel("      Sexo : ");
58     }
59 }
```

```

58
59     presentacion.setHorizontalTextPosition(SwingConstants.CENTER);
60     presentacion.setVerticalTextPosition(SwingConstants.TOP);
61     panel.add(presentacion);
62     panel.add(tEdad);
63     panel.add(edad);
64     panel.add(tIMC);
65     panel.add(testatura);
66     panel.add(estatura);
67     panel.add(tpeso);
68     panel.add(peso);
69     panel.add(ciMC);
70     panel.add(resultadoIMC);
71     panel.add(tiCC);
72     panel.add(tcintura);
73     panel.add(cintura);
74     panel.add(tcadera);
75     panel.add(cadera);
76     panel.add(sexo);
77     panel.add(hombre);
78     panel.add(mujer);
79     panel.add(cICC);
80     panel.add(resultadoICC);
81     panel.add(borrar);
82     ButtonGroup option = new ButtonGroup();
83     option.add(hombre);
84     option.add(mujer);
85
86     setSize(320, 510);
87     setVisible(true);
88     setDefaultCloseOperation(EXIT_ON_CLOSE);
89

```

```

90     ActionListener(new ActionListener() {
91         rride
92         ic void actionPerformed(ActionEvent ae) {
93
94             DecimalFormat df = new DecimalFormat("#.00");
95             String resultado = "";
96
97             JButton boton = (JButton) ae.getSource();
98
99             if (edad.getText().isEmpty() || peso.getText().isEmpty() || estatura.getText().isEmpty()) {
100
101                 JOptionPane.showMessageDialog(null, "¡Los campos estan vacios!", "ERROR", JOptionPane.WARNING_MESSAGE);
102
103             } else {
104
105                 if (boton == ciMC) {
106                     if (isNumeric(edad.getText()) && isNumeric(peso.getText()) && isNumeric(estatura.getText())) {
107                         Double mtsCms = Double.parseDouble(estatura.getText()) / 100.0;
108                         Double pesoKgs = Double.parseDouble(peso.getText());
109                         if (mtsCms < 1.40 || mtsCms > 2 || pesoKgs < 40 || pesoKgs > 160) {
110                             JOptionPane.showMessageDialog(null, "Se requiere valores dentro del rango");
111                         } else {
112                             Double uIMC = pesoKgs / Math.pow(mtsCms, 2);
113                             resultado = String.valueOf(df.format(uIMC));
114                             resultadoIMC.setText(resultado);
115
116                             if (Double.parseDouble(resultado) < 18.5) {
117                                 resultadoIMC.setText(resultado + " => bajo de Peso");
118                             } else if (Double.parseDouble(resultado) >= 18.5 && Double.parseDouble(resultado) <= 24.9) {
119                                 resultadoIMC.setText(resultado + " => Peso normal ");
120                             } else if (Double.parseDouble(resultado) >= 25 && Double.parseDouble(resultado) <= 29.9) {
121                                 resultadoIMC.setText(resultado + " => Sobre Peso ");
122

```

```

122         } else {
123             resultadoIMC.setText(resultado + " => Obesidad");
124         }
125     }
126 }
127 } else {
128     JOptionPane.showMessageDialog(null, "Se requiere valores numericos");
129 }
130 }
131 }
132 }
133 }
134 }
135 }
136 );
137

```

```

138 cICC.addActionListener(new ActionListener() {
139     @Override
140     public void actionPerformed(ActionEvent ae) {
141
142         DecimalFormat df = new DecimalFormat("#.00");
143         JButton boton2 = (JButton) ae.getSource();
144         String resICC = "";
145         if (cintura.getText().toString().isEmpty() || cadera.getText().toString().isEmpty()) {
146             JOptionPane.showMessageDialog(boton2, "¡Los campos estan vacios!", "ERROR",
147                 JOptionPane.WARNING_MESSAGE);
148         } else {
149             if (boton2 == cICC) {
150                 if (isNumeric(cintura.getText()) && isNumeric(cadera.getText())) {
151                     Double ICC = Double.parseDouble(cintura.getText()) / Double.parseDouble(cadera.getText());
152                     resICC = String.valueOf(df.format(ICC));
153
154                     if (hombre.isSelected()) {
155                         if (Double.parseDouble(resICC) < 0.95) {
156                             resultadoICC.setText(resICC + " => Riesgo Cardiovascular BAJO");
157                         } else if (Double.parseDouble(resICC) >= 0.95 && Double.parseDouble(resICC) <= 1.0) {
158                             resultadoICC.setText(resICC + " => Riesgo Cardiovascular MEDIO");
159                         } else {
160                             resultadoICC.setText(resICC + " => Riesgo Cardiovascular ALTO");
161                         }
162                     }
163                 }
164             } else {
165                 if (Double.parseDouble(resICC) < 0.80) {
166

```

```

170                     resultadoICC.setText(resICC + " => Riesgo Cardiovascular BAJO");
171                 } else if (Double.parseDouble(resICC) >= 0.80 && Double.parseDouble(resICC) <= 0.85) {
172                     resultadoICC.setText(resICC + " => Riesgo Cardiovascular MEDIO");
173                 } else {
174                     resultadoICC.setText(resICC + " => Riesgo Cardiovascular ALTO");
175                 }
176             }
177         }
178     }
179 } else {
180     JOptionPane.showMessageDialog(null, "Se requiere valores numericos");
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 });

```

```

190     borrar.addActionListener(new ActionListener() {
191         @Override
192         public void actionPerformed(ActionEvent ae) {
193             JButton btnDelete = (JButton) ae.getSource();
194             if (btnDelete == borrar) {
195                 edad.setText(null);
196                 peso.setText(null);
197                 estatura.setText(null);
198                 resultadoIMC.setText("_____");
199                 cintura.setText(null);
200                 cadera.setText(null);
201                 hombre.setSelected(false);
202                 mujer.setSelected(false);
203                 resultadoICC.setText("_____");
204             }
205         }
206     });
207 }
208

```

```

210     private static boolean isNumeric(String cadena) {
211         try {
212             Integer.parseInt(cadena);
213             return true;
214         } catch (NumberFormatException nfe) {
215             return false;
216         }
217     }
218 }
219

```

```


1
2     package indicadoresSalud;
3
4     public class Prueba {
5
6         Run | Debug
7         public static void main(String[] args) {
8
9             IndicadoresSalud e = new IndicadoresSalud();
10        }
11
12    }
13

```


RESULTADOS DE PRUEBA IMC

IMC-ICC

INDICADORES BASICOS DE RIESGOS A LA SALUD



Proporciona tu edad (>19):


CÁLCULO DEL ÍNDICE DE MASA CORPORAL (IMC)

Estura (cms): Peso(kgs.):

17.30 => bajo de Peso

IMC-ICC

INDICADORES BASICOS DE RIESGOS A LA SALUD



Proporciona tu edad (>19):


CÁLCULO DEL ÍNDICE DE MASA CORPORAL (IMC)

Estura (cms): Peso(kgs.):

23.05 => Peso normal

IMC-ICC

INDICADORES BASICOS DE RIESGOS A LA SALUD



Proporciona tu edad (>19):


CÁLCULO DEL ÍNDICE DE MASA CORPORAL (IMC)

Estura (cms): Peso(kgs.):

25.71 => Sobre Peso

IMC-ICC

INDICADORES BASICOS DE RIESGOS A LA SALUD



Proporciona tu edad (>19):

CÁLCULO DEL ÍNDICE DE MASA CORPORAL (IMC)

Estura (cms): Peso(kgs.):

31.02 => Obesidad

RESULTADOS DE PRUEBA ICC

CÁLCULO DEL INDICE DE CINTURA-CADERA (I-C-C)

Cintura (cms): Cadera (cms):

Sexo: ☒ Hombre ☐ Mujer

.41 => Riesgo Cardiovascular BAJO

CÁLCULO DEL INDICE DE CINTURA-CADERA (I-C-C)

Cintura (cms): Cadera (cms):

Sexo: ☒ Hombre ☐ Mujer

.99 => Riesgo Cardiovascular MEDIO

CÁLCULO DEL INDICE DE CINTURA-CADERA (I-C-C)

Cintura (cms): Cadera (cms):

Sexo: ☒ Hombre ☐ Mujer

1.24 => Riesgo Cardiovascular ALTO

CÁLCULO DEL INDICE DE CINTURA-CADERA (I-C-C)

Cintura (cms): Cadera (cms):

Sexo: ☐ Hombre ☒ Mujer

.41 => Riesgo Cardiovascular BAJO

CÁLCULO DEL INDICE DE CINTURA-CADERA (I-C-C)

Cintura (cms): Cadera (cms):

Sexo: ☐ Hombre ☒ Mujer

.99 => Riesgo Cardiovascular ALTO

CÁLCULO DEL INDICE DE CINTURA-CADERA (I-C-C)

Cintura (cms): Cadera (cms):

Sexo: ☐ Hombre ☒ Mujer

.80 => Riesgo Cardiovascular MEDIO

Conclusiones sobre los principales aspectos a considerar en una aplicación GUI

En conclusión el conocimiento de estos puntos clave, nos permitirán enfocarnos mejor al estudio de la materia. También Las Interfaces de usuario, como vínculo de inmersión del hombre en el entorno de trabajo tecnológico actual, realzan su importancia en el desarrollo de nuevos productos, más eficaces, eficientes e interactivos, que es lo que el mercado demanda. Los puntos, cómo los históricos y evolutivos, deben ser abordados de manera más investigativa, recordemos que "conocer el pasado nos proyecta al futuro". Otras puntualizaciones de clasificación obligarán a que investiguemos y propongamos, nuevas distribuciones clasificatorias, útiles a futuro en una carrera de desarrollo de software.