

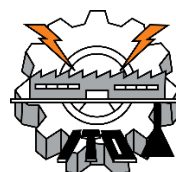


**TECNOLÓGICO NACIONAL DE MÉXICO**  
**INSTITUTO TECNOLÓGICO DE OAXACA**

**ASIGNATURA:** TOPICOS AVANZADOS DE PROGRAMACIÓN  
**CATEDRÁTICO:** HERNANDEZ ABREGO ANAYANSI CRISTINA  
**ALUMNO:** GARCÍA GARCÍA JOSÉ ÁNGEL

**PRACTICA 3- Diseño e implementación de una aplicación con  
GUI y eventos diversos.**

**OAXACA DE JÚAREZ, OAX, 26/FEBRERO/2020**



## Índice

CÓDIGO.....	3
UN CUADRO DE LOS CASOS DE PRUEBA .....	9
RESULTADOS DE LOS CASOS DE PRUEBA .....	11
BITÁCORA DE SEGUIMIENTO CON LAS ANOTACIONES SOLICITADAS DURANTE EL DESARROLLO DE LA PRÁCTICA .....	17
CONCLUSIONES SOBRE LOS PRINCIPALES ASPECTOS A CONSIDERAR EN UNA APLICACIÓN GUI.....	18



## CÓDIGO

Le dejo el código principal de la GUI.

```
package Alimentos;
/*
 * @author García García José Ángel
 */
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Component;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;

import javax.swing.*.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;

public class ManejoAlimento extends JFrame implements
ActionListener,ChangeListener,ItemListener{
    private Alimentos alimentos[];
    private final Color COLORES[] = {Color.GREEN, Color.YELLOW, Color.ORANGE,
Color.WHITE, Color.MAGENTA};
    private final String Tipo_Alimento[] =
{"Verduras","Frutas","Cereales","Leguminosas","Origen Animal"};
    private final int KCAL_TIPO ALIM[] = {25,60,70,120,40};
    private JSlider racTipoAlim[];
    private JLabel kCalTipoAlim[];
    private JPanel pKcalTipoAlim[];
    private JButton verificar;
    private JLabel kilocalorias;
    private final String TIEMPOSDECOMIDA[] = {"Desayuno","Alumerzo","Cena"};
    private JCheckBox eleccionTiempo[];
    private DefaultListModel[] modelosListas;
    private JList tiempos[];
    private JComboBox alimento;
    private JButton aceptar;
    private JLabel tipoAli;
    JLabel[] numAlimTiem;
    private int[] numAlimTie;

    public ManejoAlimento() {
        super("Practica 2");
        Container panel = getContentPane();
        JTabbedPane panelPrincipal = new JTabbedPane();
        alimentos = cargarDatos();
        panelPrincipal.addTab("Conociendo alimentos", panelConociendo());
        panelPrincipal.addTab("Eleccion de alimentos", panelAlimentos());
        panel.add(panelPrincipal);
        this.setVisible(true);
    }
}
```



```

this.setDefaultCloseOperation(EXIT_ON_CLOSE);
this.pack();
this.setResizable(false);
this.setSize(new Dimension(600, 330));
this.setLocationRelativeTo(null);
}

private Alimentos[] cargarDatos() {
    alimentos = new Alimentos[22];
    alimentos[0] = new Alimentos("Zanahoria", 0);
    alimentos[1] = new Alimentos("Lechuga", 0);
    alimentos[2] = new Alimentos("Espinaca", 0);
    alimentos[3] = new Alimentos("Acelgas", 0);
    alimentos[4] = new Alimentos("Tomate", 0);
    alimentos[5] = new Alimentos("Platano", 1);
    alimentos[6] = new Alimentos("Naranja", 1);
    alimentos[7] = new Alimentos("Pepino", 1);
    alimentos[8] = new Alimentos("Manzana", 1);
    alimentos[9] = new Alimentos("Pera", 1);
    alimentos[10] = new Alimentos("Sandia", 1);
    alimentos[11] = new Alimentos("Mango", 1);
    alimentos[12] = new Alimentos("Cereza", 1);
    alimentos[13] = new Alimentos("Piña", 1);
    alimentos[14] = new Alimentos("Pollo", 4);
    alimentos[15] = new Alimentos("Arroz", 2);
    alimentos[16] = new Alimentos("Uva", 1);
    alimentos[17] = new Alimentos("Ciruela", 1);
    alimentos[18] = new Alimentos("Coco", 1);
    alimentos[19] = new Alimentos("Mandarina", 1);
    alimentos[20] = new Alimentos("Toronja", 1);
    alimentos[21] = new Alimentos("Frijoles", 3);
    return alimentos;
}

private JPanel panelConociendo() {
    JPanel conocer = new JPanel(new BorderLayout());
    JPanel panelSuperior = new JPanel(new BorderLayout());
    JPanel panelCentro = new JPanel(new GridLayout(1, 4, 10, 0));
    JPanel panelSur = new JPanel();
    JPanel panelTitulo = new JPanel(new GridLayout(0, 1, 0, 15));
    JPanel panelEncabezado = new JPanel(new FlowLayout());
    JLabel enc1 = new JLabel("CONOCIMIENTO DE LAS KILOCALORIAS APORTADAS POR
RACIONES POR TIPO DE ALIMENTO");
    JLabel enc2 = new JLabel("Elige el numero de las raciones decada tipo de
alimento que consideres que debes consumir");
    panelTitulo.add(enc1);
    panelTitulo.add(enc2);
    panelEncabezado.add(panelTitulo);
    panelSuperior.add(panelEncabezado, BorderLayout.CENTER);
    pKcalTipoAlim = new JPanel[Tipo_Alimento.length];
    JPanel pGpoTipoAlim = new JPanel(new GridLayout());
    racTipoAlim = new JSlider[Tipo_Alimento.length];
    kcalTipoAlim = new JLabel[Tipo_Alimento.length];
    verificar = new JButton("Verifica proporcion");
    verificar.addActionListener(this);
    kilocalorias = new JLabel("Total de kilocalorais : 0");
}

```



```

panelSur.add(kilocalorias);
panelSur.add(verificar);
for(int ta = 0; ta < Tipo_Alimento.length;ta++) {
    JPanel pDatos = new JPanel(new BorderLayout());
    JLabel tit = new JLabel(Tipo_Alimento[ta]);
    tit.setHorizontalAlignment(SwingConstants.CENTER);
    pKcalTipoAlim[ta] = new JPanel();
    racTipoAlim[ta] = new JSlider();
    racTipoAlim[ta].addChangeListener(this);
    racTipoAlim[ta].setOrientation(JSlider.VERTICAL);
    racTipoAlim[ta].setPaintTicks(true);
    racTipoAlim[ta].setPaintLabels(true);
    racTipoAlim[ta].setMinimum(0);
    racTipoAlim[ta].setMaximum(10);
    racTipoAlim[ta].setValue(1);
    racTipoAlim[ta].setMajorTickSpacing(2);
    racTipoAlim[ta].setMinorTickSpacing(1);
    kCalTipoAlim[ta] = new JLabel(Tipo_Alimento[ta]);
    kCalTipoAlim[ta].setBackground(COLORES[ta]);
    kCalTipoAlim[ta].setOpaque(true);
    kCalTipoAlim[ta].setVerticalAlignment(SwingConstants.NORTH);
    kCalTipoAlim[ta].setHorizontalAlignment(SwingConstants.CENTER);
    pKcalTipoAlim[ta].setLayout(null);
    pKcalTipoAlim[ta].setPreferredSize(new Dimension(60, 0));
    pKcalTipoAlim[ta].add(kCalTipoAlim[ta]);
    pDatos.add(tit, BorderLayout.NORTH);
    pDatos.add(racTipoAlim[ta], BorderLayout.WEST);
    pDatos.add(pKcalTipoAlim[ta], BorderLayout.CENTER);
    pGpoTipoAlim.add(pDatos);
}

panelCentro.add(pGpoTipoAlim);
conocer.add(panelSur, BorderLayout.SOUTH);
conocer.add(panelSuperior, BorderLayout.NORTH);
conocer.add(panelCentro, BorderLayout.CENTER);
return conocer;
}

private JPanel panelAlimentos() {
    JPanel pAlimentos = new JPanel();
    pAlimentos.setLayout(new BorderLayout());
    JPanel panelCentro = new JPanel();
    JPanel panelCheck = new JPanel();
    panelCheck.setLayout(new GridLayout(1,0));
    panelCentro.setLayout(new BorderLayout());
    JPanel panelSeleccion = new JPanel();
    JLabel titulo = new JLabel("ELECCION DE ALIMENTOS Y ASIGNACION A UN TIEMPO");
    titulo.setHorizontalAlignment(JLabel.CENTER);
    JLabel tAlimento = new JLabel("Elige Alimento ");
    tipoAli = new JLabel("");
    alimento = new JComboBox(alimentos);
    alimento.addItemListener(this);
    tipoAli.setText(Tipo_Alimento[0]);
    JPanel panelTiempos = new JPanel();
    panelTiempos.add(new JPanel());
    panelSeleccion.add(tAlimento);

```



```

panelSeleccion.add(alimento);
panelSeleccion.add(tipoAli);
panelCentro.add(panelSeleccion, BorderLayout.NORTH);
panelCentro.add(creaPanelTiempos(), BorderLayout.CENTER);
pAlimentos.add(titulo, BorderLayout.NORTH);
pAlimentos.add(panelCentro, BorderLayout.CENTER);
return pAlimentos;
}

```

```

private JPanel creaListas() {
    JPanel pListas = new JPanel();
    JPanel listas = new JPanel();
    JPanel numAlim = new JPanel();
    tiempos = new JList[TIEMPOSDECOMIDA.length];
    numAlimTiem = new JLabel[TIEMPOSDECOMIDA.length];
    modelosListas = new DefaultListModel[TIEMPOSDECOMIDA.length];
    GridLayout dList = new GridLayout(0, TIEMPOSDECOMIDA.length+1, 5, 0);
    pListas.setLayout(new BorderLayout());
    listas.setLayout(dList);
    numAlim.setLayout(dList);
    for (int t = 0; t < TIEMPOSDECOMIDA.length; t++) {
        modelosListas[t] = new DefaultListModel();
        tiempos[t] = new JList(modelosListas[t]);
        listas.add(new JScrollPane(tiempos[t]));
    }
    for (int t = 0; t < TIEMPOSDECOMIDA.length; t++){
        numAlimTiem[t] = new JLabel();
        numAlim.add(numAlimTiem[t]);
    }
    pListas.add(listas, BorderLayout.NORTH);
    pListas.add(numAlim, BorderLayout.SOUTH);

    return pListas;
}

```

```

public JPanel creaPanelTiempos() {
    JPanel tiemposAlimento = new JPanel();
    tiemposAlimento.setLayout(new BorderLayout());
    JLabel indicacion = new JLabel("Selecciona el tiempo cuado deseas
consumirlo");
    indicacion.setHorizontalAlignment(JLabel.CENTER);
    tiemposAlimento.add(indicacion, BorderLayout.NORTH);
    JPanel eleccion = new JPanel();
    aceptar = new JButton("Agregar");
    aceptar.addActionListener(this);
    eleccionTiempo = new JCheckBox[TIEMPOSDECOMIDA.length+1];
    numAlimTie = new int[eleccionTiempo.length-1];
    eleccion.setLayout(new GridLayout(0, 4));
    for(int t = 0; t < eleccionTiempo.length-1; t++) {
        eleccionTiempo[t] = new JCheckBox(TIEMPOSDECOMIDA[t]);
        eleccionTiempo[t].addActionListener(this);
        eleccion.add(eleccionTiempo[t]);
        numAlimTie[t] = 0;
    }
    eleccionTiempo[eleccionTiempo.length-1] = new JCheckBox("En todos");
}

```



```

        eleccion.add(eleccionTiempo[eleccionTiempo.length-1]);
        eleccionTiempo[eleccionTiempo.length-1].addActionListener(this);
        aceptar.setBounds(445,93, 80, 20);
        tiemposAlimento.add(aceptar);
        tiemposAlimento.add(eleccion, BorderLayout.CENTER);
        tiemposAlimento.add(creaListas(), BorderLayout.SOUTH);
        return tiemposAlimento;
    }
    @Override
    public void actionPerformed(ActionEvent ae) {
        Object prod = ae.getSource();
        if(prod == verificar) {
            String mensaje = "La verdura sin restriccion de cantidad\nLa fruta sin
restricciones de cantidadcon variedad de colores\n"
                + "Los creales deben ser suficientes\nLas legumbres deben
ser suficientes combinadas con cereales\n"
                + "Productos de origen animal deben ser consumidos en
poca cantidad ";
            JOptionPane.showMessageDialog(this, mensaje);
        }if(prod == eleccionTiempo[0] || prod == eleccionTiempo[1] || prod ==
eleccionTiempo[2] || prod == eleccionTiempo[3]) {
            boolean seleccion = true;
            for (int i = 0; i < eleccionTiempo.length; i++) {
                if(prod == eleccionTiempo[3]) {
                    eleccionTiempo[i].setSelected(true);
                }
                seleccion = seleccion && eleccionTiempo[i].isSelected();
                eleccionTiempo[eleccionTiempo.length-1].setSelected(seleccion);
            }
        }if (prod == aceptar) {
            Alimentos aliSel = (Alimentos) alimento.getSelectedItemAt();
            for (int t = 0; t <eleccionTiempo.length-1; t++) {
                if(eleccionTiempo[t].isSelected() &&
!modelosListas[t].contains(aliSel)){
                    modelosListas[t].addElement(aliSel);
                    numAlimTiem[t].setText("Num. Ali.: " +
(++numAlimTie[t]));
                }
            }
        }
    }

    @Override
    public void stateChanged(ChangeEvent e) {
        int totalkc = 0;
        try {
            for(int ta = 0; ta < COLORES.length; ta++) {
                int h = (int)(0.5*(racTipoAlim[ta].getValue()*KCAL_TIPO_ALIM[ta]));
                int base = racTipoAlim[ta].getHeight()-
racTipoAlim[ta].getValue()*KCAL_TIPO_ALIM[ta] + h;
                kCalTipoAlim[ta].setBounds(0, base,60,h);
                kCalTipoAlim[ta].setText(""+racTipoAlim[ta].getValue()*KCAL_TIPO_ALIM[ta]+"KC");
                totalkc += racTipoAlim[ta].getValue()*KCAL_TIPO_ALIM[ta];
            }
        }
    }

```



```

        }catch(Exception ex) {}
        kilocalorias.setText("TotalKC : "+ totalkc + " ");
        this.repaint();
    }

    @Override
    public void itemStateChanged(ItemEvent e) {
        JComboBox aliSel = (JComboBox) e.getSource();
        int t = ((Alimentos)aliSel.getSelectedItem()).getTipo();
        tipoAli.setText(Tipo_Alimento[t]);
    }
}

```

## Clase Alimento

```

package Alimentos;

public class Alimentos {

    private String alimento;
    private int tipo;

    public Alimentos(String alimento,int tipo) {
        this.alimento = alimento;
        this.tipo = tipo;
    }
    public int getTipo() {
        return tipo;
    }
    public String getAlimento() {
        return alimento;
    }
    @Override
    public String toString() {
        return alimento;
    }
    public boolean equals(Alimentos ali) {
        return alimento.equals(ali.alimento) && tipo == ali.getTipo();
    }
}

```





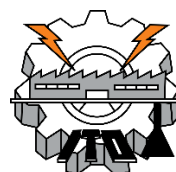
## UN CUADRO DE LOS CASOS DE PRUEBA

- **Tabla de casos de prueba**

Contiene una simulación con un total de 5 personas, que desean conocer las calorías aportadas por los alimentos que consumen en su día, también se muestra la elección de sus alimentos en cada tiempo de comida del día.

Además, se adjunta la imagen del programa con el cálculo por cada persona.

Id	Raciones de verduras	Raciones de frutas	Raciones de cereales	Raciones de leguminosas	Raciones de origen animal	Elección de elementos
01	1	0	2	1	3	<b>Desayuno</b> <ul style="list-style-type: none"> <li>- Lechuga</li> <li>- Pollo</li> <li>- Arroz</li> </ul> <b>Almuerzo</b> <ul style="list-style-type: none"> <li>- Frijoles</li> <li>- Pollo</li> </ul> <b>Cena</b> <ul style="list-style-type: none"> <li>- Pollo</li> <li>- Arroz</li> </ul>
02	4	3	1	0	2	<b>Desayuno</b> <ul style="list-style-type: none"> <li>- Lechuga</li> <li>- Manzana</li> <li>- Zanahoria</li> <li>- Pollo</li> </ul> <b>Almuerzo</b> <ul style="list-style-type: none"> <li>- Pollo</li> <li>- Arroz</li> <li>- Pera</li> </ul> <b>Cena</b> <ul style="list-style-type: none"> <li>- Espinaca</li> <li>- Acelgas</li> <li>- Sandia</li> </ul>
03	5	6	0	1	1	<b>Desayuno</b> <ul style="list-style-type: none"> <li>- Lechuga</li> <li>- Acelgas</li> <li>- Tomate</li> <li>- Manzana</li> <li>- Piña</li> </ul> <b>Almuerzo</b> <ul style="list-style-type: none"> <li>- Cereza</li> <li>- Plátano</li> <li>- Frijoles</li> <li>- Pollo</li> </ul>



						<b>Cena</b> <ul style="list-style-type: none"> <li>- Lechuga</li> <li>- Tomate</li> <li>- Pera</li> <li>- Manzana</li> </ul>
04	2	3	2	2	3	<b>Desayuno</b> <ul style="list-style-type: none"> <li>- Acelgas</li> <li>- Pollo</li> <li>- Arroz</li> <li>- Manzana</li> </ul> <b>Almuerzo</b> <ul style="list-style-type: none"> <li>- Lechuga</li> <li>- Pollo</li> <li>- Frijoles</li> <li>- Pera</li> </ul> <b>Cena</b> <ul style="list-style-type: none"> <li>- Mango</li> <li>- Pollo</li> <li>- Frijoles</li> </ul>
05	3	1	2	2	3	<b>Desayuno</b> <ul style="list-style-type: none"> <li>- Pollo</li> <li>- Tomate</li> <li>- Lechuga</li> <li>- Arroz</li> </ul> <b>Almuerzo</b> <ul style="list-style-type: none"> <li>- Manzana</li> <li>- Pollo</li> <li>- Frijoles</li> <li>- Arroz</li> </ul> <b>Cena</b> <ul style="list-style-type: none"> <li>- Frijoles</li> <li>- Pollo</li> </ul>



## RESULTADOS DE LOS CASOS DE PRUEBA

- Las imágenes siguientes son los resultados de cada persona, identificadas por su id.

Practica 2

Conociendo alimentos | **Eleccion de alimentos**

CONOCIMIENTO DE LAS KILOCALORIAS APORTADAS POR RACIONES POR TIPO DE ALIMENTO

Elige el numero de las raciones decada tipo de alimento que consideres que debes consumir

Verduras	Frutas	Cereales	Leguminosas	Origen Animal
10	10	10	10	10
8	8	8	8	8
6	6	6	6	6
4	4	4	4	4
2	2	2	2	2
0	0	0	0	0

25KC

140KC

120KC

120KC

TotalKC : 405

Verifica proporcion

Ilustración 1- Calorías de 01

Practica 2

Conociendo alimentos | **Eleccion de alimentos**

ELECCION DE ALIMENTOS Y ASIGNACION A UN TIEMPO

Elige Alimento **Frijoles** Leguminosas

Selecciona el tiempo cuado deseas consumirlo

☐ Desayuno ☒ Aluerzo ☐ Cena ☐ En todos

Lechuga	Pollo	Pollo
Pollo	Frijoles	Arroz
Arroz		

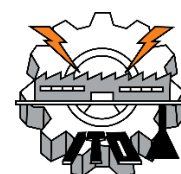
Agregar

Num. Ali.: 3

Num. Ali.: 2

Num. Ali.: 2

Ilustración 2Elección de 01



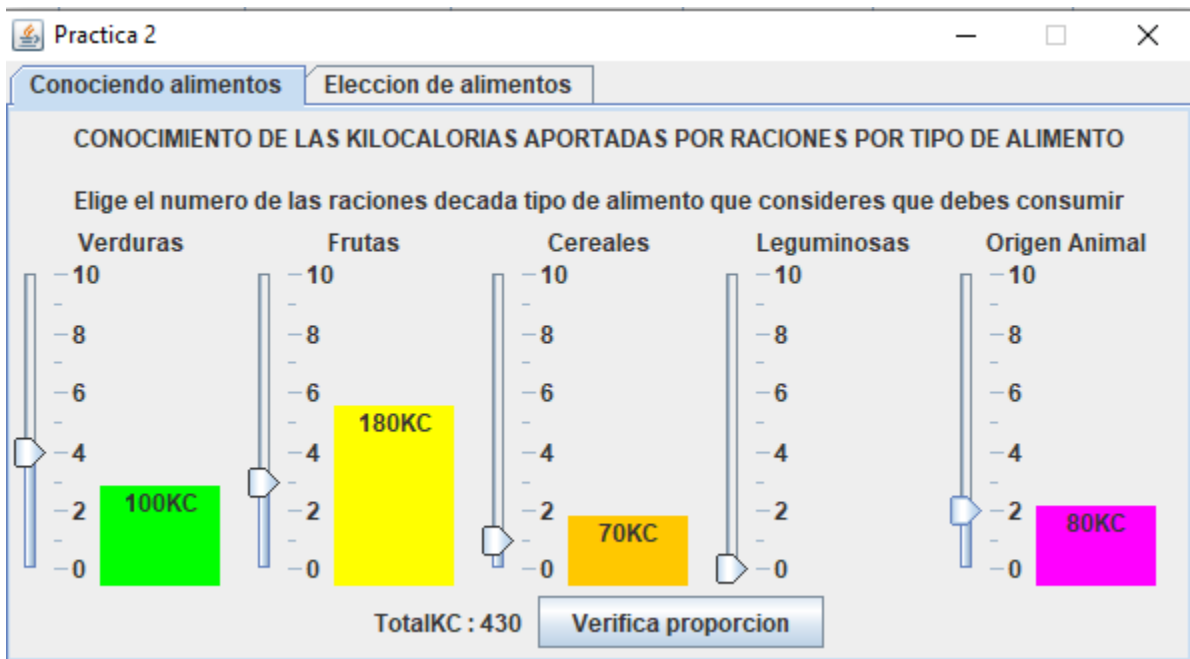


Ilustración 3 Calorías de 02

Practica 2

Conociendo alimentos | Eleccion de alimentos

### ELECCION DE ALIMENTOS Y ASIGNACION A UN TIEMPO

Elige Alimento: Sandia ▼ Frutas

Selecciona el tiempo cuando deseas consumirlo

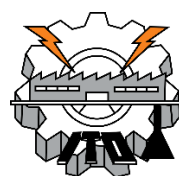
☐ Desayuno ☐ Alumerzo ☒ Cena ☐ En todos

Lechuga	Pollo	Espinaca
Manzana	Arroz	Acelgas
Zanahoria	Pera	Sandia
Pollo		

Num. Ali.: 4      Num. Ali.: 3      Num. Ali.: 3

Agregar

Ilustración 4 Elección de 02



Practica 2

Conociendo alimentos | Eleccion de alimentos

CONOCIMIENTO DE LAS KILOCALORIAS APORTADAS POR RACIONES POR TIPO DE ALIMENTO

Elige el numero de las raciones decada tipo de alimento que consideres que debes consumir

Verduras	Frutas	Cereales	Leguminosas	Origen Animal
10 8 6 4 2 0	10 8 6 4 2 0	10 8 6 4 2 0	10 8 6 4 2 0	10 8 6 4 2 0
125KC			120KC	40KC

TotalKC : 645

Verifica proporcion

Ilustración 5 Calorías de 03

Practica 2

Conociendo alimentos | Eleccion de alimentos

ELECCION DE ALIMENTOS Y ASIGNACION A UN TIEMPO

Elige Alimento  Origen Animal

Selecciona el tiempo cuando deseas consumirlo

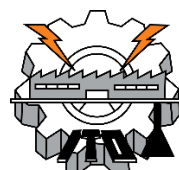
☐ Desayuno ☐ Alumerzo ☒ Cena ☐ En todos

Lechuga Acelgas Tomate Manzana Piña	Cereza Platano Frijoles Pollo	Lechuga Tomate Pera Pollo
---	--	------------------------------------

Num. Ali.: 5 Num. Ali.: 4 Num. Ali.: 4

Agregar

Ilustración 6 Elección de 03



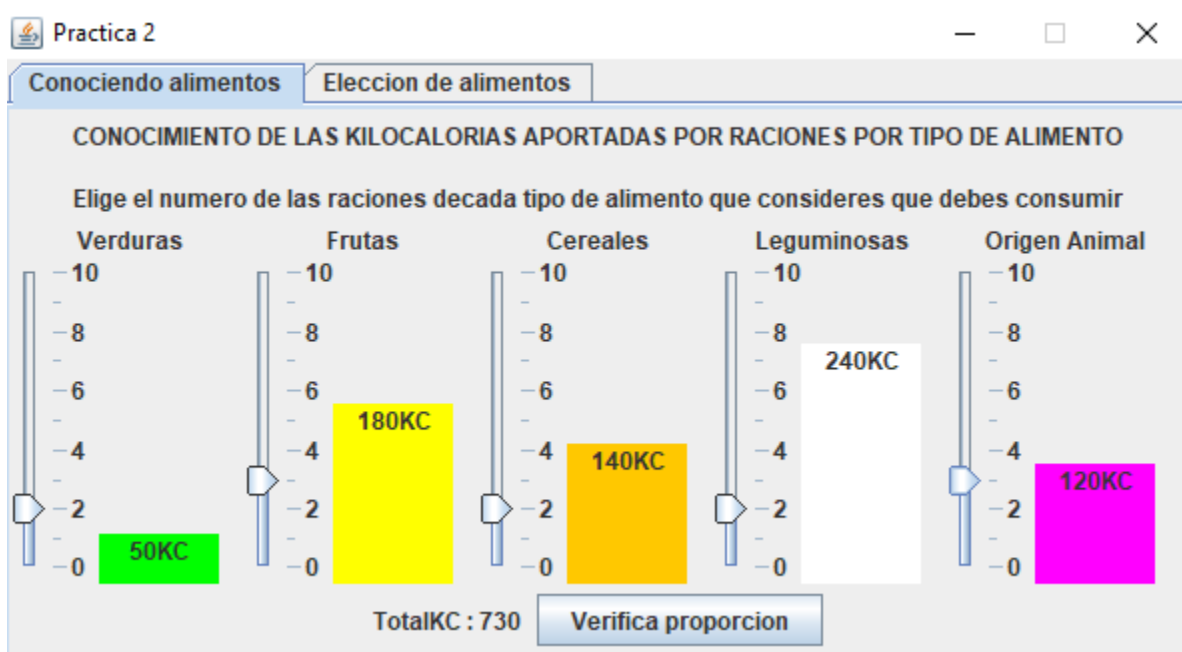


Ilustración 7 Calorías de 04

Practica 2

Conociendo alimentos | Eleccion de alimentos

### ELECCION DE ALIMENTOS Y ASIGNACION A UN TIEMPO

Elige Alimento  Leguminosas

Selecciona el tiempo cuando deseas consumirlo

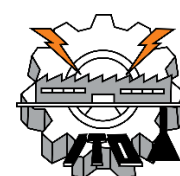
☐ Desayuno ☐ Alumerzo ☒ Cena ☐ En todos

Acelgas	Lechuga	Mango
Pollo	Pollo	Pollo
Arroz	Frijoles	Frijoles
Manzana	Pera	

Num. Ali.: 4 Num. Ali.: 4 Num. Ali.: 3

Agregar

Ilustración 8 Elección de 04



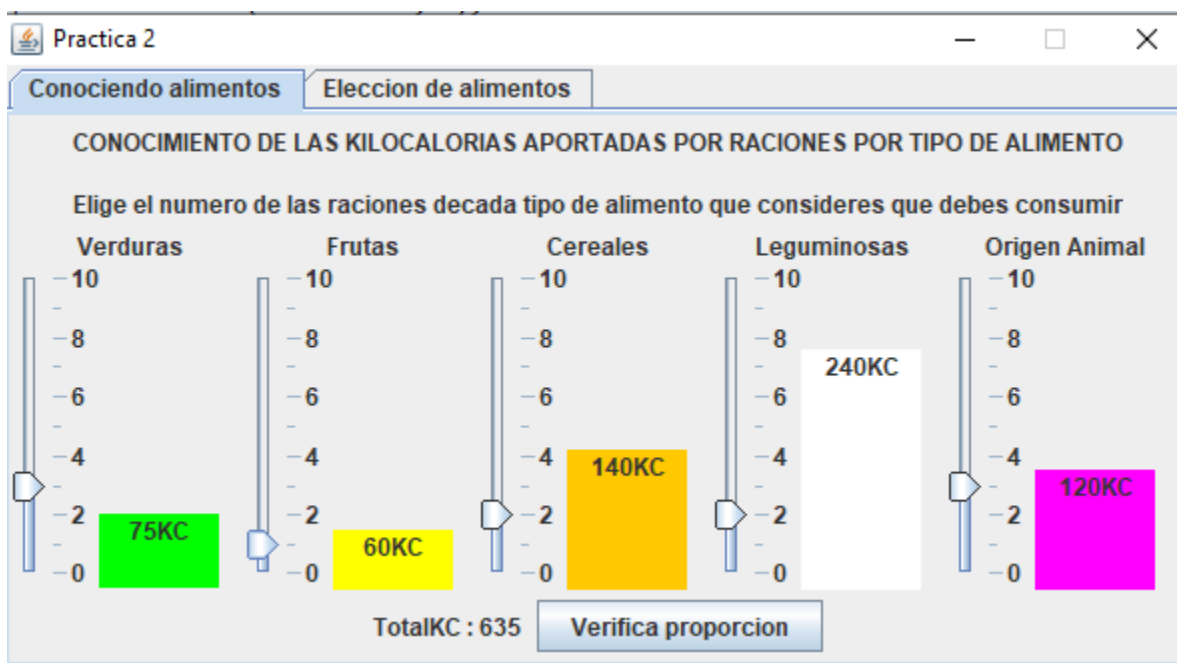


Ilustración 9 Calorías de 05

**Practica 2**

**Conociendo alimentos | Eleccion de alimentos**

**ELECCION DE ALIMENTOS Y ASIGNACION A UN TIEMPO**

Elige Alimento: **Pollo** Origen Animal

Selecciona el tiempo cuando deseas consumirlo

☐ Desayuno ☐ Alumerzo ☒ Cena ☐ En todos

Pollo	Manzana	Frijoles
Tomate	Pollo	Pollo
Lechuga	Frijoles	
Arroz	Arroz	

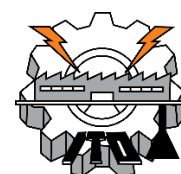
Num. Ali.: 4 Num. Ali.: 4 Num. Ali.: 2

Agregar

Ilustración 10 Elección de 05



Id	Alimentación	Calorías Totales
01	<b>Desayuno</b> - Lechuga + Pollo + Arroz <b>Almuerzo</b> - Frijoles + Pollo <b>Cena</b> - Pollo + Arroz	405 kcal
02	<b>Desayuno</b> - Lechuga + Manzana + Zanahoria + Pollo <b>Almuerzo</b> - Pollo + Arroz + Pera <b>Cena</b> - Espinaca + Acelgas + Sandia	430 kcal
03	<b>Desayuno</b> - Lechuga + Acelgas + Tomate + Manzana + Piña <b>Almuerzo</b> - Cereza + Plátano + Frijoles + Pollo <b>Cena</b> - Lechuga + Tomate + Pera + Manzana	645 kcal
04	<b>Desayuno</b> - Acelgas + Pollo + Arroz + Manzana <b>Almuerzo</b> - Lechuga + Pollo + Frijoles + Pera <b>Cena</b> - Mango + Pollo + Frijoles	730 kcal
05	<b>Desayuno</b> - Pollo + Tomate + Lechuga + Arroz <b>Almuerzo</b> - Manzana + Pollo + Frijoles + Arroz <b>Cena</b> - Frijoles + Pollo	635 kcal





## BITÁCORA DE SEGUIMIENTO CON LAS ANOTACIONES SOLICITADAS DURANTE EL DESARROLLO DE LA PRÁCTICA

Mi bitácora en este caso fue GitHub, donde actualizaba el código en ciertos periodos de tiempo, la actualización la hacía cuando consideraba que tenía un buen avance con la GUI. En la siguiente imagen se muestra los commit que realice durante el desarrollo de esta práctica que comenzó desde el sábado, dicho día en el que avance con el panel “Conociendo Alimentos”, pero fue retomada hasta el día de ayer para posteriormente terminarla hoy.

Se compromete el 26 de febrero de 2020

<b>ProyectoTerminado</b> ... ChepeAicrag12 comprometido hace 26 minutos Se terminó la GUI con lo solicitado previamente.	 ac118c8 
<b>BarrasTerminadas</b> ... ChepeAicrag12 comprometido hace 3 horas Ya se visualizan las barras	 b213f1e 
<b>ColocacionDeJFrame</b> ... ChepeAicrag12 comprometido hace 3 horas Ya se tiene el JFrame establecido con tamaño y posición correcta.	 e94d371 
<b>EventosAgregados</b> ... ChepeAicrag12 comprometido hace 8 horas Solo falta la barra de progresos	 a74d368 

Se compromete el 25 de febrero de 2020

<b>FuncionamientosBotones</b> ChepeAicrag12 comprometido hace 23 horas	 77824bc 
---	---

Se compromete el 22 de febrero de 2020

<b>Practica2</b> ... ChepeAicrag12 comprometido hace 4 días Avance de un 40% con la GUI	 b9e6798 
---	---



## CONCLUSIONES SOBRE LOS PRINCIPALES ASPECTOS A CONSIDERAR EN UNA APLICACIÓN GUI

En esta práctica se pudo observar los aspectos que se debe tener en cuenta cuando se desarrolla una GUI, algunos de estos aspectos son, la organización de los elementos en la interfaz, el uso de las barras, de igual forma el uso de botones y las acciones que estos tienen, que deben ser únicos, es decir, no sobrecargados de acciones los botones. Esta interfaz no tiene muchos aspectos de rellenar campos ya que no es de tipo formulario, es más para realizar cálculos y su aspecto visual es el principal elemento que la hace entendible para el usuario, misma vista que es complicada de desarrollar.

Otro de los aspectos que se debe tener en cuenta, es que si utilizamos el API y lo que esto nos proporciona, podemos hacer que nuestro trabajo sea bueno al momento de desarrollar la interfaz y además ahorrando tiempo, esto se pudo observar al usar un LayoutManager que era sencillo de implementar y que sus acciones en la interfaz hacían que fueran mejor y más fácil de acomodar los elementos en la interfaz, algo que no se lograba muy bien con un diseño absoluto, además que su desarrollo lleva más tiempo. Entonces debemos evitarnos tareas extras cuando ya existen cosas que nos la facilitan para poder ser eficientes al momento de desarrollar, solo se debe investigar sobre que ocupar y como usarlo.

Como mencione anteriormente, esta interfaz no hace uso de áreas de texto, solo etiquetas por lo que la validación de datos no es necesaria en este caso, no como la GUI de la practica 2. Lo que se valida son los JCheckBox para evitar no tener errores al momento de agregar los alimentos; en conclusión, de esto, podemos decir que no importa que elementos utilicemos siempre habrá que hacer validaciones para que la GUI pueda funcionar correctamente, de igual forma debemos pensar como cliente para ver los casos que se generarían en cada situación cuando se interactuara con nuestra GUI y así evitar fallos durante ejecución y que la validación sea más eficiente para evitar este tipo de problemas, esto es como pensar “en todo” con la intención de hacer que nuestra interfaz sea consistente y no llegue a tener fallos, aunque esto inevitable pero si ayudaría a reducirlos.

Concluyendo esta práctica, podemos decir que se aprendió a hacer usos de los diferentes eventos que tiene un componente. Pero lo mas destacable fue que podemos hacer que la GUI se clasifique en diferentes “vistas” para así no sobre cargar una sola y poder trabajar en diferentes, de modo que se creen pestañas que tienen ventanas en las que se realice actividades muy diferentes entre ellas, haciendo que la interfaz esté ordenada y sea mucho más adecuada para el usuario, permitiendo moverse al usuario entre la GUI.

