



ANÁLISIS DE SOFTWARE

2019 – 1^{er} Cuatrimestre

GRUPO 7

Minitienda

Integrantes

Sanchez	Fernando	36822171
Lorenz	Lautaro Enrique	37661245
Tejada	Sabrina	37790024
Riquelme	Mariano	36287422
Villca	Luis Alberto	35277730
Adrián	Morel	34437202



CÓDIGO FUENTE

ARCHIVO: Main.java

MÉTODO: Comprar

```
private void btnComprarActionPerformed(java.awt.event.ActionEvent evt) {
    boolean enviar = false;
    if((arrayProductos[0] != null)&&(prodComprados != null))
    {
        String mensajeEnviar = "<br>Gracias por comprar en nuestra tienda. En menos de 48
        horas lo recibirá.<br>";
        mensajeEnviar += "<br><table width='90%' border='0' cellpadding='0'
        cellspacing='0' align='left'><tr><td colspan='2' align='left' style='font-size:
        15px;'><b>Datos del cliente:</b></td></tr><tr><td
        colspan='2'>&nbsp;</td></tr><tr><td
        width='10%'><b>Nombre:</b></td><td>"+arrayClientes[userActual].getNombre()+"</td></tr>";
        mensajeEnviar += "<tr><td
        width='10%'><b>Apellidos:</b></td><td>"+arrayClientes[userActual].getApellidos()+"<
        /td></tr><tr><td
        width='10%'><b>UserName:</b></td><td>"+arrayClientes[userActual].getUserName()+"</td>
        </tr></table>";
        mensajeEnviar += "<br><br><br><table width='90%' border='1px' cellpadding='0'
        cellspacing='0' align='left'><tr><td colspan='2' align='left' style='font-size:
        15px;'><b>Datos de la compra:</b></td></tr><tr><td
        colspan='2'>&nbsp;</td></tr><tr>";
        mensajeEnviar += "<td width='10%'><b>Nombre:</b></td><td
        width='10%'><b>Descripción:</b></td><td width='10%'><b>Precio:</b></td><td
        width='10%'><b>Cantidad:</b></td><td width='10%'><b>Total:</b></td></tr>";
        for(int x = 0; x < prodComprados.length; x++)
        {
            if(prodComprados[x] > 0)
            {
                enviar = true;
                mensajeEnviar += "<tr><td
                align='center'>"+arrayProductos[x].getNombre()+"</td><td
                align='center'>"+arrayProductos[x].getDescripcion()+"</td><td
                align='center'>"+arrayProductos[x].getPrecio()+"</td>";
                mensajeEnviar += "<td align='center'>"+prodComprados[x]+"</td><td
                align='center'>"+arrayTotal[x]+"</td></tr>";
            }
        }
        mensajeEnviar +=
        "<tr><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>&nbsp;</td><td>
        align='center'>"+compraTotal+" &euro;</td></tr>";
        mensajeEnviar += "</table>";
        String subject = "Procesando pedido";
        if(enviar)
        {
            String[] contenido = new String[]{ arrayClientes[userActual].getNombre()+" hemos
            registrado su pedido.", "En menos de 24 horas recibirá un correo con los
            detalles." };
            JOptionPane.showMessageDialog(new JFrame(), contenido, "Procesando pedido...", J
           OptionPane.INFORMATION_MESSAGE);
            try {
                Utils.enviamail(mensajeEnviar,subject,arrayClientes[userActual]);
            }
        }
    }
}
```

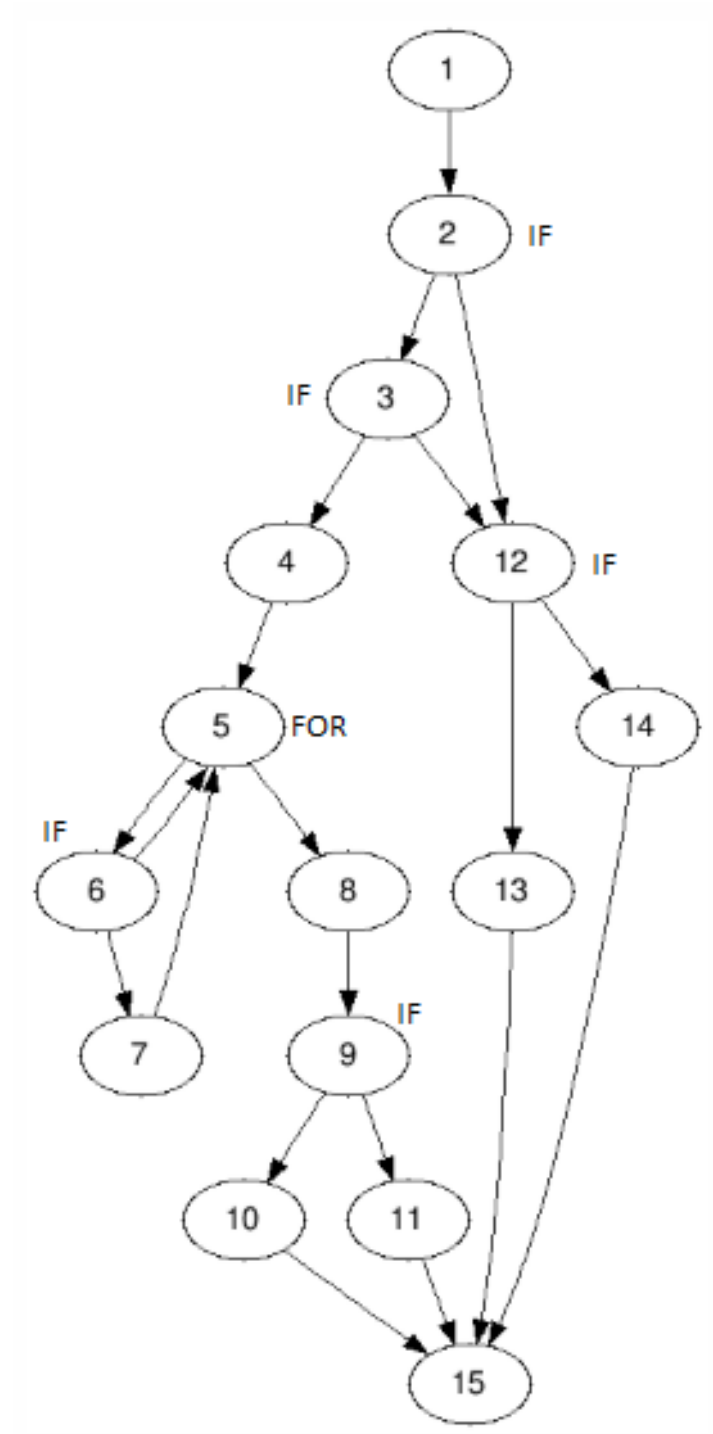


```

        } catch (NoSuchProviderException ex)
        {
        } catch (MessagingException ex)
        {
        }
        Clientes aux = new Clientes();
        aux.guardarClientes(arrayClientes);
        System.exit(0);
    }
    else
    {
        String[] contenido2 = new String[]{ "Señor
        "+arrayClientes[userActual].getNombre()+",","No ha seleccionado ning n producto
        para comprar.", " jGracias!" };
        JOptionPane.showMessageDialog(new JFrame(), contenido2, " jEscoja sus
        productos!", JOptionPane.WARNING_MESSAGE);
    }
}
else
{
    if(arrayProductos[0] == null)
    {
        String[] contenido = new String[]{ "Lo sentimos se or
        "+arrayClientes[userActual].getNombre(),"no tenemos ning n producto para
        venderle.", " ", "Contacte con el Administrador para mas
        informaci n", " jGracias!", " ", "Saliendo del Programa..." };
        JOptionPane.showMessageDialog(new JFrame(), contenido, " jSin productos!",
        JOptionPane.WARNING_MESSAGE);
        Clientes aux = new Clientes();
        aux.guardarClientes(arrayClientes);
        System.exit(0);
    }
    else
    {
        String[] contenido = new String[]{ "Se or
        "+arrayClientes[userActual].getNombre()+",","No ha seleccionado ning n producto
        para comprar.", " jGracias!" };
        JOptionPane.showMessageDialog(new JFrame(), contenido, " jEscoja sus
        productos!", JOptionPane.WARNING_MESSAGE);
    }
}
}
}

```

GRAFO DEL MÉTODO COMPRAR





RESULTADO DEL ANÁLISIS MEDIANTE HERRAMIENTA

Analisis del Metodo

Lineas de codigo	<input type="text" value="58"/>
Lineas comentadas	<input type="text" value="1"/>
Porcentaje de codigo comentado	<input type="text" value="1 %"/>
Complejidad ciclomatica	<input type="text" value="7"/>
Fan in	<input type="text" value="1"/>
Fan out	<input type="text" value="0"/>
Longitud	<input type="text" value="293"/>
Volumen	<input type="text" value="1877,95"/>
Ezfuero	<input type="text" value="18779,52"/>

VERIFICACIÓN DE RESULTADO VS GRAFO OBTENIDO

$$\text{Nodos Predicado } (6) + 1 = 7$$

$$\text{Cantidad de aristas } (20) - \text{Cantidad de nodos } (15) + 2 = 7$$

$$\text{Cantidad de regiones} = 7$$

Complejidad ciclomática verificada



MÉTODO DE MC CABE

Recorrer todas las aristas con caminos linealmente independientes, donde un camino linealmente independiente es aquel que agrega la menor cantidad de aristas que no fueron recorridas por ningún otro camino.

- 1) 1-2-12-13-15 (*camino más corto*)
- 2) 1-2-12-14-15
- 3) 1-2-3-12-13-15
- 4) 1-2-3-4-5-8-9-10-15
- 5) 1-2-3-4-5-8-9-11-15
- 6) 1-2-3-4-5-6-5-8-9-10-15
- 7) 1-2-3-4-5-6-7-5-8-9-11-15

caso	arrayProductos	prodComprados	enviar	output
1	null	longitud 0	-	SIN PRODUCTOS PARA VENDER
2	null	longitud > 0	-	SIN PRODUCTOS PARA COMPRAR
3	longitud > 0	null	-	SIN PRODUCTOS PARA COMPRAR
4	longitud > 0	longitud 0	false	SIN PRODUCTOS PARA COMPRAR
5	longitud > 0	longitud 0	true	NO FACTIBLE
6	longitud > 0	longitud > 0	true	PRODUCTOS COMPRADOS
7	longitud > 0	longitud > 0	false	NO FACTIBLE

$$Ter = \frac{\text{casos probados}}{\text{caminos L.I.}} = \frac{5}{7} = 71\%$$



MÉTODO DE SENTENCIAS

Recorrer todos los nodos con la menor cantidad de caminos posibles

- 1) 1-2-12-14-15
- 2) 1-2-12-13-15
- 3) 1-2-3-4-5-6-4-7-8-9-14-15
- 4) 1-2-3-4-7-8-10-14-15

caso	arrayProductos	prodComprados	enviar	output
1	null	-	-	SIN PRODUCTOS PARA VENDER
2	longitud > 0	null	-	SIN PRODUCTOS PARA COMPRAR
3	longitud > 0	longitud > 0	true	PRODUCTOS COMPRADOS
4	longitud > 0	0	false	SIN PRODUCTOS PARA COMPRAR

$$Ter = \frac{\text{sentencias probadas}}{\text{sentencias totales}} = \frac{14}{14} = 100\%$$



MÉTODO DE DECISIÓN-CONDICIÓN

decisión	condición	valor condición	valor lógico condición	valor lógico decisión	output
((arrayProductos[0] != null) && (prodComprados != null))	arrayProductos[0] != null	arrayProductos[0] = any	V	Evalúa siguiente condición	No genera salida
		arrayProductos[0] = null	F	F	No genera salida
	prodComprados != null	prodComprados = any	V	V	No genera salida
		prodComprados = null	F	F	No genera salida
(int x = 0; x < prodComprados.length; x++)	prodComprados.length	prodComprados.length > 0	V	V	No genera salida
		prodComprados.length = 0	F	F	No genera salida
prodComprados[x] > 0	prodComprados[x] > 0	prodComprados[x] > 0	V	V	PRODUCTOS COMPRADOS
		prodComprados[x] = 0	F	F	No genera salida
enviar	enviar	enviar = true	V	V	PRODUCTOS COMPRADOS
		enviar = false	F	F	SIN PRODUCTOS PARA COMPRAR
arrayProductos[0] == null	arrayProductos[0] == null	arrayProductos[0] = null	V	V	SIN PRODUCTOS PARA VENDER
		arrayProductos[0] = any	F	F	SIN PRODUCTOS PARA COMPRAR

$$T\epsilon r = \frac{\text{decisiones probadas}}{\text{decisiones totales}} = \frac{5}{5} = 100\%$$

$$T\epsilon r = \frac{\text{condiciones probadas}}{\text{condiciones totales}} = \frac{6}{6} = 100\%$$



PUNTOS DE FUNCIÓN

ENTRADAS	ATRIBUTOS	FICHEROS	PONDERACIÓN	SUBTOTAL
INSERTAR CLIENTE	NOMBRE	USUARIOS	BAJA	5 BAJA * 3 = 15
	APELLIDO			
	CORREO			
	USER			
	CONTRASEÑA			
INSERTAR PRODUCTO	NOMBRE	PRODUCTOS	BAJA	
	PRECIO			
	DESCRIPCIÓN			
ELIMINAR PRODUCTO	PRODUCTO SELECCIONADO	PRODUCTOS	BAJA	
ELIMINAR USUARIO	USUARIO SELECCIONADO	USUARIOS	BAJA	
INGRESO	USER	USUARIOS	BAJA	
	CONTRASEÑA			

CONSULTAS	ATRIBUTOS	FICHEROS	PONDERACIÓN	SUBTOTAL
VER CLIENTES	NOMBRE	USUARIOS	BAJA	2 BAJA * 3 = 6
	APELLIDO			
	CORREO			
	USER			
	CONTRASEÑA			
	TIPO			
VER PRODUCTOS		PRODUCTOS	BAJA	

SALIDAS	ATRIBUTOS	FICHEROS	PONDERACIÓN	SUBTOTAL
COMPRAR	NOMBRE	PRODUCTOS	BAJA	1 BAJA * 4 = 4
	PRECIO	USUARIOS		
	DESCRIPCIÓN			
	CANTIDAD			



FICHEROS LÓGICOS INTERNOS	ATRIBUTOS	ENTIDADES	PONDERACIÓN	SUBTOTAL
USUARIOS	NOMBRE	USUARIO	BAJA	2 BAJA * 7 = 14
	APELLIDO			
	CORREO			
	USER			
	CONTRASEÑA			
	TIPO			
PRODUCTOS	NOMBRE	PRODUCTO	BAJA	
	PRECIO			
	DESCRIPCIÓN			

FICHEROS LÓGICOS EXTERNOS	ATRIBUTOS	ENTIDADES	PONDERACIÓN	SUBTOTAL
-	-	-	-	-

PUNTOS DE FUNCIÓN SIN AJUSTAR = **PFSA** = 15 + 6 + 4 + 14 = **39**

Factor de Ajuste	Puntaje		VALORACIÓN
	Min	Max	
Comunicación de Datos	0	5	3
Proceso Distribuido	0	5	0
Objetivos de Rendimiento	0	5	5
Configuración de Explotación Compartida	0	4	0
Tasa de transacciones	0	5	5
Entrada de Datos en Línea	0	5	0
Eficiencia con el Usuario Final	0	5	5
Actualizaciones en Línea	0	5	0
Lógica de Proceso Interno Compleja	0	5	2
Reusabilidad del Código	0	5	3
Conversión e Instalación contempladas	0	5	5
Facilidad de Operación	0	5	5
Instalaciones Múltiples	0	5	0
Facilidad de Cambios	0	5	5
			ACT 38

PUNTOS DE FUNCIÓN AJUSTADOS = PFA = PFSA * (0.65 + (0.01 * ACT))

PFA = 39 * (0.65 + (0.01 * 38)) = **40.17**



MÉTODO DE CAJA NEGRA – CLASES DE EQUIVALENCIA

Archivo: Main.java

Método analizado: Comprar “btnComprarActionPerformed()”

Se debe hacer la prueba para saber si puedo comprar los productos seleccionados.

Condiciones:

- a) Debe existir al menos un producto creado en el sistema.
- b) Debe haber seleccionado al menos un producto para comprar.

Condición de entrada	Clase válida	Clase no válida	Regla que aplica
productos(cantidad)	> 0 (2)	0 (1)	RANGO
productosComprados(cantidad)	> 0 (4)	0 (3)	RANGO

Entrada	Clases cubiertas
productos = [producto1, producto2]; productosComprados = [producto1];	2, 4
productos = []	1
productosComprados = []	3

MÉTODO DE CAJA NEGRA – ANÁLISIS DE VALORES LÍMITE (AVL)

Dado el método del punto anterior “btnComprarActionPerformed()” como la clase válida 2 donde la cantidad de elementos debe ser mayor a cero, falta probar el caso límite donde el vector contiene un único producto, por lo que es necesario agregar una nueva entrada.

Entrada	Clases cubiertas
productos = [producto1, producto2];	2, 4
productosComprados = [producto1];	
productos = []	1
productosComprados = []	3
productos = [producto1];	2