

PRACTICA N ° 2

CARLOS FABIAN GARCES TORRES

CÓDIGO: DSUP10-GP-02

1. TEMA: Manejo de los diferentes tipos de variables y la implementación de clases mediante el desarrollo de Algoritmos y programas para manejo de datos y los conceptos de la POO.

2. TIEMPO DE DURACIÓN: 120 minutos

3. FUNDAMENTOS:

En la programación orientada a objetos (POO), las variables son elementos fundamentales para almacenar y manipular datos. Estas variables se definen dentro de clases, que son los componentes básicos de la POO. Aquí tienes una breve descripción sobre las variables y el manejo de datos en la programación orientada a objetos:

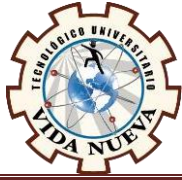
Variables: En la POO, una variable es un contenedor que puede almacenar diferentes tipos de datos, como números, cadenas de texto, booleanos, objetos, entre otros. Las variables tienen un nombre y un tipo de dato asociado, y se utilizan para representar y almacenar información.

4. OBJETIVO(S)

- Analizar el modelo de POO en JAVA con el fin de implementar una factura digital en Netbeans mediante controles WEB.
- Generar un listado de objetos útiles dentro de Netbeans, para la creación de una factura digital.
- Desarrollar una factura digital utilizando POO, para mantener una buena estructura en la ejecución del producto de software.

5. MATERIALES

- Libros



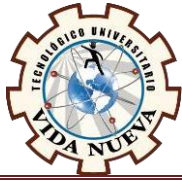
- Internet
- Campus Virtual
- Software (Netbeans)

6. EQUIPOS

- Computador

7. NORMAS DE SEGURIDAD

- **No comer, beber en el laboratorio:** Esto es importante porque cualquier derrame accidental de alimentos o bebidas podría dañar los equipos o dispositivos, lo que podría resultar en costosas reparaciones o reemplazos. Además, fumar en el laboratorio podría ser peligroso debido a los productos químicos y materiales inflamables que se utilizan en algunos experimentos.
- **Utilizar protección personal adecuada:** Los estudiantes deben utilizar protección personal adecuada, como guantes, gafas de seguridad y batas de laboratorio si es necesario. Esto es especialmente importante si se está trabajando con sustancias químicas peligrosas o dispositivos que generan calor o electricidad.



8. PREPARACIÓN PREVIA/ CONOCIMIENTOS PREVIOS

Requisito	Descripción
Clases y objetos	En programación orientada a objetos (POO), las clases son estructuras que definen propiedades y comportamientos de un objeto.
Programación orientada a objetos (POO)	La POO es un paradigma de programación que organiza el código en objetos, los cuales son instancias de clases.
Variables	En programación, las variables se utilizan para almacenar y manipular datos. Pueden tener diferentes tipos, como enteros (int), números de punto flotante (float), cadenas de texto (string), booleanos (bool), entre otros.
Atributos	Son las características de un objeto. Por ejemplo, si tienes una clase "Persona", los atributos podrían ser "nombre", "edad", "altura", etc.

9. PROCEDIMIENTO

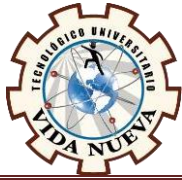
1. Abrir el entorno de trabajo Netbeans.
2. Crear el entorno gráfico para la creación de la factura digital. Ver anexo 1.
3. Analizar los elementos que contendrán en el constructor de las clases.
4. Desarrollar la programación de cada uno de los botones para la factura digital.
5. Verificar los datos obtenidos y registrar los resultados. Colocar en tabla 1.

Tabla 1

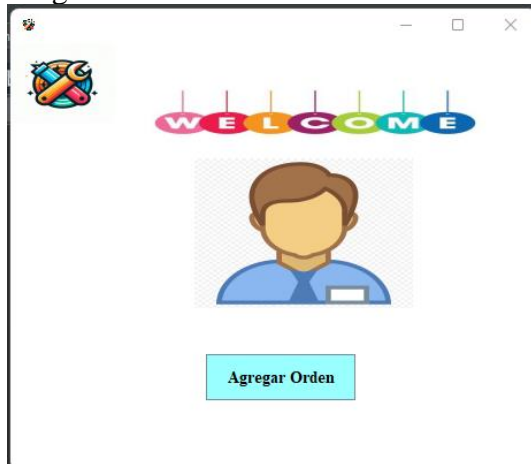
Factura digital.

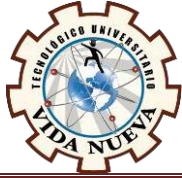
Atributos del objeto Factura	Resultados obtenidos
Subtotal	
Impuestos	
Total	

Nota. Elementos finales de la factura digital.



Programa de Factura en NEATBEANAS usando Java





Tecnológico Universitario Vida Nueva



*******FACTURA FERRETERIA EL GRAN CARLOS*******

Cliente: Carlos

Producto	Precio	Cantidad	Importe
Martillo	\$5,50	2	\$11,00
Machete	\$8,00	1	\$8,00

Subtotal: \$19,00
IVA (15%): \$2,85
Total: \$21,85

OK

```
package practica;

import java.awt.Image;
import java.awt.Toolkit;
import java.util.ArrayList;
import javax.swing.DefaultComboBoxModel;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

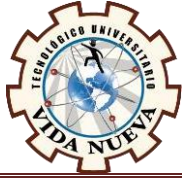
public class Tiendita extends javax.swing.JFrame {
    //Inicializamos los Atributos principales de la clase los productos disponibles

    String nombres[] = {"Carlos", "Jose", "Manuel", "Jose", "Pedro", "Juan", "Felix"};
    String productos[] = {"Martillo", "Machete", "Desarmadores", "Tornillos", "Tuercas", "Cables", "Cemento"};
    double precios[] = {5.50, 8.00, 1.50, 0.50, 0.25, 3.00, 30.00};
    double precio = 0;
    int cantidad = 0;
    DefaultTableModel modelo = new DefaultTableModel();
    //Array List va ir guardando los datos de las ventas que se va ir realizando para rellenar la tabla
    ArrayList<Venta> listaVentas = new ArrayList<Venta>();

    public Tiendita() {
        initComponents();
        //Agregamos el titulo y le damos un icono a la pagina
        this.setTitle("Ferreteria el Gran Carlos");
        //importamos Image y toolking para llamar a la imagen en la carpeta de img que creamos
        Image icono = Toolkit.getDefaultToolkit().getImage(getClass().getResource("/img/ferreteria.png"));
        this.setIconImage(icono);

        //Agregamos el Logo a un label
        Image logo = Toolkit.getDefaultToolkit().getImage(getClass().getResource("/img/ferreteria.png"));
        lbllogo.setIcon(new ImageIcon(logo.getScaledInstance(lbllogo.getWidth(), lbllogo.getHeight(), Image.SCALE_SMOOTH)));
        this.setSize(800, 800);
        this.setLocationRelativeTo(null);

        //Agregar el logo al Boton de Agregar Producto
        Image iconoAgregar = Toolkit.getDefaultToolkit().getImage(getClass().getResource("/img/mas.png"));
```



```
public Tiendita() {
    initComponents();
    //Agregamos el titulo y le damos un icono a la pagina
    this.setTitle("Ferreteria el Gran Carlos");
    //importamos Image y toolkit para llamar a la imagen en la carpeta de img que creamos
    Image icono = Toolkit.getDefaultToolkit().getImage(getClass().getResource("/img/ferreteria.png"));
    this.setIconImage(icono);

    //Agregamos el Logo a un label
    Image logo = Toolkit.getDefaultToolkit().getImage(getClass().getResource("/img/ferreteria.png"));
    lbllogo.setIcon(new ImageIcon(logo.getScaledInstance(lbllogo.getWidth(), lbllogo.getHeight(), Image.SCALE_SMOOTH)));
    this.setSize(800, 800);
    this.setLocationRelativeTo(null);

    //Agregar el logo al Boton de Agregar Producto
    Image iconoAgregar = Toolkit.getDefaultToolkit().getImage(getClass().getResource("/img/mas.png"));
    btnAgregar.setIcon(new ImageIcon(iconoAgregar.getScaledInstance(btnAgregar.getWidth(), btnAgregar.getHeight(), Image.SCALE_SMOOTH)));

    //Agregamos imagen para el boton de borrar
    Image icoBorrar = Toolkit.getDefaultToolkit().getImage(getClass().getResource("/img/eliminarF.png"));
    btnEliminar.setIcon(new ImageIcon(icoBorrar.getScaledInstance(btnEliminar.getWidth(), btnEliminar.getHeight(), Image.SCALE_SMOOTH)));

    //Agregamos el boton de imprimir Productos
    Image icoImprimir = Toolkit.getDefaultToolkit().getImage(getClass().getResource("/img/imprimir.png"));
    btnImprimirFactura.setIcon(new ImageIcon(icoImprimir.getScaledInstance(btnImprimirFactura.getWidth(), btnImprimirFactura.getHeight(), Image.SCALE_SMOOTH)));

    //Agregar productos al Combo box que seleccionamos antes
    DefaultComboBoxModel comboModel = new DefaultComboBoxModel(productos);
    DefaultComboBoxModel cboModel = new DefaultComboBoxModel(nombres);
    cboProducto.setModel(comboModel);
    cboNombre.setModel(cboModel);
    modelo.addColumn("Nombre C.");
    modelo.addColumn("Descripcion");
    modelo.addColumn("Precio U.");
}
```

```
//Impresion de la factura
StringBuilder factura = new StringBuilder();

factura.append("*****FACTURA FERRETERIA EL GRAN CARLOS*****\n");
factura.append("Cliente: ").append(cboNombre.getSelectedItem().toString()).append("\n");
factura.append("-----\n");
factura.append(String.format("%-20s %-10s %-10s %-10s\n", "Producto", "Precio", "Cantidad", "Importe"));
factura.append("-----\n");

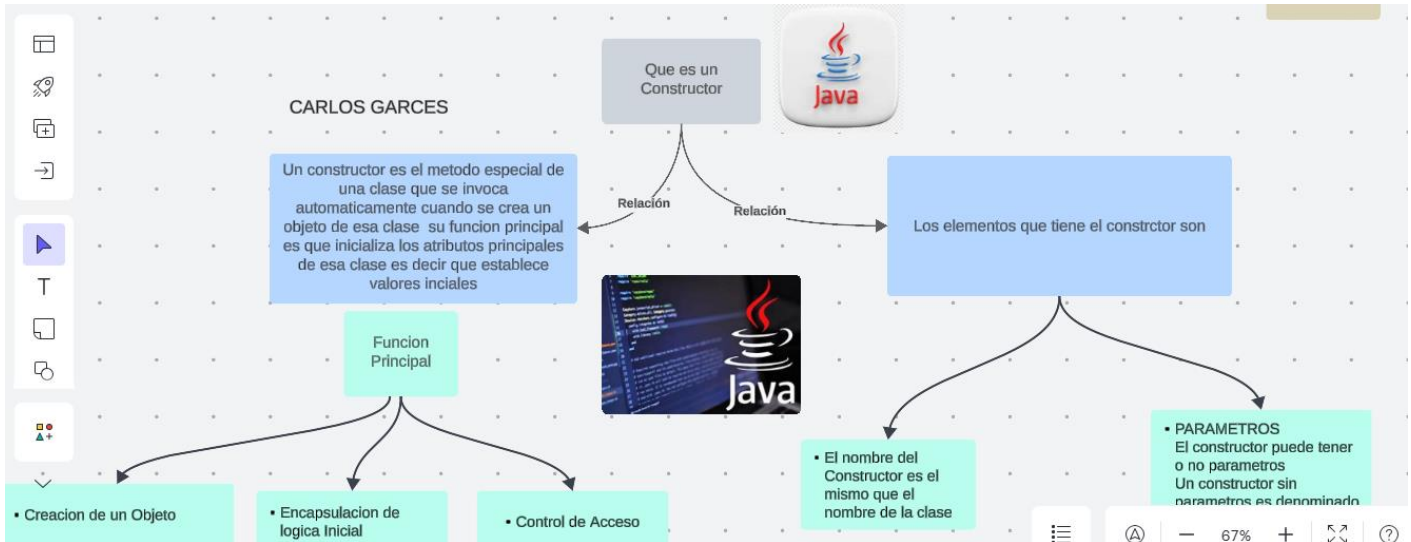
double subtotal = 0;
for (Venta venta : listaVentas) {
    factura.append(String.format("%-20s %-10s %-10d %-12s\n",
        venta.getDescripcion(),
        String.format("$%.2f", venta.getPrecio()),
        venta.getCantidad(),
        String.format("$%.2f", venta.getImporte())));
    subtotal += venta.getImporte();
}

double iva = subtotal * 0.15;
double total = subtotal + iva;

factura.append("-----\n");
factura.append(String.format("%-20s %-10.2f\n", "Subtotal:", subtotal));
factura.append(String.format("%-20s %-10.2f\n", "IVA (15%):", iva));
factura.append(String.format("%-20s %-10.2f\n", "Total:", total));
factura.append("*****\n");
// Mostramos la factura en un JOptionPane
JOptionPane.showMessageDialog(this, factura.toString(), "Factura de Compra", JOptionPane.INFORMATION_MESSAGE);
}
```


10. ORGANIZADOR GRÁFICO DE LAS IDEAS PRINCIPALES

Realizar un organizador gráfico que muestre los elementos y el uso del constructor.



11. PREGUNTAS DE REFUERZO

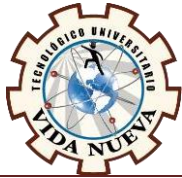
- ¿Cuál son las diferencias entre los niveles de acceso al utilizar clases?
Los niveles de acceso de java definen la visibilidad de las clases, métodos atributos, Public permite el acceso desde cualquier clase; protegected desde la misma clase, sub clases y el mismo paquete, default solo dentro del mismo paquete y private restringe el acceso a la propia clase, facilitando encapsulamiento y seguridad
- ¿Qué es instanciar una clase y coloque un ejemplo?
Instanciar una clase significa crear un objeto a partir de esa clase, al instanciar se reserva espacio en la memoria para el objeto y se inicializan los atributos a través del constructor
- ¿Cuál son los paquetes para utilizar java.lang?
es un paquete básico de Java que se importa automáticamente y contiene clases esenciales. No se requieren paquetes adicionales para utilizar java.lang, ya que está disponible de manera predeterminada. Este paquete incluye clases fundamentales como String, Math, Integer, y System, que son comunes en cualquier programa Java.

12. CONCLUSIONES:

Implementar una factura digital mediante el enfoque OOP en Java facilita un diseño modular y flexible. Mediante clases y objetos se optimiza la reutilización de código y se simplifica la integración de funcionalidades a través de controles web en NetBeans, mejorando la eficiencia y el mantenimiento del software.

NetBeans proporciona herramientas como JTextField, JTable, JButton y JComboBox, esenciales para desarrollar una factura digital. El uso eficiente de estos componentes permite el diseño de interfaces interactivas y funcionales, mejorando la experiencia del usuario en el sistema.

Desarrollar una factura digital mediante OOP asegura una estructura organizada, facilitando su escalabilidad y mantenimiento. Mediante la creación de clases como Producto, Cliente y Factura se obtiene un sistema sólido que simplifica la gestión y generación de facturas de



forma digital.

(Relacionadas a los objetivos)

13. EVALUACIÓN DEL APRENDIZAJE

Rúbrica:

CONSIDERACIONES	0,10 MUY BUENO	0, 05 BUENO	0, 01 REGULAR
Declaración y uso de variables: Evalúa la declaración correcta y uso de variables en el código.	Algoritmos simples con estructuras de control básicas.	Manipulación sencilla de variables.	Declaración de variables básicas.
Manipulación de variables: Evalúa la manipulación adecuada de variables y el uso de operadores y funciones para realizar cálculos y transformaciones.	Algoritmos más complejos con estructuras de control y buenas prácticas de programación.	Manipulación de variables con operaciones y funciones relevantes.	Declaración y uso adecuado de variables de diferentes tipos.
Desarrollo de algoritmos: Evalúa la capacidad del estudiante para diseñar algoritmos claros y estructurados para resolver problemas.	Algoritmos sofisticados con estructuras de control y optimización del código.	Manipulación avanzada de operaciones con variables y funciones complejas.	Declaración y uso eficiente de variables.

14. BIBLIOGRAFÍA

Arland, D. S. (2011). *Programación Java Script Y JQuery*. España: Anaya

Multimedia.

Arnold, K. (2001). *El Lenguaje de Programación Java*. España: Pearson.

Deitl, P. (2012). *Java como Programar* novena edición. México: Pearson.



Nota. Diseño de la factura para el desarrollo