

IPC1 Proyecto Semestre1 2023

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Introducción a la Programación y Computación 1

Sección: "E"

Manual técnico

Dominic Juan Pablo Ruano Pérez

202200075

Guatemala 14 de marzo del 2023

Introducción

El siguiente proyecto es un programa para una empresa de envíos con el cual se podría realizar la gestión tanto de usuarios, regiones, departamentos, municipios, kioscos además de poder ver reportes de las regiones con más ventas, usuarios con mas ventas entre otros, debido a que este sistema es una DEMO se usó únicamente la memoria dinámica del sistema.

Requisitos del sistema

Windows Vista SP2

Windows Server 2008 R2 SP1 (64 bits)

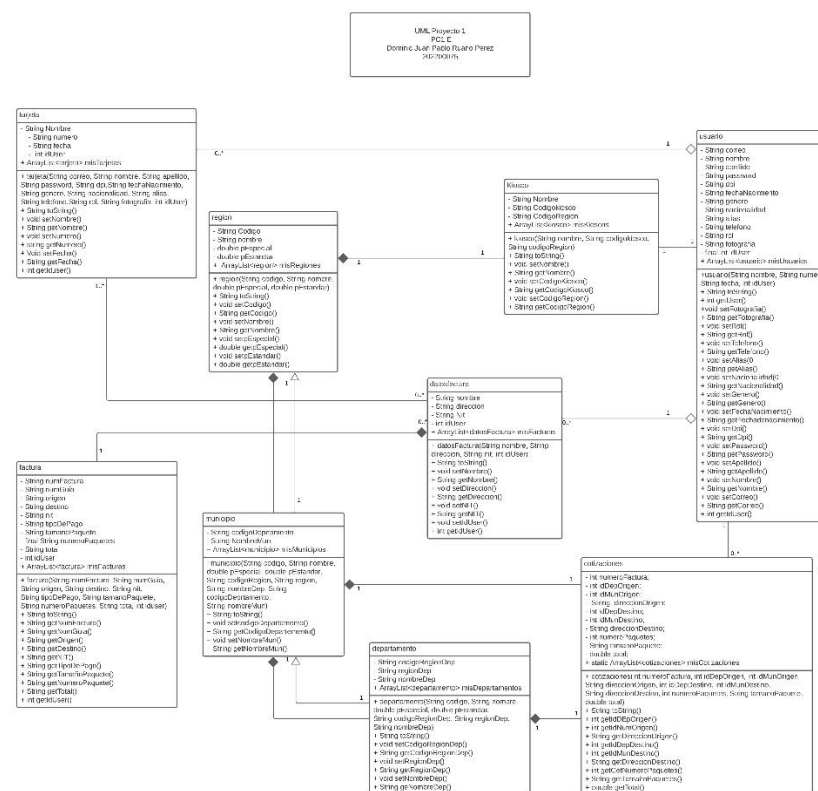
Windows Server 2012 y 2012 R2 (64 bits)

RAM: 128 MB

Espacio en disco: 124 MB para JRE

Procesador: Mínimo Pentium 2 a 266 MHz

Arquitectura del programa



Se empleo el uso de clases con los debidos constrictores para cada uno además de un manejo de los datos de cada clase guardados en Arrayalist para cada uno se utilizó herencia cuando fue necesario además de que se implemento Swing para poder mostrar gráficamente el menú y que a los usuario les sea más fácil poder interactuar con el mismo.

Tecnologías y herramientas utilizadas

- Java 11
- IntelliJ
- java.util.ArrayList
- javax.swing
- java.awt.event.ActionEvent

funcionalidades del programa

Autenticación de usuario

```
public boolean validarCredenciales(){
    idUser = -1;
    for (usuario i : usuario.misUsuarios) {
        idUser = i.getIdUser();
        if (txtCorreo.getText().equals(i.getCorreo()) && txtContraseña.getText().equals(i.getPassword())){
            return true;
        }
    }
    return false;
}
```

Esta función es llamada en el momento en el que se intenta iniciar sesión y recorre la lista de todos los usuarios creados para comparar que exista un usuario y contraseña para poder dar acceso al sistema devolviendo un verdadero si el usuario existe y las credenciales son correctas o un falso si no lo son.

Funciones de ir

```
public static void irAutenticacion(){
    Main.frame.setContentPane(new Autenticacion().Autenticacion);
    Main.frame.pack();
    Main.frame.setSize( width: 400, height: 300);
}
```

Se crearon una gran cantidad de funciones de ir para hacer más fácil el llamar una nueva ventana.

Funciones de guardar

```
private void guardarKiosco(){
    if (!txtCodigo.getText().equals("") && !txtNombre.getText().equals("")){
        kiosco.misKioscos.add(new kiosco(txtNombre.getText(), txtCodigo.getText(), cboRegion.getSelectedItem().toString()));

        txtNombre.setText("");
        txtCodigo.setText("");

        System.out.println();
        for (kiosco i: kiosco.misKioscos) {
            System.out.println(i);
        }
    }else {JOptionPane.showMessageDialog(Main.frame, "Por favor ingrese todos los campos.");}
}
```

Para cada tipo de datos existe una función empleando el constructor de la clase específica con la cual al momento de guardar el dato ya sea un kiosco, usuario, región, municipio, etc solamente se llama a su respectiva función para poder ser guardada la información en donde es debido.

Función de modificar

```
private void modificarKiosco(int num){
    if (!txtCodigo.getText().equals("") && !txtNombre.getText().equals("")){
        kiosco.misKioscos.get(num).setCodigoKiosco(txtCodigo.getText());
        kiosco.misKioscos.get(num).setNombre(txtNombre.getText());
        kiosco.misKioscos.get(num).setCodigoRegion(cboRegion.getSelectedItem().toString());

        System.out.println();
        for (kiosco i: kiosco.misKioscos) {
            System.out.println(i);
        }
    }else {JOptionPane.showMessageDialog(Main.frame, "Por favor ingrese todos los campos.");}
}
```

Con esta función la cual existe una para cada tipo de dato se modifican los datos que el usuario desea por otros empleando los setters de cada atributo.

Función eliminar

```
private void eliminarKiosco(int num){
    kiosco.misKioscos.remove(num);

    txtNombre.setText("");
    txtCodigo.setText("");

    System.out.println();
    for (kiosco i: kiosco.misKioscos) {
        System.out.println(i);
    }
}
```

Con esta función se borra el dato que el usuario desea, esto existe para cada tipo de dato.