

Manual técnico 2022

8 SEPTIEMBRE

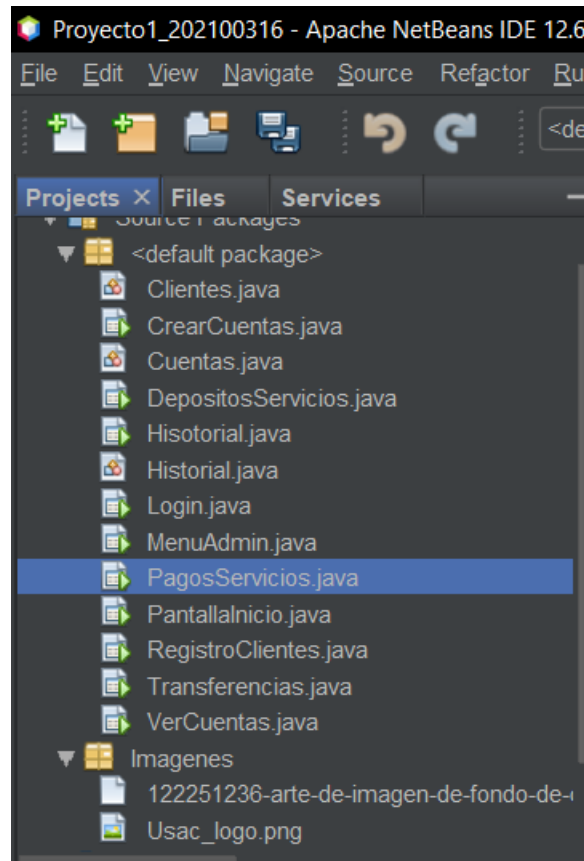
Universidad San Carlos de Guatemala
Creado por: Jose Andres Hinestroza Garcia
Carnet: 202100316



Nombre del
logotipo

Manual Técnico

La siguiente práctica se diseñó a base de lenguaje de Java en Apache NetBeans. Se realizaron tres clases para poder distinguir entre cada menú en el cual el usuario podrá interactuar, identificándose de igual manera cada frame con el menú en el cual se va a interactuar.



Login

En ella solo se valida los ingresos si el usuario es correcto así como si la contraseña que se ingreso es la correcta para poder acceder

```
private void IngresarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    String usuario = "administrador";  
    //System.out.println(Contraseña.getText());  
    if (Usuario.getText().equals("administrador") && Contraseña.getText().equals("202100316"))  
        PantallaInicio PantallaPrincipal = new PantallaInicio();  
        this.setVisible(false);  
        MenuAdmin Registro = new MenuAdmin();  
        Registro.setVisible(true);  
    }  
}
```

```

try {
    int Cui = Integer.parseInt(CUI.getText());
    if (Apellido.getText().equals("") || Nombre.getText().equals("")) {
        JOptionPane.showMessageDialog(null, "Ingreso algun dato vacio");
    } else {
        int bandera = 0;

        if (contador < 5) { // contando el 0 no debe de pasar de 4

            ListaRegistradora[contador] = new Clientes(Cui, Nombre.getText(), Apellido.getText());

            bandera = Busqueda(contador);
            if (bandera == 1) {
                JOptionPane.showMessageDialog(null, "no pueden haber CUI repetido");
            } else {
                this.setVisible(false);
                Administrador.setVisible(true);
                contador++;
            }
        } else {
            JOptionPane.showMessageDialog(null, "No se puede agregar mas clientes");
            this.setVisible(false);
            Administrador.setVisible(true);
        }
    }
}

```

En el menu de registrar clientes, se valida que en el cui no este ingresando un valor que no sea diferente a un numero como ademas que no hayan valores vacios, para posteriormente ir validando si hay CUI repetido

Busqueda()

En ella se ira validando en cada dato si existe un valor de CUI repetido y si existe retornara la variable flag 1, demostrando que por un mensaje que el cui se esta repitiendo.

```

int Busqueda(int contador) {
    int flag = 0;
    for (int i = 0; i < ListaRegistradora.length; i++) {
        if (ListaRegistradora[i] != null) {
            if (contador != i) {
                if (ListaRegistradora[i].getCUI() == ListaRegistradora[contador].getCUI()) {
                    flag = 1;
                }
            }
        }
    }
    return flag;
}

```

Se irá validando si esta posee aun hay espacio en el arreglo para poder crear una nueva cuenta, o si en efecto a alcanzado el límite.

```
if (numerocuento < 20) {
    if (tipo.equals("transferencia") || tipo.equals("deposito")) {
        verHistoria[filas][numerocuento] = new Historial(String.valueOf(cuentas.NuevasCuentas[filas],
            fecha, registro.ListaRegistradora[filas].getNombre(), registro.ListaRegistradora[filas].getDinero(),
            dinero, 0);
        numerocuento++;
    }
}
if (numerocuento < 20) {
    if (tipo.equals("retiro transferencia") || tipo.equals("pago Servicio Luz")
        || tipo.equals("pago Servicio Agua") || tipo.equals("pago Servicio Telefon")) {
        verHistoria[filas][numerocuento] = new Historial(String.valueOf(cuentas.NuevasCuentas[filas],
            fecha, registro.ListaRegistradora[filas].getNombre(), registro.ListaRegistradora[filas].getDinero(),
            0, dinero);
        numerocuento++;
    }
}
JOptionPane.showMessageDialog(rootPane, "Transacciones realizada con exito");
} else {
    JOptionPane.showMessageDialog(null, "No puede realizar más transacciones");
    Administrador.setVisible(true);
    this.setVisible(false);
}
}
```

En este metodo se envian las diferentes transacciones para que en ella puedan almacenarse cada una de las que se envian y asi poder mostrar esta lista al final

```
public void Transacciones(int fila, int cuenta, String tipo, int dinero) {
```

La cual tiene como parametro la persona "fila", la cuenta en la cual se efectuo dicha transaccion , el tipo de transaccion en la cual se valida que si se acredita o se debita y finalmente cuanto dinero se efectuo,