

MANUAL TÉCNICO LOGIN

```
public class Login extends javax.swing.JFrame {  
  
    /**  
     * Creates new form Login  
     */  
    public Login() {  
        initComponents();  
        this.getContentPane().setBackground(Color.cyan);  
    }  
}
```

Esta es la primera parte del módulo login dónde se describirá cómo funciona el mismo. En la captura de arriba se puede ver que se define una clase con nombre login y se establece el color de fondo a un color cyan.

```
private void btnIngresarActionPerformed(java.awt.event.ActionEvent evt) {  
    String user= txtUser.getText();  
    String contra= txtContra.getText();  
    try{  
  
        if(user.isEmpty() || contra.isEmpty())  
            JOptionPane.showMessageDialog(null,"Campo Vacío Revise");  
  
        else  
            if(DB.Login(user,contra)==1)  
            {  
                JOptionPane.showMessageDialog(null,"Bienvenido");  
                FormPrincipal main = new FormPrincipal();  
                this.setVisible(false);  
                main.setVisible(true);  
            }  
            else{  
                JOptionPane.showMessageDialog(null,"Usuario o Contraseña Incorrecto");  
            }  
    } catch (ClassNotFoundException ex) {  
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);  
    }  
}
```

En el manual de usuario se puede ver que tenemos un botón de ingresar. Al darle clic pasa toda esta secuencia de código. Usamos un try catch al usar el login con la base de datos (MySQL) porque cuando se llama a la base puede llegar a ocurrir una excepción SQL si llega haber algún problema de la base de datos.

Primero, revisamos si los campos del usuario y contraseña están vacíos, si ocurre se mostrará un mensaje que diga "Campo Vacío Revise". Si todo está bien, pasa al siguiente if donde se llama a la base de datos para revisar si existe el usuario y si su contraseña es correcta. La decisión te tiene si es igual a uno porque ese se consideró como un valor ideal para la decisión de poder entrar al sistema. Si se devuelve un uno, entonces se mostrará "Bienvenido" se cerrará la pantalla del login y se abrirá el módulo principal del sistema. Si la decisión devuelve un cero saltará al último else donde dirá "Usuario o Contraseña Incorrecto".

```

public class DB {
    private static Connection cnx = null;
    public static Connection conexion() throws SQLException, ClassNotFoundException {
        if (cnx == null) {
            try {
                Class.forName("com.mysql.jdbc.Driver");
                cnx = DriverManager.getConnection("jdbc:mysql://localhost/clinica_unicah", "root", "");
            } catch (SQLException ex) {
                JOptionPane.showMessageDialog(null, ex);
            } catch (ClassNotFoundException ex) {
                JOptionPane.showMessageDialog(null, ex);
            }
        }
        return cnx;
    }
}

```

Cuando se llama a DB.Login lo primero que esa función hará es ir a la función de conectar a la base. La imagen de arriba muestra cómo se conecta JDBC a MySQL.

```

public static int Login(String user, String pass) throws ClassNotFoundException {
    int res = 0;
    String usuRes = "", passRes = "";
    String sql = "";
    sql = ("select descripcion_usuario, password_usuario from usuario where descripcion_usuario='" + user + "' and password_usuario='" + pass + "'");
    try {
        DB.conexion();
        Statement s = cnx.createStatement();
        ResultSet rs = s.executeQuery(sql);
        if (rs.next()) {
            usuRes = rs.getString("descripcion_usuario");
            passRes = rs.getString("password_usuario");
            res = 1;
        } else {
            return 0;
        }
        rs.close();
        s.close();
        DB.cerrar();
        return res;
    } catch (SQLException ex) {
        res = 0;
        return res;
    }
}

```

Esta es la función DB.Login donde anteriormente se dijo que primero va a conectar la base de datos y devolverá un cero o un uno. Esta función ocupa de parámetros uno para el usuario y otro para la contraseña. Para poder ejecutar una consulta ocupamos utilizar statement, resultset y una propiedad del resultset que es next. Statement dice que ejecutaremos la consulta. Resultset ejecuta esa consulta. Se utiliza el next para poder traer datos desde la base. Si todo funciona perfectamente, tenemos una variable "res" que es la que devuelve esta función. Al antes de retornar se cierra el resultset, statement y la conexión a la base de datos.

```

private void btnSalirActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    System.exit(0);
}

```

En el formulario de Login también hay un botón salir dónde se cierra el programa con un system.exit (0).

MANUAL TECNICO DE INGRESO ATA

El ATA es la recolección de datos del paciente, para que con esta información el medico pueda recetar o elaborar una prescripción medica y con esto poder curar o rebajar los síntomas del paciente. Este se elabora para beneficio de la universidad, a modo que todo se vuelva un poco mas tecnológico, y evitar mucho papeleo, que puede llegar a hacer tedioso. La base de datos, esta constituida de la siguiente manera:

Llave Primaria	Paciente	Expediente	ATA	Medicamento
Llave Secundaria	Identidad_Paciente	Numero_Expediente	Codigo_ATA	Codigo_Medicamento
	Nombre_Paciente	Identidad_Paciente	Nombre_Paciente	Descripcion_Medicamento
	Edad_Paciente	Fecha_Expediente	Carera_Paciente	Tipo_Medicamento
	Sexo_Paciente	Hora_Ingreso	Sintoma	
	Domicilio_Paciente	Hora_Egreso	Codigo_Medicamento	
	Carrera_Paciente	Sintoma_Principal		
		HEA		
		Peso		
		Talla		
		IMC		
		Frecuencia_Cardiaca		
		Frecuencia_Respiratoria		
		Presion_Arterial		
		Temperatura		
		Glasgow		
		Indice_Choque		
		Calidad_Pulso		
		Frialdad_Distal		
		Color_Piel		
		Llenado_Capilar		
		Datos_Contribuyentes		
		Diagnostico		
		Tratamiento		
		Interconsulta		
		Comentario		

El ATA esta compuesta en su mayor parte por la tabla paciente y la tabla ATA. Se compone de ambas porque en su particularidad el ATA solo es el registro del paciente, tomar sus diagnosticos actuales, y elaborar una prescripción medica que ayude a alivianar o erradicar completamente los síntomas de el paciente. Los campos de la base de datos en sql están constituidos de la siguiente manera:

```

30
31 CREATE TABLE `paciente` (
32   `idpaciente` varchar(15) NOT NULL,
33   `idcarrera` varchar(30) NOT NULL,
34   `nombre_paciente` varchar(30) NOT NULL,
35   `apellido_paciente` int(11) NOT NULL,
36   `edad_paciente` int(11) NOT NULL,
37   PRIMARY KEY (`idpaciente`),
38   UNIQUE KEY `idpaciente_UNIQUE` (`idpaciente`)
39 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
40
41 CREATE TABLE `ata` (
42   `idata` int(11) NOT NULL AUTO_INCREMENT,
43   `iddoctor` varchar(15) NOT NULL,
44   `idmedicamento` int(11) NOT NULL,
45   `nombre_paciente_ata` varchar(50) NOT NULL,
46   `apellido_paciente_ata` varchar(50) NOT NULL,
47   `edad_paciente_ata` int(11) NOT NULL,
48   `sintomas_paciente_ata` varchar(100) NOT NULL,
49   `fecha_ata` date NOT NULL,
50   PRIMARY KEY (`idata`),
51   UNIQUE KEY `idata_UNIQUE` (`idata`),
52   KEY `iddoctor_idx` (`iddoctor`),
53   KEY `idmedicamento_idx` (`idmedicamento`),
54   CONSTRAINT `iddoctor` FOREIGN KEY (`iddoctor`) REFERENCES `doctor` (`iddoctor`) ON DELETE NO ACTION ON UPDATE NO ACTION,
55   CONSTRAINT `idmedicamento` FOREIGN KEY (`idmedicamento`) REFERENCES `medicamento` (`idmedicamento`) ON DELETE NO ACTION ON UPDATE NO ACTION
56 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
57
58 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
59
60 CREATE TABLE `expediente` (
61   `idexpediente` varchar(10) NOT NULL,
62   `idpaciente` varchar(13) NOT NULL,
63   `fecha_expediente` datetime NOT NULL,
64   `hora_ingreso` datetime NOT NULL,
65   `hora_egreso` datetime NOT NULL,
66   `sintoma_principal` varchar(50) NOT NULL,
67   `hea` varchar(200) NOT NULL,
68   `peso_paciente` double NOT NULL,
69   `talla_paciente` double NOT NULL,
70   `frecuencia_cardiaca_paciente` double NOT NULL,
71   `frecuencia_respiratoria_paciente` double NOT NULL,
72   `presion_arterial_paciente` double NOT NULL,
73   `temperatura_paciente` double NOT NULL,
74   `color_piel_paciente` varchar(15) NOT NULL,
75   `llenado_capilar_paciente` varchar(15) NOT NULL,
76   `datos_contribuyentes_paciente` varchar(250) NOT NULL,
77   `diagnostico` varchar(250) NOT NULL,
78   `tratamiento` varchar(250) NOT NULL,
79   `interconsulta` varchar(250) NOT NULL,
80   `comentario_adicional` varchar(250) NOT NULL,
81   PRIMARY KEY (`idexpediente`),
82   UNIQUE KEY `idexpediente_UNIQUE` (`idexpediente`),
83   KEY `idpaciente_idx` (`idpaciente`),
84   CONSTRAINT `idpaciente_expediente` FOREIGN KEY (`idpaciente`) REFERENCES `paciente` (`idpaciente`) ON DELETE NO ACTION ON UPDATE NO ACTION
85 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```