INNOVA COLOMBIA,
Área: Recursos Humanos

Sebastian Pabon López, y Hanna Katherine Amaya Rojas, *sebastian.pabon@pi.edu.co*, *hanna.amaya@pi.edu.co*, *IEEE Politécnico Internacional, Bogotá*

***Resumen*—Este trabajo se hizo con el fin de profundizar en los aprendizajes de la clase Estructura de Datos. Innova Colombia es la empresa a la cual se le realizó una base de datos, en dónde se tendrá el registro de la hora de entrada y salida de los empleados.**

***Abstract*--This work was done in order to deepen the learning of the Data Structure class. Innova Colombia is the company to which a database was made, where the record of the entry and exit time of the employees will be kept.**

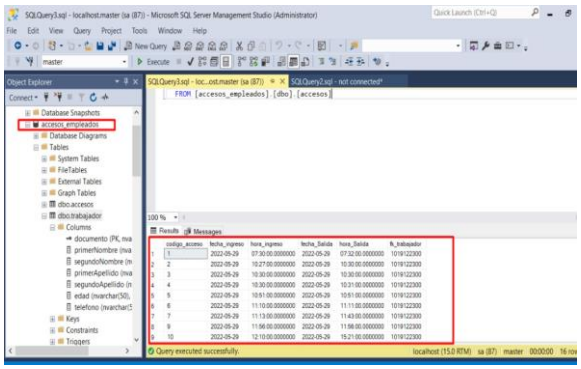
Introducción

Este documento contiene la información detallada de lo que se llevó acabo para crear una base de datos, la cual permitirá poder tener un control sobre la entrada y salida de los empleados de la empresa Innova Colombia.

El desarrollo de la base e interfaz se realizó con los lenguajes: programación, de Java a través de NetBeans y desarrollo, el sistema de gestión de bases de datos relacional Sql Server.

TABLE I
INTERFAZ GRÁFICA

Fig1. Interfaz que permite hacer en registro de entrada y salida.



.

Fig2. Base datos Sql Server.

I. DESARROLLO

A. *Interfaz*

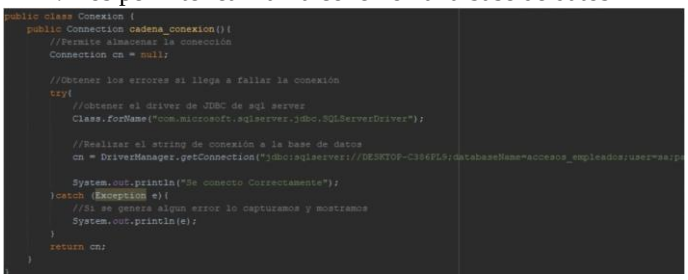
Figura1. La interfaz en java, es como un Java Class, no obstante, esta solo tiene constantes estáticas y método abstracto. Java usa la interfaz para implementar herencia múltiple. Así mismo es posible implementar diferentes interfaces Java. Es de suma importancia diferenciar los métodos, ya que, en la interfaz, estos son implícitamente públicos y abstractos. La principal función de la interfaz es que esta, pueda comunicar información a través de ella hacía algún tipo de dispositivo u o sistema.

Base de Datos

Figura2. Sql Server Management Studio es una aplicación de software lanzada por primera vez con Microsoft SQL Server 2005 que se utiliza para configurar, administrar y administrar todos los componentes dentro de Microsoft SQL Server. Es el sucesor del Enterprise Manager en SQL 2000 o antes. En este programa se crearon las tablas correspondientes para el registro de la información.

B. *Código*

1. Nos permite realizar la conexión a la base de datos



Politécnico Internacional: Sebastian Pabon y Hanna Amaya

```java
private String[] consultarTrabajador(String documento){
    try{
        //Crear una nueva conexion a la base de datos
        Connection con = cx.cadena_conexion();
        ResultSet rs;

        //Consultar información del trabajador por su documento
        PreparedStatement ps = con.prepareStatement("SELECT * FROM trabajador WHERE documento = ?");
        ps.setString(1, documento);
        rs = ps.executeQuery();

        //LLena la información del trabajador en un arreglo con la información obtenida
        String[] infoPersona = new String[7];
        int rowCount = 0;
        int codigoAcceso = -1;
        while(rs.next()) {
            rowCount++;
            //llena el arreglo con la información del trabajador
            infoPersona[0] = rs.getString("documento");
            infoPersona[1] = rs.getString("primerNombre");
            infoPersona[2] = rs.getString("segundoNombre");
            infoPersona[3] = rs.getString("primerApellido");
            infoPersona[4] = rs.getString("segundoApellido");
            infoPersona[5] = rs.getString("edad");
            infoPersona[6] = rs.getString("telefono");
        }

        return infoPersona;
    } catch(SQLException e){
        String[] infoPersona = new String[7];
        JOptionPane.showMessageDialog(null, e.toString());
        return infoPersona;
```

2. Consulta la información de un trabajador y la entrega en un arreglo

```java
private void guardarRegistro(String documento, String fecha,String hora){
    try{
        //Crear una nueva conexion a la base de datos
        Connection con = cx.cadena_conexion();
        ResultSet rs;

        //Se genera la petición para consultar si el trabajador tiene un ingreso activo
        PreparedStatement ps = con.prepareStatement("SELECT * FROM accesos WHERE fk_trabajador = ? AND fecha_Salid
        ps.setString(1, documento);
        rs = ps.executeQuery();

        //Cuenta la cantidad de registros encontrados y guarda su codigo de acceso
        int rowCount = 0;
        int codigoAcceso = -1;
        while(rs.next()) {
            rowCount++;
            codigoAcceso = rs.getInt("codigo_acceso");
        }

        //Comprueba si tiene registros para registrar la salida o el ingreso
        if(rowCount > 0 && codigoAcceso != -1){
            //Genera el ingreso del trabajador
            actualizarSalida(fecha, hora, codigoAcceso);
        }else{
            //Genera la salida del trabajador
            generarIngreso(fecha, hora, documento);
        }
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e.toString());
    }
}
```

3. Permite registrar la salida o ingreso de un trabajador

```java
private void generarIngreso(String fecha, String hora, String documento){
    try{
        //Crear una nueva conexion a la base de datos
        Connection con = cx.cadena_conexion();
        ResultSet rs;

        //Se genera la petición para actualizar un registro a la base de datos
        PreparedStatement ps = con.prepareStatement("INSERT INTO accesos (fecha_ingreso, hora_ingreso, fk_trabajado

        //Añadir los campos que seran actualizados con la nueva información
        ps.setString(1, fecha);
        ps.setString(2, hora);
        ps.setString(3, documento);

        // ps.execute udate // se guardan
        ps.executeUpdate();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e.toString());
    }
}
```

4.Permite generar el registro de ingreso de una nueva persona

```java
private void actualizarSalida(String fecha, String hora, int codigo){
    try{
        //Crear una nueva conexion a la base de datos
        Connection con = cx.cadena_conexion();
        ResultSet rs;

        //Se genera la petición para actualizar un registro a la base de datos
        PreparedStatement ps = con.prepareStatement("UPDATE accesos SET fecha_Salida=?,hora_Salida=? WHERE codigo_a

        //Añadir los campos que seran actualizados con la nueva información
        ps.setString(1, fecha);
        ps.setString(2, hora);
        ps.setInt(3, codigo);

        //ejecuta la actualización del registro
        ps.executeUpdate();
    } catch(SQLException e){
        JOptionPane.showMessageDialog(null, e.toString());
    }
}
```

5. Permite generar el registro de salida de una nueva persona

II. Conclusiones

Poder tener una base de datos ayuda a tener un control acerca de la asistencia de los empleados en este caso de la empresa Innova Colombia. En este caso el mejor programa para desarrollar esta base fue en Sql Server en dónde se crearon las respectivas tablas y conexión con Java en dónde de hicieron las interfaces las cuales son la vista de la base.

Referencias

[1] https://softtrader.es/blog-microsoft/que-es-sql-server-management-studio
[2] https://www.arkaitzgarro.com/java/capitulo-18.html