Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

Aplicación para un Sistema de Gestión Institucional utilizando Java y MySql

Briones Shariff [1]; Menéndez E. Marco [2]; Menéndez Mauro [3]; Peñafiel Nataly [4]; Sánchez Joan [5]

[1,2,3,4,5] *Universidad Técnica de Manabí-Facultad de Ciencias Informáticas –Carrera de Tecnologías de la Información, Portoviejo, Ecuador*

Resumen: La gestión Institucional permite a las escuelas, colegios y universidades tener un control multiplataforma para el manejo de las autoridades y la comunidad académica en general, esta medida fomenta la organización que debe llevar una institución académica. Por este motivo se ha desarrollado una aplicación para el control y gestión de las personas que conforman una institución educativa, el sistema cuenta con conexión a una base de datos para mantener todos los datos almacenados, mediante el uso de métodos y técnicas de programación y gestión de bases de datos. El análisis, diseño e implementación del sistema de gestión institucional fue realizado siguiendo el proceso de la programación orientada a objetos POO, que permite crear una serie de objetos y a estos definirlos en características o atributos, además de la reutilización de código mediante la herencia, el sistema está enfocado a resultados positivos que el usuario quiere visualizar en las acciones que realiza el software mediante las múltiples opciones de ingreso y consultas de datos, la interacción entre el usuario y la aplicación debe ser de sumo agrado y garantizando la calidad del software.

Palabras clave: Gestión, MySql, POO, Herencia, Software

Application for an Institutional Management System using Java and MySql

Abstract: Institutional management allows schools, colleges and universities to have multi-platform control for the management of the authorities and the academic community in general, this measure encourages the organization that an academic institution must carry. For this reason, an application has been developed for the control and management of the people who make up an educational institution, the system has a connection to a database to keep all the data stored, through the use of programming and management methods and techniques. of databases. The analysis, design and implementation of the institutional management system was carried out following the OOP object-oriented programming process, which allows creating a series of objects and defining them in characteristics or attributes, in addition to code reuse through inheritance, The system is focused on positive results that the user wants to see in the actions carried out by the software through the multiple input options and data queries, the interaction between the user and the application must be of the utmost pleasure and guarantee the quality of the software.

Keywords: Management, MySql, POO, Inheritance, Software

1. INTRODUCCIÓN

Dentro de la gestión académica existen diferentes técnicas para el control y gestión de usuarios que permiten realizar la recolección a gran escala de información. Con este propósito, se ha implementado una aplicación que facilite el desempeño de gestión de las autoridades y la comunidad en general de la Institución, que mediante el uso de un scrip en MySql permite la manipulación de la información dentro de una base de datos ya establecida con la finalidad de facilitar y cuidar la información de cada miembro de la institución.

Se desarrolló una aplicación en Java con interfaz gráfica para facilitar la interacción del administrador o administradores en el ingreso y consulta de información, adicionalmente una ventana de ingreso de usuarios para que la gestión sea privada.

[1, 2, 3, 4, 5] Estudiante

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

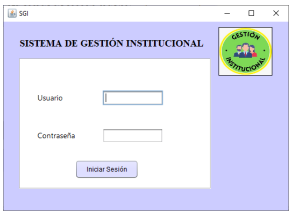

Ilustración 1. Ingreso de Usuario.
Fuente: Autor

El proyecto se realiza en un sistema con arquitectura de x64 que es la compatibilidad que comúnmente se utiliza en computadores actualmente.

Una vez inicializado el proyecto es importante ejecutar los servicios de XAMPP, el cual permite crear la conexión con la base de datos:

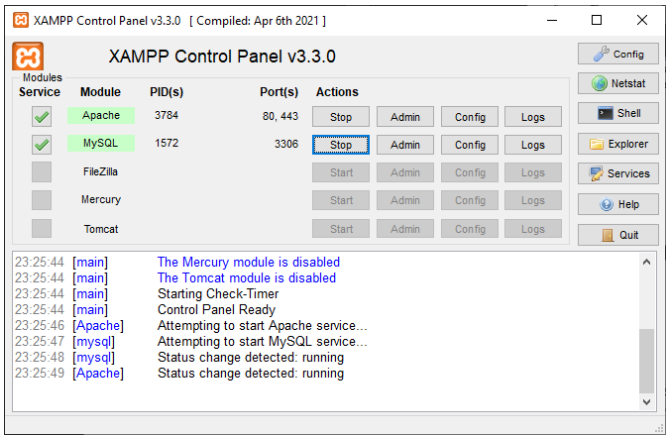

Ilustración 2. Panel de control XAMPP.
Fuente: Autor

2. METODOLOGÍA

2.1.Planificación

Para llevar a cabo este proyecto se requerirá de ciertas aplicaciones que permitirán la conexión de JAVA con MYSQL, dentro del IDE se desarrollará la GUI que se compondrá de varias ventanas que definirán acciones distintas para cada tipo de usuario que interactúe con ella.
Dichas interacciones en la GUI permitirán que el programa desarrollado se direccione en la realización de consultas, ingresos y eliminación de datos, los mismos que tendrán que ser validados para mantener la eficacia en la información.

2.2.Diseño

En el primer diseño, se realiza una interfaz de concepto accesible donde el usuario interactúa con la interfaz, de modo que el ingreso es más práctico de modo que su principal función es validar y cumplir con los ingresos de varios usuarios.

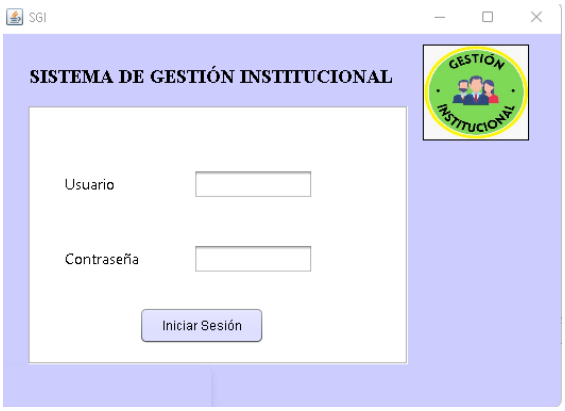

Ilustración 3. Inicio.
Fuente: Autor

El segundo diseño se muestra porque los datos en el ingreso fueron validados y por la categoría establecida en el usuario se presenta la interfaz Directivo, allí se realizarán validaciones de datos, y podrá ejecutar las funciones que tiene permitido dicho usuario.

Ilustración 4. Directivo.
Fuente: Autor

Como tercera parte del diseño se muestra ventana del usuario tipo Estudiante, en el mismo se realizan validaciones de los datos del estudiante y las funciones que tiene descritas permiten consultar, limpiar y regresar a la sesión de origen.

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

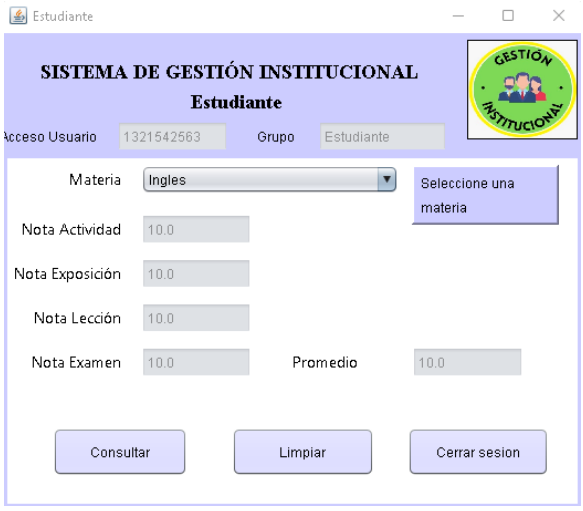

Ilustración 5. Estudiante.
Fuente: Autor

La cuarta parte del diseño hace mención a la interfaz del usuario auxiliar de servicios, en el que las únicas funciones que se permiten son: consultar y regresar al inicio.

Ilustración 6. Auxiliar de servicio.
Fuente: Autor

La última parte del diseño de este programa se basa en la interfaz del usuario de Profesor, del cual se podrá visualizar las funciones consultar el sueldo, y la más importante es el ingreso de notas que sin estos datos Estudiante no podrá visualizar sus calificaciones, además se ubica un cierre de sesión, hay que enfatizar que cada proceso esta validado.

Ilustración 7. Profesor.
Fuente: Autor

2.3.Desarrollo

El proyecto cuenta con siete diseños de interfaz, pero como parte estándar, cada clase creada hace uso de las mismas librerías y asimismo todas contemplan el mismo formato para la conexión entre Java y MySQL.

```
import java.awt.Image;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

//Variables para realizar la conexion a la BD
    int band =0;
 public String driver = "com.mysql.cj.jdbc.Driver";
 public String database = "sgi";
 public String hostname = "localhost";
 public String port = "3306";
 public String url = "jdbc:mysql://" + hostname + ":"
+ port + "/" + database + "?useSSL=false";
//public String url = "jdbc:mysql://" + hostname +
":/" + database;
 public String username = "root";
 public String password = "";
```

La interfaz principal o inicio solo contiene dos TextField para el ingreso de los datos de usuario y contraseña, sin embargo, las operaciones son desarrolladas en el botón de “Iniciar sesión”.

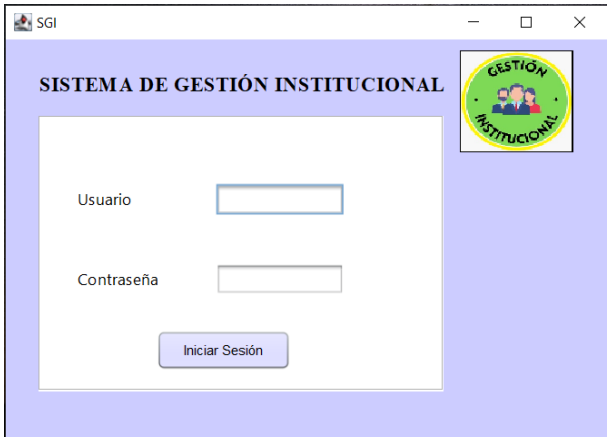

Ilustración 8. Inicio.
Fuente: Autor

Dentro de la función del botón “Iniciar sesión” se guarda el usuario y contraseña, se conecta a la base de datos y se realiza la consulta, si los datos no coinciden con los que se encuentran almacenados se

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

muestra un mensaje indicando que alguno de los datos es incorrecto.
Además, usuario tiene una validación ya que la longitud debe ser igual a 10, dado que es un número de cédula.
Si todos los datos se encuentran, se muestra una nueva ventana que dependerá de la categoría (directivo, auxiliar de servicio, profesor y estudiante) y se cierra la interfaz de inicio.

```
//Cerrar sesion
private void
jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
 UsuarioCed = JT_Usuario.getText();
 String categoria=null;
 //Dimension del usuario (cedula)
 char[] aux=JT_Usuario.getText().toCharArray();
 if(aux.length==10){
 try{
 Class.forName(driver);
 }catch(ClassNotFoundException e){
 System.out.println(e.getMessage());
 }
 try{
 //Realiza la conexion y ejecuta la consulta SQL
 //System.out.println(url);
 Connection conexion=DriverManager.getConnection(url,
username, password);
 Statement estatuto=conexion.createStatement();
 ResultSet rs=estatuto.executeQuery("select * from
usuarios where usuario='"+JT_Usuario.getText()+"' and
contra='"+JT_Contra.getText()+"'");

 while(rs.next()){
 band=1;
 Object[] t=new Object[3];
 for(int i=0;i<t.length;i++){
 t[i]=rs.getObject(i+1);
 categoria = t[2]+"";}
 //Se guarda la categoria
 CategoriaM = categoria;
 }
 //Datos no coinciden, se elimina la contraseña
 if(band==0){
 JFrame jFrame = new JFrame();
 JOptionPane. showMessageDialog(jFrame,"El usuario o
contraseña son incorrectos");
 JT_Contra.setText("");
 }
 else{
 if(categoria.equals("Directivo")){
     this.setVisible(false);
     Directivo Dire = new Directivo();
     Dire.setVisible(true);
 }else if(categoria.equals("Auxiliar de servicio")){
     this.setVisible(false);
     Auxiliar_Servicio auxi = new
Auxiliar_Servicio();
     auxi.setVisible(true);
 }else if(categoria.equals("Profesor")){
     this.setVisible(false);
     Profesor profe = new Profesor();
     profe.setVisible(true);
 }else if(categoria.equals("Estudiante")){
     this.setVisible(false);
     Estudiante est = new Estudiante();
     est.setVisible(true);
 }
}
 //Cierra la conexion
 estatuto.close();
 conexion.close();
 }catch(Exception e)
{System.out.println(e.getMessage());}
 }else
 //No coincide longitud de cedula
javax.swing.JOptionPane.showMessageDialog(this,"¡Usua
rio incorrecto!","Error",0);
 }
```

Directivo es el usuario con mayor rango, puesto que es el único que administra al resto de categorías y por ende puede crear nuevos usuarios. En esta interfaz Directivo tiene tres botones: Consultar sueldo, Administrar usuario y Cerrar Sesión.

Ilustración 9. Directivo.
Fuente: Autor

El primero botón muestra el sueldo, pero primero hace la consulta y verifica si existe dicho valor y lo presenta. Una vez seleccionado Administrar usuarios se ejecuta el llamado de otra interfaz "Operaciones" para el llenado de datos en caso de ser nuevo o para realizar consultas. Y por último el botón de "Cerrar sesión" deja de mostrar la GUI de directivo por la interfaz de inicio.

```
//Consultar sueldo
private void jButton2ActionPerformed
(java.awt.event.ActionEvent evt) {
 //Verificar usuario
 char[] aux=Usuario_A.getText().toCharArray();
 if(aux.length==10){
 try{
 Class.forName(driver);
 }catch(ClassNotFoundException e){
 System.out.println(e.getMessage());
 }
 try{
 //Realiza la conexion y ejecuta la consulta SQL
 //System.out.println(url);
 Connection conexion=DriverManager.getConnection(url,
username, password);
 Statement estatuto=conexion.createStatement();
 ResultSet rs=estatuto.executeQuery("select * from
directivo where cedula='"+Usuario_A.getText()+"'");
 //Almacena las columnas en un vector
 while(rs.next()){
 Object[] t=new Object[2];
```

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

```
 for(int i=0;i<t.length;i++){
 t[i]=rs.getObject(i+1);
 sueldo = t[1]+"";}
 }
 JFrame jFrame = new JFrame();
 JOptionPane. showMessageDialog(jFrame,"Su sueldo es:
"+sueldo);
 estatuto.close();
 conexion.close();
 }catch(Exception e)
{System.out.println(e.getMessage());}
 }else
 //En caso de no coincider usuario
 javax.swing.JOptionPane.showMessageDialog(this,
 "¡Usuario incorrecto!","Error",0);
 }
//Administrar usuario
private void jButton1ActionPerformed
(java.awt.event.ActionEvent evt) {
 //Abre el formulario Operaciones y cierra el actual
     this.setVisible(false);
     Operaciones ope = new Operaciones();
     ope.setVisible(true);
    }
//Cerrar sesion
private void jButton3ActionPerformed
(java.awt.event.ActionEvent evt) {
     this.setVisible(false);
     Inicio ini = new Inicio();
     ini.setVisible(true);
    }
```

En Auxiliar de servicio solo se muestran los datos del usuario y existen dos botones con la función de Consultar sueldo y Cerrar sesión.

Ilustración 10. Auxiliar de servicio.
Fuente: Autor

Para observar el sueldo, primero se realiza una consulta verificando los datos del auxiliar y luego presenta el valor del sueldo.
Al seleccionar Cerrar sesión se deja de mostrar la interfaz del auxiliar de servicio y se cambia a la de inicio.

```
//Consultar sueldo
private void jButton2ActionPerformed
(java.awt.event.ActionEvent evt) {
 //Verificar dimension del usuario (cedula)
  char[] aux=Usuario_A.getText().toCharArray();
  if(aux.length==10){
  try{
  Class.forName(driver);
  }catch(ClassNotFoundException e){
   System.out.println(e.getMessage());
  }
  try{
  //Realiza la conexion y ejecuta la consulta SQL
  //System.out.println(url);
Connection conexion=DriverManager.getConnection(url,
username, password);
Statement estatuto=conexion.createStatement();
ResultSet rs=estatuto.executeQuery("select * from
auxiliar_servicio where
cedula='"+Usuario_A.getText()+"'");
//Almacena las columnas en un vector
while(rs.next()){
Object[] t=new Object[2];
for(int i=0;i<t.length;i++){
t[i]=rs.getObject(i+1);
sueldo = t[1]+"";}
}
JFrame jFrame = new JFrame();
JOptionPane. showMessageDialog(jFrame,"Su sueldo es:
"+sueldo);
//Se cierra la conexion
estatuto.close();
conexion.close();x
}catch(Exception e)
{System.out.println(e.getMessage());}
}else
 javax.swing.JOptionPane.showMessageDialog(this,
 "¡Usuario incorrecto!","Error",0);
}
//Cerrar sesion
private void jButton3ActionPerformed
(java.awt.event.ActionEvent evt) {
        //Cierra la sesion y regresa al inicio
        this.setVisible(false);
        Inicio ini = new Inicio();
        ini.setVisible(true);
}
```

La interfaz de Profesor muestra los campos de usuario, categoría y materia que imparte, además tiene tres botones cada uno con diferentes funciones.

Ilustración 11. Profesor.
Fuente: Autor

La consulta del sueldo de profesor se lleva a cabo como en los casos anteriores. Sin embargo, el botón

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

Ingresar notas fue configurado para llamar a otra interfaz que se encargara de este proceso, finalmente con el botón Cerrar se deja de presentar interfaz profesor y regresa a la ventana de inicio.

```
//Consultar sueldo
private void jButton1ActionPerformed
(java.awt.event.ActionEvent evt) {
//Verificar dimension del usuario (cedula)
 char[] aux=Usuario_A.getText().toCharArray();
 if(aux.length==10){
 try{
 Class.forName(driver);
 }catch(ClassNotFoundException e){
  System.out.println(e.getMessage());
 }
 try{
 //System.out.println(url);
 Connection conexion=DriverManager.getConnection(url,
username, password);
 Statement estatuto=conexion.createStatement();
 ResultSet rs=estatuto.executeQuery("select * from
profesor where cedula='"+Usuario_A.getText()+"'");

 while(rs.next()){
 Object[] t=new Object[3];
 for(int i=0;i<t.length;i++){
 t[i]=rs.getObject(i+1);
 sueldo = t[1]+"";
 }
}
JFrame jFrame = new JFrame();
JOptionPane. showMessageDialog(jFrame,"Su sueldo es:
"+sueldo);
//Se cierra la conexion
estatuto.close();
conexion.close();
}catch(Exception e)
{System.out.println(e.getMessage());}
}else
  javax.swing.JOptionPane.showMessageDialog(this,
"¡Usuario incorrecto!","Error",0);
 }
//Ingreso de notas
private void jButton2ActionPerformed
(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Ingreso_Notas notas = new Ingreso_Notas();
        notas.setVisible(true);
}
//Cerrar sesion
private void jButton3ActionPerformed
(java.awt.event.ActionEvent evt) {
this.setVisible(false);
Inicio ini = new Inicio();
ini.setVisible(true);
}
```

Ingreso de notas se habilita luego que profesor necesite registra nuevas calificaciones, cabe recalcar que luego que estás son ingresadas no existe modificación alguna.
Este formulario consta de un comboBox con los nombres de estudiantes; varios TextField y tres botones.

Ilustración 12. Ingreso de notas.
Fuente: Autor

Primero se mostrarán los campos desactivados al igual que el botón Ingresar, dado que no contiene datos. Cuando se da click sobre Nuevo ingreso se podrá elegir al estudiante y los campos deben aparecer en blanco para su posterior llenado, consecuentemente Ingresar se desactiva y guardará los valores, volviendo a desactivar todo.
Mientras, la función que cumple el botón Regresar es volver a la interfaz de profesor.

```
//Nuevo ingreso
private void jButton2ActionPerformed
(java.awt.event.ActionEvent evt) {
        jComboBox1.setEnabled(true);
        JT_NActividad.setText("");
        JT_NActividad.setEnabled(true);
        JT_NExposicion.setText("");
        JT_NExposicion.setEnabled(true);
        JT_NLeccion.setText("");
        JT_NLeccion.setEnabled(true);
        JT_NExamen.setText("");
        JT_NExamen.setEnabled(true);
        jButton1.setEnabled(true);
    }
//Ingresar
private void jButton1ActionPerformed
(java.awt.event.ActionEvent evt) {
try{
 Class.forName(driver);
 }catch(ClassNotFoundException e){
 System.out.println(e.getMessage());
 }
 try{
 //Realiza la conexion y registra la informacion
 Connection conexion=DriverManager.getConnection(url,
username, password);
 Statement estatuto=conexion.createStatement();
 Statement estatuto2=conexion.createStatement();
 Statement estatuto3=conexion.createStatement();
 Statement estatuto4=conexion.createStatement();
 Statement estatuto5=conexion.createStatement();
ResultSet rs=estatuto.executeQuery("select cedula
from persona, usuarios where persona.cedula =
usuarios.usuario and usuarios.categoria =
'Estudiante'");
 ResultSet rs1=estatuto2.executeQuery("select nombres
from persona, usuarios where persona.cedula =
```

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

```
usuarios.usuario and usuarios.categoria =
'Estudiante'");
 ResultSet rs2=estatuto3.executeQuery("select
apellidos from persona, usuarios where persona.cedula
= usuarios.usuario and usuarios.categoria =
'Estudiante'");
 int i=1;
 while(rs.next() && rs1.next() && rs2.next()){
    if((rs2.getObject(i)+"
"+rs1.getObject(i)).equals(jComboBox1.getSelectedItem
().toString())){
ResultSet rs5=estatuto5.executeQuery("select * from
notas where cedula='"+(rs.getObject(i)+"")+"' and
materia='"+(JT_Materia.getText())+"'");
        String comprobar = "no";
        while(rs5.next()){
        comprobar = "si";
        }
        if(comprobar.equals("no")){
        band=1;
String sql="insert into notas
values('"+(rs.getObject(i)+"")+"','"+(JT_Materia.getT
ext())+"','"+Double.parseDouble(JT_NActividad.getText
())+"','"+Double.parseDouble(JT_NExposicion.getText()
)+"','"+Double.parseDouble(JT_NLeccion.getText())+"',
'"+Double.parseDouble(JT_NExamen.getText())+"')";
        estatuto4.executeUpdate(sql);
        }else {
        band=0;
        }
    }
}
  if(band==0){
javax.swing.JOptionPane.showMessageDialog(this,"¡Este
estudiante ya contiene notas en la materia
"+(JT_Materia.getText())+" !","Error",0);
        JT_NActividad.setText("");
        JT_NActividad.setEnabled(true);
        JT_NExposicion.setText("");
        JT_NExposicion.setEnabled(true);
        JT_NLeccion.setText("");
        JT_NLeccion.setEnabled(true);
        JT_NExamen.setText("");
        JT_NExamen.setEnabled(true);
        jButton1.setEnabled(true);
 }
else{
 JFrame jFrame = new JFrame();
 JOptionPane. showMessageDialog(jFrame,"Notas
registradas correctamente!");
 jComboBox1.setEnabled(false);
 JT_NActividad.setEnabled(false);
 JT_NExposicion.setEnabled(false);
 JT_NLeccion.setEnabled(false);
 JT_NExamen.setEnabled(false);
 jButton1.setEnabled(false);
 }
 estatuto.close();
 estatuto2.close();
 estatuto3.close();
 estatuto4.close();
 estatuto5.close();
 conexion.close();
 }catch(Exception e){
javax.swing.JOptionPane.showMessageDialog(this,"¡No
se pudo ingresar las notas porque uno de los campos
se encuentra vacio!","Error",0);
 }
    }
//Regresar
private void jButton3ActionPerformed
(java.awt.event.ActionEvent evt) {
        this.setVisible(false);
        Profesor profe = new Profesor();
        profe.setVisible(true);
    }
```

Estudiante solo tiene permitido ver las notas por materia. La interfaz cuenta con un comboBox, y cuatro textField donde se mostrarán las notas ingresadas por el docente y otro con el promedio.

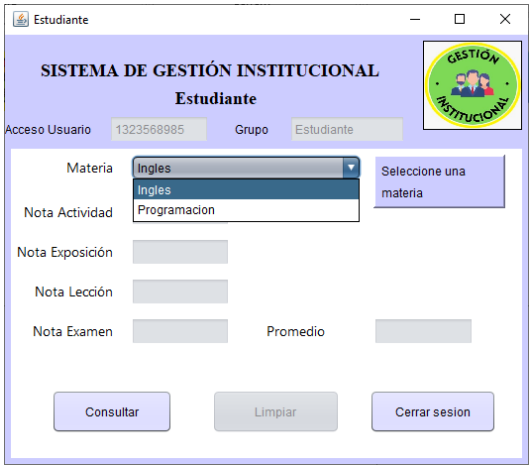

Ilustración 13. Estudiante.
Fuente: Autor

El botón Consultar se encarga de consultar los datos que luego serán exhibidos, dentro de esta función se realiza una operación matemática para obtener el promedio de las notas.
Si se desea consultar nuevamente se ubicó un mensaje indicando que se debe Limpiar, este botón dejara a los campos en blancos, listos para una nueva consulta. Cerrar sesión devuelve al usuario a la interfaz de inicio.

```
//Consultar
private void jButton1ActionPerformed
(java.awt.event.ActionEvent evt) {
 if((JT_NActividad.getText()).equals("")){
 try{
 Class.forName(driver);
 }catch(ClassNotFoundException e){
 System.out.println(e.getMessage());
 }
 try{
 //System.out.println(url);
 Connection conexion=DriverManager.getConnection(url,
username, password);
 Statement estatuto=conexion.createStatement();
 ResultSet rs=estatuto.executeQuery("select * from
notas where cedula='"+Usuario_A.getText()+"' and
materia='"+jComboBox1.getSelectedItem().toString()+"'
");

 while(rs.next()){
 band=1;
 Object[] t=new Object[6];
 for(int i=0;i<t.length;i++){
```

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

```
 t[i]=rs.getObject(i+1);
 JT_NActividad.setText(t[2]+"");
 JT_NExposicion.setText(t[3]+"");
 JT_NLeccion.setText(t[4]+"");
 JT_NExamen.setText(t[5]+"");
 }

JT_Promedio.setText(((Double.parseDouble(JT_NActivida
d.getText())+Double.parseDouble(JT_NExposicion.getTex
t())+Double.parseDouble(JT_NLeccion.getText())+Double
.parseDouble(JT_NExamen.getText()))/4)+"");
}jButton2.setEnabled(true);
  estatuto.close();
 conexion.close();
 }catch(Exception e)
{System.out.println(e.getMessage());}
    }else{
        JFrame jFrame = new JFrame();
        JOptionPane. showMessageDialog(jFrame,"Limpie
el contenido antes de realizar otra consulta");
 }
    }
//Limpiar
private void jButton2ActionPerformed
(java.awt.event.ActionEvent evt) {
    JT_NActividad.setText("");
    JT_NExposicion.setText("");
    JT_NLeccion.setText("");
    JT_NExamen.setText("");
    JT_Promedio.setText("");
    jButton2.setEnabled(false);
    }
}
//Cerrar sesion
private void jButton3ActionPerformed
(java.awt.event.ActionEvent evt) {
    this.setVisible(false);
    Inicio ini = new Inicio();
    ini.setVisible(true);
    }
```

Operaciones es Administrar usuarios que proviene de Directivos. Este formulario tiene once textFiel, y un comboBox, además posee 4 botones para disímiles funciones.

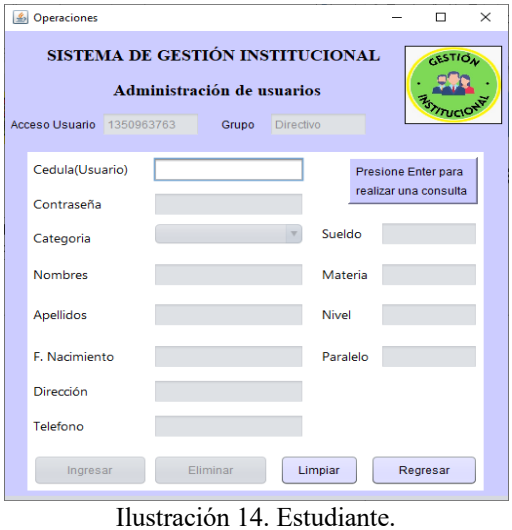

Ilustración 14. Estudiante.
Fuente: Autor

Se realiza un método con evento KeyPressed a cédula, pues si se da enter se llevan a cabo la consulta de los datos, hay que destacar que el campo sueldo no puede ser mostrado si la consulta está direccionada por un estudiante, por eso se realizó un if que compara la categoría y de acuerdo a eso se muestran los campos pertenecientes a cada tipo de usuario. En el caso que no se haya encontrado el usuario, se inicia el ingreso de un nuevo registro. Además, se creó una validación para que no se pueda eliminar el usuario directivo.
Luego se Ingresa la información y se guarda sueldo si no son estudiantes. El botón 2 Elimina, inicialmente consulta si los datos existen, después los borra, dejando los campos en blanco y también los desactiva.
Regresar tiene la función de redirigirse al interfaz directivo. Limpiar deja a cada campo en blanco.

```
//Evento KeyPressed
private void JT_CedulaKeyPressed
(java.awt.event.KeyEvent evt) {
//Verifica, no se encuentra ingresada una cedula
 if (VCed.equals("")){
 String categoria=null;
 //Verifica la dimension de la cedula
 char[] aux=JT_Cedula.getText().toCharArray();
 if(evt.getKeyCode()==10){
 if(aux.length==10){
 VCed = JT_Cedula.getText();
 try{
 Class.forName(driver);
 }catch(ClassNotFoundException e){
 System.out.println(e.getMessage());
 }
 try{
 //Realiza la conexion y ejecuta la consulta sql
 //System.out.println(url);
 Connection conexion=DriverManager.getConnection(url,
username, password);
 Statement estatuto=conexion.createStatement();
 ResultSet rss=estatuto.executeQuery("select * from
usuarios where usuario='"+JT_Cedula.getText()+"'");

 while(rss.next()){
 band=1;
 Object[] t=new Object[3];
 for(int i=0;i<t.length;i++){
 t[i]=rss.getObject(i+1);
 categoria = t[2]+"";}
 jComboBox1.addItem(categoria);
 }
//Realiza la consulta a la tabla persona
 ResultSet rs=estatuto.executeQuery("select * from
persona where cedula='"+JT_Cedula.getText()+"'");
 while(rs.next()){
 band=1;
 Object[] t=new Object[6];
 for(int i=0;i<t.length;i++)
 t[i]=rs.getObject(i+1);
 JT_Nombres.setText(t[1]+"");
 JT_Apellidos.setText(t[2]+"");
 JT_Nacimiento.setText(t[3]+"");
 JT_Direccion.setText(t[4]+"");
```

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

```
 JT_Telefono.setText(t[5]+"");
 }
 if(band==1){
 //Consulta sueldo si no son estudiantes
if(jComboBox1.getSelectedItem().toString().equals("Di
rectivo")){
 ResultSet rs1=estatuto.executeQuery("select * from
directivo where cedula='"+JT_Cedula.getText()+"'");
 while(rs1.next()){
 band=1;
 Object[] t=new Object[2];
 for(int i=0;i<t.length;i++)
 t[i]=rs1.getObject(i+1);
 JT_Sueldo.setText(t[1]+"");
 }
 }else
if(jComboBox1.getSelectedItem().toString().equals("Au
xiliar de servicio")){
 ResultSet rs2=estatuto.executeQuery("select * from
auxiliar_servicio where
cedula='"+JT_Cedula.getText()+"'");
 while(rs2.next()){
 band=1;
 Object[] t=new Object[2];
 for(int i=0;i<t.length;i++)
 t[i]=rs2.getObject(i+1);
 JT_Sueldo.setText(t[1]+"");
 }
 }else
if(jComboBox1.getSelectedItem().toString().equals("Pr
ofesor")){
 ResultSet rs3=estatuto.executeQuery("select * from
profesor where cedula='"+JT_Cedula.getText()+"'");
 while(rs3.next()){
 band=1;
 Object[] t=new Object[3];
 for(int i=0;i<t.length;i++)
 t[i]=rs3.getObject(i+1);
 JT_Sueldo.setText(t[1]+"");
 JT_Materia.setText(t[2]+"");
 }
 }else
if(jComboBox1.getSelectedItem().toString().equals("Es
tudiante")){
 ResultSet rs3=estatuto.executeQuery("select * from
estudiante where cedula='"+JT_Cedula.getText()+"'");
 while(rs3.next()){
 band=1;
 Object[] t=new Object[3];
 for(int i=0;i<t.length;i++)
 t[i]=rs3.getObject(i+1);
 JT_Nivel.setText(t[1]+"");
 JT_Paralelo.setText(t[2]+"");
 }
 }
 }
 //Si no existen registros registrar un nuevo usuario
 if(band==0){
 JFrame jFrame = new JFrame();
 JOptionPane. showMessageDialog(jFrame,"Registre el
nuevo usuario");
 JT_Contra.setEnabled(true);
 jComboBox1.addItem("Auxiliar de servicio");
 jComboBox1.addItem("Directivo");
 jComboBox1.addItem("Estudiante");
 jComboBox1.addItem("Profesor");
 jComboBox1.setEnabled(true);
 JT_Nombres.setEnabled(true);
 JT_Apellidos.setEnabled(true);
 JT_Nacimiento.setEnabled(true);
 JT_Direccion.setEnabled(true);
 JT_Telefono.setEnabled(true);
 JT_Sueldo.setEnabled(true);
 jButton1.setEnabled(true);
 //Validacion de usuario actual no eliminarse
 }else if
((Usuario_A.getText()).equals(JT_Cedula.getText()))
 {
 jButton2.setEnabled(false);
 }else {
 jButton2.setEnabled(true);
 JT_Sueldo.setEnabled(false);
 JT_Materia.setEnabled(false);
 JT_Nivel.setEnabled(false);
 JT_Paralelo.setEnabled(false);
 }
 //Se cierra la conexion
 estatuto.close();
 conexion.close();
 }catch(Exception
e){System.out.println(e.getMessage());}
 }else
 //Mensaje de error - cedula mal ingresada

javax.swing.JOptionPane.showMessageDialog(this,"¡Cédu
la Incorrecta!","Error",0);
 }
 //Se muestra un mensaje de error en caso de que no
se haya borrado la informacion anterior.
 }else {

javax.swing.JOptionPane.showMessageDialog(this,"¡Limp
ie el contenido antes de realizar una nueva
consulta!","Error",0);
 }
}
//Ingresar
private void jButton1ActionPerformed
(java.awt.event.ActionEvent evt) {
try{
 Class.forName(driver);
 }catch(ClassNotFoundException e){
 System.out.println(e.getMessage());
 }
 try{
 //Realiza la conexion a la BD y registra la
informacion
 Connection conexion=DriverManager.getConnection(url,
username, password);
 Statement estatuto=conexion.createStatement();
 String sql2="insert into usuarios
values('"+JT_Cedula.getText()+"','"+JT_Contra.getText
()+"','"+jComboBox1.getSelectedItem().toString()+"')"
;
 String sql="insert into persona
values('"+JT_Cedula.getText()+"','"+JT_Nombres.getTex
t()+"','"+JT_Apellidos.getText()+"','"+JT_Nacimiento.
getText()+"','"+JT_Direccion.getText()
+"','"+JT_Telefono.getText()+"')";

 //Guarda sueldo si no son estudiantes

if(jComboBox1.getSelectedItem().toString().equals("Di
rectivo")){
 String sql3="insert into directivo
values('"+JT_Cedula.getText()+"','"+Double.parseDoubl
e(JT_Sueldo.getText())+"')";
 estatuto.executeUpdate(sql3);
 } else
if(jComboBox1.getSelectedItem().toString().equals("Au
xiliar de servicio")){
 String sql4="insert into auxiliar_servicio
values('"+JT_Cedula.getText()+"','"+Double.parseDoubl
e(JT_Sueldo.getText())+"')";
 estatuto.executeUpdate(sql4);
```

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

```
 } else
if(jComboBox1.getSelectedItem().toString().equals("Pr
ofesor")){
 String sql5="insert into profesor
values('"+JT_Cedula.getText()+"','"+Double.parseDoubl
e(JT_Sueldo.getText())+"','"+JT_Materia.getText()+"')
";
 estatuto.executeUpdate(sql5);
 } else
if(jComboBox1.getSelectedItem().toString().equals("Es
tudiante")){
 String sql6="insert into estudiante
values('"+JT_Cedula.getText()+"','"+Integer.parseInt(
JT_Nivel.getText())+"','"+JT_Paralelo.getText()+"')";
 estatuto.executeUpdate(sql6);
 }
 //System.out.println(sql);
 estatuto.executeUpdate(sql2);
 estatuto.executeUpdate(sql);
 //Realiza una consulta para verificar que la
informacion esta ingresada
 ResultSet rs=estatuto.executeQuery("select * from
persona where cedula='"+JT_Cedula.getText()+"'");
 while(rs.next()){
 band=1;
 Object[] t=new Object[6];
 for(int i=0;i<t.length;i++)
 t[i]=rs.getObject(i+1);
 JT_Nombres.setText(t[1]+"");
 JT_Apellidos.setText(t[2]+"");
 JT_Nacimiento.setText(t[3]+"");
 JT_Direccion.setText(t[4]+"");
 JT_Telefono.setText(t[5]+"");
 }
 //si no ingresa la informacion - mensaje de error
 if(band==0){
javax.swing.JOptionPane.showMessageDialog(this,"¡Usua
rio no ingresado!","Error",0);
 //En caso de que se registro la informacion, se
muestra un mensaje indicandolo y se deshabilitan los
campos y el boton de ingresar
 }else{
 JFrame jFrame = new JFrame();
 JOptionPane. showMessageDialog(jFrame,"Usuario
registrado correctamente!");
 JT_Contra.setText("");
 JT_Contra.setEnabled(false);
 jComboBox1.setEnabled(false);
 JT_Nombres.setEnabled(false);
 JT_Apellidos.setEnabled(false);
 JT_Nacimiento.setEnabled(false);
 JT_Direccion.setEnabled(false);
 JT_Telefono.setEnabled(false);
 jButton1.setEnabled(false);
 JT_Sueldo.setEnabled(false);
 JT_Paralelo.setEnabled(false);
 JT_Materia.setEnabled(false);
 JT_Nivel.setEnabled(false);
 }
 //Se cierra la conexion
 estatuto.close();
 conexion.close();
 }catch(Exception e){
javax.swing.JOptionPane.showMessageDialog(this,"¡No
se pudo ingresar la informacion porque uno de los
campos se encuentra vacio!","Error",0);
 }
    }
//Eliminar
private void jButton2ActionPerformed
(java.awt.event.ActionEvent evt) {
 try{
 Class.forName(driver);
 }catch(ClassNotFoundException e){
 System.out.println(e.getMessage());
 }
 try{
 //Realiza la conexion a la BD y eliminar la
informacion
 Connection conexion=DriverManager.getConnection(url,
username, password);
 Statement estatuto=conexion.createStatement();
 String sql="delete from usuarios where
usuario='"+JT_Cedula.getText()+"'";
 String sql2="delete from persona where
cedula='"+JT_Cedula.getText()+"'";
 estatuto.executeUpdate(sql);
 estatuto.executeUpdate(sql2);
 //Consulta sueldo si no son estudiantes
if(jComboBox1.getSelectedItem().toString().equals("Di
rectivo")){
 String sql3="delete from directivo where
cedula='"+JT_Cedula.getText()+"'";
 estatuto.executeUpdate(sql3);
 }else
if(jComboBox1.getSelectedItem().toString().equals("Au
xiliar de servicio")){
 String sql4="delete from auxiliar_servicio where
cedula='"+JT_Cedula.getText()+"'";
 estatuto.executeUpdate(sql4);
 } else
if(jComboBox1.getSelectedItem().toString().equals("Pr
ofesor")){
 String sql5="delete from profesor where
cedula='"+JT_Cedula.getText()+"'";
 estatuto.executeUpdate(sql5);
 } else
if(jComboBox1.getSelectedItem().toString().equals("Es
tudiante")){
 String sql6="delete from estudiante where
cedula='"+JT_Cedula.getText()+"'";
 estatuto.executeUpdate(sql6);
 }
 //Realiza una consulta para verificar que la
informacion esta ingresada
 ResultSet rs=estatuto.executeQuery("select * from
persona where cedula='"+JT_Cedula.getText()+"'");
 while(rs.next()){
 band=1;
 Object[] t=new Object[6];
 for(int i=0;i<t.length;i++)
 t[i]=rs.getObject(i+1);
 JT_Nombres.setText(t[1]+"");
 JT_Apellidos.setText(t[2]+"");
 JT_Nacimiento.setText(t[3]+"");
 JT_Direccion.setText(t[4]+"");
 JT_Telefono.setText(t[5]+"");
 }

 //En caso de no ingresar la informacion se muestra
un mensaje de error y se pide intentar eliminar de
nuevo
 if(band==0){
 javax.swing.JOptionPane.showMessageDialog(this,"¡No
se pudo eliminar el usuario!","Error",0);

 //En caso de que se elimino correctamente la
informacion se muestra un mensaje y borra todos los
campos y deshabilita la eliminacion
 }else{
 JFrame jFrame = new JFrame();
 JOptionPane. showMessageDialog(jFrame,"Usuario
eliminado correctamente!");
        band =0;
        JT_Cedula.setText("");
        JT_Contra.setText("");
```

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

```
        JT_Contra.setEnabled(false);
        jComboBox1.removeAllItems();
        jComboBox1.setEnabled(false);
        JT_Nombres.setText("");
        JT_Nombres.setEnabled(false);
        JT_Apellidos.setText("");
        JT_Apellidos.setEnabled(false);
        JT_Nacimiento.setText("");
        JT_Nacimiento.setEnabled(false);
        JT_Direccion.setText("");
        JT_Direccion.setEnabled(false);
        JT_Telefono.setText("");
        JT_Telefono.setEnabled(false);
        JT_Sueldo.setText("");
        JT_Sueldo.setEnabled(false);
        JT_Materia.setText("");
        JT_Materia.setEnabled(false);
        JT_Nivel.setText("");
        JT_Nivel.setEnabled(false);
        JT_Paralelo.setText("");
        JT_Paralelo.setEnabled(false);
        jButton2.setEnabled(false);
        VCed = "";
 }
 //Se cierra la conexion
 estatuto.close();
 conexion.close();
 JT_Sueldo.setText("");
 }catch(Exception
e){System.out.println(e.getMessage());}
    }
//Limpiar
private void jButton4ActionPerformed
(java.awt.event.ActionEvent evt) {
     //Limpia todo el contenido del formulario
        band =0;
        JT_Cedula.setText("");
        JT_Contra.setText("");
        JT_Contra.setEnabled(false);
        jComboBox1.removeAllItems();
        jComboBox1.setEnabled(false);
        JT_Nombres.setText("");
        JT_Nombres.setEnabled(false);
        JT_Apellidos.setText("");
        JT_Apellidos.setEnabled(false);
        JT_Nacimiento.setText("");
        JT_Nacimiento.setEnabled(false);
        JT_Direccion.setText("");
        JT_Direccion.setEnabled(false);
        JT_Telefono.setText("");
        JT_Telefono.setEnabled(false);
        JT_Sueldo.setText("");
        JT_Sueldo.setEnabled(false);
        JT_Materia.setText("");
        JT_Materia.setEnabled(false);
        JT_Paralelo.setText("");
        JT_Paralelo.setEnabled(false);
        JT_Nivel.setText("");
        JT_Nivel.setEnabled(false);
        jButton1.setEnabled(false);
        jButton2.setEnabled(false);
        VCed = "";
}
//Regresar
private void jButton3ActionPerformed
(java.awt.event.ActionEvent evt) {
   //Cierra la ventana, regresa a la anterior
        this.setVisible(false);
        Directivo Dire = new Directivo();
        Dire.setVisible(true);
    }

private void jComboBox1ActionPerformed
(java.awt.event.ActionEvent evt) {
    if(VCed.equals(JT_Cedula.getText())){

if(jComboBox1.getSelectedItem().toString().equals("Es
tudiante")){
     JT_Sueldo.setEnabled(false);
     JT_Nivel.setEnabled(true);
     JT_Paralelo.setEnabled(true);
     JT_Materia.setEnabled(false);
 }else if (jComboBox1.isEnabled()== true){
if(jComboBox1.getSelectedItem().toString().equals("Pr
ofesor")){
     JT_Materia.setEnabled(true);
     JT_Sueldo.setEnabled(true);
     JT_Nivel.setEnabled(false);
    JT_Paralelo.setEnabled(false);
    }else
if(jComboBox1.getSelectedItem().toString().equals("Di
rectivo") ||
jComboBox1.getSelectedItem().toString().equals("Auxil
iar de servicio")){
    JT_Sueldo.setEnabled(true);
    JT_Nivel.setEnabled(false);
    JT_Paralelo.setEnabled(false);
    JT_Materia.setEnabled(false);
    }
 }
}
```

2.4.Pruebas

Para el proceso de pruebas se creó la BD “sgi” y luego se ejecutó el programa en Netbeans ingresando nuevos datos que se reflejan en la página de phpAdmin de Mysql.

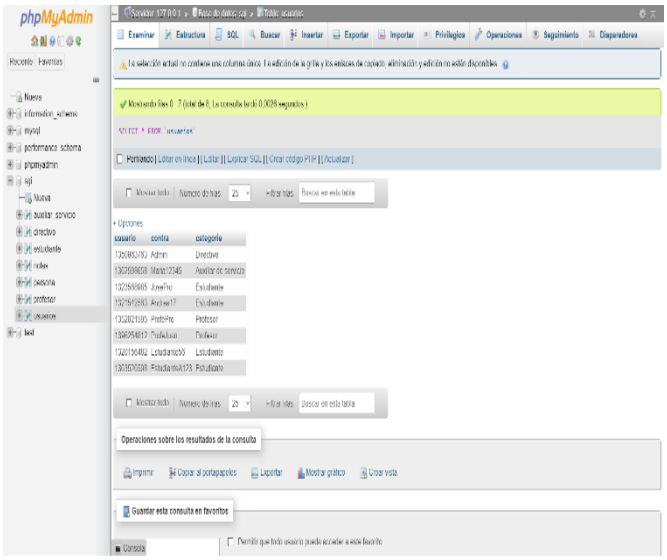

Ilustración 15. Base de datos – tabla usuarios.
Fuente: Autor

Al tener diferentes usuarios, los resultados obtenidos de las pruebas son desiguales pues depende de la categoría.

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

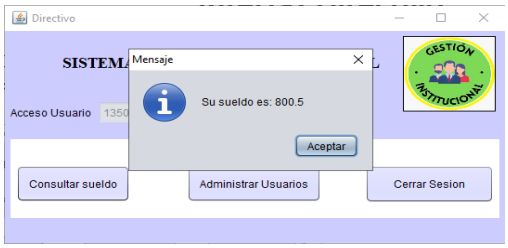

Ilustración 16. Sueldo Directivo.
Fuente: Autor

Prueba de ingreso de nuevo usuario perteneciente a Administrar usuario de Directivo.

Ilustración 17. Administrar usuario – ingreso de datos.
Fuente: Autor

Eliminar usuario desde Administrar usuario de Directivo.

Ilustración 18. Administrar usuario – eliminar dato.
Fuente: Autor

Notas ingresadas por Profesor.

Ilustración 19. Ingreso de nota.
Fuente: Autor

Consulta de notas – Estudiante.

Ilustración 20. Estudiante – Consulta nota.
Fuente: Autor

Luego de realizar varias pruebas, se corrigieron errores que se encontraban en varios procesos, pero finalmente se logró dar solución, dando paso a la fase de prueba donde todas interfaces fueron capaces de responder a ingresos, consultas y además se hizo uso de todas las opciones brindadas para corroborar el completo funcionamiento del programa.

3. RESULTADOS Y DISCUSIÓN

3.1.Identificación de contenido.

En Java la POO (Programación Orientada a Objetos) permite crear múltiples objetos o clases, las cuales pueden utilizarse para manipular la información en forma de campos y el código en forma de métodos. Java maneja un entorno de desarrollo integrado (IDE) llamado NetBeans, en el

Revista Nexos Científicos
Julio-Diciembre 2022 pp. 18-31
Volumen 1, Número 2
Fecha de recepción: septiembre 2022
ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

cual se hizo la creación de código e interfaces gráficas (GUI) para la interacción directa del administrador y el programa. Los objetos en la programación son entidades o instancias dentro de la clase que definimos para la solución de un problema real, contiene propiedades (métodos y atributos).

MySQL es un sistema de gestión de base de datos de código abierto basado en un lenguaje de consulta, el cual permite el uso correcto y manejo de la información que sea registrada con anterioridad. El acceso a esta información permite la manipulación y consulta de los datos.

Con ayuda de GUI (interfaz de usuario gráfica), los usuarios tienen la ventaja de manipular con seguridad la aplicación y esta medida es muy sencilla y simple de entender. Además, cuenta con una interfaz agradable a la vista del usuario.

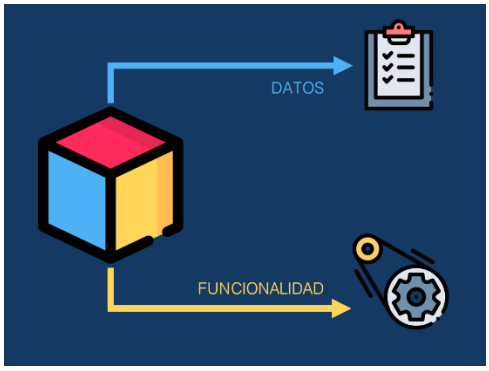

Ilustración 21. Funcionalidad POO.
Fuente: (CMS-ITTUX, 2022)

XAMPP es una herramienta la cual permite probar un desarrollo web basado en PHP desde el propio ordenador sin la necesidad de contar con acceso a internet, también cuenta con la configuración ya establecida del servidor desde que se instala la herramienta, mostrando la información en el navegador.

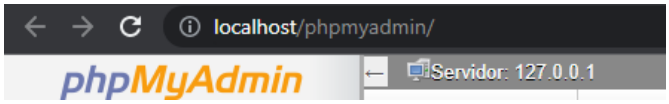

Ilustración 22. Servidor XAMPP.
Fuente: (Autor)

3.2.Proceso y Búsqueda.

Dentro de este proceso se identificó la estructura de la información mediante el uso de las clases y objetos, las cuales son definidas a partir de las características (métodos y atributos) para manejar la información de manera correcta.

Esta medida permite que la información para cada miembro de la institución se almacene de manera estructurada, haciendo uso de los métodos de ingreso, consulta y asignación de calificaciones para los estudiantes.

La consulta de información está incluida dentro de la aplicación, una vez creada la BD y ejecutado el scrip que contiene los datos de creación de tablas y atributos a través del servidor XAMPP. De esta manera el sistema de gestión institucional no generara errores durante su ejecución.

3.3.Resultados.

Del análisis obtenido durante la ejecución del programa obtuvimos resultados deseados al momento de realizar las consultas, la información de cada usuario se muestra de acuerdo a sus privilegios y las validaciones mantienen la integridad de los datos.

En la ejecución del programa podemos encontrar como primera instancia una ventana de login, para ello cada usuario (estudiantes, autoridades, directivos, etc) del sistema cuentan con sus credenciales de ingreso definidos con anterioridad, permitiendo gestionar los datos según el cargo que ocupa dentro de la institución.

Cada cargo cuenta con características diferentes a los anteriores, sin embargo, mantienen opción de consulta y cada una de ellas se ha realizado de manera efectiva, es decir no existe ningún error al momento de realizar una consulta.

La interfaz gráfica está realizada con tonos claros y cálidos (colores pastel), para así dar una vista agradable y una buena comunicación entre los administradores y el programa.

El scrip para la base de datos funciona sin inconvenientes una vez creada la base de datos, el manejo de los datos es de manera sencilla e inmediata y los servicios de XAMPP funcionan correctamente.

ISSN: 2773-7489
Correo: editor@istvidanueva.edu.ec
URL: http://nexoscientificos.vidanueva.edu.ec/index.php/ojs/index
Fecha de aceptación: noviembre 2022

4. CONCLUSIONES

Luego de finalizar y hacer las pruebas de funcionamiento, podemos concluir que:

- El programa está diseñado para la gestión de usuarios dentro de cualquier institución educativa, de esta manera obtener acceso al contenido de manera sencilla, rápida y sobre todo evitando pérdida de tiempo al momento de consultar la información.
- A medida que se desarrollaba la aplicación surgieron muchas ideas diferentes a la original, pero luego de varias conversaciones con el equipo de trabajo se pudo definir las funciones a implementar y como estas mejoraban el código fuente.
- Existieron algunos problemas durante la ejecución del programa, tuvimos que implementar nuevas medidas para mejorar el rendimiento del mismo.
- El proceso de consulta cuenta con una gran cantidad de opciones, ya que, como está conectada a una base de datos se pueden hacer múltiples consultas. Es evidente que el programa cuenta con ciertas características para el manejo y la recolección de datos.
- Se obtuvieron grandes conocimientos a medida que se desarrollaba el programa, sobre el funcionamiento de una base de datos y la creación de software, también se logró comprender conceptos avanzados sobre el uso correcto del lenguaje de programación (Java).

REFERENCIAS

Annevk, F. a. (29 de 06 de 2020). *Whatwg*. Obtenido de Whatwg: https://dom.spec.whatwg.org

CMS-ITTUX. (2022). POO. 879-882.

Microsoft. (01 de 01 de 2020). *https://docs.microsoft.com/*. Obtenido de https://docs.microsoft.com/: https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.datavisualization.charting.chart?view=netframework-4.8

Mozilla. (01 de 01 de 2020). *https://developer.mozilla.org/*. Obtenido de https://developer.mozilla.org/: https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview