

Universidad de San Carlos de Guatemala

Centro Universitario de Occidente

División de Ciencias de la Ingeniería

Área Profesional

Introducción a la Programación y Computación 2

Sección "A"

Ing. José Moisés Granados Guevara



“MANUAL TECNICO PROYECTO 1”

Melvin Eduardo Ordoñez Sapón RA | 202230552

Quetzaltenango, septiembre del 2024

“Id y enseñad a todos”

ACERCA DEL PROGRAMA:

IDE utilizado: Apache NetBeans IDE 22

JDK utilizado: Versión 21.0.4

Programado en Sistema Operativo: Ubuntu 24.04 LTS

Terminal Utilizada: Terminal de Linux

Lenguaje de Programación Utilizado: Java - JSP

Base de datos utilizado: MySQL 24.04.1

Compatibilidad: Windows, Linux, macOS

METODOS IMPORTANTES.

Con el siguiente metodo nos conectamos a la base de datos

```
public static Connection conectar() throws SQLException {
    try {
        // Registro del driver explícitamente
        Class.forName("com.mysql.cj.jdbc.Driver");
        connection = DriverManager.getConnection(URL_MYSQL, USER, PASSWORD);
        //System.out.println("Conexión a la base de datos exitosa");
        return connection;
    } catch (ClassNotFoundException e) {
        throw new SQLException("Driver no encontrado", e);
    } catch (SQLException ex) {
        ex.printStackTrace();
        throw new SQLException("Error al conectar con la base de datos", ex);
    }
}
```

El siguiente metodo se utiliza para registrar a un nuevo usuario a la base de datos

```
public boolean registrarUsuario(Usuario usuario) {
    if (existeUsuario(usuario.getNombreUsuario())) {
        return false;
        // Significa que el usuario ya existe en la base de datos
    }

    String consulta = "INSERT INTO usuarios (nombre_usuario, contraseña, perfil, rol) VALUES";
    (?, ?, ?, ?);

    try {
        PreparedStatement statement = connection.prepareStatement(consulta);
        statement.setString(1, usuario.getNombreUsuario());
        statement.setString(2, usuario.getContrasena());
        statement.setString(3, usuario.getTexto()); // Se usa String aquí
        statement.setString(4, usuario.getRol().toString());

        // Si la inserción fue exitosa, executeUpdate devuelve el número de filas afectadas
        int filasAfectadas = statement.executeUpdate();

        // Si se afectó al menos una fila, devuelve true, significa que la inserción fue exitosa
        return filasAfectadas > 0;
    }
```

```

    } catch (SQLException e) {
        // Si ocurre una excepción, puedes manejarla y devolver false
        System.out.println("Error al registrar usuario: " + e.getMessage());
        return false;
    }
}

```

este metodo se encarga de iniciar sesion con un usuario existente en la base de datos

```

public Usuario iniciarSesion(String nombreUsuario, String contrasena) {
    Seguridad seguridad = new Seguridad();
    Usuario usuarioObtenido = obtenerUsuario(nombreUsuario);

    if (usuarioObtenido == null) {
        return null;
    }
    // Si la contraseña coincide con el hash almacenado, devuelve un objeto Usuario
    if (seguridad.verificarContrasena(contrasena, usuarioObtenido.getContrasena())) {
        System.out.println("Contraseña correcta");
        return usuarioObtenido;
    }

    // Si no se encontró un usuario con las credenciales dadas, devuelve null
    return null;
}

```

para actualizar los datos de algun usuario del sistema se ejecuta el siguiente metodo

```

public void actualizarUsuario(String nombreUsuario, String texto, String fotoPerfilPath) {
    String consulta = "UPDATE usuarios SET perfil = ?, foto_perfil = ? WHERE nombre_usuario = ?";

    try (PreparedStatement stmt = connection.prepareStatement(consulta)) {
        stmt.setString(1, texto);
        stmt.setString(2, fotoPerfilPath);
        stmt.setString(3, nombreUsuario);
        stmt.executeUpdate();
    } catch (SQLException e) {

```

```

        e.printStackTrace();
    }
}

```

para crear una revista se utiliza el siguiente metodo que agrega toda la informacion de una revista a la base de datos

```

public boolean crearRevista(Revista revista) {
    if (existeRevista(revista.getNombre())) {
        return false;
        // Significa que la revista ya existe en la base de datos
    }

    String consulta = "INSERT INTO revistas (nombre_revista, descripcion, categoria,
    fecha_creacion, id_autor, costo, url_pdf) VALUES (?, ?, ?, ?, ?, ?, ?)";

    try {

        PreparedStatement statement = connection.prepareStatement(consulta);
        statement.setString(1, revista.getNombre());
        statement.setString(2, revista.getDescripcion());
        statement.setString(3, revista.getCategoria());

        // Convertir la fecha de String a java.sql.Date
        java.sql.Date sqlDate = java.sql.Date.valueOf(revista.getFechaCreacion()); // Asegúrate que
        revista.getFechaCreacion() esté en formato "YYYY-MM-DD"

        statement.setDate(4, sqlDate);
        statement.setString(5, revista.getAutor());
        statement.setDouble(6, revista.getCosto());
        statement.setString(7, revista.getArchivoPDF());
        int filasAfectadas = statement.executeUpdate();

        return filasAfectadas > 0;
    } catch (SQLException e) {
        // Si ocurre una excepción, puedes manejarla y devolver false
        System.out.println("Error al crear revista: " + e.getMessage());
        return false;
    }
}

```

este metodo es el encargado de suscribir a un usuario de tipo suscriptor a una revista
seleccionada por el cliente

```
// Método para suscribirse a una revista
public boolean suscribirUsuarioAREvista(String nombreUsuario, String nombreRevista, String
fechaSuscripcion) {
    String consulta = "INSERT INTO suscripciones (nombre_usuario, nombre_revista,
fecha_suscripcion) VALUES (?, ?, ?)";

    try (PreparedStatement statement = connection.prepareStatement(consulta)) {
        statement.setString(1, nombreUsuario);
        statement.setString(2, nombreRevista);
        statement.setString(3, fechaSuscripcion);

        int filasAfectadas = statement.executeUpdate();
        return filasAfectadas > 0;
    } catch (SQLException e) {
        System.out.println("Error al suscribirse a la revista: " + e.getMessage());
        return false;
    }
}
```

este metodo es uno de los principales, ya que dependiendo del rol del usuario que inicio
sesión este lo redirige al portal correspondiente

```
private void redirigirInicio(HttpServletRequest request, HttpServletResponse response,
Usuario usuario) throws ServletException, IOException {
    Rol rol = usuario.getRol();
    System.out.println("Rol del usuario: " + rol.name());

    switch (rol) {
        case SUScriptor:
            response.sendRedirect("homeSuscriptor");
            //response.sendRedirect("inicio/suscriptor/suscriptorHome.jsp");
            break;
        case EDITor:
            //response.sendRedirect("inicio/editor/editorHome.jsp");
            response.sendRedirect("homeEditor");
            break;
    }
}
```

```

case ANUNCIANTE:
    response.sendRedirect("inicio/anunciante/anuncianteHome.jsp");
    break;
case ADMINISTRADOR:
    response.sendRedirect("inicio/administrador/administradorHome.jsp");
    break;
default:
    System.out.println("Rol desconocido: " + rol);
    //response.sendRedirect("portal/login.jsp?error=Rol desconocido");
    break;
}
}

```

Es utilizado por el servlet que es el encargado de iniciar la sesion a un usuario existente

```

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
    String nombreUsuario = request.getParameter("nombre_usuario");
    String contrasena = request.getParameter("contrasena");

    UsuarioDB usuarioDB = new UsuarioDB();
    Usuario usuario = usuarioDB.iniciarSesion(nombreUsuario, contrasena);

    if (usuario != null) {
        // Guardar el usuario en la sesión
        HttpSession session = request.getSession();
        session.setAttribute("usuario", usuario);
        // Redirigir a la pagina de inicio correspondiente
        redirigirInicio(request, response, usuario);
    } else {
        // Si el login falla, pasar el mensaje de error al JSP y reenviar la solicitud
        request.setAttribute("errorMensaje", "Credenciales incorrectas, por favor intenta
nuevamente.");
        RequestDispatcher dispatcher = request.getRequestDispatcher("portal/login.jsp");
        dispatcher.forward(request, response);
    }
}

```