DISEÑO DETALLADO H1

Gestión de documentación de servicio social.

**Historial de Versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versión** | **Fecha** | **Detalle** | **Autor** |
| 01 | 19/11/18 | Diseño general de las clases de conexión. | JAGG |

**Contenido**

[**Modelo de Clases** 2](#_Toc530727216)

[Diagrama de Clases. 2](#_Toc530727217)

[**Modelo de presentación.** 3](#_Toc530727218)

[Diagrama de presentación. 3](#_Toc530727219)

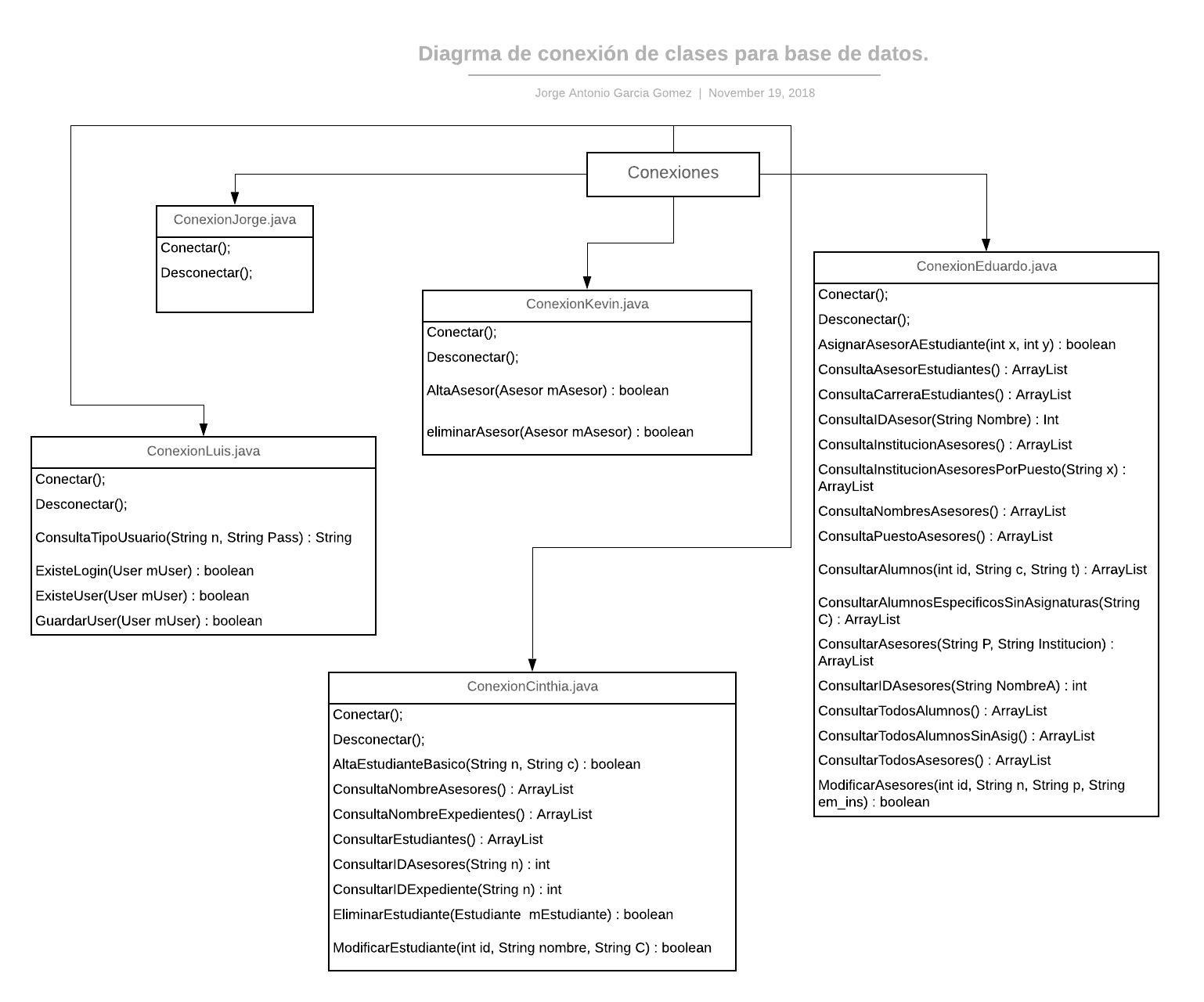
[Diagrama de procesos 3](#_Toc530727220)

[Especificación de métodos ¿qué hacen?: 3](#_Toc530727221)

# **Modelo de Clases**

## Diagrama de Clases.

Paquete conexiones



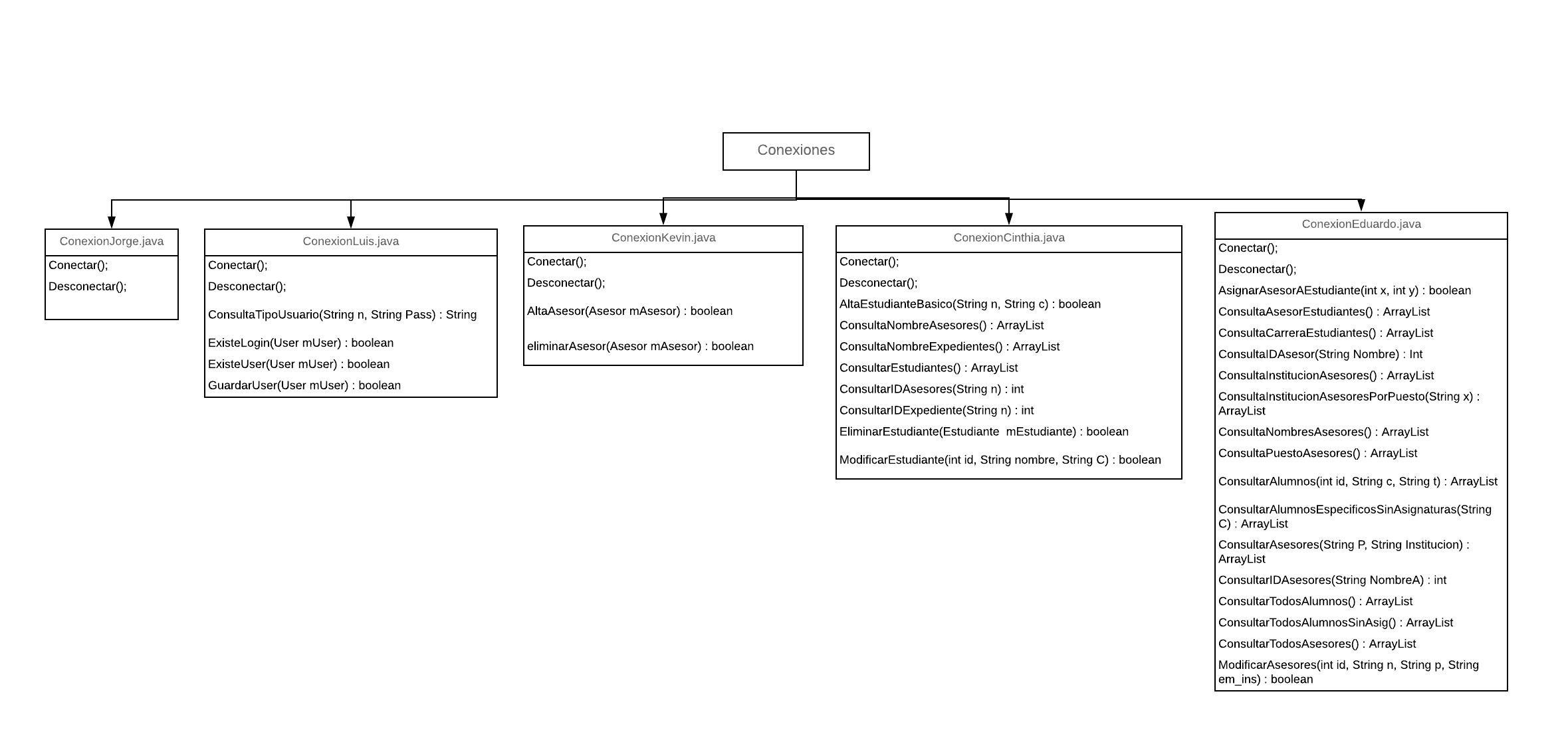
# **Modelo de presentación.**

## Diagrama de presentación.

En este caso, al ser solo las clases que permiten la conexión entre el entorno gráfico con la base de datos **no existe un diagrama de presentación** debido a que estas clases están implementadas a nivel de código y la única forma de comprobación que existe de que estas funcionan es que las clases conecten con la base de datos recibiendo un valor booleano (true) como confirmación.

**Modelo de procesos.**

## Diagrama de procesos



## Especificación de métodos ¿qué hacen?:

1. **Clase ConexionJorge.java**
   1. **Método Conectar():** Este método se utiliza para conectar a la base de datos, establece una conexión para que el usuario pueda ejecutar sus instrucciones SQL.
   2. **Método Desconectar():** Este método es utilizado para cerrar conexiones que puede haber cuando el usuario utiliza el método Conectar().
2. **Clase ConexionLuis.java**
   1. **Método Conectar():** Este método se utiliza para conectar a la base de datos, establece una conexión para que el usuario pueda ejecutar sus instrucciones SQL.
   2. **Método Desconectar():** Este método es utilizado para cerrar conexiones que puede haber cuando el usuario utiliza el método Conectar().
   3. **ConsultaTipoUsuario():** Este método se utiliza para saber el tipo de usuario que intenta iniciar sesión por medio del login.
   4. **ExisteLogin():** Este método se utiliza para saber si el usuario que se intenta iniciar sesión en el login, realmente está registrado en la base de datos.
   5. **ExisteUser():** Este método verifica si el usuario que se está ingresando ya existe.
   6. **GuardarUser():** Este método guarda al usuario en caso de que este sea registrado.
3. **Clase ConexionKevin.java**:
   1. **Método Conectar():** Este método se utiliza para conectar a la base de datos, establece una conexión para que el usuario pueda ejecutar sus instrucciones SQL.
   2. **Método Desconectar():** Este método es utilizado para cerrar conexiones que puede haber cuando el usuario utiliza el método Conectar().
   3. **Método AltaAsesor():** Este método se utiliza para dar de alta un asesor en la base de datos.
   4. **Método eliminarAsesor():** Este método se utiliza para dar de baja un asesor que fue dado de alta en la base de datos.
4. **Clase ConexionCinthia.java:**
   1. **Método Conectar():** Este método se utiliza para conectar a la base de datos, establece una conexión para que el usuario pueda ejecutar sus instrucciones SQL.
   2. **Método Desconectar():** Este método es utilizado para cerrar conexiones que puede haber cuando el usuario utiliza el método Conectar().
   3. **Método AltaEstudianteBasico():** Este método se utiliza para dar de alta a un estudiante en la base de datos.
   4. **Método ConsultaNombreAsesores():** Este método se utiliza para consultar el nombre del asesor.
   5. **Método ConsultaNombreExpediente():** Este método se utiliza para consultar los nombres de los expedientes registrados en la base de datos.
   6. **Método ConsultarEstudiantes():** Este método se utiliza para consultar a los estudiantes y todos sus campos.
   7. **Método ConsultarIDAsesores():** Este método consulta al asesor completo registrado en la base de datos.
   8. **Método ConsultarIDExpediente():** Este método consulta el expediente completo registrado en la base de datos.
   9. **Método EliminarEstudiante():** Este método elimina a un estudiante completo de la base de datos.
   10. **Método ModificarEstudiante():** Este método modifica varios campos de un estudiante en caso de que fueron ingresados erroneamentes.
5. **Clase ConexionEduardo.java:** 
   1. **Método Conectar():** Este método se utiliza para conectar a la base de datos, establece una conexión para que el usuario pueda ejecutar sus instrucciones SQL.
   2. **Método Desconectar():** Este método es utilizado para cerrar conexiones que puede haber cuando el usuario utiliza el método Conectar().
   3. **Método AsignarAsesorAEstudiante():** Este método asgina un asesor a un estudiante dados de alta en la base de datos.
   4. **Método ConsultaAsesorEstudiante():** Este método hace una consulta de los asesores que están asignados a un estudiante.
   5. **Método ConsultaCarreraEstudiante():** Este método hace una consulta de la carrera de los estudiantes.
   6. **Método ConsultaIDAsesor():** Este método consulta a todo un asesor que fue registrado en la base de datos.
   7. **Método ConsultaInstitucionAsesores():** Este método realiza una consulta de las instituciones a las que pertenecen los asesores.
   8. **Método ConsultaInstitucionAsesoresPorPuesto():** Este método realizar una consulta de las instituciones a los que pertenecen los asesores por su puesto asginado.
   9. **Método ConsultaNombreAsesores():** Este método realiza una consulta de los nombres de los asesores.
   10. **Método ConsultaPuestoAsesores():** Este método consulta solo el puesto de los asesores.
   11. **Método ConsultaAlumnos():** Este método consulta a un alumno con sus campos determinados.
   12. **Método ConsultaAlumnosEspecificosSinAsignaturas():** Este método consulta a los alumnos sin asignaturas asignadas en la base de datos.
   13. **Método ConsutaAsesores():** Este método hace una consulta de un asesor determinado para mostrar sus campos respectivos.
   14. **Método ConsultarIDAsesores():** Este método hace una consulta de un asesor determinado por medio de su ID.
   15. **Método ConsultarTodosAlumnos():** Este método hace una consulta de todos los alumnos registrados en la base de datos.
   16. **Método ConsultarTodosAlumnosSinAsig():** Este método hace una consulta de todos los alumnos registrados en la base de datos sin las asignaturas.
   17. **Método ConsultarTodosAsesores():** Este método hace una consulta de todos los asesores registrados en la base de datos.
   18. **Método ModificarAsesores():** Este método hace una modificación de un determinado asesor para corregir un determinado campo.