|  |
| --- |
| MANUAL DEL SISTEMA |
| CBI (Aplicación para control de servicio) |
|  |
| **Andrés Felipe Rodríguez Velilla** |
| **David Enrique Padilla Fonseca**  **Marialejandra Torrenegra Anaya** |

**TABLA DE CONTENIDO**

[**1 MODELO DE NEGOCIO** 3](#_Toc498639567)

[1.1 Casos de uso de mundo real 3](#_Toc498639568)

[1.1.1 Diagrama de casos de uso de mundo real 3](#_Toc498639569)

[1.1.2 Descripción detallada de cada caso de uso 5](#_Toc498639570)

[1.2 Modelo de dominio 7](#_Toc498639571)

[1.3 Proceso de negocio 8](#_Toc498639572)

[1.3.1 Diagrama de actividades 8](#_Toc498639573)

[1.4 Glosario 13](#_Toc498639574)

[**2 REQUISITOS** 14](#_Toc498639575)

[2.1 Visión 14](#_Toc498639576)

[2.2 Especificación de requisitos 15](#_Toc498639577)

[2.3 Requisitos priorizados 16](#_Toc498639578)

[2.4 Diagrama de casos de uso de requisitos 17](#_Toc498639579)

[**3 MODELO DE DISEÑO** 18](#_Toc498639580)

[3.1 Vista lógica 18](#_Toc498639581)

[3.1.1 Diagrama de componentes 18](#_Toc498639582)

[3.1.2 Diagrama de clases 19](#_Toc498639583)

[3.2 Vista de escenarios 27](#_Toc498639584)

[3.2.1 Diagrama general de casos de uso 28](#_Toc498639585)

[3.2.2 Diseño interfaz gráfica de usuario 30](#_Toc498639586)

[3.3 Vista de procesos 34](#_Toc498639587)

[3.3.1 Diagrama de secuencia de los casos de uso más representativos 34](#_Toc498639588)

[**4 MODELO DE IMPLEMENTACIÓN** 36](#_Toc498639589)

[4.1 Vista de desarrollo 36](#_Toc498639590)

[4.1.1 Diagrama de paquetes 36](#_Toc498639591)

[4.2 Vista física 37](#_Toc498639592)

[4.2.1 Diagrama de despliegue 37](#_Toc498639593)

[**5 BIBLIOGRAFÍA** 39](#_Toc498639594)

# **MODELO DE NEGOCIO**

Para la realización de este proyecto y tener una clara descripción del problema a solucionar se utilizaron técnicas de recolección de información, tal como la entrevista.

Para la entrevista se interrogo al encargado de la iglesia, en este caso el pastor, con el fin de conocer como es el proceso administrativo y de control en la iglesia, y si este estaría dispuesto a mejorarlo. Las preguntas de la entrevista se encuentran en anexos (*Ver entrevista anexada en el CD*).

La recolección anteriormente mencionada permitió conocer más a fondo el proceso de registro y control de la iglesia y la perspectiva que el pastor tiene de este.

## **Casos de uso de mundo real**

### **Diagrama de casos de uso de mundo real**

A continuación, se presenta el diagrama de casos de uso del mundo real, el cual refleja la interacción del actor(es) involucrados en el contexto del problema con el contexto del problema, en este caso los actores son dos, el pastor y el congregante, y en el diagrama se observa cómo estos se relacionan con el contexto al que pertenecen.

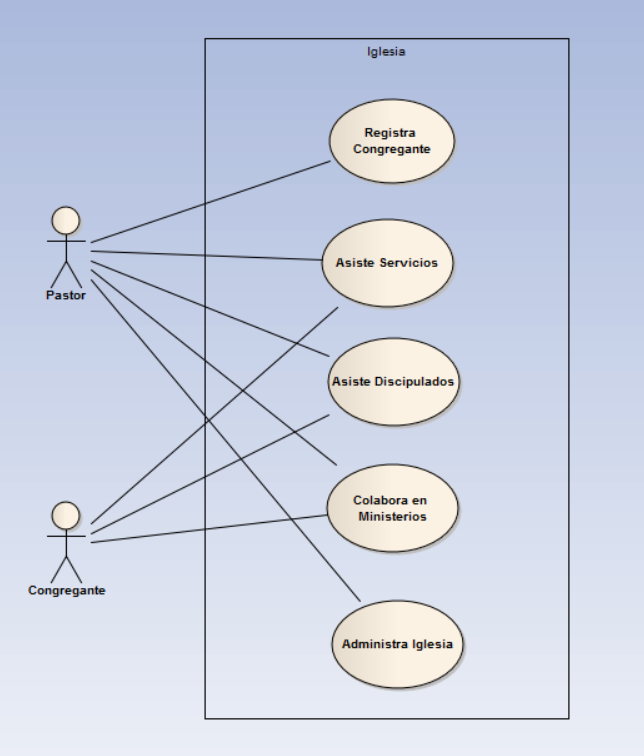


Figura N°1. Diagrama de casos de uso de mundo real.

En el diagrama anterior, se observa la relación que tiene el pastor y el congregante con cada uno de los casos de uso pertenecientes al contexto real del problema, en este caso la iglesia. El pastor es un actor involucrado en cuatro (5) casos, los cuales son, ‘registra congregante’, ‘asiste servicios’, ‘asiste discipulado’, ‘colabora en ministerios’ y ‘administra iglesia’. por su parte el congregante participa e interviene en tres (3) casos, que son, ‘asiste servicios’, ‘asiste discipulados’ y ‘colabora en ministerios’.

### **Descripción detallada de cada caso de uso**

|  |
| --- |
| Caso de uso 1: Registra Congregante |
| Actor principal: Congregante |
| Personal involucrado e intereses:  Congregante: Espera realizar registro y obtener o no sobre de diezmo para ingresar a la iglesia |
| Precondiciones: Ninguno |
| Postcondiciones: Llegar a la entrada de la iglesia |
| Flujo principal:  1. Congregante se ubica en el final de la fila  2. Congregante avanza cada vez que el Congregante de adelante avanza  3. Congregante llega a la entrada de la iglesia |
| Flujo alternativo: No hay |
| Frecuencia: Cada vez que un Congregante desee entrar en la iglesia |
| Prioridad: Primero en implementarse al encontrar la fila de la iglesia |
| Uso de tecnologías: Ninguna |
| Preguntas abiertas: Ninguna por el momento |

|  |
| --- |
| Caso de uso 2: Asiste Servicio |
| Actor principal: Congregante |
| Personal involucrado e intereses:  Congregante: Asiste a la iglesia para participar en el servicio de esta.  Pastor: Dirige el servicio de la iglesia. |
| Precondiciones: Haber realizado la fila para el registro de la iglesia |
| Postcondiciones: Ingreso a la iglesia |
| Flujo principal:   1. Congregante realiza registro 2. Pastor permite acceso de congregante 3. Congregante ingresa a la iglesia 4. Congregante asiste a servicio |
| Flujo alternativo:  1a. Si el congregante no realiza el registro de manera correcta, este no puede ingresar a la iglesia. |
| Frecuencia: Cada vez que un Congregante desee asistir a un servicio |
| Prioridad: primero en implementar al llegar a la iglesia |
| Uso de tecnologías: Ninguna |
| Preguntas abiertas: Ninguna por el momento |

|  |
| --- |
| Caso de uso 3: Asiste discipulado |
| Actor principal: Congregante |
| Personal involucrado e intereses:  Congregante: Cursa discipulado respectivo  Pastor: Que congregante este en el discipulado respectivo |
| Precondiciones: Estar registrado en la iglesia |
| Postcondiciones: Asiste a discipulado |
| Flujo principal:  1. Congregante ingresa a iglesia  2. Pastor indica los grupos de discipulado  3. Congregante se ubica en grupo de discipulado al que pertenece  4. Pastor dirige los discipulados   1. Congregante cursa discipulado |
| Flujo alternativo:  1a. Si es primera vez que el Congregante ingresa, Pastor lo ubica en el nivel 1 de discipulado |
| Frecuencia: Cada vez que un Congregante deba asistir a discipulado |
| Prioridad: Segundo en implementar al llegar a la iglesia |
| Uso de tecnologías: Ninguna |
| Preguntas abiertas: Ninguna por el momento |

|  |
| --- |
| Caso de uso 4: Colabora en ministerio |
| Actor principal: Congregante |
| Personal involucrado e intereses:  Congregante: Pertenece a un ministerio  Pastor: Lidera ministerio |
| Precondiciones: Estar registrado en la iglesia |
| Postcondiciones: Asiste a ministerio |
| Flujo principal:  1. Congregante ingresa a iglesia  2. Pastor ubica los ministerios  3. Congregante se ubica en el ministerio al que pertenece  4. Pastor lidera los ministerios |
| Flujo alternativo:  1a. Si es primera vez que el Congregante ingresa, Pastor lo ubica en el ministerio básico |
| Frecuencia: Cada vez que un Congregante deba asistir a ministerio |
| Prioridad: tercero en implementar al llegar a la iglesia |
| Uso de tecnologías: Ninguna |
| Preguntas abiertas: Ninguna por el momento |

|  |
| --- |
| Caso de uso 5: Administra iglesia |
| Actor principal: Pastor |
| Personal involucrado e intereses:  Pastor: Dirige, lidera y revisa todas las actividades de la iglesia y relacionadas a esta |
| Precondiciones: Ninguno |
| Postcondiciones: Ninguno |
| Flujo principal:  1. Pastor controla registro de Congregante  2. Pastor lidera ministerio  3. Pastor dicta discipulado  4. Pastor brinda servicio  5. Pastor maneja iglesia |
| Flujo alternativo: No hay |
| Frecuencia: Siempre |
| Prioridad: Siempre |
| Uso de tecnologías: Ninguna |
| Preguntas abiertas: Ninguna por el momento |

## **Modelo de dominio**

A continuación, se presenta el diagrama referente al modelo de dominio del contexto del problema en base a los requisitos del sistema. En este diagrama se observan cómo interactúan los distintos objetos involucrados en el problema y las relaciones que tienen entre ellos.

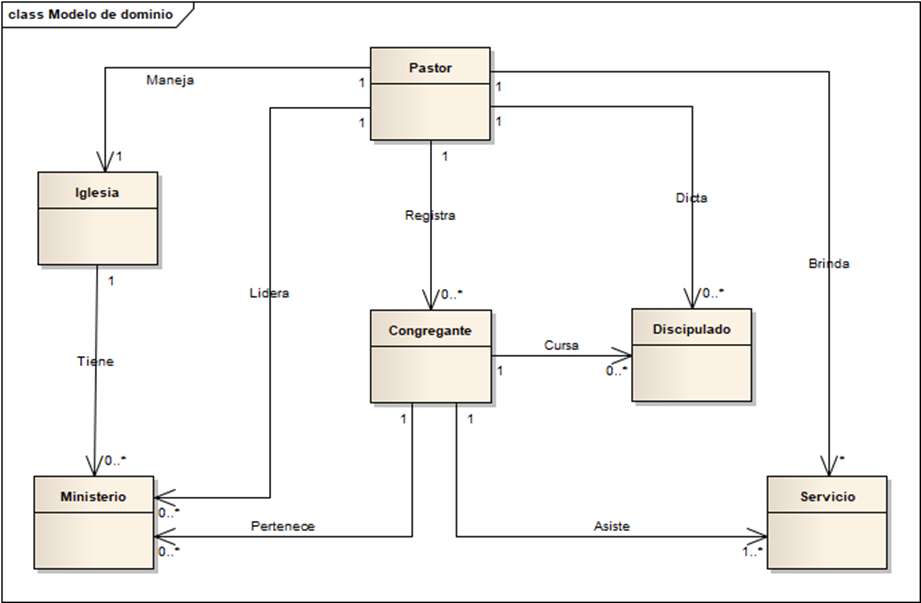


Figura N°2. Diagrama de modelo de dominio.

Para diseñar este modelo de dominio nos basamos en la información descrita en la descripción del problema y la técnica utilizada para recolectar información (entrevista) e identificamos los posibles objetos que actúan en el problema, de allí descartamos los que consideramos irrelevantes y procedimos a implementarlos en el diagrama UML, a continuación, establecimos las respectivas relaciones y su cardinalidad.

Ahora describiremos el modelo de dominio, comenzando desde la iglesia que es manejada por un pastor, la iglesia puede tener muchos ministerios quien a su vez los dirige el pastor. El pastor predica o brinda servicios a los cuales asisten los miembros o congregantes, éstos son registrados en el software por el pastor y así mismo los congregantes pueden cursar discipulados que también son dictados por el pastor.

## **Proceso de negocio**

### **Diagrama de actividades**

Los siguientes diagramas se elaboraron con base a la información obtenida por medio de nuestro entrevistado quien nos explicó de manera clara como era la lógica del proceso de registro y control, a partir de los casos de uso del mundo real.

Los diagramas presentados a continuación hacen referencia a los diagramas de actividades, el general y los referentes a los casos de uso más relevantes del contexto del problema.

#### **Diagrama de actividades general**

A continuación, se presenta el diagrama de actividades general del problema en base a los casos de uso del mismo; en él se muestra el flujo de las actividades que integran el problema.

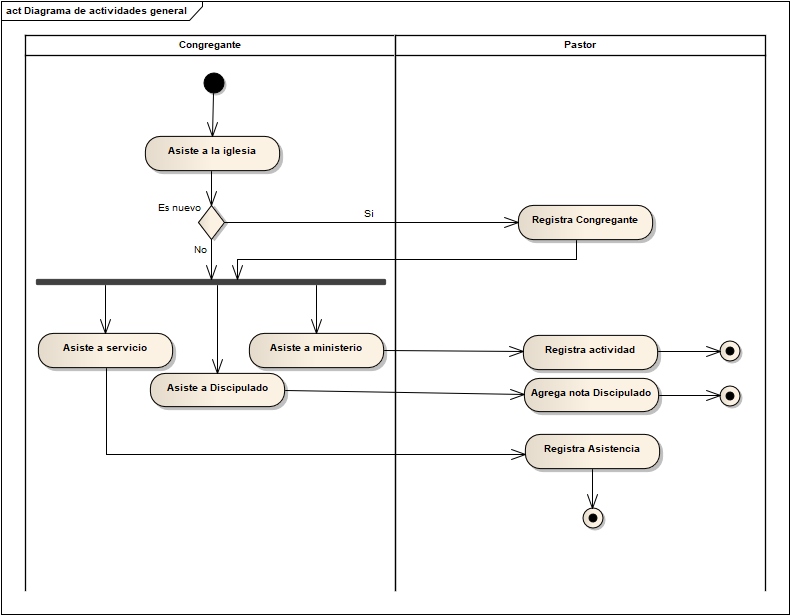


Figura N°3. Diagrama de actividades general.

En este diagrama de actividades se muestra de forma general la información acerca del funcionamiento de la gestión de registro y control de los congregantes de la iglesia, que funcionara como apoyo para el desarrollo de la solución.

Al iniciar, la persona (congregante) llega a la iglesia, si es una persona nueva debe proceder a registrarse, caso contrario, si esta persona es antigua. Seguido a esto, se realiza el debido control de la asistencia, así como también de ser requerido, la consulta de registros y la adición de notas a los discipulados.

#### **Diagrama de actividades específicos**

A continuación, se presentan cada uno de los diagramas referentes a los casos de uso más relevantes e importantes del contexto real del problema; los cuales muestran el flujo de actividades de un caso específico del problema.

*Diagrama de actividades: Registra congregante*

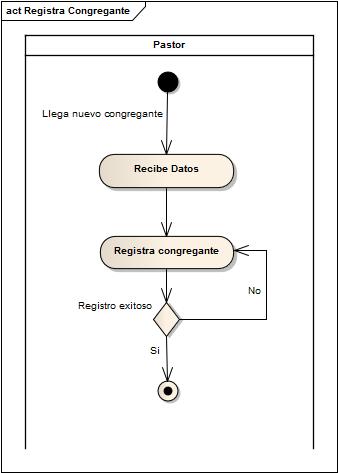


Figura N°4. Diagrama de actividades, registra congregante.

Para controlar el registro, el congregante llega a la iglesia y suministra los datos que sean necesarios para esto, el pastor recibe estos datos y los verifica, en caso de que estos datos sean incorrectos, el congregante debe suministrar nuevamente lo datos de tal forma que no se tengas errores; en caso contrario, de que estos datos sean correctos y tengan concordancia con la base de datos, el pastor permite el ingreso del congregante a la iglesia.

*Diagrama de actividades: Asiste servicio*

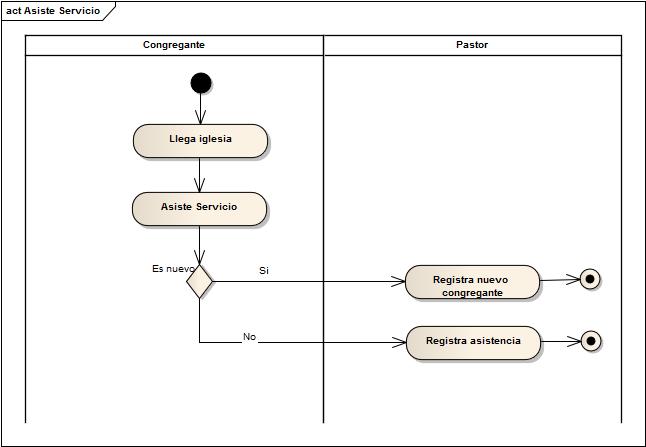


Figura N°5. Diagrama de actividades, asiste servicio.

Para asistir al servicio, el congregante llega a la iglesia, asiste al servicio, si este es nuevo, debe realizar el proceso de registro requerido, de lo contrario, si es antiguo, se registra la asistencia; estas dos últimas actividades son realizadas por el Pastor.

*Diagrama de actividades: Asiste discipulado*

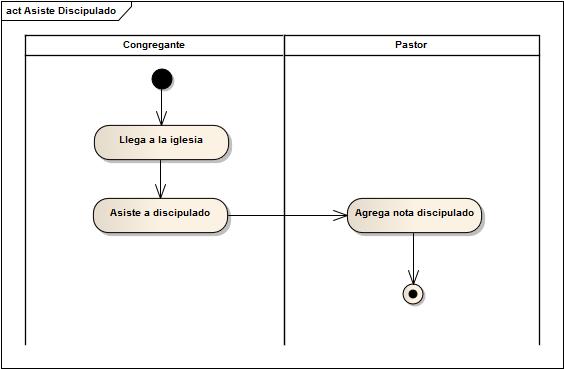
**

Figura N°6. Diagrama de actividades, asiste discipulado.

Para asistir al discipulado, el congregante llega a la iglesia, asiste al discipulado y el pastor agrega la nota de este al discipulado al cual pertenezca.

*Diagrama de actividades: Colabora ministerio*

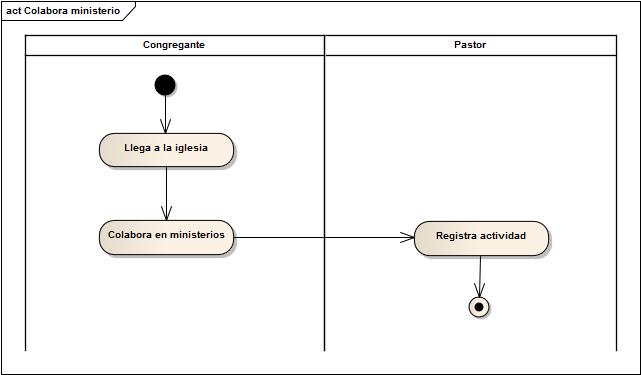
**

Figura N°7. Diagrama de actividades, asiste ministerio

Para colaborar en el ministerio, el congregante debe llegar a la iglesia, colabora en el ministerio y el pastor registra la actividad dependiendo del ministerio al que este pertenezca.

## **Glosario**

* **Asistente:** Persona que asiste a la iglesia, escucha la palabra y que siguiendo las normativas de la iglesia puede ser miembro de ésta.
* **Comunidad Cristiana:** grupo de personas que asociadas tienen como objetivo llevar una vida o gustos en común basadas en el apoyo y ayuda mutua. Se basan en alabar la vida del hijo de Dios hecho hombre, su vida, su sacrificio y su paso por la tierra.
* **Congestión:** Obstrucción ocasionada por la aglomeración excesiva de datos personales.
* **Congregación:** grupo de religiosos que viven en comunidad, obedientes a una misma regla y bajo una misma dirección
* **Congregante:** Participante o miembro de una congregación.
* **Discipulado:** Grupo de personas que escuchan, estudian e interpretan el contenido de la Biblia más a fondo.
* **Miembro:** Persona perteneciente a una congregación local.
* **Ministerio:** Servicio prestado a Dios ni a la gente. Ministerio en la iglesia tiene como meta la edificación de los individuos, sirviendo a Dios con madurez corporativa en Cristo.
* **Ministro:** Persona que sirve a Dios Suele ser el líder de la congregación y de los ministerios que ofrece la iglesia.
* **Nota:** Calificación sacada en cada uno de los cortes o periodos que ofrece el discipulado.
* **Pastor:** Prelado o cualquier otro eclesiástico que tiene a su cargo un grupo de fieles, sobre todo en las Iglesias reformadas
* **Servicio o Culto:** El culto es el centro de la vida y de la actividad de la iglesia. Podríamos decir que es su rostro. El mundo, instintivamente, identifica la iglesia con lo que ocurre en el culto.

*\*La definición y/o concepto de los términos previamente descritos en el glosario fue tomada de la página web* ***Word Reference****.*

# **REQUISITOS**

## **Visión**

La visión del proyecto a desarrollar se centra en que sea un software de calidad, donde el pastor de la iglesia Centro Bíblico Internacional Un Oasis de Esperanza pueda tener una aplicación, la cual le facilite el proceso de registro, asistencia, control y asignación de notas académicas de discipulados para cada congregante. También se manejarán los ministerios de dicha iglesia para saber que personas están involucradas en cada ministerio y cuál es su cargo.

Entre las características y funciones del producto software tenemos:

* Fácil manejo del software para el pastor de la iglesia.
* Capacidad de adaptarse a cualquier tipo de cambios en el producto.
* Capacidad para guardar datos personales de cada congregante.
* Capacidad de calcular promedios para las notas obtenidas por los congregantes en cada uno de los diferentes discipulados.

## **Especificación de requisitos**

La especificación de los requisitos del software se presenta en el documento de especificación de requisitos, este documento muestra una descripción completa y detallada del comportamiento del software desarrollado (CBI), para delimitar su alcance. Esto mediante los requisitos funcionales que debe ser capaz de realizar, las restricciones que debe cumplir y los requisitos no funcionales para tener en cuenta en la implementación de la solución para desarrollar un producto software de calidad. Este documento se encuentra en calidad de anexo en la carpeta de Anexos del proyecto.

## **Requisitos priorizados**

|  |  |
| --- | --- |
| **Requisito** | **1** |
| Nombre | Administrar Congregante |
| Descripción | El software debe permitir al administrador, registrar a una persona. |
| Tipo | Requisito |
| Prioridad | Alta-Esencial |

|  |  |
| --- | --- |
| **Requisito** | **2** |
| Nombre | Ver Congregante |
| Descripción | El software debe tener la posibilidad de consultar el registro de una persona en la aplicación. |
| Tipo | Requisito |
| Prioridad | Alta-Esencial |

|  |  |
| --- | --- |
| **Requisito** | **3** |
| Nombre | Control asistencia |
| Descripción | El software debe permitir al administrador llevar un control de la asistencia de la persona. |
| Tipo | Requisito |
| Prioridad | Alta-Esencial |

|  |  |
| --- | --- |
| **Requisito** | **4** |
| Nombre | Agrega o ver nota discipulado |
| Descripción | El software debe permitir agregar notas a los congregantes pertenecientes a los distintos discipulados. |
| Tipo | Requisito |
| Prioridad | Media-Should |

|  |  |
| --- | --- |
| **Requisito** | **5** |
| Nombre | Verificar registro |
| Descripción | El software podrá verificar si el congregante que se presenta ante el Pastor o al cual el Pastor desea gestionar algún dato se encuentra previamente registrado, por medio de su nombre o identificación. |
| Tipo | Requisito |
| Prioridad | Alta - Must |

## **Diagrama de casos de uso de requisitos**

A continuación, se presenta el diagrama de casos de uso a nivel de requisitos, en el cual se observan los casos referentes a los requisitos del sistema y como los actores intervienen en cada uno de ellos; en este caso se observa como el pastor quien es el actor involucrado, se relaciona con cada uno de los casos.

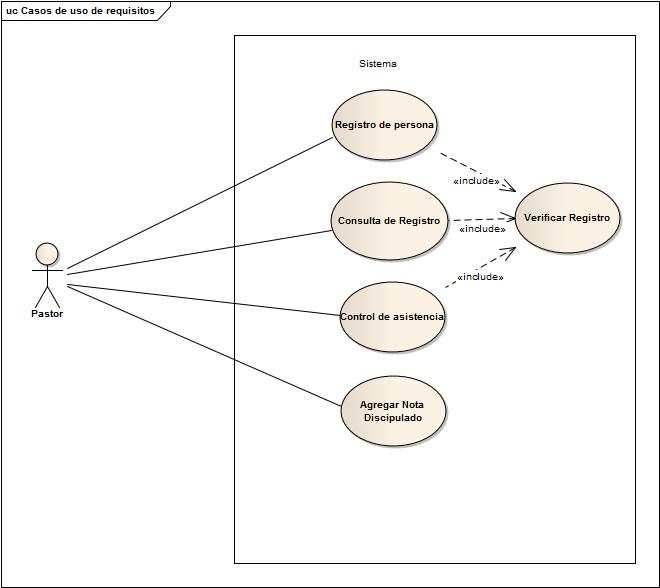


Figura N°8. Diagrama de casos de uso de requisitos.

# **MODELO DE DISEÑO**

Los pasos para diseñar un producto software son: primero, se debe establecer y documentar los requisitos del sistema, luego diseñar y documentar la estructura, seguido a esto, diseñar y documentar las interfaces, para luego diseñar y documentar la jerarquía de usos, más adelante diseñar y documentar las estructuras internas, más tarde escribir programas y finalmente hacer mantenimiento de los anteriores.

Los pasos dichos anteriormente, son indispensables para el desarrollo de un producto software de calidad, ya que siguiendo esta cadena de pasos podemos llegar a realizar una buena arquitectura, documentación y programación de nuestro sistema y poder así brindar un buen mantenimiento del software para que al momento de que haya que hacer un cambio, se cambie la documentación y así poder mejorar el desarrollo de dicho producto software.

## **Vista lógica**

### **Diagrama de componentes**

A continuación, se presenta el diagrama de componentes del sistema, el cual muestra la relación existente entre los componentes que conforman el sistema referente a la solución del problema.

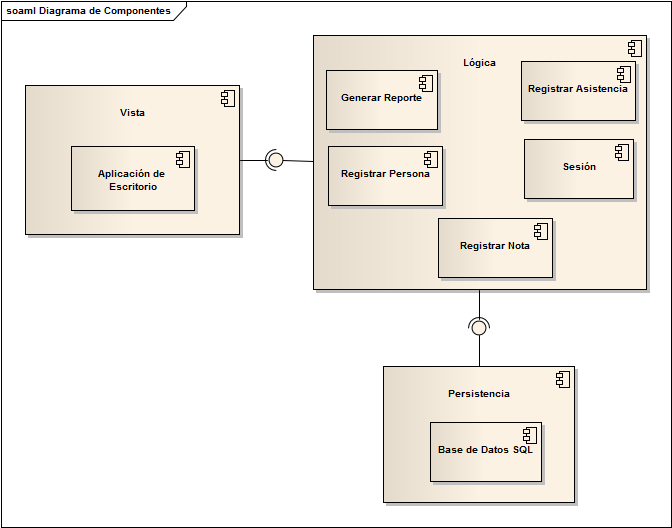


Figura N°9. Diagrama de componentes.

En el diagrama se pueden apreciar las relaciones entre los componentes del sistema donde el componente principal, Lógica, hace uso de la Persistencia para poder hacer el desarrollo de las operaciones del programa con la información guardada en la base de datos. Lógica a su vez es consumida por Vista para visualizar los resultados de las consultas y registros realizados en el componente de lógica.

### **Diagrama de clases**

#### **Diagrama de clases general**

En este diagrama se pueden observar los componentes, las relaciones entre ellos y cuales componentes se ven involucrados en tal relación. La clase Iglesia es creada para manejar la mayor parte del sistema y para hacer uso de las demás clases del sistema, las clases Congregante y Pastor heredan de la clase Persona, ya que ambas clases antes mencionadas tienen semejanzas en uso de variables, pero no todas son las mismas. La clase ActividadMinisterio hereda de Ministerio, ya que los ministerios en una iglesia pueden tener personas sirviendo, y también pueden tener actividades que son programadas. La clase NotaDiscipulado hereda de Discipulado, ya que la iglesia puede tener varias etapas o cursos de discipulados y a éstos se les puede agregar notas por cortes. Finalmente, la clase Asistencia tiene la tarea de llevar un registro de asistencias para tener en cuenta que días asiste y no asiste un congregante a la iglesia.

Imagen que contiene captura de pantalla, texto

Descripción generada con confianza muy alta

Figura N°10. Diagrama de clases general.

#### **Diagrama de clases de los casos de uso más representativos**

A continuación, se presentan cada uno de los diagramas referentes a los diagramas de clases de los casos de uso a nivel de requisitos, mas importantes y/o relevantes; estos diagramas de clases muestran las relaciones existentes entre los objetos que integran las funcionalidades del software. Los casos de uso a nivel de requisitos considerados los más relevantes son, ‘registrar persona’, ‘agregar nota discipulado’ y ‘control asistencia’.

*Diagrama de clases: Registro persona*

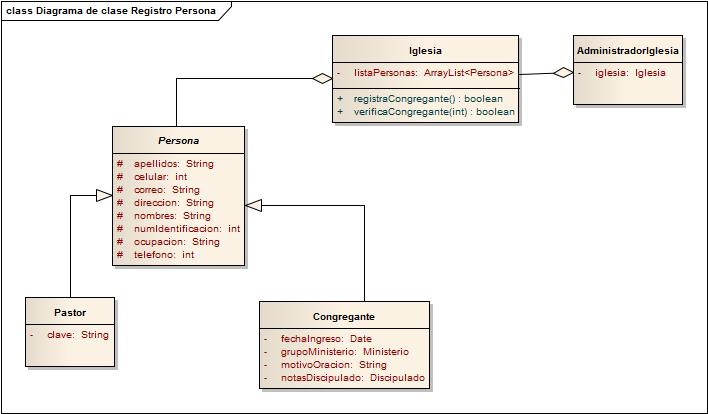


Figura N°11. Diagrama de clases, registro persona.

En este diagrama observamos dos tipos de relaciones entre componentes y que componentes se ven involucrados en tal relación. La clase Persona es creada para ser una clase general de la cual heredan Pastor y Congregante. La clase iglesia es creada para crearse un ArrayList de tipo Persona, por lo cual se encuentra relacionada con la clase Persona. La clase AdministradorIglesia se crea haciendo uso del patrón controlador, ya que permite separar la lógica del negocio con la interfaz de presentación.

*Diagrama de clases: Agregar nota discipulado*

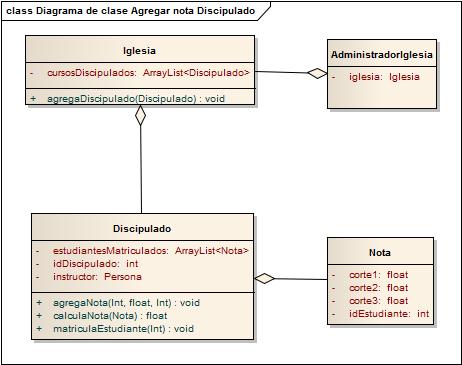
**

Figura N°12. Diagrama de clases, agregar nota discipulado.

En este diagrama observamos un tipo de relación entre los componentes y los componentes involucrados en tal relación. La clase Nota, es la clase en la cual se van a almacenar el id del congregante (estudiante) y la nota de los tres cortes del curso. La clase Discipulado es la encargada de guardar el instructor a cargo, el ArrayList de notas, el cual es la lista de estudiantes matriculados en el curso y un id de curso, el cual lo identifica. Esta clase contiene los métodos de agregar nota, calcular la nota final y agregar estudiantes al curso. La clase Iglesia es aquella donde se crea la instancia de los cursos de discipulado, por lo tanto, esta sería la clase creadora, es decir, se hace uso en este diagrama del patrón creador. La clase AdministradorIglesia se crea haciendo uso del patrón controlador, ya que permite separar la lógica del negocio con la interfaz de presentación.

*Diagrama de clases: Control asistencia*

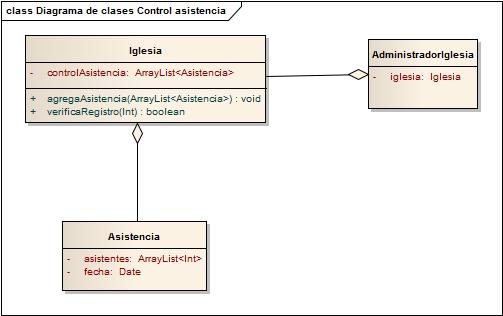
**

Figura N°13. Diagrama de clases, control asistencia.

En este diagrama observamos un tipo de relación entre los componentes y los componentes involucrados en tal relación. Contiene una clase Asistencia que posee un ArrayList de enteros, el cual se refiere al número de identificación de cada congregante que ha asistido al servicio; el identificador seria en este caso la fecha. La clase Iglesia es aquella donde se crea la instancia de los cursos de discipulado, por lo tanto, esta sería la clase creadora, es decir, se hace uso en este diagrama del patrón creador. La clase AdministradorIglesia se crea haciendo uso del patrón controlador, ya que permite separar la lógica del negocio con la interfaz de presentación.

#### **Diseño de clase (lógica, atributos y métodos)**

*Iglesia*

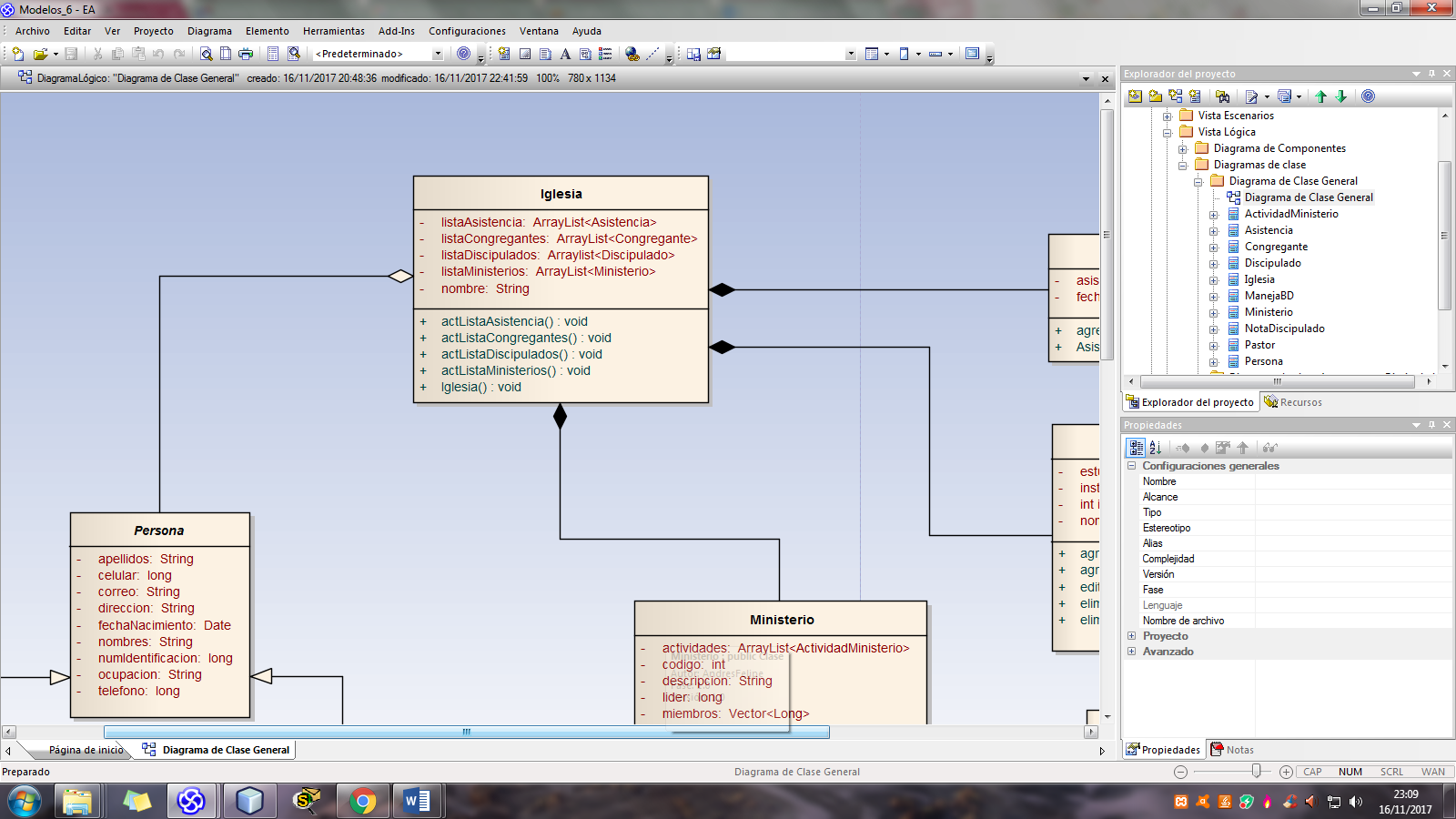


Figura N°14. Clase Iglesia.

*Persona*

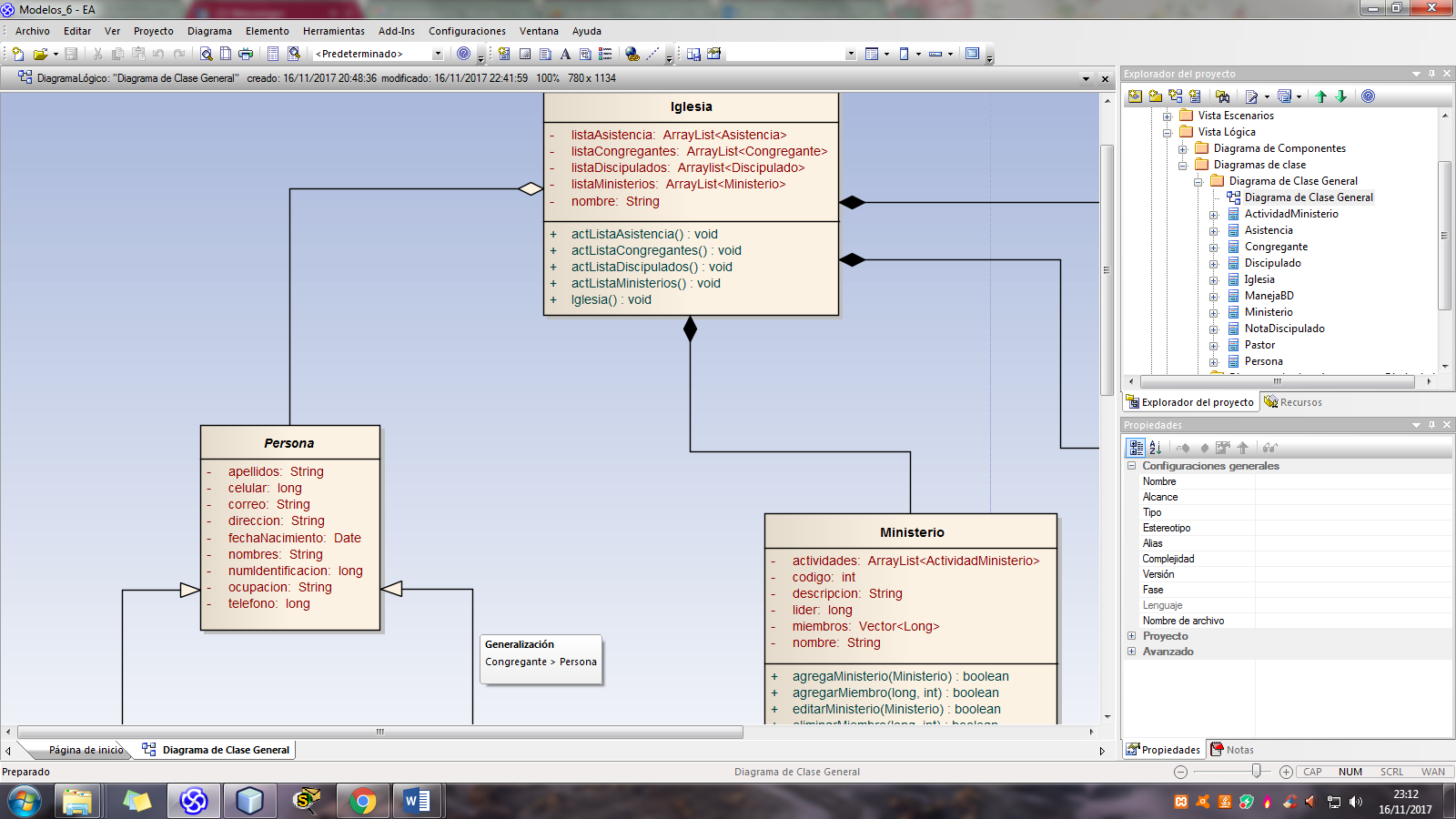


Figura N°15. Clase Persona.

*Pastor*

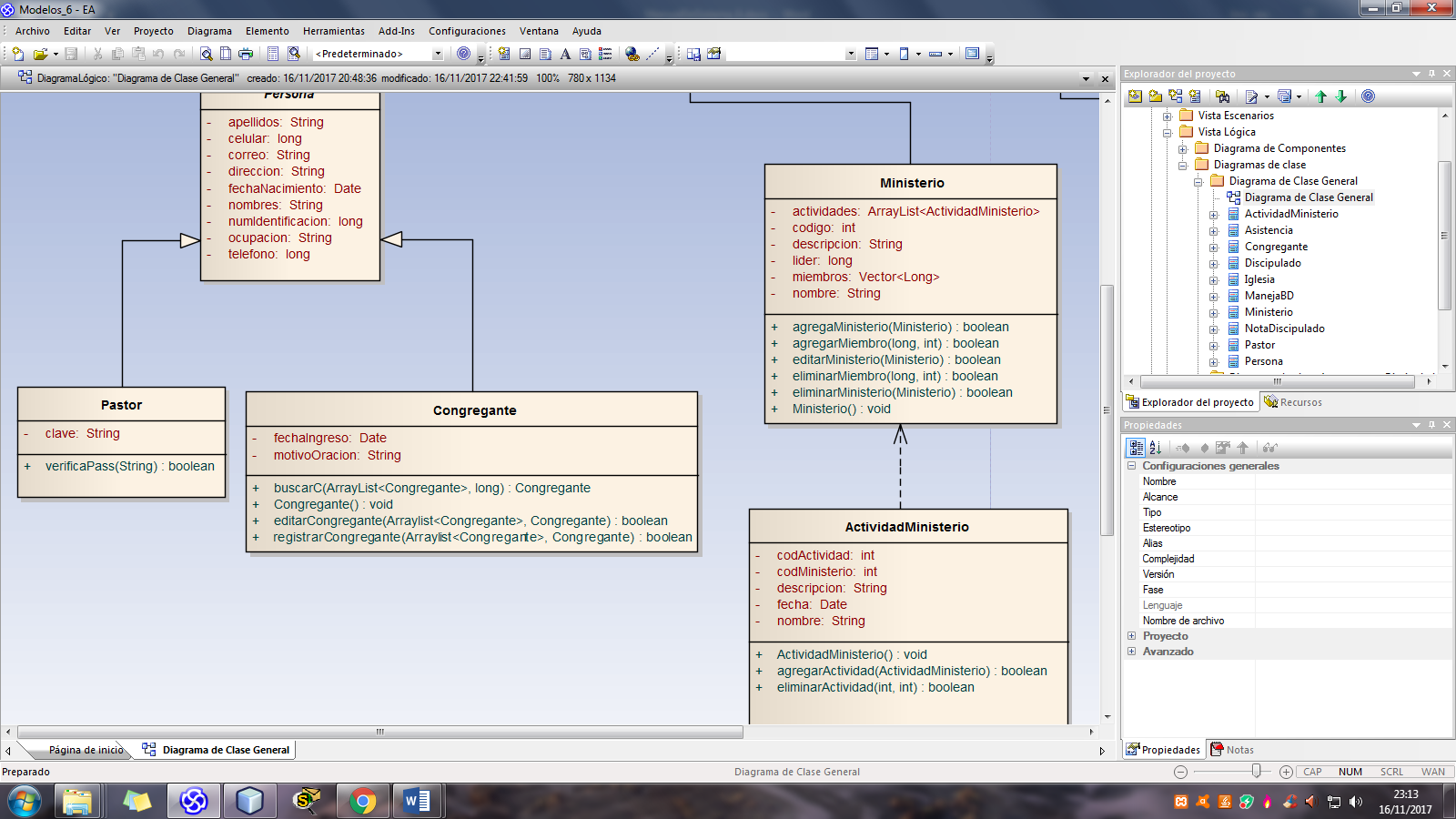


Figura N°16. Clase Pastor.

*congregante*

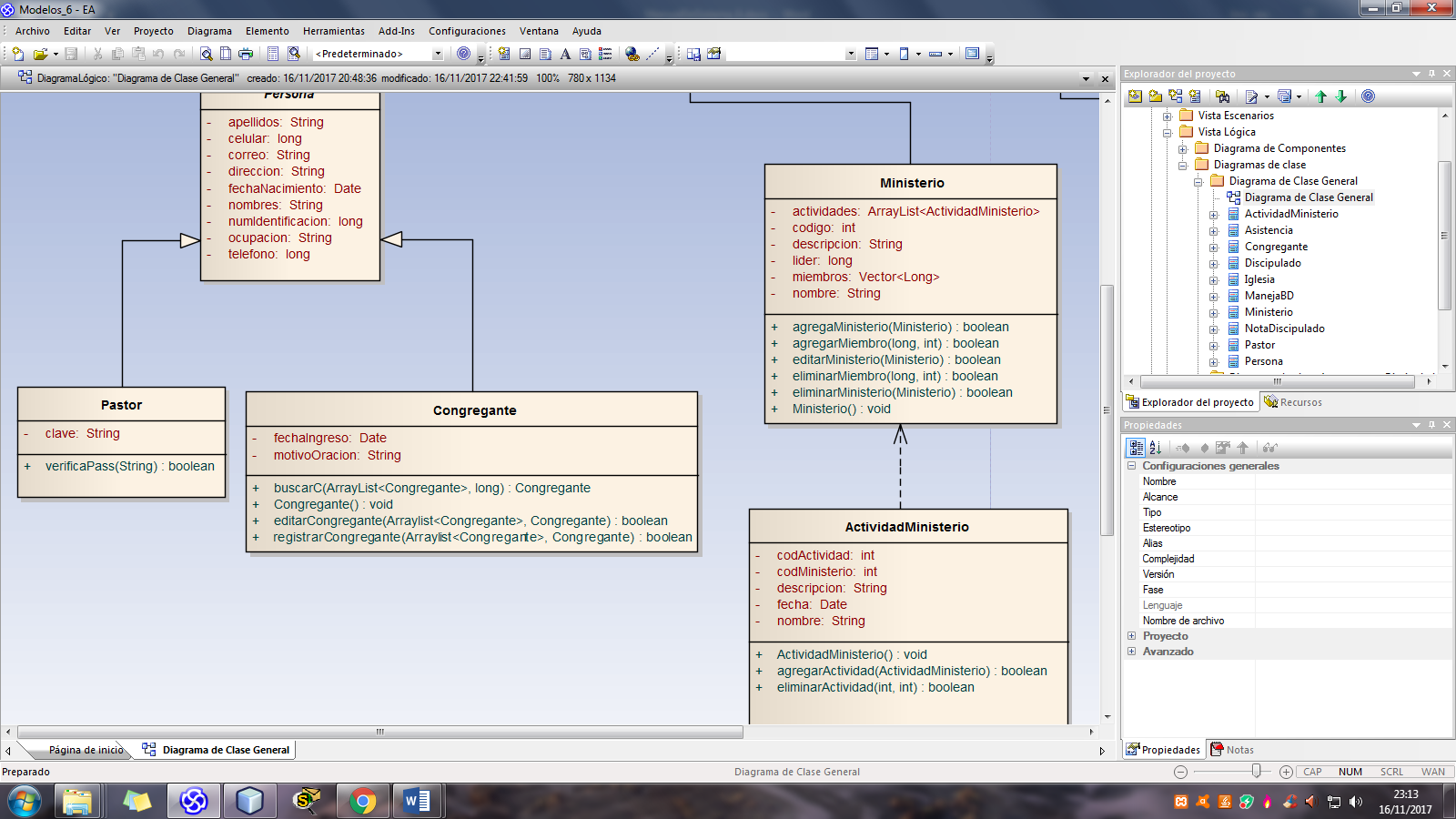


Figura N°17. Clase Congregante.

*Ministerio*

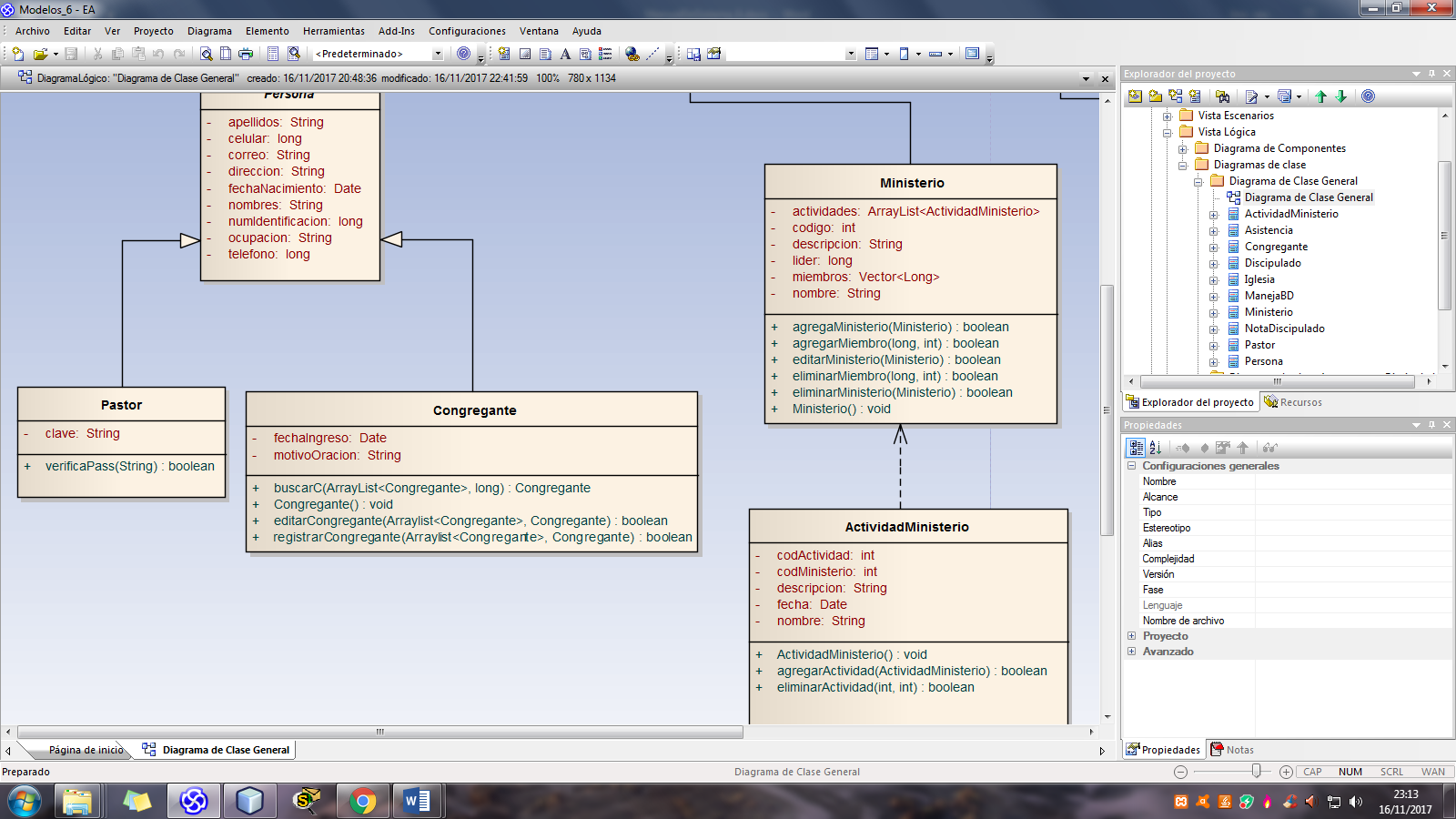


Figura N°18. Clase Ministerio.

*ActividadMinisterio*

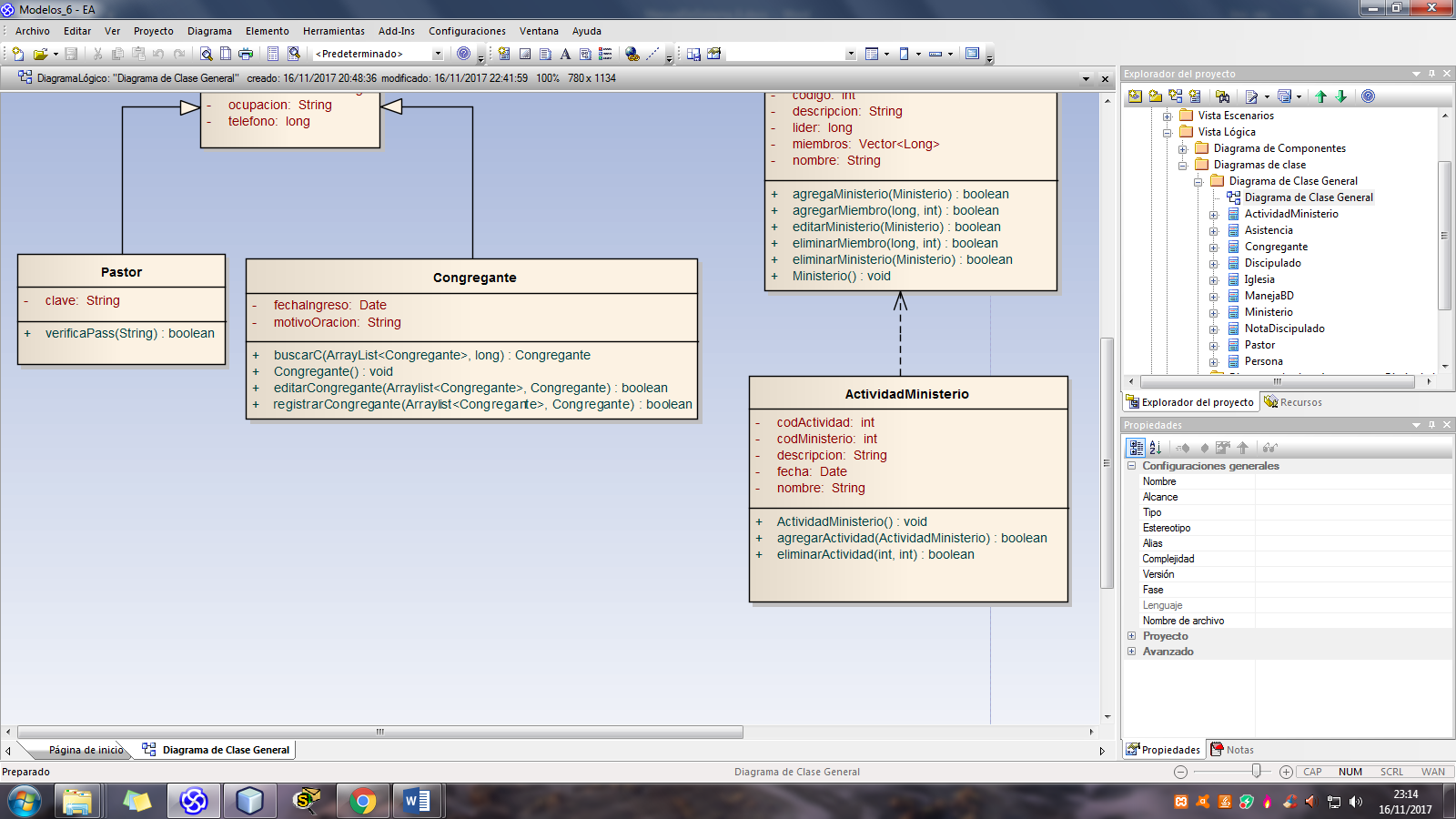


Figura N°19. Clase ActividadMinisterio.

*Asistencia*

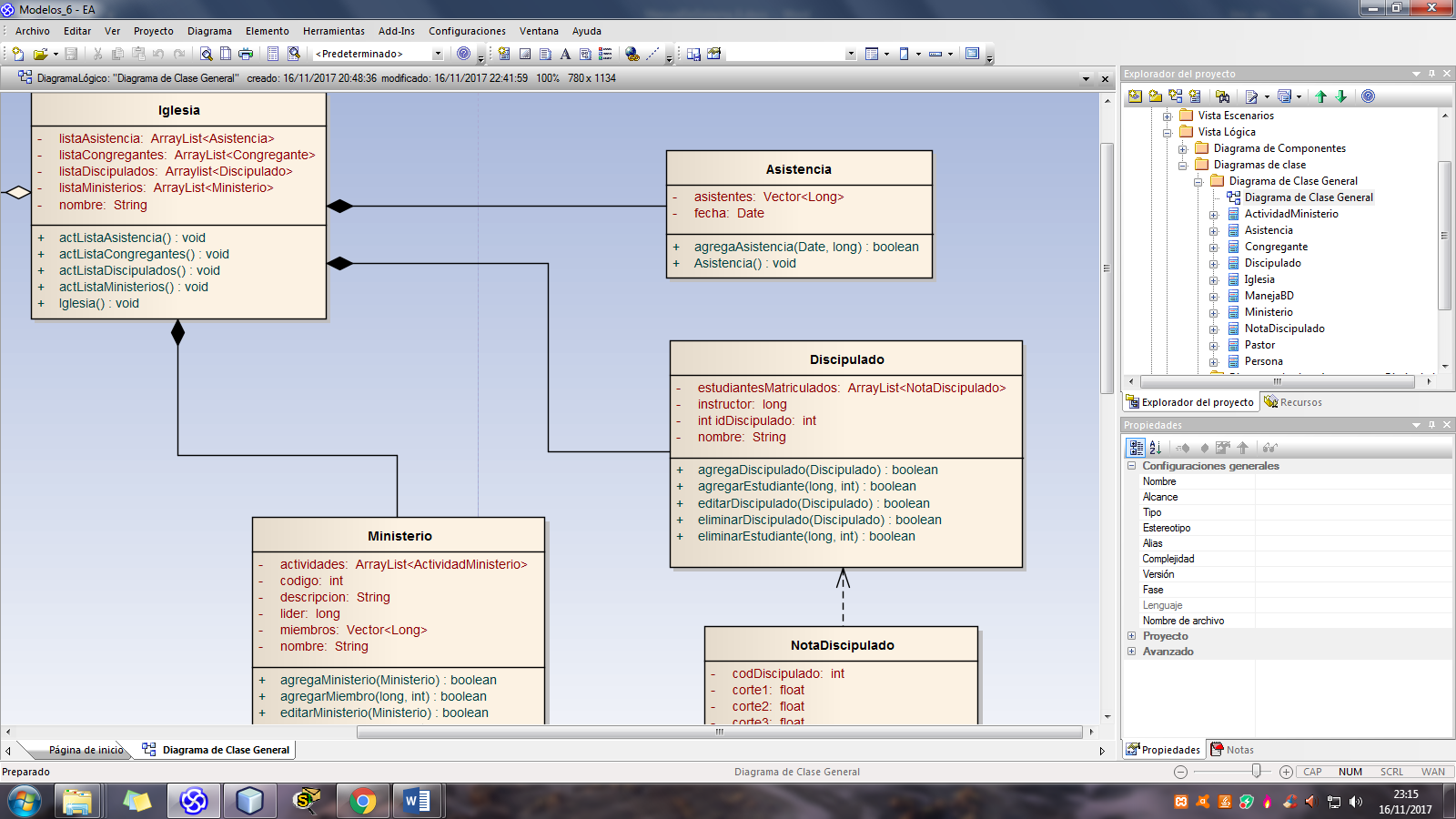


Figura N°20. Clase Asistencia.

*Discipulado*

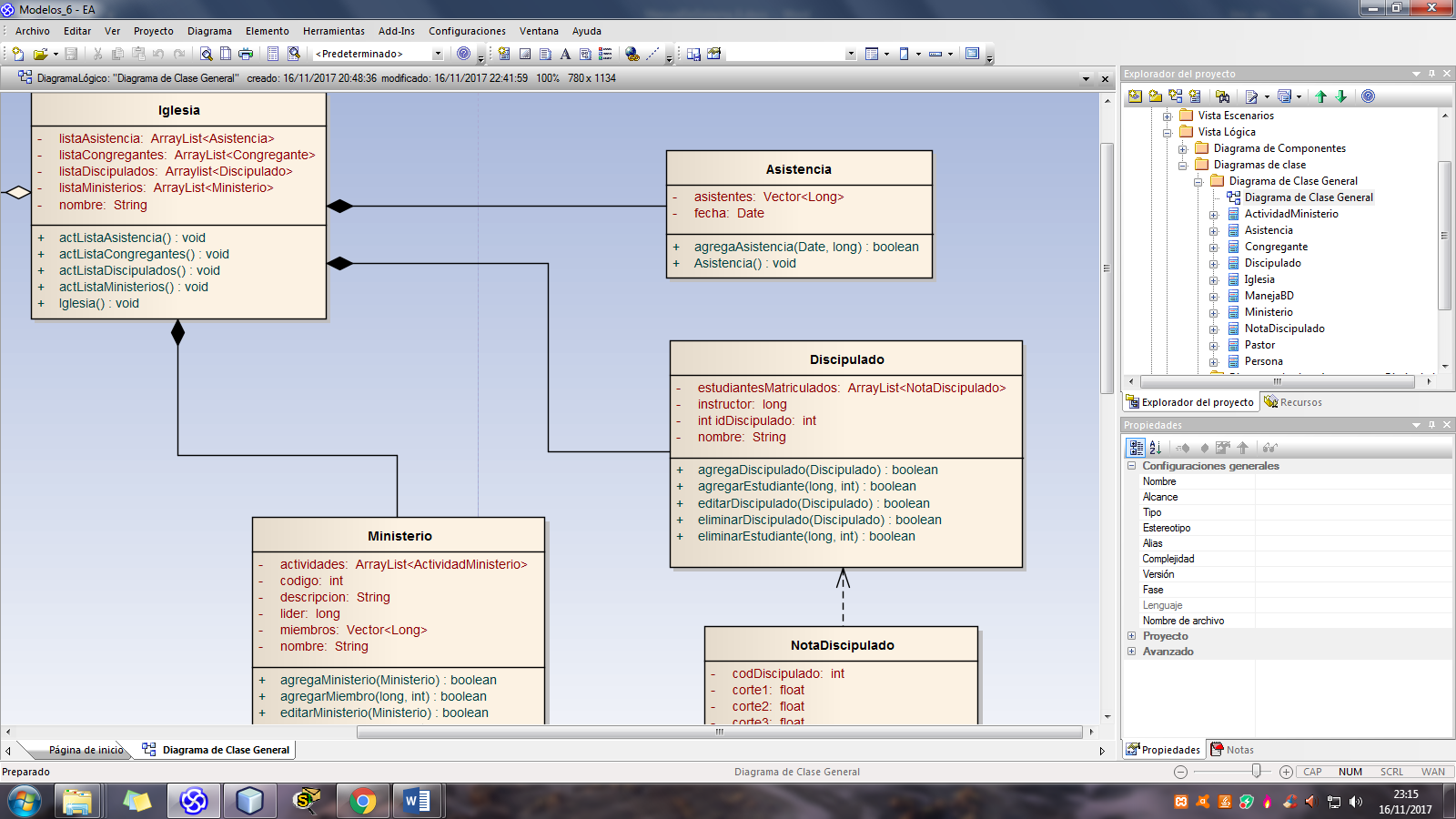


Figura N°21. Clase Discipulado.

*NotaDiscipulado*

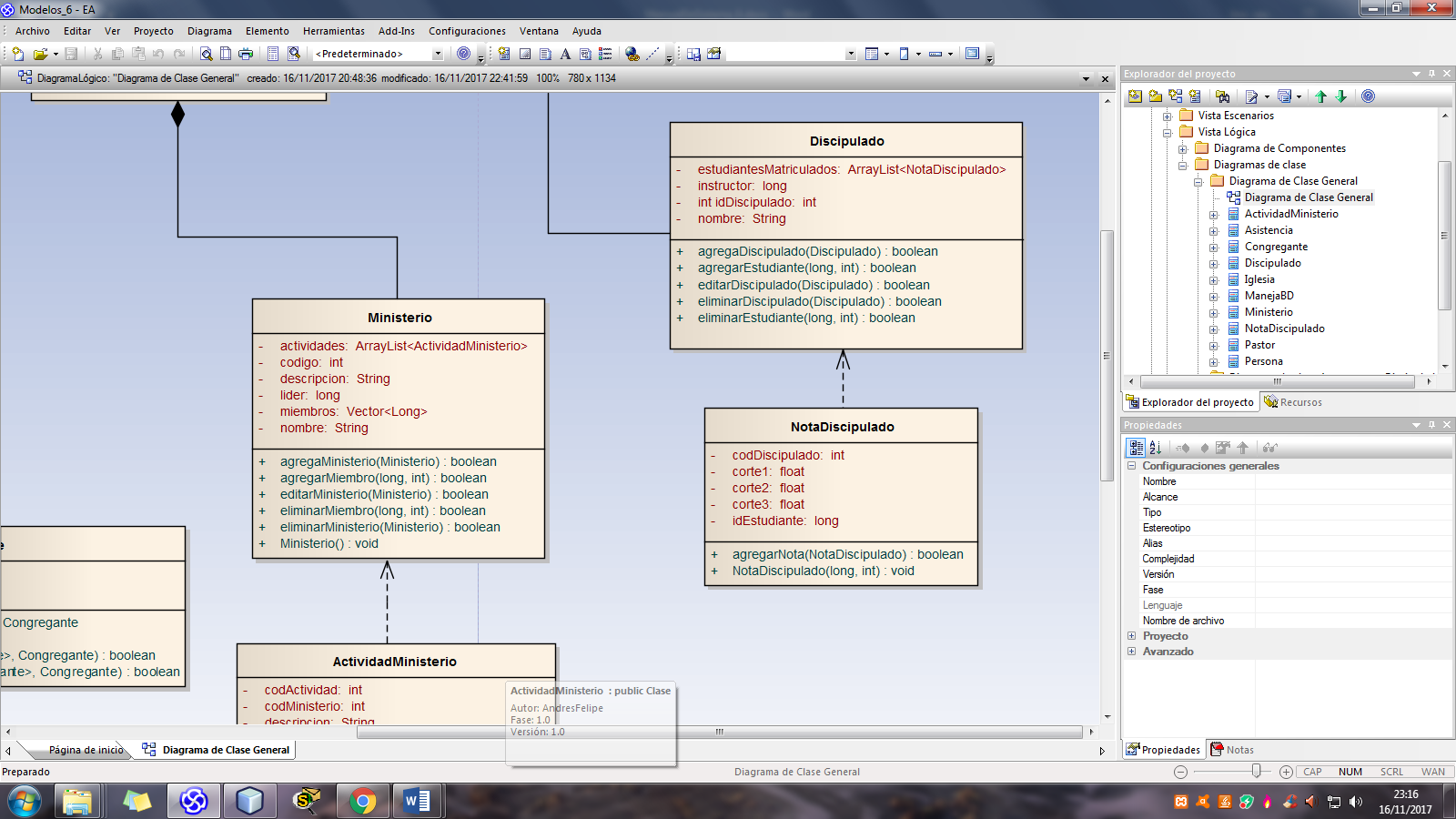


Figura N°22. Clase NotaDiscipulado.

## **Vista de escenarios**

### **Diagrama general de casos de uso**

A continuación, se presenta el diagrama de casos de uso a nivel de diseño, el cual representa los casos de uso que permite realizar el software para el usuario.

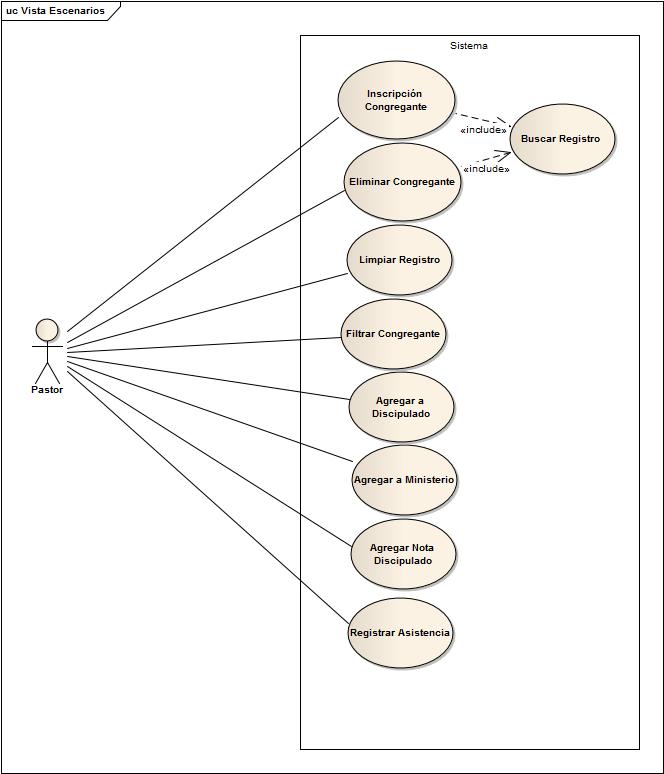


Figura N°23. Diagrama de casos de uso de diseño.

El diagrama general de casos de uso nos muestra las diferentes funcionalidades del software CBI a la que el usuario, en este caso el Pastor, puede acceder y hacer uso del mismo.

Contamos con un actor Activo Pastor y 10 casos de uso principales funcionalidades disponibles dentro del sistema. Entre estas funcionalidades tenemos:

* **Buscar registro:** Este caso de uso describe la posibilidad del pastor de poder buscar información de los congregantes registrados en la iglesia, mostrarla en pantalla y el cómo hacerlo.
* **Agregar congregante:** Este caso de uso describe la posibilidad del pastor de poder agregar (registrar) un congregante que ingresa a la iglesia y el cómo hacerlo.
* **Eliminar congregante:** Este caso de uso muestra la capacidad del pastor de poder eliminar a uno o varios congregantes que ingresaron y registraron en la iglesia y el cómo hacerlo.
* **Limpiar registro:** En este caso de uso el pastor está en la capacidad de limpiar la tabla que muestra los congregantes que ingresaron a la iglesia, junto con su base de datos, y el cómo hacerlo.
* **Filtrar congregante:** En este caso de uso se describe como el pastor puede filtrar la tabla de los registrados con base en un campo especifico, para mostrar la información de un congregante especifico.
* **Agregar a discipulado:** En este caso se describe como el pastor puede agregar un congregante a un discipulado acorde con el conocimiento y niveles previos que este posea.
* **Agregar a ministerio:** En este caso de uso se describe como el pastor puede agregar un congregante a un ministerio, teniendo en cuenta sus capacidades y conocimientos.
* **Agregar nota a discipulado:** En este caso de uso se describe como el pastor puede agregar una nota cuantitativa a un determinado congregante de un determinado discipulado acorde a sus evaluaciones y avances en el nivel de discipulado en el que se encuentre.
* **Registrar asistencia:** En este caso de uso se describe como el pastor puede llevar un registro diario de la asistencia a la iglesia, es decir, quienes asisten y quienes no a la iglesia y sus servicios, actividades y grupos, y con qué frecuencia lo hacen.

### **Diseño interfaz gráfica de usuario**

Para el desarrollo de la interfaz de usuario se tuvieron en cuenta los principios de diseño aprendidos en clase para cumplir con el objetivo de llevar a cabo un producto software de calidad.

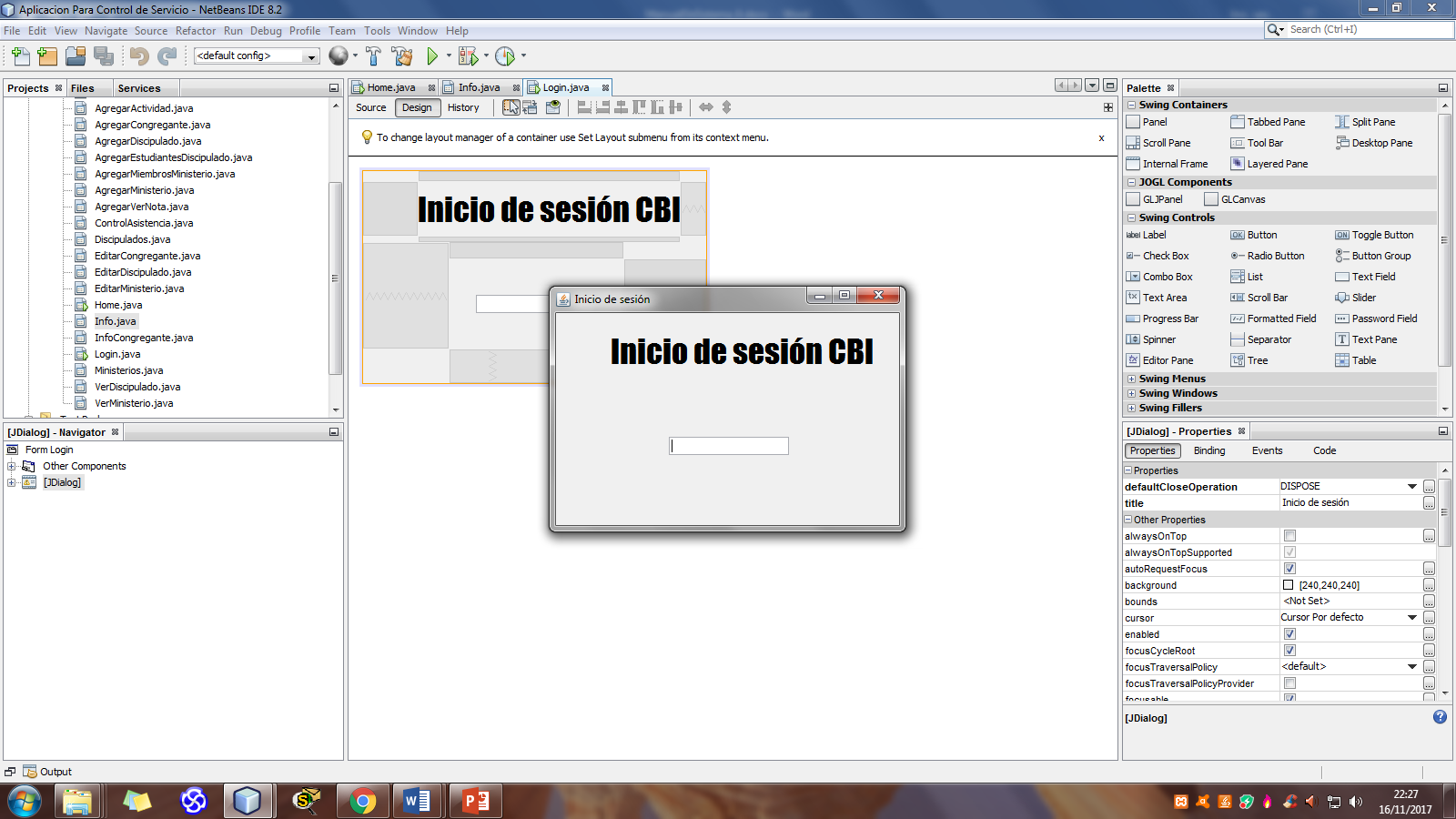


Figura N°24. Interfaz inicio de sesión.

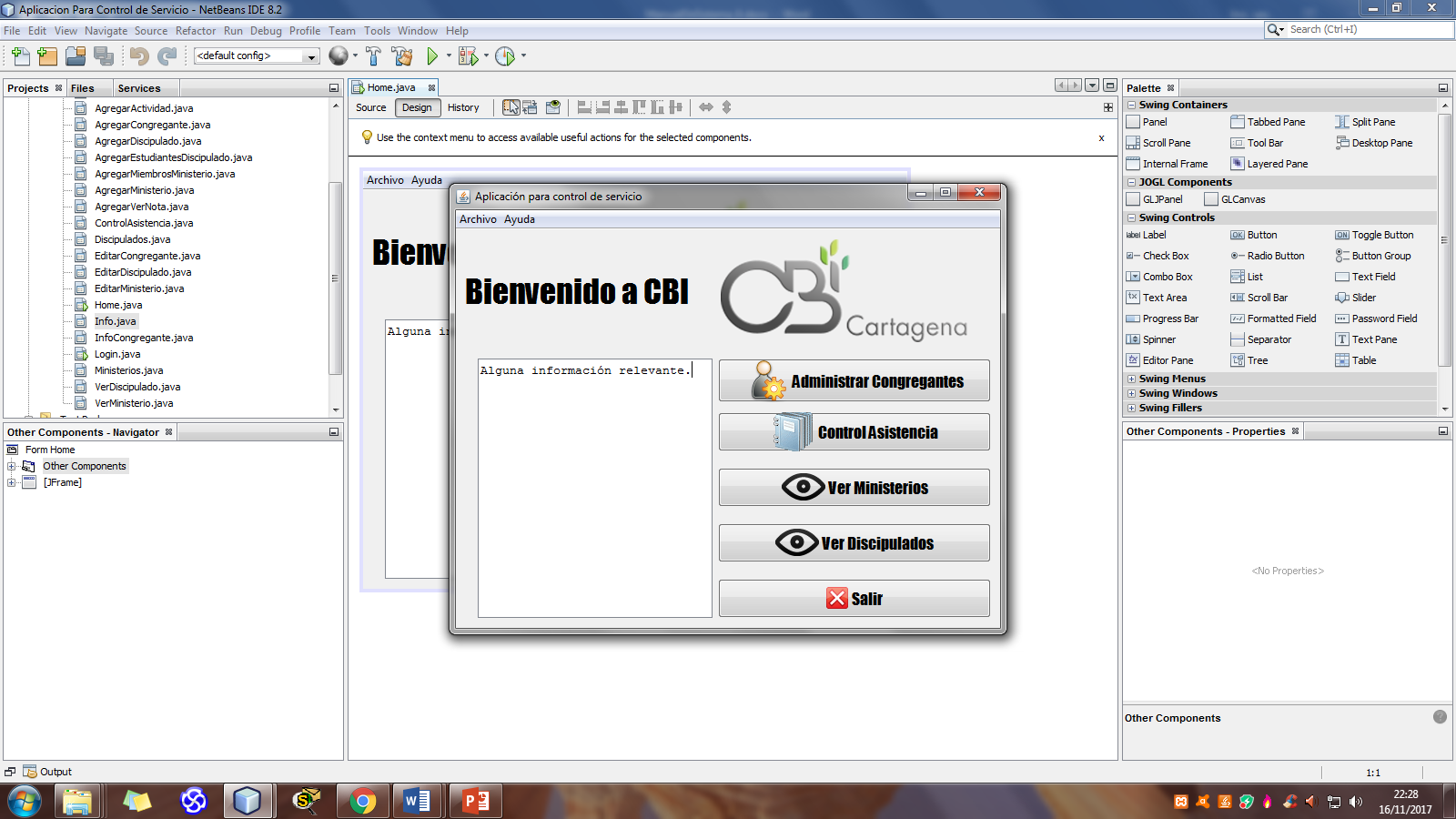


Figura N°25. Interfaz de entrada.

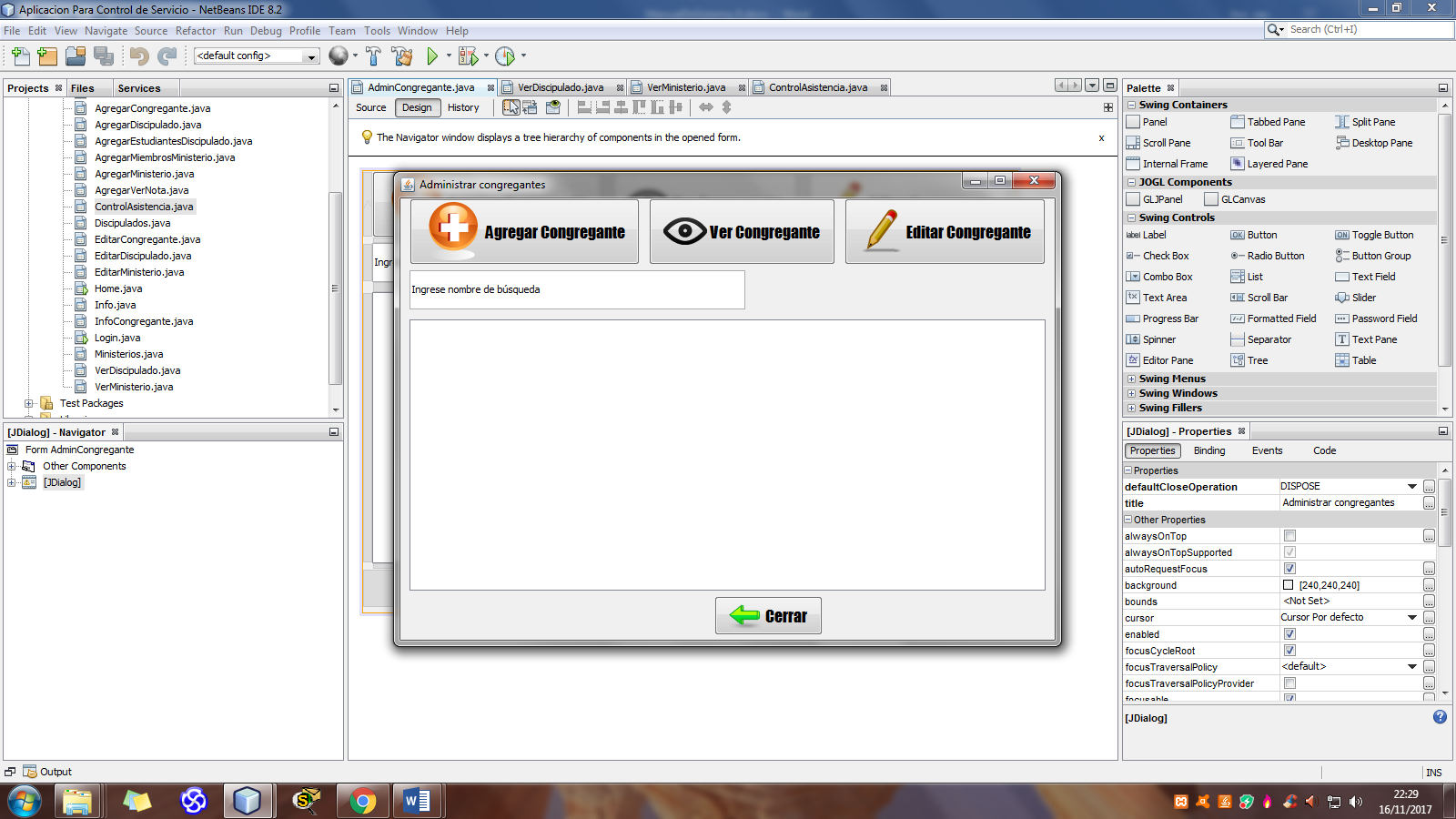


Figura N°26. Interfaz de administrar congregantes.

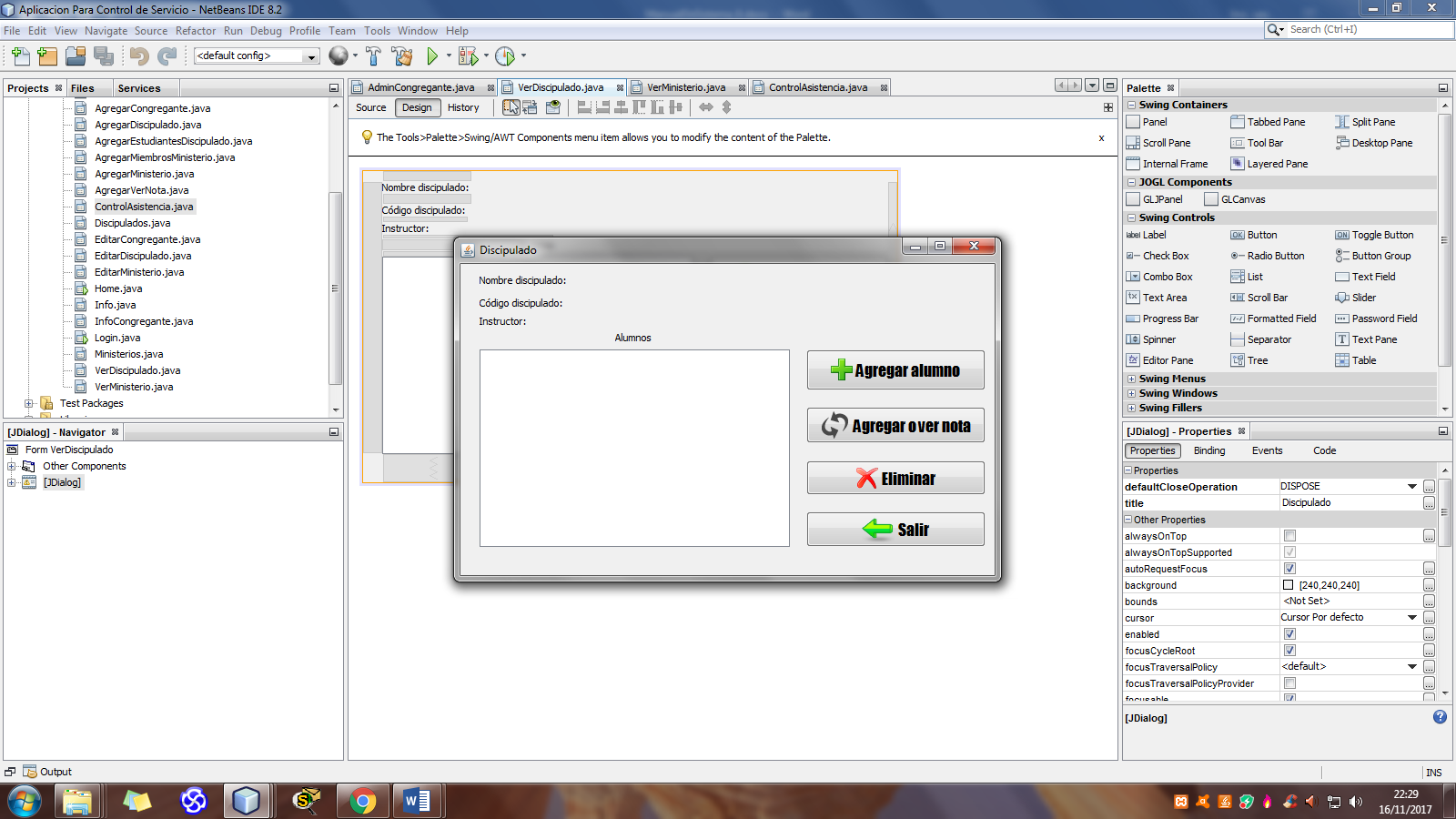


Figura N°27. Interfaz de ver discipulados.

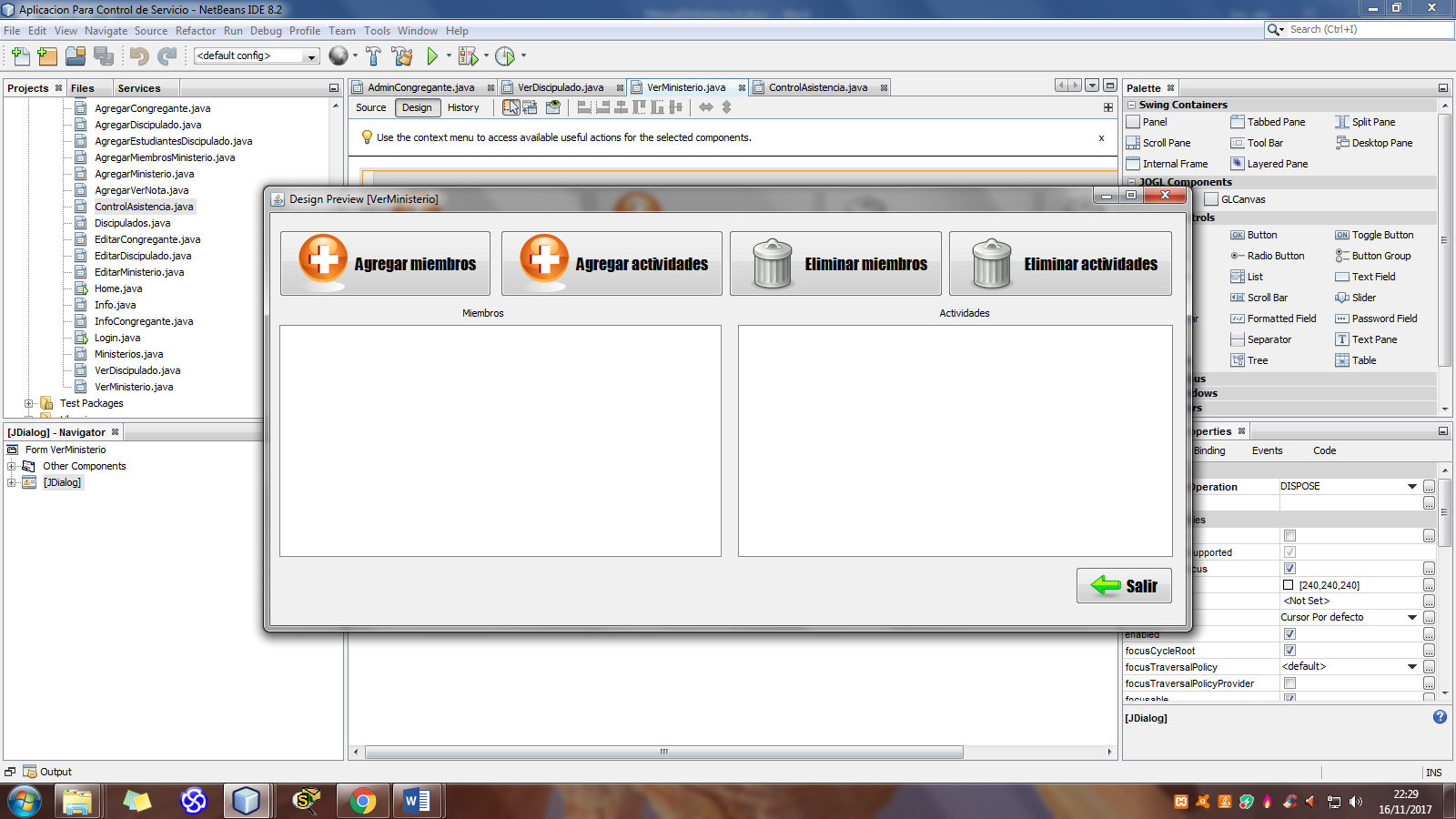


Figura N°28. Interfaz de ver ministerios.

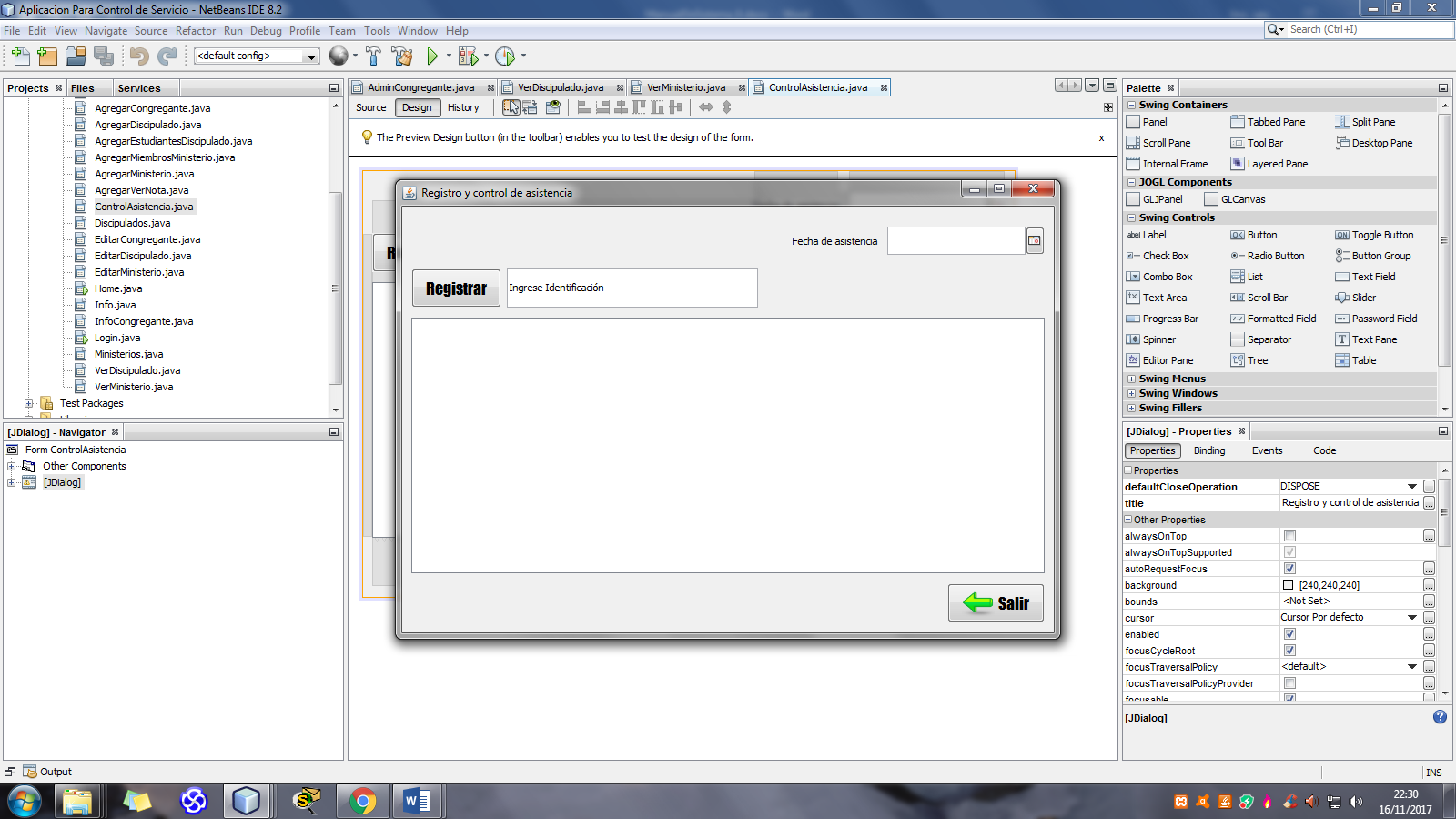


Figura N°29. Interfaz de control de asistencia.

* **Principio estético:** Se aplicaron colores poco llamativos y que fueran tolerantes a la vista del usuario; encargado, quien hará uso del programa por varias horas seguidas y no pretendemos que este resulte con la vista cansada o fastidiado.
* **Fácil de comprender:** Desarrollamos una vista que resultara bastante clara para el usuario, una vista simple con solo lo necesario. Además, desarrollamos vistas que tuvieran consistencia por lo que se puede observar que estas vistas son muy similares, permitiendo así que el encargado aprenda rápido el uso de ellas.
* **Control:** El sistema dispone de un bloque de texto que muestra al usuario cualquier tipo de error o información detallada requerida. Además, los botones se encuentran configurados para habilitarse solo al momento en que tal acción pueda ser ejecutada, de lo contrario no será posible hacer uso de ellos.
* **Eficiencia:** Implementamos un botón en cada una de estas vistas que permita ir directamente a la otra, así ahorrando trabajo y tiempo.
* **Flexibilidad:** La interfaz es simple, ajustándose a las posibles habilidades o capacidades con las que cuenta quien hará uso del sistema.

## **Vista de procesos**

### **Diagrama de secuencia de los casos de uso más representativos**

A continuación, se muestran los diagramas de secuencia de los casos de uso más representativos del sistema, los cuales describen que sucede en el sistema a través del tiempo y como los elementos se comunican entre sí.

*Diagrama de secuencia: Registro congregante*

*Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta*

Figura N°30. Diagrama de casos de uso de diseño.

*Diagrama de secuencia: Verificar inscripción congregante*

*Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta*

Figura N°31. Diagrama de casos de uso de diseño.

# **MODELO DE IMPLEMENTACIÓN**

## **Vista de desarrollo**

### **Diagrama de paquetes**

A continuación, se presenta el diagrama de paquetes en relación con el producto software, el cual representa la forma como son implementados los elementos lógicos del sistema.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Figura N°32. Diagrama de paquetes.

En el anterior diagrama se observa como son implementados los elementos lógicos que integran el sistema como son el modelo, el cual contiene la base de datos y la lógica del sistema; la GUI que se refiere a la interfaz gráfica del usuario y las tablas, que contiene la tabla referente a la lista de los congregantes.

## **Vista física**

### **Diagrama de despliegue**

A continuación, se muestra el diagrama de despliegue del sistema, el cual representa los elementos físicos que requiere el producto software para su despliegue.

Imagen que contiene captura de pantalla

Descripción generada con confianza muy alta

Figura N°33. Diagrama de despliegue.

En el anterior diagrama se observan los elementos físicos que permiten el despliegue del sistema. En primer lugar, está el PC el cual es el primer medio para llevar a cabo la ejecución de nuestro producto software, seguido a este, sigue Windows, luego de esta, sigue la máquina virtual donde se ejecuta el programa *.jar,* el cual tiene una conexión SQL con My Sql, donde se encuentra ubicada la base de datos; de tal manera que la integración y relación de cada uno de los elementos físicos antes mencionados, permiten el despliegue del producto software en cuestión.

# **BIBLIOGRAFÍA**

PARNAS, D. L. Y CLEMENTS, P. C. A rational design process: How and why to fake it. **En el texto:** (Parnas and Clements, 2005). Parnas, D. and Clements, P. (2005). A rational design process: How and why to fake it. In: *Lecture Notes in Computer Science*. Springer, pp. https://link.springer.com/chapter/10.1007/3-540-15199-0\_6.