

## **I. CAPÍTULO I: EL PROYECTO**

### **I.1 PRESENTACIÓN DEL PROYECTO**

#### **I.1.1 Título**

Mejoramiento del sistema de gestión de la información del proyecto comunitario “Guadalquivir”.

#### **I.1.2 Área del Proyecto**

Área Administrativa.

#### **I.1.3 Responsable del Proyecto**

Univ. Omar IbáñezIbáñez

Carrera de Ingeniería Informática.

#### **I.1.4 Entidades Asociadas**

U.A.J.M.S. “Universidad Autónoma Juan Misael Saracho”

Proyecto comunitario “Guadalquivir”

#### **I.1.5 Grupo Responsable del Proyecto**

Omar Anheló IbáñezIbáñez – Estudiante de la carrera de ingeniería informática

#### **I.1.6 Personal Vinculado al Proyecto**

### I.1.7 Director de Proyecto

Apellido Paterno	Apellido Materno	Nombre	C.I.
<b>Ibáñez</b>	<b>Ibáñez</b>	<b>Omar Anhelo</b>	<b>7142681</b>
Profesión	Carrera o Unidad	Facultad	
<b>Estudiante de 5to año</b>	<b>Ingeniería informática</b>	<b>Ciencias y tecnología</b>	
Telf. Oficina	Celular	Correo electrónico	Firma
--	<b>79245007</b>	<b>Omar—ag@hotmail.com</b>	

Tabla: 1 Director del proyecto

### I.1.8 Participantes equipo de trabajo

Categoría	Nombres y Apellidos	Profesión	C.I.	Firma
<b>Director</b>	Omar A. Ibáñez Ibáñez	Estudiante	7142681	
<b>Analista y Diseñador de Sistemas</b>	Omar A. Ibáñez Ibáñez	Estudiante	7142681	
<b>Asesor</b>	Lic. Efrain Torrejón	Lic.		

		Informática		
<b>Programador</b>	Omar A. Ibáñez Ibáñez	Estudiante	7142681	
<b>Ing. De Software</b>	Omar A. Ibáñez Ibáñez	Estudiante	7142681	

Tabla: 2 Participantes del proyecto

#### I.1.9 Empresas/Instituciones/Organizaciones participantes/cooperantes

<b>Nombre: Proyecto Comunitario “Guadalquivir”</b>			
<b>Dirección: Barrio Pedro Antonio Flores</b>		<b>Teléf. Oficina:</b>	
<b>Nombre y Apellidos</b>	<b>Cargo</b>	<b>C.I.</b>	<b>Firma</b>
Lic. María Teresa Flores Adrián	Administradora		

Tabla: 3 Organización Cooperante

### I.1.10 Actividades previstas para los integrantes del equipo de investigación

Responsables	Actividades
<p style="text-align: center;"><b>Director</b></p>	<ul style="list-style-type: none"> <li>• Planificar y controlar el calendario del proyecto.</li> <li>• Seguimiento del cronograma y cumplimiento de las actividades</li> <li>• Árbol de Problemas y Objetivos</li> <li>• Asignación de recursos para las distintas actividades planificadas para el desarrollo del proyecto</li> <li>• Administrar el desarrollo del proyecto.</li> <li>• Asegurar la calidad e integridad el proyecto</li> <li>• Elaborar Manual de Usuario para los involucrados</li> <li>• Capacitar a los involucrados con el sistema en el correcto manejo del mismo.</li> </ul>
<p><b>Analista y Diseñador de Sistemas</b></p>	<ul style="list-style-type: none"> <li>• Recolección, especificación y validación de requisitos, tratando los mismos con el cliente y los usuarios.</li> <li>• Establecer la arquitectura del sistema.</li> <li>• Elaborar los diferentes modelados de Análisis y Diseño.</li> <li>• Colaborar en las pruebas de funcionalidad del sistema.</li> </ul>

<b>Programador</b>	<ul style="list-style-type: none"> <li>• Construcción del sistema de información.</li> <li>• Colaboración en la elaboración de las pruebas funcionales.</li> <li>• Colaboración de las validaciones con el usuario.</li> </ul>
<b>Asesor</b>	Encargado de la revisión en el proceso de desarrollo del proyecto, esta supervisión se realizara de acuerdo a las fechas acordadas, entre el universitario con el Administrador de la institución
<b>Ingeniero de Software</b>	<p>El Ingeniero de Software realizara las siguientes actividades:</p> <ul style="list-style-type: none"> <li>• Gestión de requisitos, gestión de configuración y cambios.</li> <li>• Preparación de las pruebas funcionales.</li> </ul>

Tabla: 4 Actividades previstas

## **I.2 Descripción del proyecto**

### **I.2.1 Resumen ejecutivo del proyecto**

La institución a la cual va dirigido el siguiente proyecto es el “PROYECTO COMUNITARIO GUADALQUIVIR” es una organización comunitaria que está afiliada a la organización mundial Chilfund, con sede en Bolivia. *El propósito de esta organización es el de Ayudar a que los niños y niñas carentes, excluidos y vulnerables tengan la oportunidad de convertirse en adultos, jóvenes, padres y madres de familia y líderes que lleven cambios duraderos y positivos a sus hogares o comunidades.*

*El proyecto Comunitario “Guadalquivir”. Se encarga de brindar apoyo a las familias pertenecientes a este programa como ser; apoyo académico, apoyo alimenticio, apoyo psicológico, colaboran en el ambiente de la convivencia familiar, realizan gestiones de apadrinamientos que es una de las bases fundamentales para la sostenibilidad del programa entre otros.*

En la época de modernidad que atravesamos nos vemos envueltos en un entorno donde el uso y desarrollo de las TIC han crecido a pasos agigantados. Haciendo que esta tendencia sea más bien una necesidad en cada organización, empresa, entidad, institución, etc. donde se maneje información. . Esto abre campo a los ingenieros informáticos los cuales tienen varias tareas pero una de las más importantes es de manipular la información de forma sistematizada en otras palabras manejar la información de manera más rápida y sencilla mediante un software.

Las TIC se definen como sistemas tecnológicos mediante los que se recibe, manipula y procesa información, y que facilitan la comunicación entre dos o más interlocutores. Por lo tanto las TIC son algo más que informática y computadoras puesto que funcionan en conexión con otras mediante una red. También son algo más que tecnologías de emisión y difusión (como televisión y radio) puesto que no solo

dan cuenta de la divulgación de la información, sino que además permiten una comunicación interactiva.

*El actual proyecto tiene como fin* Incorporar las TIC's (Tecnología de Información y Comunicación) para mejorar el área administrativa del proyecto "Guadalquivir"

Esto debido a la preocupación creciente del área administrativa sobre el manejo y disociación de la información que se maneja en la institución, creando una deficiencia al momento del manejo y archivado de los mismos. Tomando en cuenta el inmenso crecimiento poblacional en la ciudad, y por consecuencia el mayor manejo de datos de información pertenecientes a la institución

Para lograr el propósito planteado, el proyecto al cual de aquí en adelante llamaremos **Mejoramiento del sistema de gestión de la información del proyecto comunitario "Guadalquivir"** consiste en crear un entorno tecnológico donde se pueda realizar la gestión de los datos usando un sistema informático que permita llevar un control más minucioso y eficaz. De esta forma crear un entorno de trabajo donde suceda la interacción hombre-máquina. Este proyecto facilitará a la organización en las tareas como ser: registro de datos, búsqueda de datos específicos, evaluación de crecimiento, generar reportes, registrar actividades, asistencias, entre otros

### **I.2.2 Descripción y fundamentación del proyecto**

El desarrollo del proyecto **Mejoramiento del sistema de gestión de la información del proyecto comunitario "Guadalquivir"** se orienta al desarrollo de un sistema computarizado para automatizar los datos de las familias que pertenecen a la institución. Debido a que el principal problema que se pudo observar es una deficiencia en el control y acceso de la información. Lo fundamental es coadyuvar a mejorar la administración de información.

El proyecto consta de 2 componentes:

**1. Sistema de gestión automatizado para acceso, control, registro, de información del proyecto Guadalquivir:**

Se trata de desarrollar un Sistema informático que facilitara la manipulación de toda la información de los miembros afiliados al proyecto Guadalquivir. Porque la falta de datos actualizados de los afiliados no permite llevar un buen control de los mismos. Generando pérdida de información, demora en la búsqueda de datos,

La solución planteada con el sistema informático es la de almacenar la información en una base de datos y así evitar la pérdida de información y mejorar la emisión de reportes actualizados y rápidos

Se podrá optimizar el uso de las tecnologías y se podrá tener mejor organización en cuanto al manejo de la información que es de vital importancia a la hora de realizar informes.

**2. Capacitación al personal encargado de la administración de la institución**

Para poder llegar a conocer el funcionamiento del sistema es necesario realizar talleres de socialización del sistema y de las TIC's al personal de la empresa, con el objetivo de que estos tengan un conocimiento básico de lo que son las tecnologías de la información y comunicación y que el buen uso de estas crear situaciones de ventajas para el crecimiento institucional.

El presente proyecto se enmarca en los componentes mencionados anteriormente que contribuyen al propósito general del proyecto que es el de lograr un Sistema de acceso, registro y control de la información mejorado. Se trata de mejorar todo el proceso de gestión de la información con el desarrollo de los componentes definidos anteriormente y así encaminar a la institución en un ambiente tecnológico para la realización de las actividades es decir, incorporar a la tecnología en la administración de la institución. La realización del propósito da lugar a complementar el proyecto

cuyo fin es Incorporar las TIC's para mejorar el área administrativa del proyecto Guadalquivir. Quiere decir que el presente proyecto va ayudar a la institución a involucrarse más con el uso de las TIC's para mejorar todo el proceso de administración de los datos de los afiliados del proyecto Guadalquivir, De esta forma el área administrativa de la institución se verá fortalecida y mejorada en cuanto al manejo de la tecnología en sus procesos de gestión de datos.

a) **Árbol de problemas**

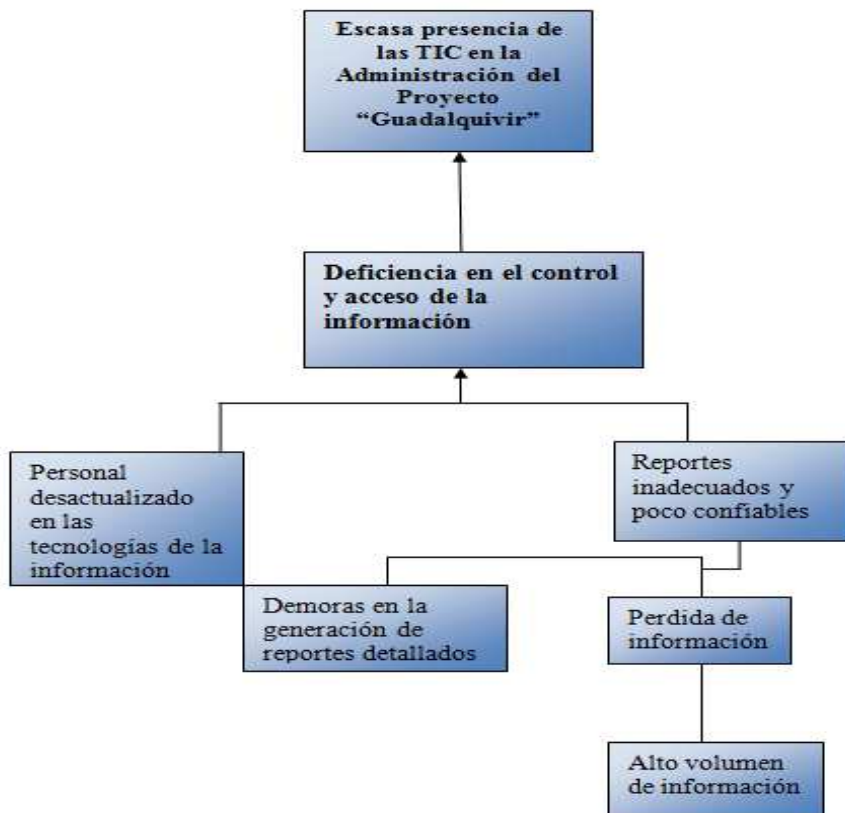
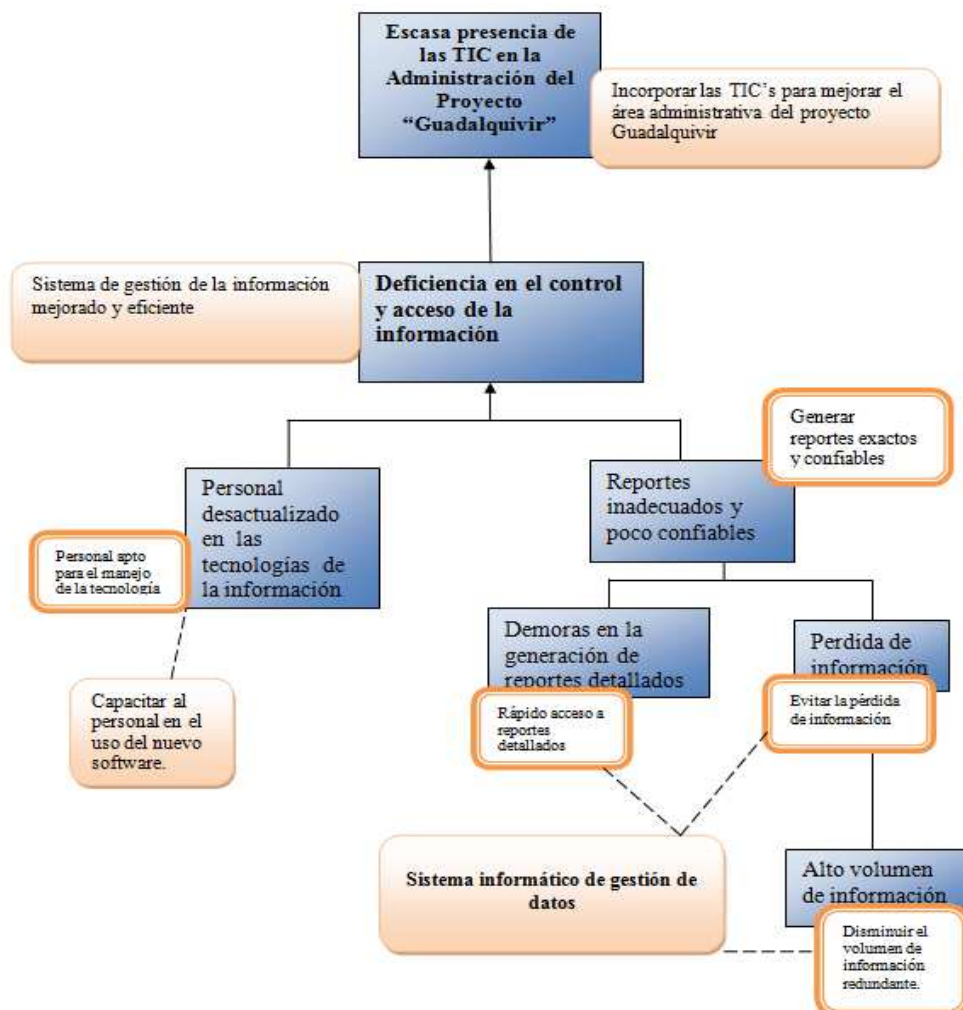


Figura N° 1 Árbol de problemas

b) **Árbol de objetivos**Figura N° 2 **Árbol de objetivos**

**c) Situación planteada con y sin proyecto**

<b>Situación sin proyecto</b>	<b>Situación con proyecto</b>
<p>El Proyecto Guadalquivir no cuenta con ningún sistema automático que contribuya con un mejor manejo de la información.</p> <p>El control se lo hace por medio de tarjetas donde se anota de forma manual la información de cada familia y luego se lo archiva.</p> <p>En una recopilación de información se pueden observar las siguientes falencias:</p> <ul style="list-style-type: none"> <li>• Registro de datos de familias deficiente</li> <li>• Demora en generar reportes para brindar información</li> <li>• Demora en la búsqueda de datos específicos</li> <li>• Dificultad en la identificación de los miembros de la institución</li> <li>• Personal de la institución desactualizado en la TIC's</li> </ul>	<p>El proyecto Guadalquivir cuenta con un sistema de gestión de la información que mejora todo el proceso de registro, control y acceso a la información. Este sistema ayuda mucho para llevar un control más minucioso de datos de los miembros de la institución.</p>

Tabla: 5 Situación planteada con y sin proyecto

### **I.2.3 Objetivos**

#### **Objetivo General:**

Mejorar el Sistema de gestión de la información del proyecto comunitario “Guadalquivir”

#### **Objetivo específico:**

1. Desarrollar un Sistema informático de gestión para acceso, control, registro, de información del proyecto Guadalquivir
2. Capacitar al personal sobre el manejo del **Sistema de gestión**

### **I.2.4 Matriz de Marco lógico del proyecto**

<b>Resumen Narrativo del Proyecto</b>	<b>Indicadores</b>	<b>Medios de Verificación</b>	<b>Supuestos</b>
<p><b>Fin</b></p> <p>Incorporar las TIC's para mejorar el área administrativa del proyecto Guadalquivir</p>	<p>Al finalizar el primer año de ejecutado el proyecto aproximadamente la mitad de las actividades administrativas del Proyecto comunitario Guadalquivir han mejorado gradualmente en cuanto al manejo de su información</p>	<p>Encuestas al personal encargado para verificar la conformidad con el nuevo sistema de manejo de información</p>	<ul style="list-style-type: none"> <li>• La institución brinda toda la información necesaria para la realización del proyecto</li> <li>• Interés de la institución en la ejecución del proyecto</li> </ul>
<p><b>Propósito</b></p> <p>Sistema de acceso, registro y control de la información mejorado</p>	<p>Al finalizar el primer año de ejecutado el proyecto, se ha mejorado el sistema de gestión de la información que se maneja en la institución respecto del año base 2011 donde no existía un sistema computarizado</p>	<p>Informe avalado por la administradora de la institución que certifica que el sistema de manejo de la información esta mejorado</p>	<ul style="list-style-type: none"> <li>• Plena disposición por parte del personal para el manejo y uso del sistema</li> <li>• Recursos económicos para la sostenibilidad del proyecto</li> </ul>

<b>Componentes</b>			
<p>1. Sistema de gestión automatizado para acceso, control, registro, de información del proyecto Guadalquivir</p> <p>2. Capacitación al personal encargado de la administración de la institución</p>	<p>1. En el mes de noviembre del 2013 se concluye el desarrollo del proyecto, automatizando el 70% de las tareas que realiza la administradora de la institución</p> <p>2. A partir del mes de diciembre de 2013 se realizara la capacitación dirigida a 3 personas en el uso del sistema con una duración de 2 horas discontinuas, se obtiene una tasa de asistencia del personal involucrado del 66% (dos personas)</p>	<ul style="list-style-type: none"> <li>• Carta de conformidad realizada por la administradora.</li> <li>• Registro del control de asistencia</li> <li>• Fotografías que avalen el momento de la capacitación</li> <li>• Manual de usuario</li> </ul>	<ul style="list-style-type: none"> <li>• Se cuenta con el material necesario para llevar adelante el desarrollo del sistema de gestión</li> <li>• Apoyo constante de la administradora de la institución para el desarrollo del proyecto</li> <li>• La administración de la institución brinda el apoyo necesario y la autorización para la capacitación del personal en cuanto al manejo y uso del Sistema</li> </ul>

<b>Actividades</b>			
<p data-bbox="360 459 741 703"><b>1. Sistema de gestión automatizado para acceso, control, registro, de información del proyecto Guadalquivir</b></p> <p data-bbox="409 882 741 963">1.1. Análisis y Diseño del Sistema</p> <p data-bbox="409 1142 741 1230">1.2. Determinación de requerimientos</p>	<p data-bbox="757 459 1196 592">El sistema se realizó desde el 10 de marzo hasta el 29 de noviembre de 2013</p>	<p data-bbox="1214 890 1583 1023">Documento de análisis de diseño revisado por el Docente de Taller III.</p> <p data-bbox="1214 1182 1583 1315">Documento de especificación de requerimientos de software</p>	<p data-bbox="1601 459 1968 541">Información disponible sobre las tareas de la institución</p> <p data-bbox="1601 1067 1968 1200">Disponibilidad de equipos de computación para instalar el sistema y la base de datos</p>

<p>1.3. Programación del Sistema</p> <p>1.4. Pruebas del Sistema</p> <p>1.5. Presentación del Sistema</p> <p>1.6. Elaboración de material de apoyo para el uso del software elaborado (<i>Manual de usuario</i>)</p> <p><b>2. Capacitación al personal encargado de la administración de la</b></p>	<p>La capacitación se la realiza en fecha 09 de diciembre, con una duración de 2 horas</p>	<p>Presentación del sistema cumpliendo requisitos establecidos en la fase de análisis y diseño</p>	<p>Existencia y disponibilidad de recursos económicos.</p>
---	--	--	--

<p><b>institución</b></p> <p>2.1. Planificación de actividades para la capacitación en el manejo del Sistema de Gestión</p> <p>2.2. Construcción del programa para la capacitación en el Sistema de Gestión y Control</p>		<p>Documento con especificaciones sobre la capacitación al personal revisado por el docente.</p>	<p>Disponibilidad de tiempo e interés del personal involucrado para la capacitación</p>
---	--	--	---

Tabla: 6 Matriz de marco lógico

### **I.2.5 Metodología de trabajo**

#### **Desarrollo del proyecto**

Se empleará la metodología RUP, esta metodología es un proceso de ingeniería de software que ofrece un enfoque disciplinado para asignar tareas y responsabilidades dentro de la organización del desarrollo.

Solo se abarcará las tres primeras fases que son:

**1ra. Inicio:** En esta fase se realiza la actividad “Determinación de requerimientos”, donde se identifica los requisitos del producto desde la perspectiva del usuario. Se elaboran los casos de uso del negocio para comprender la actual situación los costos dentro del Proyecto “Guadalquivir”, además el Plan de Desarrollo del Proyecto, el calendario de actividades y un cronograma de control de fechas de inicio y entrega de cada documento.

**2da. Elaboración:** Se realiza la actividad “Análisis y Diseño del Sistema”, donde se identifica y especifica los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento de los objetivos.

También se realizarán los diagramas UML (de casos de uso, de secuencia, de estado, de clases) para modelar, especificar y visualizar la interacción entre los actores y los elementos que conforman el sistema al igual que el funcionamiento de cada uno de ellos. Se desarrollarán los prototipos de interfaces de usuario.

La aceptación del diseño representado mediante diagramas, por parte del usuario, marcan el final de esta fase.

**3ra. Construcción:** Se realiza las actividades “Programación del Sistema” y “Pruebas del Sistema”, donde el producto es construido en base a 2 iteraciones, cada una abarcará el 50% de la programación del producto. Se aplican las pruebas correspondientes.

**a) Descripción y relación de las estrategias con los objetivos**

Estrategias	Objetivos Específicos
<p>Recopilación de requerimientos.</p> <p>Análisis de toda la información recolectada, para diseñar una solución objetiva.</p> <p>Programación de todo el diseñado.</p> <p>Pruebas para garantizar la calidad del sistema.</p> <p>Documentación del manual de usuario.</p> <p>Elaboración del material de capacitación.</p> <p>Impartir curso de capacitación al personal.</p>	<p>1. Desarrollar un Sistema de gestión automatizado para acceso, control, registro, de información del proyecto Guadalquivir</p> <p>2. Capacitar al personal sobre el manejo del <b>Sistema de gestión</b></p>

Tabla: 7 Descripción y relación de las estrategias con los objetivos

**I.2.6 Resultados esperados**

Con el desarrollo del proyecto se espera:

- Mejorar y organizar todo el proceso de registro de información en la institución
- El sistema podrá facilitar el acceso a la información que se maneja en la institución

Concientizar al personal sobre la importancia del uso de las tecnologías para el manejo de información

## **Capítulo II: Componentes**

### **Componente I**

#### **I.1 Sistema de gestión automatizado para registro y control de información del proyecto Guadalquivir:**

##### **I.1.1 Introducción**

La adopción de nuevas tecnologías en el sector académico, especialmente en las universidades estuvo asociada en sus inicios, a incipientes programas de informática que resultaron poco significativos. Dada la cantidad de información que entra, se procesa y genera, se tienen que realizar acciones entre las que se puede destacar: la incorporación de nuevas tecnologías (redes, multimedia, internet) y el desarrollo de modernos sistemas de información con el propósito de superar la situación planteada.

Este documento provee una visión global del enfoque de desarrollo propuesto, muestra una versión preliminar preparada para ser incluida en la propuesta al sistema a elaborar como respuesta al proyecto de la asignatura de Taller III de la Carrera de Ingeniería Informática de la Facultad de Ciencias y Tecnología de la Universidad Autónoma “Juan Misael Saracho”.

El proyecto ha sido ofertado por el universitario Omar A. Ibáñez Ibáñez, basado en la metodología de Rational Unified Process (RUP) en la que únicamente se procederá a cumplir con las tres primeras fases que marca la metodología, constando únicamente en la tercera fase de dos iteraciones. Es importante destacar esto puesto que utilizaremos la terminología RUP, de acuerdo a las características del proyecto, seleccionaremos los roles de los participantes, las actividades a realizar y los artefactos que serán generados. Se incluirá el detalle para las fases de Inicio, Elaboración y adicionalmente se realizaran las fases posteriores de Construcción para dar una visión global de todo proceso.

El software desarrollado para el proyecto “mejoramiento del sistema de gestión de la información del proyecto comunitario “Guadalquivir””. Tiene como nombre

“Sistema de gestión automatizado para acceso, control, registro, de información”  
**SARCI.**

**SARCI** es un sistema de gestión que utiliza el modelo de software incremental ya que en su etapa de construcción el sistema se divide en procesos que fueron desarrollados jerárquicamente, este modelo nos permite utilizar datos de los primeros procesos para desarrollar los posteriores

#### **I.1.1.1 Objetivos**

##### **General:**

Desarrollar un sistema que realice el control y registro de las familias afiliadas al “PROYECTO GUADALQUIVIR”

##### **Específicos:**

- Diseñar un entorno del sistema dinámico fácil y amigable para la presentación de la información, utilizando tecnologías existentes.
- Gestionar y automatizar los procesos de usuarios, familias, afiliados, patrocinadores y actividades que se desarrollan en el “Proyecto GUADALQUIVIR”.
- Diseñar una base de datos Orientada al modelo Entidad Relación con Postgres 8.4
- Utilizar el lenguaje orientado a objetos java como lenguaje de programación.
- Aplicar metodologías RUP, utilizando herramientas case para su desarrollo y documentación, para realizar análisis y diseño del sistema (UML).

#### **I.1.1.2 Alcances y limitaciones**

##### **Alcances:**

El sistema contemplara los siguientes aspectos:

- Controlar y organizar el alto índice de información que se maneja en la institución
- Controlar y evaluar un listado previo de familias candidatas a ser afiliadas al proyecto
- Prevenir el desordenamiento y pérdida de datos
- Controlar y hacer seguimiento de los patrocinadores que apoyan a niños del proyecto
- Registrar al personal para manejar el sistema y los procesos que conlleva.
- Registrar a las familias que pertenecen y a las familias que quieren ser parte del “Proyecto Guadalquivir”.
- Mantener información completa y actualizada de cada familia
- Registrar a los niños afiliados con sus respectivos patrocinadores
- Registrar a los patrocinadores
- Registrar y programar actividades
- Realizar informes generales con los datos almacenados en el sistema

**Limitaciones:**

El sistema no contemplará lo siguiente:

- El sistema será desarrollado bajo plataforma Windows.
- El Sistema no incluirá la Gestión de Inventarios.
- El sistema no incluirá la Gestión Contable.
- El sistema será restringido, solo usuarios autorizados podrán acceder al sistema.
- Ingresos y Egresos.
- Balance.
- No contempla la parte financiera.

### **I.1.1.3 Suposiciones y restricciones**

#### ***I.1.1.3.1 Suposiciones***

- Equipo de computación. En el que incluye un computador, impresora. Para la implementación del sistema.
- Disponibilidad económica
- La disponibilidad de otorgar toda la información necesaria para la creación del software.
- Personal con conocimientos básicos de computación.

#### ***I.1.1.3.2 Restricciones***

Como restricciones el proyecto establece las siguientes:

- Tiempo de ejecución variable según el desarrollo de la materia
- Restricción de acceso al sistema mediante uso de claves
- Restricción de información a personal no autorizado

### **I.1.1.4 Entregables del proyecto**

A continuación se indican y describen cada uno de los artefactos que serán generados y utilizados por el proyecto, que constituyen los entregables. Esta lista constituye la configuración de RUP desde la perspectiva de artefactos, y que proponemos para este proyecto.

Es preciso destacar que de acuerdo a la metodología RUP (y de todo proceso iterativo e incremental), todos los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, solo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada

iteración y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos. Esto será indicado más adelante cuando se presenten los objetivos de cada iteración.

### **1) Plan de Desarrollo del Software**

Es el presente documento.

### **2) Visión**

Este documento define la visión del producto desde la perspectiva del Cliente, especificando las necesidades y características del producto. Constituye una base de acuerdo en cuanto a los requisitos del sistema

### **3) Glosario**

Es un documento que define los principales términos usados en el proyecto. Permite establecer una terminología consensuada.

### **4) Modelo de Casos de Uso del Negocio**

Es un modelo de las funciones de negocio vistas desde la perspectiva de los actores externos (agentes de registro, solicitantes finales, otros sistemas etc.) permite situar al sistema en el contexto organizacional haciendo énfasis en los objetivos en este ámbito. Este modelo se representa con un Diagrama de Casos de Uso usando estereotipos específicos para este modelo.

### **5) Requerimientos funcionales**

Sección donde se describen todos los requerimientos funcionales del sistema, los servicios y la funcionalidad que se espera

### **6) Modelo de Objetos del Negocio**

Es un modelo que describe la realización de cada caso de uso del negocio, estableciendo los actores internos, la información que en términos generales manipulan y los flujos de trabajo (workflows) asociados al caso de uso del negocio.

## **7) Modelo de Casos de Uso**

El modelo de Casos de Uso presenta las funciones del sistema y los actores que hacen uso de ellas. Se representa mediante Diagramas de Casos de Uso.

## **8) Especificación de los Casos de Uso**

Para los casos de usos que lo requieran (cuya funcionalidad no sea evidente o que no baste con una simple descripción narrativa) se realiza una descripción detallada utilizando una plantilla de documento, donde se incluye: pre-condiciones, post-condiciones y flujo de eventos.

## **9) Modelo de Análisis y Diseño**

Este modelo establece la realización de los casos de uso en clases y pasando desde una representación en términos de análisis (sin incluir aspectos de implementación) hacia una de diseño (incluyendo una orientación hacia el entorno de implementación), de acuerdo al avance del proyecto.

## **10) Modelo de Datos**

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Clases.

## **11) Diseño de Pantallas de Usuario**

Se trata de prototipos que permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como: dibujos a mano en papel, dibujos con alguna herramienta gráfica o prototipos ejecutables interactivos, siguiendo ese orden de acuerdo al avance del proyecto. Sólo los de este último tipo serán entregados al final de la fase de Elaboración, los otros

serán desechados. Asimismo, este artefacto será desechado en la fase de Construcción en la medida que el resultado de las iteraciones vayan desarrollando el producto final.

### **12) Modelo de Implementación**

Es una colección de componentes y los subsistemas que los contienen.

Los componentes incluyen, ficheros ejecutables, código fuente, y todo lo necesario para la implementación y despliegue del sistema.

### **13) Modelo de Despliegue**

Este modelo muestra el despliegue la configuración de tipos de nodos del sistema, en los cuales se hará el despliegue de los componentes.

### **14) Casos de Prueba**

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de las pruebas y los resultados esperados

### **15) Manual de Instalación**

Este documento incluye las instrucciones para realizar la instalación del producto.

### **16) Material de Apoyo al Usuario Final**

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Manual de Usuario.

### **17) Producto**

Los ficheros del producto empaquetados y almacenadas en un disco con los mecanismos apropiados para facilitar su instalación. El producto, a partir de la primera iteración de la fase de Construcción es desarrollado incremental e iterativamente.

#### ***1.1.1.4.1 Organización del proyecto***

##### ***1.1.1.4.1.1 Participantes del proyecto***

La institución todavía no incluye a un personal como responsable del proyecto, solo se cuenta con personal que se estime conveniente para proporcionar los requisitos y usar el sistema

El resto del personal encargado del desarrollo general del proyecto está formado por los siguientes puestos de trabajo.

**Jefe de Proyecto:** Univ. Omar Anheló Ibáñez Ibáñez, estudiante de último semestre de la carrera de ingeniería informática de la Facultad de Ciencias y Tecnología de la UAJMS.

Con una moderada experiencia en metodologías de desarrollo, en particular la notación UML y el proceso de desarrollo RUP.

**Analista de Sistemas:** El perfil establecido pertenece al universitario Omar Anheló Ibáñez Ibáñez con conocimientos básicos de UML, con experiencia en sistemas similares al a línea del proyecto.

**Programador:** Univ. Omar Anheló Ibáñez Ibáñez con experiencia en el desarrollo de prototipos afines al proyecto

**Ingeniero de Software:** El perfil estará a cargo del Univ. Omar Anheló Ibáñez Ibáñez, que participara realizando labores de gestión de requisitos, configuración, diseño de datos, desarrollo de pruebas funcionales entre otros.

##### ***1.1.1.4.1.2 Interfaces Externas***

La institución definirá los participantes del proyecto que proporcionara los requisitos e información del sistema para el desarrollo del mismo.

El equipo de desarrollo interactuara periódicamente con los participantes para especificación y validación del producto.

### ***1.1.1.4.1.3 Roles y Responsabilidades***

Descripción general de las principales responsabilidades de cada uno de los cargos en el equipo de desarrollo durante las fases de inicio y elaboración de acuerdo con los roles que desempeñan RUP.

<b>Responsables</b>	<b>Actividades</b>
<b>Jefe de Proyecto</b>	<ul style="list-style-type: none"><li>• Planificar y controlar el calendario del proyecto.</li><li>• Seguimiento del cronograma y cumplimiento de las actividades</li><li>• Árbol de Problemas y Objetivos</li><li>• Asignación de recursos para las distintas actividades planificadas para el desarrollo del proyecto</li><li>• Administrar el desarrollo del proyecto.</li><li>• Asegurar la calidad e integridad el proyecto</li><li>• Elaborar Manual de Usuario para los involucrados</li><li>• Capacitar a los involucrados con el sistema en el correcto manejo del mismo.</li></ul>
<b>Analista y Diseñador</b>	<ul style="list-style-type: none"><li>• Recolección, especificación y validación de requisitos, tratando los mismos con el cliente y los usuarios.</li><li>• Establecer la arquitectura del sistema.</li></ul>

<b>de Sistemas</b>	<ul style="list-style-type: none"> <li>• Elaborar los diferentes modelados de Análisis y Diseño.</li> <li>• Colaborar en las pruebas de funcionalidad del sistema.</li> </ul>
<b>Programador</b>	<ul style="list-style-type: none"> <li>• Construcción del sistema de información.</li> <li>• Colaboración en la elaboración de las pruebas funcionales.</li> <li>• Colaboración de las validaciones con el usuario.</li> </ul>
<b>Ingeniero de Software</b>	<p>El Ingeniero de Software realizara las siguientes actividades:</p> <ul style="list-style-type: none"> <li>• Gestión de requisitos, gestión de configuración y cambios.</li> <li>• Preparación de las pruebas funcionales.</li> </ul>

Tabla: 1Roles y Responsabilidades

***1.1.1.4.2 Gestión del Proceso***

***1.1.1.4.2.1 Presupuesto /Justificación***

<b>ITEM</b>	<b>RUBROS</b>	<b>Aporte Universidad</b>	<b>Otro Aporte</b>	<b>TOTAL (Bs.)</b>
<b>20000</b>	<b>SERVICIOS NO PERSONALES</b>			
	<b>21000. Servicios Básicos</b>			
	<b>22000. Servicios de transporte</b>		60.-	
	<b>23000. Alquileres</b>			
	<b>24000. Mantenimiento y reparación</b>		250.-	
	<b>25000. Servicios Profesionales y Comerciales</b>			
	<b>Sub total rubro</b>		310.-	310.-
<b>30000</b>	<b>MATERIALES Y SUMINISTROS</b>			
	<b>32000. Productos de Papel, Cartón e Impresos</b>		800.-	
	<b>39000. Productos Varios.</b>		50.-	
	<b>Sub total rubro</b>			850.-

	<b>TOTAL</b>			1160.-
	<b>TOTAL + 40% Incentivo</b>			1584.-

**a) SUB GRUPO 22000. Descripción de los gastos de viajes y transporte de personal**

<b>Partida</b>	<b>Personal</b>	<b>Lugar</b>	<b>N° de viajes</b>	<b>Costo unitario*</b>	<b>Costo total</b>
22100	Pasajes	Ciudad de Tarija	60	1	60.-
<b>Total</b>					60.-

**b) SUB GRUPO 24000. Descripción mantenimiento y reparación**

<b>Partida</b>	<b>Mantenimiento y reparación de equipo y maquinaria</b>	<b>Costo unitario</b>	<b>Tiempo mes</b>	<b>Costo total</b>
24100	Mantenimiento y Reparación de Edificios y Equipos			
24300	Otros Gastos por Mantenimiento y Reparación	250.-		
<b>Total</b>				250.-

**c) SUB GRUPO 32000. Descripción del gasto de Productos de Papel, Cartón e Impresos**

<b>Partida</b>	<b>Tipo de material *</b>	<b>Cantidad</b>	<b>Costo/Unidad</b>	<b>Total</b>
32100	Papel de Escritorio	8	98	784
32200	Productos de Artes Gráficas, Papel y Cartón	4	4	16
32300	Libros y Revistas			
32400	Textos de Enseñanza			
32500	Periódicos			
<b>Total</b>				800

\* Se refiere a la adquisición de; papel y cartón en sus diversas formas y clases, impresos y publicaciones, periódicos, revistas, libros, fotocopias, etc.

**b) SUB GRUPO 39000. Productos varios**

<b>Partida</b>	<b>Mantenimiento y reparación de equipo y maquinaria</b>	<b>Costo unitario</b>	<b>Costo total</b>
3910	Productos varios	50.-	50.-
24300	Otros Gastos por Mantenimiento y Reparación		

<b>Total</b>		50.-
--------------	--	------

#### ***I.1.1.4.2.2 Plan del Proyecto***

Representación de la organización en fases e iteraciones, también se presenta el calendario de presentaciones

#### ***II.1.1.4.2.5.1 Plan de las fases***

El desarrollo se llevará a cabo en base a fases con una o más iteraciones en cada una de ellas. La siguiente tabla muestra una la distribución de tiempos y el número de iteraciones de cada fase (para las fases de Construcción y Transición es sólo una aproximación muy preliminar)

<b>Fase</b>	<b>Nro. Iteraciones</b>	<b>Duración Iteraciones</b>	<b>Duración Total</b>
Fase de Inicio	1	3 semanas	03 marzo – 24 marzo
Fase de Elaboración	1	8 semanas	18 mayo -17 julio
Fase de Construcción	2	16 semanas	20 julio – 20 noviembre

Fase de Transición	2	1 semana	21 noviembre – 29 noviembre
--------------------	---	----------	--------------------------------

Tabla: 2 Plan de Fases

En la siguiente tabla se describen los hitos que marcan el final de cada fase

Descripción	Hito
Fase de Inicio	En esta fase desarrollarán los requisitos del producto desde la perspectiva del usuario, los cuales serán establecidos en el artefacto Visión. Los principales casos de uso serán identificados y se hará un refinamiento del Plan de Desarrollo del Proyecto. La aceptación del cliente /usuario del artefacto Visión y el Plan de Desarrollo marcan el final de esta fase.
Fase de Elaboración	En esta fase se analizan los requisitos y se desarrolla un prototipo de arquitectura (incluyendo las partes más relevantes y / o críticas del sistema). Al final de esta fase, todos los casos de uso correspondientes a requisitos que serán implementados en la primera release de la fase de Construcción deben estar analizados y diseñados (en el Modelo de Análisis / Diseño). La revisión y aceptación del prototipo de la arquitectura del sistema marca el final de esta fase. En nuestro caso particular, por no incluirse las fases siguientes, la revisión y entrega de todos los artefactos hasta este punto de desarrollo también se incluye como

	<p>hito. La primera iteración tendrá como objetivo la identificación y especificación de los principales casos de uso, así como su realización preliminar en el Modelo de Análisis / Diseño, también permitirá hacer una revisión general del estado de los artefactos hasta este punto y ajustar si es necesario la planificación para asegurar el cumplimiento.</p>
Fase de Construcción	<p>En la fase de construcción se terminan de analizar y diseñar todos los casos de uso, refinando el modelo de análisis y diseño. El producto se construye en base a 2 iteraciones, cada una produciendo una release a la cual se le aplican las pruebas y se valida con el cliente y usuario. Se comienza la elaboración de material de apoyo al usuario. El hito que marca el fin de esta fase es la versión de la release 2.0, con la capacidad operacional parcial del producto que se haya considerado como crítica, lista para ser entregada a los usuarios para pruebas beta.</p>
Fase de Transición	<p>El hito que marca el fin de esta fase incluye la entrega de toda la documentación del proyecto y todo el material de apoyo al usuario, también contempla la capacitación realizada a los usuarios y la defensa final del proyecto realizado.</p>

Tabla: 3 Descripción de fases

### ***1.1.1.4.2.3 Calendario del Proyecto***

Un calendario que contiene las principales tareas del proyecto incluyendo las fases de inicio y elaboración.

Como se ha comentado, el proceso iterativo e incremental de RUP está caracterizado por la realización en paralelo de todas las disciplinas de desarrollo a lo largo del proyecto, con lo cual la mayoría de los artefactos son generados muy tempranamente en el proyecto pero van desarrollándose en mayor o menor grado de acuerdo a la fase e iteración del proyecto. La siguiente figura muestra este enfoque marcando de manera sombreada el énfasis de cada disciplina en un momento determinado del proceso de desarrollo.

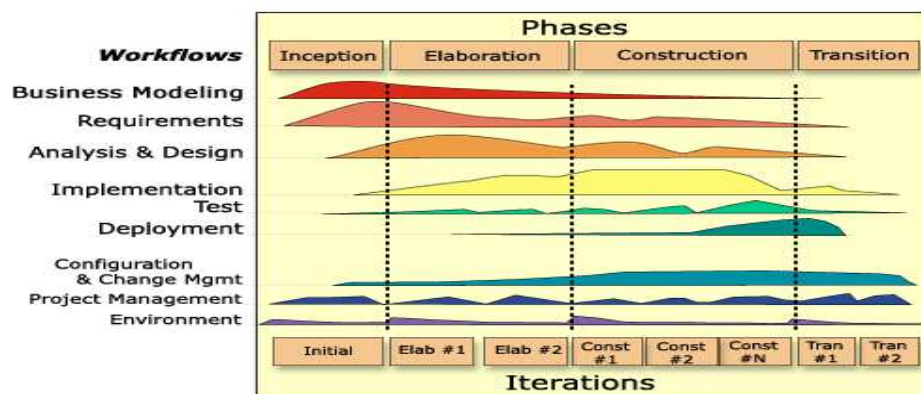


Figura N° 1 Metodología RUP

A continuación se muestra un calendario que fue establecido para la realización de este proyecto, donde la fecha de aprobación de la materia indica cuando el producto pasara a un estado de completitud suficiente para someterse a una revisión y aprobación, pero esta restricción no quita la posibilidad de su posterior cambio o mejora.

<b>Disciplinas / Artefactos generados o modificados durante la Fase de Elaboración</b>	<b>Comienzo</b>	<b>Aprobación</b>
<b>Modelado del Negocio</b>		
Modelo de Casos de Uso del Negocio y Modelo de Objetos del Negocio	25-marzo	
<b>Requisitos</b>		
Glosario	13-abril	
Visión	15- abril	
Modelo de Casos de Uso	15- abril	
<b>Análisis / Diseño</b>		
Modelo de Análisis / Diseño	25-mayo	
Modelo de Datos	06-julio	
<b>Implementación</b>		
Prototipos de Interfaces de Usuario	28-agosto	
Modelo de Implementación	31-septiembre	
<b>Gestión de Cambios y Configuración</b>	Durante todo el proyecto	
<b>Gestión del proyecto</b>		

Plan de Desarrollo del Software en su versión 1.0	29 noviembre	
<b>Ambiente</b>	Durante todo el proyecto	

Tabla: 4 Disciplinas / Artefactos Fase de Inicio



<b>2.2</b>	Realizar cronograma de capacitación.	<b>2</b>										<b>X</b>
<b>2.3</b>	Realizar capacitación a los Usuarios	<b>1</b>										<b>X</b>

Tabla: 5 Cronograma de Actividades

### ***1.1.1.5.1.1 Seguimiento y control del proyecto***

#### **Control de Plazos**

<b>Fase</b>	<b>Nº de Iteración</b>	<b>Inicio</b>
<b>Inicio</b>	Primera	25 - marzo
<b>Elaboración</b>	Primera	29 – mayo
<b>Construcción</b>	Primera	27 – julio
<b>Transición</b>	Primera	29 - noviembre

Tabla: 6 Control de Plazos

### ***1.1.1.5.1.2 Control de Calidad***

#### **• Introducción**

El control de calidad es un modelo planeado y sistemático de todas las acciones necesarias para proporcionar la confianza de que el artículo o producto se ajuste a los requisitos técnicos establecidos.

La preparación de un plan de control de calidad del software para cada proyecto de software es una de las principales responsabilidades del grupo de control de calidad del software.

El control de calidad realizará las siguientes funciones:

Durante el análisis y diseño, se presentaran un plan de verificación del software y un plan de prueba de aceptación. El plan de verificación describe los métodos que se ocuparán para revisar que los documentos de diseño satisfagan los requisitos, y que el código fuente sea consistente con las especificaciones de requisitos y con la documentación del diseño.

El plan de prueba del código fuente es un componente importante del plan de verificación del software.

El plan de prueba de aceptación incluye casos de prueba, resultados esperados y capacidades demostradas por cada caso de prueba. A menudo, el personal de control de calidad trabajará con el cliente para desarrollar un solo plan de prueba de aceptación. En otros casos el cliente desarrollará un plan de prueba de aceptación independiente del plan de control de calidad. De cualquier forma, el personal de control de calidad debe desarrollar un plan de prueba de aceptación doméstico.

Al terminar los planes de verificación y de aceptación se realizará una revisión de verificación del software para evaluar cuan adecuados son los planes.

Durante la evolución del producto, se realizarán auditorías en el proceso para verificar que los productos de trabajo sean consistentes y estén completos.

Los elementos que sufrirán auditoría por consistencia incluyen especificaciones de interfaces para hardware, software y personas; diseño interno contra especificaciones funcionales; código fuente contra documentación. En la práctica, sólo ciertas porciones críticas del Sistema pueden someterse a auditorías intensivas.

Antes de la entrega del Sistema, se realiza una auditoría funcional y una auditoría física.

La primera reconfirma el cumplimiento de todos los requisitos. La auditoría física verifica que el código fuente y todos los documentos asociados estén completos, sean consistentes tanto internamente, como uno con otro, y que estén listos para enviarse. El resumen de verificación del software se prepara para describir los resultados de todas las revisiones, auditorías y pruebas efectuadas por el personal de control de calidad, a través del ciclo de desarrollo.

Dicho personal, a veces se encarga de los acuerdos para los recorridos, inspecciones, y revisiones de logros principales. Además, el personal de control de calidad conduce el proyecto póstumo, escribe es documento del legado del proyecto, y proporciona una custodia a largo plazo de los registros del proyecto.

El grupo de control de calidad trabajará con el grupo de desarrollo para obtener el plan de pruebas y el código fuente, que especifica los objetivos de las pruebas, los criterios para la terminación de las pruebas, el plan de integración del Sistema, los métodos que se usaran en módulos particulares, además, entradas de prueba particulares y resultados esperados.

El plan de pruebas de código fuente tiene cuatro tipos de pruebas: pruebas de función, de desempeño, de tensión, y estructuradas. Las dos primeras se basan en las especificaciones de requisitos y se diseñaron para demostrar que el Sistema satisface sus requisitos los que a su vez se deben redactar en términos cuantificables y que se puedan probar.

Los casos de prueba funcional especifican condiciones operativas comunes valores de entradas comunes y resultados esperados comunes, también prueban el comportamiento dentro, sobre, y más allá de las fronteras funcionales.

Las pruebas de desempeño se proyectan para verificar el tiempo de respuesta, rendimiento, la utilización de memorias primarias y secundarias y las tasas de tráfico en los canales de datos y los enlaces de comunicación.

Las pruebas de tensión se diseñan para sobrecargar un Sistema de varias maneras.

Las pruebas de estructura se relacionan con el examen de la lógica interna de procesamiento de un Sistema de software. Las rutinas particulares llamadas y las rutas lógicas recorridas a lo largo de las rutinas son los objetos importantes.

- **Propósito**

- Detectar problemas.
- Delimitar el área problemática.
- Estimar factores que probablemente provoquen el problema.
- Determinar si el efecto tomado como problema es verdadero o no.

- Prevenir errores debido a omisión, rapidez o descuido.
- Confirmar los efectos de mejora.
- Detectar desfases.
- Realizar pruebas en cada versión.
- **Objetivos**
  - Aumentar la satisfacción del cliente.
  - Equilibrar el esfuerzo en múltiples demandas.
  - Obtener el mejor producto.
  - Disfrutar de una ventaja competitiva.
  - Disponer de métricas objetivas de valoración.
  - Ahorrar tiempo y dinero.

## I.1.2 Definiciones Acrónimos y Abreviaturas

### I.1.2.1 Definiciones

<b>Institución/proyecto</b>	Edificio donde se instalan las dependencias del proyecto comunitario “Guadalquivir”
<b>Usuario</b>	Persona que va interactuar de forma directa con el sistema
<b>Afiliado</b>	Persona que proporcionara información para almacenarla en la base de datos del sistema

Tabla: 7 Definiciones

### I.1.2.2 Acrónimos

<b>BBDD, BD</b>	Base de datos, Base de datos.
<b>SARCI</b>	Sistema de Registro y Control de la Información
<b>TIC</b>	Tecnologías de la Información y Comunicación.
<b>RUP</b>	Proceso Unificado Racional
<b>UML</b>	Lenguaje Unificado de Modelado

Tabla: 8 Acrónimos

### I.1.3 Requerimientos funcionales

#### I.1.3.1 Interfaces de Usuario

<b>ID del Requerimiento</b>	<b>Req (01)</b>
<b>Nombre del Requerimiento</b>	Interfaz
<b>Característica</b>	La interfaz de usuario debe ser amigable y de fácil manejo para el usuario. Orientada a formularios accesibles desde un navegador web
<b>Prioridad:</b> Alta	

Tabla: 9 Interfaz usuario

#### I.1.3.2 Interfaces Hardware

No son necesarias.

### I.1.3.3 Interfaces Software

<b>ID del Requerimiento</b>	<b>Req (02)</b>
<b>Nombre del Requerimiento</b>	Interfaz de Software
<b>Característica</b>	Por el momento, no habrá ninguna interfaz software con sistemas externos, solo se tendrá enlaces hacia paginas relacionadas y de contenido similar.
<b>Prioridad:</b> Media	

Tabla: 10 Interfaz software

### I.1.3.4 Interfaces de Comunicación

<b>ID del Requerimiento</b>	<b>Req (03)</b>
<b>Nombre del Requerimiento</b>	Interfaz de Comunicación
<b>Característica</b>	Uso de la red e infraestructura proporcionada por la institución.
<b>Prioridad:</b> Media	

Tabla: 11 Interfaz de Comunicación

### I.1.3.5 Requisitos funcionales

SARCI es un sistema de gestión que interactuara con el usuario a través de una interfaz entendible y de fácil manejo. A continuación se describen las funcionalidades que debe proporcionar el sistema.

<b>ID del Requerimiento</b>	<b>Req (04)</b>
<b>Nombre del Requerimiento</b>	Validación
<b>Característica</b>	El sistema controlara el acceso al sistema solo para usuarios autorizados y con roles asignados por el administrador.
<b>Prioridad:</b> Alta	

Tabla: 12 Req. Validación

<b>ID del Requerimiento</b>	<b>Req (05)</b>
<b>Nombre del Requerimiento</b>	Presentación
<b>Característica</b>	El sistema mostrará a los usuarios los datos y atributos de cada personal.
<b>Prioridad:</b> Alta	

Tabla: 13 Req. Presentación

<b>ID del Requerimiento</b>	<b>Req (06)</b>
<b>Nombre del Requerimiento</b>	<b>Registrar usuario</b>
<b>Característica</b>	El sistema permitirá adicionar usuarios y asignar un rol
<b>Prioridad:</b> Alta	

Tabla: 14 Req. Usuario

<b>ID del Requerimiento</b>	<b>Req (07)</b>
<b>Nombre del Requerimiento</b>	<b>presentación del menú</b>
<b>Característica</b>	el sistema mostrara a los usuarios un menú contemplado de acuerdo al rol que este tenga con los procesos que sean necesarios
<b>Prioridad:</b> Alta	

Tabla: 15Presentación Menú

**Req (08) Actualizar datos:** prioridad: media. El usuario del sistema podrá adicionar modificar, borrar datos siempre y cuando su rol se lo permita.

**Req (09) almacenamiento de datos.** Prioridad: alta. El sistema almacenara toda la información registrada en una base de datos

**Req (10).-Simplicidad y facilidad de manejo:** El sistema ofrecerá al usuario un entorno simple y fácil para su manejo.

#### **Cargos de usuario del sistema**

**Req (11).- cargos y funciones:** Descripción de los cargos y funciones se describe en la tabla siguiente

Tipo de usuario	Administrador (dirección)
Habilidades	Conocimientos de PC básicos
Funciones	1. Ingreso a los datos de personal 2. Planificar y tomar decisiones de acuerdo al resultado de los

	<p>reportes</p> <p>3. Podrá tener acceso a la mayor parte del sistema.</p>
--	--

Tipo de usuario	Usuario (secretaria, educador)
Habilidades	Conocimientos de PC básicos
Funciones	<ol style="list-style-type: none"> <li>1. Ingreso a los datos de personal</li> <li>2. Planificar y tomar decisiones de acuerdo al resultado de los reportes junto con el administrador</li> <li>3. Podrá tener acceso al sector de actividades y control de asistencia</li> <li>4. También podrá tener acceso a las listas de familias, afiliados, patrocinadores</li> <li>5. no puede eliminar datos sin consentimiento del administrador</li> </ol>

Tabla: 16Cargos y funciones

### **Gestión de familias**

**Req (12) consulta registrar.** Prioridad: alta el sistema permite adicionar datos de familias mediante formularios

**Req (13) consulta. Presentación.** Prioridad: media. El sistema mostrara al usuario una lista detallada y dinámica de las familias registradas

**Req (14) consulta. Actualizar.** Prioridad: media. El sistema permite realizar la actualización de datos de las familias registradas mediante listados y permitiendo dar de altas y bajas.

#### **Gestión familias candidatas**

**Req (15) consulta registrar y listar.** Prioridad: baja. El sistema permite adicionar datos de familias candidatas a ingresar al proyecto y también permite ver un listado

#### **Gestionar afiliado**

**Req (16) consulta registrar.** Prioridad: alta el sistema permite adicionar datos de cada miembro de las familias es decir de todos los afiliados a la institución mediante llenado de formularios

**Req (17) consulta. Listar.** Prioridad: media. El sistema mostrara al usuario una lista detallada y dinámica de los afiliados registrados

**Req (18) consulta. Actualizar.** Prioridad: media. El sistema permite realizar la actualización de datos (adicionar, borrar, modificar).

#### **Control de asistencia a actividades**

**Req (19) consulta registrar.** Prioridad: alta. el sistema permite registrar la asistencia de las familias o afiliados a las actividades programadas por la institución.

#### **Gestionar patrocinador**

**Req (20) consulta registrar.** Prioridad: alta el sistema permite adicionar datos de los patrocinadores que brindan apoyo a la institución. Para el registro de estos datos se usaran formularios

**Req (21) consulta. Listar.** Prioridad: media. El sistema mostrara al usuario una lista detallada y dinámica de los patrocinadores registrados

**Req (22) consulta. Actualizar.** Prioridad: media. El sistema permite realizar la actualización de datos (adicionar, borrar, modificar).

### **Gestionar actividades**

**Req (23) consulta registrar.** Prioridad: alta el sistema permite registrar las actividades a realizar por la institución

**Req (24) consulta. Listar.** Prioridad: media. El sistema mostrara al usuario una lista detallada y dinámica de las actividades registradas

**Req (25) consulta. Actualizar.** Prioridad: media. El sistema permite realizar la actualización de datos (adicionar, borrar, modificar).

### **Requisitos no funcionales**

#### **Requisitos de rendimiento**

**Req (28)** el tiempo de respuesta de las consultas debe ser el esperable en cualquier aplicación web. Comparándola con un entorno de una aplicación *online*.

**Req (29)** generación de reportes deben ser exactos para poder tomarlos en cuenta en la administración de la institución debido a que son de gran importancia para hacer informes finales.

**Req(30)** capacidad de almacenamiento de la base de datos debe ser superior a 5000 registros

#### **Restricciones de Diseño**

**Req (31)** para el desarrollo del sistema SARCI se usara la metodología RUP de manera que se puedan incorporar fácilmente cambios y nuevas funciones.

#### **Requisitos Tecnológicos**

**Req (32):** En cuanto a la infraestructura informática necesaria se requiere un servidor web con un Sistema de Gestión de Base de Datos.

## **Atributos del Sistema**

### **Fiabilidad**

**Req(33):** Cualquier tipo de ingreso al sistema SARCI deberá ser exitosamente realizada previamente ingresando correctamente su nombre de usuario y contraseña.

### **Disponibilidad**

**Req (34):** el sistema SARCI se encontrara disponible en los equipos de la oficina central de la institución. Disponible solo para usuarios autorizados

### **Seguridad**

**Req (35)** solo se podrá ingresar al sistema mediante la validación de usuario y clave

**Req (36)** el sistema tendrá distintos tipos de usuario ingresando cada uno con su clave correspondiente y realizando las funciones que le corresponden

**Req (37)** se registrara la fecha y hora de los ingresos al sistema.

### **Mantenimiento**

**Req (38):** El sistema requerirá el mínimo posible de mantenimiento y, siempre que sea posible.

### **Otros Requisitos**

**Req (39):** En la base de datos se mantendrá información de toda la información que maneja la institución.

## **I.1.3.6 Glosario**

### ***I.1.3.6.1 Introducción***

Este documento recoge todos y cada uno de los términos manejados a lo largo de todo el Proyecto. Se trata de un diccionario informal de datos y definiciones de la

nomenclatura que se maneja, de tal modo que se crea un estándar para todo el proyecto.

#### ***1.1.3.6.2 Propósito***

El propósito de este glosario es definir con exactitud y sin ambigüedad la terminología manejada en el proyecto. También sirve como guía de consulta para la clarificación de los puntos conflictivos o poco esclarecedores del proyecto, para aclarar algunas definiciones de Términos que se manejan en el Proyecto y que pueden ser desconocidos por el lector.

#### ***1.1.3.6.3 Alcance***

El alcance del presente documento se extiende a todos los componentes del proyecto.

#### ***1.1.3.6.4 Organización del glosario***

El presente documento está organizado por definiciones de términos ordenados de forma ascendente según la ordenación alfabética tradicional del español.

#### ***1.1.3.6.5 Definiciones***

Se presenta todos los términos manejados en el desarrollo de todo el proyecto

**Institución.-** Se refiere al proyecto comunitario “Guadalquivir”

**Sarci.-** nombre del sistema a desarrollar “Sistema automatizado de registro y control de la información”

**Afiliado.-** Persona que está inscrita en la institución

**Administrativo.-** persona encargada de la administración de la institución

**Usuario.-** persona habilitada por el sistema para interactuar con el mismo

## **I.1.4 Marco Teórico**

### **I.1.4.1 Definiciones**

#### **I.1.4.2 Sistema de información computarizado**

Los sistemas de información computarizados tienen un soporte informático, es decir se desarrollan en un entorno usuario - computadora, utilizando hardware y software computacional, redes de telecomunicaciones, técnicas de administración de bases de datos computarizadas y otras formas de tecnología de información.

#### **I.1.4.3 Sistemas de gestión de base de datos**

Sistema de gestión de base de datos) o en inglés Database management system (DBMS), es una agrupación de programas que sirven para definir, construir y manipular una base de datos.

Definir una base de datos: consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán.

Construir una base de datos: es el proceso de almacenar los datos sobre algún medio de almacenamiento.

Manipular una base de datos: incluye funciones como consulta, actualización, etc. de bases de datos.

Si el sistema soporta bases de datos relacionales se llama RDBMS en inglés o SGBDR en español.

#### ***1.1.4.3.1 Funciones de los sistemas de gestión de base de datos***

En la manipulación de una base de datos, los SGBD deben incluir un control de concurrencia, o sea, deben permitir a varios usuarios tener acceso "simultáneo" a la base de datos.

- Controlar la concurrencia implica que si varios usuarios acceden a la base de datos, la actualización de los datos se haga de forma controlada para que no haya problemas.
- Un SGBD también debe encargarse de cumplir las reglas de integridad y redundancias.
- Otra función importante en un SGBD es su capacidad de realizar copias de seguridad y de recuperación de datos.
  
- Restricción de accesos no autorizados.
- Suministrar múltiples interfaces de usuario.
- Representar relaciones complejas entre los datos.

#### ***1.1.4.3.2 Arquitectura de los sistemas de gestión de base de datos***

El grupo ANSI-SPARC (American National Standard Institute - Standards Planning and Requirements Committee) propuso una arquitectura de tres niveles para los SGBD cuyo objetivo principal era el de separar los programas de aplicación de la BD física. En esta arquitectura el esquema de una BD se define en tres niveles de abstracción distintos:

**Nivel interno o físico:** el más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador. Describe la estructura física de la BD mediante un esquema interno. Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la

información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, etcétera.

**Nivel externo o de visión:** es el más cercano a los usuarios, es decir, es donde se describen varios esquemas externos o vistas de usuarios. Cada esquema describe la parte de la BD que interesa a un grupo de usuarios en este nivel se representa la visión individual de un usuario o de un grupo de usuarios.

**Nivel conceptual:** describe la estructura de toda la BD para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento. Representa la información contenida en la BD. En la Figura 1.1 se representan los niveles de abstracción de la arquitectura ANSI.

#### **I.1.4.4 Herramientas tecnológicas de desarrollo**

##### ***I.1.4.4.1 Java***

Es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

##### ***I.1.4.4.2 Velocity***

Velocity es un motor de plantillas basado en Java. Les permite a los diseñadores de páginas hacer referencia a métodos definidos dentro del código Java. Los diseñadores Web pueden trabajar en paralelo con los programadores Java para desarrollar sitios de acuerdo al modelo de Modelo-Vista-Controlador (MVC), permitiendo que los diseñadores se concentren únicamente en crear un sitio bien diseñado y que los programadores se encarguen solamente de escribir código de primera calidad.

Velocity separa el código Java de las páginas Web, haciendo el sitio más mantenible a largo plazo y presentando una alternativa viable.

Velocity se puede utilizar para crear páginas web, SQL, PostScript y cualquier otro tipo de salida de plantillas. Se puede utilizar como una aplicación independiente para generar código fuente y reportes, o como un componente integrado en otros sistemas.

#### ***1.1.4.4.3 Html***

Html es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

Es **un** lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos

#### ***1.1.4.4.4 Hojas de estilo css***

CSS son las siglas de Cascading Style Sheets - Hojas de Estilo en Cascada - que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla.

El lenguaje CSS se basa en una serie de reglas que rigen el estilo de los elementos en los documentos estructurados, y que forman la sintaxis de las hojas de estilo. Cada regla consiste en un selector y una declaración, esta última va entre corchetes y consiste en una propiedad o atributo, y un valor separados por dos puntos.

#### ***1.1.4.4.5 PostgreSQL***

Es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global DevelopmentGroup).

#### ***1.1.4.4.6 4PGADMIN III POSTGRESQL 8.4***

PgAdmin es un proyecto de software libre publicado bajo la licencia de PostgreSQL. El software está disponible en fuente y el formato binario de la red de servidores espejos de PostgreSQL.

Como muchos otros proyectos open source, el desarrollo de PostgreSQL no es manejado por una sola compañía sino que es dirigido por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo.

#### ***1.1.4.4.7 JQuery***

jQuery es una biblioteca de Javascript que nos permite simplificar la manera de poder interactuar con los documentos en HTML.

Entre muchas de sus funciones, nos permite agregar interacción con la tecnología AJAX en páginas web y blogs, crear y desarrollar animaciones y manejar eventos.

jQuery, al igual que otras bibliotecas existentes en Internet, nos proporciona una serie de funcionalidades basadas en Javascript que de otra manera requerirían de mucho más código.

#### ***1.1.4.4.8 ENTERPRISE ARCHITECT***

Es una herramienta “modelado visual” de UML. La herramienta está diseñada para una gama amplia de usuarios como ser ingenieros de software, analistas de sistemas, analistas comerciales y arquitectos de sistemas o para cualquiera que está interesado en construir sistemas, software de gran potencia que usan un acercamiento orientado a objeto fiablemente.

Enterprise Architect combina el poder de la última especificación UML 2.1 con alto rendimiento, interfaz intuitiva, para traer modelado avanzado al escritorio, y para el equipo completo de desarrollo e implementación. Con un gran conjunto de características y un valor sin igual para el dinero, EA puede equipar a su equipo entero, incluyendo analistas, evaluadores, administradores de proyectos, personal del control de calidad, equipo de desarrollo y más, por una fracción del costo de algunos productos competitivos.

Enterprise Architect es una herramienta comprensible de diseño y análisis UML, cubriendo el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. El Lenguaje Unificado de Modelado provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente.

#### ***1.1.4.4.9 Eclipse***

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también

para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente, como BitTorrent o Azureus.

#### **I.1.4.5 Especificación de requerimientos**

##### ***I.1.4.5.1 Metodologías de desarrollo***

Es de suma importancia elegir la metodología adecuada, así como las herramientas de implementación adecuadas, es por ello que la metodología RUP basada en UML nos proporciona todas las bases para llevar al éxito la elaboración del software.

RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Rational Unified Process. Proceso Unificado de Rational, metodología del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo.

Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios.

##### ***I.1.4.5.2 Proceso de ingeniería de software***

##### ***I.1.4.5.3 Lenguaje unificado de modelado UML***

**Lenguaje Unificado de Modelado** (UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group) gracias a Jesús Moreno.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

#### ***1.1.4.5.4 Diagramas de UML***

En UML Existen dos tipos de diagramas los diagramas que nos dan una vista estática y dinámica.

Estos diagramas nos describen el comportamiento del sistema en el tiempo

**Diagramas de Casos de Uso:** Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.

**Diagrama de Clases:** El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clases incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

**Diagrama de Componentes:** Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

**Diagramas de Despliegue:** Los diagramas de despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución de instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo.

### *Diagramas Estáticos*

Estos diagramas nos describen el comportamiento del sistema en el tiempo.

**Diagramas de Casos de Uso:** Casos de Uso es una técnica para capturar información de cómo un sistema o negocio trabaja o de cómo se desea que trabaje. No pertenece estrictamente al enfoque orientado a objeto, es una técnica para captura de requisitos.

**Diagrama de Clases:** El Diagrama de Clases es el diagrama principal para el análisis y diseño. Un diagrama de clases presenta las clases del sistema con sus relaciones estructurales y de herencia. La definición de clases incluye definiciones para atributos y operaciones. El modelo de casos de uso aporta información para establecer las clases, objetos, atributos y operaciones.

**Diagrama de Componentes:** Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes de Ada, bibliotecas cargadas dinámicamente, etc. Las relaciones de dependencia se utilizan en los diagramas de

componentes para indicar que un componente utiliza los servicios ofrecidos por otro componente.

**Diagramas de Despliegue:** Los diagramas de despliegue muestran la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución de instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo.

### *Diagramas Dinámicos*

**Diagramas de Secuencia:** Diagrama que muestra las interacciones entre los objetos organizadas en una secuencia temporal. En participar muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados.

Representa una interacción, un conjunto de comunicaciones entre objetos organizadas visualmente por orden temporal. A diferencia de los diagramas de colaboración, los diagramas de secuencia incluyen secuencia temporales pero no incluyen las relaciones entre objetos. Pueden existir de forma de descriptor (describiendo todos los posibles escenarios) y en forma de instancia (describiendo un escenario real).

**Diagramas de Actividades:** Un estado de actividad representa una actividad: un paso en el flujo de trabajo o la ejecución de una operación. Un grafo de actividades describe grupos secuenciales y concurrentes de actividades. Los grafos de actividades se muestran en diagramas de actividades. Las actividades se enlazan por transiciones

automáticas. Cuando una actividad termina se desencadena el paso a la siguiente actividad. Un diagrama de actividad es provechoso para entender el comportamiento de alto nivel de la ejecución de un sistema, sin profundizar en los detalles internos de los mensajes. Los parámetros de entrada y salida de una acción se pueden mostrar usando las relaciones de flujo que conectan la acción y un estado de flujo de objeto.

## **I.1.5 Ingeniería de requerimientos**

### **I.1.5.1 Modelo de casos de usodel negocio**

#### ***I.1.5.1.1 Introducción***

Un Caso de Uso del Negocio define qué debe ocurrir en el negocio cuando este se realiza, describe el comportamiento de una sucesión de acciones que produce un resultado de valor para un Actor particular del negocio. Es decir, un Caso de Uso del Negocio describe una secuencia de acciones realizadas en el negocio que produce un resultado de valor observable para un actor individual del negocio. Por tanto, desde la perspectiva de un actor individual, un caso de uso del negocio define el flujo de trabajo completo que produce los resultados deseados.

El modelado de Casos de Uso del Negocio se basa en dos diagramas principales, el modelo de casos de uso del negocio y los modelos de objetos del negocio.

#### ***I.1.5.1.2 Propósito***

- Representar la funcionalidad provista de la organización como un todo.
- Comprender mejor el funcionamiento de la organización.

#### ***I.1.5.1.3 Alcance***

- Identificar los objetos de Negocio

- Describe los procesos del Negocio

#### 1.1.5.1.4 Diagrama de casos de uso del negocio

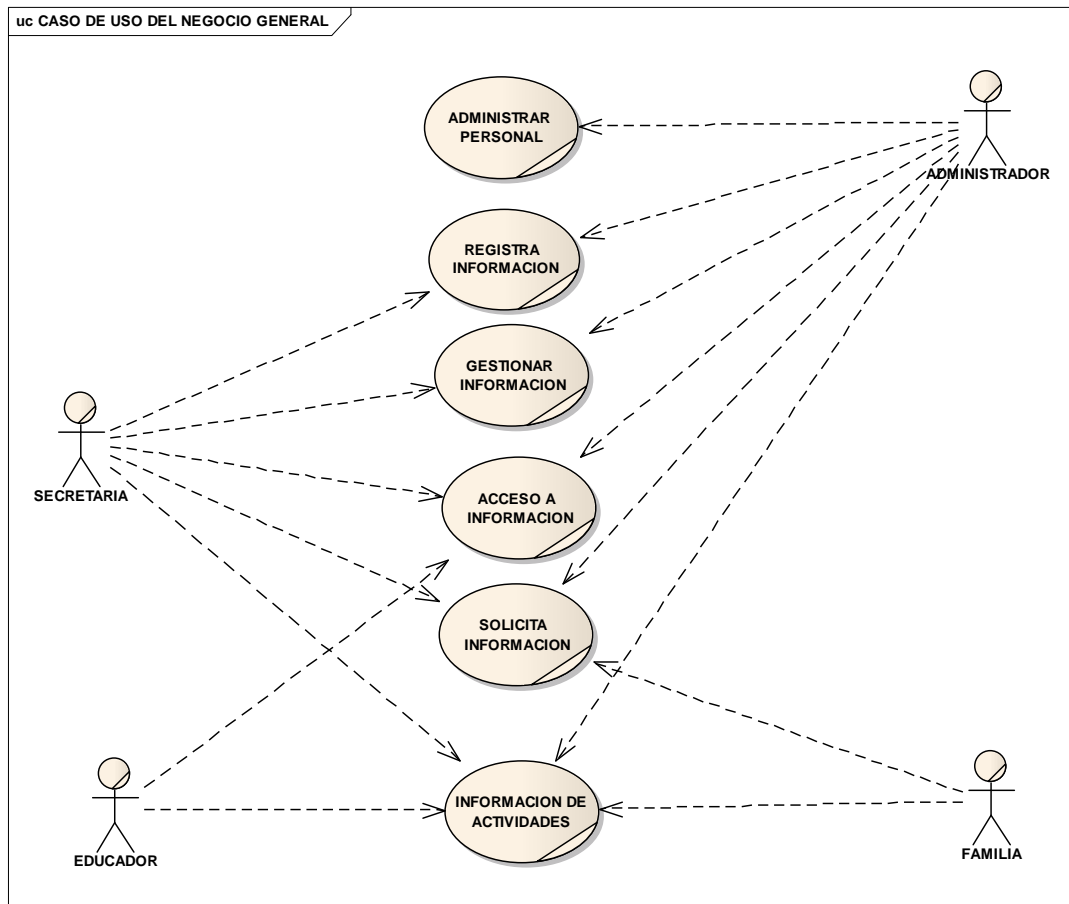


Figura N° 2 Caso de uso del negocio general

#### ***1.1.5.1.5 Descripción de los actores del negocio***

**Administrador.-** Es el encargado de supervisar todo los registros de información que se hacen en la institución,

**Secretaria.-** se encarga de registrar toda la información perteneciente a los afiliados a l proyecto “Guadalquivir”

**Educador.-** Encargado de las actividades que se realizan en la institución, y seguimiento a afiliados

**Familia.-** son los beneficiarios del funcionamiento de la institución

#### **1.1.5.2 Modelo de objetos del negocio**

##### ***1.1.5.2.1 Introducción***

Es un modelo que establece la forma actual en que la institución trabaja sus operaciones, y establecerá una idea del proceso a seguir para automatizar dichos procesos.

##### ***1.1.5.2.2 Propósito***

El modelo de objetos del negocio tiene como propósito evidenciar, los procesos que se realizan en el negocio.

##### ***1.1.5.2.3 Alcance***

- Describir operaciones de relaciones entre el administrador y los solicitantes que en todo caso son estudiantes universitarios.
- Describe los procesos del negocio
- Identificar y definir los objetos de negocio

### *1.1.5.2.4 Modelo de Objeto del Negocio*

#### *1.1.5.2.4.1 Registrar información*

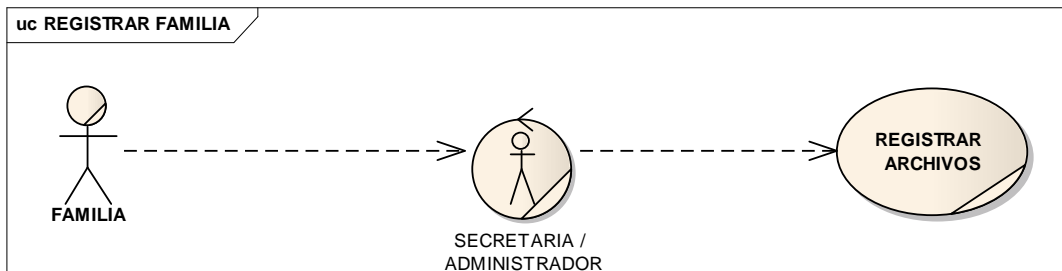


Figura N° 3 Caso de uso del negocio registrar familia

#### *1.1.5.2.4.2 Solicitar información*

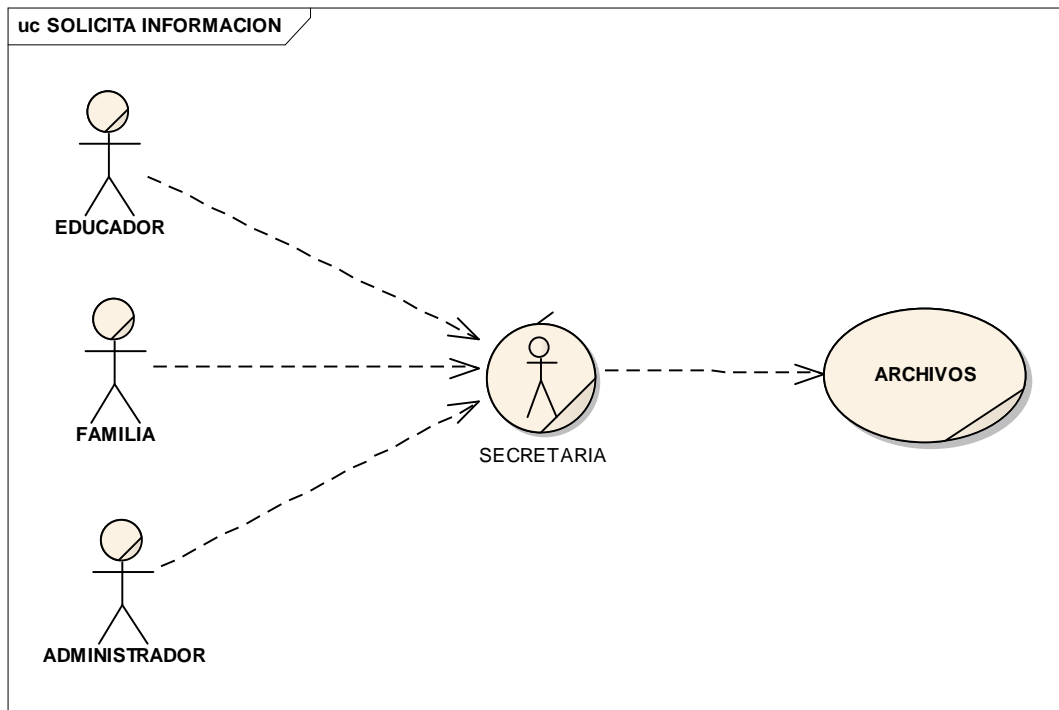




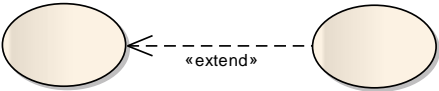
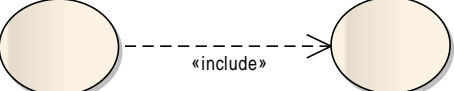
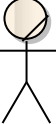


Figura N° 4 Caso de uso del negocio- solicitar información

### I.1.5.3 Estereotipos UML

<b>Caso de uso del negocio</b>	
<b>Objeto de negocio</b>	
<b>Caso de uso</b>	
<b>Comunicación</b>	
<b>Relación de Extensión</b>	
<b>Relación e Inclusión</b>	
<b>Actor del negocio</b>	


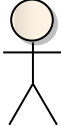



<b>Trabajador de negocio</b>	
<b>Actor</b>	
<b>Representa una clase</b>	
<b>Representa un controlador</b>	
<b>Representa una interfaz</b>	

Figura N° 5 Estereotipos UML

## **I.1.5.4 DIAGRAMA DE CASOS DE USO**

### ***I.1.5.4.1 Introducción***

El diagrama de casos de uso nos representa la forma en cómo un Cliente (Actor) opera con el sistema en desarrollo, además de la forma, tipo y orden en como los elementos interactúan

### ***I.1.5.4.2 Propósito***

- Involucrar a los usuarios en las etapas iniciales del análisis y el diseño del sistema.
- Ayuda a obtener los requerimientos desde el punto de vista del usuario.
- Modelar el contexto del sistema.
- Modelar los requerimientos del sistema.
- Identificar los procesos del sistema.

### ***I.1.5.4.3 Alcance***

- Describe lo que el sistema de software realizará dentro del negocio.
- Describe los alcances del sistema.
- Describe los procesos de sistema y de usuarios.

### ***I.1.5.4.4 Identificación de Actores***

#### **➤ Actor – Administrador**

El actor Administrador tiene la responsabilidad de administrar todos los módulos.

Los módulos en específico son:

### ***Modulo Administrativo***

- Gestionar Usuarios.
- Ingresar al Sistema
- Gestionar cargo
- Gestionar menús
- Gestionar familia
- Gestionar familia candidata
- Gestionar patrocinador
- Gestionar afiliados
- Gestionar actividades
- Controlar asistencia
- Gestionar correspondencia
- Registrar nivel educativo
- Gestionar salud
- Reportes
- Backup base de datos

**Actor – Secretaria**

- Gestionar Usuarios.
- Ingresar al Sistema
- Gestionar familia
- Gestionar familia candidata
- Gestionar patrocinador
- Gestionar afiliados
- Registrar nivel educativo
- Controlar asistencia
- Gestionar actividades
- Gestionar salud
- Reportes

**Actor – Educador**

- Ingresar al Sistema
- Gestionar actividades
- Controlar asistencia
- Reportes.

### 1.1.5.4.5 Diagramas de Casos de Uso

CASO DE USO: General del Sistema

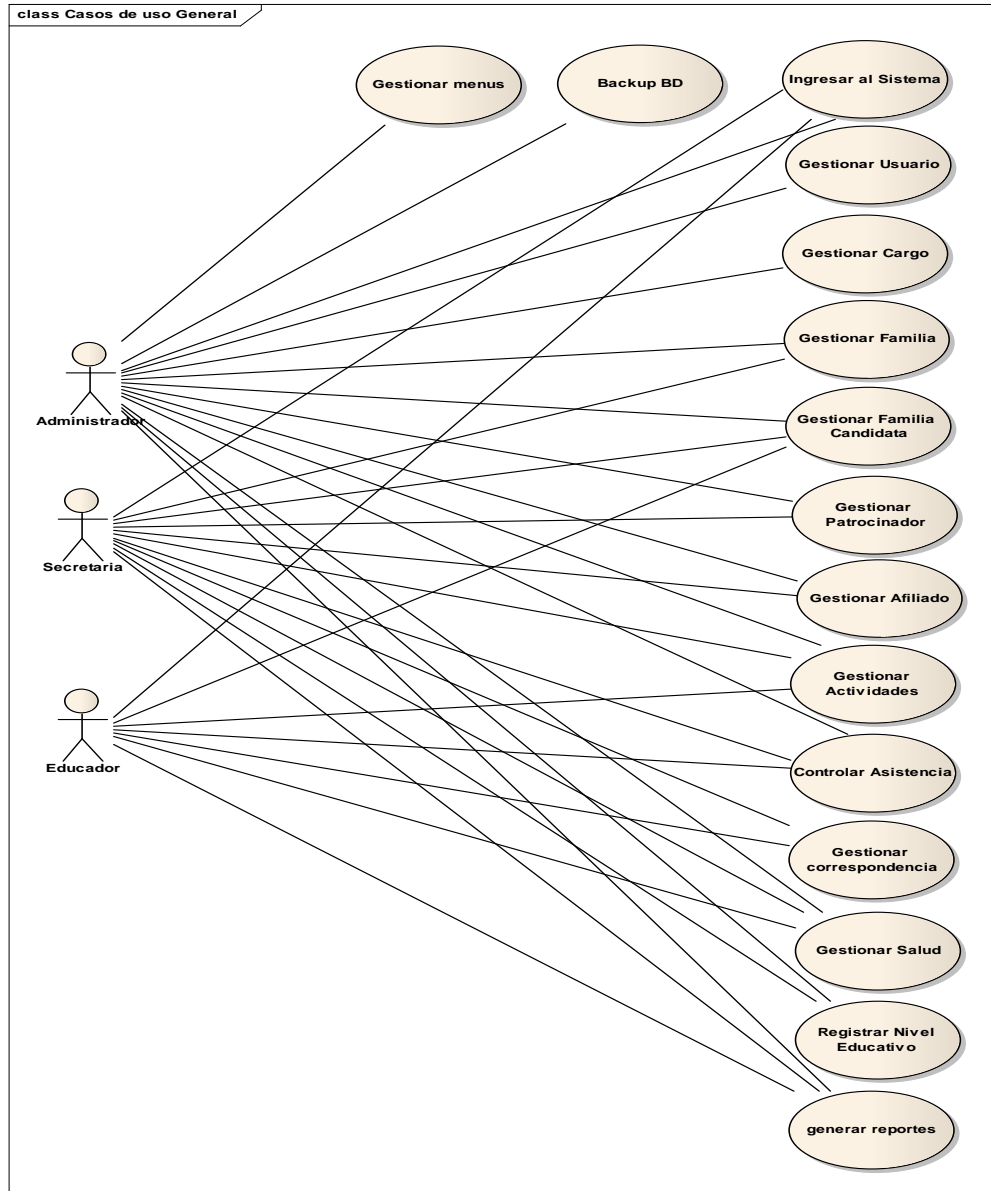


Figura N° 6 Caso de uso general

**I.1.5.5 Diagrama de modelo de casos de uso**

**Caso de uso Ingresar al Sistema**

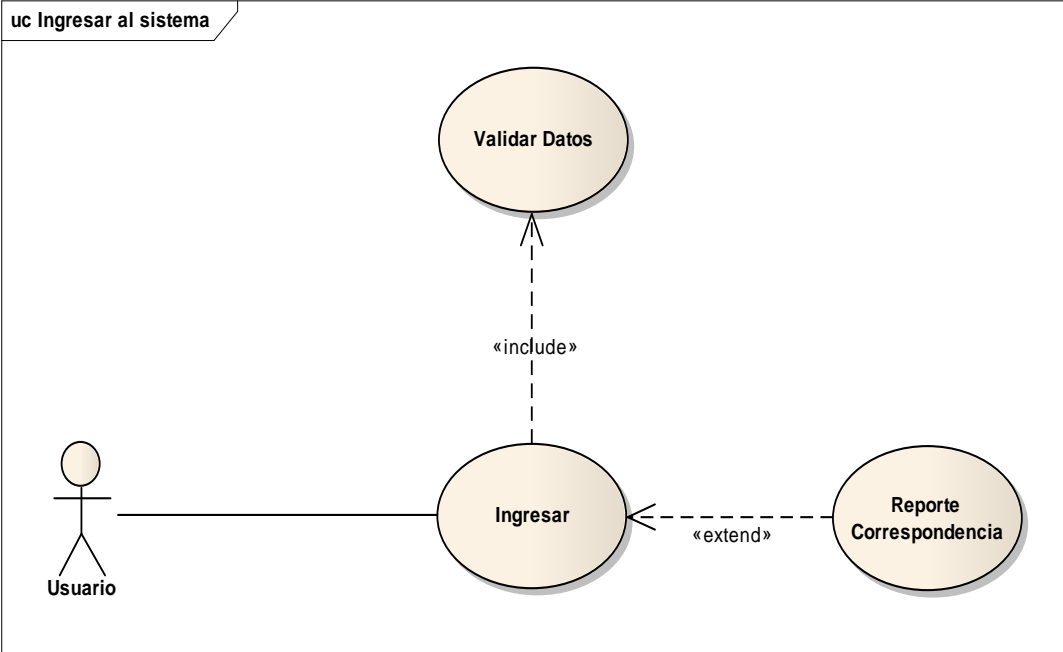


Figura N° 7 Ingresar al Sistema

### Caso de uso Gestionar Usuario

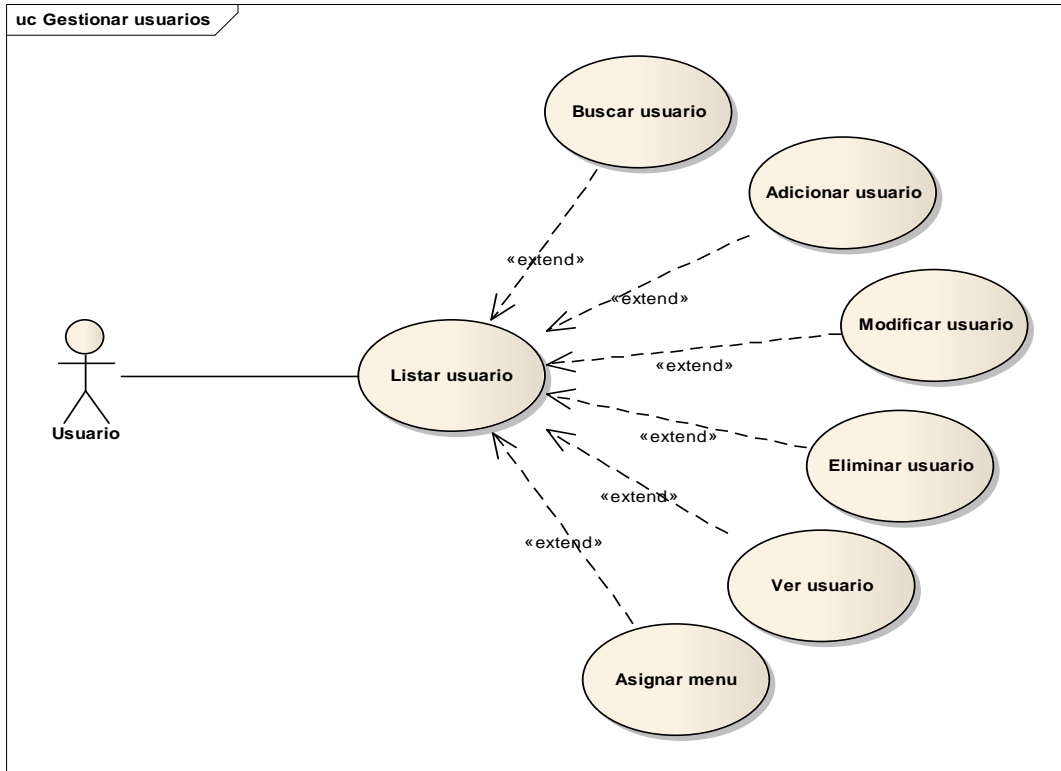


Figura N° 8 Gestionar Usuario

### Caso de Uso Gestionar Cargo

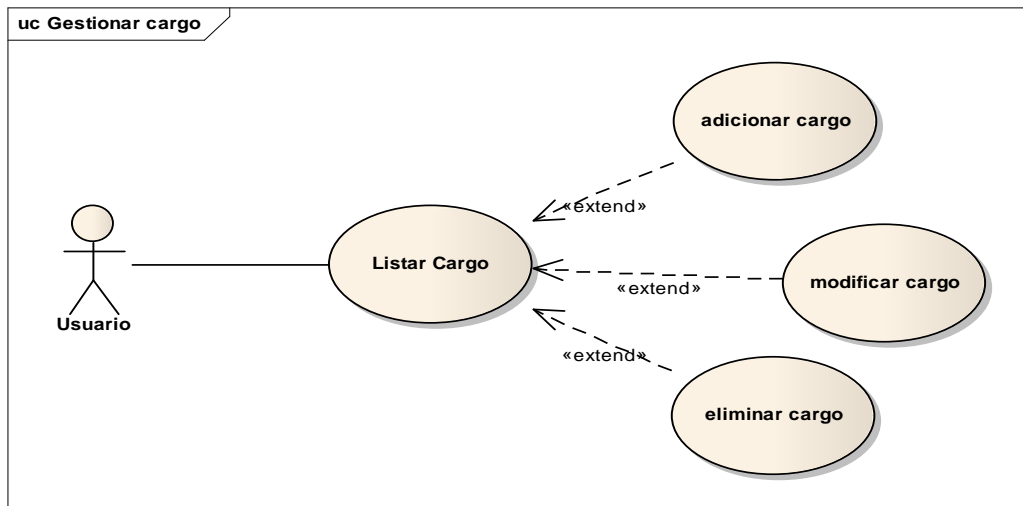


Figura N° 9 Gestionar Cargo

## Caso de Uso Gestionar Familia

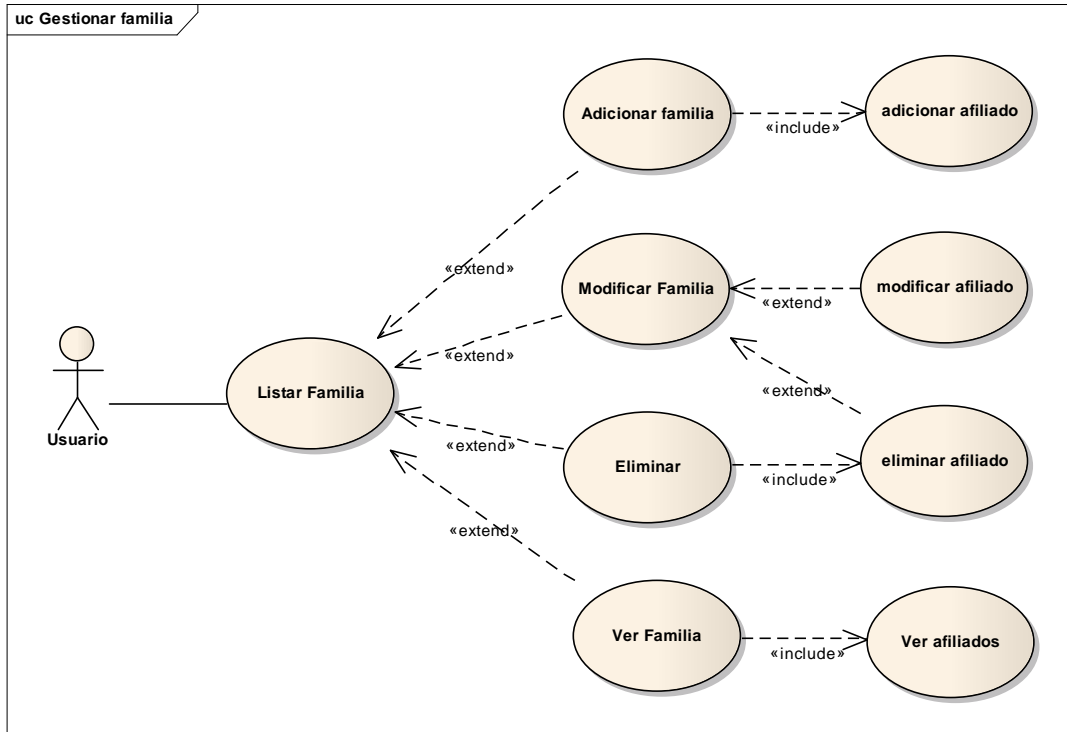


Figura N° 10 Gestionar Familia

## Caso de Uso Gestionar Familia Candidata

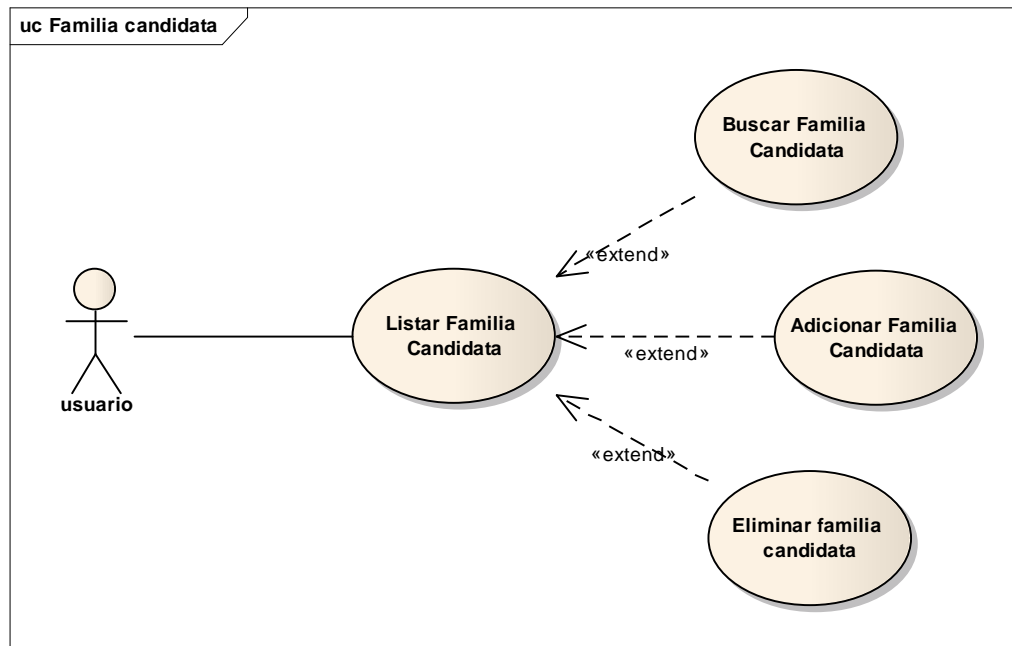


Figura N° 11 Gestionar Familia Candidata

## Caso de Uso Gestionar Patrocinador

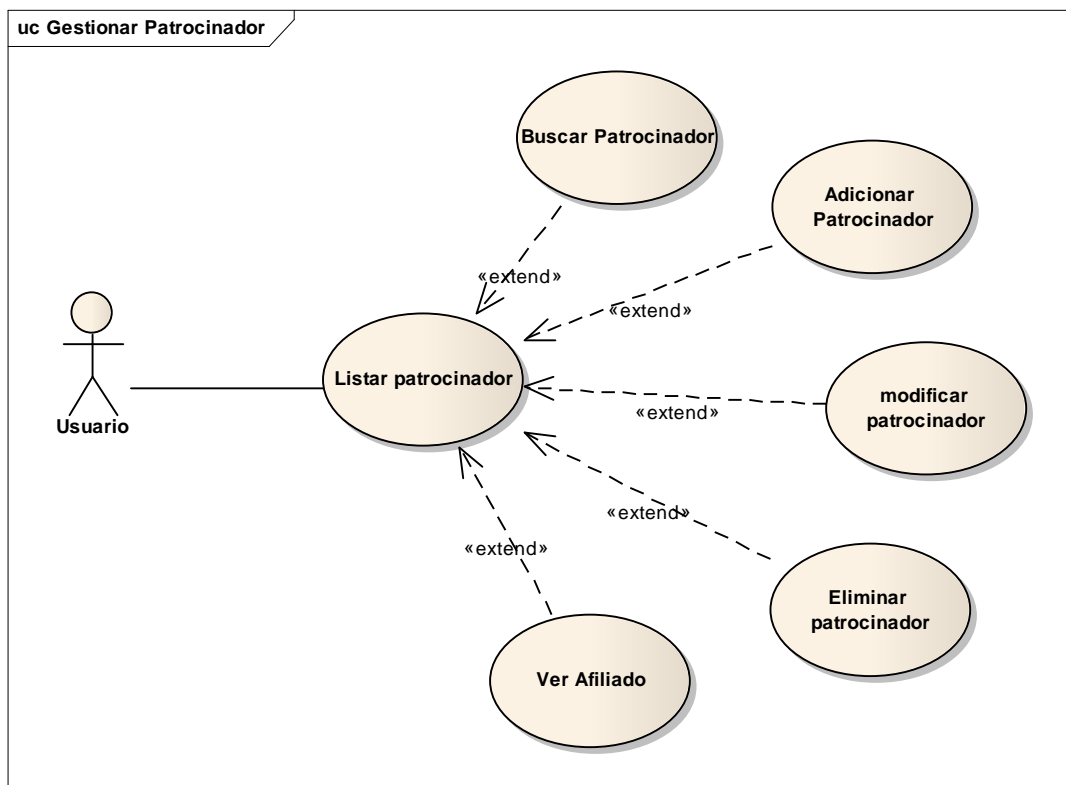


Figura N° 12 Gestionar Patrocinador

## Caso de Uso Gestionar Afiliado

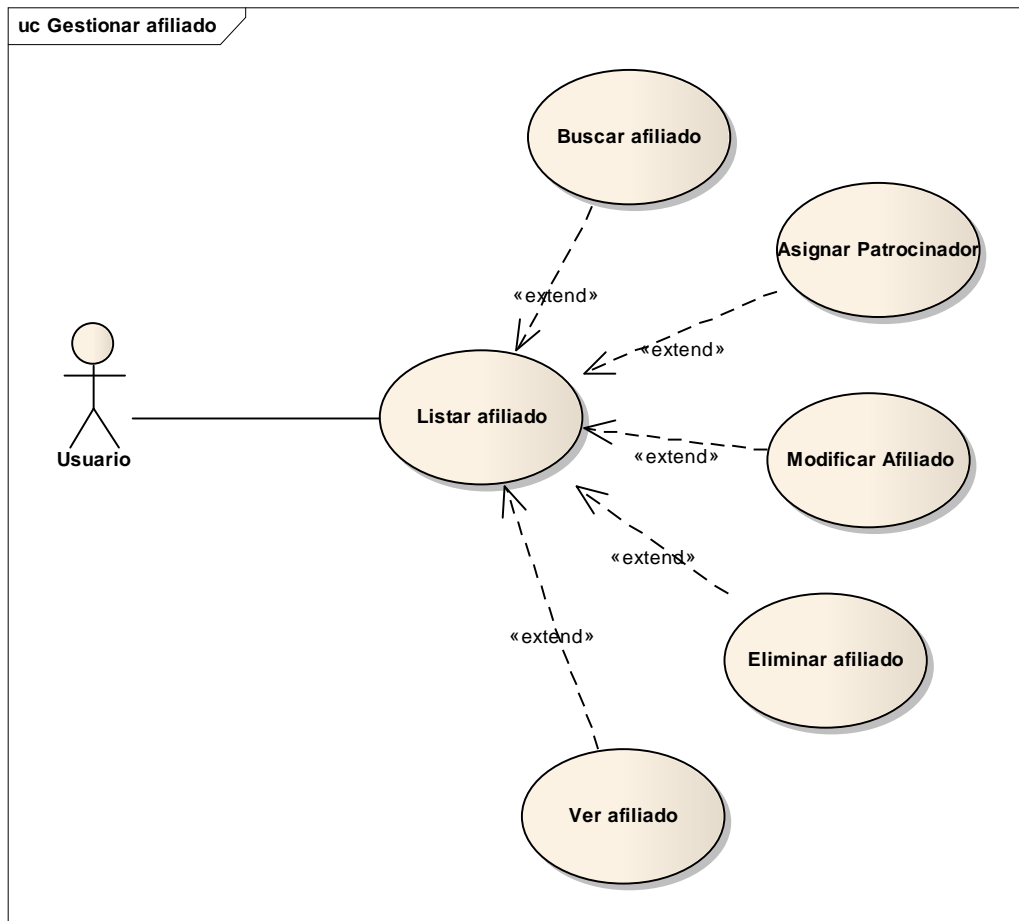


Figura N° 13 Gestionar Afiliado

## Caso de Uso Gestionar Actividades

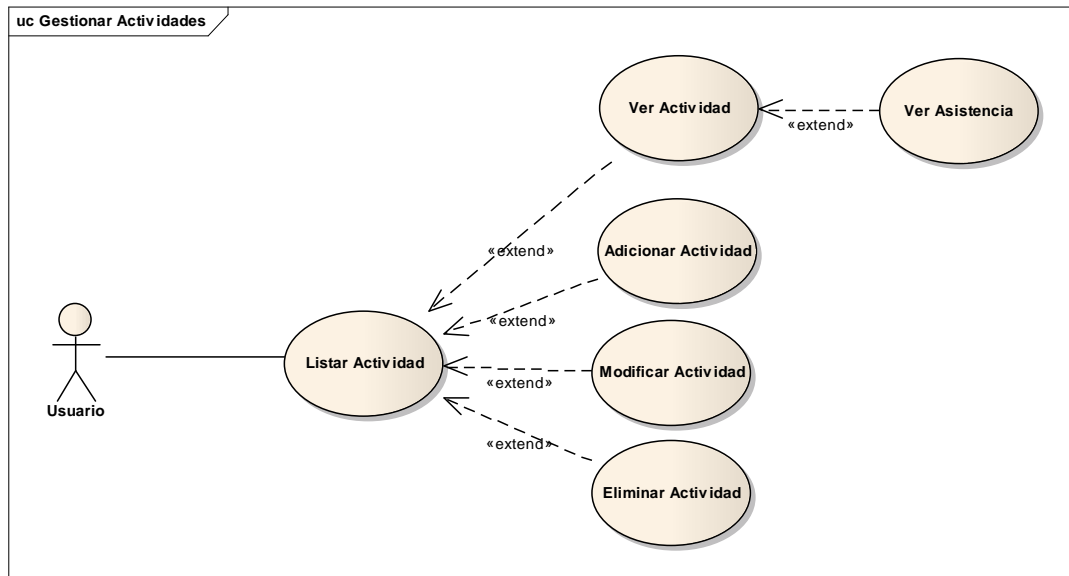


Figura N° 14 Gestionar Actividades

## Caso de Uso Controlar Asistencia

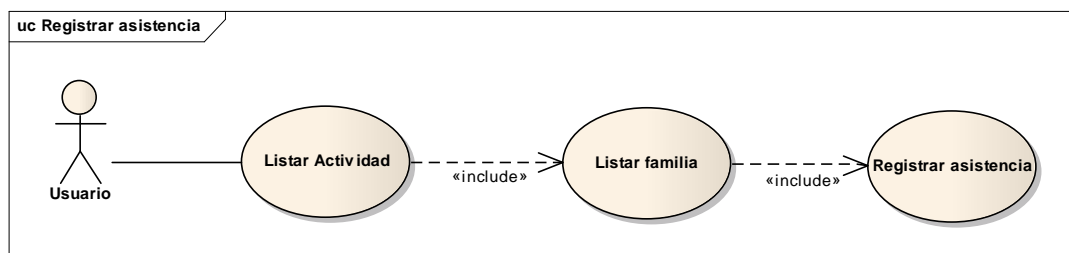


Figura N° 15 Controlar Asistencia

## Caso de Uso Registrar Nivel Educativo

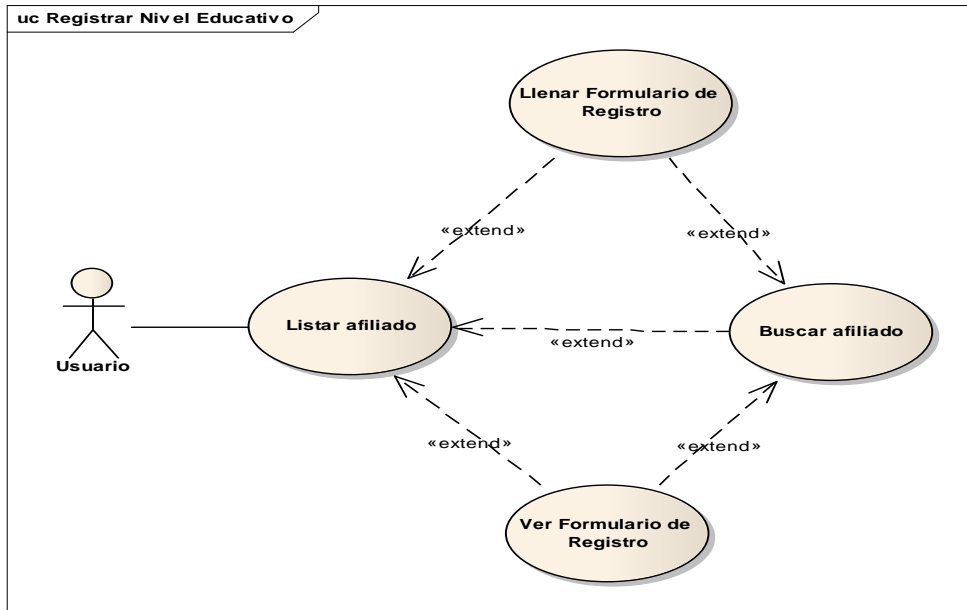


Figura N° 16 Registrar Nivel Educativo

## Caso de Uso Gestionar Salud

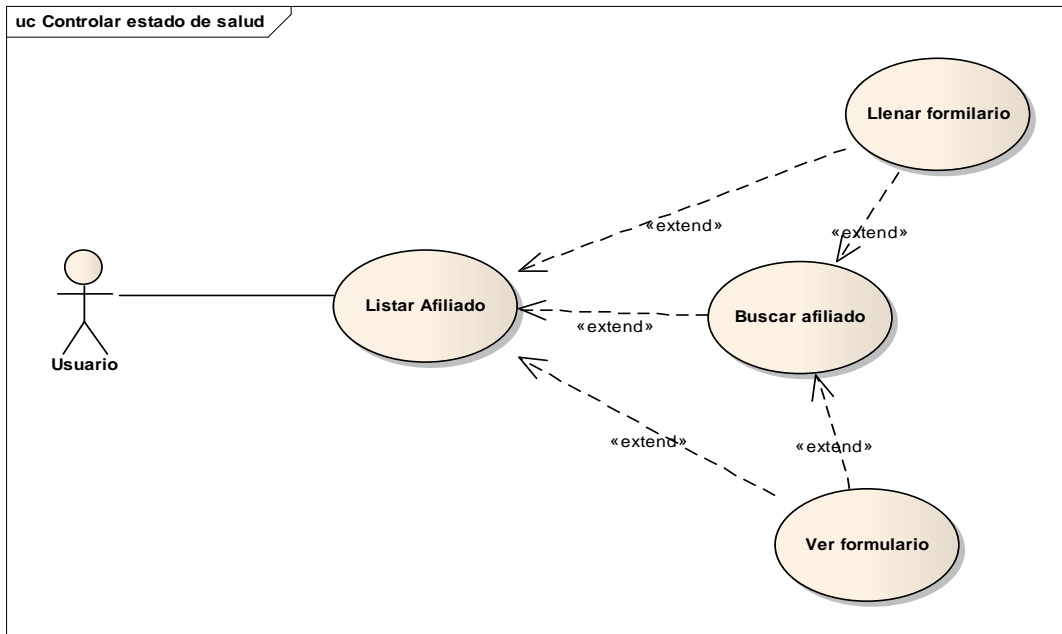


Figura N° 17 Controlar Estado de Salud

### Caso de Uso: Gestionar Correspondencia

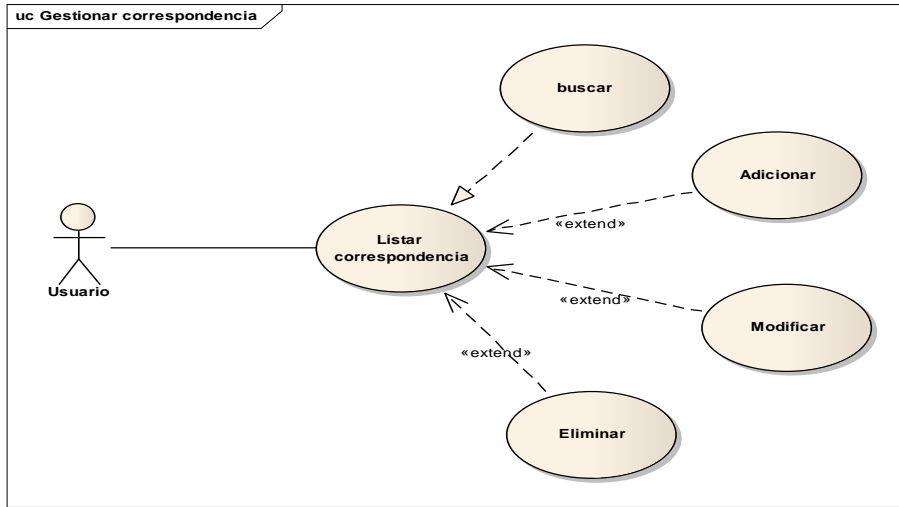


Figura N° 18 Gestionar correspondencia

### Caso de Uso Generar Reportes

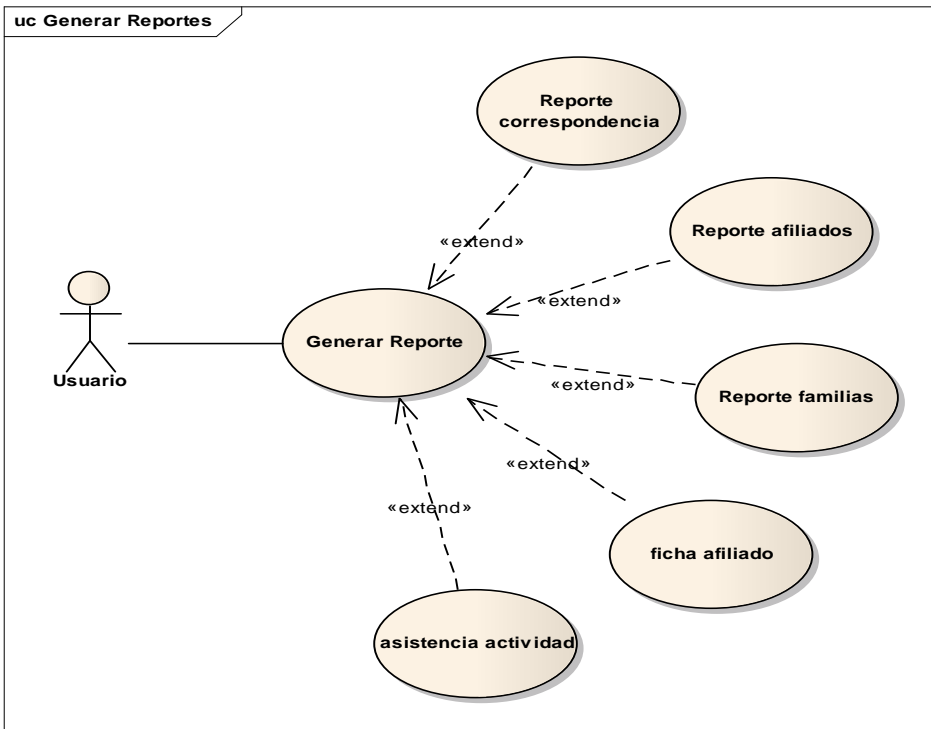


Figura N° 19 Generar Reporte

## Caso de uso Backup base de datos

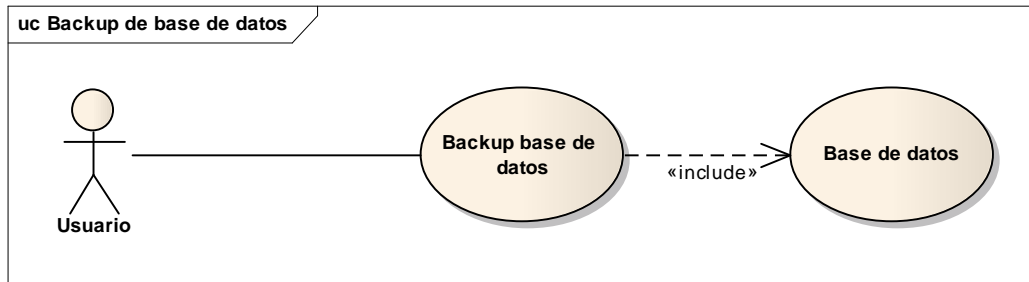


Figura N° 20 Backup base de datos

### I.1.5.6 Especificación de casos de uso

Para los Casos de Uso que lo se ejecuta una descripción detallada utilizando una plantilla de documento, donde se incluyen: precondiciones, postcondiciones, flujo de eventos, requisitos no-funcionales asociados. También, para casos de uso cuyo flujo de eventos sea complejo podrá adjuntarse una representación gráfica mediante un Diagrama de Actividad.

#### *I.1.5.6.1 Introducción*

Las Especificaciones de los Casos de Uso son una descripción detallada de los Casos de Uso del Sistema.

#### *I.1.5.6.2 Propósito*

Interpretar y describir los Casos de Uso.

#### *I.1.5.6.3 Alcance*

Describe los procesos internos de los Casos de Uso.

Detalla los flujos de los Casos de Uso según lo establecido por la organización.

#### *1.1.5.6.4 Especificación de los Casos de Uso*

##### **Caso de uso** Ingresar al Sistema

Casos de Uso	Ingresar al sistema
Actores	Administrador, Secretaria, Educador
Tipo	Básico
Propósito	Validar a un usuario ya registrado para el uso del sistema SARCI
Resumen	En este caso de uso, se inicia por el usuario. Valida al usuario mediante un usuario y contraseña a verificarse con su respectivo registro de usuario, para que pueda utilizar el sistema SARCI
Precondiciones	Estar registrado como usuario con un Usuario y Contraseña en la Base de Datos.
Flujo Principal	Se presenta al Usuario la Pantalla Inicial. El usuario ingresa datos de Usuario y Contraseña.
Subflujos	1. Si los datos del formulario están vacíos, el sistema muestra un mensaje "Revisar datos"
Excepciones	Ninguno

Tabla: 17 Caso de uso: Ingresar al Sistema

<b>Caso de uso</b>	<b>Validar datos</b>
Actores	Usuario-Administrador, Base de datos
Tipo	Inclusión
Propósito	Validar datos del Usuario.
Resumen	Este caso de uso inicia al presionar el Botón Ingresar cuando el Administrador introduce sus datos de Usuario y Contraseña en la pantalla inicial
Precondiciones	Estar registrado como usuario con un Usuario y Contraseña en la Base de Datos.
Flujo principal	Para validar los datos se ingresa presionando el botón “Ingresar”, si introduce correctamente los datos ingresa al sistema visualizando la Pantalla Principal del sistema Si los datos son incorrectos se visualiza el mensaje
Subflujos	1. Si los datos no son válidos el sistema muestra el mensaje “Datos incorrectos”
Excepciones	Si uno de los campos se encuentra vacío el administrador no podrá continuar con el proceso de ingresar.

Tabla: 18 Caso de uso: validar datos

<b>Caso de uso</b>	<b>Reporte Correspondencia</b>
Actores	Administrador, Secretaria, educador, Base de datos
Tipo	Extensión
Propósito	Imprimir lista de correspondencia del día
Resumen	En este caso de uso se puede ver la lista de los afiliados que tiene correspondencia, y se puede imprimir
Precondiciones	Ingresar a la pantalla inicial del sistema
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario ingresa a la pantalla Ingresar al Sistema</li> <li>2. Selecciona la opción imprimir correspondencia</li> <li>3. El sistema muestra la hoja de asistencia e imprime el reporte</li> </ol>
Subflujos	ninguno
Excepciones	ninguno

Tabla: 19Caso de uso Reporte correspondencia

## **Pantalla Principal**

Casos de Uso	Ninguno (No es un caso de uso, es la pantalla principal del sistema a la que se refieren todos los casos de uso)
Actores	Administrador, Educador, Secretaria (no tienen acceso a todos los casos de uso solo algunos).
Tipo	Ninguno
Propósito	Permite al administrador y encargado ver todos los menús disponibles del sistema y acceder a los mismos.
Resumen	En esta pantalla se ven todos los menús del sistema (casos de uso).
Precondiciones	Se requiere haber ejecutado antes el caso de uso Ingresar al Sistema.
Flujo Principal	Se presenta la pantalla Principal del sistema, el Administrador y el Encargado pueden manejar el sistema, tienen la opción de escoger alguna actividad de los menús.  Si la actividad seleccionada es “SALIR” se saldrá del sistema.

Tabla: 20 caso de uso: Principal

### Caso de uso Gestionar Usuario

<b>Caso de uso</b>	<b>Listar Usuario</b>
Actores	Usuario- Base de datos
Tipo	Básico.
Propósito	Gestionar los datos de los Usuarios mediante los siguientes procesos: <ol style="list-style-type: none"><li>1. Adicionar Usuario</li><li>2. Modificar Usuario</li><li>3. Eliminar Usuario</li><li>4. Ver Usuario</li></ol>
Resumen	En este caso de uso el Usuario Administrador realiza la gestión de los registros de los Usuarios del Proyecto “Guadalquivir” adicionando, modificando y eliminando los registros de los Usuarios.
Precondiciones	Haber ejecutado el caso de uso Ingresar al sistema.  El usuario debe estar en la pantalla principal
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario elige la opción Gestionar Usuarios</li><li>2. El sistema muestra la lista de usuarios existentes</li><li>3. Se visualiza las opciones de Adicionar, modificar, eliminar, ver.</li></ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 21 Caso de uso: listar usuario

### Caso de uso Buscar Usuario

<b>Caso de uso</b>	<b>Buscar Usuario</b>
Actores	Usuario- Base de datos
Tipo	Básico.
Propósito	Buscar Usuarios
Resumen	En este caso de uso el Usuario Administrador puede realizar la búsqueda de los usuarios del sistema
Precondiciones	Haber ejecutado el caso de uso Ingresar al sistema. Y Listar usuario
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario inserta datos del apellido paterno del usuario que busca</li><li>2. Selecciona la opción “buscar”</li><li>3. El sistema muestra el resultado de la búsqueda</li></ol>
Subflujos	Ninguno
Excepciones	Si el sistema no reconoce el dato insertado, no se muestra nada

Tabla: 22 Caso de uso: Buscar usuario

### Caso de uso Adicionar Usuario

Caso de uso	Adicionar Usuario
Actores	Administrador
Tipo	Extensión
Propósito	Adicionar Nuevo Usuario
Resumen	En este caso de uso el Administrador adiciona un nuevo Usuario
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Usuarios
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario elige la opción “adicionar nuevo usuario”</li><li>2. El sistema muestra un formulario para ingresar datos</li><li>3. Se ingresan los datos</li><li>4. Si el usuario elige la opción “Guardar” , el sistema adiciona el usuario a la base de datos</li><li>5. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</li></ol>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error

Excepciones	<ol style="list-style-type: none"> <li>1. cuando se hace el registro de un usuario que ya existe en la base de datos.</li> <li>2. no puede continuar porque la base de datos no permite campos vacíos</li> </ol>
-------------	--

Tabla: 23 Caso de uso: adicionar usuario

### Caso de Uso Modificar Usuario

Caso de uso	Modificar Usuario
Actores	Administrador
Tipo	Extensión
Propósito	Modificar datos de Usuario
Resumen	En este caso de uso el Administrador modifica los datos de un Usuario
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Usuarios y adicionar usuario
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la opción “modificar usuario”</li> <li>2. El sistema muestra un formulario para modificar datos</li> <li>3. Se ingresan los datos</li> <li>4. Si el usuario elige la opción “Modificar” , el sistema modifica el usuario a la base de datos</li> </ol> <p>Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</p>

Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	El sistema muestra un mensaje de confirmación ¿Seguro de modificar?

Tabla: 24 Caso de uso: Modificar usuario

### Caso de Uso Eliminar usuario

Caso de uso	Eliminar Usuario
Actores	Administrador
Tipo	Extensión
Propósito	Eliminar un Usuario
Resumen	En este caso de uso el Administrador Elimina a un Usuario
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Usuarios
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona un usuario de la lista</li> <li>2. Selecciona la opción "Eliminar"</li> <li>3. Si el usuario acepta la eliminación, el sistema eliminara al usuario, caso contrario volverá a la pantalla anterior</li> </ol>
Excepciones	Verificación de seguridad al momento de eliminar un dato. Mediante un mensaje emergente de confirmación

Tabla: 25 Caso de uso: Eliminar usuario

### Caso de uso Ver usuario

Caso de uso	Ver Usuario
Actores	Administrador
Tipo	Extensión
Propósito	ver un Usuario
Resumen	En este caso de uso el Administrador Puede Ver los datos de un usuario
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Usuarios
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario selecciona un usuario</li><li>2. Selecciona la opción "Ver Usuario"</li><li>3. El sistema muestra una pantalla con los datos del usuario</li></ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 26 Caso de uso: ver usuario

### Caso de uso Asignar Menú

Caso de uso	Asignar Menú
Actores	Administrador
Tipo	Extensión
Propósito	Asignar menú a usuario
Resumen	En este caso de uso el Administrador puede asignar menús a los usuarios
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Usuarios
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario selecciona un usuario</li><li>2. Selecciona la opción “asignar menú”</li><li>3. El sistema muestra una pantalla con los menús disponibles</li><li>4. El usuario selecciona los menús que requiera</li><li>5. Selecciona la opción asignar</li><li>6. Caso contrario se vuelve a la pantalla anterior</li></ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 27 Caso de uso: Asignar Menú

### **Caso de Uso Gestionar Cargo**

<b>Caso de uso</b>	<b>Listar Cargo</b>
Actores	Administrador
Tipo	Extensión
Propósito	Listar los cargos del proyecto
Resumen	En este Caso de uso el administrador puede visualizar a los cargos
Precondiciones	Haber ejecutado el caso de uso ingresar al sistema y estar dentro de la pantalla principal del sistema
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario selecciona la opción “Gestionar Cargo”</li><li>2. El sistema muestra la lista de los cargos existentes</li></ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 28 Caso de uso: Listar cargo

### **Caso de Uso adicionar cargo**

<b>Caso de uso</b>	<b>Adicionar Cargo</b>
Actores	Administrador
Tipo	Extensión
Propósito	Registrar nuevo Cargo.
Resumen	Este caso de uso, se inicia por el Administrador. Agrega un Cargo al sistema
Precondiciones	El sistema debe haber validado correctamente al usuario para poder agregar un nuevo Cargo.
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la opción “adicionar nuevo cargo”</li> <li>2. El sistema muestra un formulario para ingresar datos</li> <li>3. Se ingresan los datos</li> <li>4. Si el usuario elige la opción “Guardar” , el sistema adiciona el cargo a la base de datos</li> <li>5. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</li> </ol>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	El sistema está contemplado con 3 cargos de manera fija

Tabla: 29 Caso de uso: Adicionar cargo

## Caso de Uso Modificar Cargo

Caso de uso	Modificar Cargo
Actores	Administrador
Tipo	extensión
Propósito	Modificar datos del Cargo registrado.
Resumen	Este caso de uso, se inicia por el Administrador. Modifica un cargo agregado.
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario selecciona el cargo a modificar</li><li>2. El usuario elige la opción “modificar”</li><li>3. El sistema muestra un formulario para ingresar datos</li><li>4. Se ingresan los datos</li><li>5. Si el usuario elige la opción “modificar” , el sistema modifica el cargo a la base de datos</li></ol> <p>Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</p>
Subflujos	<p>Actualizar información</p> <p>Se actualiza la lista en la pantalla de los datos modificados del Cargo.</p> <p>Mensaje: ¿Seguro de Modificar?</p>

Tabla: 30 Caso de uso: Modificar cargo

### Caso de Uso Eliminar Cargo

Caso de uso	Eliminar Cargo
Actores	Administrador
Tipo	extensión
Propósito	Eliminar cargo
Resumen	Este caso de uso, se inicia por el administrador, elimina un cargo agregado.
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario selecciona el cargo a eliminar</li><li>2. El usuario elige la opción "Eliminar"</li><li>3. El sistema muestra un mensaje de confirmación</li><li>4. Si el usuario elige la opción "sí" , el sistema elimina el cargo de la base de datos</li><li>5. Si el usuario elige la opción "no", se regresa a la pantalla anterior</li></ol>
Subflujos	Actualizar información  Se actualiza la lista en la pantalla de los datos del Cargo.

Tabla: 31 Caso de uso: Eliminar cargo

### Caso de Uso Gestionar Familia

<b>Caso de uso</b>	<b>Listar Familia</b>
Actores	Administrador, secretaria
Tipo	Básico.
Propósito	Gestionar los datos de las Familias mediante los siguientes procesos: <ol style="list-style-type: none"><li>1. Adicionar Familia</li><li>2. Ver familia</li><li>3. Eliminar Familia</li></ol>
Resumen	En este caso de uso el Usuario Administrador realiza la gestión de los registros de las familias del Proyecto “Guadalquivir” adicionando y eliminando.
Precondiciones	Haber ejecutado anteriormente el caso de uso ingresar al sistema y estar en la pantalla principal del sistema
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario selecciona la opción “Gestionar Familia”</li><li>2. El sistema muestra la lista de familias existentes</li><li>3. El sistema muestra las opciones de adicionar, modificar , eliminar, ver</li></ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 32 Caso de uso: gestionar familia

### Caso de uso Buscar Familia

<b>Caso de uso</b>	<b>Buscar Familia</b>
Actores	Usuario- Base de datos
Tipo	Extensión
Propósito	Buscar Familias
Resumen	En este caso de uso el Usuario Administrador puede realizar la búsqueda en el sistema
Precondiciones	Haber ejecutado el caso de uso Ingresar al sistema. Y Listar familia
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario inserta datos del apellido paterno de la familia que busca</li><li>2. Selecciona la opción “buscar”</li><li>3. El sistema muestra el resultado de la búsqueda</li></ol>
Subflujos	Ninguno
Excepciones	Si el sistema no reconoce el dato insertado, no se muestra nada

Tabla: 33 Caso de uso: Buscar Familia

### Caso de Uso Adicionar Familia

<b>Caso de uso</b>	<b>Adicionar Familia</b>
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Adicionar Nueva Familia
Resumen	En este caso de uso el Administrador adiciona un nueva Familia
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Familia
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario selecciona la opción “Adicionar nueva familia”</li><li>2. El sistema muestra un formulario para llenar datos</li><li>3. El usuario inserta datos</li><li>4. Si elige la opción “guardar” el sistema adiciona la familia a la base de datos, caso contrario se vuelve a la pantalla anterior</li></ol>
Subflujos	<ol style="list-style-type: none"><li>1. Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error</li></ol>
Excepciones	<ol style="list-style-type: none"><li>1. Cuando se hace el registro de una familia que ya existe en la base de datos.</li><li>2. no puede continuar porque la base de datos no permite campos incorrectos</li></ol>

Tabla: 34 Caso de uso: adicionar familia

**Caso de Uso Adicionar Afiliado**

<b>Caso de uso</b>	<b>Adicionar Afiliado</b>
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Adicionar Nueva afiliado
Resumen	En este caso de uso el Administrador adiciona un nuevo afiliado
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Familia, adicionar Familia
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario adiciona la nueva familia</li> <li>2. El sistema muestra un formulario para llenar datos de los afiliados</li> <li>3. El usuario inserta datos</li> <li>4. Si elige la opción “guardar” el sistema adiciona al afiliado a la familia en la base de datos, caso contrario se vuelve a la pantalla anterior</li> </ol>
Subflujos	<ol style="list-style-type: none"> <li>1. Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error</li> </ol>
Excepciones	<ol style="list-style-type: none"> <li>1. Cuando se hace el registro de un afiliado que ya existe en la base de datos.</li> <li>2. no puede continuar porque la base de datos no permite campos incorrectos</li> </ol>

Tabla: 35 Caso de uso: adicionar afiliado

### Caso de uso Modificar Familia

<b>Caso de uso</b>	<b>Modificar Familia</b>
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Modifica Familia
Resumen	En este caso de uso el Administrador modifica los datos que forman parte de la familia
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Familia y adicionar familia
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario selecciona la familia</li><li>2. Selecciona la opción “modificar”</li><li>3. El sistema muestra una lista de los integrantes de la familia</li><li>4. El usuario selecciona al integrante a modificar</li><li>5. El sistema muestra un formulario para modificar datos</li><li>6. El usuario modifica los datos que desee</li><li>7. Si se desea guardar la modificación se elige la opción “modificar”, caso contrario se vuelve a la pantalla anterior</li></ol>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error.

Excepciones	<ol style="list-style-type: none"> <li>1. cuando se hace el registro de una persona que ya existe en la base de datos.</li> <li>2. no puede continuar porque la base de datos no permite campos vacíos</li> </ol>
-------------	---

Tabla: 36 Caso de uso: adicionar miembros

### Caso de Uso Eliminar Familia

Caso de uso	Eliminar Familia
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Eliminar una Familia
Resumen	En este caso de uso el Administrador Elimina a una Familia
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Familia
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona una familia</li> <li>2. Elige la opción “eliminar”</li> <li>3. Si el usuario acepta la eliminación, el sistema eliminara la familia, caso contrario volverá a la pantalla anterior</li> </ol>
Subflujos	
Excepciones	Verificación de seguridad al momento de eliminar un dato. Mediante un mensaje emergente de confirmación

Tabla: 37 Caso de uso: eliminar familia

**Caso de uso Ver Familia**

<b>Caso de uso</b>	<b>Ver Familia</b>
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Ver los datos de una familia
Resumen	En este caso de uso el usuario puede ver los datos de la familia
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Familia
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona una familia</li> <li>2. Elige la opción "Ver Familia"</li> <li>3. El sistema muestra una pantalla con todos los datos de la familia seleccionada</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 38 Caso de uso: Ver familia

### **Caso de Uso Gestionar Familia Candidata**

<b>Caso de uso</b>	<b>Listar Familias Candidatas</b>
Actores	Administrador , secretaria
Tipo	Básico.
Propósito	Gestionar los datos de las familias candidatas mediante los siguientes procesos:  <ol style="list-style-type: none"><li>1. Adicionar Familias Candidatas</li><li>2. Eliminar Familias Candidatas</li></ol>
Resumen	En este caso de uso el Usuario realiza la gestión de los registros de las familias que aspiran ingresar al Proyecto “Guadalquivir” adicionando y eliminando los registros de las Familias Candidatas
Precondiciones	Estar dentro del Sistema. Validarse como administrador
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario Elige la opción “Gestionar Familia Candidata”</li><li>2. El sistema muestra un listado de las familias candidatas existentes</li></ol>
Subflujos	

Excepciones	
-------------	--

Tabla: 39 Caso de uso: Listar familias candidatas

### Caso de uso Buscar Familia Candidata

Caso de uso	Buscar Familia Candidata
Actores	Usuario- Base de datos
Tipo	Básico.
Propósito	Buscar familia candidata
Resumen	En este caso de uso el Usuario Administrador puede realizar una búsqueda de datos en el sistema
Precondiciones	Haber ejecutado el caso de uso Ingresar al sistema. Y Listar Familia Candidata
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario inserta datos del apellido paterno de la familia que busca</li> <li>2. Selecciona la opción “buscar”</li> <li>3. El sistema muestra el resultado de la búsqueda</li> </ol>
Subflujos	Ninguno
Excepciones	Si el sistema no reconoce el dato insertado, no se muestra nada

Tabla: 40 Caso de uso Buscar Familia Candidata

### Caso de uso adicionar familia candidata

<b>Caso de uso</b>	<b>Adicionar Familia Candidata</b>
Actores	Administrador , secretaria
Tipo	Extensión
Propósito	Adicionar Nueva Familias Candidatas
Resumen	En este caso de uso el Administrador adiciona una nueva Familia Candidatas.
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar familia candidata
Flujo principal	<ol style="list-style-type: none"><li>1. El usuario elige la opción “adicionar familia candidata”</li><li>2. El sistema muestra un formulario para ingresar datos</li><li>3. Se ingresan los datos</li><li>4. Si el usuario elige la opción “Guardar” , el sistema adiciona la familia a la base de datos</li><li>5. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</li></ol>

Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	<ol style="list-style-type: none"> <li>1. cuando se hace el registro de una familia que ya existe en la base de datos.</li> <li>2. no puede continuar porque la base de datos no permite campos vacíos</li> </ol>

Tabla: 41 Caso de uso: adicionar familia candidata

### Caso de uso Modificar Familia Candidata

Caso de uso	Modificar familia Candidata
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Modificar datos de familias candidatas
Resumen	En este caso de uso el Administrador modifica los datos de una Familia Candidata
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Familia Candidata
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la opción “modificar”</li> <li>2. El sistema muestra un formulario para modificar datos</li> <li>3. Se ingresan los datos</li> <li>4. Si el usuario elige la opción “modificar” , el sistema modifica La familia en la base de</li> </ol>

	<p>datos</p> <ol style="list-style-type: none"> <li>1. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</li> </ol>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	no puede continuar porque la base de datos no permite campos vacíos

Tabla: 42 Caso de uso: Modificar Familia Candidata

<b>Caso de uso</b>	<b>Eliminar Familia Candidata</b>
Actores	Administrador , secretaria
Tipo	Extensión
Propósito	Eliminar una familia candidata
Resumen	En este caso de uso el Administrador Elimina a una Familia Candidata.
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso listar familia candidata y adicionar familia candidata
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona una familia</li> <li>2. Elige la opción “eliminar”</li> <li>3. Si el usuario acepta la eliminación, el sistema eliminara la familia, caso contrario volverá a la pantalla anterior</li> </ol>
Subflujos	

Excepciones	Verificación de seguridad al momento de eliminar un dato. Mediante un mensaje emergente de confirmación
-------------	--

Tabla: 43 Caso de uso: Eliminar familia candidata

### Caso de Uso Gestionar Patrocinador

Caso de uso	Listar Patrocinadores
Actores	Administrador, secretaria
Tipo	Básico.
Propósito	Gestionar los datos de los Patrocinadores mediante los siguientes procesos: <ol style="list-style-type: none"> <li>1. Adicionar Patrocinador</li> <li>2. Modificar Patrocinador</li> <li>3. Eliminar Patrocinador</li> </ol>
Resumen	En este caso de uso el Usuario realiza la gestión de los registros de los Patrocinadores que brindan apoyo a los niños afiliados al Proyecto “Guadalquivir” adicionando, modificando y eliminando los registros de las Patrocinadores.
Precondiciones	Haber ejecutado anteriormente el caso de uso ingresar al sistema y estar en la pantalla principal
Flujo principal	1. El Usuario selecciona la opción “Gestionar Patrocinadores”

	2. El sistema muestra la lista de los patrocinadores existentes
--	---

Tabla: 44 Caso de uso: Listar patrocinadores

### Caso de uso Buscar patrocinador

<b>Caso de uso</b>	<b>Buscar Patrocinador</b>
Actores	Usuario- Base de datos
Tipo	Básico.
Propósito	Buscar patrocinador
Resumen	En este caso de uso el Usuario Administrador puede realizar una búsqueda de datos en el sistema
Precondiciones	Haber ejecutado el caso de uso Ingresar al sistema. Y Listar Patrocinadores
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario inserta datos del apellido paterno del patrocinador que busca</li> <li>2. Selecciona la opción “buscar”</li> <li>3. El sistema muestra el resultado de la búsqueda</li> </ol>
Subflujos	Ninguno

Excepciones	Si el sistema no reconoce el dato insertado, no se muestra nada
-------------	---

Tabla: 45 Caso de uso Buscar patrocinador

### Caso de Uso Adicionar Patrocinador

Caso de uso	Adicionar Patrocinador
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Adicionar Nuevo Patrocinador
Resumen	En este caso de uso el Administrador adiciona un nuevo Patrocinador
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Patrocinador
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la opción “adicionar nuevo patrocinador”</li> <li>2. El sistema muestra un formulario para ingresar datos</li> <li>3. Se ingresan los datos</li> <li>4. Si el usuario elige la opción “Guardar” , el sistema adiciona al patrocinador a la base de</li> </ol>

	<p>datos</p> <p>5. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</p>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	<ol style="list-style-type: none"> <li>1. cuando se hace el registro de un usuario que ya existe en la base de datos.</li> <li>2. no puede continuar porque la base de datos no permite campos vacíos</li> </ol>

Tabla: 46 Caso de uso: Adicionar patrocinador

#### **Caso de uso Modificar Patrocinador**

<b>Caso de uso</b>	<b>Modificar Patrocinador</b>
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Modificar datos de Patrocinador
Resumen	En este caso de uso el Administrador modifica los datos de un Patrocinador
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Patrocinador y adicionar patrocinador
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la opción “modificar”</li> <li>2. El sistema muestra un formulario para modificar datos</li> <li>3. Se ingresan los datos</li> <li>4. Si el usuario elige la opción “modificar” , el</li> </ol>

	<p>sistema modifica el patrocinador en la base de datos</p> <p>2. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</p>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	no puede continuar porque la base de datos no permite campos vacíos

Tabla: 55 Caso de uso: Modificar patrocinador

### Caso de Uso Eliminar Patrocinador

Caso de uso	Eliminar Patrocinador
Actores	Administrador, Secretaria
Tipo	Extensión
Propósito	Eliminar un Patrocinador
Resumen	En este caso de uso el Administrador Elimina a un Patrocinador
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Patrocinador
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona un patrocinador</li> <li>2. Elige la opción “eliminar”</li> <li>3. Si el usuario acepta la eliminación, el sistema eliminara el patrocinador, caso contrario volverá a la pantalla anterior</li> </ol>

Subflujos	
Excepciones	Verificación de seguridad al momento de eliminar un dato. Mediante un mensaje emergente de confirmación

Tabla: 47 Caso de uso: Eliminar patrocinador

### Caso de uso Ver Afiliado

Caso de uso	Ver Afiliado
Actores	Administrador, Secretaria
Tipo	Extensión
Propósito	Ver a que afiliado patrocina
Resumen	En este caso de uso se puede visualizar el afiliado patrocinado
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Patrocinador
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona un patrocinador</li> <li>2. Elige la opción “ver Afiliado”</li> <li>3. El sistema muestra una pantalla con el perfil del afiliado patrocinado</li> </ol>
Subflujos	Si no tiene ningún patrocinado, el sistema muestra un

	mensaje informativo “no tiene afiliado”
Excepciones	

Tabla: 48 Caso de uso: Ver afiliado

### Caso de uso Gestionar Afiliados

<b>Caso de uso</b>	<b>Listar Afiliado</b>
Actores	Administrador, secretaria
Tipo	Básico.
Propósito	Gestionar los datos de los afiliados mediante los siguientes procesos: <ol style="list-style-type: none"> <li>1. Buscar</li> <li>2. Modificar</li> <li>3. Eliminar</li> <li>4. Ver</li> </ol>
Resumen	En este caso de uso el Usuario realiza la gestión de los registros de los Afiliados que forman parte del proyecto “Guadalquivir”

Precondiciones	Haber ejecutado anteriormente el caso de uso ingresar al sistema y estar en la pantalla principal
Flujo principal	<p>3. El Usuario selecciona la opción “Gestionar Afiliados”</p> <p>4. El sistema muestra la lista de los afiliados existentes</p>

Tabla: 49 Caso de uso: listar afiliados

### Caso de uso buscar afiliados

<b>Caso de uso</b>	<b>Buscar Afiliado</b>
Actores	Usuario- Base de datos
Tipo	Básico.
Propósito	Buscar Afiliados
Resumen	En este caso de uso el Usuario Administrador puede realizar una búsqueda de datos en el sistema

Precondiciones	Haber ejecutado el caso de uso Ingresar al sistema. Y Listar afiliados
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario inserta datos del apellido paterno del afiliado que busca</li> <li>2. Selecciona la opción “buscar”</li> <li>3. El sistema muestra el resultado de la búsqueda</li> </ol>
Subflujos	Ninguno
Excepciones	Si el sistema no reconoce el dato insertado, no se muestra nada

Tabla: 50 Caso de uso: buscar afiliado

### **Caso de uso Modificar Afiliado**

<b>Caso de uso</b>	<b>Modificar Afiliado</b>
Actores	Administrador, secretaria
Tipo	Extensión
Propósito	Modificar datos del Afiliado
Resumen	En este caso de uso el Administrador modifica los datos de un Afiliado
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Afiliado
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la opción “modificar”</li> <li>2. El sistema muestra un formulario para</li> </ol>

	<p>modificar datos</p> <ol style="list-style-type: none"> <li>3. Se ingresan los datos</li> <li>4. Si el usuario elige la opción “modificar” , el sistema modifica el Afiliado en la base de datos</li> <li>3. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</li> </ol>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	no puede continuar porque la base de datos no permite campos vacíos

Tabla: 51 Caso de uso: Modificar Afiliado

### Caso de Uso Eliminar Afiliado

Caso de uso	Eliminar Afiliado
Actores	Administrador, Secretaria
Tipo	Extensión
Propósito	Eliminar un Afiliado
Resumen	En este caso de uso el Administrador Elimina a un Afiliado del sistema
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Afiliado
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona un Afiliado</li> <li>2. Elige la opción “eliminar”</li> <li>3. Si el usuario acepta la eliminación, el sistema</li> </ol>

	eliminar al Afiliado, caso contrario volverá a la pantalla anterior
Subflujos	
Excepciones	Verificación de seguridad al momento de eliminar un dato. Mediante un mensaje emergente de confirmación

Tabla: 52 Caso de uso: Eliminar Afiliado

### Caso de uso Ver Afiliado

Caso de uso	Ver Afiliado
Actores	Administrador, Secretaria
Tipo	Extensión
Propósito	Ver los datos de un afiliado
Resumen	En este caso de uso el usuario puede revisar los datos de un afiliado
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Afiliado
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona un afiliado</li> <li>2. Elige la opción “ver Afiliado”</li> </ol>

	3. El sistema muestra una pantalla con el perfil del afiliado
Subflujos	ninguno
Excepciones	ninguno

Tabla: 53 Caso de uso: Ver afiliado

### **Caso de uso asignar patrocinador**

<b>Caso de uso</b>	<b>Asignar Patrocinador</b>
Actores	Administrador, Secretaria
Tipo	Extensión
Propósito	Asignar un patrocinador a un afiliado
Resumen	En este caso de uso el usuario puede asignar un patrocinador a un afiliado, mediante las listas de patrocinadores disponibles
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar Afiliado

Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona un afiliado sin patrocinador</li> <li>2. Elige la opción “asignar patrocinador”</li> <li>3. El sistema muestra una pantalla con los patrocinadores disponibles</li> <li>4. El usuario selecciona el patrocinador correspondiente. Y elige la opción “Asignar”</li> <li>5. El sistema asigna al patrocinador y afiliado a la base de datos</li> <li>6. Caso contrario se vuelve a la pantalla anterior</li> </ol>
Subflujos	ninguno
Excepciones	Solo se permite un patrocinador por afiliado

Tabla: 54 Caso de uso: Asignar patrocinador

#### **Caso de Uso** Gestionar Actividades

<b>Caso de uso</b>	<b>Listar Actividad</b>
Actores	Administrador, secretaria, educador
Tipo	Inclusión
Propósito	<p>Gestionar las actividades que se realizan en el Proyecto “Guadalquivir”</p> <ol style="list-style-type: none"> <li>1. Adicionar Actividad</li> <li>2. Modificar Actividad</li> <li>3. Eliminar Actividad</li> </ol>

Resumen	En este caso de uso el Usuario realiza la gestión de las actividades extracurriculares de las familias afiliadas; buscando, adicionando, modificando y eliminando los registros de las familias que cumplan con dichas actividades.
Precondiciones	Estar dentro del Sistema en la pantalla principal
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario Elige la opción “Gestionar Actividad”</li> <li>2. El sistema muestra un listado de las actividades registradas en el sistema</li> </ol>
Subflujos	Ninguno

Tabla: 55 Caso de uso: Listar actividad

### **Caso de Uso Adicionar Actividad**

<b>Caso de uso</b>	<b>Adicionar Actividad</b>
Actores	Administrador, secretaria, educador
Tipo	Extensión
Propósito	Adicionar Nueva Actividad extracurricular.
Resumen	En este caso de uso el Usuario adiciona nuevas Actividades extracurriculares.

Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar actividades
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Adicionar nueva actividad”</li> <li>2. El sistema muestra un formulario para llenar datos</li> <li>6. El usuario ingresan los datos</li> <li>7. Si el usuario elige la opción “Guardar” , el sistema adiciona la actividad a la base de datos</li> <li>3. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</li> </ol>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	no puede continuar porque la base de datos no permite campos vacíos

Tabla: 56 Caso de uso: Adicionar actividad

### **Caso de Uso Modificar Actividad**

<b>Caso de uso</b>	<b>Modificar Actividad</b>
Actores	Administrador, secretaria, educador
Tipo	Extensión
Propósito	Modificar datos de las actividades Extracurriculares.
Resumen	En este caso de uso el Administrador modifica los datos de las actividades extracurriculares que se realizan en el

	Proyecto “Guadalquivir”
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar actividades y adicionar actividad
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la actividad que desee modificar</li> <li>2. El usuario selecciona la opción Modificar</li> <li>3. El sistema muestra un formulario para modificar datos</li> <li>4. El usuario ingresan los datos</li> <li>5. Si el usuario elige la opción “Modificar” , el sistema modifica los datos en la base de datos</li> <li>6. Si el usuario elige la opción “Cancelar”, se regresa a la pantalla anterior</li> </ol>
Subflujos	Si los datos ingresados no son válidos el sistema muestra un mensaje respectivo al error
Excepciones	<p>no puede continuar porque la base de datos no permite campos vacíos</p> <p>Mensaje: ¿Seguro de Modificar?</p>

Tabla: 57 Caso de uso: Modificar Actividad

### Caso de uso Eliminar Actividad

<b>Caso de uso</b>	<b>Eliminar Actividad</b>
Actores	Administrador, secretaria, educador

Tipo	Extensión
Propósito	Eliminar una Actividad Extracurricular.
Resumen	En este caso de uso el Administrador Elimina a una Actividad Extracurricular.
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar actividades y adicionar actividad
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la actividad que desee eliminar</li> <li>2. Elige la opción “eliminar”</li> <li>3. Si el usuario acepta la eliminación, el sistema eliminara la actividad caso contrario volverá a la pantalla anterior</li> </ol>
Subflujos	
Excepciones	Verificación de seguridad al momento de eliminar un dato. Mediante un mensaje emergente de confirmación

Tabla: 58 Caso de uso: Eliminar actividad

### Caso de uso ver actividad

<b>Caso de uso</b>	<b>Ver Actividad – Ver asistencia</b>
Actores	Administrador, secretaria, educador
Tipo	Extensión

Propósito	Ver los datos de la actividad
Resumen	En este caso de uso el Usuario puede ver los datos de una actividad en específico con una lista de los asistentes a dicha actividad
Precondiciones	Se requiere haber ejecutado anteriormente el caso de uso Listar actividades y adicionar actividad
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la actividad que desee ver</li> <li>2. Elige la opción “ver actividad”</li> <li>3. El sistema muestra los datos de la actividad</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 59 Caso de uso: ver asistencia

#### **Caso de Uso** Controlar Asistencia

<b>Caso de uso</b>	<b>Listar Actividades</b>
Actores	Administrador, secretaria, educador
Tipo	Básico

Propósito	Controlar las actividades existentes, para que se pueda controlar la asistencia a dichas actividades.
Resumen	En este caso de uso el Usuario realiza la gestión de la asistencia a las actividades que se realizan al Proyecto “Guadalquivir”.
Precondiciones	Estar dentro del sistema. En la pantalla principal del sistema
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Controlar Asistencia”</li> <li>2. El sistema visualiza la lista de todas las actividades.</li> <li>3. El usuario elige la actividad que desee registrar asistencia</li> <li>4. Selecciona la opción “marcar asistencia”</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 60 Caso de uso: Listar actividades

### **Caso de Uso Listar Familia**

<b>Caso de uso</b>	<b>Listar Familia</b>
Actores	Administrador, secretaria, educador
Tipo	Inclusión.
Propósito	Listar las familias afiliadas en el Proyecto.

Resumen	Lista a las familias afiliadas al Proyecto “Guadalquivir” para registrar su asistencia a las actividades que se realizan cada gestión.
Precondiciones	Se requiere haber ejecutado el caso de uso Controlar Asistencia
Flujo principal	1. El sistema Muestra un listado de las familias registradas
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 61 Caso de uso: Listar familia

### Caso de Uso Registrar Asistencia

<b>Caso de uso</b>	<b>Registrar Asistencia</b>
Actores	Administrador, secretaria, educador
Tipo	Inclusión.

Propósito	Controlar la asistencia a las actividades realizadas.
Resumen	En este caso de uso el Usuario realiza el registro de la asistencia de las familias a las actividades que se realizan al Proyecto “Guadalquivir”.
Precondiciones	Se requiere haber ejecutado el caso de uso Listar actividades y listar familia
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona las familias asistentes a la actividad</li> <li>2. El usuario selecciona la opción “Guardar” para registrar la asistencia de las familias seleccionadas</li> <li>3. El sistema guarda los datos en la base de datos</li> <li>4. Si el usuario no desea registrar cambios presiona la opción “Cancelar” y se vuelve a la página anterior</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 62 Caso de uso: Registrar asistencia

### **Caso de Uso Gestionar Salud**

<b>Caso de uso</b>	<b>Listar afiliado</b>
--------------------	------------------------

Actores	Administrador
Tipo	Básico
Propósito	Registrar datos generales de salud de los afiliados
Resumen	En este Caso de uso el Administrador Puede registrar datos correspondientes a la salud de los afiliados, para tener un control más exacto sobre la información en cuanto a la salud.
Precondiciones	Haber Ejecutado el caso de uso Ingresar al Sistema, estar dentro de la pantalla principal del sistema
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Controlar Estado de Salud”</li> <li>2. El sistema muestra una lista de todos los afiliados al sistema</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 63 Caso de uso: Listar afiliado

#### **Caso de uso Buscar afiliado**

<b>Caso de uso</b>	<b>Buscar afiliados</b>
Actores	Administrador, secretaria

Tipo	extensión
Propósito	Buscar a los afiliados del Proyecto
Resumen	En este Caso de uso el usuario puede realizar una búsqueda del afiliado que se requiera
Precondiciones	Haber Ejecutado el caso de uso Ingresar al Sistema, y el caso de uso Controlar estado de salud
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario ingresa los datos de apellido paterno</li> <li>2. Selecciona la opción “buscar”</li> <li>3. El sistema muestra un listado con los resultados de la búsqueda</li> </ol>
Subflujos	Ninguno
Excepciones	Si el sistema no reconoce el dato insertado, no se muestra nada

Tabla: 64 Caso de uso: Buscar Afiliado

### **Caso de uso Llenar formulario**

<b>Caso de uso</b>	<b>Llenar Formulario</b>
Actores	Administrador, secretaria

Tipo	extensión
Propósito	Ingresar datos referentes a la salud del afiliado
Resumen	En este caso de uso el usuario puede guardar datos referentes a la salud básica de cada afiliado del proyecto
Precondiciones	Haber Ejecutado el caso de uso Ingresar al Sistema, y el caso de uso Controlar estado de salud
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona el afiliado al que desea registrar datos de salud</li> <li>2. El usuario elige la opción “Llenar Formulario”</li> <li>3. El sistema muestra un formulario para introducir datos</li> <li>4. El usuario ingresa los datos correspondientes</li> <li>5. Si el usuario desea guardar la información selecciona la opción “Guardar” caso contrario selecciona la opción “Cancelar” y se vuelve a la pantalla anterior</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 65 Caso de Uso Llenar Formulario

### **Caso de Uso Ver Formulario**

<b>Caso de uso</b>	<b>Ver Formulario</b>
--------------------	-----------------------

Actores	Administrador, secretaria
Tipo	extensión
Propósito	Ver datos de formulario
Resumen	En este Caso de uso el usuario puede ver los datos del formulario de salud del afiliado
Precondiciones	Haber Ejecutado el caso de uso Ingresar al Sistema, y el caso de uso Controlar estado de salud
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona el afiliado que desee revisar los datos de salud</li> <li>2. El usuario elige la opción “Ver Formulario”</li> <li>3. El sistema muestra un formulario con los datos de salud del afiliado seleccionado</li> <li>4. Si el usuario elige la opción “Volver” se regresa a la pantalla anterior</li> </ol>
Subflujos	Ninguno
Excepciones	El formulario es estático, sin opciones de modificar

Tabla: 66 Caso de uso Ver Formulario

### **Caso de Uso Registrar Nivel Educativo**

<b>Caso de uso</b>	<b>Listar afiliado</b>
Actores	Administrador
Tipo	Básico
Propósito	Listar a todos los afiliados del proyecto
Resumen	En este Caso de uso el administrador puede visualizar a los afiliados
Precondiciones	Haber ejecutado el caso de uso ingresar al sistema y estar dentro de la pantalla principal
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la opción “Registrar nivel educativo”</li> <li>2. El sistema muestra un listado de los afiliados registrados</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 67 Caso de uso Listar Afiliado

<b>Caso de uso</b>	<b>Buscar afiliados</b>
--------------------	-------------------------

Actores	Administrador, secretaria
Tipo	extensión
Propósito	Buscar a los afiliados del Proyecto
Resumen	En este Caso de uso el usuario puede realizar una búsqueda del afiliado que se requiera
Precondiciones	Haber Ejecutado el caso de uso Ingresar al Sistema, y el caso de uso Controlar estado de salud
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario ingresa los datos de apellido paterno</li> <li>2. Selecciona la opción “buscar”</li> <li>3. El sistema muestra un listado con los resultados de la búsqueda</li> </ol>
Subflujos	Ninguno
Excepciones	Si el sistema no reconoce el dato insertado, no se muestra nada

Tabla: 68 Caso de uso Buscar formulario

### **Caso de uso Llenar formulario**

<b>Caso de uso</b>	<b>Llenar Formulario de registro</b>
Actores	Administrador, secretaria
Tipo	Inclusión
Propósito	Llenar datos de nivel educativo
Resumen	En este Caso de uso el administrador puede llenar un formulario preestablecido con datos sobre el nivel educativo de los afiliados
Precondiciones	Haber ejecutado el caso de uso Listar afiliado
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona el afiliado al que desea registrar datos de educación</li> <li>2. El usuario elige la opción “Llenar Formulario”</li> <li>3. El sistema muestra un formulario para introducir datos</li> <li>4. El usuario ingresa los datos correspondientes</li> <li>5. Si el usuario desea guardar la información selecciona la opción “Guardar” caso contrario selecciona la opción “Cancelar” y se vuelve a la pantalla anterior</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 69 Caso de uso: Llenar formulario de registro

<b>Caso de uso</b>	<b>Ver Formulario de Registro</b>
--------------------	-----------------------------------

Actores	Administrador
Tipo	Extensión
Propósito	Ver los datos de nivel académico de los afiliados al proyecto
Resumen	En este Caso de uso el administrador puede ver los datos de nivel académico de los afiliados al proyecto
Precondiciones	Haber ejecutado el caso de uso Listar afiliado
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona el afiliado que desee revisar los datos de nivel académico</li> <li>2. El usuario elige la opción “Ver Formulario”</li> <li>3. El sistema muestra un formulario con los datos de salud del afiliado seleccionado</li> <li>4. Si el usuario elige la opción “Volver” se regresa a la pantalla anterior</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 70 Caso de uso: Ver formulario de Registro

### **Caso de Uso Generar Reporte**

<b>Caso de uso</b>	<b>Generar Reporte</b>
Actores	Administrador, secretaria
Tipo	extensión
Propósito	Generar reportes específicos
Resumen	En este Caso de uso permite al usuario generar diferentes tipos de reportes específicos
Precondiciones	Haber Ejecutado el caso de uso Ingresar al Sistema
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario selecciona la opción “Generar Reportes”</li> <li>2. El sistema muestra las opciones válidas para generar reportes</li> <li>3. El usuario elige un tipo de reporte</li> <li>4. El sistema muestra un formulario para llenar datos requeridos para generar el reporte</li> <li>5. El usuario llena el formulario,</li> <li>6. Si el usuario elige la opción “aceptar” el sistema despliega el reporte</li> <li>7. Caso contrario se vuelve a la pantalla anterior</li> </ol>
Subflujos	Ninguno
Excepciones	Si el sistema no reconoce el dato insertado, no se muestra nada

Tabla: 71 Caso de uso: generar reporte

### **Caso de uso Backup BD**

<b>Caso de uso</b>	<b>Backup BD</b>
Actores	Administrador
Tipo	extensión
Propósito	Hacer un respaldo de la base de datos
Resumen	En este Caso de uso el usuario puede realizar un backup de la base de datos para evitar perdida de información
Precondiciones	Haber Ejecutado el caso de uso Ingresar al Sistema
Flujo principal	<ol style="list-style-type: none"> <li>1. El usuario elige la opción “Backup BD”</li> <li>2. El sistema realiza un backup automático de la base de datos y lo guarda en una carpeta especifica</li> </ol>
Subflujos	Ninguno
Excepciones	Ninguno

Tabla: 72 Caso de uso: backup

## **I.1.6 Análisis y diseño**

### **I.1.6.1 MODELO DE DIAGRAMA DE ACTIVIDADES**

#### ***I.1.6.1.1 Introducción***

Un diagrama de actividades muestra la interacción de un conjunto de objetos en una aplicación a través de las actividades realizadas y se modela para los casos de uso más relevantes. En un diagrama de actividades, se puede mostrar el flujo de los datos que pasan de una acción a otra. Sin embargo, en un diagrama de actividades no se describe la estructura de los datos.

#### ***I.1.6.1.2 Propósito***

- Comprender la estructura y la dinámica del sistema deseado para la organización.
- Mostrar operaciones que se pasan entre objetos.

#### ***I.1.6.1.3 Alcance***

- Describe los procesos de sistema y los clientes.
- Identificar y definir los procesos de los casos de uso según los objetivos de la organización.
- Definir un diagrama de actividad para los casos de uso más importantes del sistema.

### 1.1.6.1.4 DIAGRAMA DE ACTIVIDAD

#### 1.1.6.1.4.1 Ingresar al sistema

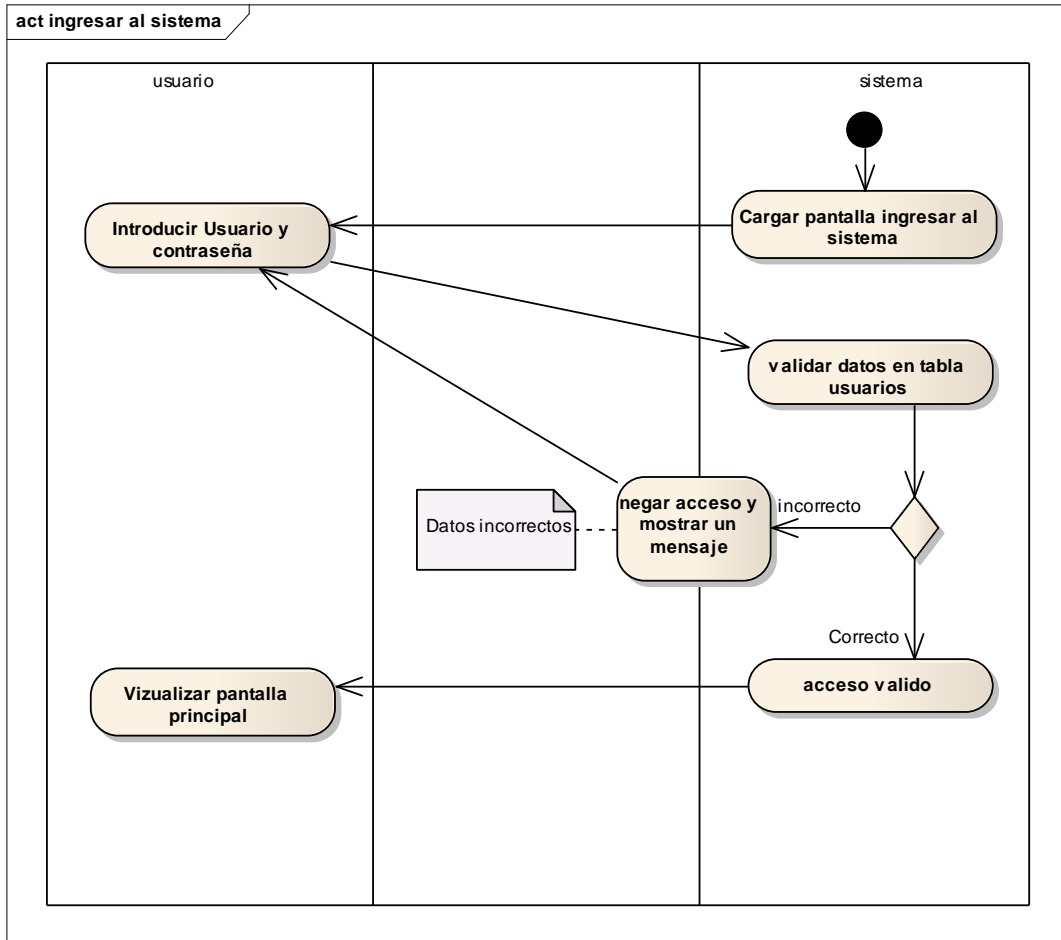


Figura N° 21 Diagrama de actividad: Ingresar al sistema

*1.1.6.1.4.2 Gestionar usuarios*

**Listar usuario**

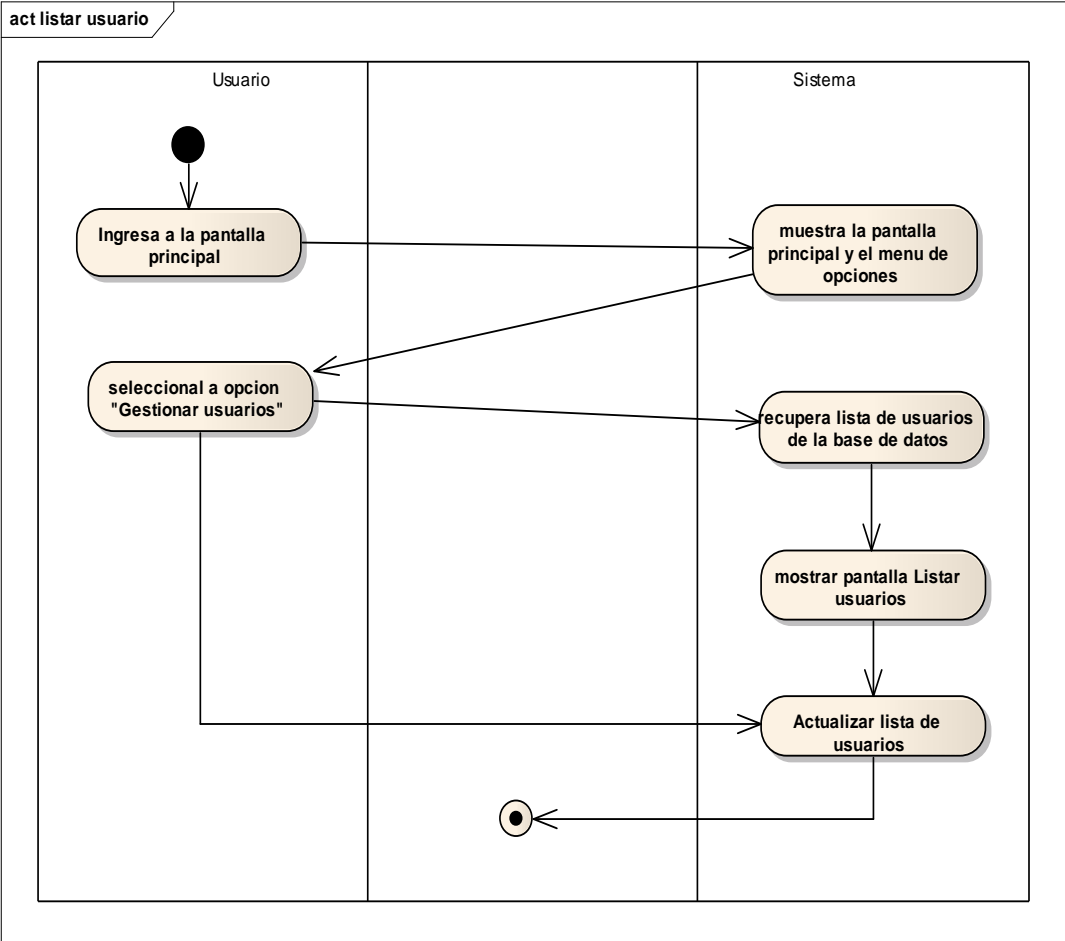


Figura N° 22 Diagrama de actividad: Listar Usuario

## Adicionar usuario

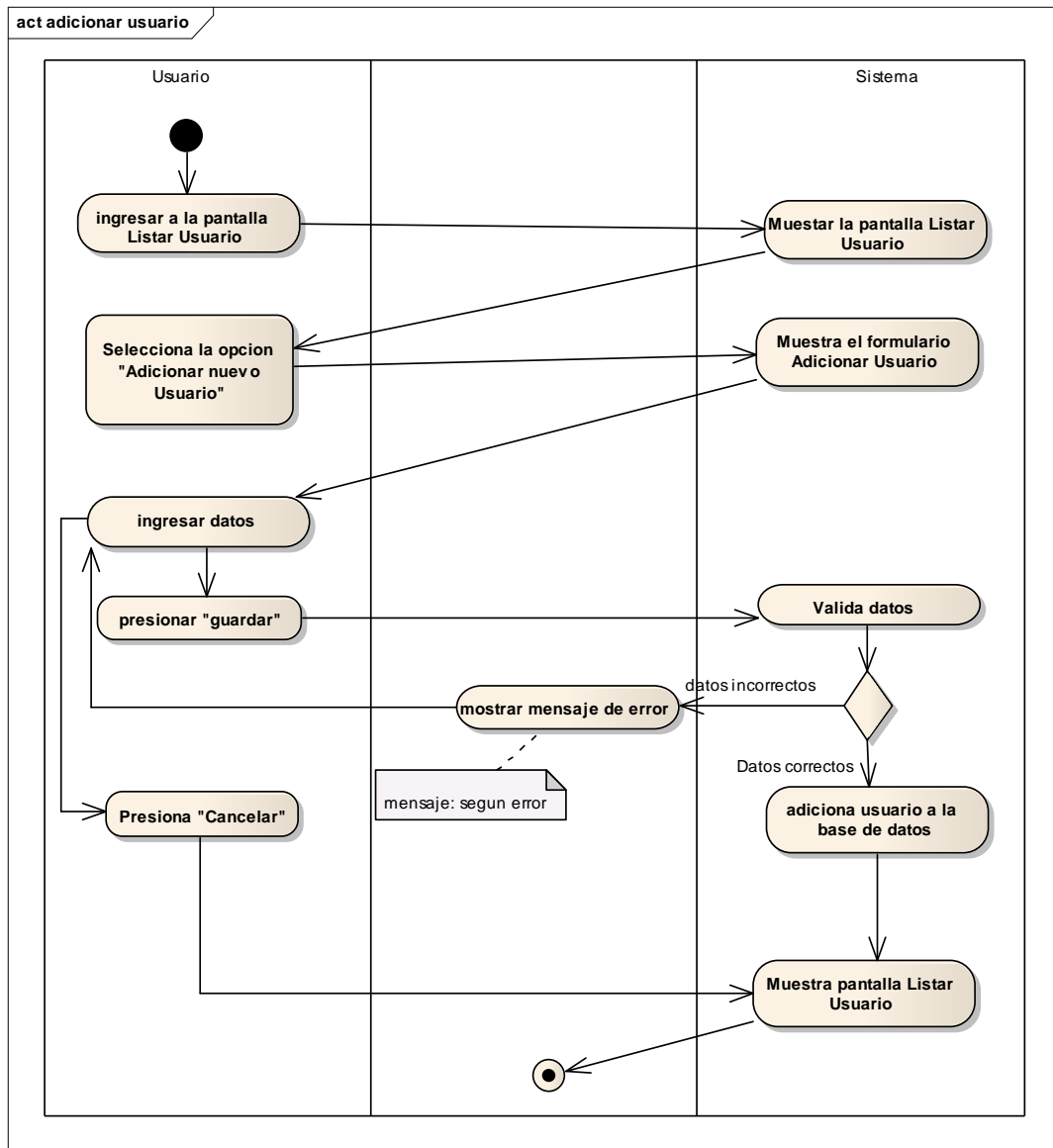


Figura N° 23 Diagrama de actividad: adicionar Usuario

## Modificar Usuarios

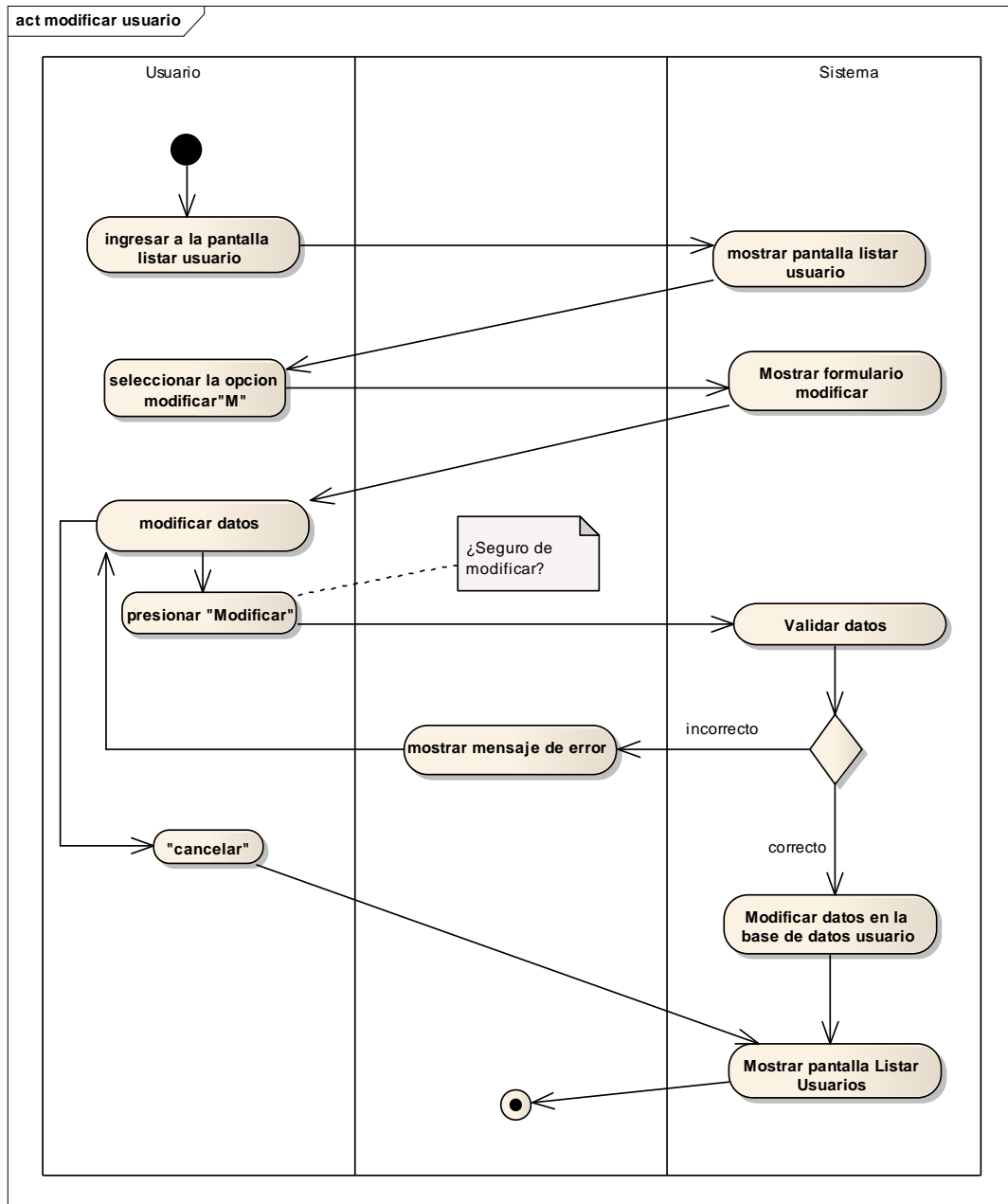


Figura N° 24 Diagrama de actividad: modificar Usuario

## Eliminar Usuario

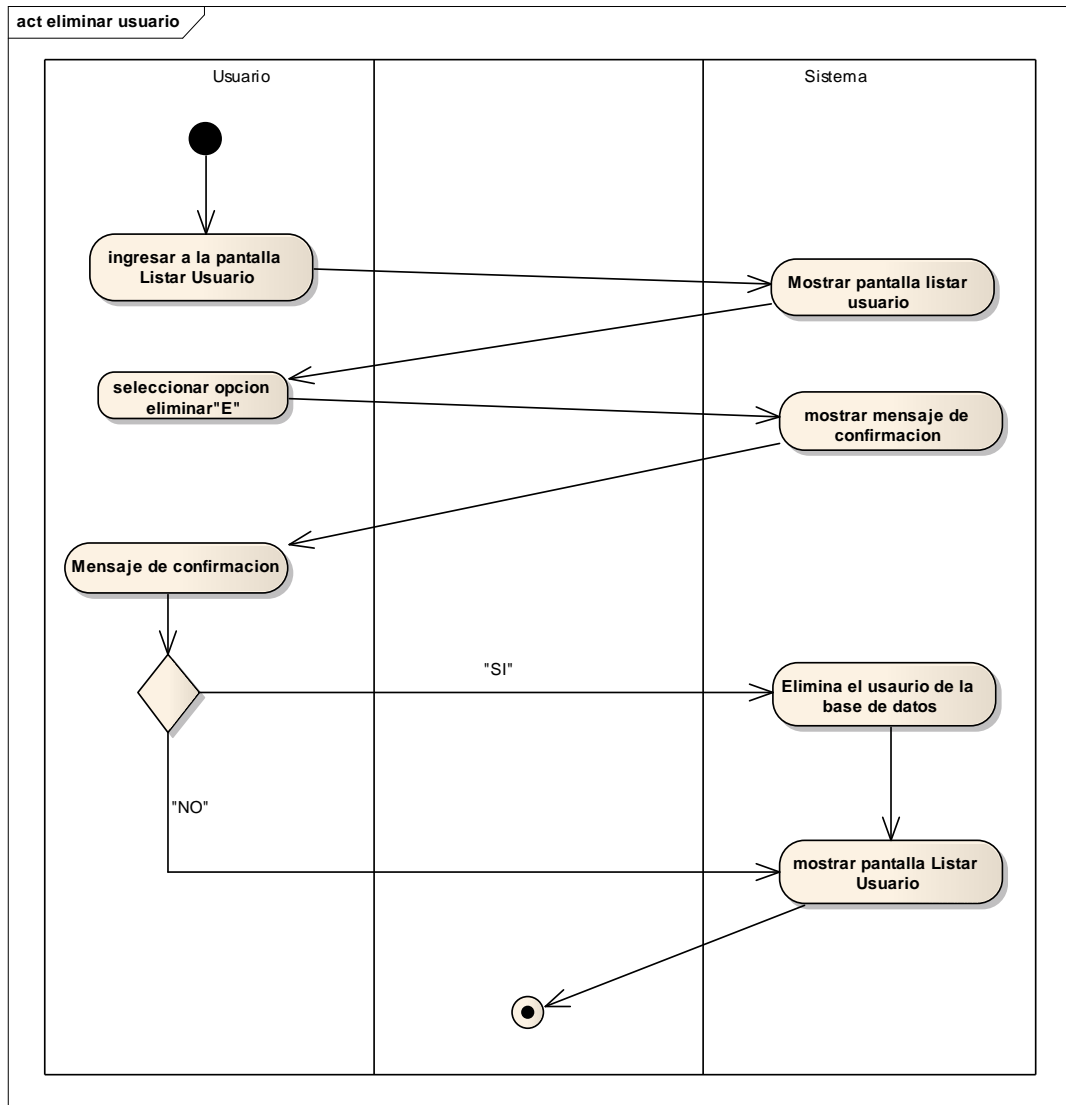


Figura N° 25 Diagrama de actividad: Eliminar Usuario

## Buscar Usuario

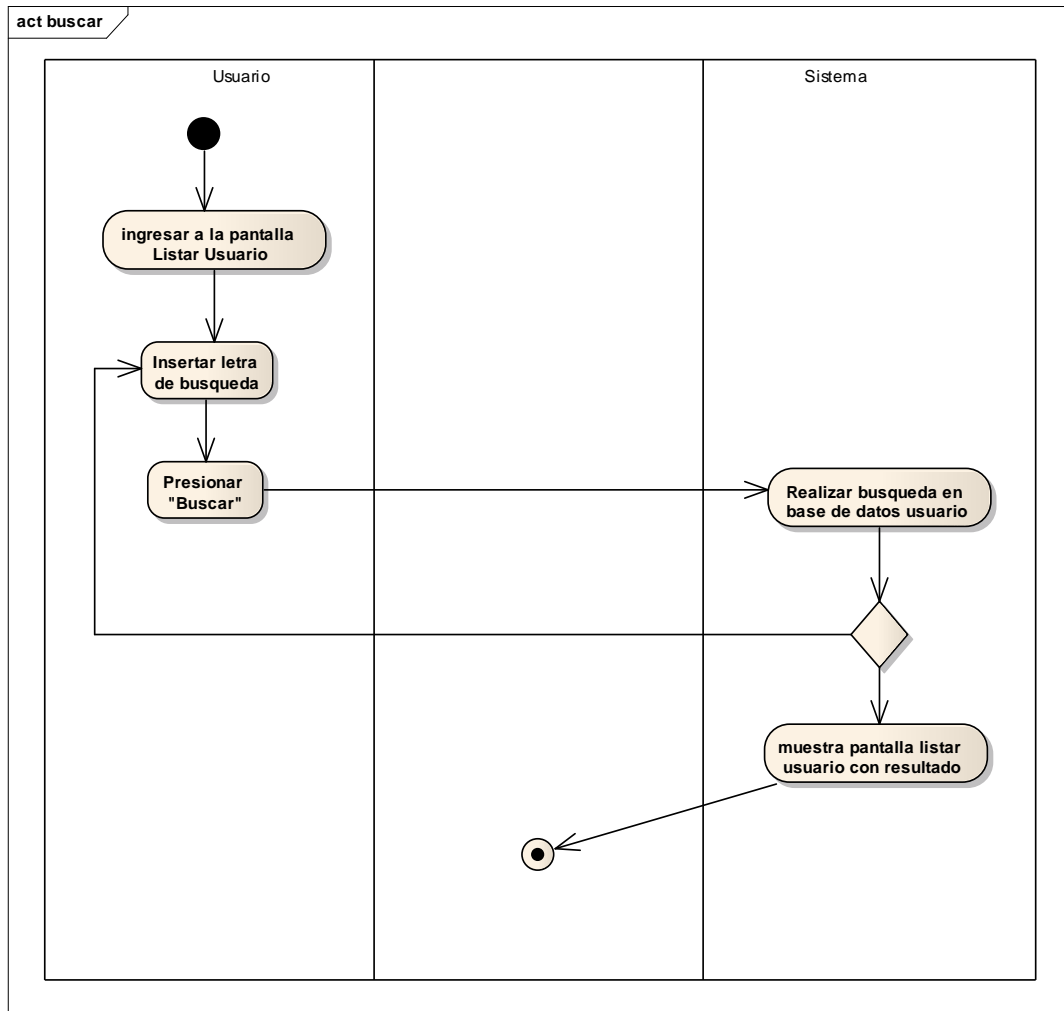


Figura N° 26 Diagrama de actividad: Buscar Usuario

## Ver Usuario

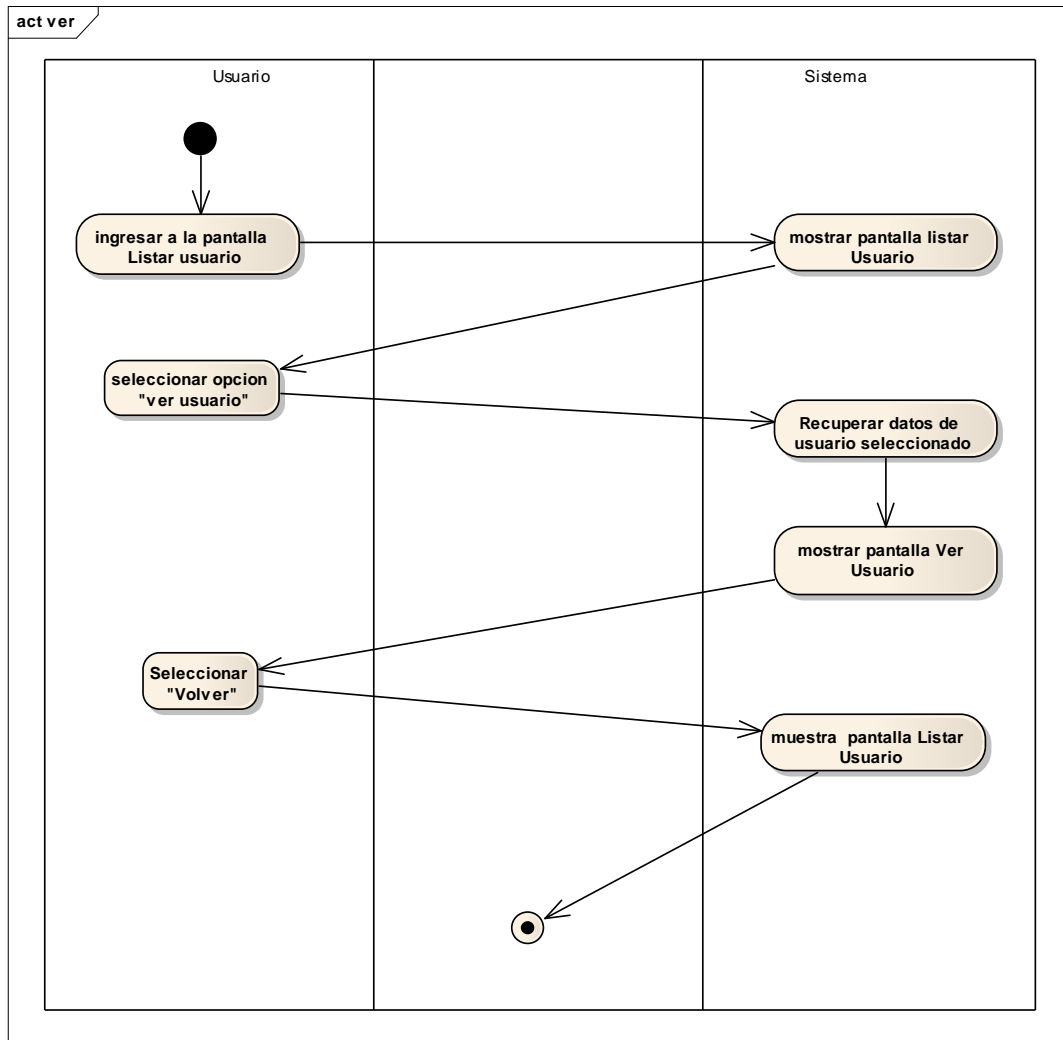


Figura N° 27 Diagrama de actividad: Ver Usuario

### 1.1.6.1.4.3 Gestionar familia

#### Listar familia

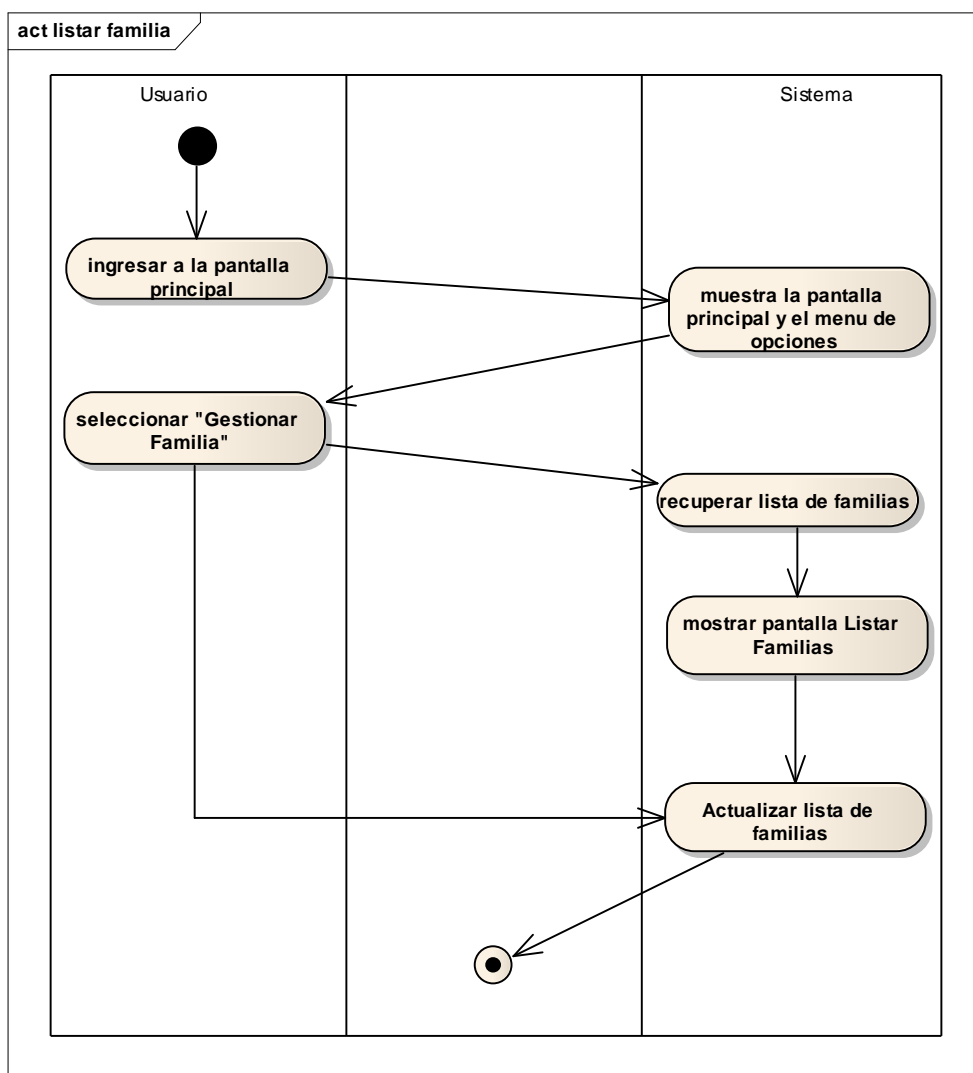


Figura N° 28 Diagrama de actividad: Listar familia

## Adicionar Familia

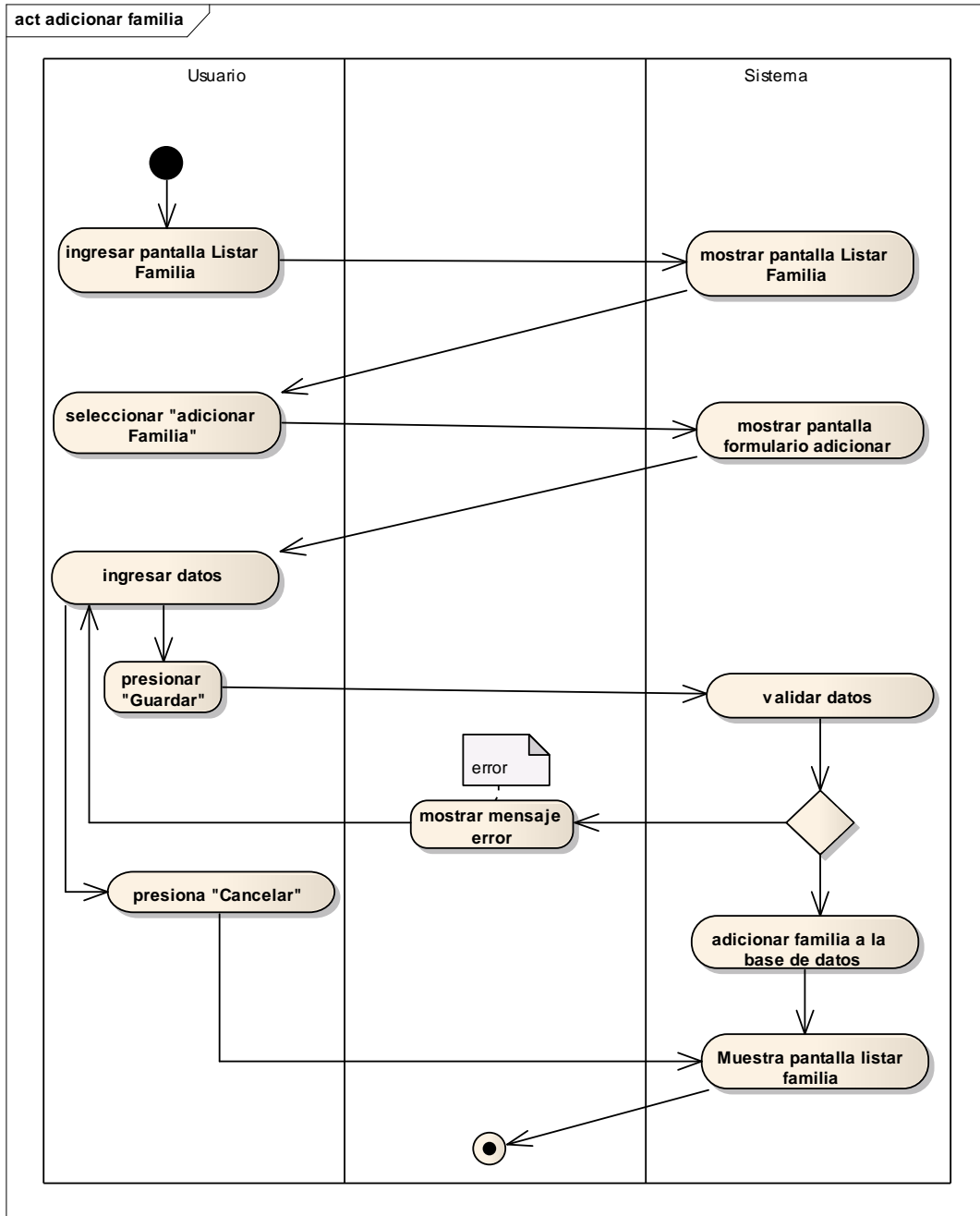


Figura N° 29 Diagrama de actividad: Adicionar familia

## Modificar familia

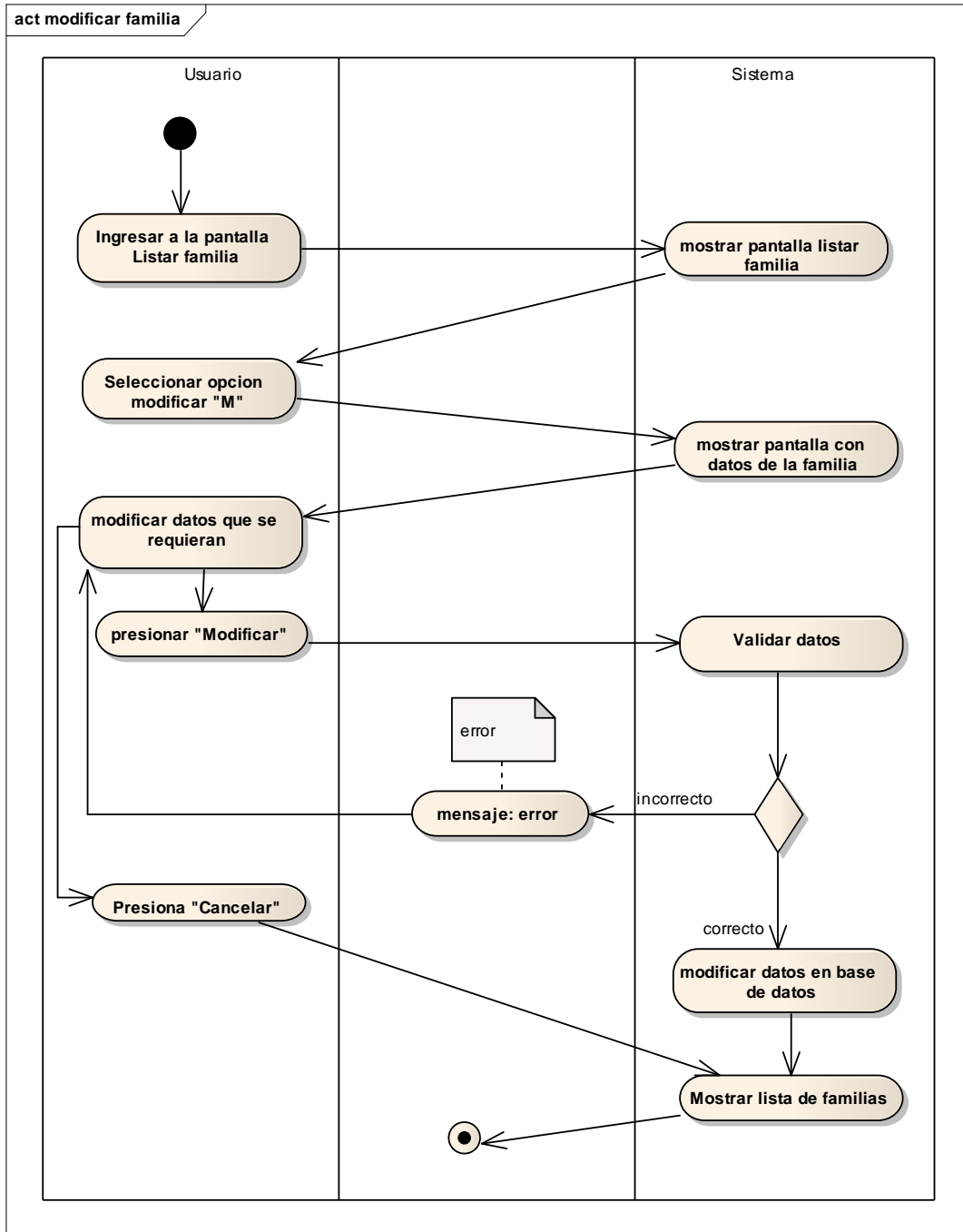


Figura N° 30 Diagrama de actividad: modificar familia

## Eliminar Familia

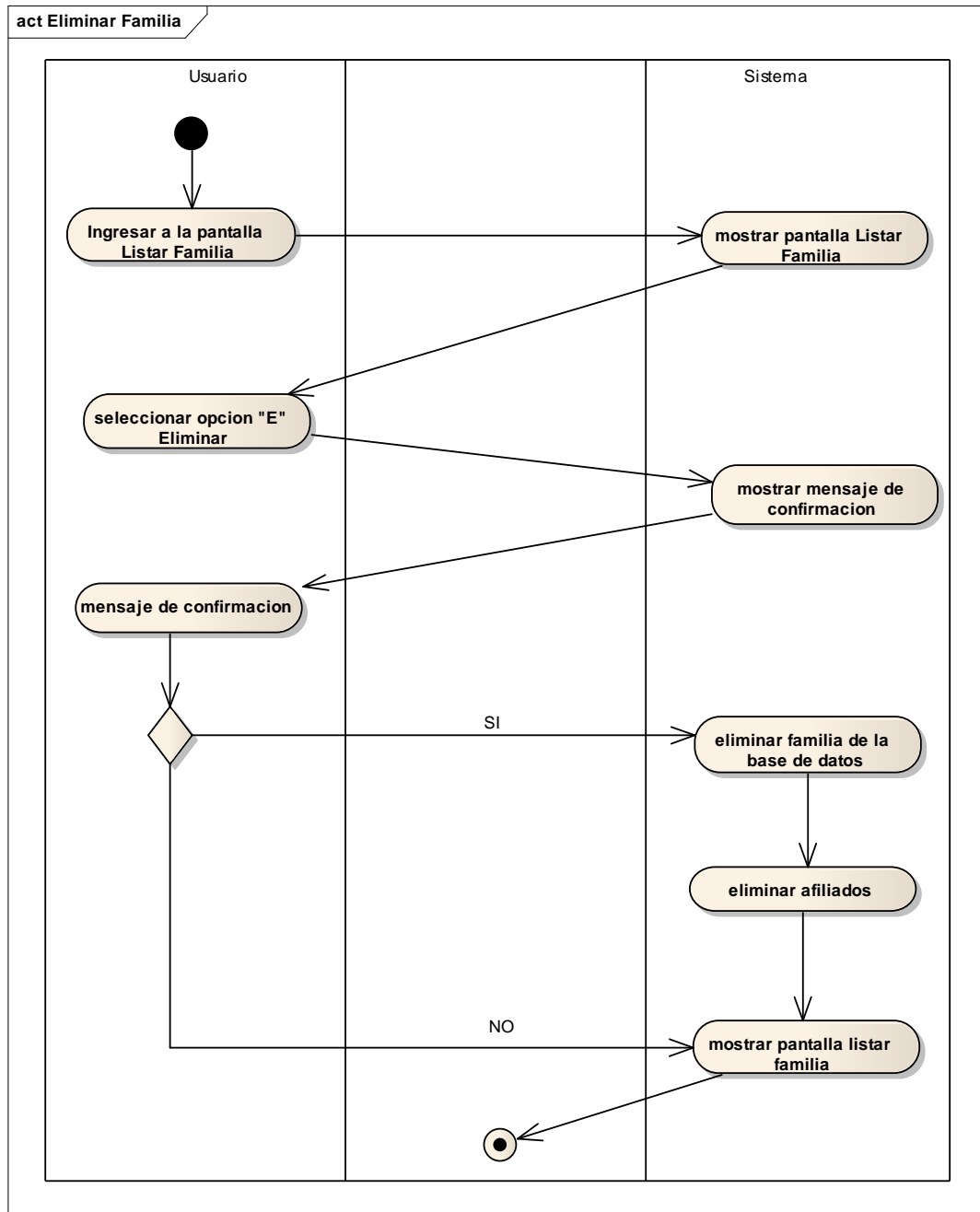


Figura N° 31 Diagrama de actividad: Eliminar familia

## Ver familia

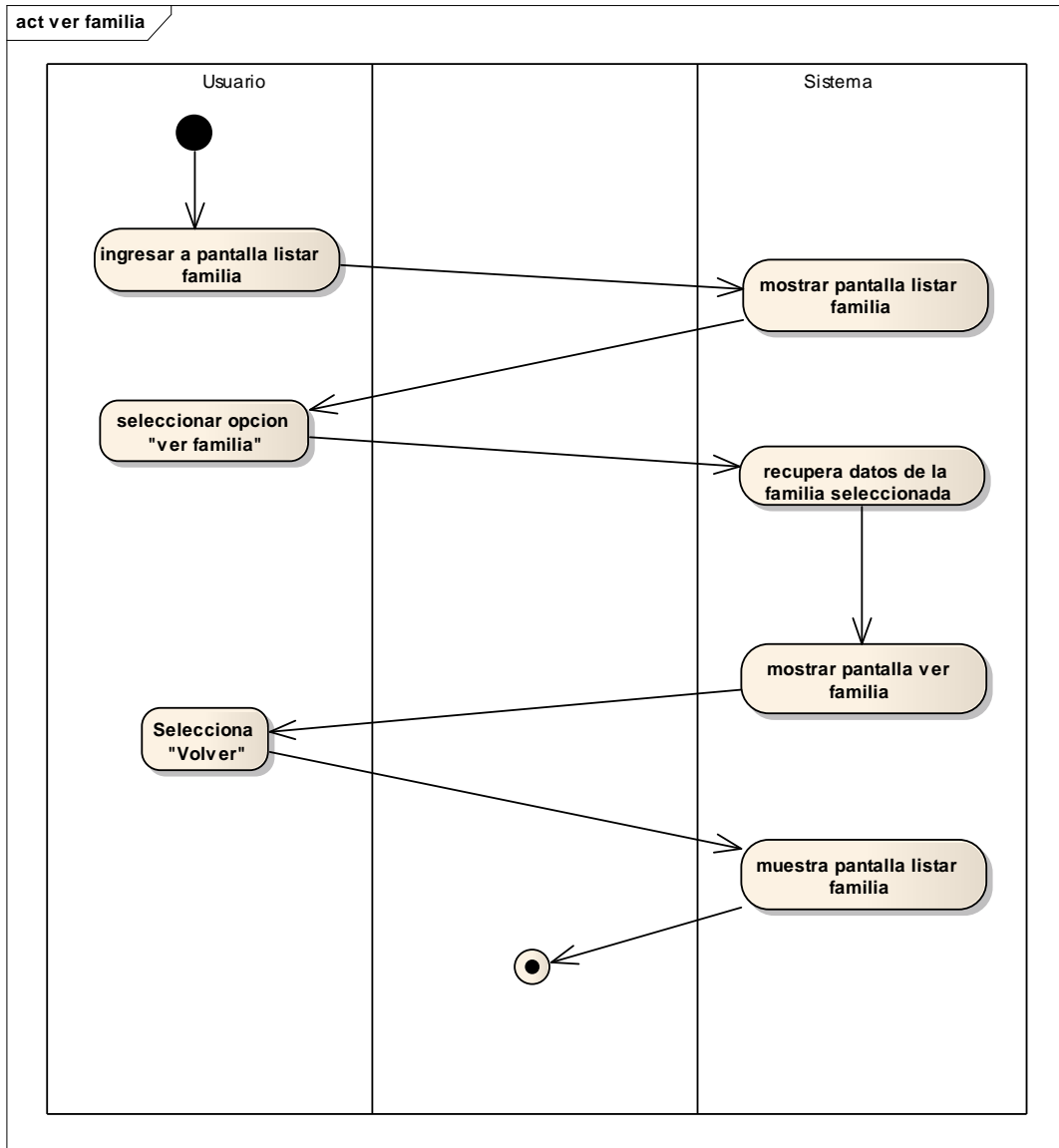


Figura N° 32 Diagrama de actividad: Ver familia

*I.1.6.1.4.4 Caso de uso: Gestionar Familia Candidata*

**Listar Familia candidata**

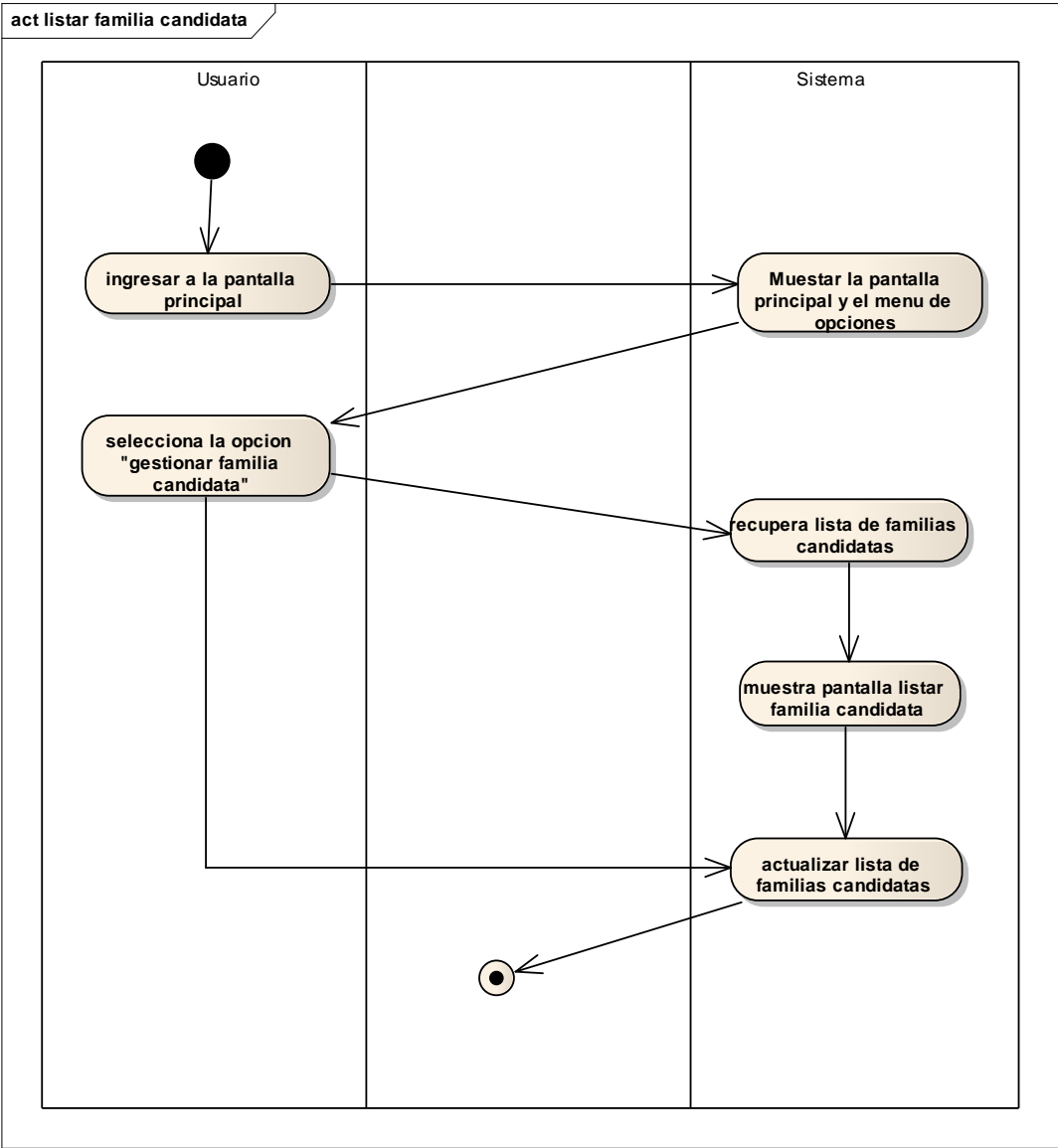


Figura N° 33 Diagrama de actividad: Listar familia candidata

## Adicionar Familia candidata

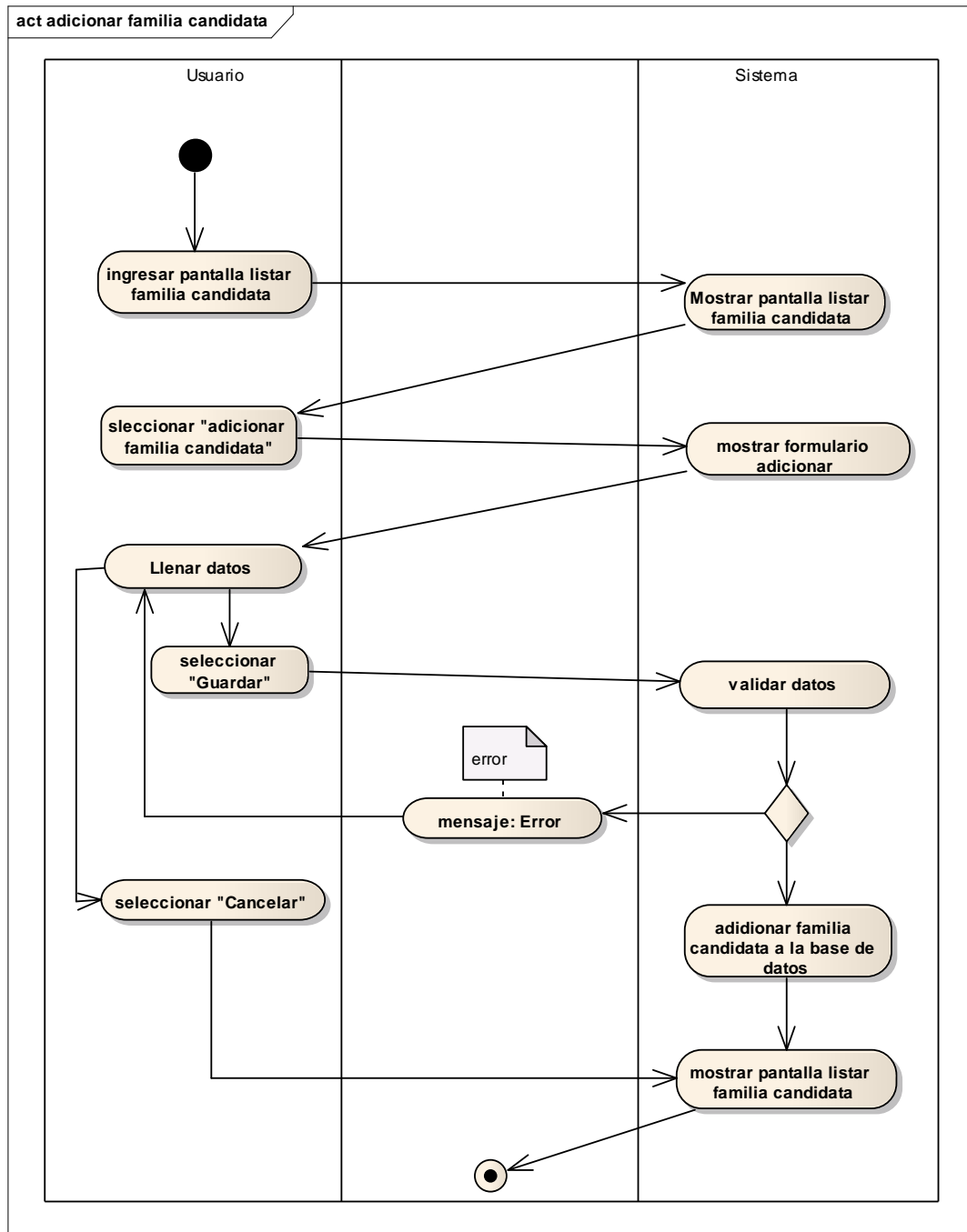


Figura N° 34 Diagrama de actividad: Adicionar familia candidata

## Eliminar Familia candidata

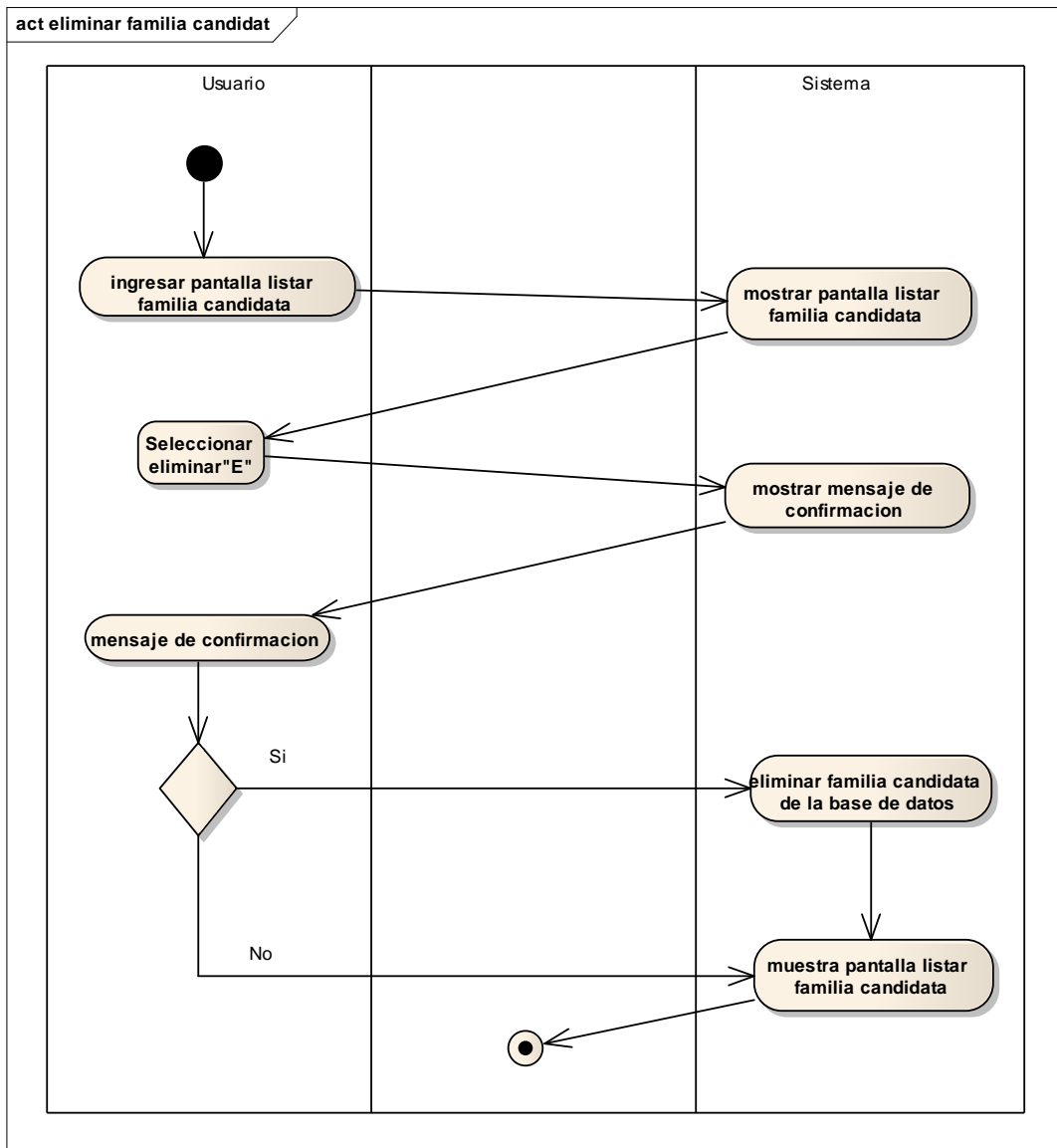


Figura N° 35 Diagrama de actividad: Eliminar familia candidata

*1.1.6.1.4.5 Caso de uso: Gestionar Patrocinadores*

**Listar Patrocinadores**

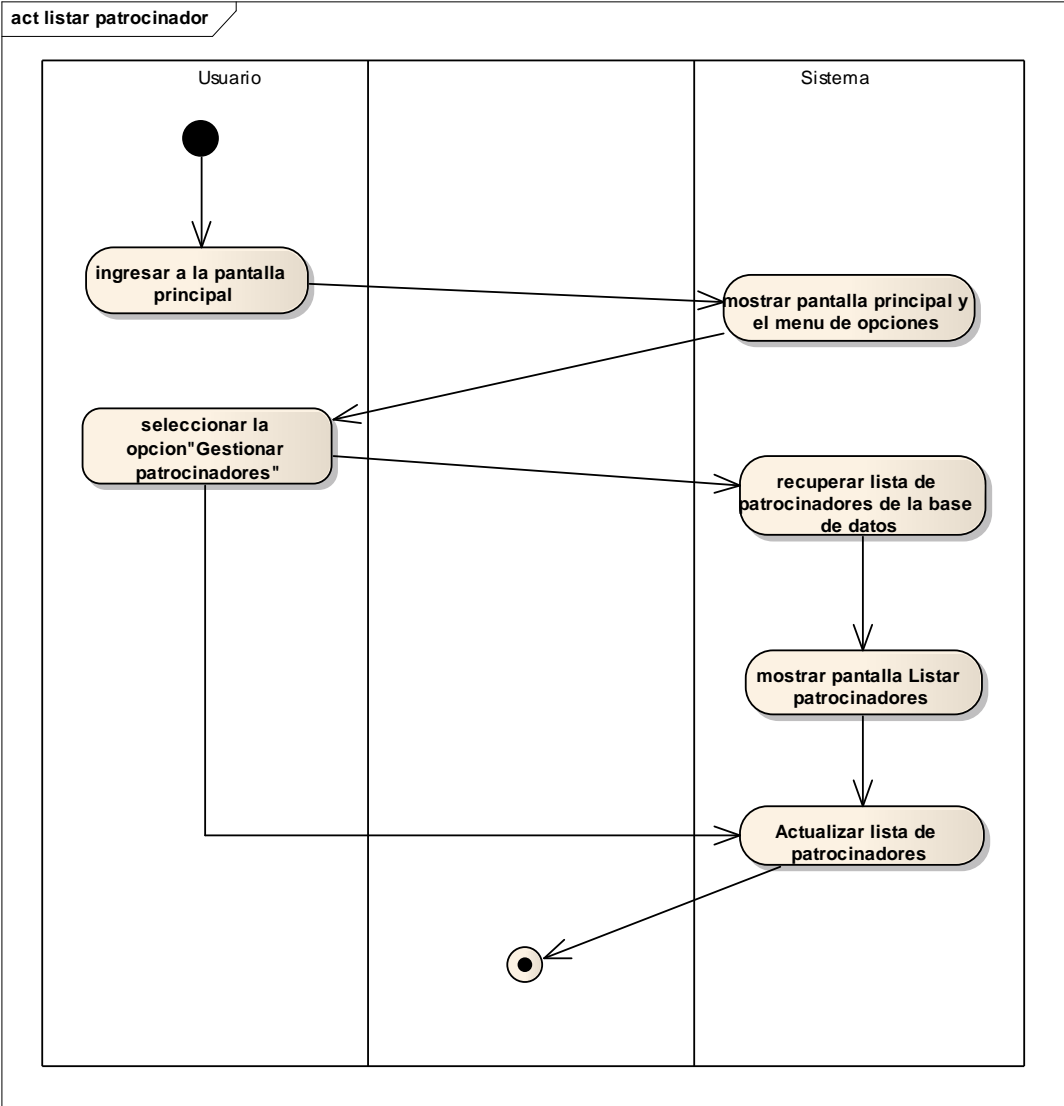


Figura N° 36 Diagrama de actividad: listar patrocinador

## Buscar Patrocinador

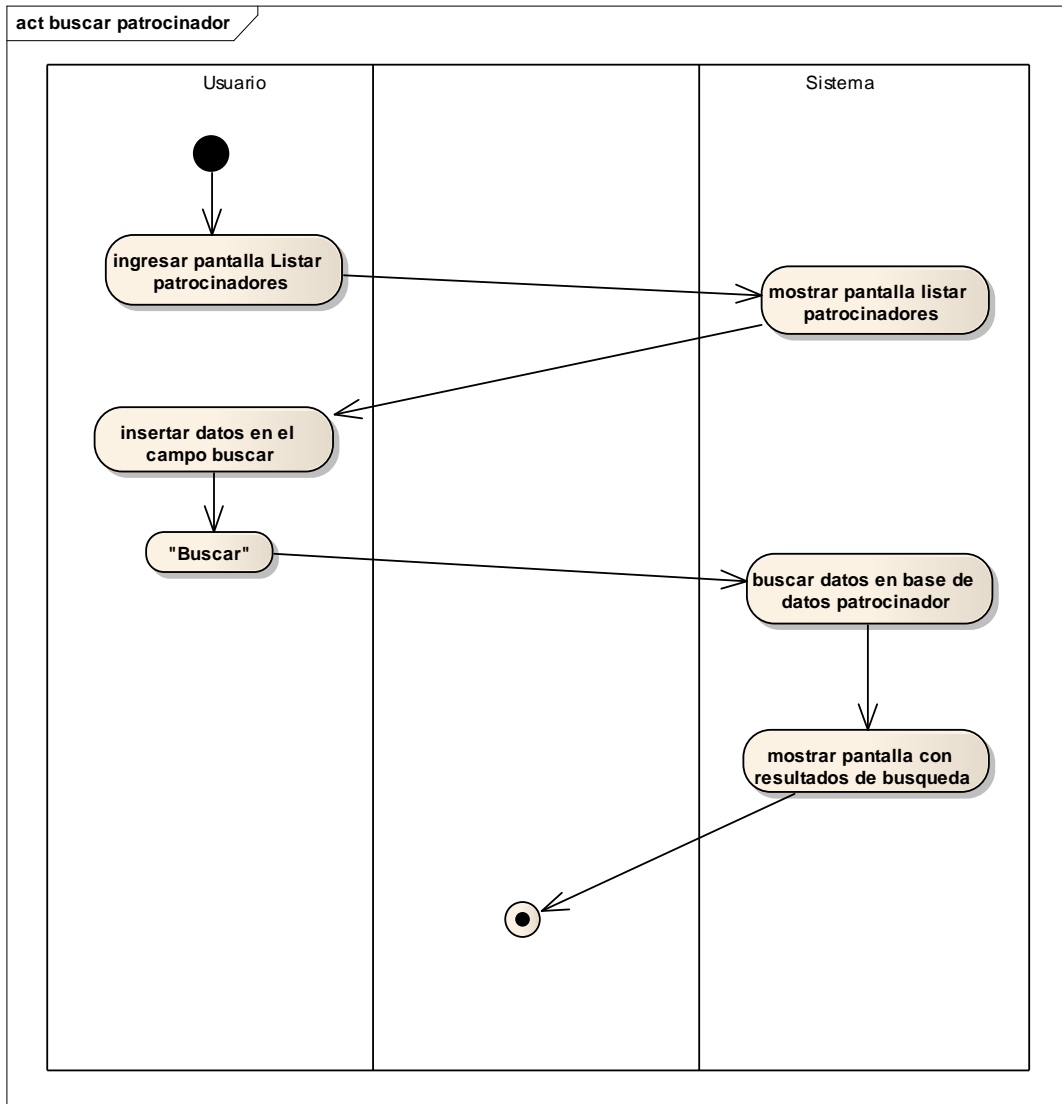


Figura N° 37 Diagrama de actividad: Buscar patrocinador

## Adicionar patrocinador

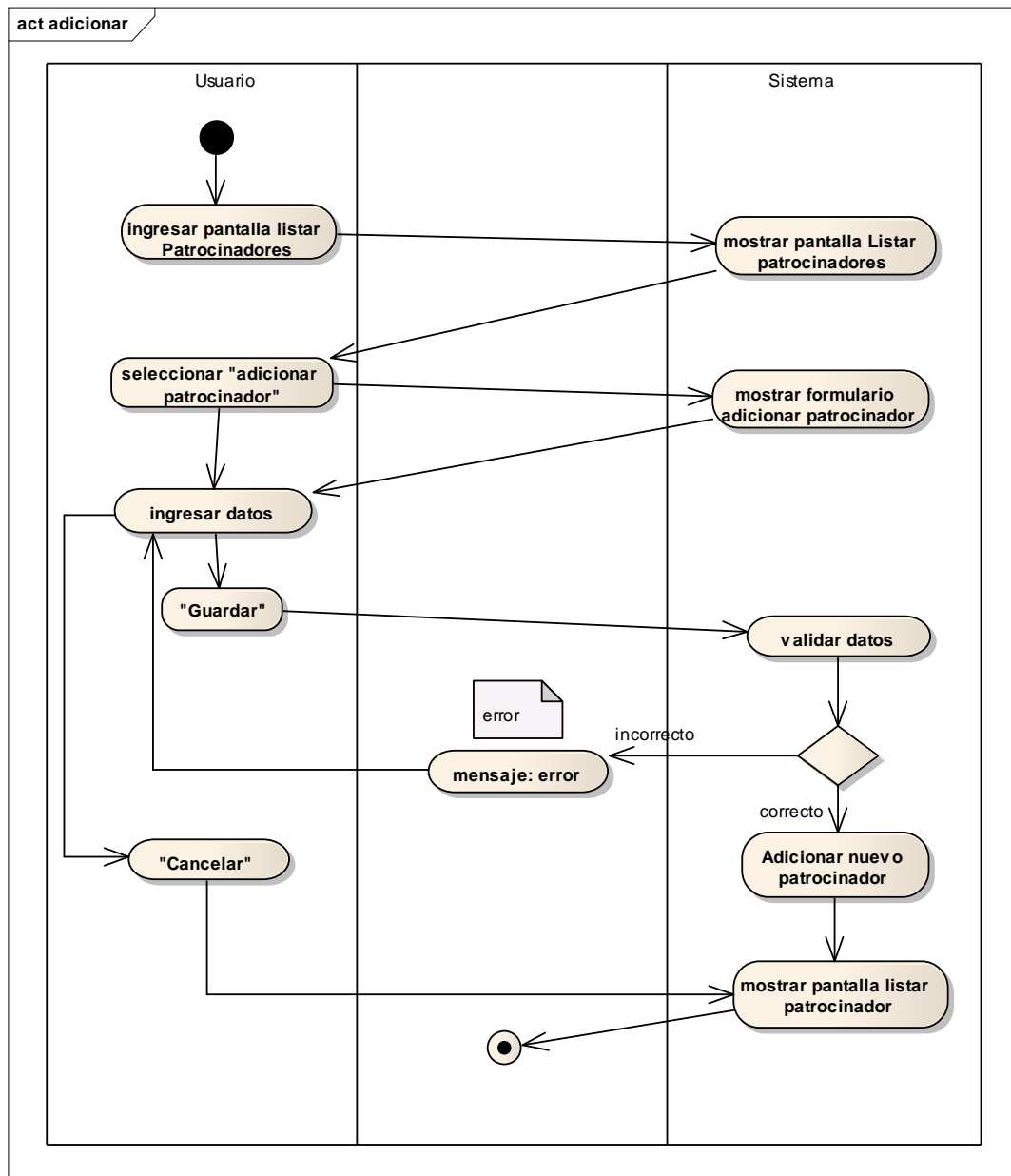


Figura N° 38 Diagrama de actividad: Adicionar patrocinadores

## Modificar Patrocinador

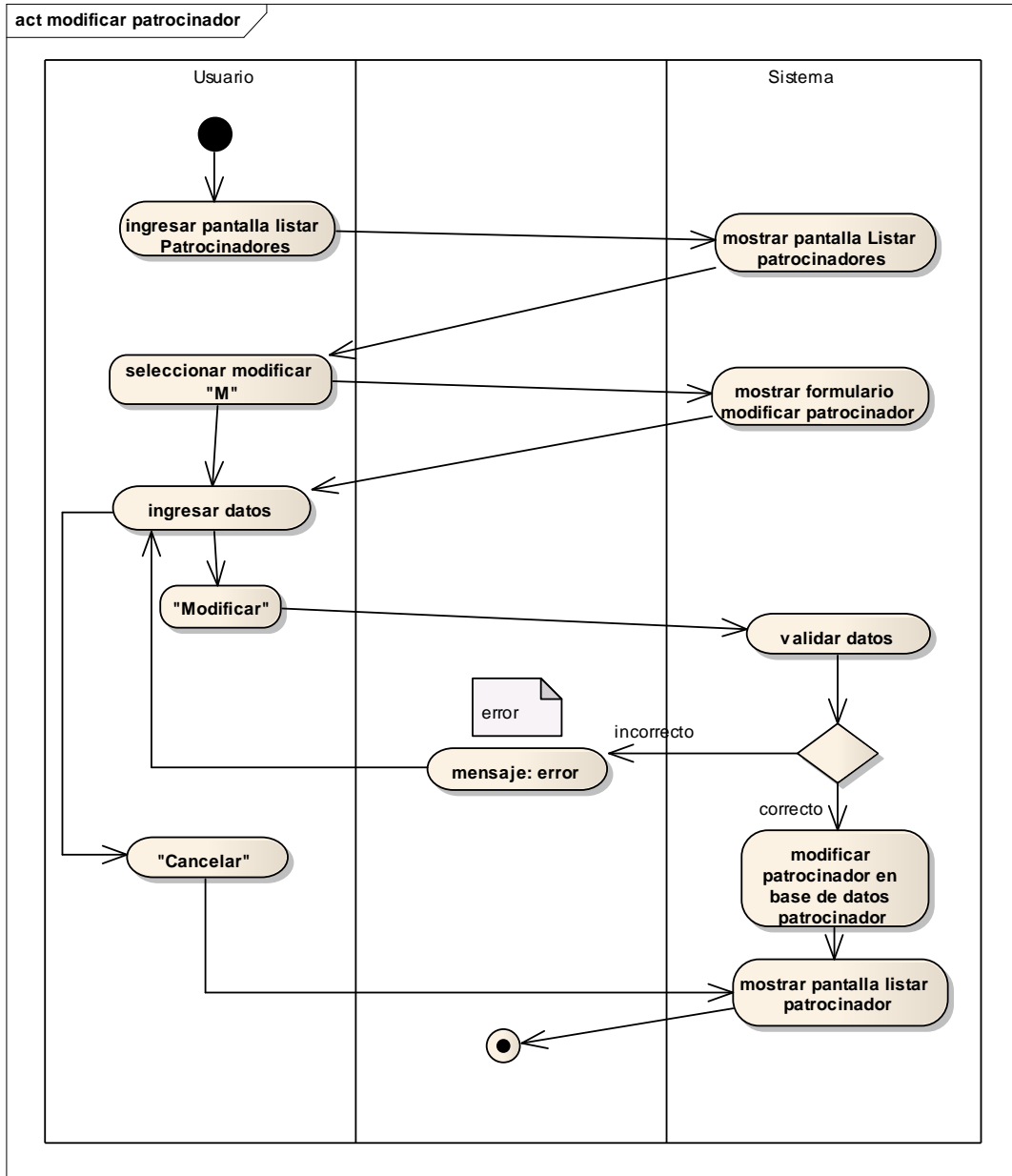


Figura N° 39 Diagrama de actividad: Modificar Patrocinadores

## Eliminar Patrocinador

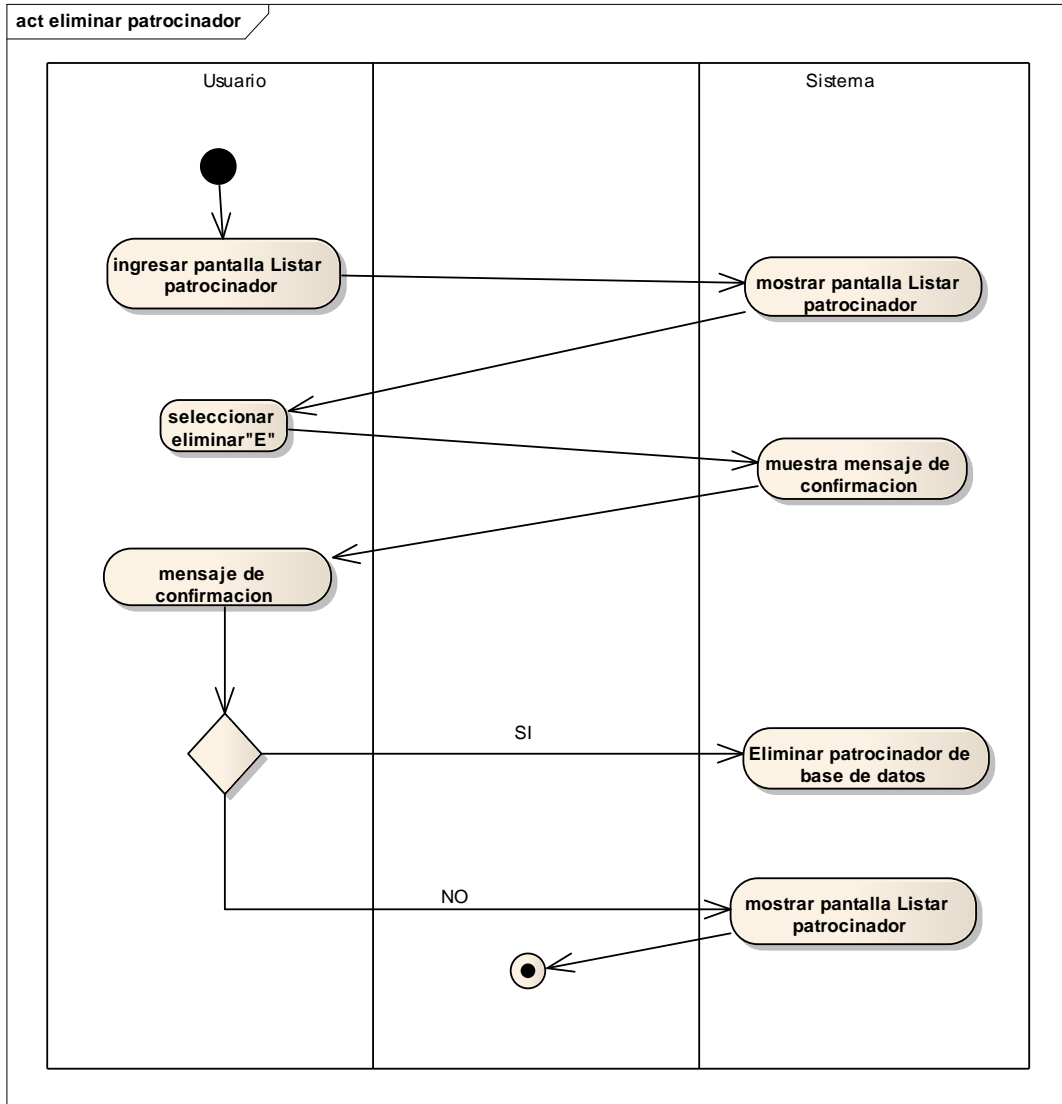


Figura N° 40 Diagrama de actividad: Eliminar Patrocinador

## Ver Afiliado (x patrocinador)

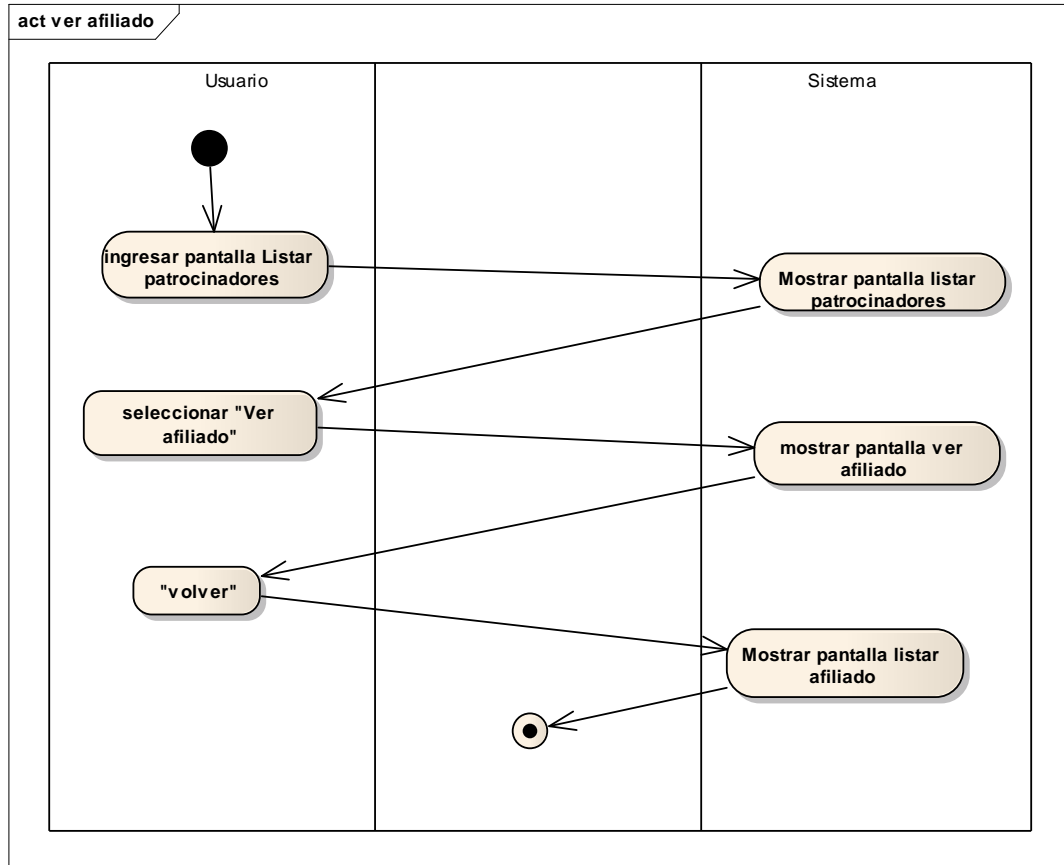


Figura N° 41 Diagrama de actividad: Ver afiliado

*1.1.6.1.4.6 Caso de Uso Gestionar Actividades*

**Listar Actividades**

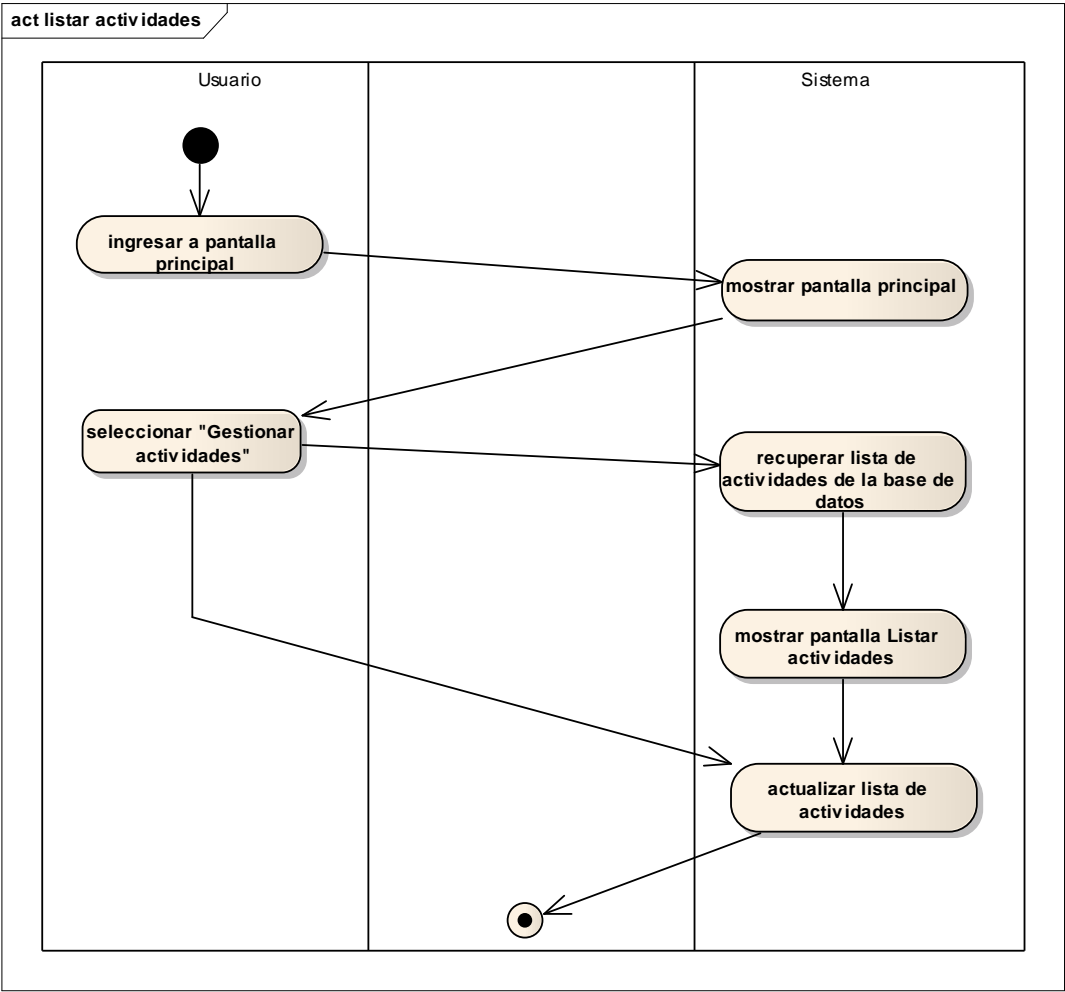


Figura N° 42 Diagrama de actividad: Listar Actividades

## Adicionar Actividades

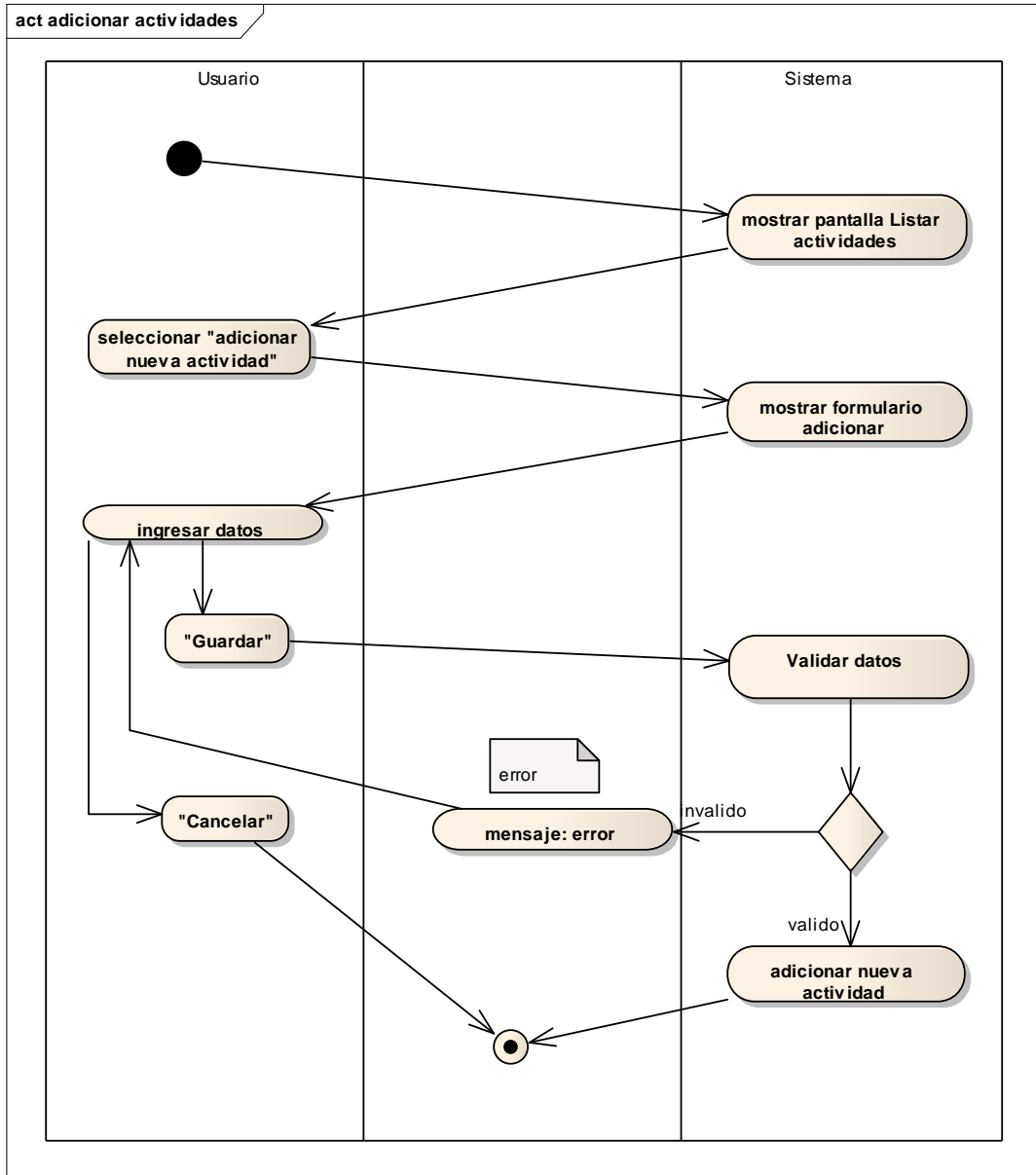


Figura N° 43 Diagrama de actividad: Adicionar Actividades

## Modificar Actividades

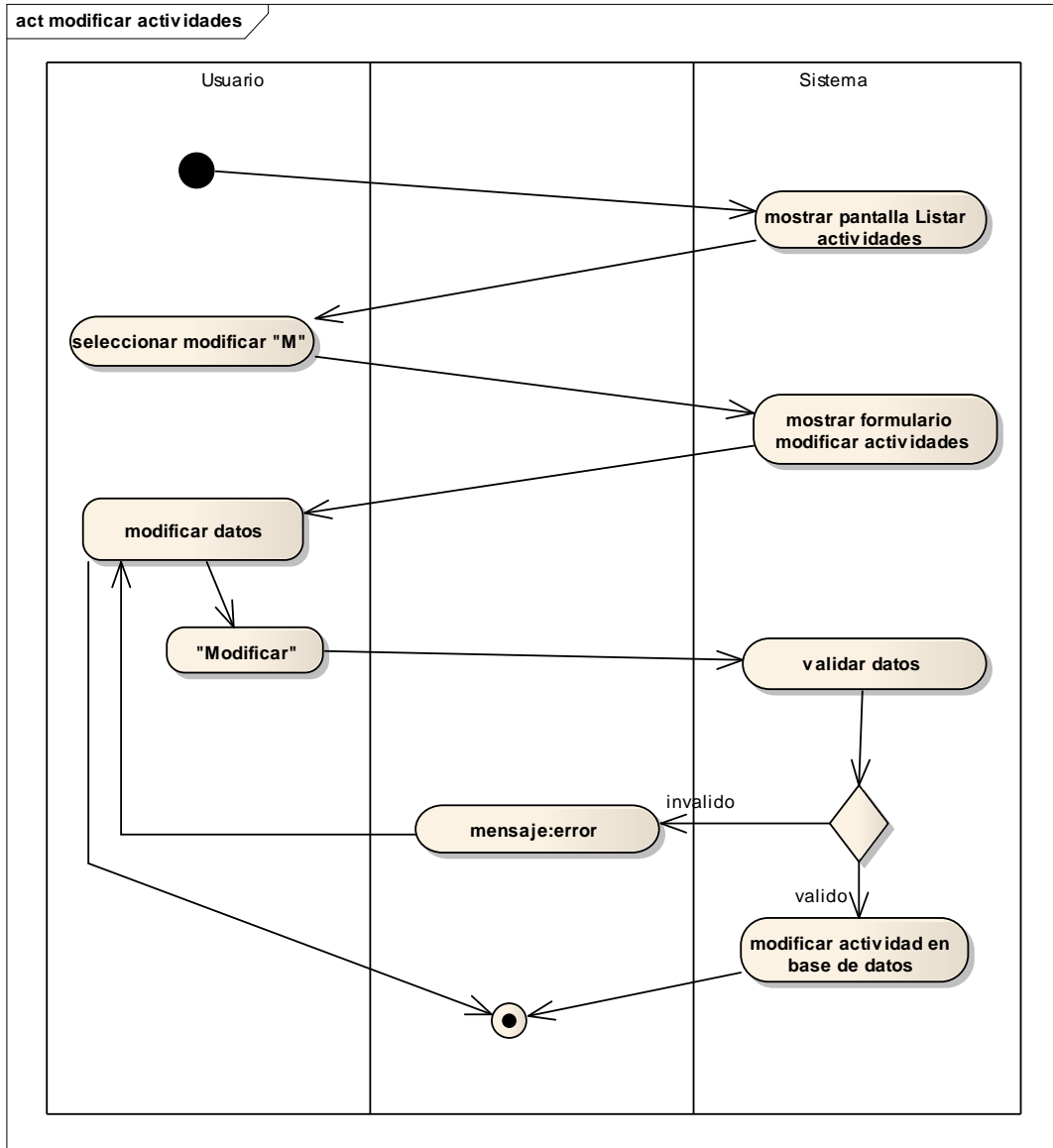


Figura N° 44 Diagrama de actividad: Modificar Actividades

## Eliminar Actividades

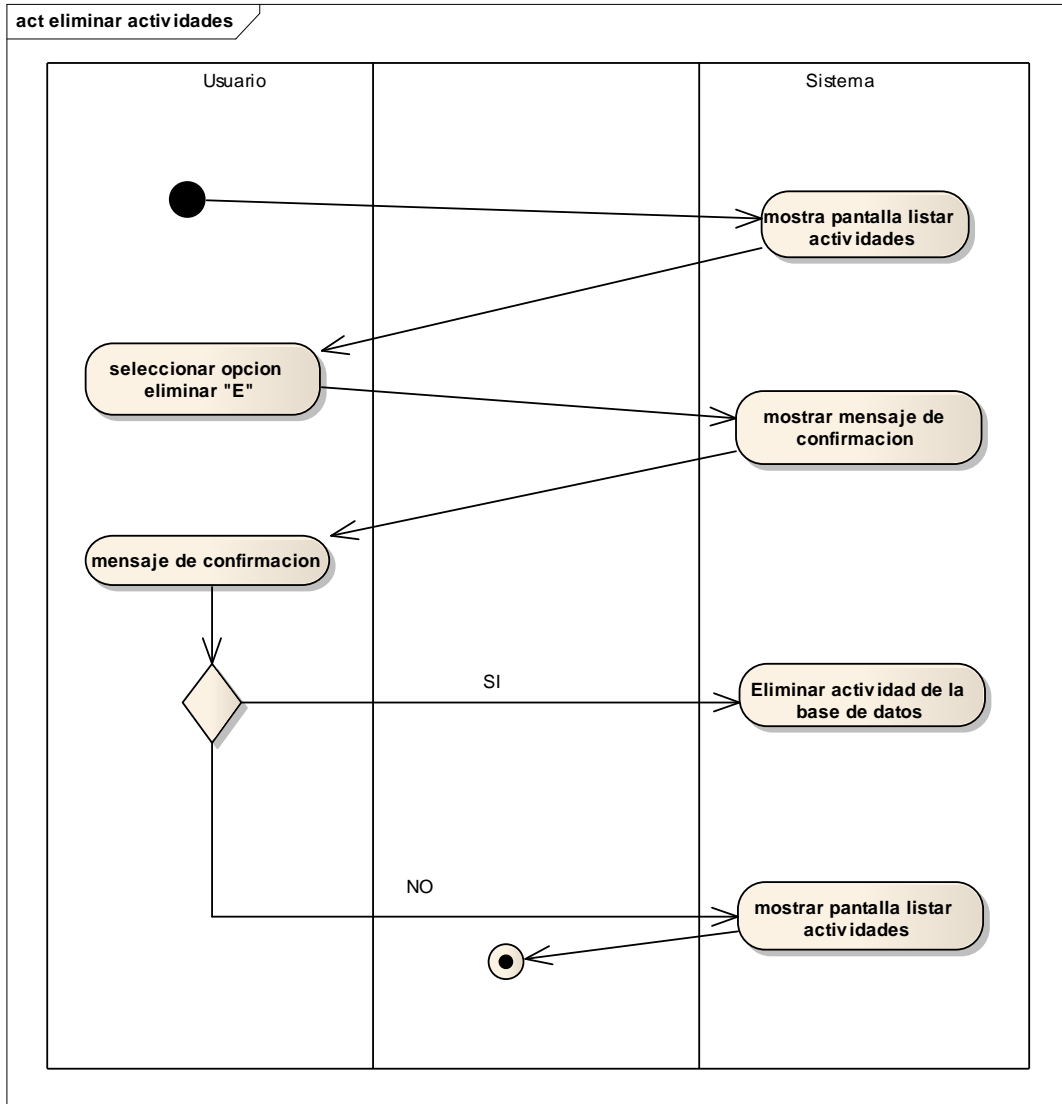


Figura N° 45 Diagrama de actividad: Eliminar Actividades

## Ver Asistencia

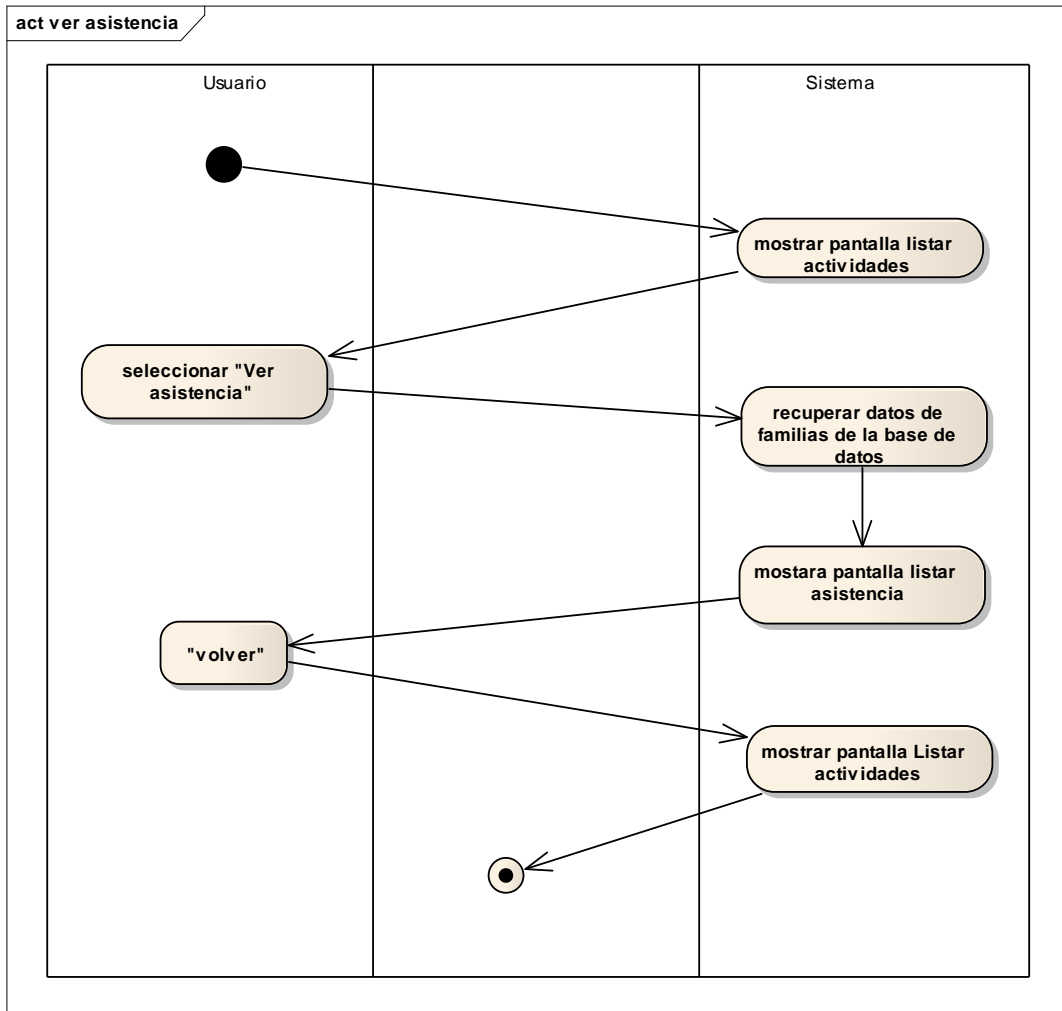


Figura N° 46 Diagrama de actividad: Ver asistencia

*1.1.6.1.4.7 Caso de Uso: Gestionar Afiliado*

**Listar afiliado**

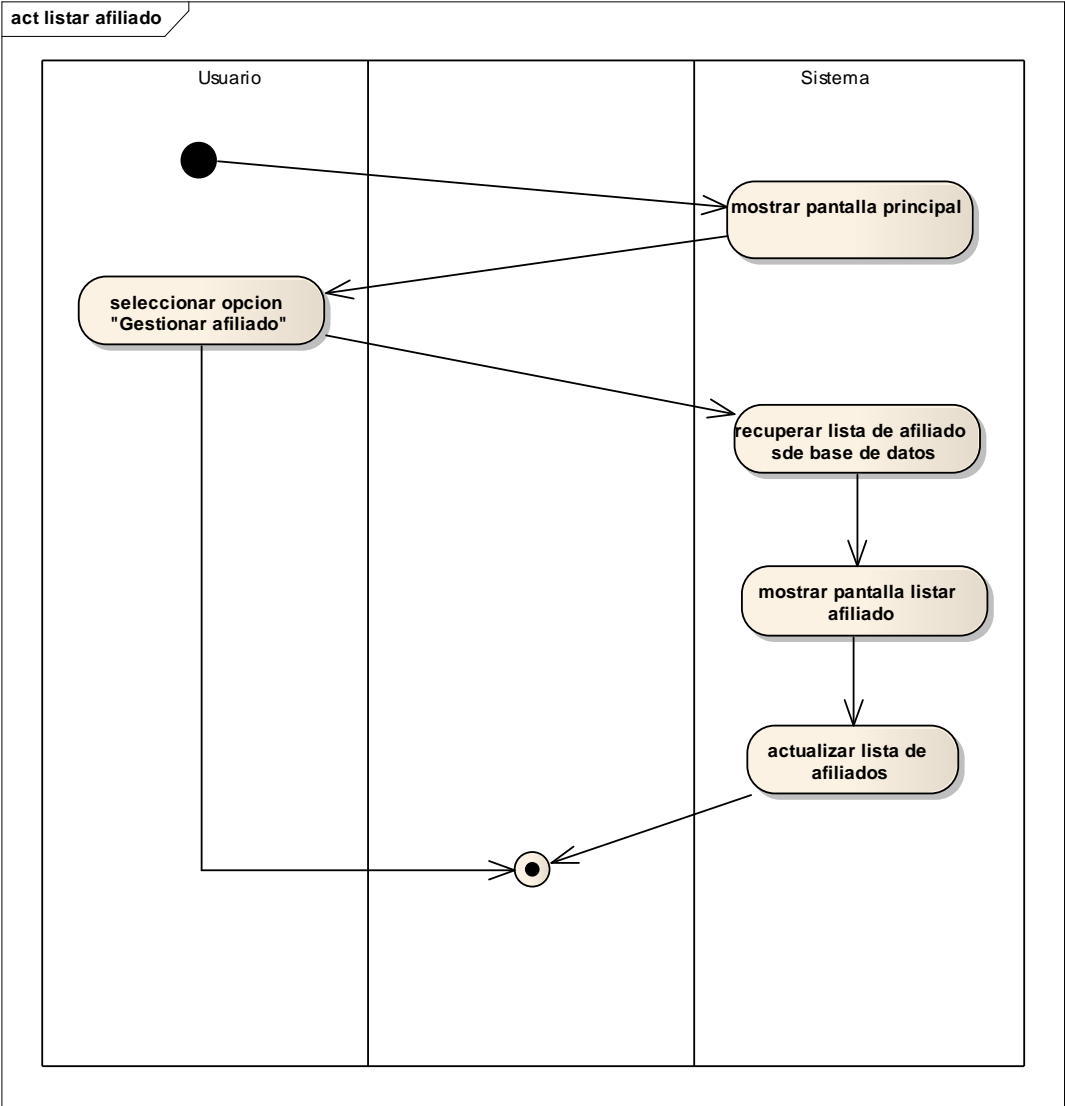


Figura N° 47 Diagrama de actividad: listar afiliado

## Modificar afiliado

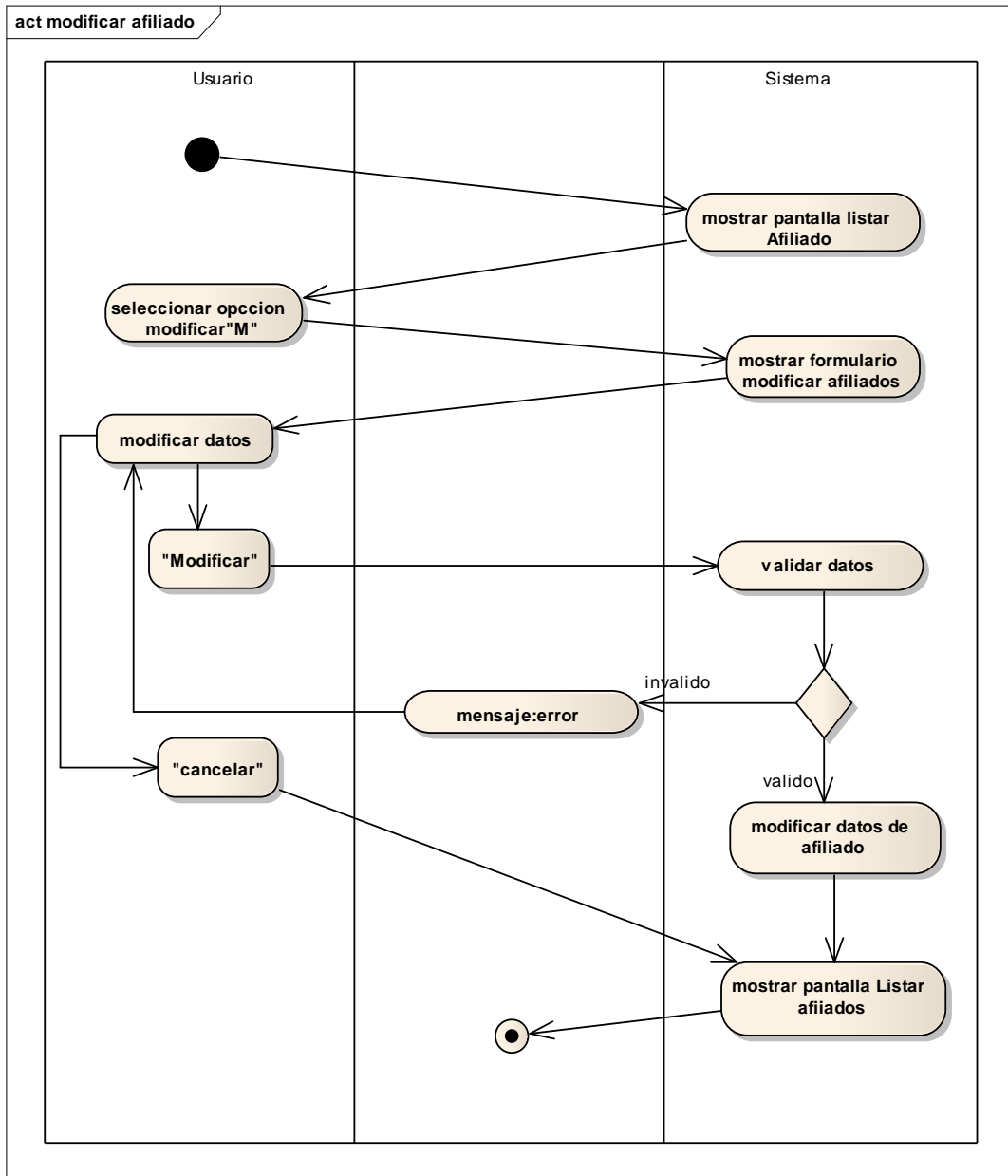


Figura N° 48 Diagrama de actividad: Modificar afiliado

## Eliminar afiliado

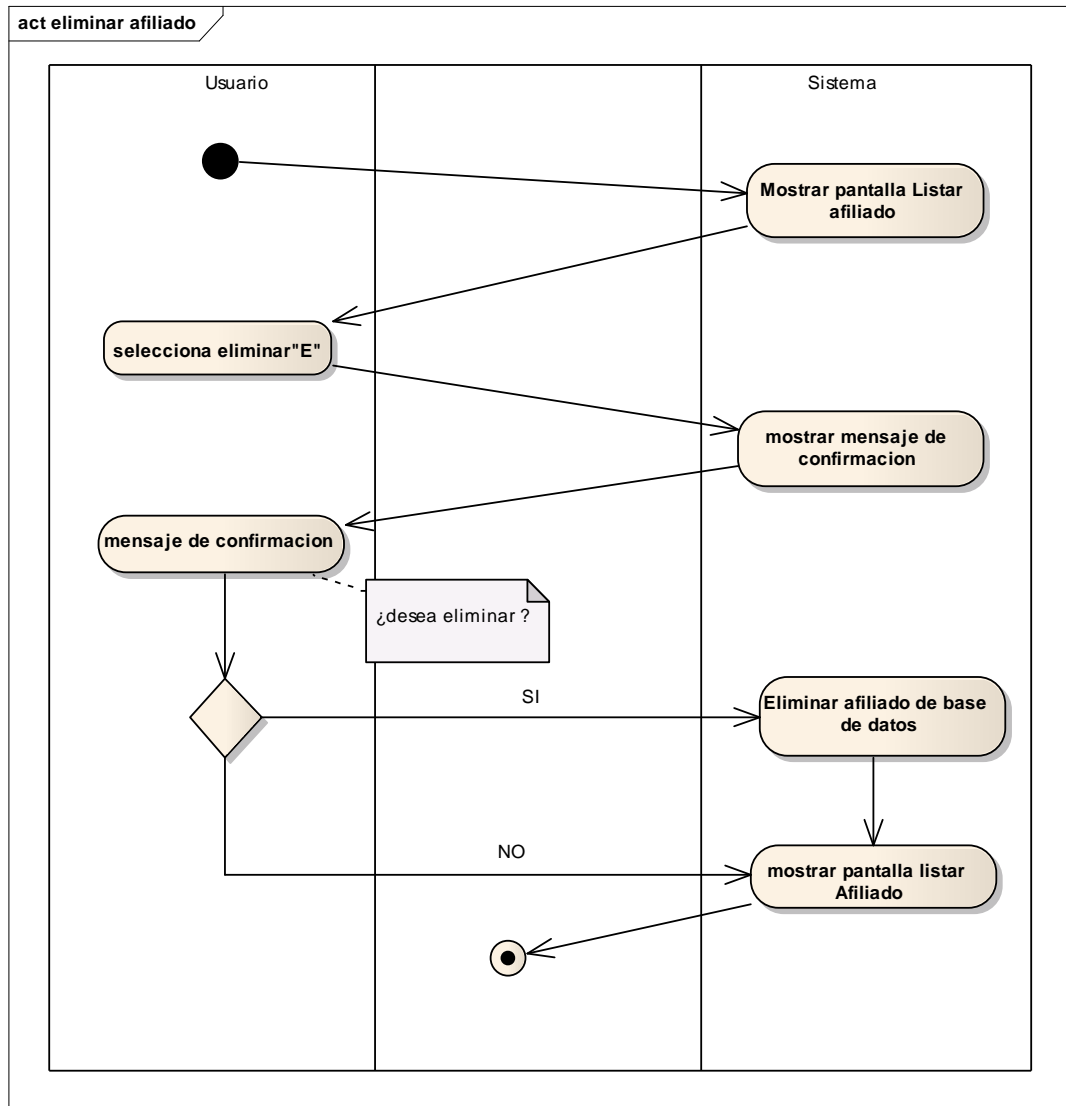


Figura N° 49 Diagrama de actividad: Eliminar afiliado

## Ver afiliado

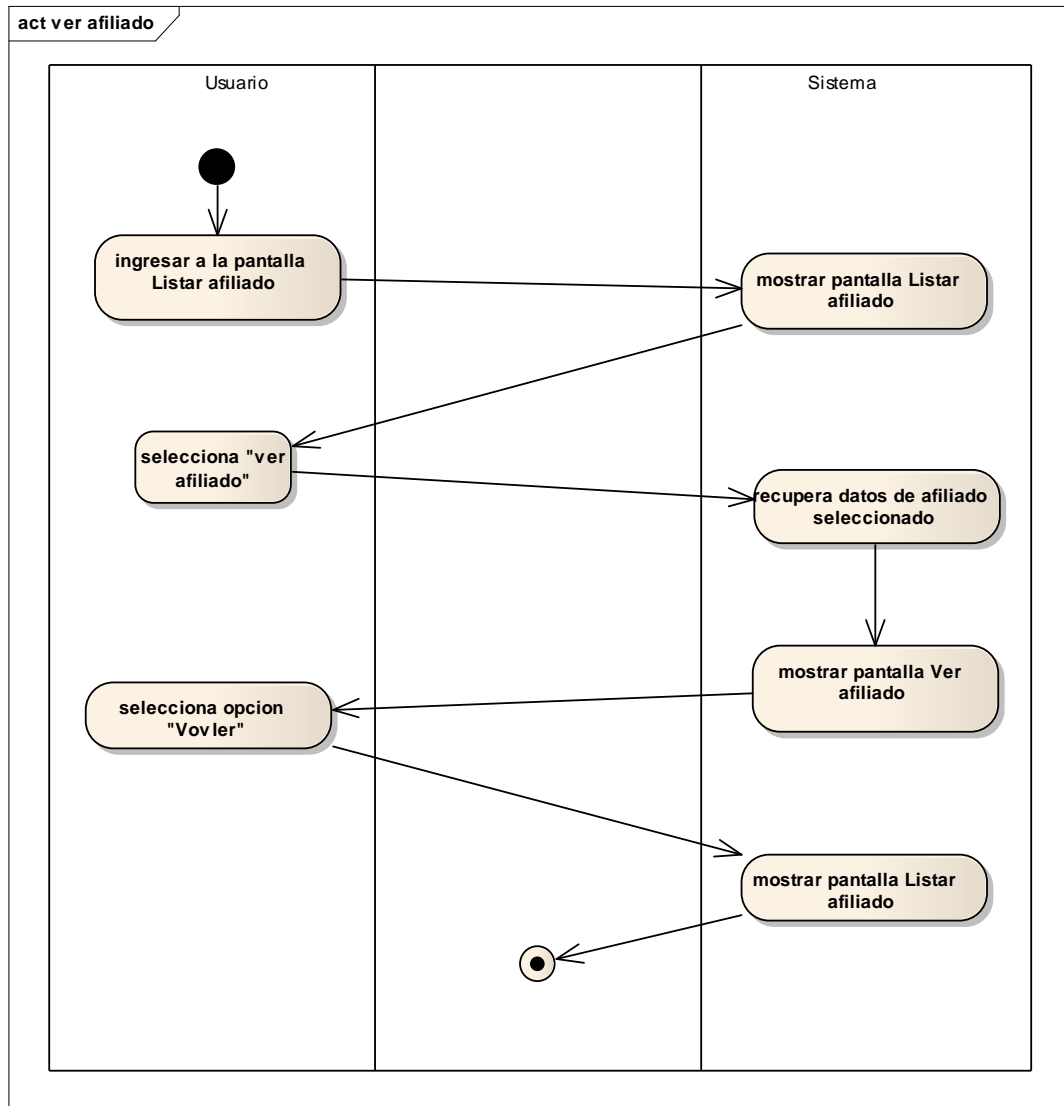


Figura N° 50 Diagrama de actividad: Ver Afiliado

***1.1.6.1.4.8 Caso de Uso: gestionar Salud***

**Listar afiliado**

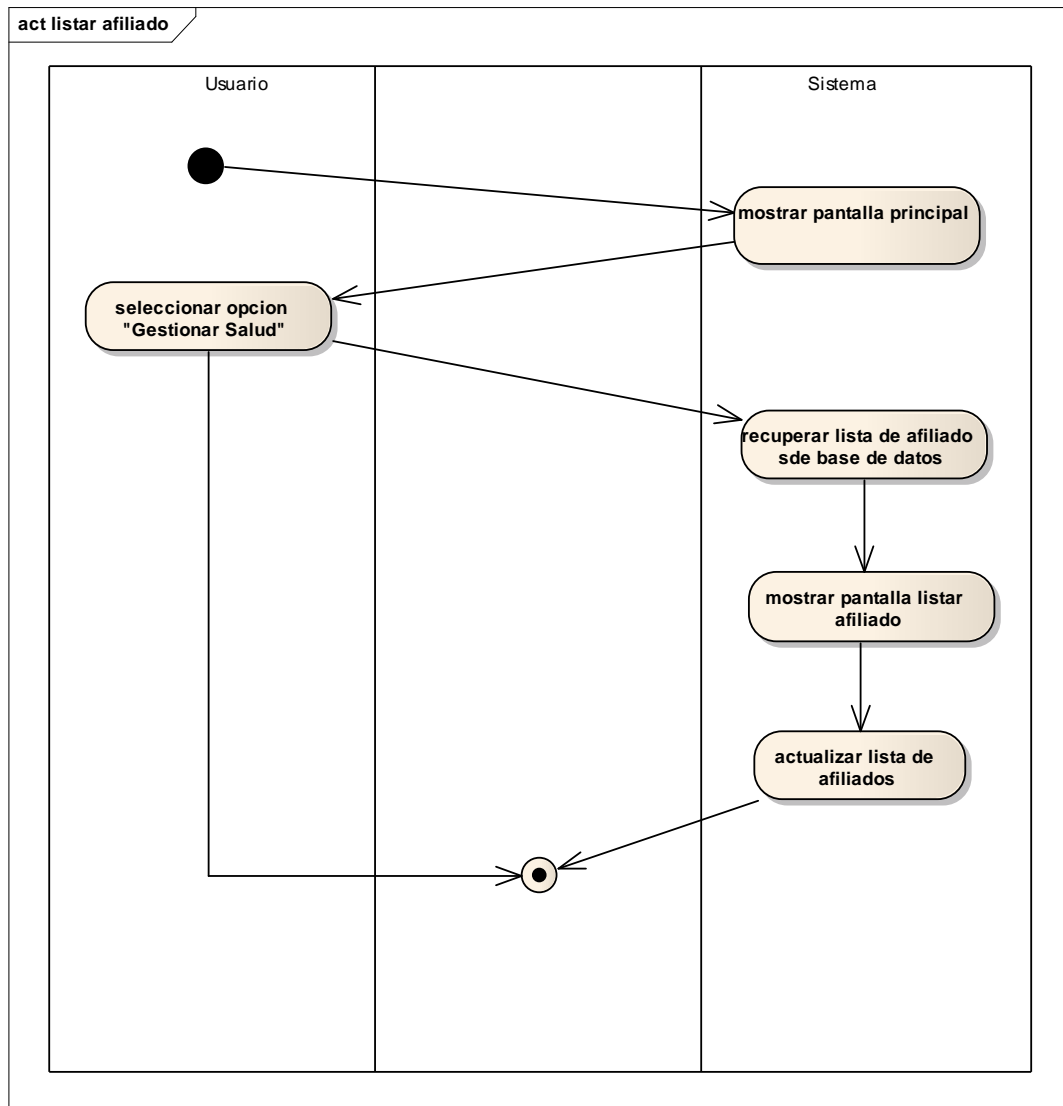


Figura N° 51 Diagrama de actividad: Listar Afiliado

## Buscar afiliado

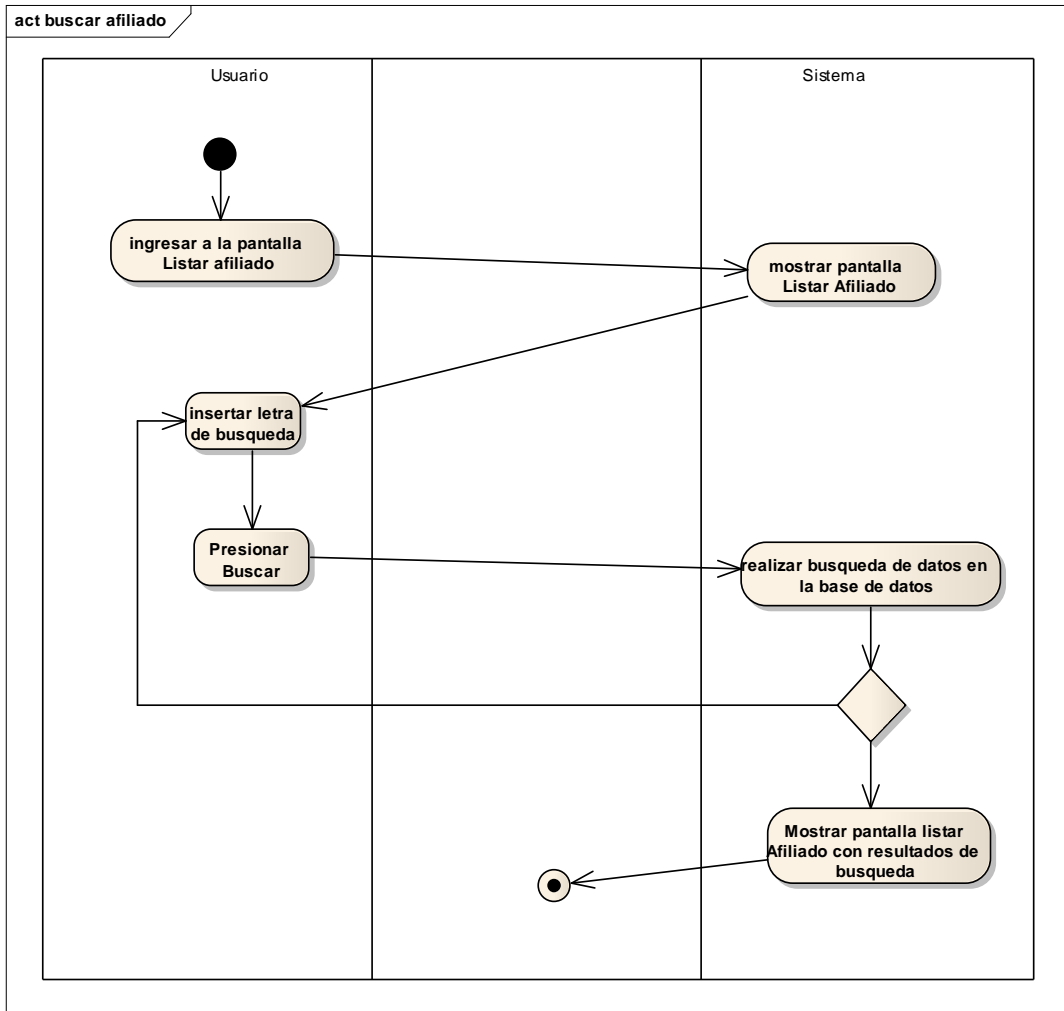


Figura N° 52 Diagrama de actividad: Buscar Afiliado

## Llenar Formulario

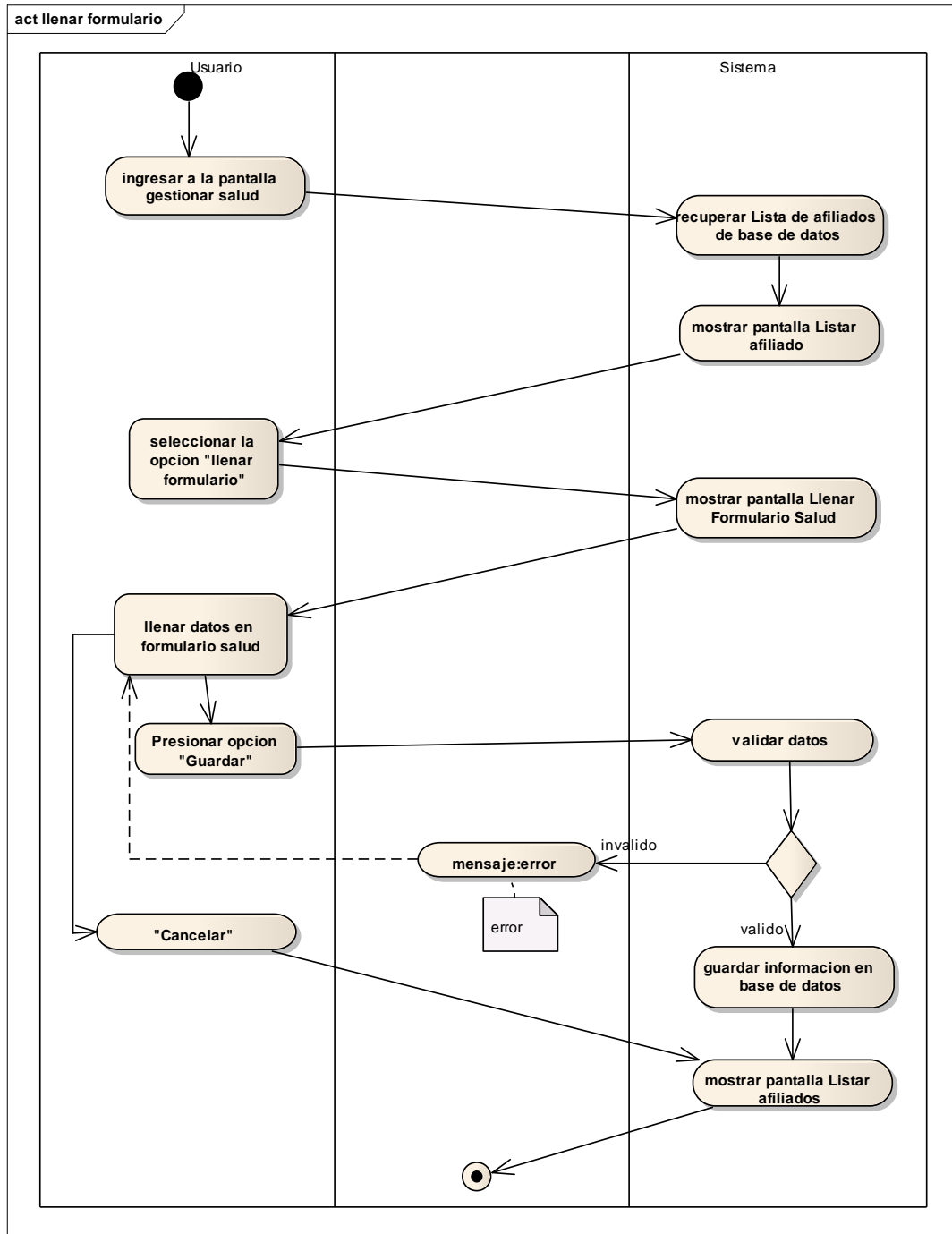


Figura N° 53 Diagrama de actividad: Llenar Formulario

## Ver Formulario

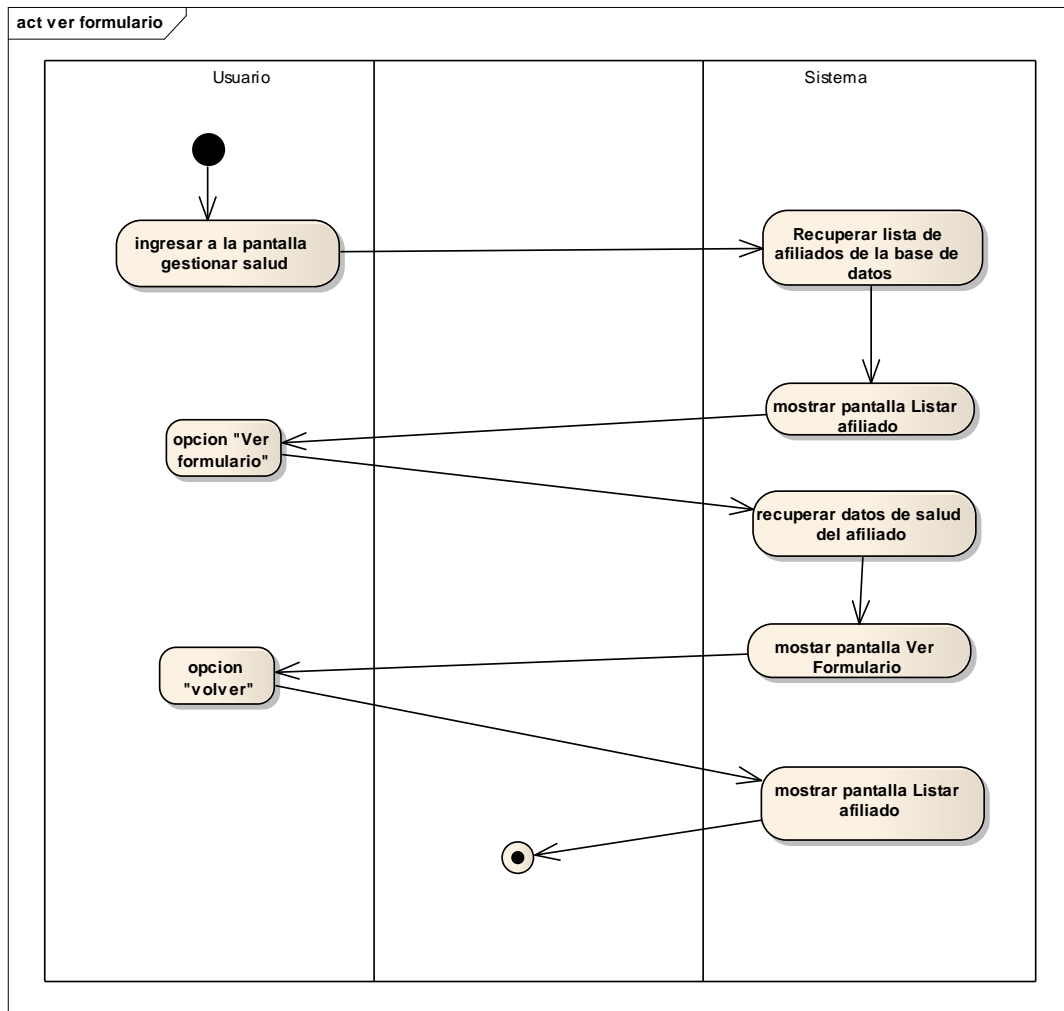


Figura N° 54 Diagrama de actividad: Ver Formulario

**1.1.6.1.4.9 Caso de uso: Controlar Asistencia**

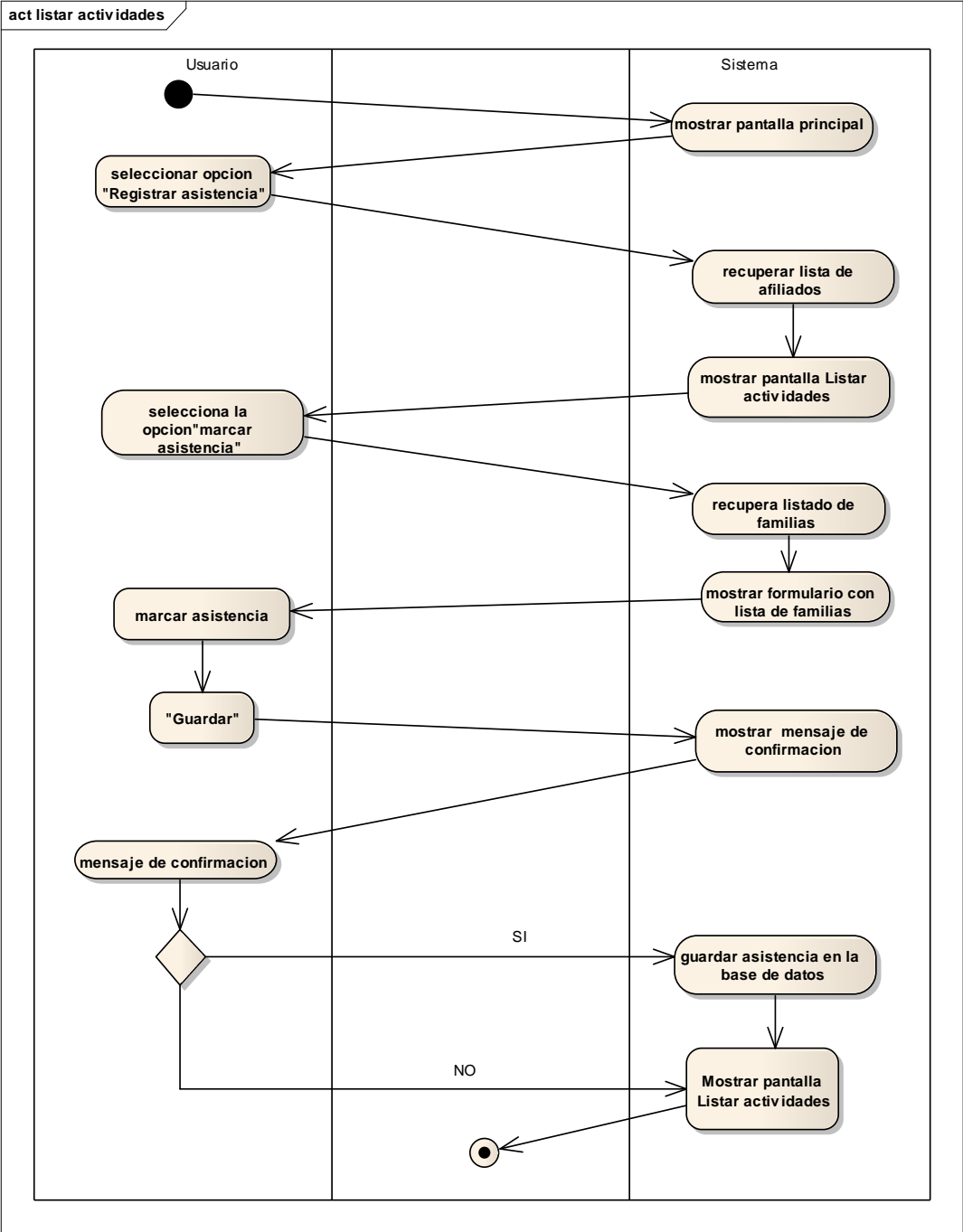


Figura N° 55 Diagrama de actividad: controlar asistencia

## Caso de Uso: Registrar nivel educativo

### Listar afiliado

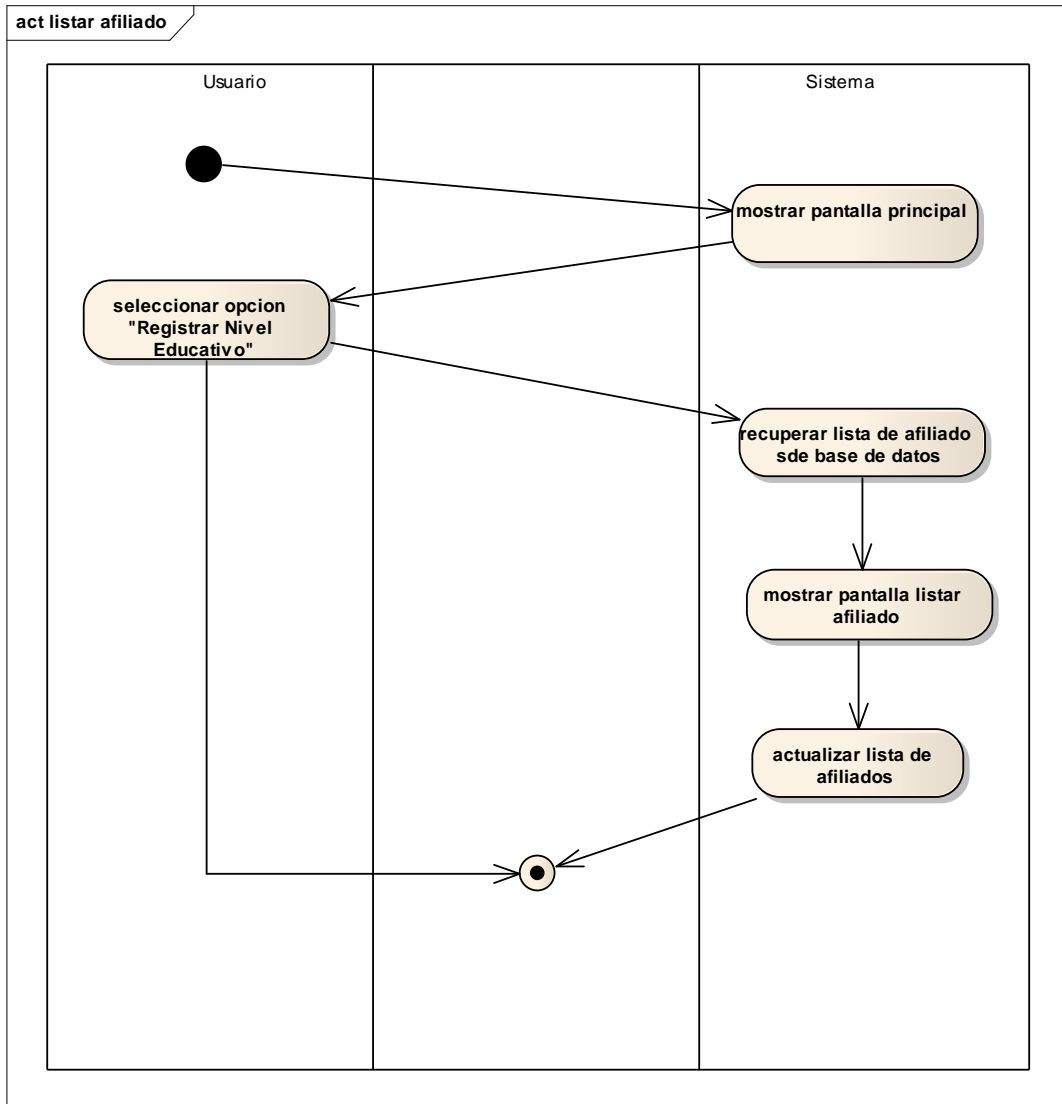


Figura N° 56 Diagrama de actividad: Listar afiliado

## Buscar afiliado

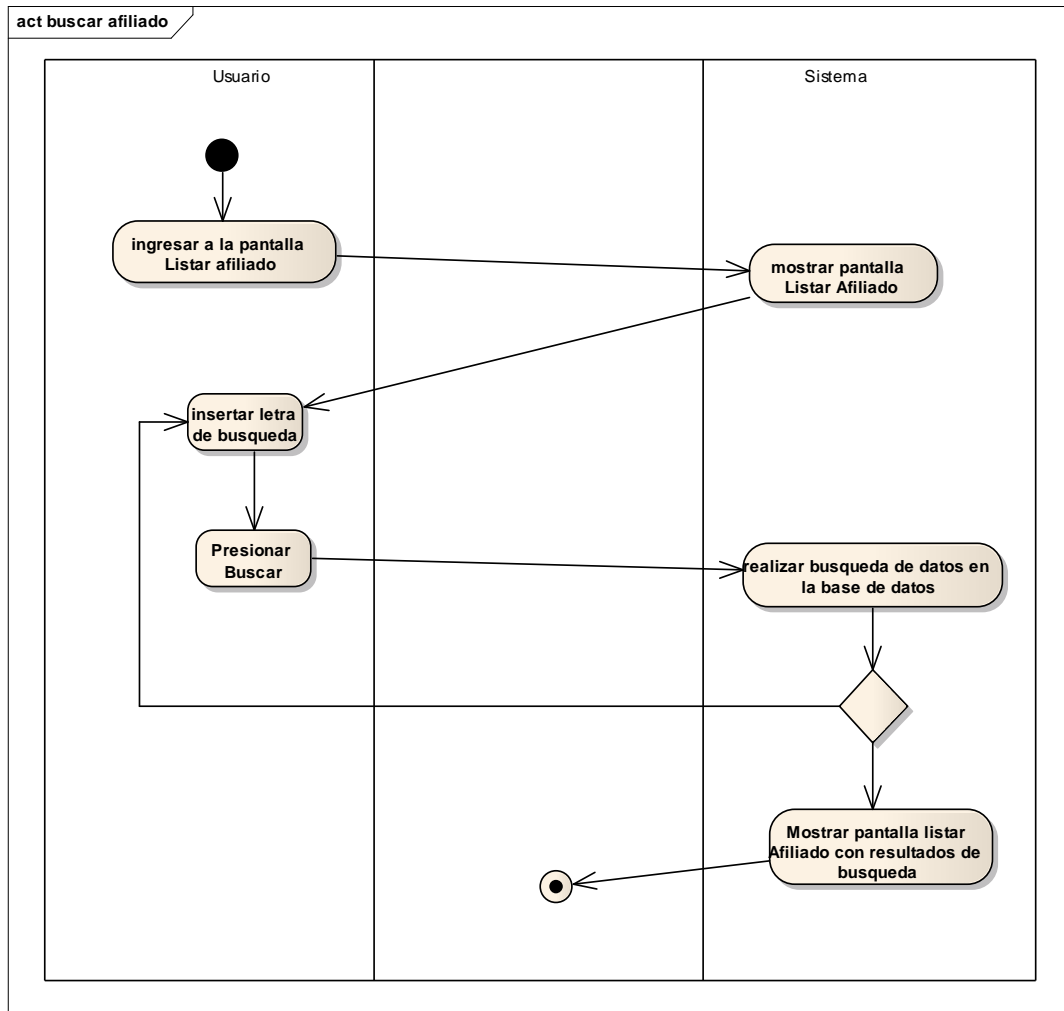


Figura N° 57 Diagrama de actividad: Buscar afiliado

## Llenar formulario

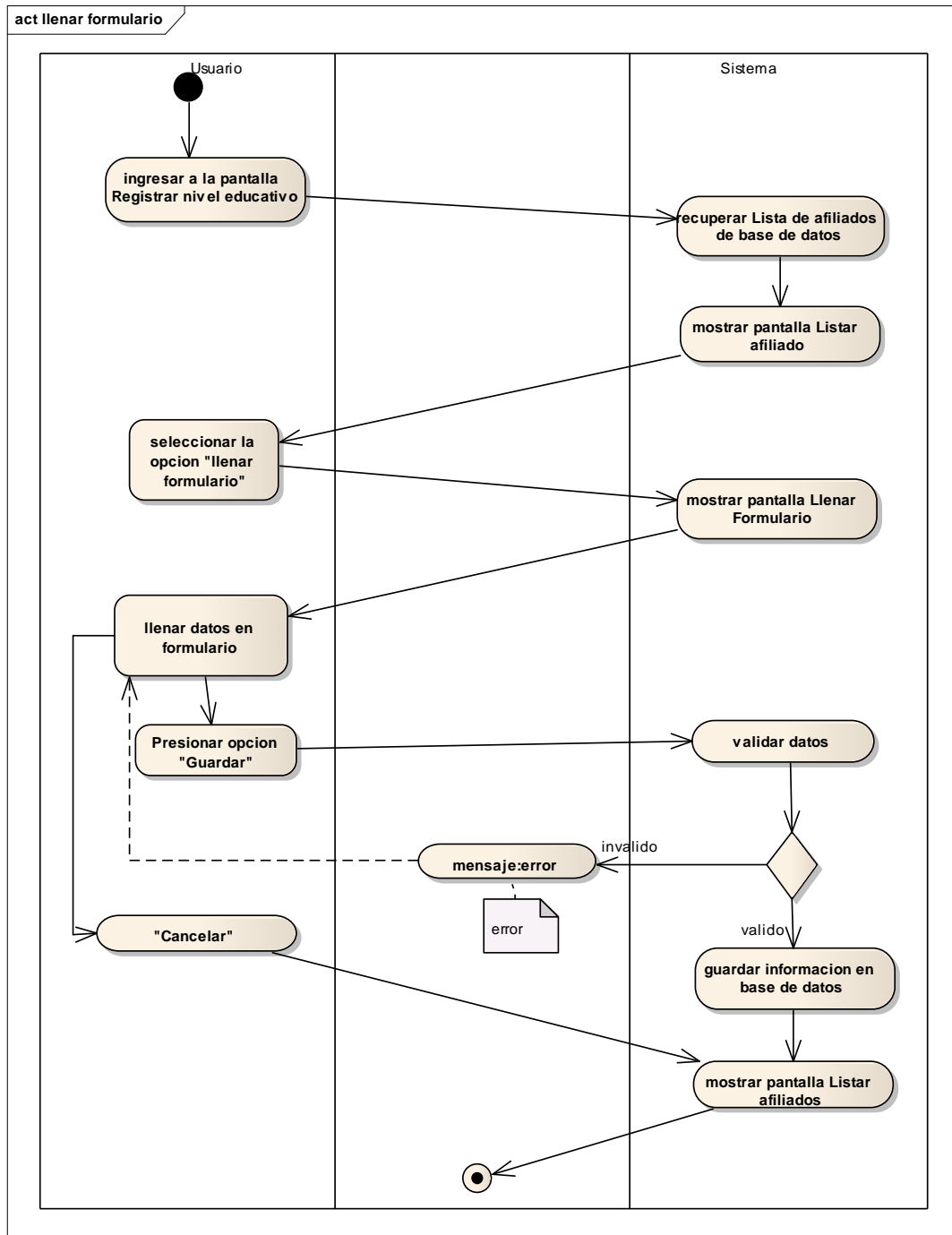


Figura N° 58 Diagrama de actividad: Llenar formulario datos educativos

## Ver Formulario

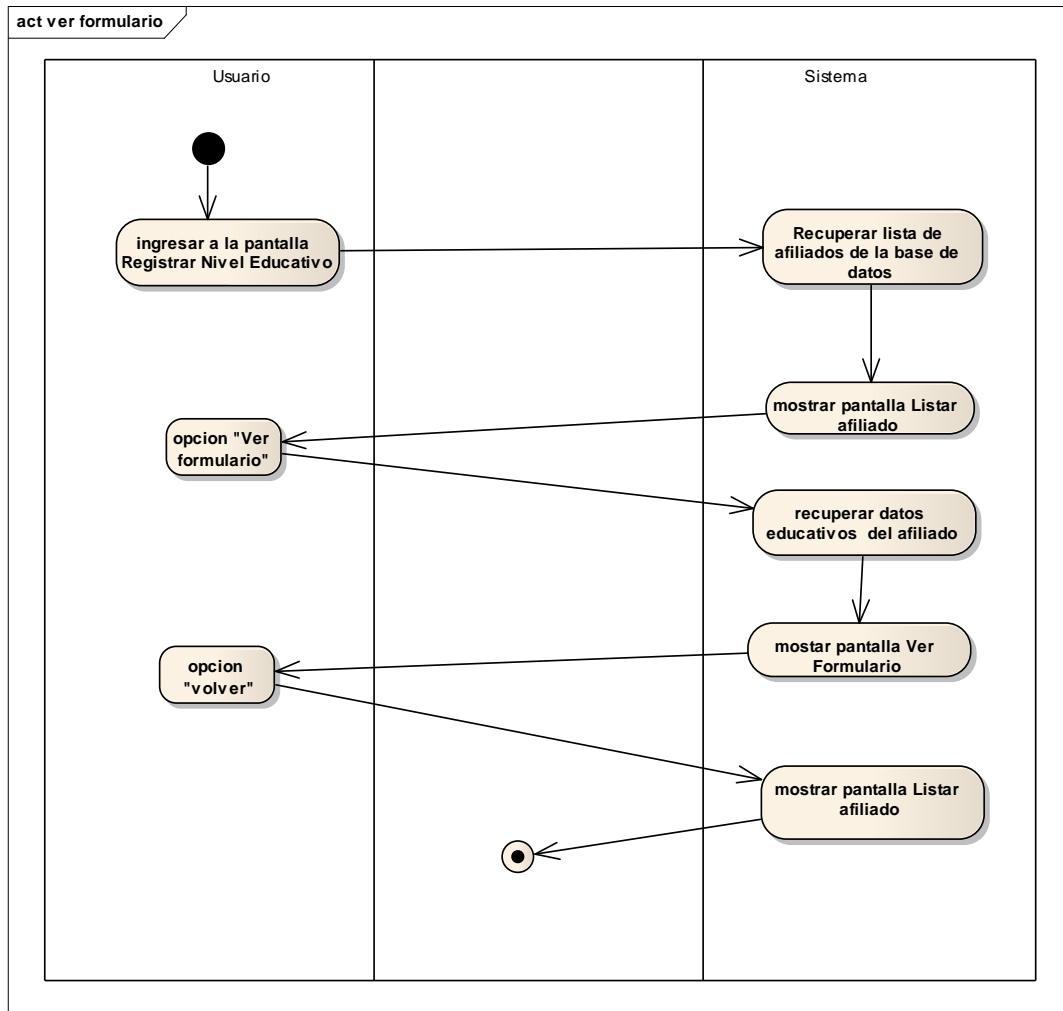


Figura N° 59 Diagrama de actividad: Ver Formulario Datos educativos

## **I.1.6.2 Modelado de diagramas de secuencia**

### ***I.1.6.2.1 Introducción***

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino.

Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones.

### ***I.1.6.2.2 Propósito***

- Comprender la dinámica del sistema deseado para la organización
- Identificar clases de análisis y diseño
- Comprender la interacción de los actores de sistema.
- 

### ***I.1.6.2.3 Alcance***

Describir la dinámica de sistema en el tiempo de vida de las clases u objetos

Definir un diagrama de secuencia para los casos de uso más relevantes del sistema

### 1.1.6.2.4 Diagramas de secuencia

#### 1.1.6.2.4.1 Ingresar al sistema

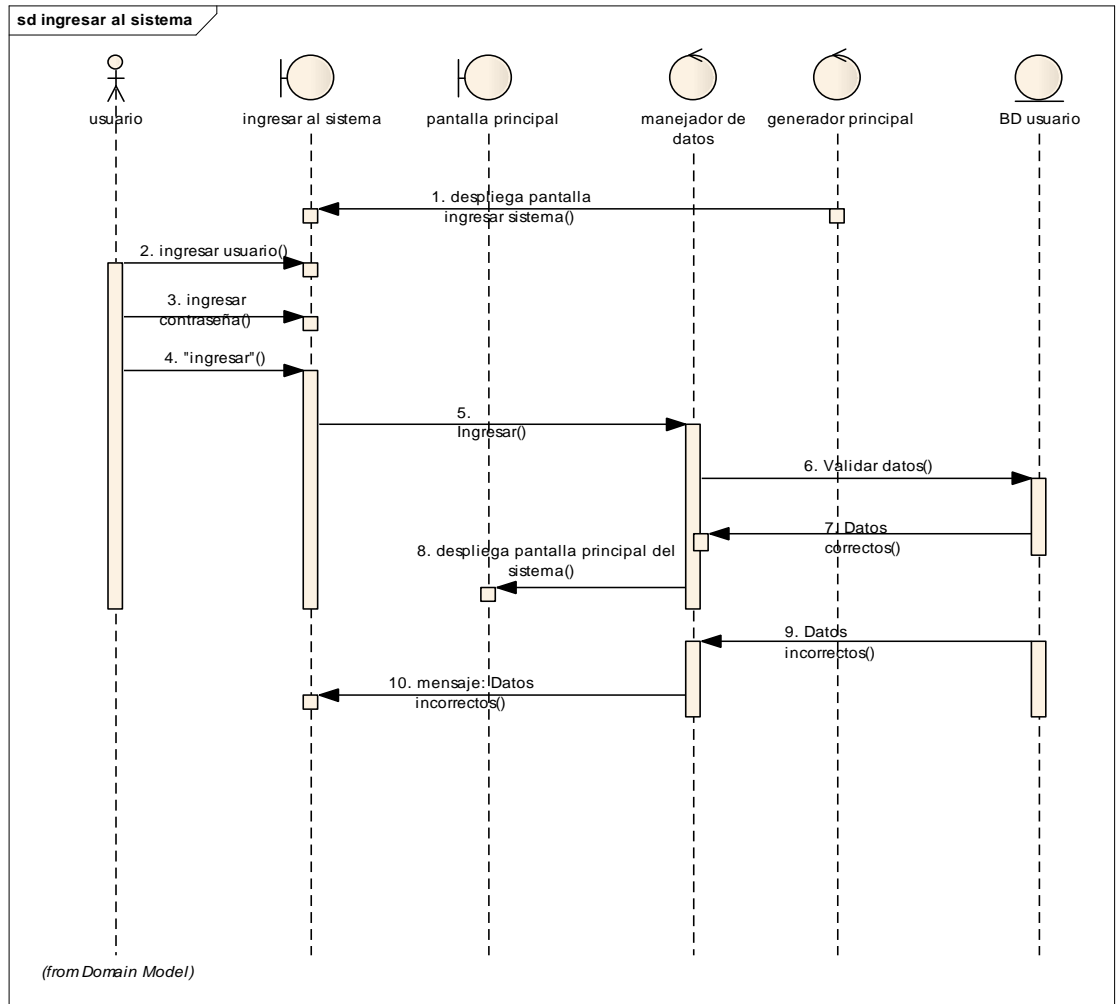


Figura N° 60 Diagrama de secuencia: Ingresar al sistema

**I.1.6.2.4.2 Gestionar usuario**

**Listar Usuario**

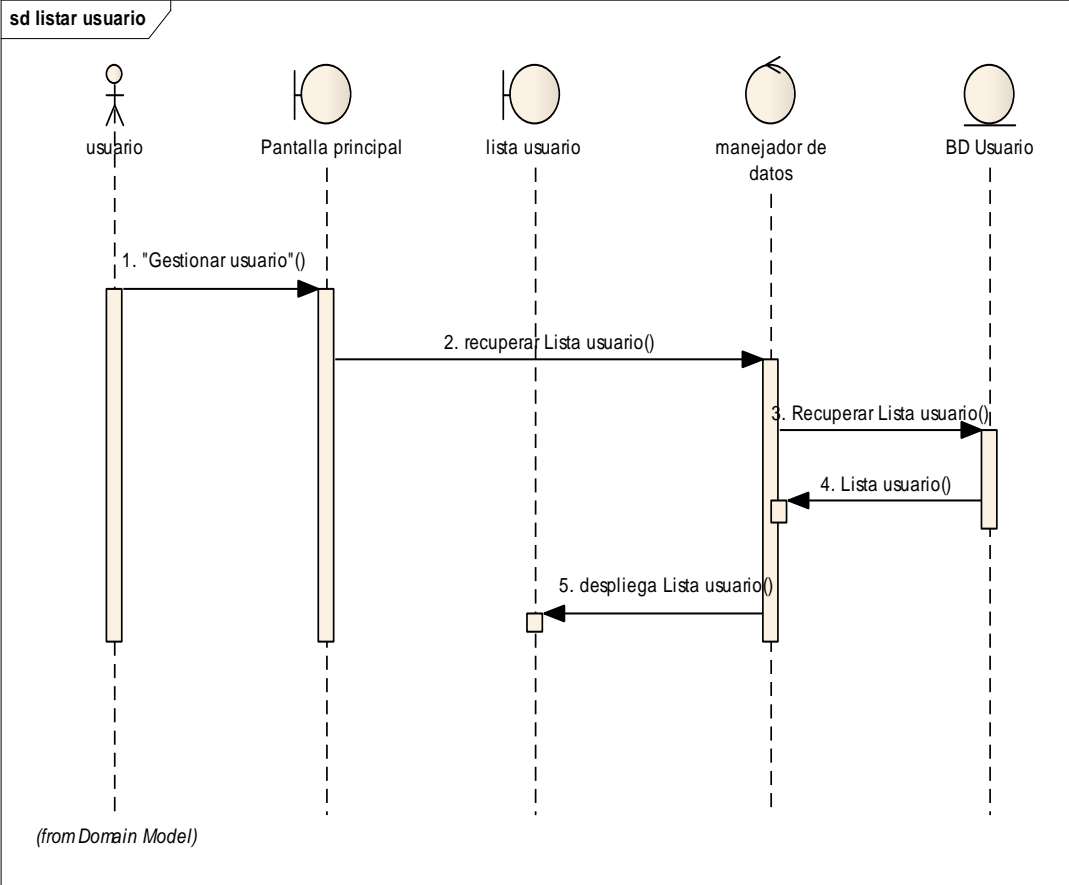


Figura N° 61 Diagrama de secuencia: Listar usuario

## Adicionar Usuario

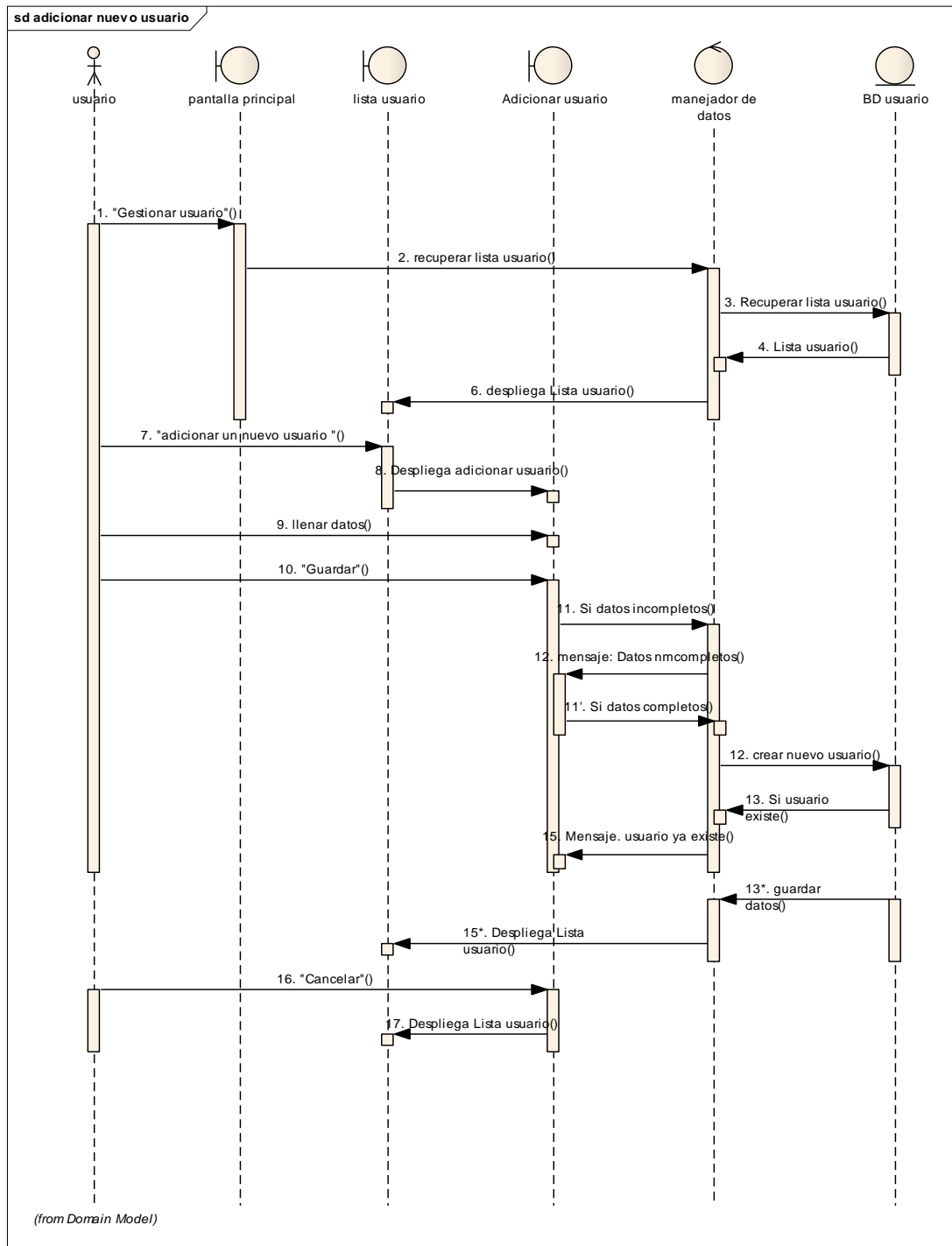


Figura N° 62 Diagrama de secuencia: adicionar usuario

## Modificar Usuario

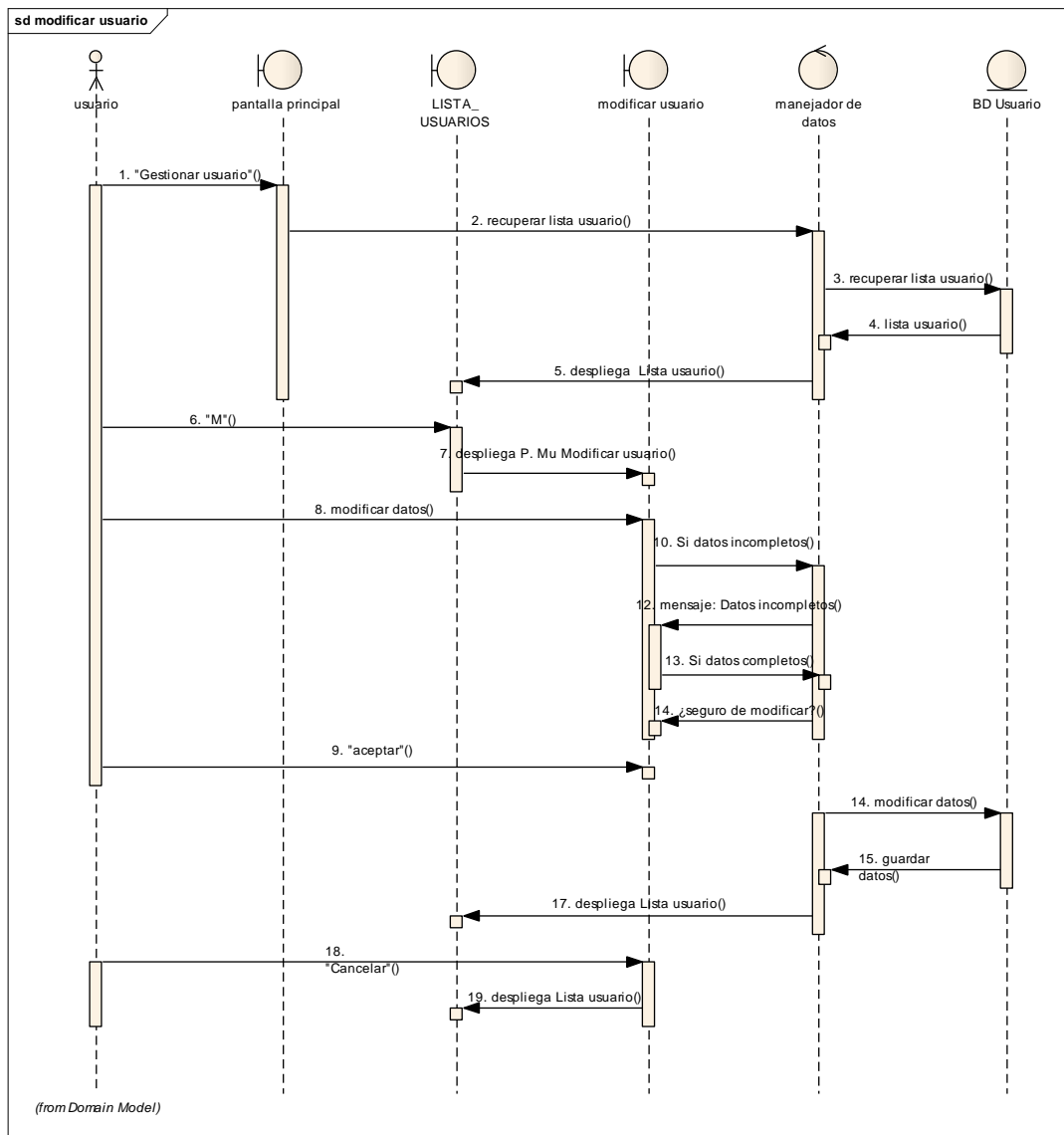


Figura N° 63 Diagrama de secuencia: modificar usuario

## Eliminar Usuario

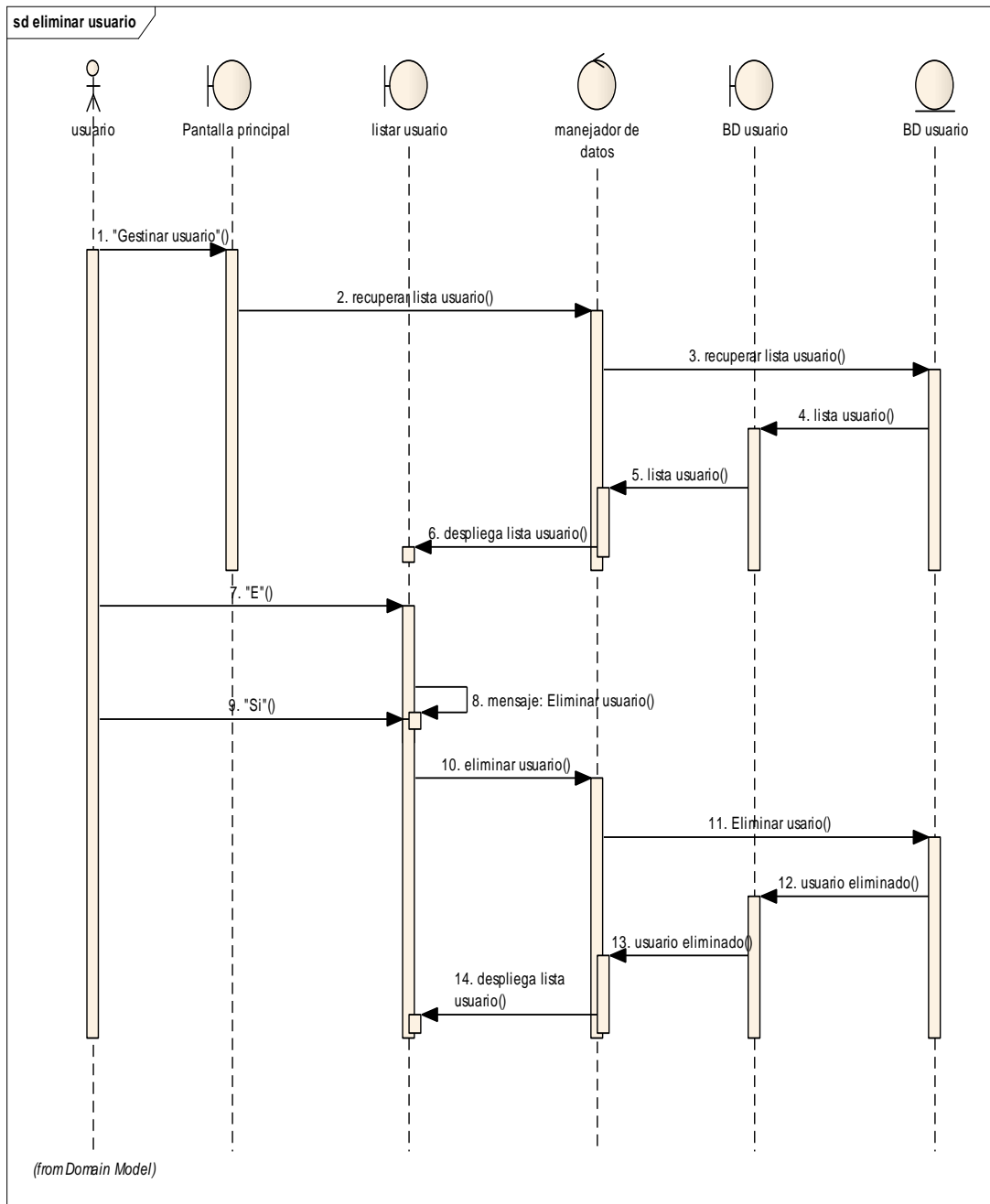


Figura N° 64 Diagrama de secuencia: eliminar usuario

**I.1.6.2.4.3 Gestionar familia**

**Listar Familia**

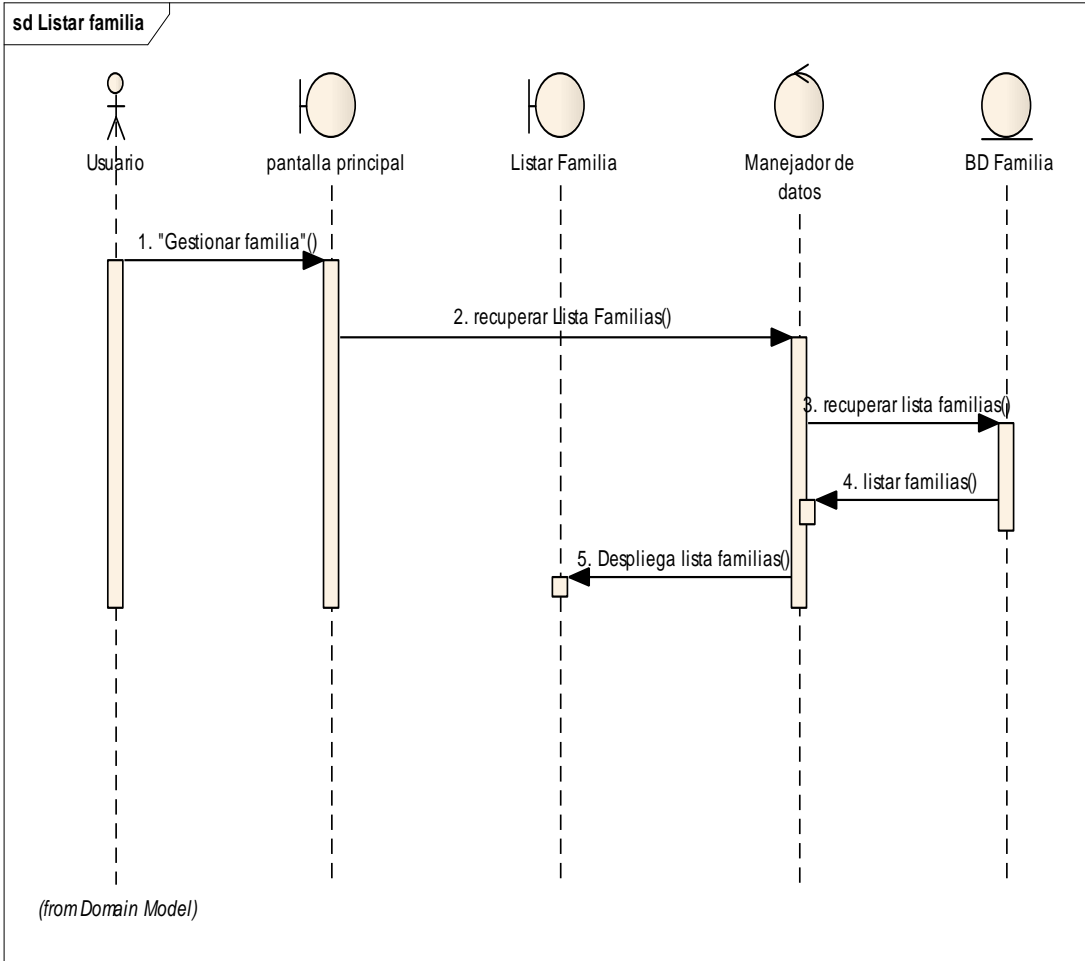


Figura N° 65 Diagrama de secuencia: Listar familia

## Adicionar Familia

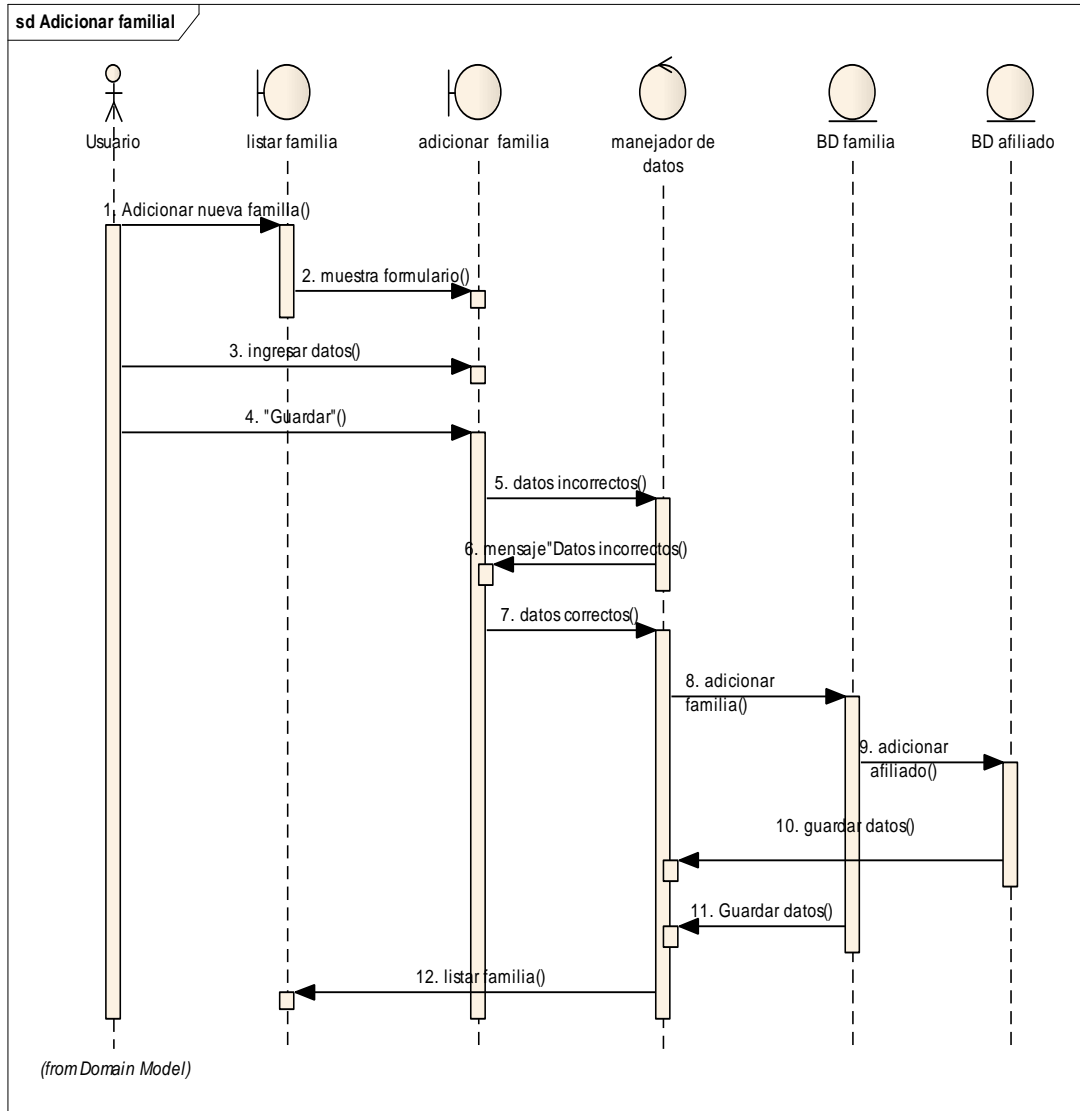


Figura N° 66 Diagrama de secuencia: Adicionar familia

## Modificar familia

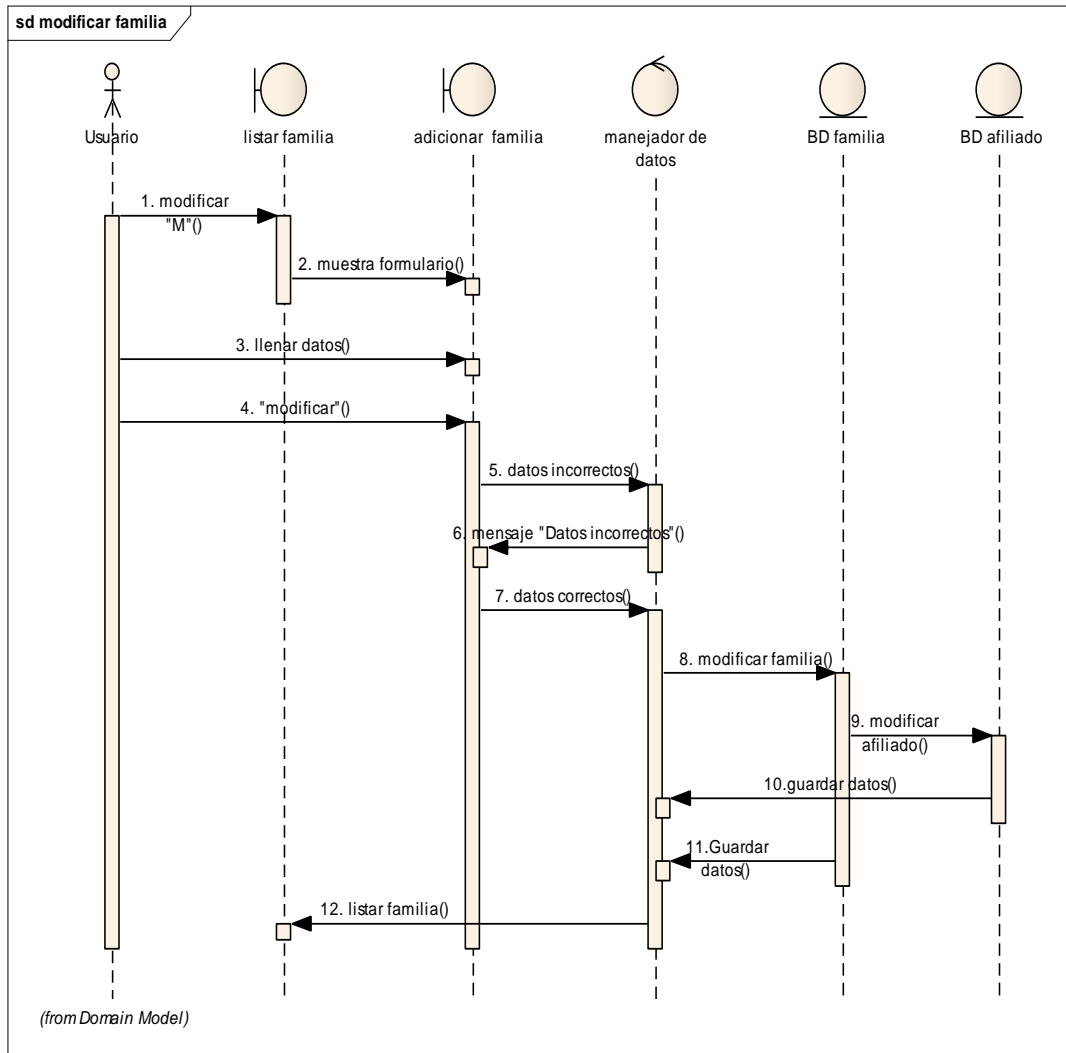


Figura N° 67 Diagrama de secuencia: Modificar familia

## Eliminar familia

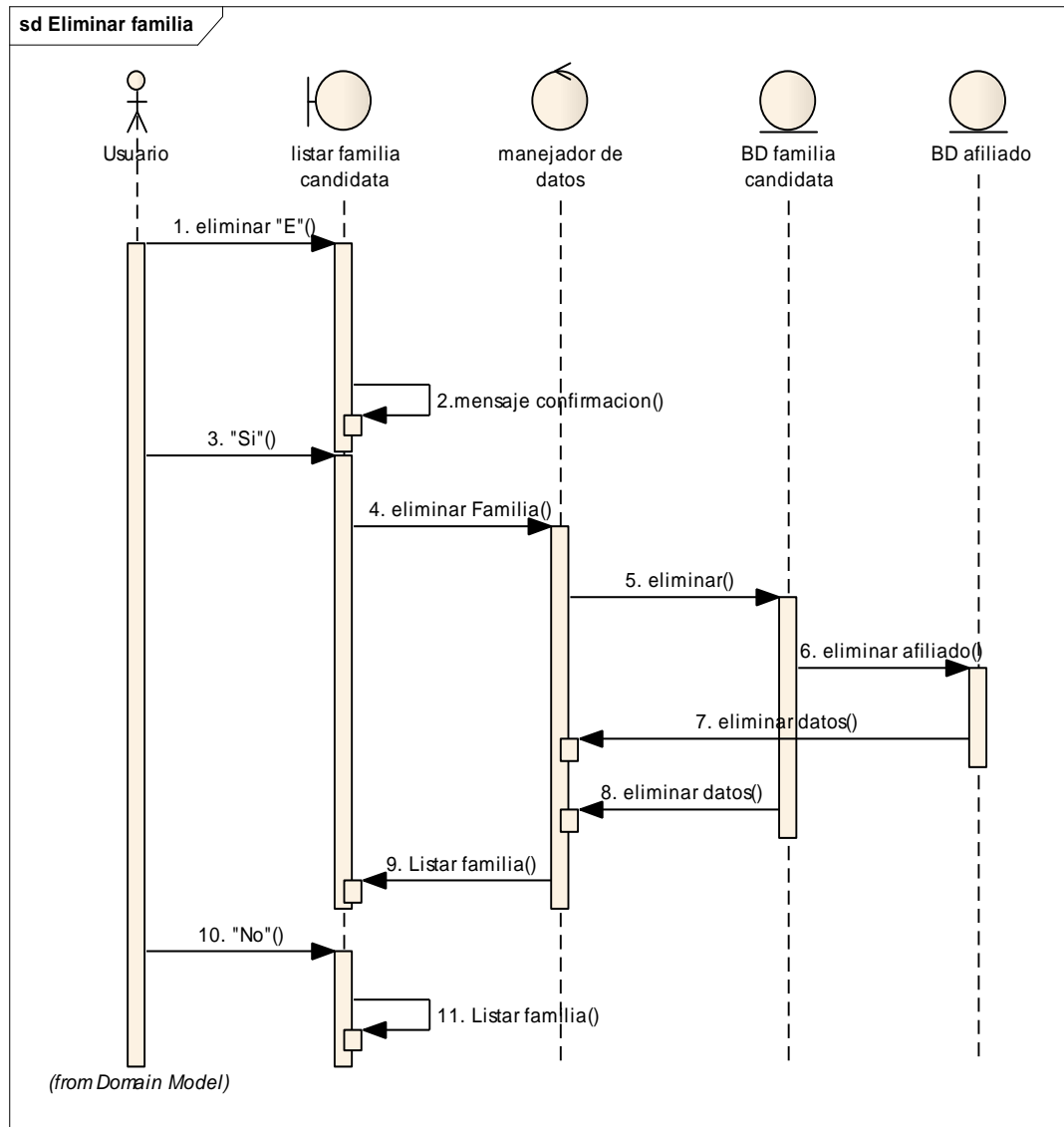


Figura N° 68 Diagrama de secuencia: Eliminar familia

**I.1.6.2.4.4 Gestionar Familia Candidata**

**Listar familia candidata**

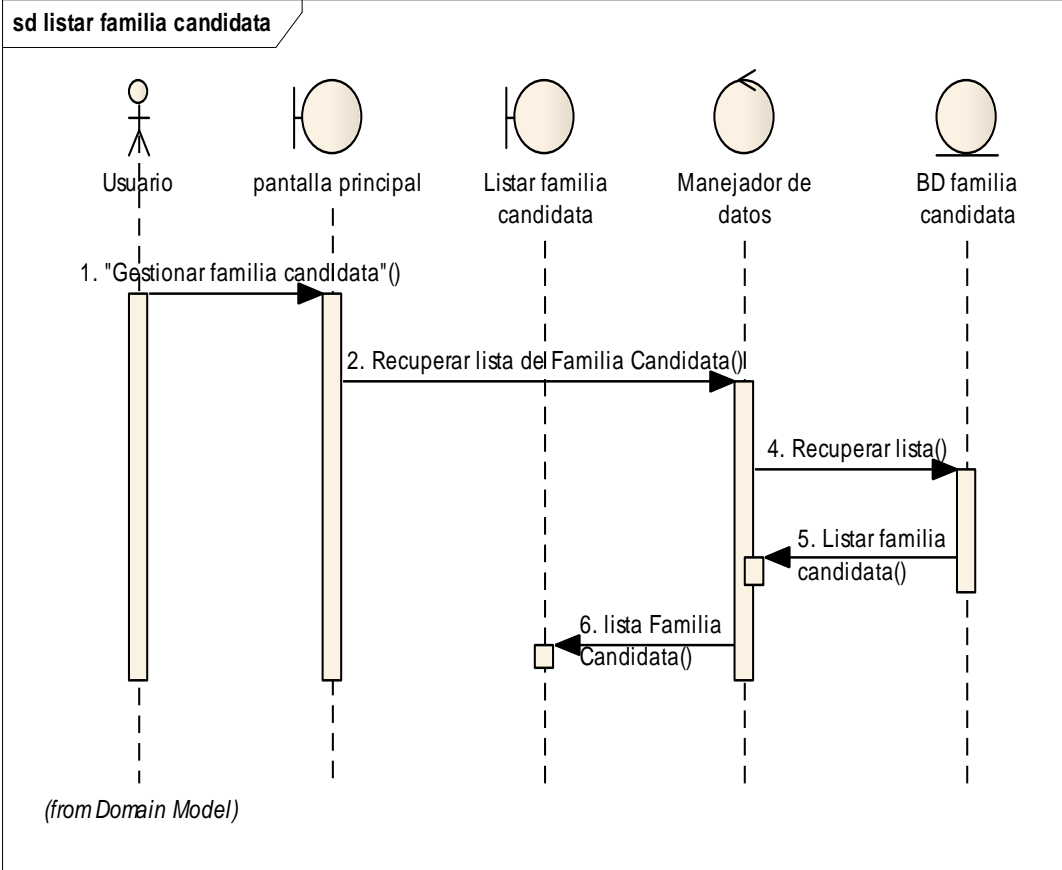


Figura N° 69 Diagrama de secuencia: Listar familia candidata

## Adicionar familia candidata

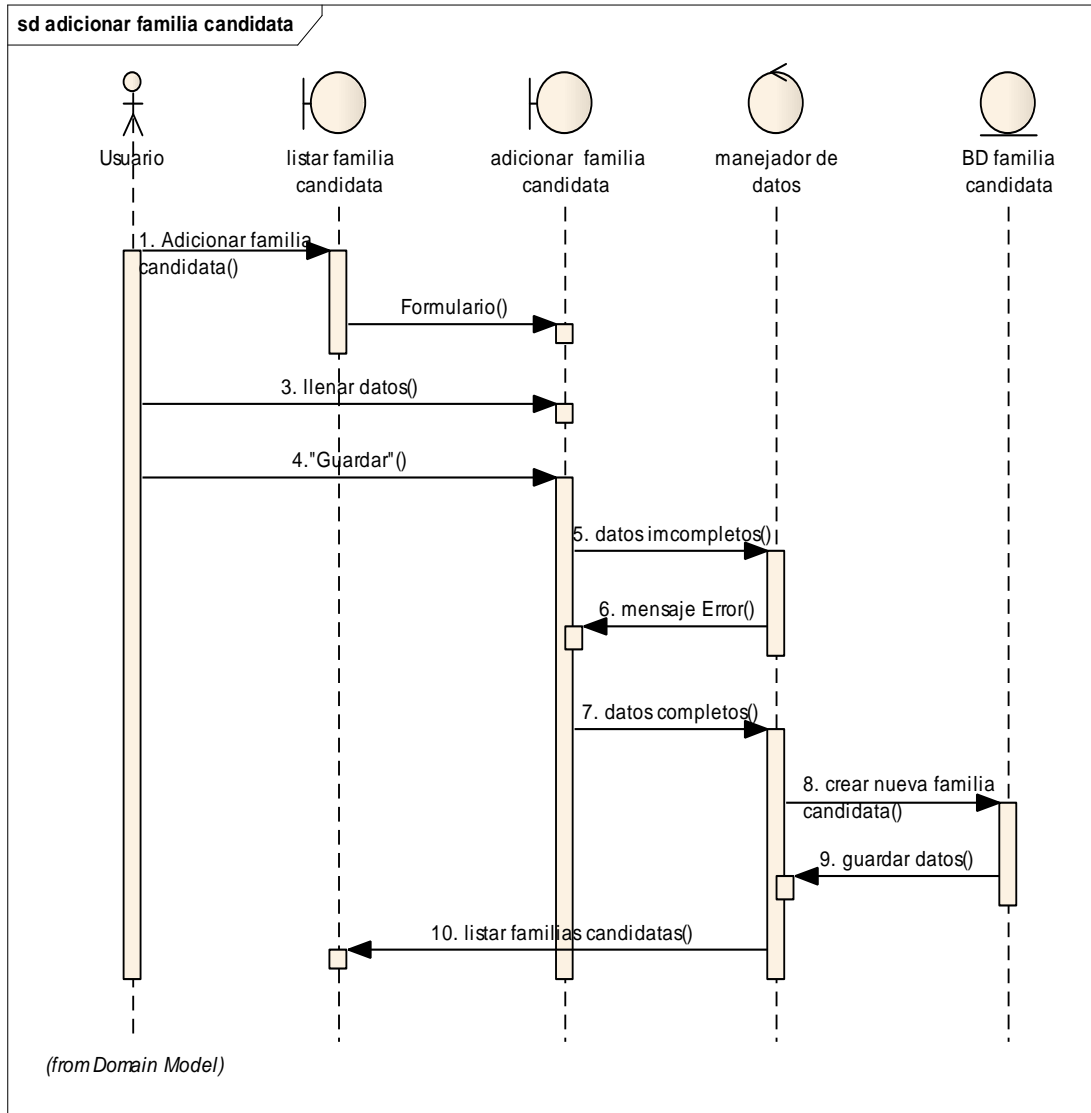


Figura N° 70 Diagrama de secuencia: Adicionar familia candidata

## Eliminar familia candidata

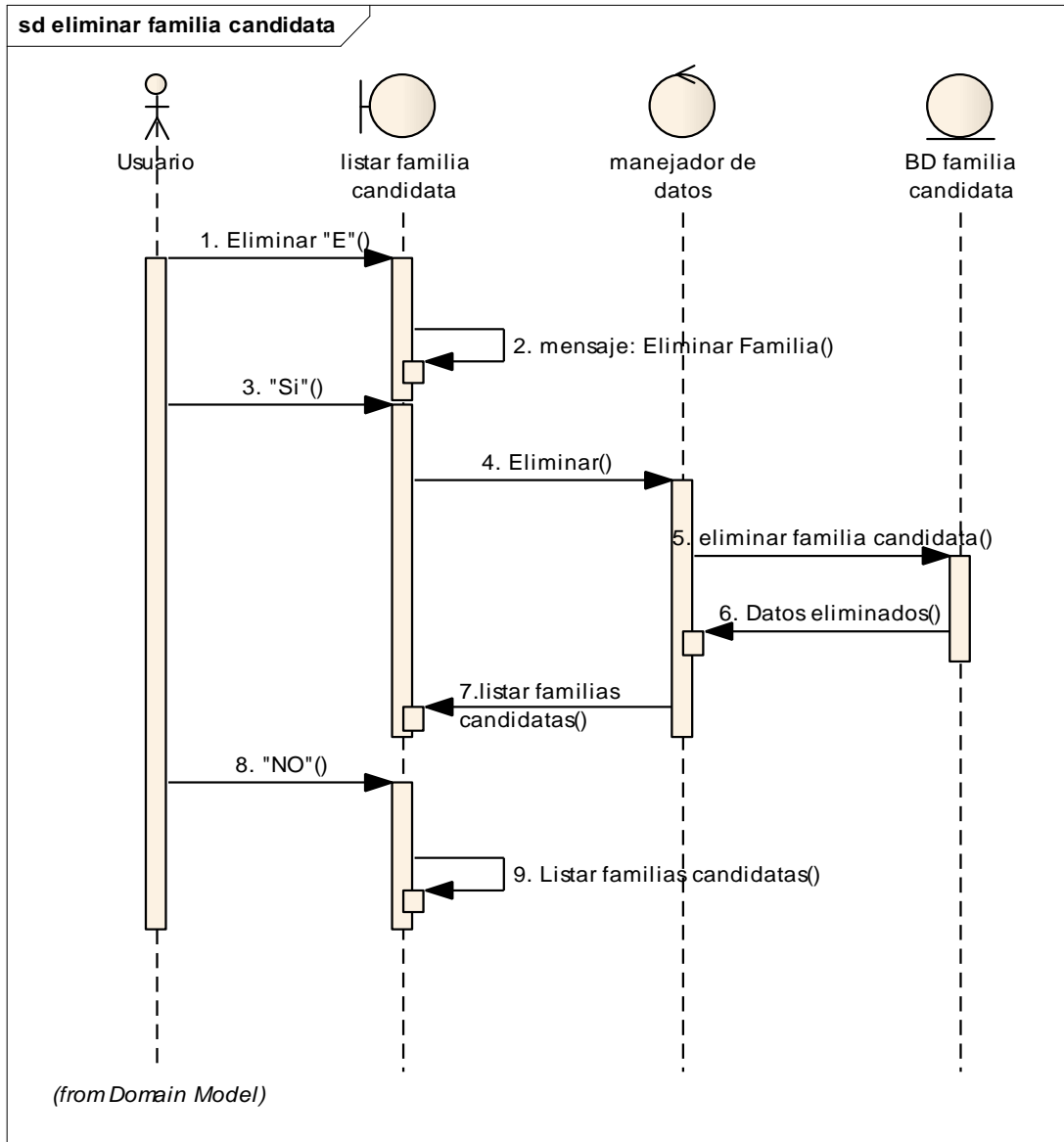


Figura N° 71 Diagrama de secuencia: Eliminar familia candidata

**I.1.6.2.4.5 Gestionar Patrocinador**

**Listar Patrocinador**

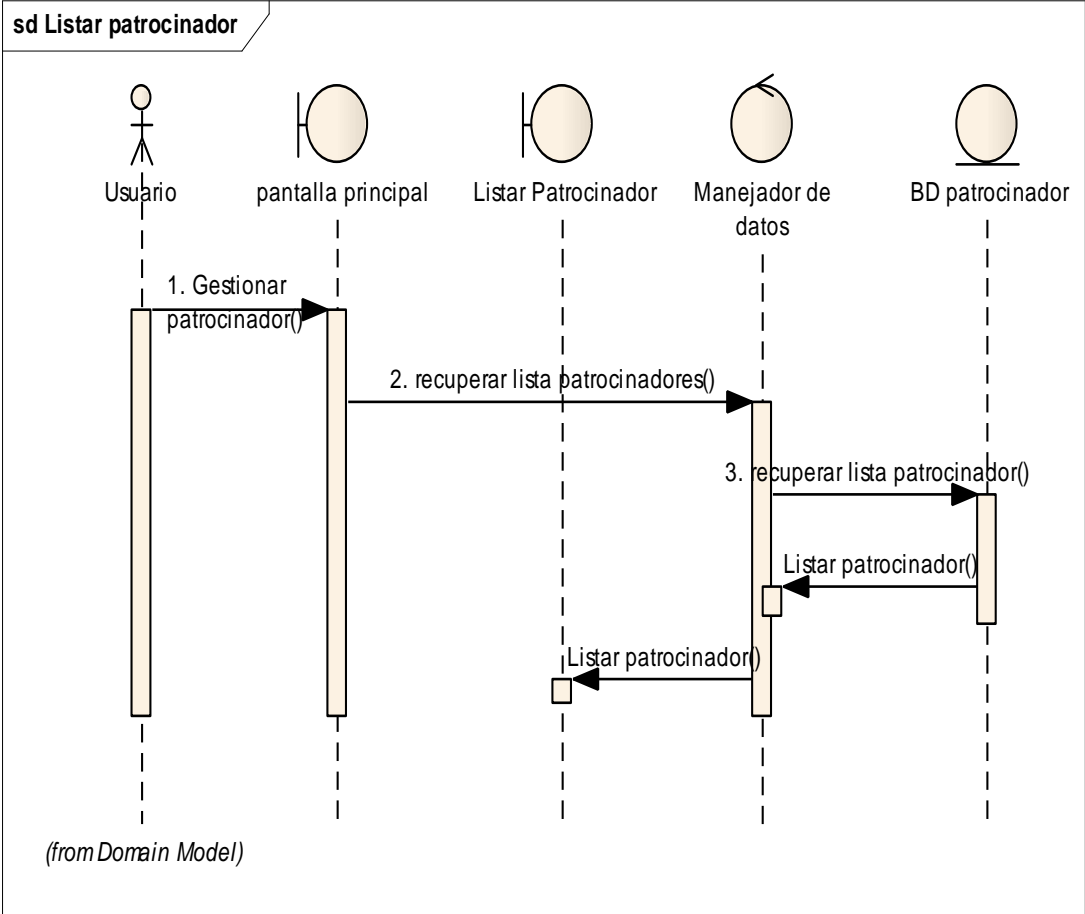


Figura N° 72 Diagrama de secuencia: Listar Patrocinador

## Adicionar Patrocinador

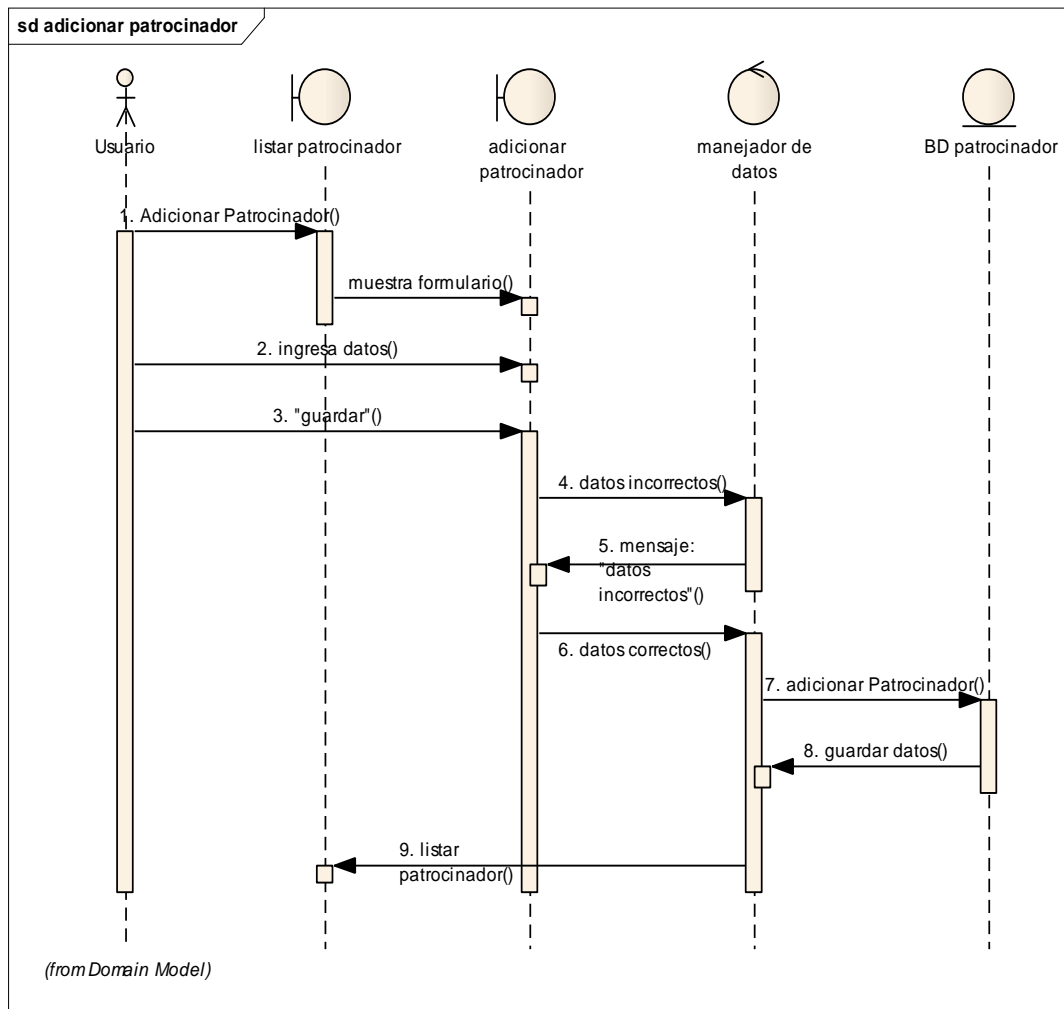


Figura N° 73 Diagrama de secuencia: Adicionar Patrocinador

## Modificar Patrocinador

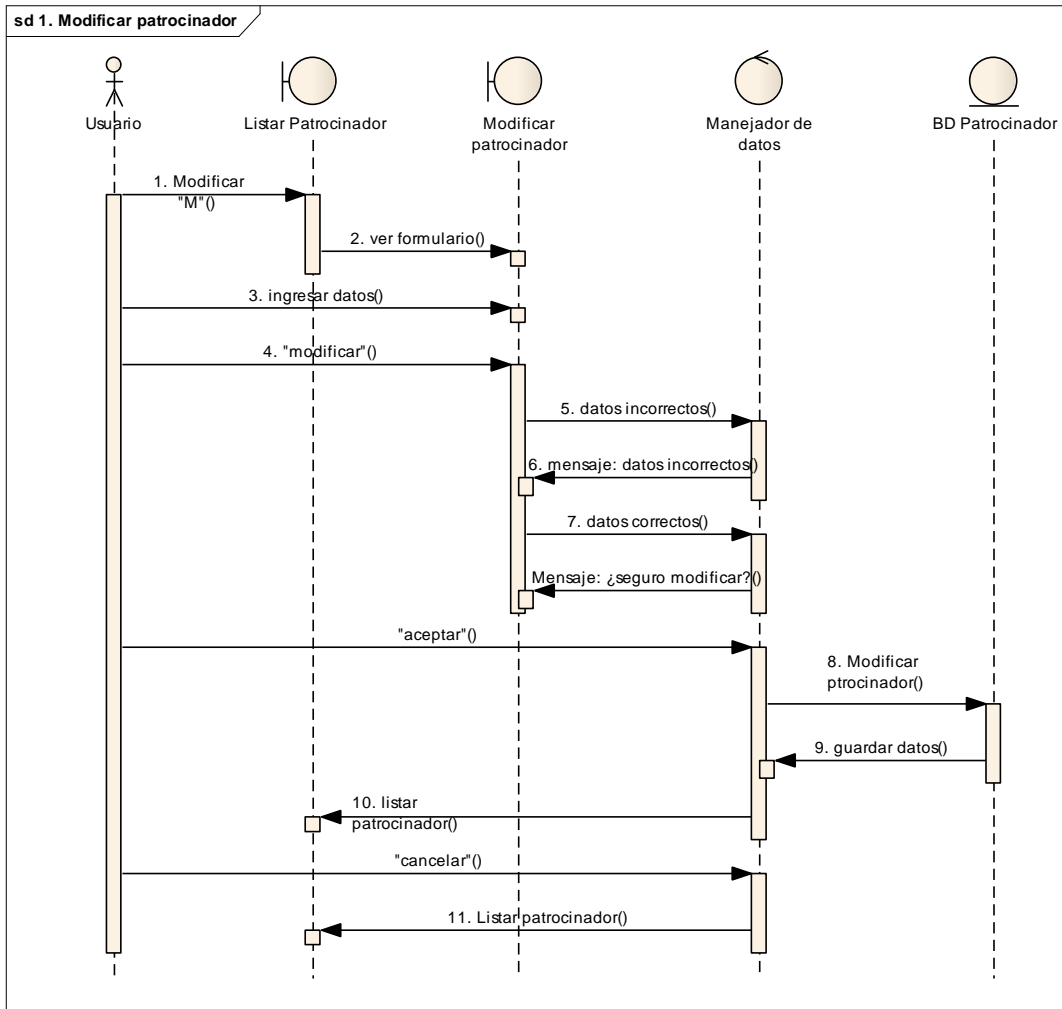


Figura N° 74 Diagrama de secuencia: Modificar Patrocinador

## Eliminar Patrocinador

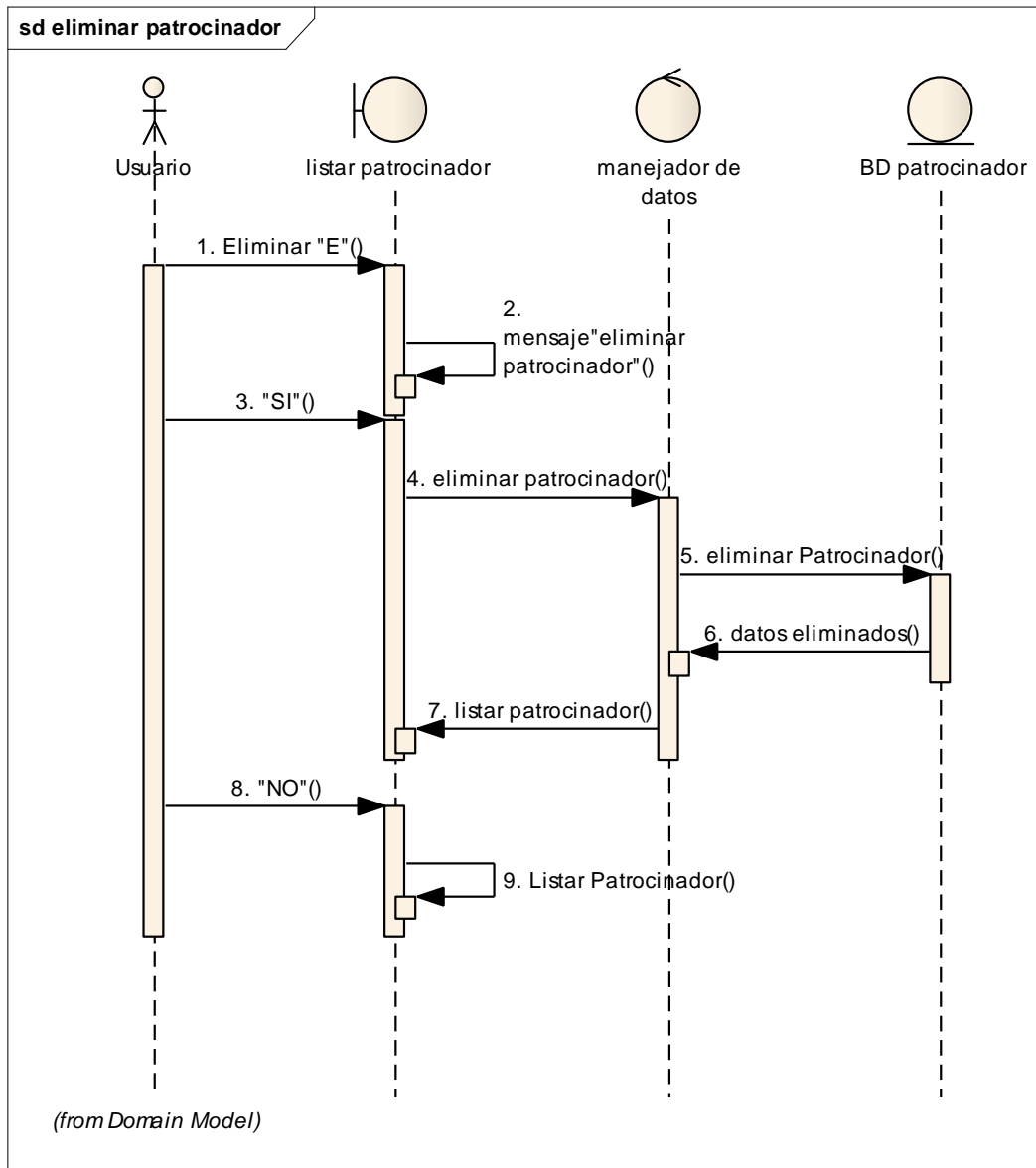


Figura N° 75 Diagrama de secuencia: Eliminar Patrocinador

### 1.1.6.2.4.6 Gestionar afiliado

#### Listar afiliado

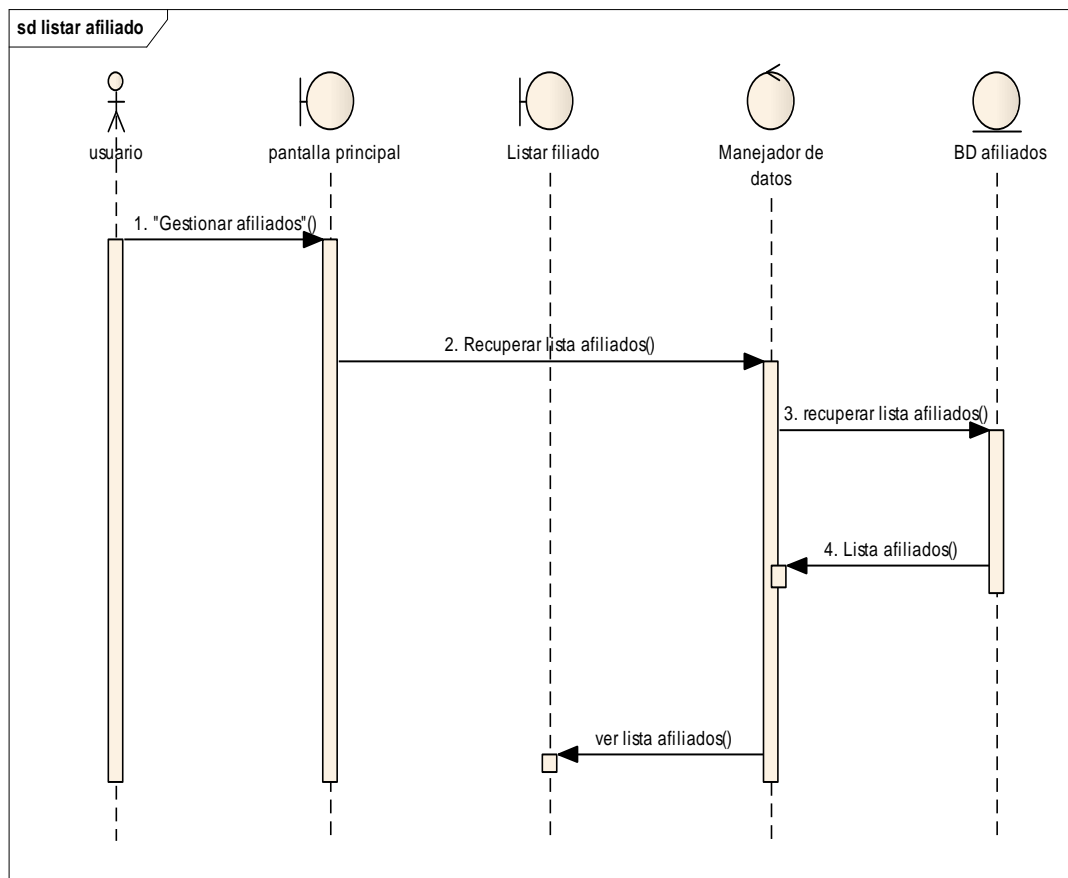


Figura N° 76 Diagrama de secuencia: Listar afiliado

## Modificar afiliado

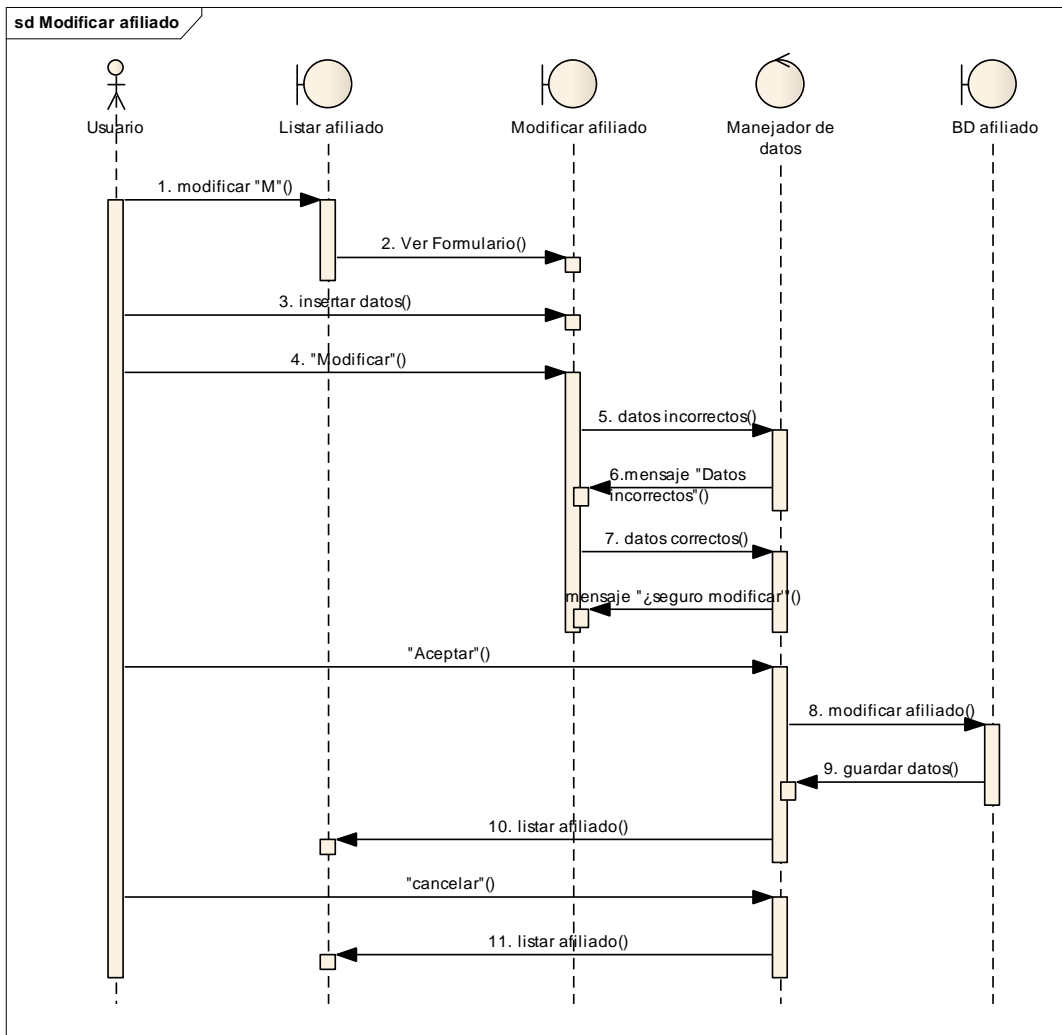


Figura N° 77 Diagrama de secuencia: Modificar afiliado

## Eliminar afiliado

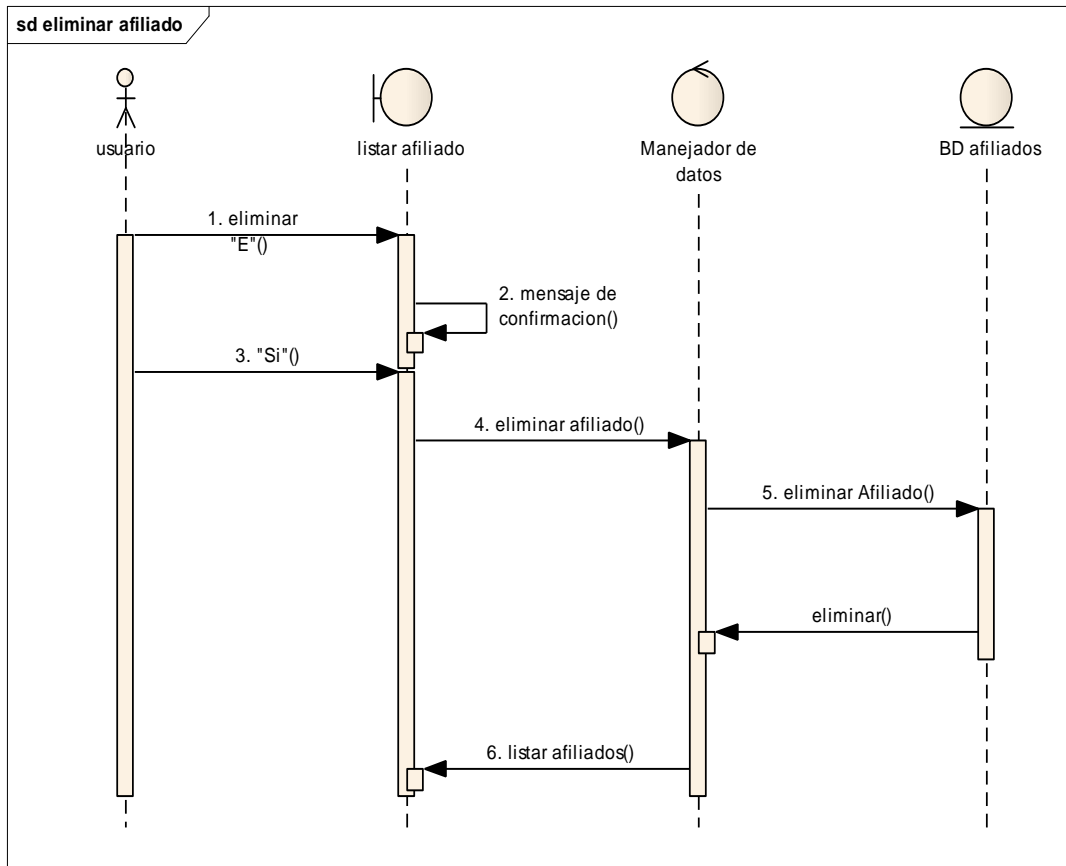


Figura N° 78 Diagrama de secuencia: Eliminar afiliado

**1.1.6.2.4.7 Gestionar Actividades**

**Listar Actividades**

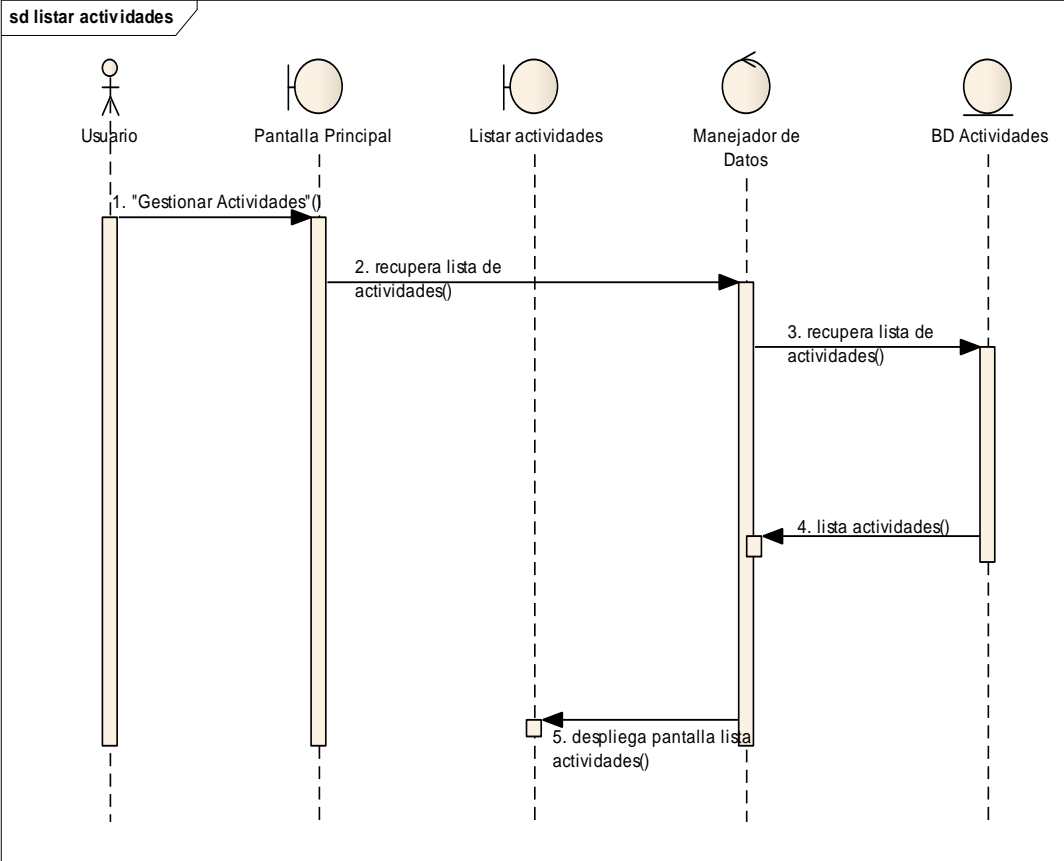


Figura N° 79 Diagrama de secuencia: Listar Actividades

## Adicionar Actividades

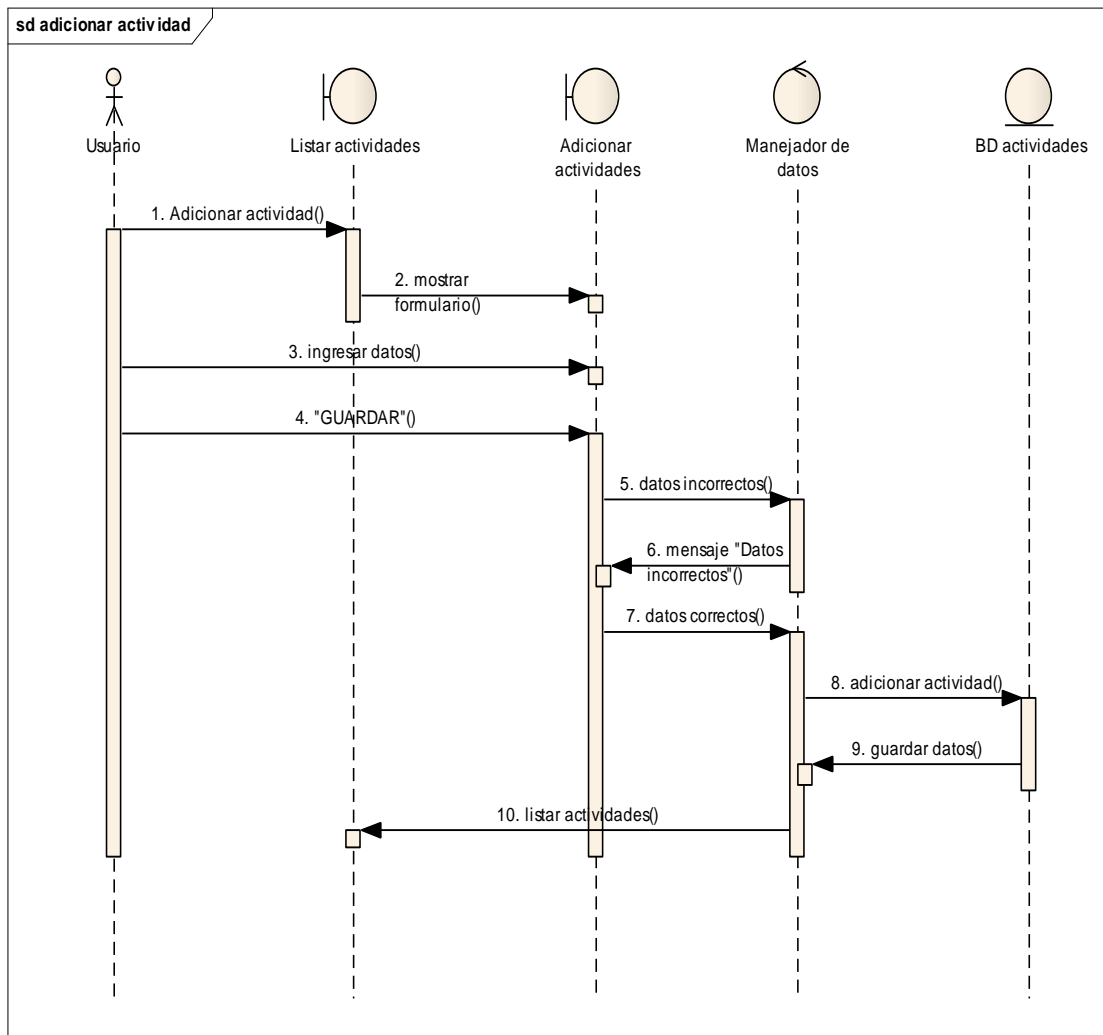


Figura N° 80 Diagrama de secuencia: Adicionar Actividades

## Modificar Actividades

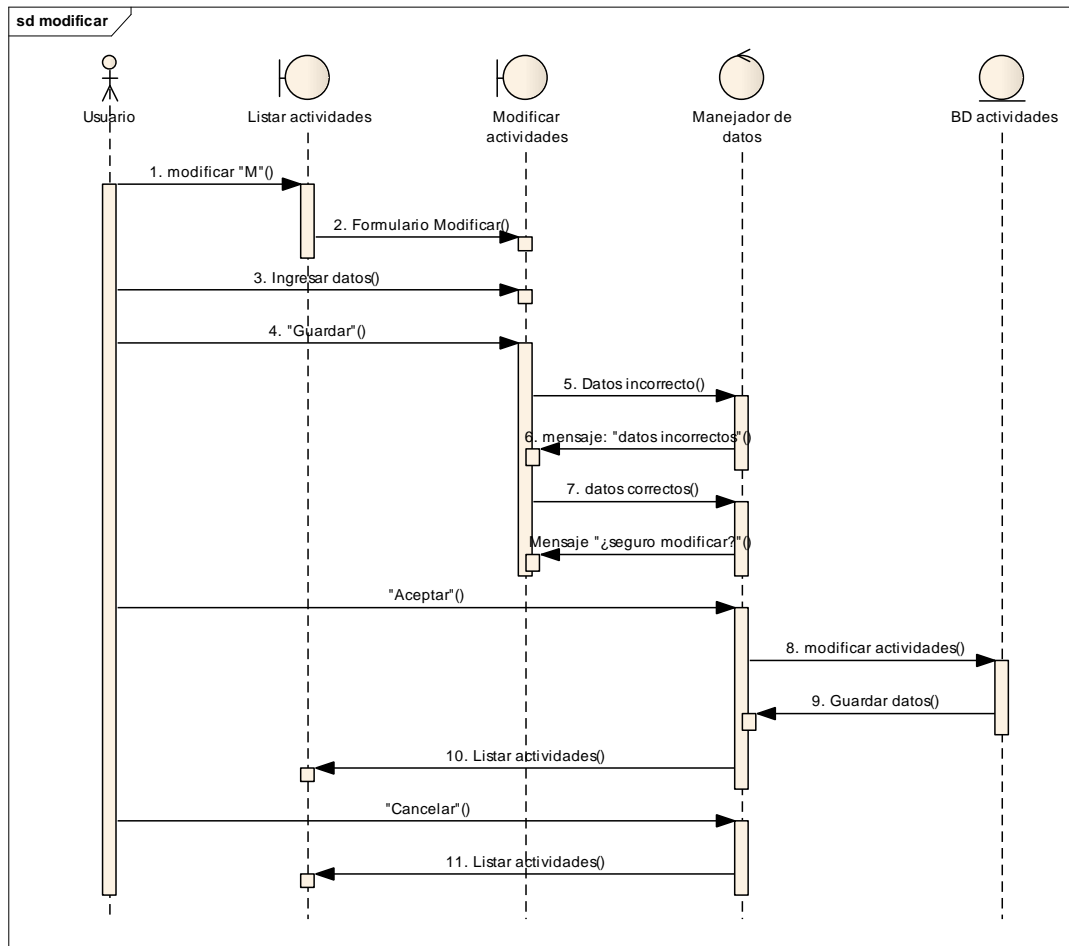


Figura N° 81 Diagrama de secuencia: Modificar Actividades

## Eliminar Actividades

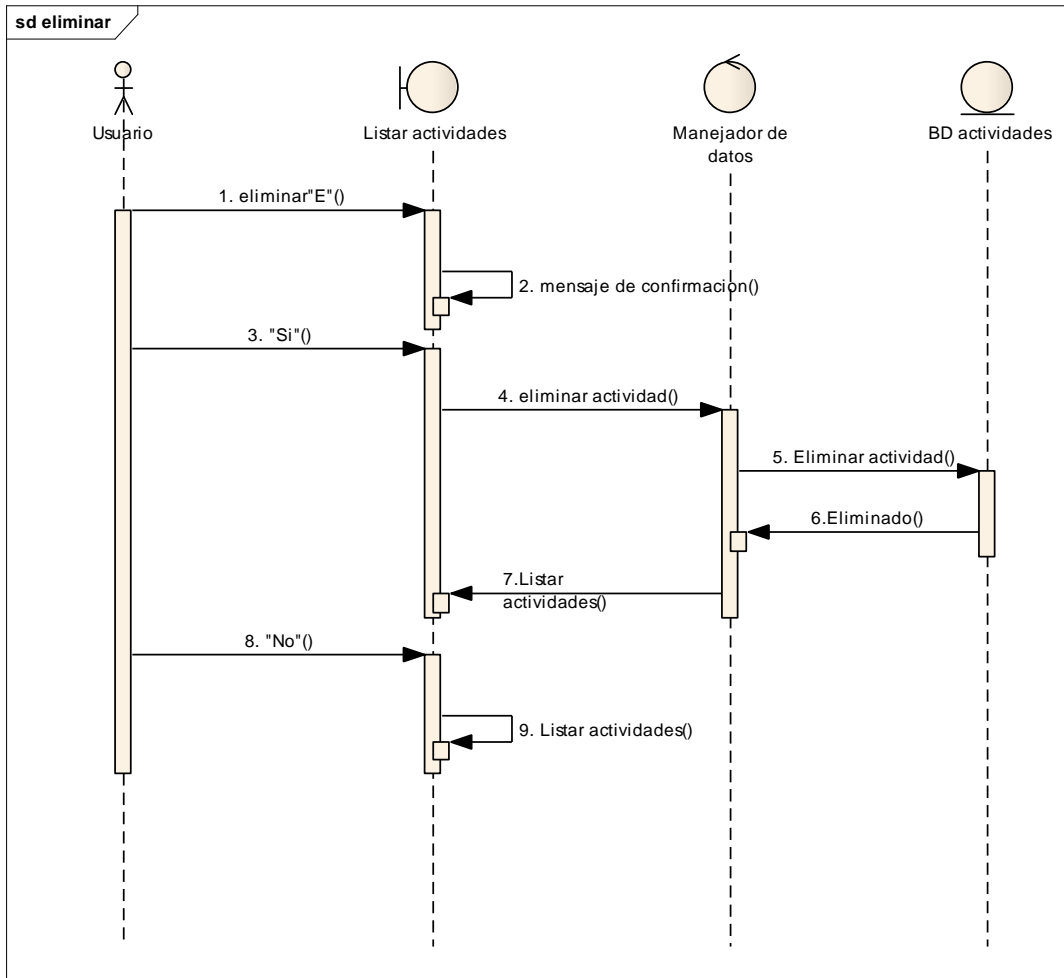


Figura N° 82 Diagrama de secuencia: Eliminar Actividades

**I.1.6.2.4.8 Registrar Nivel Educativo**

**Listar afiliado**

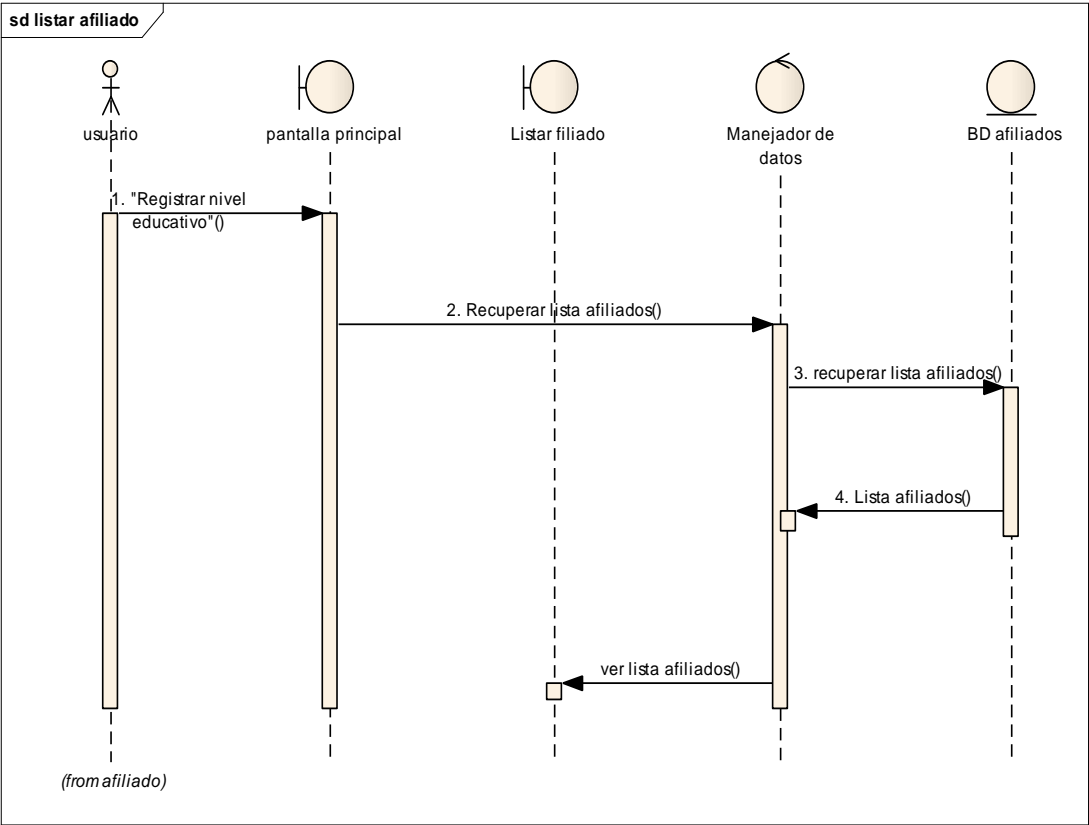


Figura N° 83 Diagrama de secuencia: Listar afiliado

## Llenar formulario

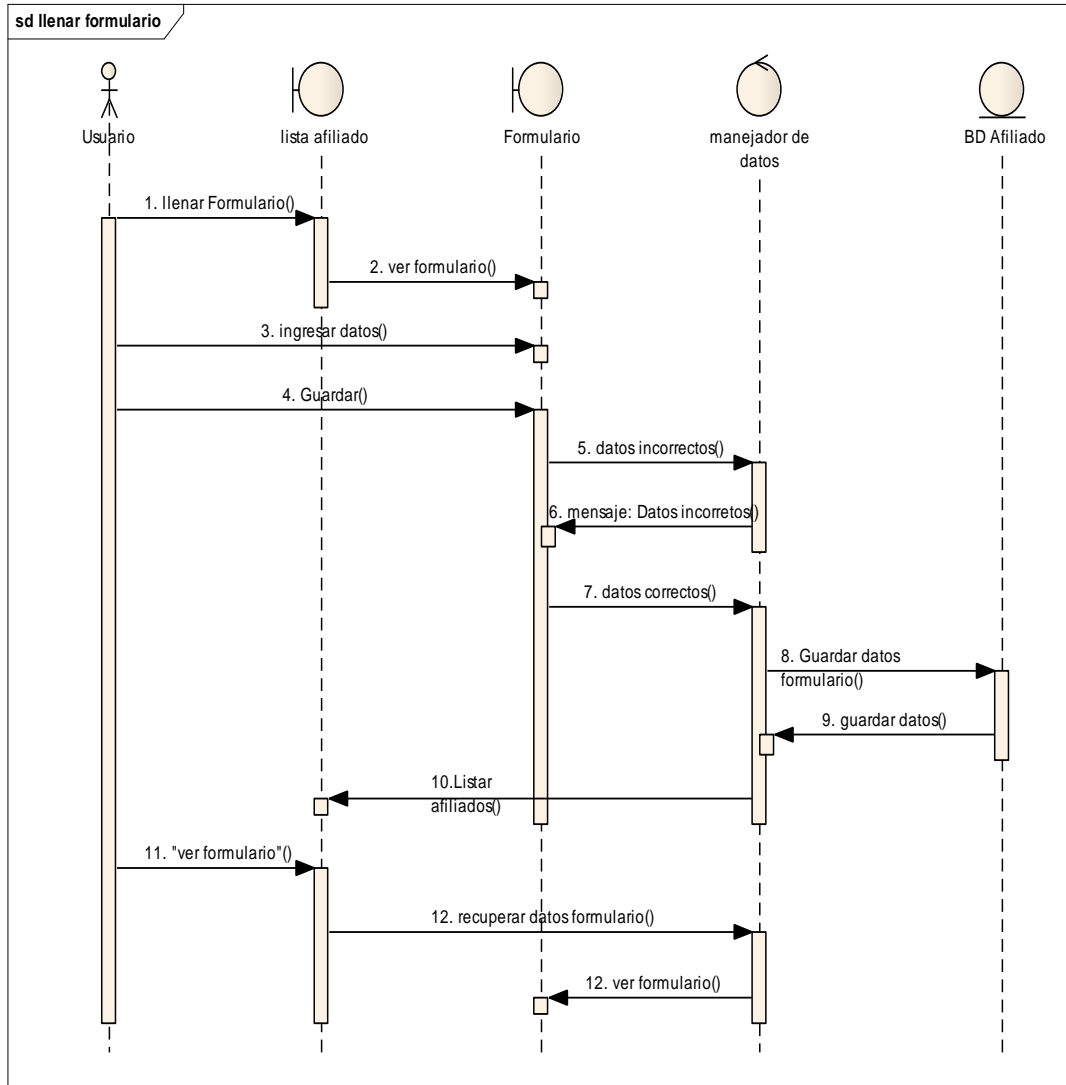


Figura N° 84 Diagrama de secuencia: Llenar formulario

**1.1.6.2.4.9 Gestionar Salud**

**Listar afiliado**

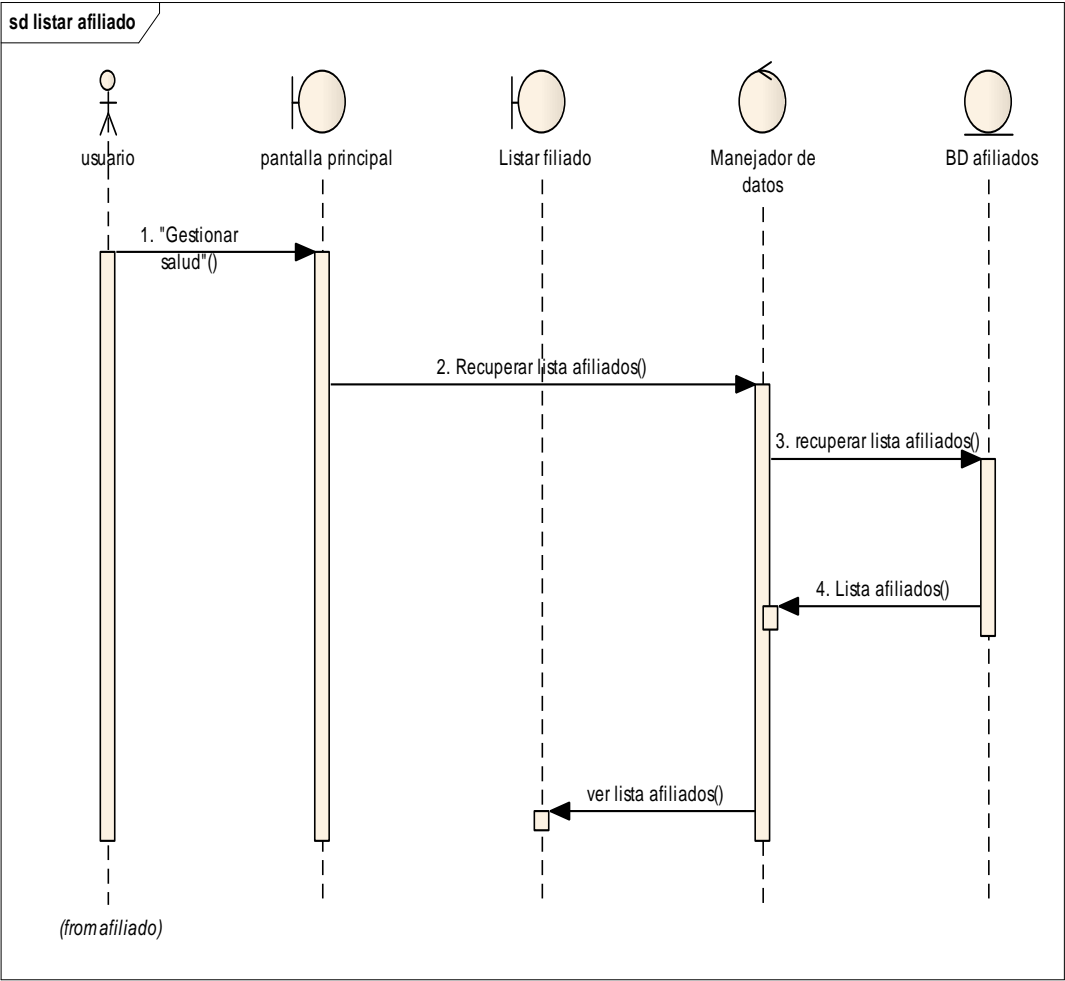


Figura N° 85 Diagrama de secuencia: Listar afiliado

## Llenar formulario Salud

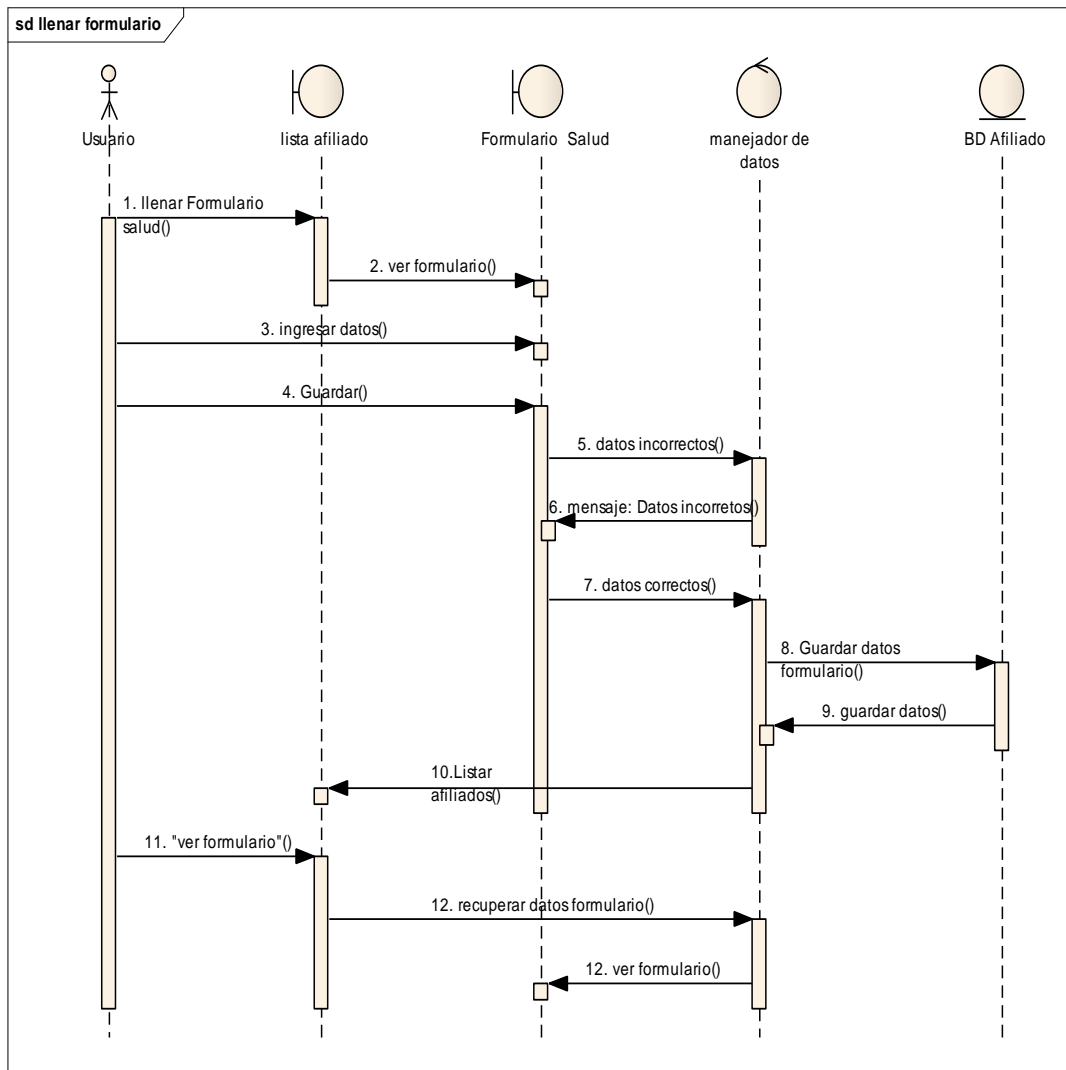


Figura N° 86 Diagrama de secuencia: Llenar formulario salud

### I.1.6.2.4.10 Gestionar Cargo

#### Listar cargo

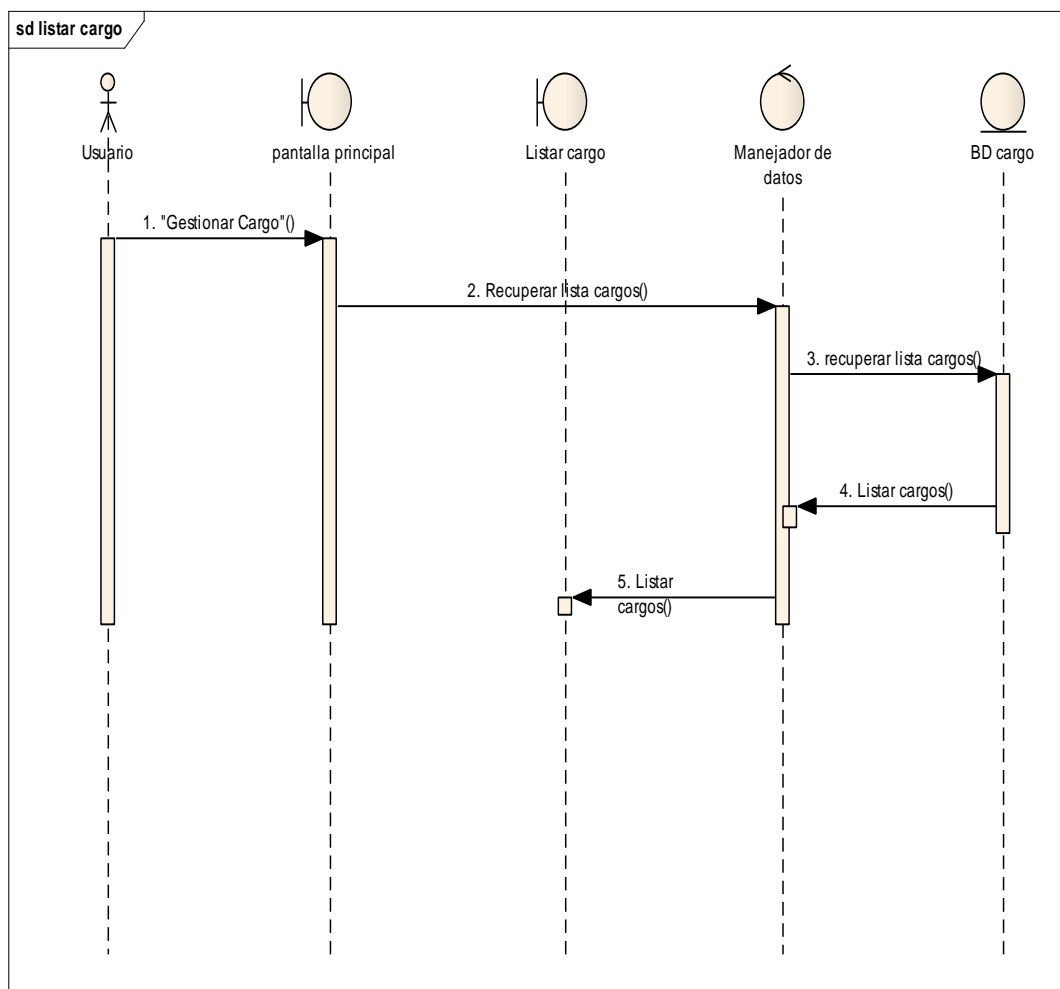


Figura N° 87 Diagrama de secuencia: Listar cargo

## Adicionar Cargo

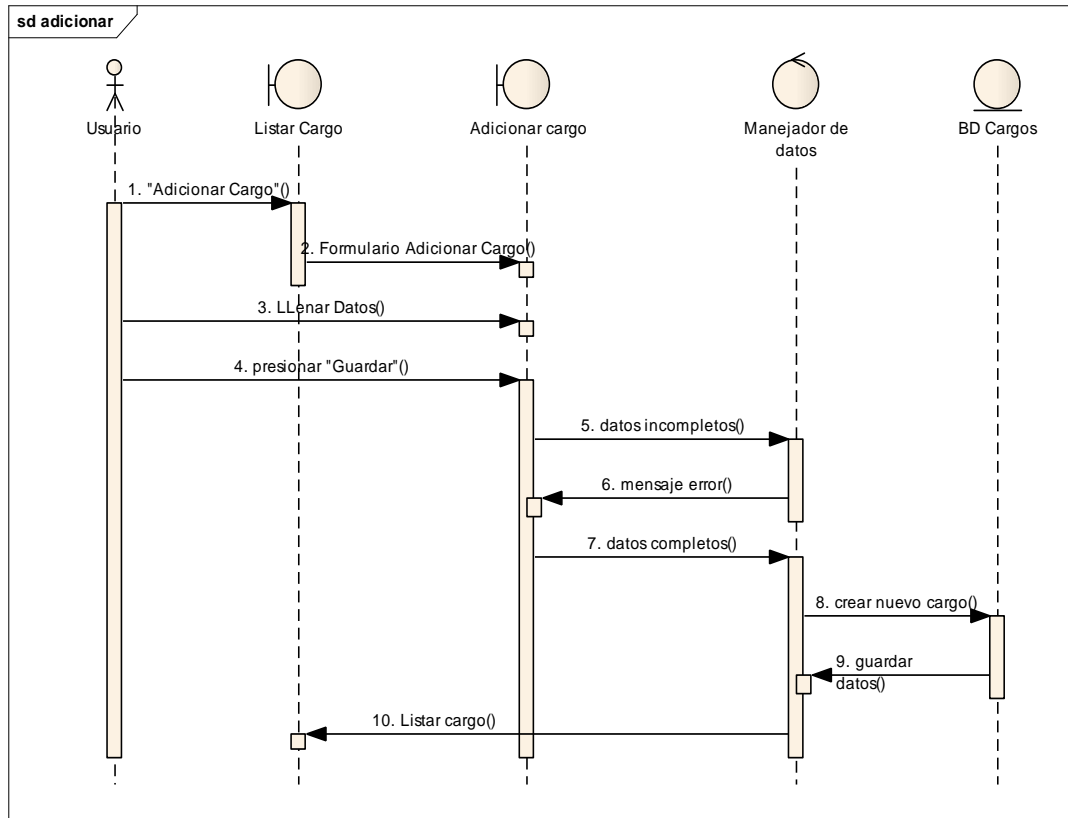


Figura N° 88 Diagrama de secuencia: Adicionar cargo

# Modificar Cargo

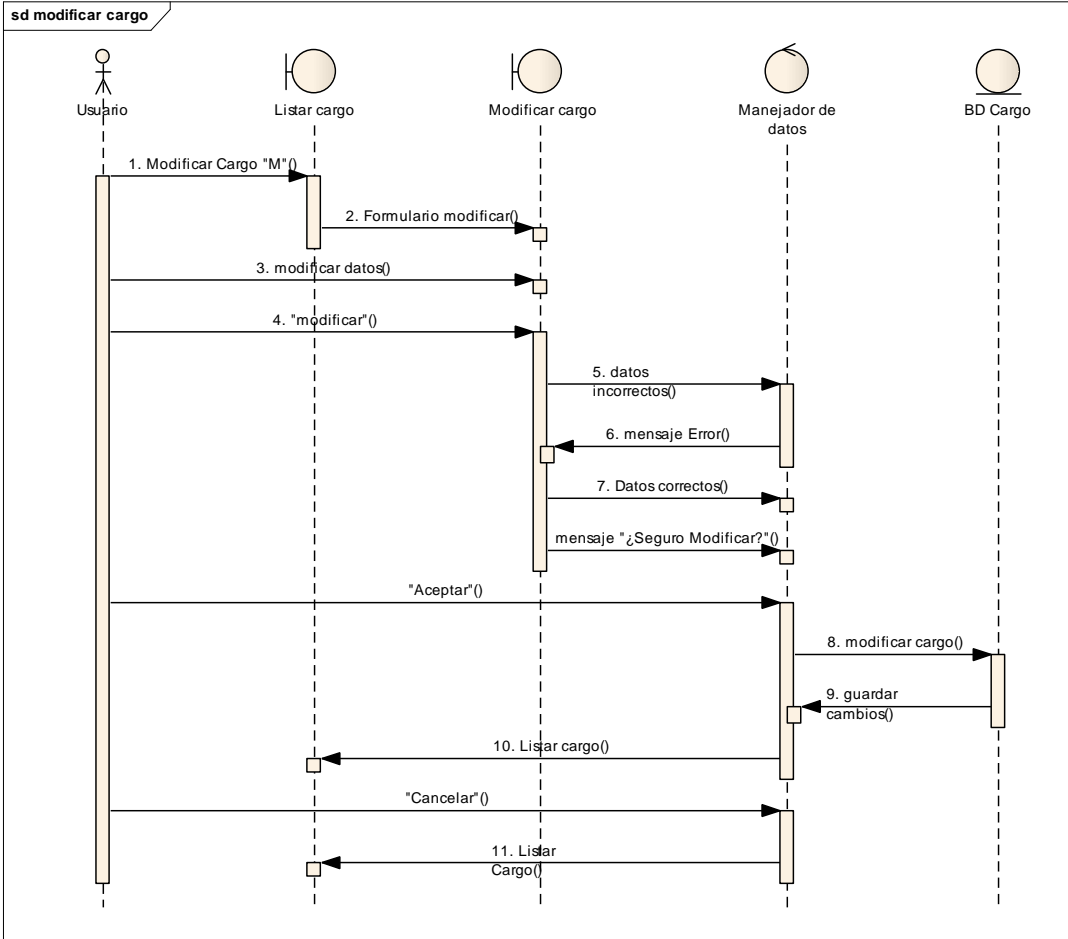


Figura N° 89 Diagrama de secuencia: Modificar cargo

# Eliminar Cargo

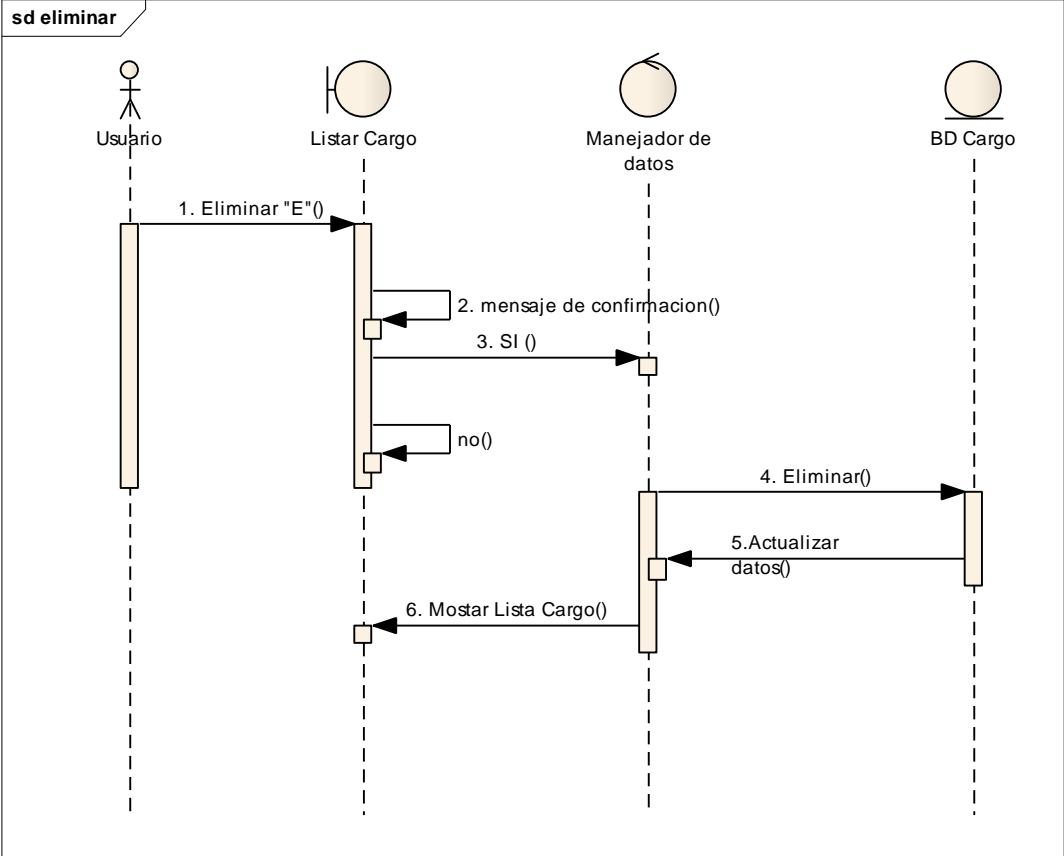


Figura N° 90 Diagrama de secuencia: Eliminar cargo

### **I.1.6.3 PROTOTIPOS DE LA INTERFAZ DE USUARIOS**

#### ***I.1.6.3.1 Introducción***

A continuación se presentan los prototipos de Interfaz gráfica que usara el sistema. Las interfaces graficas del Sistema propuesto serán fácilmente adaptables sin sufrir ningún cambio en el performance de nuestra aplicación en diferentes plataformas Windows

#### ***I.1.6.3.2 Propósito***

Los Prototipos de Interfaz gráfica tienen el propósito de mostrar y describir las interfaces que implementará el sistema para que le usuario pueda comprender mejor y tener una idea de la apariencia y la productividad que podrá tener al usar las interfaces.

#### ***I.1.6.3.3 Alcance***

Los prototipos de interfaz gráfica no representan a las interfaces graficas definitivas que tendrá el sistema. Pueden sufrir algunas modificaciones leves.

#### ***I.1.6.3.4 DISEÑO DE LAS PANTALLAS***

Cada una de las siguientes pantallas son de salida y a su vez, pueden contener enlaces que se convierten en pantallas de entrada. A continuación se presentan los prototipos:

### 1.1.6.3.4.1 Ingresar al sistema

#### Pantalla Ingresar al sistema

Esta pantalla es la pantalla inicial del sistema, en esta pantalla se puede ver una pequeña introducción al sistema, como ser: calendario, iniciar sesión, lista de correspondencia, actividades próximas, información de la institución

**SARC** Sistema Automático de Registro y Control de Información

**Calendario** Fecha: Noviembre 2013

L	M	M	J	V	S	D
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

**BIENVENIDOS**

**INGRESAR AL SISTEMA**

Proyecto Comunitario "Guadalquivir"

**Informacion**

Para ingresar al sistema, se debe contar con un usuario y clave, otorgado por la administradora del Proyecto Comunitario Guadalquivir

**CORRESPONDENCIA**

1. Colque paredes Juan
2. Perez Ruiz Carlos damian
3. Morales Copa MARIA
4. Llanos Lopez Carla
5. Coria Torrez Alba

[Imprimir correspondencia](#)

**ACTIVIDADES PROXIMAS**

DOMINGO -- ASEO GENERAL  
15 - NOV -- REUNION GENERAL  
25 - NOV -- DEPORTE

**INFORMACION**

Email: [proguadalquivir@hotmail.com](mailto:proguadalquivir@hotmail.com)  
Telefono: 66-25336  
Celular: 79245007  
Fax: 66-56998 - 25336

[www.chilefund.com](http://www.chilefund.com)

SARC

Figura N° 91 Pantalla: Ingresar al Sistema

## Iniciar sesión

Es un pequeño formulario donde los usuarios registrados pueden introducir su usuario y clave para ingresar al sistema



INICIAR ..Sesion

USUARIO:

CLAVE:

Figura N° 92 Pantalla: iniciar sesión

## Pantalla mensaje error de acceso

Si los datos de usuario y clave son incorrectos

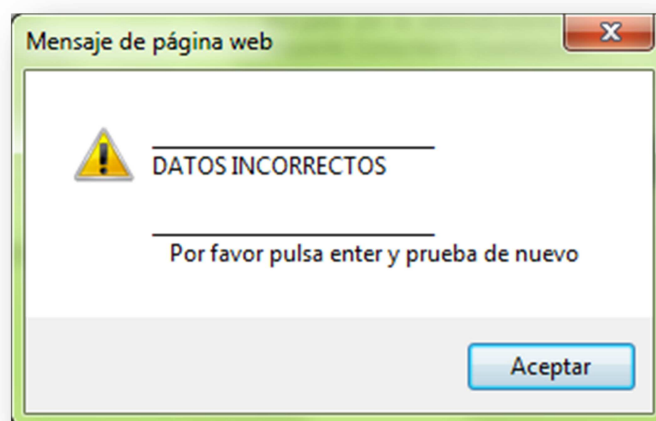


Figura N° 93 Pantalla Mensaje error de acceso

## Pantalla Mensaje Campos vacíos

Mensaje de alerta cuando se introducen campos vacíos

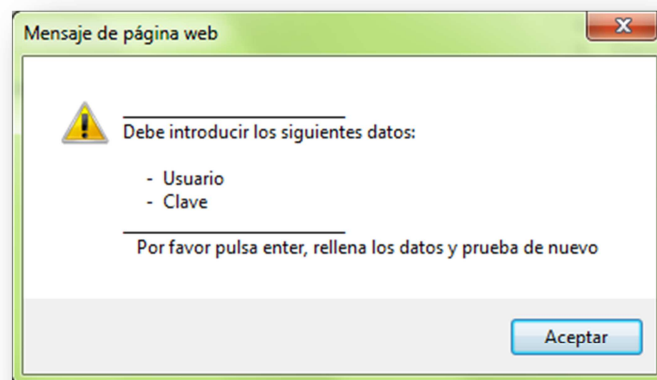


Figura N° 94 Pantalla Mensaje Campos vacíos

#### ***I.1.6.3.4.2 Pantalla Principal del sistema***

### **Pantalla Principal**

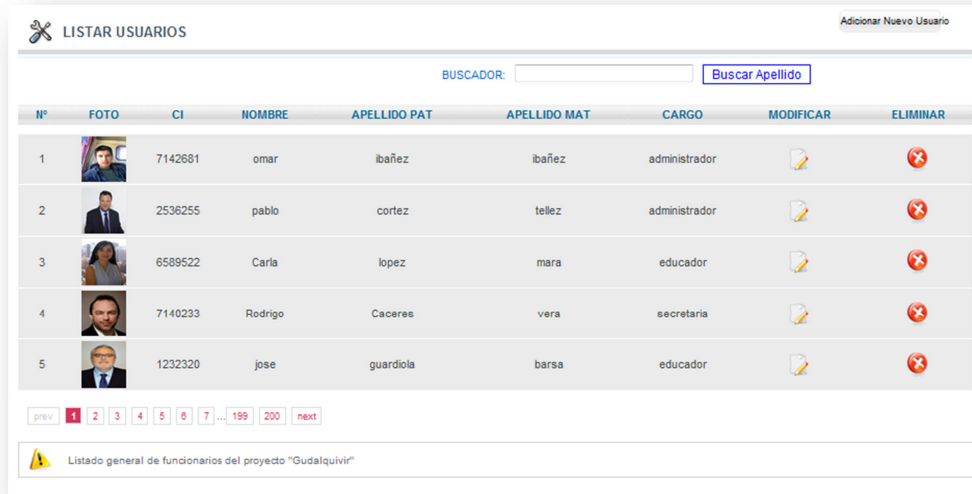
Esta pantalla se muestra una vez validado los datos de usuario y clave en la base de datos, es la pantalla principal del sistema con acceso a todos los procesos



Figura N° 95 Pantalla: Principal

### I.1.6.3.4.3 Gestionar usuarios

#### Pantalla: Listar Usuarios



LISTAR USUARIOS Adicionar Nuevo Usuario

BUSCADOR:

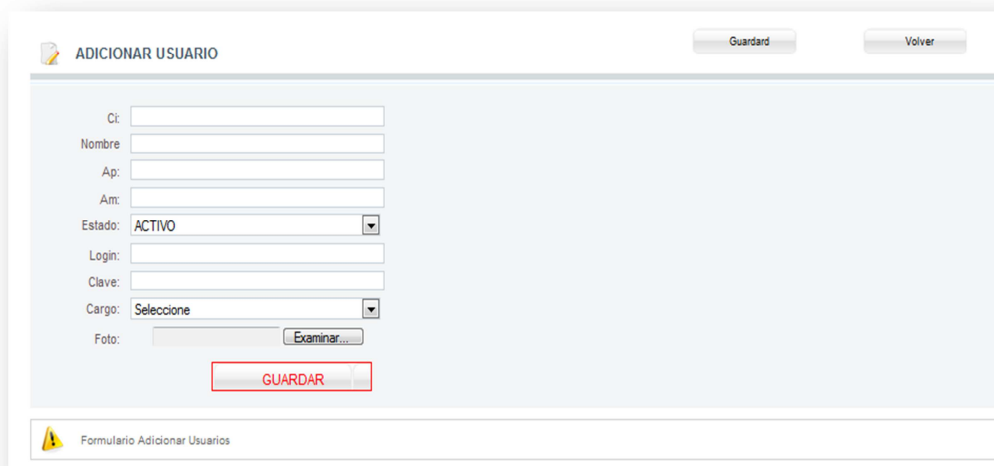
Nº	FOTO	CI	NOMBRE	APELLIDO PAT	APELLIDO MAT	CARGO	MODIFICAR	ELIMINAR
1		7142681	omar	ibañez	ibañez	administrador		
2		2536255	pablo	cortez	telez	administrador		
3		6589522	Carla	lopez	mara	educador		
4		7140233	Rodrigo	Caceres	vera	secretaria		
5		1232320	jose	guardiola	barsa	educador		

prev 1 2 3 4 5 6 7 ... 199 200 next

Listado general de funcionarios del proyecto "Gudalquivir"

Figura N° 96 Pantalla: Listar Usuarios

#### Pantalla: Adicionar Usuarios



ADICIONAR USUARIO

CI:

Nombre

Ap:

Am:

Estado: ACTIVO

Login:

Clave:

Cargo: Seleccione

Foto:

Formulario Adicionar Usuarios

Figura N° 97 Pantalla: Adicionar Usuarios

**Pantalla: Modificar Usuarios**

**MODIFICAR USUARIO**

Guardar Volver

Cod\_u: 62 +

Ct: 7142681

Nombre: omar

Ap: Ibañez

Am: Ibañez

Estado: ACTIVO

Login: omar

Clave: 123

Cargo: administrador

Modificar

Formulario Modificar Usuario

Figura N° 98 Pantalla: Modificar Usuarios

## Pantalla: Eliminar Usuarios

LISTAR USUARIOS Adicionar Nuevo Usuario

BUSCADOR:

Nº	FOTO	CI	NOMBRE	APELLIDO PAT	APELLIDO MAT	CARGO	MODIFICAR	ELIMINAR
1		7142681	omar	ibañez	ibañez	administrador		
2		2536255	pablo	cortez	tellez	administrador		
3		6589522	Carla	lopez	mara	educador		
4		7140233	Rodrigo	Caceres	vera	secretaria		
5		1232320	jose	guardiola	barsa	educador		

prev 1 2 3 4 5 6 7 ... 199 200 next

Listado general de funcionarios del proyecto "Gudalquivir"

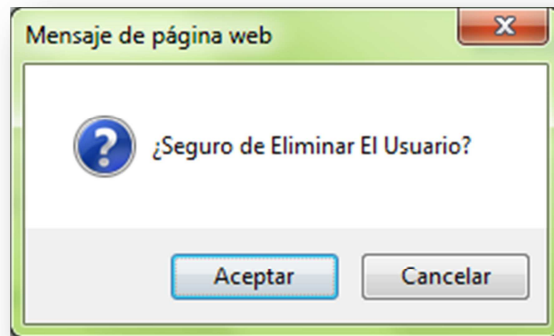
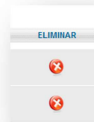


Figura N° 99 Pantalla: Mensaje Eliminar Usuarios

## Pantalla: Asignar Menú

**USUARIO:**

Nombre: omar ibañez ibañez  
Cargo: administrador  
Foto: 

**ASIGNAR**

MENÚ		
N°	MENU	SELECCIONAR
1	GESTIONAR ACTIVIDADES	<input checked="" type="checkbox"/>
2	GESTIONAR USUARIOS	<input checked="" type="checkbox"/>
3	GESTIONAR FAMILIA	<input checked="" type="checkbox"/>
4	GESTIONAR PATROCINADORES	<input checked="" type="checkbox"/>
5	GESTIONAR FAMILIAS CANDIDATAS	<input checked="" type="checkbox"/>
6	CONTROLAR ASISTENCIA	<input checked="" type="checkbox"/>
7	GESTIONAR SALUD	<input checked="" type="checkbox"/>
8	REGISTRAR NIVEL EDUCATIVO	<input checked="" type="checkbox"/>
9	GESTIONAR AFILIADOS	<input checked="" type="checkbox"/>
10	GESTIONAR CARGO	<input checked="" type="checkbox"/>
11	GESTIONAR CORRESPONDENCIA	<input checked="" type="checkbox"/>
12	HACER BACKUP	<input checked="" type="checkbox"/>

Volver

Figura N° 100 Pantalla: Asignar Menú

#### I.1.6.3.4.4 Gestionar familia

### Pantalla: Listar familia

The screenshot displays a web interface for managing families. At the top left, there is a logo with a pair of scissors and the word 'FAMILIAS'. At the top right, there is a button labeled 'Adicionar Nueva Familia'. Below the header, there is a search bar with the label 'BUSCADOR:' and a button labeled 'Buscar Apellido'. The main content is a table with the following columns: 'Nº', 'REPRESENTANTE', 'CODIGO TARJETA', 'Nº INTEGRANTES', 'VER FAMILIA', 'MODIFICAR', and 'ELIMINAR'. The table contains three rows of data. Below the table, there is a pagination control with buttons for 'prev', '1', '2', '3', '4', '5', '6', '7', '...', '199', '200', and 'next'. At the bottom, there is a warning icon and a message: 'Listado general de las familias del proyecto "Gudalquivir"'. The table data is as follows:

Nº	REPRESENTANTE	CODIGO TARJETA	Nº INTEGRANTES	VER FAMILIA	MODIFICAR	ELIMINAR
1	leon carg ss	63	3	<a href="#">ver</a>		
2	jose river fole	64	2	<a href="#">ver</a>		
3	padre asd sad	65	2	<a href="#">ver</a>		

Figura N° 101 Pantalla: Listar familia

## Pantalla: Adicionar familia

The screenshot shows a web form titled "ADICIONAR FAMILIA" with two main sections: "Integrantes" and "REPRESENTANTE".

**Integrantes:**

- Integrantes:
- Dirección:
- Fecha: **Miércoles, 20 de Noviembre**

**REPRESENTANTE:**

- Ci:
- Nombre:
- Apellido Paterno:
- Apellido Materno:
- Fecha de Nac:
- Sexo: **Masculino** (dropdown)
- Telefono:
- Price:
- Estado: **Activo** (dropdown)
- Parentesco: **Padre** (dropdown)
- Afilado: **S** (dropdown)
- Foto:

Buttons:

**INTEGRAR NUEVA FAMILIA, Y EL REPRESENTANTE**

Figura N° 102 Pantalla: adicionar familia

## Pantalla: adicionar afiliado

The screenshot shows a web form titled "ADICIONAR FAMILIA" with two main sections: "Integrantes" and "AFILIADOS".

**Integrantes:**

- Integrantes: **2**
- Dirección: **la loma**
- Fecha: **Miércoles, 20 de Noviembre**

**AFILIADOS:**

- Ci:
- Nombre:
- Apellido Paterno:
- Apellido Materno:
- Fecha de Nac:
- Sexo: **Masculino** (dropdown)
- Telefono:
- Price:
- Estado: **Activo** (dropdown)
- Parentesco: **Padre** (dropdown)
- Afilado: **S** (dropdown)
- Foto:

Buttons:

**INTEGRAR MIEMBROS DE LA FAMILIA**

Figura N° 103 Pantalla: adicionar afiliado

## Pantalla: Modificar Familia

<b>TARJETA: 74</b> Codigo: 73 Nombre: Carlos Direccion: Rosedal Telf: 72996681 Integrantes: 2	PADRES / APODERADO						
	N°	CI	FOTO	NOMBRE	FECHA DE NACIMIENTO	MODIFICAR	ELIMINAR
	1	2145236		Carlos	1965-11-06		
	HIJOS						
	N°	CI	FOTO	NOMBRE	FECHA DE NACIMIENTO	MODIFICAR	ELIMINAR
	1	1234569		Limbania Torrico Lopez	1988-05-10		

 MODIFICAR FAMILIA

Figura N° 104 Pantalla: Modificar Familia

## Pantalla: Eliminar Familia

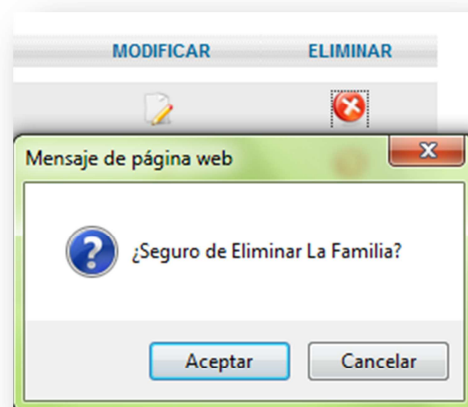


Figura N° 105 Pantalla: Eliminar Familia

## Pantalla: Ver Familia

**TARJETA: 74**

Codigo: 73  
Nombre: Carlos Torrico Orozco  
Direccion: Rosedal  
Telf: 72996681  
Integrantes: 2  
familia:

**PADRES / APODERADO**

N°	CI	FOTO	NOMBRE	FECHA DE NACIMIENTO
1	2145236		Carlos Torrico Orozco	1965-11-06

**HIJOS**

N°	CI	FOTO	NOMBRE	FECHA DE NACIMIENTO
1	1234569		Limbania Torrico Lopez	1988-05-10

Figura N° 106 Pantalla: Ver Familia

### I.1.6.3.4.5 Gestionar Familia Candidata

## Pantalla: Listar Familia Candidata

**FAMILIAS CANDIDATAS** Adicionar Familia Candidata

BUSCADOR:  Buscar Apellido

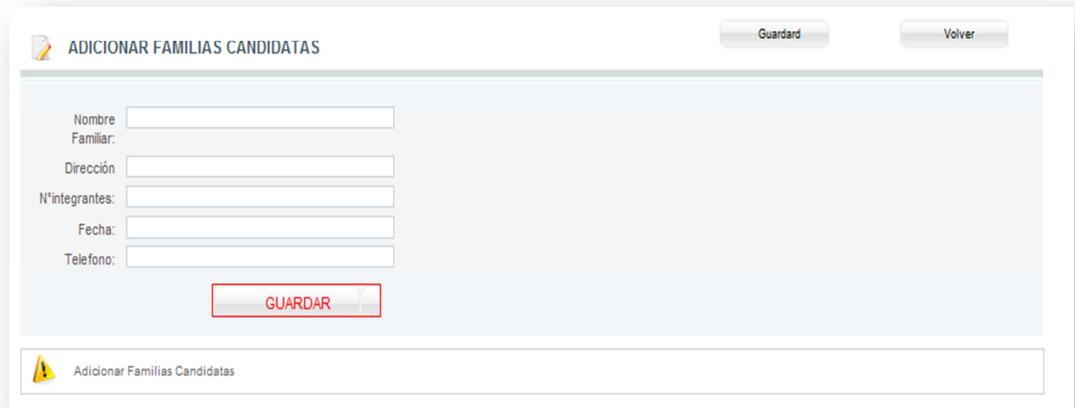
N°	NOMBRE FAMILIA	DIRECCION	N° INTEGRANTES	FECHA	TELEFONO	MODIFICAR	ELIMINAR
1	rojas	las palmas	8	20-11-2013	72965585		
2	Gustavo Flores	B/ luis espinal	5	20-11-2013	72965585		

prev 1 2 3 4 5 6 7 ... 199 200 next

Listado general de funcionarios del proyecto "Gudalquivir"

Figura N° 107 Pantalla: Listar Familia Candidata

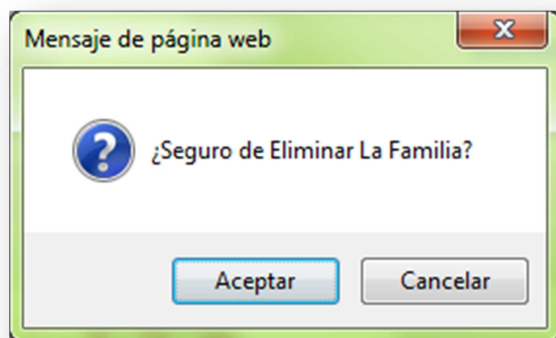
### **Pantalla: Adicionar Familia Candidata**



The screenshot shows a web form titled "ADICIONAR FAMILIAS CANDIDATAS". At the top right, there are two buttons: "Guardar" and "Volver". The form contains several input fields: "Nombre", "Familiar:", "Dirección", "N°integrantes:", "Fecha:", and "Telefono:". A red-bordered button labeled "GUARDAR" is positioned below the input fields. At the bottom left, there is a yellow warning icon and the text "Adicionar Familias Candidatas".

Figura N° 108 Pantalla: Adicionar Familia Candidata

### **Pantalla: Eliminar Familia Candidata**



The screenshot shows a confirmation dialog box titled "Mensaje de página web". It features a question mark icon and the text "¿Seguro de Eliminar La Familia?". At the bottom, there are two buttons: "Aceptar" and "Cancelar".

Figura N° 109 Pantalla: Eliminar Familia Candidata

### I.1.6.3.4.6 Gestionar patrocinadores

#### Pantalla: Listar Patrocinadores

The screenshot displays a web interface for managing sponsors. At the top left, there is a logo and the text 'PATROCINADORES'. At the top right, there is a link 'Adicionar Nuevo PatrocinadorR'. Below this is a search bar with the label 'BUSCADOR:' and a button 'Buscar Apellido'. The main content is a table with the following columns: 'N°', 'NOMBRE', 'APELLIDO PAT', 'APELLIDO MAT', 'NACIONALIDAD', 'VER', 'MODIFICAR', and 'ELIMINAR'. The table contains two rows of data. Below the table is a pagination control with buttons for 'prev', '1', '2', '3', '4', '5', '6', '7', '...', '199', '200', and 'next'. At the bottom, there is a warning icon and the text 'Listado de Patrocinadores'.

N°	NOMBRE	APELLIDO PAT	APELLIDO MAT	NACIONALIDAD	VER	MODIFICAR	ELIMINAR
1	Nelson	Ramos	Velez	Argentina			
2	josuelo	muller	kurt	alemania			

Figura N° 110 Pantalla: Listar Patrocinadores

## Pantalla: Adicionar Patrocinador

ADICIONAR PATROCINADOR

Guardar Volver

Nombre

Apellido Paterno:

Apellido Materno:

Nacionalidad:

Correo Electronico:

Telefono:

Guardar

Adicionar Datos del Patrocinador

Figura N° 111 Pantalla: Adicionar Patrocinador

## Pantalla: Modificar Patrocinador

MODIFICAR PATROCINADOR

Guardar Volver

Codigo: 156

Cod\_p: 17

Nombre: Nelson

Apellido Paterno: Ramos

Apellido Materno: Velez

nacionalidad: Argentina

Modificar

Modificar los Datos del Patrocinador

Figura N° 112 Pantalla: Modificar Patrocinador

### I.1.6.3.4.7 Gestionar Actividades

#### Pantalla: Listar Actividades

N°	NOMBRE DE ACTIVIDAD	FECHA INICIO	FECHA FIN	VER	MODIFICAR	ELIMINAR
1	aseo	2013-11-11	2013-11-26	ver		
2	Campeonato	2013-11-21	2013-11-27	ver		
3	Reunion gral	2013-11-22	2013-11-22	ver		

prev 1 2 3 4 5 6 7 ... 199 200 next

Listado general de Actividades del proyecto "Gudaquivir"

Figura N° 113 Pantalla: Listar Actividades

#### Pantalla: Adicionar Actividades

Nombre:

Fecha Inicio:

Fecha Fin:

Hora:

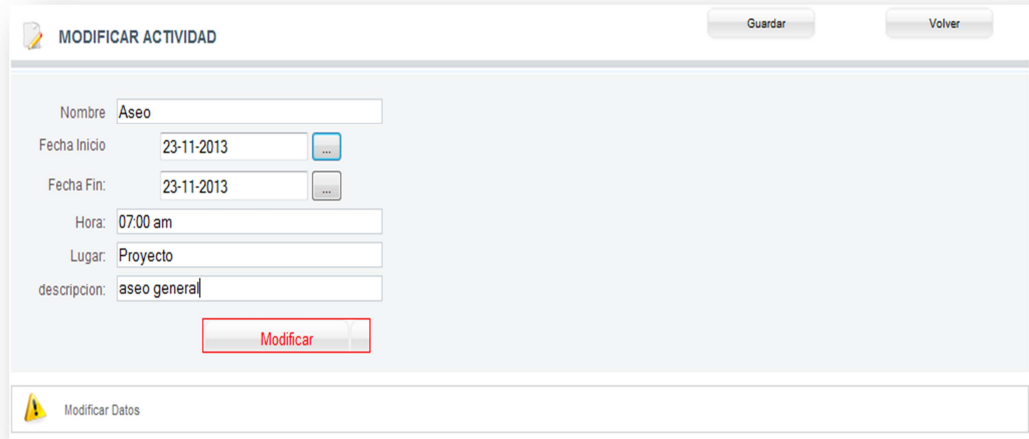
Lugar:

descripcion:

Adicionar Datos de Nueva Actividad

Figura N° 114 Pantalla: Adicionar Actividades

### Pantalla: Modificar Actividad



The screenshot shows a web application window titled "MODIFICAR ACTIVIDAD". At the top right, there are two buttons: "Guardar" and "Volver". The main area contains several input fields: "Nombre" with the value "Aseo", "Fecha Inicio" and "Fecha Fin" both set to "23-11-2013", "Hora" set to "07:00 am", "Lugar" set to "Proyecto", and "descripcion" set to "aseo general". A "Modificar" button is located below the description field. At the bottom left, there is a warning icon and the text "Modificar Datos".

Figura N° 115 Pantalla: Modificar Actividad

### Pantalla: Eliminar Actividad

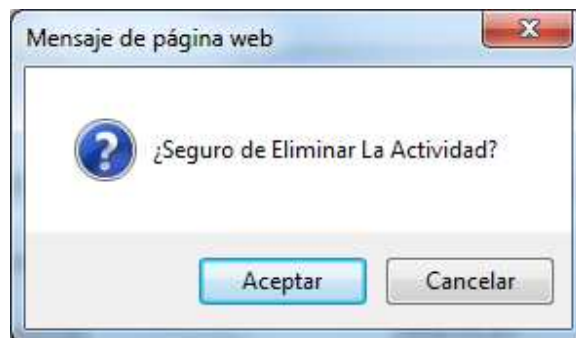


Figura N° 116 Pantalla: Eliminar Actividad

### I.1.6.3.4.8 Controlar Asistencia

#### Pantalla: Listar Actividades



The screenshot shows a web interface for 'CONTROLAR ASISTENCIA'. At the top right, there is a button 'Adicionar Nueva Actividad'. Below the header, there is a search bar labeled 'BUSCADOR:' with a 'Buscar Actividad' button. The main content is a table with the following data:

N°	NOMBRE DE ACTIVIDAD	FECHA INICIO	FECHA FIN	ASISTENCIA
1	ASEO	2013-11-11	2013-11-26	
2	PASEO	2013-11-22	2013-12-03	
3	CURSO TALLER	2013-12-02	2013-12-03	
4	SEMINARIO	2013-12-19	2013-12-20	

Below the table is a pagination control with buttons for 'PREV', '1', '2', '3', '4', '5', '6', '7', '...', '199', '200', and 'next'. At the bottom, there is a warning icon and the text 'Listado general de Actividades del proyecto "Gudalquivir"'. The title 'CONTROLAR ASISTENCIA' is on the left, and 'Adicionar Nueva Actividad' is on the right.

Figura N° 117 Pantalla: Listar Actividades

#### Pantalla: Registrar Asistencia



The screenshot shows a web interface for 'MARCAR ASISTENCIA'. At the top right, there are buttons for 'REGISTRAR' and 'Volver'. Below the header, there is a section 'Datos Actividad:' with the following information:

ACTIVIDAD: SEMINARIO  
LUGAR: Auditorio  
FECHA: 2013-12-19  
HORA: 15:00

Below this is a table for marking attendance:

N°	REPRESENTANTE	PRESENTE
1	Familia rojas torrez	<input type="checkbox"/>
2	Familia perez colque	<input type="checkbox"/>
3	Familia torrico Orozco	<input type="checkbox"/>
4	Familia palma montes	<input type="checkbox"/>
5	Familia ronaldo cr	<input type="checkbox"/>
6	Familia ingles ddd	<input type="checkbox"/>
7	Familia a d	<input type="checkbox"/>
8	Familia carg es	<input type="checkbox"/>
9	Familia river fole	<input type="checkbox"/>
10	Familia asó sad	<input type="checkbox"/>

At the bottom, there is a warning icon and the text 'Listado general de Familias del proyecto "Gudalquivir"'. The title 'MARCAR ASISTENCIA' is on the left, and 'REGISTRAR' and 'Volver' are on the right.

Figura N° 118 Pantalla: Registrar Asistencia

## Pantalla: Ver Asistencia

**Familias con Asistencia** VOLVER

**Datos Actividad:**

ACTIVIDAD: SEMINARIO  
LUGAR: Auditorio  
FECHA: 2013-12-19  
HORA: 15:00

Nº	FAMILIA
1	Familia rojas torrez
2	Familia perez colque
3	Familia palma montes
4	Familia Torrico Orozco

Listado de familias asistentes

Figura N° 119 Pantalla: Ver Asistencia

### I.1.6.3.4.9 Gestionar Correspondencia

## Pantalla: Listar Correspondencia

**LISTAR CORRESPONDENCIA** Adicionar Nueva correspondencia

BUSCADOR:  Buscar Apellido

Nº	AFILIADO	FECHA DE RECIBO	FECHA DE ENTREGA	REMITE	ESTADO	ELIMINAR
1	Limbania Torrico Lopez	2013-12-10		Ricardo Caruzo Lombardi	<a href="#">Pendiente</a>	
2	Limbania Torrico Lopez	2013-12-15		Ricardo Caruzo Lombardi	<a href="#">Pendiente</a>	
3	Daniel roque carpio	2013-12-11		Nelson Ramos Velez	<a href="#">Pendiente</a>	
4	Daniel roque carpio	2013-12-12		josuelo muller kurt	<a href="#">Pendiente</a>	

prev 1 2 3 4 5 6 7 ... 199 200 next

Listado general de Correspondencias\*

Figura N° 120 Pantalla: Listar Correspondencia

### I.1.6.3.4.10 Gestionar Afiliados

#### Pantalla: Listar Afiliados

The screenshot shows a web interface titled "LISTAR AFILIADOS". At the top, there is a search bar labeled "BUSCADOR:" with a "Buscar Afiliado" button. Below the search bar is a table with the following columns: N°, FOTO, CI, NOMBRE, APELLIDO PATERNO, APELLIDO MATERNO, FECHA NAC, VER, MODIFICAR, ELIMINAR, and ASIGNAR PATROCINADOR. The table contains two rows of data:

N°	FOTO	CI	NOMBRE	APELLIDO PATERNO	APELLIDO MATERNO	FECHA NAC	VER	MODIFICAR	ELIMINAR	ASIGNAR PATROCINADOR
1		1234569	Limbania	Torrico	Lopez	1988-05-10				
2		1793519	Daniel	roque	carpio	2013-12-04				

Below the table is a pagination control with buttons for "prev", "1", "2", "3", "4", "5", "6", "7", "199", "200", and "next". At the bottom, there is a warning icon and the text "Listado general de Afiliados del proyecto 'Gudalquivir'".

Figura N° 121 Pantalla: Listar Afiliados

#### Pantalla: modificar afiliado

The screenshot shows a web interface titled "MODIFICAR AFILIADO". At the top right, there is a "Volver" button. The form contains the following fields:

- Ci: 1234569
- Nombre: Limbania
- Ap: Torrico
- Am: Lopez
- Fcha\_nac: 1988-05-10
- Sexo: MASCULINO (dropdown menu)
- Edad: 25
- Estado: ACTIVO (dropdown menu)
- Parentesco: Padre (dropdown menu)
- Foto Anterior:
- Foto:
- 

At the bottom, there is a warning icon and the text "Formulario Modificar Afiliado".

Figura N° 122 Pantalla: modificar afiliado

## Pantalla: Ver Afiliado



Figura N° 123 Pantalla: Ver Afiliado

## Pantalla: Asignar Patrocinador



Figura N° 124 Pantalla: Asignar Patrocinador

### 1.1.6.3.4.11 Reportes

#### Pantalla: Reportes

CORRESPONDENCIA		REPORTE CORRESPON
FICHA DE FAMILIA	Familia: Seleccione	REPORTE AFILIADOS
AFILIADOS		REPORTE AFILIADOS
FAMILIAS		REPORTE FAAMILIAS

REPORTES

Figura N° 125 Pantalla: Reportes

Reportes:

Proyecto Comunitario 'Guadalquivir'  
Lista Correspondencia

**CORRESPONDENCIA**

---

N°	NOMBRE	FECHA	REMIITE
1	Ricardo Caruzzo Lombardi	2013-12-10	Limbania Torrico Lopez
2	Ricardo Caruzzo Lombardi	2013-12-15	Limbania Torrico Lopez
3	Nelson Ramos Velez	2013-12-11	Daniel roque carpio
4	josuelo muller kurt	2013-12-12	Daniel roque carpio

Proyecto Comunitario 'Guadalquivir'  
Lista General de Afiliados

**LISTA DE AFILIADOS**

---

N°	CI	NOMBRE	AP	AM	FECHA_NAC
1	1234569	Limbania	Torrigo	Lopez	1988-05-10
2	1793519	Daniel	roque	carpio	2013-12-04

Figura N° 126 Reportes

#### **I.1.6.4 Base de datos**

#### **I.1.6.5 Modelado de diagrama de clases**

##### ***I.1.6.5.1 Introducción***

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

##### ***I.1.6.5.2 Propósito***

- Identificar clases de análisis y diseño
- Comprender la estructura del sistema deseado para la organización

##### ***I.1.6.5.3 Alcance***

- Describir las clases y objetos de diseño del sistema en su segunda iteración
- Identificar y definir los objetos del sistema según los objetivos del sistema deseado aprobado por la organización

##### ***I.1.6.5.1 Diagrama de clases***

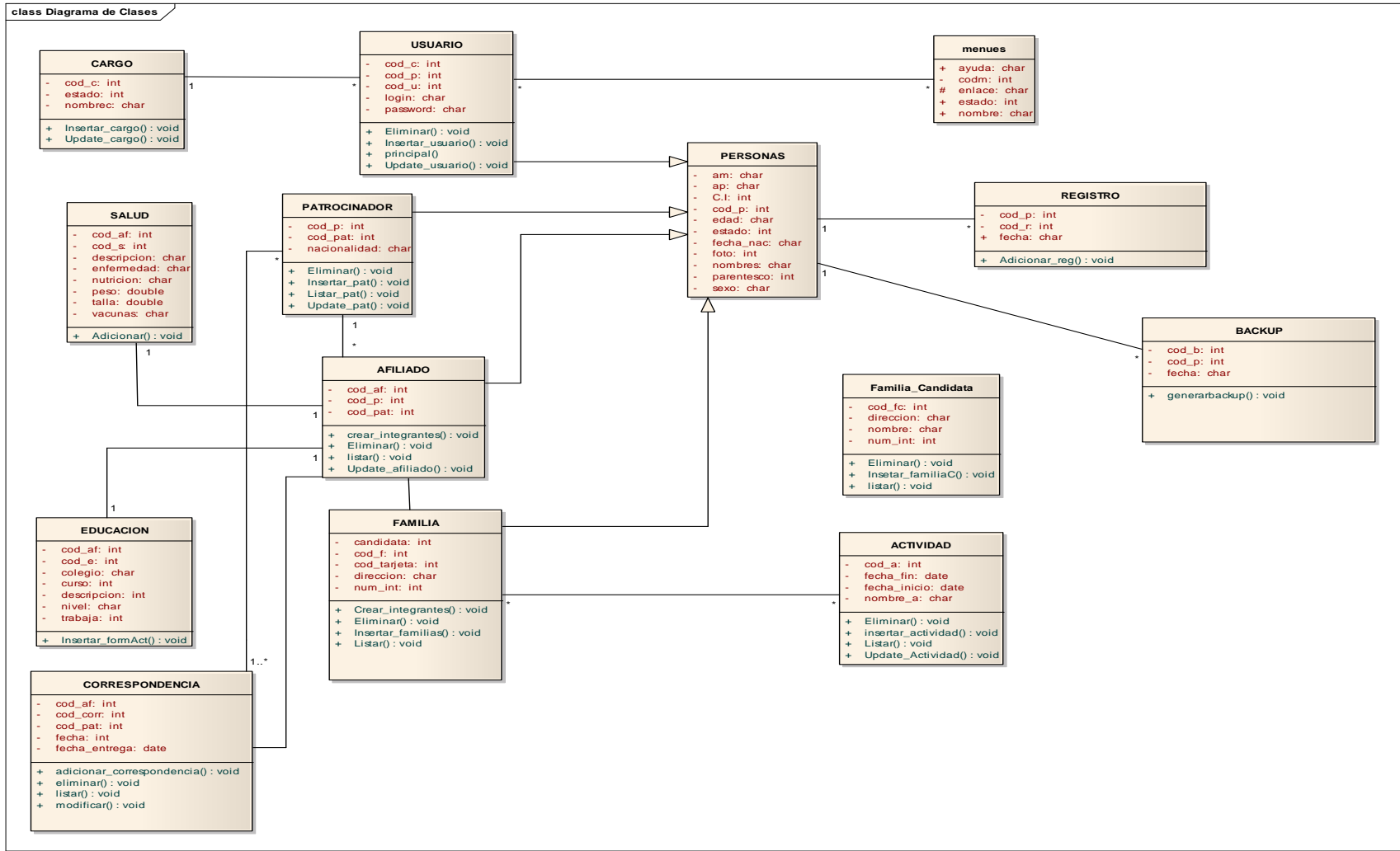


Figura N° 127 Diagrama de Clases



### **I.1.6.6 Creación de la base de datos**

La base de datos de nuestro sistema cuenta con las siguientes tablas:

```
CREATE DATABASE sarci
WITH OWNER = postgres
ENCODING = 'UTF8'
LC_COLLATE = 'Spanish_Bolivia.1252'
LC_CTYPE = 'Spanish_Bolivia.1252'
CONNECTION LIMIT = -1;
```

#### **Persona**

```
CREATE TABLE persona
(
cod_p serial NOT NULL,
ci integer NOT NULL,
nombre character varying(30),
ap character varying(30),
am character varying(30),
fecha_nac date,
edad integer,
sexo character varying(10),
estado integer,
parentesco character varying(30),
foto character varying(100),
```

```
telefono integer,  
correo character varying(100),  
CONSTRAINT persona_pkey PRIMARY KEY (cod_p)  
)
```

### **Usuario**

```
CREATE TABLE usuario  
(  
cod_u serial NOT NULL,  
cod_p integer NOT NULL,  
"login" character varying(15),  
"password" character varying(20),  
cod_c integer,  
estado smallint,  
CONSTRAINT usuario_pkey PRIMARY KEY (cod_u),  
CONSTRAINT codca FOREIGN KEY (cod_c)  
REFERENCES cargo (cod_c) MATCH SIMPLE  
ON UPDATE NO ACTION ON DELETE NO ACTION,  
CONSTRAINT usuario_cod_p_fkey FOREIGN KEY (cod_p)  
REFERENCES persona (cod_p) MATCH SIMPLE  
ON UPDATE NO ACTION ON DELETE NO ACTION  
)
```

### **Actividad**

```
CREATE TABLE actividad
(
    cod_a serial NOT NULL,
    nombre_a character varying(50),
    fecha_inicio date,
    fecha_fin date,
    CONSTRAINT actividad_pkey PRIMARY KEY (cod_a)
)
```

### **Afiliado**

```
CREATE TABLE afiliado
( cod_af serial NOT NULL,
  cod_p integer NOT NULL,
  cod_pat integer,
  CONSTRAINT afiliado_pkey PRIMARY KEY (cod_af),
  CONSTRAINT afiliado_cod_p_fkey FOREIGN KEY (cod_p)
    REFERENCES persona (cod_p) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT afiliado_cod_pat_fkey FOREIGN KEY (cod_pat)
    REFERENCES patrocinador (cod_pat) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

## Backup

```
CREATE TABLE backup
(
    cod_b integer NOT NULL,
    cod_p integer NOT NULL,
    fecha character varying(20),
    CONSTRAINT cod_back PRIMARY KEY (cod_b),
    CONSTRAINT back_persona FOREIGN KEY (cod_p)
        REFERENCES persona (cod_p) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

## Cargo

```
CREATE TABLE cargo
(
    cod_c serial NOT NULL,
    estado integer,
    nombrec character varying(50),
    CONSTRAINT cargo_pkey PRIMARY KEY (cod_c)
)
```

## **Correspondencia**

```
CREATE TABLE correspondencia
(
  cod_corr serial NOT NULL,
  cod_af integer NOT NULL,
  cod_pat integer NOT NULL,
  fecha_recib character varying(30),
  fecha_entrega character varying(30),
  estado smallint,
  CONSTRAINT corresp PRIMARY KEY (cod_corr),
  CONSTRAINT corr_af FOREIGN KEY (cod_af)
    REFERENCES afiliado (cod_af) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT corr_pat FOREIGN KEY (cod_pat)
    REFERENCES patrocinador (cod_pat) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

## **Educacion**

```
REATE TABLE educacion
(
  cod_e serial NOT NULL,
  cod_af integer NOT NULL,
```

```
colegio character varying(20),
nivel character varying(10),
curso character varying(20),
trabaja character varying(10),
descripcion character varying(200),
CONSTRAINT edu_pkey PRIMARY KEY (cod_e),
CONSTRAINT edu_cod_af_fkey FOREIGN KEY (cod_af)
    REFERENCES afiliado (cod_af) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

### **Fam\_act**

```
CREATE TABLE fam_act(
cod_f integer NOT NULL,
cod_a integer NOT NULL,
CONSTRAINT actfam_pkey PRIMARY KEY (cod_f, cod_a),
CONSTRAINT actfamilia_cod_p_fkey FOREIGN KEY (cod_a)
    REFERENCES actividad (cod_a) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT familiacod_f_fkey FOREIGN KEY (cod_f)
    REFERENCES familia (cod_f) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

## **Familia**

```
CREATE TABLE familia
(
  cod_f serial NOT NULL,
  cod_p integer NOT NULL,
  num_int integer,
  candidata integer,
  cod_tarjeta serial NOT NULL,
  direccion character varying(100),
  CONSTRAINT familia_pkey PRIMARY KEY (cod_f),
  CONSTRAINT familia_cod_p_fkey FOREIGN KEY (cod_p)
    REFERENCES persona (cod_p) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

## **Familia candidata**

```
CREATE TABLE familia_candidata
(
  cod_fc serial NOT NULL,
  nombre character varying(100),
  direccion character varying(100),
  num_int integer,
  CONSTRAINT familiac_pkey PRIMARY KEY (cod_fc)
)
```

## **Patrocinador**

```
CREATE TABLE patrocinador
(
  cod_pat serial NOT NULL,
  cod_p integer NOT NULL,
  nacionalidad character varying(20),
  CONSTRAINT patrocinador_pkey PRIMARY KEY (cod_pat),
  CONSTRAINT patrocinador_cod_p_fkey FOREIGN KEY (cod_p)
    REFERENCES persona (cod_p) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

## **Salud**

```
REATE TABLE salud
(
  cod_s serial NOT NULL,
  cod_af integer NOT NULL,
  peso double precision,
  talla double precision,
  nutricion character varying(30),
  descripcion character varying(400),
  enfermedad character varying(50),
  vacunas character varying(50),
  CONSTRAINT salud_pkey PRIMARY KEY (cod_s),
```

```
CONSTRAINT salud_cod_af_fkey FOREIGN KEY (cod_af)
REFERENCES afiliado (cod_af) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

### **Per\_Familia**

```
CREATE TABLE perfamilia
(
cod_p integer NOT NULL,
cod_f serial NOT NULL,
CONSTRAINT perfam_pkey PRIMARY KEY (cod_p),
CONSTRAINT fam_cod_f_fkey FOREIGN KEY (cod_f)
REFERENCES familia (cod_f) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT perfamilia_cod_p_fkey FOREIGN KEY (cod_p)
REFERENCES persona (cod_p) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

### **Usumenu**

```
REATE TABLE usumenu
(
cod_u integer NOT NULL,
codm integer NOT NULL,
estado smallint,
```

```
CONSTRAINT pk_usumenu PRIMARY KEY (cod_u, codm),
CONSTRAINT usumenu_cod_u_fkey FOREIGN KEY (cod_u)
REFERENCES usuario (cod_u) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT usumenu_codm_fkey FOREIGN KEY (codm)
REFERENCES menus (codm) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

### **Cargo**

```
CREATE TABLE cargo
(
cod_c serial NOT NULL,
estado integer,
nombrec character varying(50),
CONSTRAINT cargo_pkey PRIMARY KEY (cod_c)
)
```

### **Menus**

```
CREATE TABLE menus
(
codm serial NOT NULL,
nombre character varying(30) NOT NULL,
ayuda character varying(200),
```

```
estado smallint NOT NULL,  
enlace character varying(50),  
CONSTRAINT pk_menus PRIMARY KEY (codm)  
)
```

```
CREATE TABLE correspondencia
```

```
(  
cod_corr serial NOT NULL,  
cod_af integer NOT NULL,  
cod_pat integer NOT NULL,  
fecha_recib character varying(30),  
fecha_entrega character varying(30),  
estado character varying(1) DEFAULT 1,  
CONSTRAINT corresp PRIMARY KEY (cod_corr),  
CONSTRAINT corr_af FOREIGN KEY (cod_af)  
REFERENCES afiliado (cod_af) MATCH SIMPLE  
ON UPDATE NO ACTION ON DELETE NO ACTION,  
CONSTRAINT corr_pat FOREIGN KEY (cod_pat)  
REFERENCES patrocinador (cod_pat) MATCH SIMPLE  
ON UPDATE NO ACTION ON DELETE NO ACTION  
)
```

```
CREATE TABLE menu
(
  codm integer NOT NULL,
  codp integer NOT NULL,
  CONSTRAINT pk_mepro PRIMARY KEY (codm, codp),
  CONSTRAINT mepro_codm_fkey FOREIGN KEY (codm)
    REFERENCES menues (codm) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT mepro_codp_fkey FOREIGN KEY (codp)
    REFERENCES procesos (codp) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

### I.1.6.7 Tablas: descripción de atributos

#### Tabla Persona

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_p	integer	Si	Si	Código de persona
ci	integer	Si	No	Cedula de identidad
nombre	character varying(30)	No	No	Nombre de la persona
ap	character varying(30)	No	No	Apellido paterno
am	character varying(30)	No	No	Apellido materno
fecha_nac	date	No	No	Fecha de nacimiento
edad	integer	No	No	Edad
sexo	character varying(10)	No	No	Sexo
estado	integer	No	No	Indicador de registro (activo o inactivo)
parentesco	character varying(50)	No	No	Parentesco familiar

Tabla: 73 Descripción de tabla: Persona

### Tabla Usuario

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_u	integer	Si	Si	Código usuario
cod_p	integer	Si	No	Código de persona
login	character varying(28)	No	No	Login del usuario
password	character varying(28)	No	No	Clave del usuario
cod_c	integer	No	No	Código de cargo
estado	smallint	No	No	Indicador de registro (activo o inactivo)

Tabla: 74 Descripción de tabla: Usuario

### Tabla Familia Candidata

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_fc	integer	Si	Si	Código de familia candidata
nombre	character varying(30)	No	No	Nombre general de la familia
direccion	character varying(30)	No	No	Dirección de casa

num_int	integer	No	No	Número de integrantes
---------	---------	----	----	-----------------------

Tabla: 75 Descripción de tabla: familia candidata

Tabla Actividad

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_a	integer	Si	Si	Código de actividad
nombre_a	character varying(50)	No	No	Nombre de la actividad
fecha_inicio	date	No	No	Fecha inicio de actividad
fecha_fin	date	No	No	Fecha fin de actividad

Tabla: 76 Descripción de tabla: Actividad

**Tabla Patrocinador**

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_pat	integer	Si	Si	Código de patrocinador
cod_p	integer	Si	No	Código de persona
nacionalidad	character varying(20)	No	No	Nacionalidad

Tabla: 77 Descripción de tabla: Patrocinador

### Tabla Afiliado

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_af	integer	Si	Si	Código de afiliado
cod_p	integer	Si	No	Código de persona
cod_pat	integer	No	No	Código de patrocinador

Tabla: 78 Descripción de tabla: Afiliado

### Tabla Salud

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_s	integer	Si	Si	Código de salud
cod_af	integer	Si	No	Código de afiliado
peso	double precision	No	No	Peso del afiliado
talla	double precision	No	No	Talla del afiliado
nutricion	character varying(30)	No	No	Estado de nutrición
Descripción	character varying(400)	No	No	Descripción
enfermedad	character varying(50)	No	No	Padecimiento de enfermedad

vacunas	character varying(50)	No	No	Vacunas
---------	--------------------------	----	----	---------

Tabla: 79 Descripción de tabla: Salud

### Tabla Educación

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_e	integer	Si	Si	Código de educación
cod_af	integer	Si	No	Código de afiliado
colegio	character varying(20)	No	No	Nombre de colegio
nivel	character varying(10)	No	No	Nivel de curso
curso	character varying(20)	No	No	Grado que cursa
trabaja	character varying(10)	No	No	Si trabaja o No
Descripción	character varying(200)	No	No	detalles

Tabla: 80 Descripción de tabla: Educación

### Tabla Fam\_act

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_f	integer	Si	Si	Código de familia
cod_a	integer	Si	Si	Código de actividad

Tabla: 81 Descripción de tabla: Fam\_act

### Tabla Cargo

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_c	integer	Si	Si	Código del cargo
estado	integer	No	No	Indicador de registro
nombrec	character varying(30)	No	No	Nombre del cargo

Tabla: 82 Descripción de tabla: Cargo

### Tabla Familia

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_f	integer	Si	Si	Código de familia
num_int	integer	No	No	Número de integrantes
candidata	integer	No	No	Familia candidata
cod_tarjeta	integer	Si	No	Código de tarjeta

direccion	character varying(100)	No	No	Dirección de vivienda
-----------	---------------------------	----	----	-----------------------

Tabla: 83 Descripción de tabla: Familia

### Tabla Perfamilia

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_p	integer	Si	Si	Código de persona
cod_f	integer	Si	No	Código de familia

Tabla: 84 Descripción de tabla: perfamilia

### Tabla Correspondencia

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_corr	integer	Yes	Yes	Código de correspondencia
cod_af	integer	Yes	No	Código afiliado
cod_pat	integer	Yes	No	Código del patrocinador

fecha_recib	character varying(30)	No	No	Fecha de recibo de correspondencia
fecha_entrega	character varying(30)	No	No	Fecha de entrega
estado	smallint	No	No	Indicador Activo / inactivo

Tabla: 85 Descripción de tabla: correspondencia

### Tabla Backup

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_b	integer	Yes	Yes	Código de backup
cod_p	integer	Yes	No	Código persona
fecha	character varying(20)	No	No	Fecha de backup

### Tabla Patrocinador

Nombre	Tipo dato	No Nulo	PK	Descripción
cod_pat	integer	Si	Si	Código de patrocinador
cod_p	integer	Si	No	Código de persona
nacionalidad	character varying(20)	No	No	Nacionalidad

#### ***1.1.6.7.1 Diccionario de datos***

**Persona.-** Está clase se identifica a través de un código único. También esta clase tiene los ci de las personas, los datos nombre, apellido paterno y materno (ap, am), sexo, fecha de nacimiento y edad.

**Usuario.-** La clase tiene un código único y que no se puede repetir, También alberga el login y el password, los cuales permiten el ingreso al sistema

**Cargo.-** La clase tiene un código único, un nombre con el que se identifica.

**Familia.-** la clase se identifica a través de dos códigos únicos: cod\_f, cod\_tarjeta; para poder acceder al número que integra la familia (num\_int), también saber si es una familia que es candidata o no (candidata) y si esta familia esta aun afiliada al proyecto o no (estado).

**Perfamilia.-** la clase tiene dos atributos el código de persona y el código de familia para saber que una persona Y pertenece a una X familia.

**Familia\_Candidata.-** a través de sus atributos podemos registrar a las familias que necesiten del apoyo del la ONG. Estos son: cod\_fc, nombre, dirección, num\_int.

**Actividad.-** la clase tiene un código único cod\_a, y tiene nombre\_a el nombre de la actividad y fecha inicio, fecha fin de la actividad.

**Fam\_act.-** esta clase tiene dos atributos que son los códigos de la clase familia y actividad, con la cual podemos ver qué familia asiste a las actividades extracurriculares.

**Patrocinador.-** la clase tiene un código único que no se puede repetir `cod_pat`, que nos permite identificar a la persona que está patrocinando a un niño afiliado, y está relacionado con la clase Persona.

**Afiliado.-** la clase se identifica a través de un código único `cod_af` y está relacionado con la clase Persona, hereda `cod_p` y `cod_pat` con los cuales se puede acceder al nombre del niño afiliado y el padrino que lo ayuda.

**Salud.-** es una clase donde se registra los datos de salud del afiliado, como ser su peso, talla, nutrición, enfermedades, discapacidades, vacunas.

**Educación.-** al igual que las anteriores la clase posee un código único, `cod_e` y está relacionado con la clase afiliado, tiene atributos como: colegio, nivel, curso, trabaja y una descripción.

## **I.1.6.8 CASOS DE PRUEBA**

### ***I.1.6.8.1 Introducción***

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la misma, y los resultados esperados. Estos casos de prueba son aplicados como prueba llevara asociado un procedimiento de ensayo son las instrucciones para realizar la prueba y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

### ***I.1.6.8.2 Propósito***

- La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- Un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubres un error no detectado hasta entonces.

### ***I.1.6.8.3 Alcance***

- describir los casos de prueba de los formularios principales del sistema.
- Identificar y definir estos casos de prueba para evitar que existan errores y para obtener una salida correcta.
- Para la prueba de los datos en los formularios más importantes del sistema se utilizara el Método de Partición Equivalente.

### ***I.1.6.8.4 Control de la Calidad***

Definimos la calidad del software como el conjunto de cualidades que lo caracterizan y que determina su utilidad y existencia. La calidad se puede medir y depende de los

sistemas o programas; puede medirse al final del producto y se puede controlar también durante todas sus etapas de vida.

El control de calidad es una serie de inspecciones, revisiones y pruebas utilizadas a lo largo del ciclo de desarrollo para asegurar que cada producto cumple con los requisitos y que lean sido asignados. El siguiente grafico permite identificar los factores de la calidad del software a través del modelo de McCall.

### **Revisión del Producto**

- Facilidad de mantenimiento.
- Flexibilidad.
- Facilidad de prueba.

### **Transición del Producto**

- Portabilidad.
- Reusabilidad.
- Interoperabilidad.

### **Operaciones del Producto**

- Corrección.
- Fiabilidad.
- Eficiencia.
- Integridad.
- Facilidad de uso.

#### ***1.1.6.8.5 Prueba de Caja Negra***

Los métodos de prueba de Caja Negra se centran en los requisitos funcionales del software, obtener conjuntos de condiciones de entrada que ejerciten completamente todo los requisitos funcionales de un programa.

La prueba de Caja Negra trata de un enfoque complementario que intenta descubrir diferentes tipos de errores. Los casos de prueba de la Caja Negra pretenden demostrar que:

- Las funciones del Software son operativas.
- La entrada de datos se acepta de forma adecuada.
- Se produce una salida correcta.
- La integridad de la información externa se mantiene.

Se derivan conjunto de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales de un programa.

La prueba de Caja Negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de Interfaces.
- Errores de estructuras de datos o en accesos a base de datos externas.
- Errores de Rendimiento.
- Errores de inicialización y de terminación.

Los casos de prueba deben satisfacer los siguientes criterios:

- Reducir en un coeficiente que es mayor que uno, el se casos de prueba adicionales que se deben diseñar para alcanzar una prueba razonable.
- Que digan algo sobre la presencia o ausencia de clases de errores en lugar de un error asociado solamente con la prueba, en particular, que se encuentra disponible.

### ***I.1.6.8.6 MODELOS DE CASOS DE PRUEBA***

#### ***I.1.6.8.6.1 Formulario ingreso al sistema***

The image shows a login form with the title "INICIAR ..Sesion". It has two input fields: "USUARIO:" containing the text "omar" and "CLAVE:" containing three dots. Below the fields is a red button labeled "INICIAR".

Figura N° 128 *Formulario Ingresar al sistema*

<b>Condiciones de Entrada</b>	<b>Condiciones Validas</b>	<b>Condiciones Invalidas</b>
*Valor de entrada de Usuario	1. Alfanumérico	2. Otro valor
Tamaño de Usuario	3. $4 \leq \text{Valor} \leq 28$	4. Valor $< 4$ 5. Valor $> 28$
*Valor de entrada de Clave	6. Alfanumérico	7. Otro valor
Tamaño de Clave	8. $4 \leq \text{valor} \leq 28$	9. Valor $< 4$ 10. Valor $> 28$

Tabla: 86 Condición de entrada: ingresar al sistema

Usuario	Clave	Camino	Clase
omar	123	1-3-6-8	Valida
Omar-#+	++	1-3-7-9	Invalida

Tabla: 87 Camino de condición de entrada: Ingresar al sistema

#### *1.1.6.8.6.2 Gestionar usuario*

#### **Formulario Adicionar usuario**

Figura N° 129 Formulario Adicionar usuario

Condiciones de Entrada	Condiciones Validas	Condiciones Invalidas
Valor de entrada de ci	1. numérico	2. Otro valor
Tamaño de ci	3. $7 \leq \text{Valor} \leq 7$	4. Valor $< 7$

		5. Valor >7
Valor de entrada de nombre	6. Alfanumérico	7. Otro valor
Tamaño de nombre	8. $4 \leq \text{valor} \leq 30$	9. Valor <4 10. Valor >30
Valor de entrada de ap	11. alfanumérico	12.. otro valor
Tamaño de ap	13. $4 \leq \text{valor} \leq 30$	14. Valor <4 15. Valor >30
Valor de entrada de am	16. alfanumérico	17.. otro valor
Tamaño de am	18. $4 \leq \text{valor} \leq 30$	19. Valor <4 20. Valor >30
Valor de entrada de estado	21. Estado de selección	22. Otro valor
Valor de entrada de login	22. Alfanumérico	23. Otro valor
Tamaño de login	24. $4 \leq \text{Valor} \leq 28$	25. Valor <4 26. Valor >28
Valor de entrada de clave	27. Alfanumérico	28. Otro valor
Tamaño de clave	29. $4 \leq \text{Valor} \leq 28$	30. Valor <4 31. Valor >28
Valor de entrada de cargo	32. estado de selección	33. otro valor

Tabla: 88 Condición de entrada: formulario adicionar usuario

## Formulario modificar usuario

**MODIFICAR USUARIO**

Cod\_u: 62 +

Ci: 7142681

Nombre: omar

Ap: ibañez

Am: ibañez

Estado: ACTIVO

Login: omar

Clave: 123

Cargo: administrador

**Modificar**

Formulario Modificar Usuario

Figura N° 130 Formulario: modificar Usuario

Condiciones de Entrada	Condiciones Validas	Condiciones Invalidas
Valor de entrada de ci	1. numérico	2. Otro valor
Tamaño de ci	3. $7 \leq \text{Valor} \leq 7$	4. Valor < 7 5. Valor > 7
Valor de entrada de nombre	6. Alfanumérico	7. Otro valor
Tamaño de nombre	8. $4 \leq \text{valor} \leq 30$	9. Valor < 4 10. Valor > 30

Valor de entrada de ap	11. alfanumérico	12.. otro valor
Tamaño de ap	13. $4 \leq \text{valor} \leq 30$	14. Valor <4 15. Valor >30
Valor de entrada de am	16. alfanumérico	17.. otro valor
Tamaño de am	18. $4 \leq \text{valor} \leq 30$	19. Valor <4 20. Valor >30
Valor de entrada de estado	21. Estado de selección	22. Otro valor
Valor de entrada de login	22. Alfanumérico	23. Otro valor
Tamaño de login	24. $4 \leq \text{Valor} \leq 28$	25. Valor <4 26. Valor >28
Valor de entrada de clave	27. Alfanumérico	28. Otro valor
Tamaño de clave	29. $4 \leq \text{Valor} \leq 28$	30. Valor <4 31. Valor >28
Valor de entrada de cargo	32. estado de selección	33. otro valor

Tabla: 89 Condición de entrada: formulario Modificar usuario

## Formulario Adicionar Familia

Figura N° 131 Formulario: Adicionar Familia

Condiciones de Entrada	Condiciones Validas	Condiciones Invalidas
Valor de entrada de integrantes	1. Numérico	2. Otro valor
Tamaño de integrantes	3. $0 < \text{valor} < 100$	4. $\text{valor} \Rightarrow 100$ 5. $\text{valor} \leq 0$
Valor de entrada de dirección	5. Alfanumérico	6. Otro valor
Tamaño de dirección	7. $4 \leq \text{valor} \leq 100$	8. $\text{Valor} < 4$ 9. $\text{Valor} > 100$
Valor de entrada de fecha	10. Formato fecha sistema	11. otro valor

Valor de entrada de ci	12. numérico	13. Otro valor
Tamaño de ci	14. $7 \leq \text{Valor} \leq 7$	15. Valor <7 16. Valor >7
Valor de entrada de nombre	17. Alfanumérico	18. Otro valor
Tamaño de nombre	19. $4 \leq \text{valor} \leq 30$	20. Valor <4 21. Valor >30
Valor de entrada de ap	22. alfanumérico	23.. otro valor
Tamaño de ap	24. $4 \leq \text{valor} \leq 30$	25. Valor <4 26. Valor >30
Valor de entrada de am	27. alfanumérico	28.. otro valor
Tamaño de am	29. $4 \leq \text{valor} \leq 30$	30. Valor <4 31. Valor >30
Valor de entrada de fecha de nac	32. Fecha de Calendario de la forma (día-mes-año)	33. otro valor
Valor de entrada de sexo	34. estado de selección	35. otro valor
Valor de entrada de telefono	35. Alfanumérico	36. Otro valor
Tamaño de telefono	37. $5 \leq \text{Valor} \leq 10$	38. Valor <5 39. Valor >10

Valor de entrada de estado	32. estado de selección	33. otro valor
Valor de entrada de parentesco	34. estado de selección	35. otro valor
Valor de entrada de afiliado	36. estado de selección	37. otro valor
Valor de entrada de foto	38. url de ubicación de foto	39. otro valor

Tabla: 90 Condición de entrada: adicionar Familia

### Formulario Adicionar afiliados

The image shows a web form titled "AFILIADOS". It contains the following fields and controls:

- CI:
- Nombre:
- Apellido Paterno:
- Apellido Materno:
- Fecha de Nac:
- Sexo:
- Telefono:
- Price:
- Estado:
- Parentesco:
- Afiliado:
- Foto:
- 

Figura N° 132 Formulario: adicionar afiliados

Condiciones de Entrada	Condiciones Validas	Condiciones Invalidas
------------------------	---------------------	-----------------------

Valor de entrada de integrantes	2. Numérico	2. Otro valor
Tamaño de integrantes	3. $0 < \text{valor} < 100$	4. $\text{valor} \Rightarrow 100$ 5. $\text{valor} \leq 0$
Valor de entrada de dirección	7. Alfanumérico	8. Otro valor
Tamaño de dirección	7. $4 \leq \text{valor} \leq 100$	8. $\text{Valor} < 4$ 9. $\text{Valor} > 100$
Valor de entrada de fecha	10. Formato fecha sistema	11. otro valor
Valor de entrada de ci	12. numérico	13. Otro valor
Tamaño de ci	14. $7 \leq \text{Valor} \leq 7$	15. $\text{Valor} < 7$ 16. $\text{Valor} > 7$
Valor de entrada de nombre	17. Alfanumérico	18. Otro valor
Tamaño de nombre	19. $4 \leq \text{valor} \leq 30$	20. $\text{Valor} < 4$ 21. $\text{Valor} > 30$
Valor de entrada de ap	22. alfanumérico	23.. otro valor
Tamaño de ap	24. $4 \leq \text{valor} \leq 30$	25. $\text{Valor} < 4$ 26. $\text{Valor} > 30$
Valor de entrada de am	27. alfanumérico	28.. otro valor

Tamaño de am	29. $4 \leq \text{valor} \leq 30$	30. Valor <4 31. Valor >30
Valor de entrada de fecha de nac	32. Fecha de Calendario de la forma (día-mes-año)	33. otro valor
Valor de entrada de sexo	34. estado de selección	35. otro valor
Valor de entrada de teléfono	35. Alfanumérico	36. Otro valor
Tamaño de teléfono	37. $5 \leq \text{Valor} \leq 10$	38. Valor <5 39. Valor >10
Valor de entrada de estado	32. estado de selección	33. otro valor
Valor de entrada de parentesco	34. estado de selección	35. otro valor
Valor de entrada de afiliado	36. estado de selección	37. otro valor
Valor de entrada de foto	38. url de ubicación de foto	39. otro valor

Tabla: 91 Condiciones de entrada: Formulario Adicionar Afiliados

### ***1.1.6.8.6.3 Gestionar Patrocinadores***

#### **Formulario Adicionar Patrocinador**

ADICIONAR PATROCINADOR

Nombre

Apellido Paterno:

Apellido Materno:

Nacionalidad:

Correo Electronico:

Telefono:

Adicionar Datos del Patrocinador

Figura N° 133 Formulario: Adicionar Patrocinador

<b>Condiciones de Entrada</b>	<b>Condiciones Validas</b>	<b>Condiciones Invalidas</b>
Valor de entrada de nombre	1. Alfanumérico	2. Otro valor
Tamaño de nombre	3. $4 \leq \text{valor} \leq 30$	4. Valor < 4 5. Valor > 30
Valor de entrada de apellido paterno	6. alfanumérico	7. otro valor
Tamaño de apellido paterno	8. $4 \leq \text{valor} \leq 30$	9. Valor < 4 10. Valor > 30
Valor de entrada de apellido materno	11. alfanumérico	12. otro valor
Tamaño de apellido	13. $4 \leq \text{valor} \leq 30$	14. Valor < 4 15. Valor > 30

materno		
Valor de entrada de nacionalidad	16. Alfanumérico	17. Otro valor
Tamaño de nacionalidad	18. $4 \leq \text{Valor} \leq 20$	19. Valor <4 20. Valor >20
Valor de entrada de correo electrónico	21. Alfanumérico	22. Otro valor
Tamaño de correo electrónico	23. $4 \leq \text{Valor} \leq 50$	24. Valor <4 25. Valor >50
Valor de entrada de teléfono	26. Alfanumérico	27. Otro valor
Tamaño de teléfono	28. $5 \leq \text{Valor} \leq 9$	29. Valor >9 30. Valor < 5

Tabla: 92 Condición de entrada: Adicionar patrocinador

### Formulario Modificar Patrocinador

**MODIFICAR PATROCINADOR**

Codigo: 156 +

Cod\_p: 17

Nombre: Nelson

Apellido Paterno: Ramos

Apellido Materno: Velez

nacionalidad: Argentina

**Modificar**

Modificar los Datos del Patrocinador

Figura N° 134 Formulario: Modificar patrocinador

<b>Condiciones de Entrada</b>	<b>Condiciones Validas</b>	<b>Condiciones Invalidas</b>
Valor de entrada de nombre	1. Alfanumérico	2. Otro valor
Tamaño de nombre	3. $4 \leq \text{valor} \leq 30$	4. Valor <4 5. Valor >30
Valor de entrada de apellido paterno	6. alfanumérico	7. otro valor
Tamaño de apellido paterno	8. $4 \leq \text{valor} \leq 30$	9. Valor <4 10. Valor >30
Valor de entrada de apellido materno	11. alfanumérico	12. otro valor
Tamaño de apellido materno	13. $4 \leq \text{valor} \leq 30$	14. Valor <4 15. Valor >30
Valor de entrada de nacionalidad	16. Alfanumérico	17. Otro valor
Tamaño de nacionalidad	18. $4 \leq \text{Valor} \leq 20$	19. Valor <4 20. Valor >20
Valor de entrada de correo electronico	21. Alfanumérico	22. Otro valor
Tamaño de correo electronico	23. $4 \leq \text{Valor} \leq 50$	24. Valor <4 25. Valor >50
Valor de entrada de teléfono	26. Alfanumérico	27. Otro valor

Tamaño de teléfono	28. $5 \leq \text{Valor} \leq 9$	29. Valor >9 30. Valor < 5
--------------------	----------------------------------	-------------------------------

Tabla: 93 Condición de entrada: Modificar patrocinador

#### ***I.1.6.8.6.4 Gestionar Familia Candidata***

#### **Formulario Adicionar Familia Candidata**

Figura N° 135 Formulario: Adicionar familia Candidata

<b>Condiciones de Entrada</b>	<b>Condiciones Validas</b>	<b>Condiciones Invalidas</b>
Valor de entrada de nombre familiar	1. Alfanumérico	2. Otro valor
Tamaño de nombre	3. $4 \leq \text{valor} \leq 30$	4. Valor <4 5. Valor >30
Valor de entrada de apellido dirección	6. alfanumérico	7. otro valor

Tamaño de apellido dirección	8. $4 \leq \text{valor} \leq 100$	9. Valor < 4 10. Valor > 100
Valor de entrada de fecha	11. Calendario con formato (día-mes - año)	12. otro valor
Valor de entrada de teléfono	13. Alfanumérico	14. Otro valor
Tamaño de teléfono	15. $5 \leq \text{Valor} \leq 9$	16. Valor > 9 17. Valor < 5

Tabla: 94 Condición de entrada: Adicionar familia Candidata

#### ***1.1.6.8.6.5 Gestionar actividades***

#### **Formulario Adicionar Actividades**

The screenshot shows a web form titled "ADICIONAR ACTIVIDAD". At the top right, there are two buttons: "Guardar" and "Volver". The form contains the following fields:

- Nombre:
- Fecha Inicio:  with a calendar icon
- Fecha Fin:  with a calendar icon
- Hora:
- Lugar:
- descripcion:

At the bottom center, there is a "Guardar" button. At the bottom left, there is a warning icon and the text "Adicionar Datos de Nueva Actividad".

Figura N° 136 Formulario: Adicionar actividades

Condiciones de Entrada	Condiciones Validas	Condiciones Invalidas
Valor de entrada de nombre	1. Alfanumérico	2. Otro valor
Valor de entrada de fecha_inicio	3. Calendario con formato (día-mes - año)	4. otro valor
Valor de entrada de fecha_fin	5. Calendario con formato (día-mes - año)	6. Otro valor

Tabla: 95 Condición de entrada: Adicionar Actividad

### Formulario Modificar actividades

**MODIFICAR ACTIVIDAD** Guardar Volver

Nombre:

Fecha Inicio:  ...

Fecha Fin:  ...

Hora:

Lugar:

descripcion:

Modificar Datos

Figura N° 137 Formulario: modificar actividad

Condiciones de Entrada	Condiciones Validas	Condiciones Invalidas
Valor de entrada de nombre	1. Alfanumérico	2. Otro valor
Valor de entrada de fecha_inicio	3. Calendario con formato (año-mes - dia)	4. otro valor
Valor de entrada de fecha_fin	5. Calendario con formato (año-mes - dia)	6. Otro valor

Tabla: 96 Condición de entrada: Modificar Actividad

#### ***1.1.6.8.6 Gestionar Salud***

Formulario Salud

FORMULARIO

**FORMULARIO SALUD**

Nombre:

PESO:

TALLA:

**ESTADO DE NUTRICION**

Baja

Media

Normal

**Padecimiento de discapacidad**

Sin descripciones especiales

**ENFERMEDAD:**

Vacunas

Figura N° 138 Formulario: salud

<b>Condiciones de Entrada</b>	<b>Condiciones Validas</b>	<b>Condiciones Invalidas</b>
Valor de entrada de nombre	1. Alfanumérico	2. Otro valor
Tamaño de nombre	3. $4 \leq \text{valor} \leq 30$	4. Valor < 4 5. Valor > 30
Valor de entrada de peso	6. Numero decimal	7. otro valor
Tamaño de peso	8. $4 \leq \text{valor} \leq 8$	9. Valor < 4 10. Valor > 8
Valor de entrada talla	11. Numero decimal	12. otro valor
Tamaño de talla	13. $4 \leq \text{valor} \leq 8$	14. Valor < 4 15. Valor > 8
Valor de entrada de estado de nutrición	16. Alfanumérico	17. Otro valor
Tamaño de nutrición	18. $5 \leq \text{Valor} \leq 30$	19. Valor > 30 20. Valor < 5
Valor de entrada de enfermedad	21. alfanumérico	22. Otro valor
Tamaño de enfermedad	23. $5 \leq \text{Valor} \leq 50$	24. Valor > 50 25. Valor < 5
Valor de entrada de vacuna	26. Alfanumérico	27. Otro valor
Tamaño de vacuna	28. $5 \leq \text{Valor} \leq 50$	29. Valor > 50 30. Valor < 5

Valor de entrada de descripción	31. alfanumérico	32. Otro Valor
Tamaño de descripción	33. $5 \leq \text{Valor} \leq 500$	34. Valor $> 500$ 35. Valor $< 5$

Tabla: 97 Condición de entrada: Formulario salud

### ***1.1.6.8.7 Modelado de datos***

#### ***1.1.6.8.7.1 Introducción***

A continuación se describen la creación de los controladores usados en el software, apuntando a su conexión con la base de datos. Para expresar este modelo se utiliza la implementación en el lenguaje de programación usado en el desarrollo del proyecto

#### ***1.1.6.8.7.2 Propósito***

Verificar el uso del lenguaje de programación en el proyecto

Mostrar los métodos implementados

Realizar la especificación de métodos

#### ***1.1.6.8.7.3 Alcance***

Descripción de las clases en lenguaje de programación Java

Código fuente perteneciente a las clases más relevantes de cada controlador

#### ***1.1.6.8.7.4 Descripción de código fuente***

## **Conexión**

Clase para realizar la conexión con la base de datos

### **DB.java**

```
public class DB {
    private String User ="postgres";
    private String Password = "7142681";//clave de la base de datos
    private String Host = "127.0.0.1";
    private String Base = "sarci"; //nombre de la base de datos
    private Connection con= null;
    private int port = 5432;
    private ResultSet rs = null;
    private Statement stmt= null;
    private PreparedStatement p = null;
    private SQLException Mensaje = null;
    static private DB conexion= null;
    public DB (){
        try
        {
            Class.forName("org.postgresql.Driver");
        }
        catch (Exception e) {
            System.out.println("no se puede cargar el driver");
        }
    }
    try {
        this.con =
        DriverManager.getConnection("jdbc:postgresql://" +this.getHost()+ "/" +this.getBase(
        )+":"+this.port,this.getUser(),this.getPassword());
    }
    catch (SQLException se) {
        this.setMensaje(se);
    }
}
    public static DB getDb() {
        if (conexion ==null)
            conexion = new DB();
        return conexion;
    }
    public void abm(String Consulta){
        try {
```

```

        this.p = this.con.prepareStatement(Consulta);
        p.executeUpdate();
    }
    catch (SQLException se) {
        this.setMensaje(se);
        //se.printStackTrace(System.err);
    }
}
public ResultSet consulta(String consulta){
    ResultSet valor = null;
    try {

        this.stmt = this.con.createStatement();
        valor= stmt.executeQuery(consulta);
    }
    catch (SQLException se) {
        this.setMensaje(se);
        //se.printStackTrace(System.err);
    }
    return valor;
}

public String generarId (String tabla,String campo) {
    String valor = null;
    try {
        this.stmt = this.con.createStatement();
        this.rs = stmt.executeQuery("select
max(to_number("+campo+", '9999'))+1 as valor from "+tabla);
        this.rs.next();
        valor = rs.getString("valor");
        if (valor==null){
            valor ="1"; }
    }
    catch (SQLException se) {
        this.setMensaje(se);
        //se.printStackTrace(System.err);
    }
    return valor;
}

public boolean buscar(String tabla,String campo,String campob) {
    String valor = null;
    boolean band = false;

```

```

        try {
            this.stmt = this.con.createStatement();
            this.rs = stmt.executeQuery("select max(*) as valor from
"+tabla+" where "+campo+"="+campob);
            this.rs.next();
            valor = rs.getString("valor");
            if (valor==null){
                band = false; }
            else {
                band = true;
            }
        }
        catch (SQLException se) {
            this.setMensaje(se);
            //se.printStackTrace(System.err);
        }
        return band;
    }

    public String getUser(){
        return this.User;
    }
    public void setUser(String u){
        this.User=u;
    }
    public String getPassword(){
        return this.Password;
    }
    public void setPassword(String p){
        this.Password=p;
    }
    public String getHost(){
        return this.Host;
    }
    public void setHost(String h){
    public void setMensaje(SQLException se){
        this.Mensaje=se;
    }
    public void destroy (){
        try {
            this.con.close();
        }

```

```

        catch (SQLException se) {
            //out.println("SQLException" + se.getMessage());
            se.printStackTrace(System.err);
        }
    }
    public PreparedStatement prepareStatement(String Consulta) throws
    SQLException {
        return con.prepareStatement(Consulta);
    }
}

```

### **Principal**

```

public class principal extends VelocityServlet {
    static principal p;
    PreparedStatement pstmt;
    PreparedStatement cons;
    PreparedStatement menu;
    DB db=new DB();
    ResultSet res = null;
    public Template handleRequest(HttpServletRequest
    request,HttpServletResponse response, Context ctx) throws SQLException{
        Template template=null;

        String login    = request.getParameter("usuario");
        String password  = request.getParameter("clave");
        String codigo="" ;
        String cargo="" ;
        String cod_u="" ;
        p = new principal();
        boolean existe = p.usuarioExiste(login, password);
        pstmt= db.prepareStatement("SELECT cod_c FROM usuario where
login='"+login+"' and password='"+password+"'");
        res = pstmt.executeQuery();
        while (res.next()) {
            codigo= res.getString(1);
            //codigo= "1";
        }
        cons= db.prepareStatement("select nombrec from cargo
where("+codigo+"=cod_c)");
        ResultSet res1 = cons.executeQuery();
        while (res1.next()) {
            cargo = res1.getString(1);

```

```

        }

        menu= db.prepareStatement("SELECT cod_u FROM usuario where
login='"+login+"' and password='"+password+"'");
        ResultSet res2 = menu.executeQuery();
        while (res2.next()) {
            cod_u = res2.getString(1);
        }

        if (existe==true) {

            try {
                request.getSession().setAttribute("xnombre", p.usuarioNombre(login,
password));
                request.getSession().setAttribute("mensaje",cargo);
                request.getSession().setAttribute("cod_u",cod_u);
                ctx.put("mensaje", "ADMINISTRADOR");

                template=Velocity.getTemplate("/plantillas/frames/principal.html");

            }
            catch (Exception e) {
                e.printStackTrace();

            }
            // }else{
            /*if(codigo.equals("2")){
                try {
                    request.getSession().setAttribute("xnombre",
p.usuarioNombre(login, password));
                    request.getSession().setAttribute("mensaje","EDUCADOR");
                    ctx.put("mensaje", "EDUCADOR");

                template=Velocity.getTemplate("/plantillas/principal/principal1.html");

            }
            catch (Exception e) {
                e.printStackTrace();

            }
        }

```

```

        }else{
            if(codigo.equals("3")){
                try {
                    request.getSession().setAttribute("xnombre",
p.usuarioNombre(login, password));
                    request.getSession().setAttribute("mensaje","SECRETARIA");
                    ctx.put("mensaje", "SECRETARIA");

template=Velocity.getTemplate("/plantillas/principal/principal2.html");

                }
                catch (Exception e) {
                    e.printStackTrace();

                }
            }
        }
    }
}
}*/

    }
else {
    try {
        ctx.put("mensaje1", "incorrecto");
        ctx.put("mensaje2", "incorrecto");
        template=Velocity.getTemplate("/gindex.html");

    }
    catch (Exception e) {
        e.printStackTrace();

    }

    }
    return template;
}
public boolean usuarioExiste(String login,String password){

    boolean existe=false;
    try {
        pstmt= db.prepareStatement("SELECT cod_u,login,password
FROM usuario where login='"+login+"' and password='"+password+" ");
        res = pstmt.executeQuery();

```

```

        while (res.next()) {
            existe=true;
        }
    } catch (Exception e) {
        // TODO: handle exception
    }
    return existe;
}

public String usuarioNombre(String login,String clave){
    String nom="";
    try {
        pstmt= db.prepareStatement("SELECT P.nombre,P.ap,P.am
FROM usuario U,persona P where u.login='"+login+"' and u.password='"+clave+"'
and p.cod_p=u.cod_p;");
        res = pstmt.executeQuery();
        while (res.next()) {
            nom = res.getString(1) + " " + res.getString(2) + " " +
res.getString(3);//recupera el nombre del q ingreso
        }
    } catch (Exception e) {
        // TODO: handle exception
    }
    return nom;
}
}
}

```

### **Abm usuario**

Clase para listar usuarios

#### **GestUsuarios.java**

```
import conexion.DB;
```

```
public class GestUsuarios extends VelocityServlet{
```



```

        while(res.next()){

                                Map usuario = new HashMap();//creando un
tipo matriz hasmap                                usuario.put("cod_u", res.getString("cod_u"));

                                usuario.put("foto", res.getString("foto"));
                                usuario.put("ci", res.getString("ci"));
                                usuario.put("nombre",
res.getString("nombre"));

                                usuario.put("ap", res.getString("ap"));
                                usuario.put("am", res.getString("am"));
                                /*usuario.put("login", res.getString("login"));
                                usuario.put("password",
res.getString("password"));*/

                                usuario.put("nombrec",
res.getString("nombrec"));

                                lista.add(usuario);

                                context.put("lista", lista);//
                                template =
Velocity.getTemplate("/plantillas/Usuario/ver.html");
                                }
catch(Exception e){System.out.println("!!!!--ERROR-ff-!!!! "+e.getMessage());
System.err.println("Exception cauuuuught: " + e.getMessage());}
        catch (final Exception e) {
                System.err.println("Exception caught: " + e.getMessage());
        }

return template;
}
}

```

### **Abm familia**

#### **Listar.java**

```

import conexion.DB;
public class Listar extends VelocityServlet{
        public Template handleRequest(HttpServletRequest request,HttpServletResponse response, Context context) {
                Template template = null;
                try {
DB base = new DB();

```

```

ResultSet res=base.consulta("select f.cod_f,P.cod_p,P.nombre,P.ap,P.am,
f.cod_tarjeta,f.num_int, f.candidata from persona P, familia f where
f.cod_p=P.cod_p ");
System.out.println("consulta");
String cod_p=request.getParameter("cod_f");
ResultSet res1=base.consulta("delete from familia where cod_f='"+cod_p+"'");
String omar="omar";
ArrayList lista = new ArrayList();
        try{
                while(res.next()){

                                Map usuario = new HashMap();//creando un
tipo matriz hasmap                                usuario.put("cod_f", res.getString("cod_f"));
                                usuario.put("ap", res.getString("nombre")+
"+res.getString("ap")+ " "+res.getString("am"));
                                usuario.put("cod_tarjeta",
res.getString("cod_tarjeta"));
                                usuario.put("num_int",
res.getString("num_int"));
                                usuario.put("candidata",
res.getString("candidata"));
                                usuario.put("cod_p", res.getString("cod_p"));

                                /* if(res.getString("candidata")==="1"){
                                usuario.put("candidata",
res.getString("SI"));
                                }else {usuario.put("candidata",
res.getString("NO"));}*/

                                //usuario.put("numero",res.getString("numero"));
                                lista.add(usuario);
                }
                context.put("omar",omar);//
context.put("lista", lista);//
                template =
Velocity.getTemplate("/plantillas/Familia/ver.html");
                // template =
Velocity.getTemplate("/plantillas/controlactividad/ver.html");

```

```

    }
    catch(Exception e){System.out.println("!!!!--ERROR-- listar
familia!!!! "+e.getMessage());}

    //Aqui finaliza el codigo
    } catch (final Exception e) {
        System.err.println("Exception caught ojo aqui: " +
e.getMessage());
    }
return template;
}
}

```

**Abm actividad**  
**Listar.java**

```

import conexion.DB;

public class Listar extends VelocityServlet{
    public Template handleRequest(HttpServletRequest request,HttpServletResponse response, Context context) {
        Template template = null;
        try {
DB base = new DB();
ResultSet res=base.consulta("select cod_a,nombre_a,fecha_inicio,fecha_fin,estado
from actividad");
System.out.println("!!!!consulta afiliado-!!! ");

ArrayList lista = new ArrayList();

        try{

            while(res.next()){

                Map actividad = new HashMap();//creando un
tipo matriz hasmap

                actividad.put("cod_a", res.getString("cod_a"));

```

```

        actividad.put("nombre_a",
res.getString("nombre_a"));
        actividad.put("fecha_inicio",
res.getString("fecha_inicio"));
        actividad.put("fecha_fin",
res.getString("fecha_fin"));
        actividad.put("estado", res.getString("estado"));

        lista.add(actividad);
    }

    context.put("lista", lista);//
    template =
Velocity.getTemplate("/plantillas/actividad/ver.html");

    }
catch(Exception e){System.out.println("!!!!--ERROR-- en afiliado!!!!
"+e.getMessage());}

    } catch (final Exception e) {
        System.err.println("Exception caught: " +
e.getMessage());
    }

return template;
}
}

```

### **Abm patrocinador**

#### **Listar\_pat.java**

```

import conexion.DB;

public class Listar_pat extends VelocityServlet{
    public Template handleRequest(HttpServletRequest request,HttpServletResponse response, Context context) {
        Template template = null;
        try {
DB base = new DB();

```

```

ResultSet res=base.consulta("select
P.cod_p,P.foto,PA.cod_pat,PA.nacionalidad,P.nombre,P.ap, P.am from persona P,
patrocinador PA where (P.cod_p=PA.cod_p) ");

/*String cod_p=request.getParameter("cod_ue");
ResultSet res1=base.consulta("delete from usuario where cod_u="" +cod_p+""");
response.sendRedirect("ver_Usuario");*/
ArrayList lista = new ArrayList();

        try{

            while(res.next()){

                Map pastrocinador = new HashMap();//creando
un tipo matriz hasmap                pastrocinador.put("cod_pat",
res.getString("cod_pat"));                pastrocinador.put("nacionalidad",
res.getString("nacionalidad"));                pastrocinador.put("nombre",
res.getString("nombre"));                pastrocinador.put("ap", res.getString("ap"));
                pastrocinador.put("am", res.getString("am"));
                pastrocinador.put("cod_p",
res.getString("cod_p"));                pastrocinador.put("foto", res.getString("foto"));

                lista.add(pastrocinador);
            }
//      System.out.println("ERROR "+lista);
context.put("lista", lista);//
template =
Velocity.getTemplate("/plantillas/patrocinador/ver.html");

        }

```

```

catch(Exception e){System.out.println("!!!!--ERROR-- en patrocinador!!!!
"+e.getMessage());}

//Aqui finaliza el codigo
    } catch (final Exception e) {
        System.err.println("Exception caught: " +
e.getMessage());
    }
return template;
}
}

```

## **I.1.6.9 MODELO DE DIAGRAMA DE COMPONENTES**

### ***I.1.6.9.1 Introducción***

Los Diagramas de Componentes ilustran las piezas del software, controladores embebidos, etc. que conformarán un sistema. Un diagrama de Componentes tiene un nivel más alto de abstracción que un diagrama de clase, usualmente un componente se implementa por una o más clases en tiempo de ejecución. Estos son bloques de construcción, como eventualmente un componente puede comprender una gran porción de un sistema.

Los componentes son similares en práctica a los diagramas de paquete como los límites definidos y se usan para agrupar elementos en estructuras lógicas.

### ***I.1.6.9.2 Objetivo***

Se utilizan para modelar la vista estática de un sistema. Muestra la organización y las dependencias entre un conjunto de componentes. No es necesario que un diagrama incluya todos los componentes del sistema, normalmente se realizan por partes. Cada diagrama describe un apartado del sistema.

Uno de los usos principales es que puede servir para ver que componentes pueden compartirse entre sistemas o entre diferentes partes de un sistema.

#### ***1.1.6.9.3 Alcance***

- Definir una abstracción precisa con una interfaz definida.
- Solo hacer parte de la vista física de un sistema.
- Describir la forma en que se comunican los componentes.

**1.1.6.9.3.1 Diagrama de componente general**

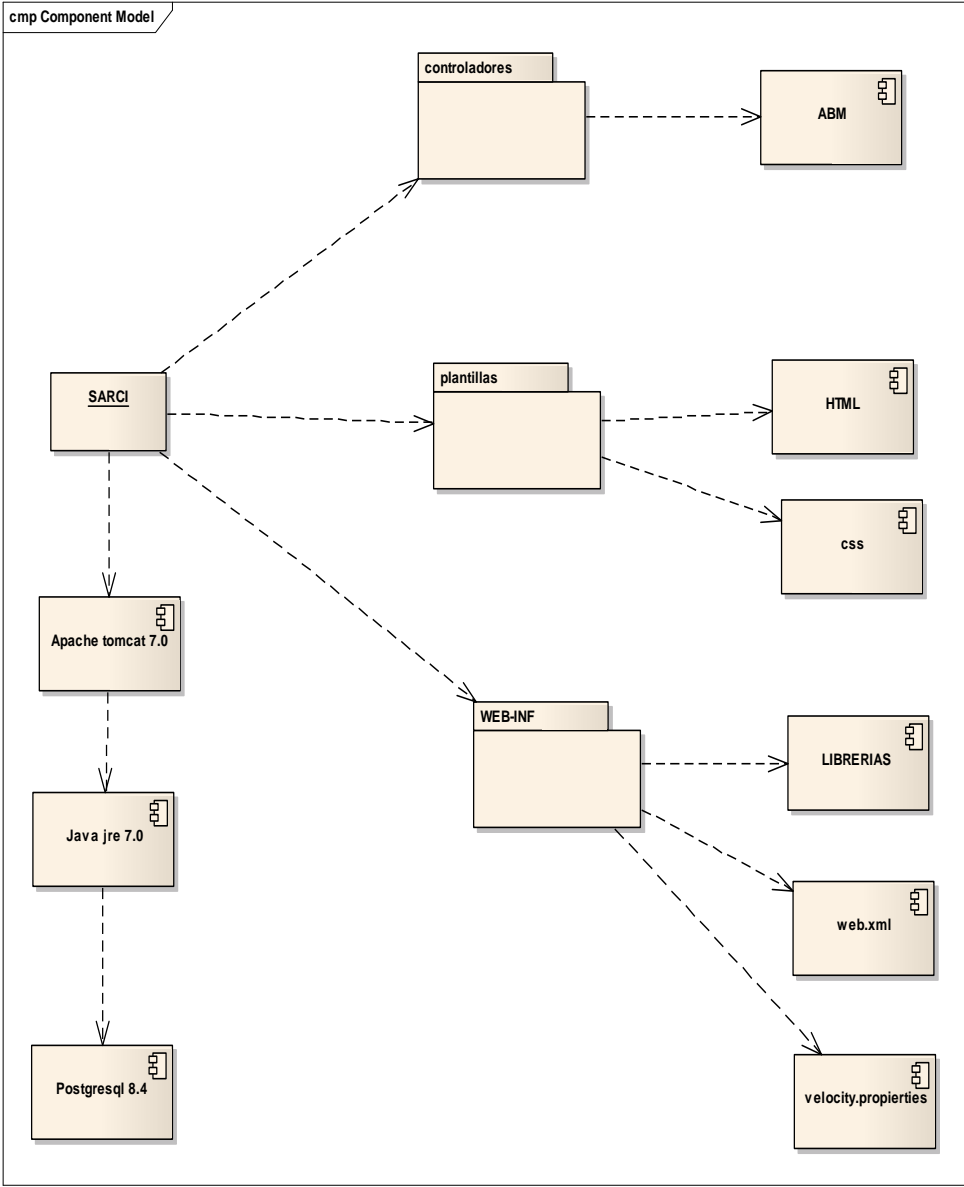


Figura N° 139 Diagrama de componente: General

**1.1.6.9.3.2 Modelado de componentes: Paquete Controladores**

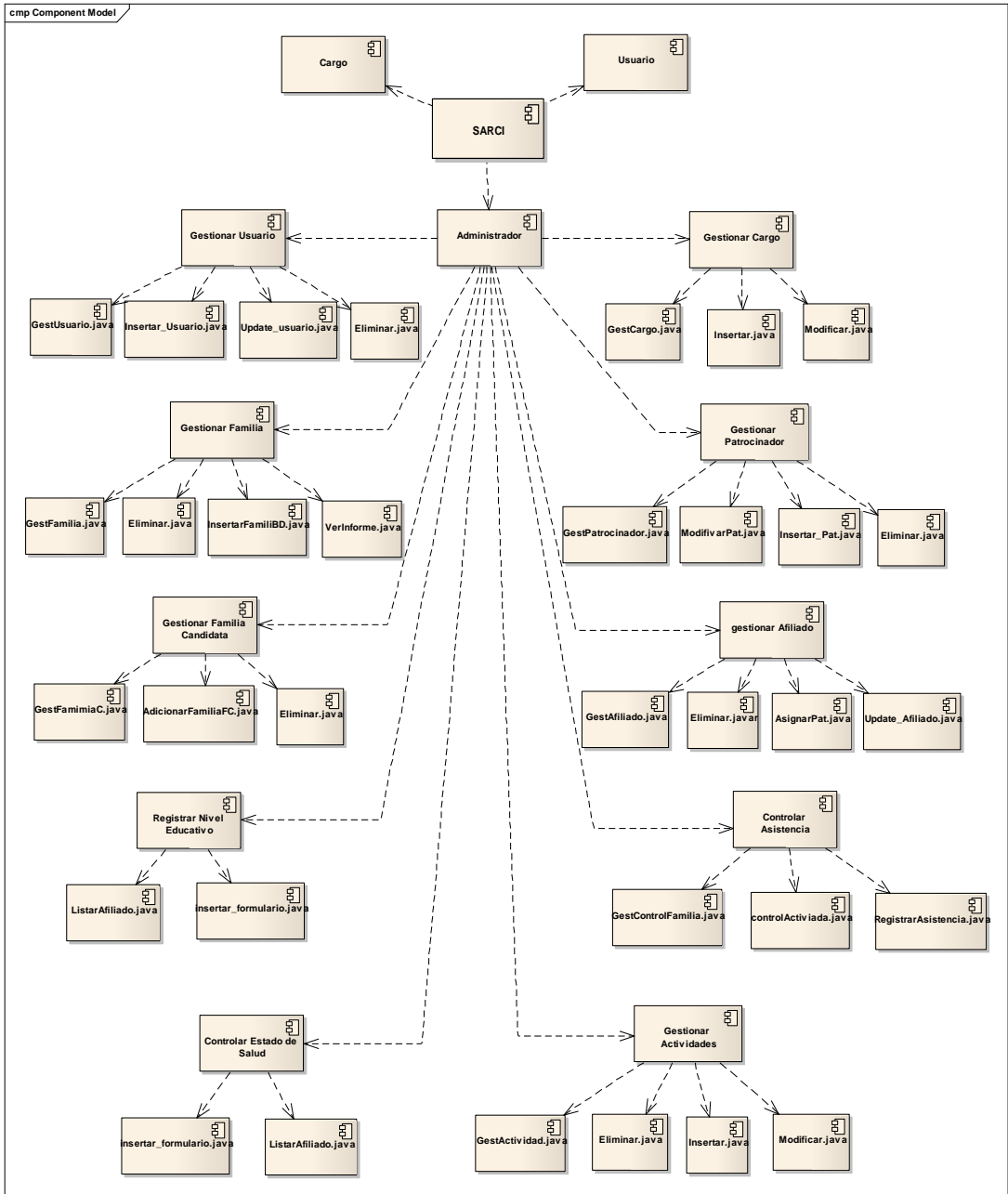


Figura N° 140 Diagrama de componentes: Paquete controladores

## **I.2 CAPACITACION DE USUARIOS DEL SISTEMA**

### **I.2.1 Introducción.**

La capacitación en los distintos niveles constituye un beneficio en las mejores inversiones en cuanto a los recursos humanos y será un beneficio que conduzca a dar mayor facilidad para el usuario, este beneficio consiste en tener automatizado la mayor parte de los registros y programas para crear una mejor organización al tener un sistema que ayudara el trabajo que se desempeña, dándose respuestas positivas y rápidas para cualquier petición.

Para la implementación de un sistema dentro de la organización es imprescindible capacitar a los involucrados con el manejo del sistema. Lo cual lograra que se pueda sacar los máximos beneficios al sistema y se reducirá enormemente los riesgos, errores que puedan surgir al momento de interactuar con el sistema.

### **I.2.2 Objetivo.**

El objetivo principal de la capacitación al personal del Proyecto comunitario “Guadalquivir” que interactuara con el sistema automatizado es lograr que dichos usuarios cuenten con la preparación necesaria y adecuada para el buen manejo y aprovechamiento del sistema, lo cual se traducirá directamente en obtener beneficios del sistema que es el principal objetivo en sí del presente proyecto.

### **I.2.3 Alcances y Limitaciones.**

Se capacito solo al personal involucrado al sistema para evitar que personas ajenas a la administración hagan uso del sistema de manera arbitraria.

### **I.2.4 Especificaciones de Capacitación**

Para la ejecución de la capacitación se elaborara un manual de usuario con el objetivo de que dicho manual sea usado en caso de surgir alguna duda sobre el manejo del sistema. Dicho manual cuenta con imágenes impresas de las pantallas del sistema y los posibles mensajes que podrían salir al usuario.

## **I.2.5 Metodología**

### **Metodología de Enseñanza Socializada**

Con esta metodología se pretende realizar una enseñanza dirigida al jefe y personal de la institución, con lo que se busca una integración social sin descuidar la individualización.

Para llevarlas a cabo se desarrolló el curso usando las siguientes técnicas.

- **Técnica Expositiva**

Consiste en la exposición oral, por parte del capacitador, en la que se estimulara la participación del personal capacitado, este requiere una buena motivación para atraer la atención de los participantes.

- **Técnica de la Experiencia**

La experiencia es un procedimiento eminentemente activo y que procura:

- Exponer de forma presencial los pasos a seguir para realizar alguna acción.
- Explicar el funcionamiento de algo mientras es observado.
- Comprobar, con razones lo que va a suceder, partiendo de experiencias.
- Conferir confianza para actuar en el terreno de la realidad de manera lógica.
- Fortalecer la confianza en sí mismo.
- Formar la mentalidad científica.
- Orientar para solucionar problemas.
- Enriquece el caudal de informaciones, que mejor contribuyen a interpretar la realidad.

Estas técnicas se las realiza para que los participantes puedan interactuar con el sistema web de una manera adecuada.

## **I.2.6 Justificación de la capacitación**

La capacitación se justifica plenamente por la falta de conocimiento y experiencia en el manejo de algún sistema informático, para el proceso de información que se

manipula en la obtención de datos que se necesitan saber, también para dar a conocer los beneficios que nos brindan los sistemas de gestión computarizados.

### **I.2.7 Medios a Utilizar**

Materiales de capacitación:

- Materiales impresos : guías del curso, manuales de usuario
- Materiales audios visuales: equipo de computación, proyector
- Otros materiales: marcadores, cuaderno, hojas, lápiz
- Diapositivas

### **I.2.8 Contenido de la capacitación**

En la capacitación dictada, se expuso el siguiente contenido:

- Introducción a la TIC's
- Resumen del proyecto
- Descripción de objetivos
- Matriz de marco lógico
- Explicación del funcionamiento del sistema
- Interacción con el uso del sistema
- Demostración del uso del sistema
- Dudas y preguntas
- Conclusión

### **I.2.9 Medios de verificación**

- Certificados de participación (Anexos)
- Lista de participación(Anexos)
- Carta de conformidad(Anexos)

Fotos:





### **I.2.10 Conclusiones**

Durante la capacitación se hizo evidente el interés, la entrega y la atención del personal en recibir la capacitación. Tomando en cuenta la importancia del sistema en el desempeño de sus actividades

Otro rasgo que sobresalió durante la capacitación es la información sobre uso de tecnologías de información y comunicación. Dentro de la institución estas tecnologías son de verdadera importancia ya que con ellas se puede aumentar la productividad y el desempeño del personal de la unidad.

## I. Conclusiones y recomendaciones

### I.1 Conclusiones

Luego de haber concluido el presente proyecto denominado “*Mejoramiento del sistema de gestión de la información del proyecto “Guadalquivir”*”, se puede llegar a concluir que el objetivo fue realizado exitosamente, proporcionando a la institución de un sistema de gestión que le permitirá una mejora en el manejo de la información de sus afiliados.

Durante la elaboración del presente proyecto se puede observar que la información con la que trabaja el proyecto comunitario GUADALQUIVIR es de importancia considerable. Tomando este aspecto se concluye con que el desarrollo del presente proyecto será de mucha utilidad en el desempeño futuro del manejo de información.

Para poder realizar el proyecto se tuvo que cumplir con dos componentes: el componente sistema y el componente capacitación, nombrados en términos genéricos. Para cumplir el primer componente denominado “**Mejoramiento del sistema de gestión de la información del proyecto comunitario “Guadalquivir”**”, se tuvo que seguir con una metodología, la cual permitió del desarrollo del sistema con éxito, cumpliendo con estándares de requerimientos, análisis y diseño.

Se debe exponer que el éxito del proyecto se debe al buen uso de las metodologías RUP que nos permite lograr mejoras en el software, gracias a sus procesos de iteración para alcanzar mayores resultados.

El uso de UML nos permite lograr un sistema confiable, pero al mismo tiempo se logró observar que entre más complejo es el sistema que se desea crear más beneficios presenta el uso de UML, las razones de esto son evidentes, sin embargo, existen dos puntos claves : El primero se debe a que mediante un plano de visión global resulta más fácil detectar las dependencias y dificultades implícitas del

sistema, y la segunda razón radica en que los cambios en una etapa inicial (Análisis) resultan más fáciles de realizar que en una etapa final de un sistema como lo sería la fase de desarrollo de software

## **I.2 Recomendaciones**

En caso de que existan cambios del personal involucrado con el manejo del sistema, se recomienda asignar un personal interno (Usuario) para la capacitación del nuevo personal para que cuenten con la preparación adecuada en el uso del sistema.

Para futuros usuarios del sistema se recomienda a la administración exigir conocimientos básicos de computación y otorgar los manuales correspondientes.

Se recomienda llevar a cabo una planificación periódica para salvaguardar la información (Backup), con el motivo de evitar pérdidas de información.

Se recomienda la adquisición de nuevos equipos de computación de última generación, para poder optimizar el rendimiento del sistema.