

1.1 INTRODUCCION

En nuestra sociedad, en la mayoría de las instituciones se manejan diferentes tipos de documentos, los mismos que a la larga deben conservarse o guardarse, de tal manera que cuando se los requiera, estos documentos estén a disposición de las personas. Muchos de ellos, no están adecuadamente organizados, y no cuentan con un sistema de información automatizado. No obstante, el siglo XXI es testigo de una revolución: la importancia y el valor de la información están en constante crecimiento; el avance de la ciencia y la tecnología ha cambiado drásticamente la forma en que ésta pueda obtenerse.

Es evidente que la informática es la ciencia que ha provocado este cambio, a través del desarrollo de sistemas de información computarizado para el procesamiento y el manejo de información, además la informática es una ciencia que tiene su aplicación de alguna u otra forma en todos los campos de la ciencia y tecnología, desde procesos administrativos hasta el control de aparatos más sofisticados. El alcance que tiene sobrepasó todas las dimensiones imaginables que han marcado el pensamiento e historia del hombre.

Hoy en día el acceso rápido y oportuno a la información es una necesidad constante que instituciones como los Centros Médicos necesitan de tal manera que puedan contar con información de manera inmediata y oportuna.

Gracias al avance de la informática que ha producido enormes transformaciones en todos los ámbitos de la vida social, que modifica continuamente en nuestra sociedad numerosas actividades industriales, económicas, políticas, y por supuesto muchas actividades relacionadas con la salud y con la atención a la salud, podemos encontrar una gran gama de herramientas que facilitan poder llegar al objetivo deseado.

La informatización de la historia clínica (HC) se ha convertido durante los últimos años en objetivo de la mayor parte de los trabajos de la Informática Médica.

En el presente trabajo se quiere dar solución al tratamiento de la información que se maneja en el INTRAID, proveyéndola de un sistema automatizado en sus fases de

análisis y diseño, para el manejo de las historias clínicas de los pacientes que son tratados en esta institución.

1.2 ANTECEDENTES

1.2.1 ANTECEDENTES DEL SISTEMA ACTUAL

Después de haber realizado una visita a la institución, se pudo constatar que la misma no cuenta con un sistema especializado para el manejo de las historias clínicas, las mismas que se registran de manera manual, lo cual significa una tarea costosa y morosa, ocasionando diversos problemas con el manejo de las mismas; como ser los siguientes: desorden en el manejo de las historias clínicas debido a la gran cantidad, pérdida total o parcial de las historias clínicas, los tratamientos de los pacientes se ven obstaculizados al no tener los documentos de manera rápida y oportuna, etc.

Las historias clínicas están estructuradas en diez secciones: datos personales, motivo de consulta, enfermedad actual, antecedentes personales, antecedentes laborales, antecedentes físicos, consumo de sustancias, antecedentes sexuales, área familiar, tratamientos realizados, examen físico, examen mental, examen neurológico, diagnósticos, propuesta terapéutica, evolución y resumen, genograma; las mismas que son llenadas por diferentes profesionales, de acuerdo a la patología que presenta el paciente.

1.3 PLANTEAMIENTO DEL PROBLEMA

El volumen elevado de la información y su tratamiento manual, no permiten una gestión adecuada de la información de los pacientes del INTRAID, debido a que existe desorden en el manejo de las historias clínicas; lo que ocasiona que la evolución-seguimiento y tratamiento del paciente se vean obstaculizados al no tener el documento de manera rápida y oportuna.

1.4 OBJETIVO GENERAL Y ESPECIFICOS

1.4.1 OBJETIVO GENERAL

Analizar y diseñar un Sistema de Información Automatizado para optimizar el registro de historias clínicas de los pacientes del INTRAID, a través del uso de la metodología RUP.

1.4.2 OBJETIVOS ESPECIFICOS

- Determinar requerimientos aplicando el Método Isac, para especificar las necesidades de los usuarios; haciendo un análisis de la situación actual.
- Proponer un modelo de Historia Clínica Automatizada, para lograr un mejor manejo de la información.
- Diseñar la Base de Datos, con un excelente soporte de búsquedas y consultas.
- Aplicar la metodología RUP, apoyada en la notación UML para el análisis y diseño del sistema.

1.5 JUSTIFICACION

La afluencia de pacientes al INTRAID tiene una media de 15 a 20 pacientes diarios, en un turno de 12 horas, cada sesión o consulta tiene una duración de una hora según las entrevistas verbales que se realizan.

Según entrevistas realizadas a todo el personal que hace uso de las historias clínicas, se hace notar la necesidad de tener un sistema de información automatizado para el registro de historias clínicas.

La sustitución del tradicional soporte en papel de la información por un sistema de información automatizado responde a varias necesidades, entre ellas:

- Resolver los dos problemas clásicos del manejo de historias clínicas:
- El crecimiento del volumen de información, que llega a crear varios problemas de espacio físico.

- El inevitable riesgo de traspapelación, pérdida parcial o total de las historias clínicas de un paciente y su deterioro.
- Poner a disposición la información de forma fácil y accesible, para brindar una mejor atención al paciente y facilitar el trabajo de los involucrados en el manejo de historias clínicas.

1.5.1. JUSTIFICACION SOCIAL

Al proponer el presente trabajo: Automatización del Registro de Historias Clínicas de los Pacientes del INTRAID, tiene una justificación social, pues el mismo plantea el servicio al público con mayor rapidez y seguridad; además de tratar a pacientes con problemas sociales, que pertenecen a nuestra sociedad.

1.5.2. JUSTIFICACION ACADEMICA

El trabajo también se justifica ya que se aplicará el conocimiento adquirido en los años de estudio dentro de la Carrera de Ingeniería Informática, esto también conduce a realizar una investigación sobre nuevas metodologías lo cual servirá de experiencia en la formación profesional.

1.5.3. JUSTIFICACION TÉCNICA

En la actualidad existe en nuestro medio tecnología que permite resolver problemas de gestión en el manejo de información en los centros de salud; tecnología que es libre, fácil de acceder y usar. Teniendo en cuenta que además la institución cuenta con los equipos necesarios para la implementación de este sistema.

1.6. ALCANCES Y LIMITACIONES

1.6.1 ALCANCES

El sistema abarcará los siguientes módulos:

Registro de Historias Clínicas.- Este módulo almacenará la información relacionada con la historia clínica de cada paciente, dicho historial contendrá información muy

importante como ser: nombres y apellidos, fecha de nacimiento, edad, sexo, escolaridad, antecedentes personales, antecedentes físicos, etc.

Reportes.- Este módulo permitirá acceder a todos los reportes que ofrecerá el sistema, como ser cantidad de pacientes, patologías frecuentes, relación número de varones y mujeres, relación de edades, porcentajes y frecuencias.

1.6.2 LIMITACIONES

El sistema sólo realizará el registro de las historias clínicas de los pacientes que sean atendidos en el INTRAID, teniendo una base de datos con toda la información de los mismos y pudiendo realizar las tareas de altas, bajas y modificaciones.

- No se realizará la elaboración de estadísticas.

2.1 ANTECEDENTES DE LA INSTITUCION

El INTRAID es un Instituto Nacional de Prevención, Tratamiento y Rehabilitación de Drogodependencia y Salud mental.

El INTRAID-Tarija fue creado el 12 de agosto de 1988 con la finalidad de brindar atención en los aspectos psicosociales de la salud y en diciembre de 1990 amplía sus servicios con la creación del Centro de Tratamiento y Rehabilitación en Drogodependencia y Alcoholismo.

Los servicios que brinda, tienden a mejorar la salud de la población, dando cobertura a aquellos programas prioritarios del Plan Estratégico de Salud.

En el Área de Prevención y Promoción de la Salud, se ha elaborado material didáctico entre el que citamos:

- Un Manual de Prevención Integral en la comunidad.
- Trípticos sobre alcohol-alcoholismo, drogas, inhalables, marihuana, autoestima, adolescencia, comunicación y relaciones familiares y tabaco.
- Un manual teórico práctico de prevención del consumo excesivo de alcohol.

2.1.1 Recursos Humanos de la Institución

El INTRAID está compuesto por el siguiente personal:

1 Director

1 Jefe de Prevención y Promoción

1 Médico Psiquiatra

1 Médico General

3 Psicólogos

1 Trabajadora Social

6 Enfermeras Auxiliares

1 Secretaria

1 Cocinera

2 Porteros

2.2 HISTORIA CLINICA¹

En toda institución médica la Historia Clínica es el archivo más importante, pues contiene información vital para la gestión médica, administrativa y legal. Es el elemento clave para el ejercicio profesional del personal sanitario.

Se puede definir como un documento donde se recoge la información que procede de la práctica clínica relativa a un enfermo y donde se resumen todos los procesos a que ha sido sometido.

Antiguamente, cuando apenas había hospitales y el médico atendía individualmente todas las necesidades del paciente, sus historias clínicas eran como un cuaderno de notas donde se registraban los datos más importantes según su criterio.

A pesar del alto valor que tienen los historiales clínicos en todos los aspectos (sanitario, jurídico, docente, investigador, etc.), no existen demasiadas disposiciones legales sobre la obligatoriedad de su existencia, los documentos que la componen y su estructura.

Y de las diferentes maneras que se pueden ordenar los documentos de una historia, de forma homogénea para todo el hospital, generalmente el *cronológico por episodios* es el más utilizado. En él se agrupan los documentos separando la información de los episodios de ingreso (desde la fecha de ingreso en el hospital hasta la fecha de alta), de la documentación de los episodios de consulta.

Lo mejor: cada paciente tiene un solo número de historia.

Si la historia es única para cada paciente en un hospital y su gestión está centralizada desde un archivo único, se garantiza que todos los sucesivos episodios de ese enfermo queden conservados juntos.

Evidentemente, si esas condiciones se cumplen, la recuperación posterior de datos se facilita y es mucho más segura y sencilla que si el mismo enfermo tuviera varias historias de distintos episodios almacenadas en distintos archivos más pequeños, en varios servicios clínicos del hospital, pues estarían incontroladas y descoordinadas.

Cada paciente tiene un número de referencia obligatoriamente único, lo cual permite colgar de él todos los registros asistenciales que se le presten: programaciones quirúrgicas, citaciones en consultas, listas de espera, hospitalizaciones, urgencias, etc. Todos los datos quedan simultáneamente registrados con su número de historia.

Es algo parecido a la cuenta corriente del banco, de la que podemos pedir en cualquier momento un extracto y ver todos los movimientos que ha tenido, y desglosados por conceptos. Aquí ocurre algo similar, teniendo la posibilidad de consultar a la base de datos todas las asistencias practicadas al titular de ese número de historia.

2.2.1 Qué entendemos por historia clínica informatizada

Es aquella en la que la información se captura de forma mecanizada, es decir que no se utiliza papel y bolígrafo para escribir, sino que se registran mediante el teclado de un ordenador.

Una definición más técnica sería la de un conjunto global y estructurado de información, en relación con la asistencia médico-sanitaria de un paciente individual, cuyo soporte permite que sea almacenada, procesada y transmitida mediante sistemas informáticos.

¹ http://www.biocom.com.ar/sistema/historias_clinicas/historia_clinica_informatica.html

2.3 METODOLOGIA DE DESARROLLO

La metodología usada en la realización del presente trabajo es la metodología RUP.

2.3.1 METODO ISAC

El método ISAC es un método orientado al cliente, que nos asegura que se comprendan las necesidades de información, a partir de un análisis de problemas, encuentra soluciones reales para problemas reales, para el desarrollo de este método se requiere de la participación de usuarios y desarrolladores.

ISAC presenta cuatro estados:

1. Análisis de Cambio
2. Estudio de Actividad
3. Análisis de la Información
4. Implementación

Para cada estado se realizan preguntas que analizan los problemas, para luego proponer paquetes de soluciones y escoger el más adecuado.

Las preguntas para cada estado del método ISAC son:

Estado 1: Análisis de Cambio

¿Qué quiere la organización?

¿Qué actividades actualmente ejecuta?

¿Cuáles son los problemas?

¿Quiénes tienen los problemas?

Estado 2: Estudio de Actividades

¿Cuáles actividades deben ser reagrupadas en el sistema de información?

¿Qué son los niveles de actualidad?

¿Qué actividades son manuales, cuáles automatizadas?

¿Qué prioridades?

Estado 3: Análisis de Información

Hacer modelo conceptual del comportamiento externo del sistema de información

Estado 4: Implementación

¿Qué tecnología usar para el sistema de información?

¿Qué procedimientos manuales se realizarán?

Se seleccionó la metodología ISAC para realizar la ingeniería de requerimientos debido a las características que presenta, enfatizando en la posibilidad de dar soluciones reales a problemas reales. Al aplicar esta metodología se encuentra que se adecúa al sistema en estudio.

2.3.2 PROCESO UNIFICADO DE RATIONAL²

La notación UML, es bastante independiente del proceso, lo que significa que se puede utilizar con diferentes procesos de ingeniería de software. El Proceso Unificado de Rational (PUR) es uno de esos enfoques de ciclo de vida que se adapta especialmente bien a UML. El PUR es un proceso *iterativo*, un enfoque iterativo propone una comprensión incremental del problema a través de refinamientos sucesivos y un crecimiento incremental de una solución efectiva a través de varios ciclos.

Como parte del enfoque iterativo se encuentra la flexibilidad para acomodarse a nuevos requisitos o a cambios tácticos en los objetivos del negocio.

Entre las actividades del Proceso Unificado de Rational, destaca la creación de modelos más que documentos sobre papel. Los modelos (especialmente los

representados por UML), proponen representaciones ricas, desde el punto de vista semántico, del sistema software que se está desarrollando.

El desarrollo del Proceso Unificado de Rational, está *centrado en la arquitectura*. El proceso se centra primero en establecer al principio una arquitectura software que guíe el desarrollo del sistema.

Las actividades de desarrollo bajo el Proceso Unificado de Rational están *dirigidas por los casos de uso*. Enfatizan la construcción de sistemas basados en una amplia comprensión de cómo se utilizará el sistema que se entregue. Las nociones de casos de uso y los escenarios, se utilizan para guiar el flujo de procesos desde la captura de los requisitos hasta las pruebas, y para probar caminos que se puedan reproducir durante el desarrollo del sistema.

El Proceso Unificado de Rational soporta técnicas *orientadas a objetos*, cada modelo es orientado a objetos.

El Proceso Unificado de Rational impulsa el *control de calidad*.

El Proceso Unificado de Rational es *configurable*, es adaptable y se puede configurar para cubrir las necesidades que van desde pequeños equipos de desarrollo de software hasta grandes empresas.

La notación UML (no hay que confundir con las metodologías que utiliza dicha notación), se ha convertido desde finales de los 90 en un estándar para modelar con tecnología orientada a objetos, todos aquellos elementos que configuran la arquitectura de un sistema de información y, por extensión de los procesos de negocio de una organización.

De la misma manera que los planos de un arquitecto disponen el esquema director a partir del cual levantamos un edificio, los diagramas UML suministran un modelo de referencia para formalizar los procesos, reglas de negocio, objetos y componentes de

²http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html

una organización. La interacción de todos estos elementos es una representación de nuestra realidad.

Nuestros límites para entender esta realidad están en nuestro lenguaje. El mundo es la suma total de lo que podemos definir, organizar y visualizar. Cabe preguntarse ¿de qué manera un modelo en UML representa nuestra experiencia?. Enseñar a utilizar un lenguaje formal siempre es problemático. Es necesario *mostrar* cómo este lenguaje puede ser aplicado a la realidad tal como la conocemos y tal como la compartimos con los demás.

Fases de desarrollo

Fase de concepción o inicio: durante la fase de concepción se establece la misión del proyecto y el glosario de conceptos, estimación de esfuerzo y cronograma, apoyo en patrones de funcionalidad y análisis, umbral de riesgo, aprobación del anteproyecto, éste es un proceso secuencial no iterativo.

Fase de formalización o elaboración: en esta fase se analiza el dominio del problema, se establece una arquitectura base sólida, se desarrolla un plan de proyecto y elimina los elementos de mayor riesgo para el desarrollo exitoso del proyecto. Las decisiones de arquitectura requieren una visión global del sistema. Los productos de esta fase son: funcionalidad, priorización de los Casos de Uso, Plan Director de Proyecto: Iteraciones, especificación de los Casos de Uso, análisis, diseño, pruebas de certificación, es un proceso iterativo, modelo de casos de uso con descripciones detalladas, otros requerimientos no funcionales o no asociados a casos de uso, descripción de la Arquitectura del Software, un prototipo ejecutable de la arquitectura, y un manual de usuario preliminar.

Fase de construcción: en esta fase se debe poner el diseño en acción, se debe desarrollar el código, actualizar el diseño, mapeo de la base de datos, interface gráfica de usuario: navegación, pruebas de certificación, es un proceso iterativo.

Los productos de esta fase son: el producto de software integrado y corriendo en la plataforma adecuada y manuales de usuario.

Fase de transición: el objetivo es traspasar el software desarrollado a la comunidad de usuarios. Una vez instalados surgirán nuevos elementos que implicarán nuevos desarrollos (ciclos).

Incluye: compilación y despliegue de componentes, pruebas de certificación, actualización del modelo de referencia, actualización del diseño, y otros diagramas, pruebas beta para validar el producto con las expectativas del cliente, ejecución paralela con sistemas antiguos, conversión de datos, entrenamiento de usuarios, distribuir el producto, documentación de usuario, documentación de administrador de sistema, plan de formación, plan de soporte.

Se seleccionó RUP (abreviatura en ingles) o PUR (en español) apoyada en la notación UML porque es un enfoque que se adapta especialmente bien a UML, además es una metodología de desarrollo actual que permite la producción de un software de mayor calidad que satisface las necesidades de los usuarios finales y se adecúa al desarrollo de sistemas de gestión.

2.3.3 UML³



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). 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.

³<http://usuarios.lycos.es/ooper/uml.html>

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo.

Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Rational o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, sólo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos.

2.3.3.1 Diagramas básicos de UML

El lenguaje UML tiene ocho diagramas básicos, que son utilizados para modelar la estructura y el comportamiento de un sistema.

Los diagramas que modelan la estructura del sistema son:

- Diagramas de clases

- Diagramas de componentes

- Diagramas de despliegue

Los diagramas que modelan el comportamiento del sistema son:

- Diagramas de casos de uso

- Diagramas de actividades

- Diagramas de secuencia y colaboración

Diagramas de estado y transición

En la realización del presente trabajo sólo se usará los diagramas que se describen a continuación.

2.3.3.1.1 Diagrama de Clases ⁴

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones. Son el pilar básico del modelado UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer, como para mostrar cómo puede ser constituido.

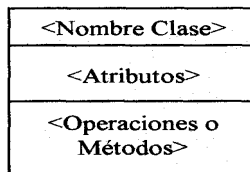
Un diagrama de clases está compuesto por los siguientes elementos:

- Clase
- Relaciones

Clase:

Es la unidad básica que encapsula toda la información de un Objeto (un objeto es una instancia de una clase). A través de ella podemos modelar el entorno en estudio.

En UML, una clase es representada por un rectángulo que posee tres divisiones:



En donde:

Superior: contiene el nombre de la Clase.

Intermedio: contiene los atributos (o variables de instancia) que caracterizan a la Clase (pueden ser private, protected o public).

⁴ <http://www.dcc.uchile.cl/~psalinas/uml/modelo.html#clase>

Inferior: contiene los métodos u operaciones, los cuales son la forma como interactúa el objeto con su entorno (dependiendo de la visibilidad: `private`, `protected` o `public`).

Atributos:

Los atributos o características de una Clase pueden ser de tres tipos, los que definen el grado de comunicación y visibilidad de ellos con el entorno, estos son:

public (+, 

private (-, 

protected (#, 

Métodos:

Los métodos u operaciones de una clase son la forma en cómo ésta interactúa con su entorno, estos pueden tener las características:

public (+, 

private (-, 

protected (#,

Relaciones entre Clases:

Ahora ya definido el concepto de Clase, es necesario explicar cómo se pueden interrelacionar dos o más clases (cada uno con características y objetivos diferentes).

Antes es necesario explicar el concepto de cardinalidad de relaciones: en UML, la cardinalidad de las relaciones indica el grado y nivel de dependencia, se anotan en cada extremo de la relación y éstas pueden ser:

- **uno o muchos:** 1..* (1..n)
- **0 o muchos:** 0..* (0..n)
- **número fijo:** m (m denota el número).

2.3.3.1.2 Diagramas de Casos de Uso⁵

El diagrama de Casos de Uso 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 (operaciones o casos de uso). Un diagrama de Casos de Uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación o sistema y cómo se relaciona con su entorno.

Un diagrama de Casos de Uso consta de los siguientes elementos:

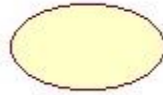
Actor: un Actor es un rol que un usuario juega con respecto al sistema. Es importante destacar el uso de la palabra rol, pues con esto se especifica que un Actor no necesariamente representa a una persona en particular, sino más bien la labor que realiza frente al sistema.



Actor

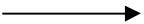
⁵<http://www.dcc.uchile.cl/~psalinas/uml/casosuso.html>

Caso de uso: es una operación/tarea específica que se realiza tras una orden de algún agente externo, sea desde una petición de un actor o bien desde la invocación de otro caso de uso.

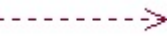


Caso de Uso

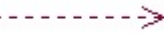
Relaciones:

Asociación 

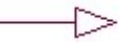
Es el tipo de relación más básica que indica la invocación desde un actor o caso de uso a otra operación (caso de uso). Dicha relación se denota con una flecha simple.

Extensión (<<Extend>>) 

Es la inserción de comportamiento adicional en un Caso de Uso base que no tiene conocimiento sobre él. Dicha relación se denota con una flecha punteada.

Inclusión (<<Include>>) 

Inserción de comportamiento adicional en un Caso de Uso base que describe explícitamente la inserción. Dicha relación se denota por una flecha punteada.

Generalización 

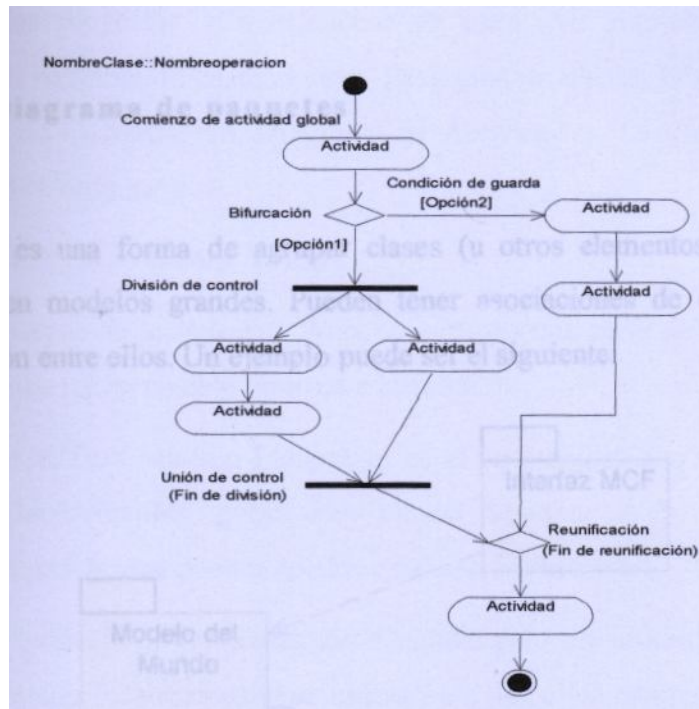
Es una relación entre un caso de uso general y un caso de uso más específico, que hereda y añade propiedades de aquel. Este tipo de relación está orientado exclusivamente para Casos de Uso (y no para actores).

2.3.3.1.3 Diagramas de Actividad

El Diagrama de Actividad es una especialización del Diagrama de Estado, organizado respecto de las acciones y usado para especificar:

- Un método
- Un caso de uso

- Un proceso de negocio: Workflow



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 actividades 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.

Un diagrama de actividades puede dar detalle a un caso de uso, o un mensaje en un objeto. Permiten representar transiciones internas al margen de las transiciones o eventos externos.

2.3.3.1.4 Diagramas de Secuencia⁶

Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes de los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación.

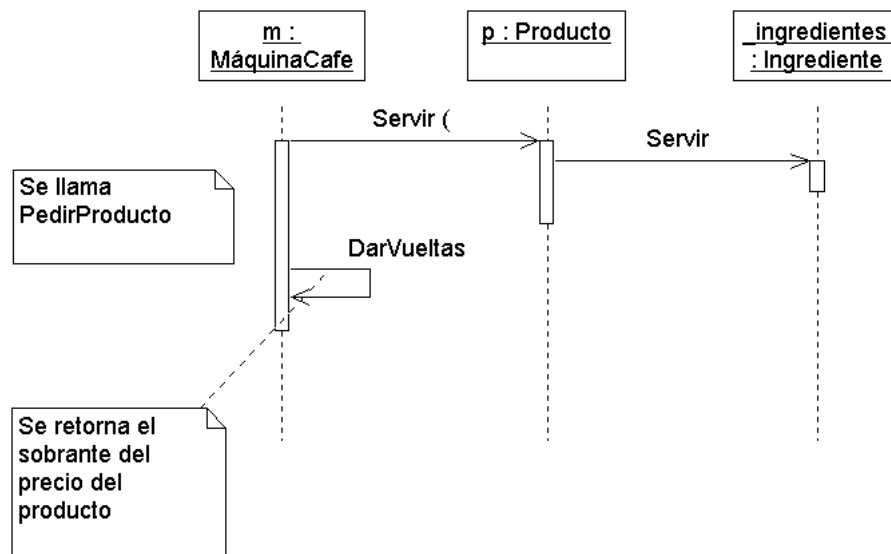


Diagrama de Secuencia

Línea de vida de un objeto

Un objeto se representa como una línea vertical punteada con un rectángulo de encabezado y con rectángulos a través de la línea principal que denotan la ejecución de métodos. El rectángulo de encabezado contiene el nombre del objeto y el de su clase, en un formato nombreObjeto: nombreClase.

⁶<http://www.cs.ualberta.ca/~pfiguero/soo/uml/secuencia01.html>

Activación

Muestra el periodo de tiempo en el cual el objeto se encuentra desarrollando alguna operación, bien sea por sí mismo o por medio de delegación a alguno de sus atributos. Se denota como un rectángulo delgado sobre la línea de vida del objeto.

Mensaje

El envío de mensajes entre objetos se denota mediante una línea sólida dirigida, desde el objeto que emite el mensaje hacia el objeto que lo ejecuta.

2.3.3.1.5 Diagramas de Colaboración ⁷

Un diagrama de colaboración es una forma de representar interacción entre objetos, alterna al diagrama de secuencia. A diferencia de los diagramas de secuencia, pueden mostrar el contexto de la operación (cuáles objetos son atributos, cuáles temporales) y ciclos en la ejecución.

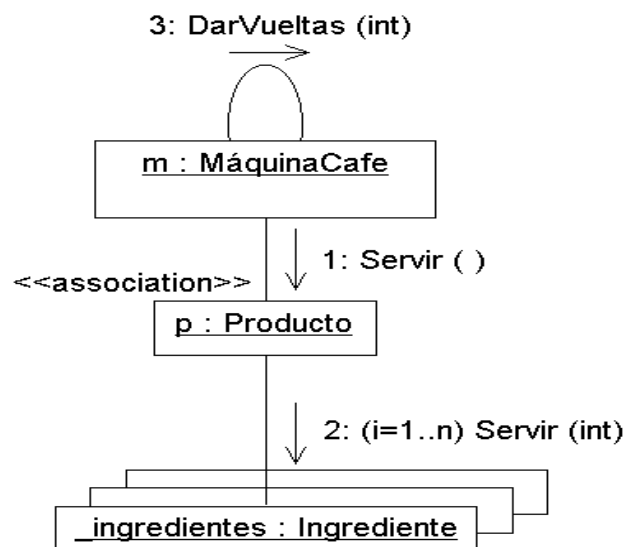


Diagrama de Colaboración

⁷<http://www.cs.ualberta.ca/~pfiguero/soo/uml/colaboracion01.html>

Objeto

Un objeto se representa con un rectángulo, que contiene el nombre y la clase del objeto en un formato *nombreObjeto: nombreClase.* +

Enlaces

Un enlace es una instancia de una asociación en un diagrama de clases. Se representa como una línea continua que une a dos objetos. Está acompañada por un número que indica el orden dentro de la interacción y por un estereotipo que indica que tipo de objeto recibe el mensaje. Pueden darse varios niveles de subíndices para indicar anidamiento de operaciones. Los estereotipos indican si el objeto que recibe el mensaje es un atributo (asociación y se asume por defecto), un parámetro de un mensaje anterior, si es un objeto local o global.

Flujo de mensajes

Expresa el envío de un mensaje. Se representa mediante una flecha dirigida cercana a un enlace.

Marcadores de creación y destrucción de objetos

Puede mostrarse en la gráfica cuáles objetos son creados y destruidos, agregando una restricción con la palabra *new* o *delete*, respectivamente, cercana al rectángulo del objeto.

Objeto compuesto

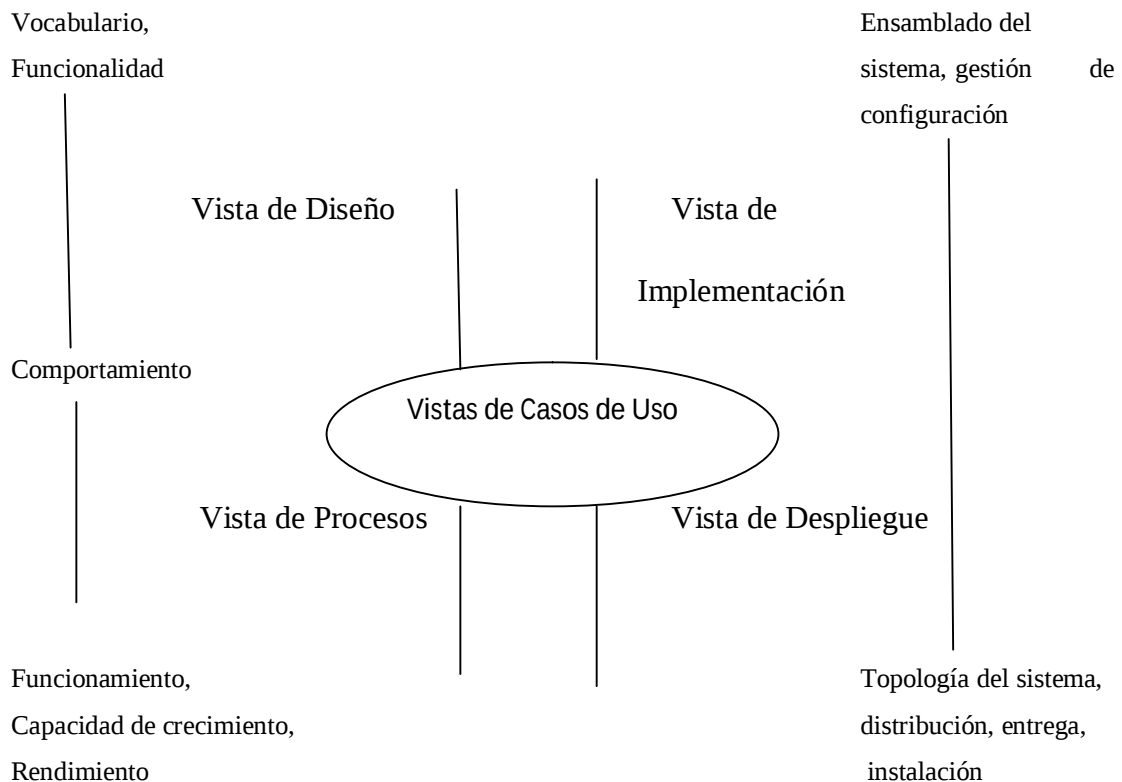
Es una representación alternativa de un objeto y sus atributos. En esta representación se muestran los objetos contenidos dentro del rectángulo que representa al objeto que los contiene. Un ejemplo es el siguiente objeto ventana.

2.3.3.2 Modelo de la Arquitectura de un Software con UML

La arquitectura es un conjunto de decisiones significativas sobre:

- La organización de un sistema software.
- La selección de elementos estructurales y sus interfaces a través de los cuales se construye el sistema.
- Su comportamiento, como se especifica en las colaboraciones entre esos elementos.
- La composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente más grandes.
- El estilo arquitectónico que guía esta organización: los elementos estáticos y sus interfaces, sus colaboraciones y su composición.
- La arquitectura del software no tiene que ver solamente con la estructura y el comportamiento, sino también con el uso, la funcionalidad, el rendimiento, la capacidad de adaptación, la reutilización, la capacidad de ser comprendido, las restricciones económicas y de tecnología y los compromisos entre alternativas así como los aspectos estéticos.

La arquitectura puede describirse a través de cinco vistas:



La vista de casos de uso de un sistema comprende los casos de uso que describen el comportamiento del sistema como es percibido por los usuarios finales, analistas y encargados de prueba. Esta vista no especifica realmente la organización de un sistema software, más bien existe para especificar las fuerzas que configuran la arquitectura del sistema.

La vista de diseño de un sistema comprende las clases, interfaces y colaboraciones que forman el vocabulario del problema y su solución. Esta vista soporta los requisitos funcionales, es decir, los servicios que debería proporcionar a los usuarios finales.

La vista de procesos de un sistema comprende los hilos y procesos que forman los mecanismos de sincronización y concurrencia del sistema. Esta vista cubre principalmente el funcionamiento, capacidad de crecimiento y rendimiento del sistema.

La vista de implementación de un sistema comprende los componentes y archivos que se utilizan para ensamblar y hacer disponible el sistema físico. Esta se preocupa principalmente de la gestión de configuraciones de las distintas versiones de un sistema, a partir de componentes y archivos un tanto independientes y que pueden ensamblarse de varias formas para producir un sistema en ejecución.

La vista de despliegue de un sistema contiene los nodos que forman la topología hardware sobre la que se ejecuta el sistema, esta vista se preocupa principalmente de la distribución, entrega e instalación de las partes que constituyen el sistema físico.

Cada una de estas cinco puede existir por sí misma, de modo que diferentes usuarios puedan centrarse en las cuestiones de la arquitectura del sistema que más les interese.

2.3.4 SISTEMA DE PROCESAMIENTO DE BASE DE DATOS

La tecnología de base de datos se desarrolló para superar las limitaciones de los sistemas de procesamiento de archivos.

Los programas de procesamiento de archivos acceden a los archivos de los datos almacenados. En contraste, los programas de procesamiento de base de datos acuden al DBMS para acceder a los almacenados. Esta diferencia es significativa pues hace más fácil la tarea de programar la aplicación. los programadores no deben preocuparse por la forma en la que los datos se almacenan.

2.3.4.1 DBMS⁸

Un Sistema de Administración de Base de Datos (DBMS) es un programa de computadora para administrar un depósito de datos permanente y autodescriptivo.

Este depósito de datos se denomina Base de Datos y está almacenado en uno o más archivos. Una base de datos es un registro auto descriptivo de registros integrados.

Es autodescriptivo porque además de contener los datos del usuario contiene una descripción su misma estructura.

También se puede decir que una base de datos es un conjunto de registros o archivos que están organizados para un propósito en particular.

2.4 HERRAMIENTAS USADAS PARA EL DISEÑO

2.4.1 ENTERPRISE ARCHITEC

Enterprise Architec es una herramienta de uso muy sencillo, que aborda el diseño y análisis UML y cubre el desarrollo de software desde la captura de requerimientos a lo largo de las etapas de análisis, diseño, pruebas y mantenimiento. EA es una herramienta multi-usuario, Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Además, permite generar documentación e informes flexibles y de alta calidad.

⁸Rumbaugh, M. Blaha, Eddy, Lorensen, Modelado y Diseño Orientado a Objetos Metodología

Velocidad, estabilidad y rendimiento

El Lenguaje Unificado de Modelado ofrece beneficios significativos para ayudar a construir modelos software rigurosos, donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect lo realiza de un modo fácil, rápido y flexible.

Trazabilidad de extremo a extremo

Enterprise Architect proporciona trazabilidad completa desde el análisis de requerimientos y los artefactos de diseño, hasta la implementación y el despliegue. En combinación con la asignación de recursos y tareas que incorpora, los equipos de Gestión de Proyectos y Calidad están dotados con toda la información necesaria para ayudarles a controlar los proyectos y sus entregas.

Construido sobre la base de UML 2.1

Las bases de Enterprise Architect están sustentadas en la especificación de UML 2- pero no se detiene ahí! Usa perfiles UML para extender el dominio de modelado, mientras que la validación del modelo asegura la integridad del proyecto. Combina los procesos de negocio, información y flujos de trabajo en un modelo usando las extensiones gratuitas para BPMN y Erikson –Penker.

EA ayuda a administrar la complejidad permitiendo analizar las dependencias, gestionar modelos muy grandes, control de versiones con proveedores CVS o SCC, Líneas Base por cada instante de tiempo, capacidad de comparar (diff) para seguir los cambios del modelo, interfaz intuitiva y de alto rendimiento con una vista del proyecto en forma de “explorador”.

EA permite la generación de documentos y herramientas de informes con un editor de plantilla WYSIWIG, que puede alcanzar niveles de calidad extraordinarios. Genera informes detallados y complejos de EA con la información que necesita en el formato que su compañía o cliente demanda.

EA soporta generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP. También

hay Add-ins gratis para CORBA y Python disponibles. Con un editor de código fuente con "resaltador de sintaxis" incorporado, EA le permite navegar y explorar su modelo de código fuente en el mismo ambiente. Para aquellos que trabajan en Eclipse o Visual Studio.Net, Sparx Systems también vende puentes livianos para estas IDE's, permitiéndole modelar en EA y saltar directamente al código fuente en su editor preferido. Las plantillas de generación de código le permiten personalizar el código fuente generado a las especificaciones de su compañía.

EA le ayuda a visualizar sus aplicaciones soportando ingeniería inversa de un amplio rango de lenguajes de desarrollo de software y esquemas de repositorios de base de datos. Ingrese frameworks completos desde código fuente o archivos Java .jar - o aún ensambladores binarios .Net! Importando frameworks y librerías de código, Ud. puede maximizar la re-utilización y entendimiento de su inversión existente.

EA soporta transformaciones de arquitectura avanzada dirigida por Modelos (MDA) usando plantillas de transformaciones de desarrollo y fáciles de usar. Con transformaciones incorporadas para DDL, C#, Java, EJB y XSD, Ud. puede rápidamente desarrollar soluciones complejas desde los simples "modelos independientes de plataforma" (MIP) que son el objetivo en "modelos específicos de plataforma" (MEP). Un MIP se puede usar para generar y sincronizar múltiples MIP's - proveyendo un aumento de productividad significativo.

2.4.2 RATIONAL ROSE

Entre las herramientas CASE a emplear para el desarrollo del sistema se propone: **"Rational Rose"**, esta herramienta propone la utilización de cuatro tipos de modelo para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo complejo que representa el dominio del problema y el sistema de software.

Rational Rose utiliza un proceso de desarrollo iterativo controlado (Controlled Iterative Process Development), donde el desarrollo se lleva a cabo en una secuencia

de iteraciones. Cada iteración comienza con una primea aproximación del análisis, diseño e implementación para identificar los riesgos del diseño, los cuales se utilizan para conducir la iteración, primero se identifican los riesgos y después se prueba la aplicación para que estos se hagan mínimos. También permite que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en el espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

Rational Rose puede generar código en distintos lenguajes de programación a partir de un diseño en UML. Otra característica Rational Rose es que proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa, se puede obtener información sobre su diseño.¹

2.4.3 WINDOWS XP

Windows XP (cuyo nombre en clave inicial fue *Whistler*) es una versión de [Microsoft Windows](#), línea de sistemas operativos desarrollado por [Microsoft](#). Lanzado al mercado el [25 de octubre](#) de [2001](#), actualmente es el sistema operativo más utilizado del planeta (con una cuota de mercado del 58,4%) y se considera que existen más de 400 millones de copias funcionando.³ Las letras "XP" provienen de la palabra *eXPeriencia* (*eXPerience* en inglés).

Dispone de versiones para varios entornos informáticos, incluyendo [PCs](#) domésticos o de negocios, [equipos portátiles](#), "[netbooks](#)", "[tablet PC](#)" y "[media center](#)". Sucesor de [Windows 2000](#) junto con Windows ME, y antecesor de [Windows Vista](#), es el primer sistema operativo de Microsoft orientado al consumidor que se construye con un núcleo y arquitectura de [Windows NT](#) disponible en versiones para plataformas de [32](#) y [64 bits](#).

A diferencia de versiones anteriores de Windows, al estar basado en la [arquitectura de Windows NT](#) proveniente del código de [Windows 2000](#), presenta mejoras en la

estabilidad y el rendimiento. Tiene una [interfaz gráfica de usuario](#) (GUI) perceptiblemente reajustada (denominada *Luna*), la cual incluye características rediseñadas, algunas de las cuales se asemejan ligeramente a otras GUI de otros sistemas operativos, cambio promovido para un uso más fácil que en las versiones anteriores. Se introdujeron nuevas capacidades de gestión de software para evitar el "[DLL Hell](#)" (*infierno de las DLLs*) que plagó las viejas versiones. Es también la primera versión de Windows que utiliza la activación del producto para reducir la piratería del software, una restricción que no sentó bien a algunos usuarios. Ha sido también criticado por las vulnerabilidades de seguridad, integración de [Internet Explorer](#), la inclusión del reproductor Windows Media Player y aspectos de su interfaz.

2.5 HERRAMIENTAS ALTERNATIVAS PARA LA IMPLEMENTACION

2.5.1 GESTORES DE BASES DE DATOS

Un Sistema Gestor de Base de Datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de Datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir:

- Definir una base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir la base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular la base de datos: realizar consultas, actualizarla, generar informes.

Así se trata de un software de propósito general. Ejemplo de SGBD son Oracle, SQLServer de Microsoft, MySQL, Firebird, etc.

Algunas de las características deseables de un Sistema Gestor de Base de Datos SGBD son:

- Control de la redundancia: la redundancia de datos tiene varios efectos negativos (duplicar el trabajo al actualizar, desperdicia espacio en disco,

puede provocar inconsistencia de datos) aunque a veces es deseable por cuestiones de rendimiento.

- Restricción de los accesos no autorizados: cada usuario ha de tener unos permisos de acceso y autorización.
- Cumplimiento de las restricciones de integridad: el SGBD ha de ofrecer recursos para definir y garantizar el cumplimiento de las restricciones de integridad.

2.5.1.1 PostgreSQL⁹

PostgreSQL es un Sistema Gestor de Base de Datos Relacionales Orientadas a Objetos. PostgreSQL es la base de datos de código abierto más avanzada hoy en día disponible, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python)

Seguridad

La seguridad de la base de datos está implementada en varios niveles:

- Protección de los ficheros de la base de datos. Todos los ficheros almacenados en la base de datos están protegidos contra escritura por cualquier cuenta que no sea la del superusuario de Postgres.
- Las conexiones de los clientes al servidor de la base de datos están permitidas, por defecto, únicamente mediante sockets Unix locales y no mediante sockets TCP/IP.
- Las conexiones de los clientes se pueden restringir por dirección IP y/o por nombre de usuario mediante el fichero pg_hba.conf situado en PG_DATA.
- Las conexiones de los clientes pueden ser autenticadas mediante otros paquetes externos.

⁹http://www.linux.net.mx/lucas/Postgresql_es/web/navegable/todopostgresql/overview.html

- A cada usuario de Postgres se le asigna un nombre de usuario y (opcionalmente) una contraseña. Por defecto, los usuarios no tienen permiso de escritura a bases de datos que no se hayan creado.
- Los usuarios pueden ser incluidos en *grupos*, y el acceso a las tablas puede restringirse en base a esos grupos.

Autenticación de Usuarios

- *Autenticación* es el proceso mediante el cual el servidor de la base de datos y el postmaster se aseguran de que el usuario que está solicitando acceso a la base de datos es en realidad quien dice ser. Todos los usuarios que quieren utilizar Postgres se comprueban en la tabla `pg_user` para asegurarse que están autorizados a hacerlo.

Estas características colocan a Postgres en la categoría de las Bases de Datos identificadas como *objetos-relacionales*. Nótese que éstas son diferentes de las referidas como *orientadas a objetos*, que en general no son bien aprovechables para soportar lenguajes de Bases de Datos relacionales tradicionales. Postgres tiene algunas características que son propias del mundo de las bases de datos orientadas a objetos.

2.5.1.2 MySQL

MySQL Database Server es la base de datos de código fuente abierto más usada del mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. La exclusiva separación del core server del manejador de tablas, permite funcionar a MySQL bajo control estricto de transacciones o con acceso a disco no transaccional ultrarrápido.

Que es MySQL?

- **MySQL es un sistema de administración de bases de datos.** Una base de datos es una colección estructurada de datos. Ésta puede ser desde una simple lista de compras a una galería de pinturas o el vasto monto de información en una red corporativa. Dado que los computadores son muy buenos manejando grandes cantidades de información, los administradores de bases de datos juegan un papel central en computación, como aplicaciones independientes o como parte de otras aplicaciones.
- **MySQL es un sistema de administración relacional de bases de datos.** Una base de datos relacional archiva datos en tablas separadas en vez de colocar todos los datos en un gran archivo. Esto permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas sobre pedido.
- **MySQL es software de fuente abierta.** Fuente bien abierta significa que es posible para cualquier persona usarlo y modificarlo. Cualquier persona puede bajar el código fuente de MySQL y usarlo sin pagar. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus necesidades. MySQL usa el GPL (GNU General Public License) para definir qué puede hacer y qué no puede hacer con el software en diferentes situaciones. Si usted no se ajusta al GLP o requiere introducir código MySQL en aplicaciones comerciales, usted puede comprar una versión comercial licenciada.
- **Porqué usar MySQL Server?** MySQL Database Server es muy rápido, confiable y fácil de usar. MySQL Server también tiene un práctico set de características desarrollado en cercana cooperación con nuestros usuarios. MySQL Server fue desarrollado inicialmente para manejar grandes bases de datos y ha sido usado exitosamente por muchos años en ambientes de producción de alta demanda. A través de constante desarrollo, MySQL Server ofrece hoy una rica variedad de funciones. Su conectividad, velocidad y

seguridad hacen a MySQL altamente satisfactorio para acceder bases de datos en internet.

2.5.1.3 FIREBID

Es un sistema de administración de base de datos relacional (o RDBMS) (Lenguaje consultas: SQL) de código abierto, basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000. Su código fue reescrito de C a C++. El proyecto se desarrolla activamente, el 18 de abril de 2008 fue liberada la versión 2.1 y el 26 de diciembre de 2009 fue liberada la versión 2.5.0 RC1.

Características

- Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Linux, HP-UX, FreeBSD, Mac OS, Solaris y Microsoft Windows.
- Ejecutable pequeño, con requerimientos de hardware bajos.
- Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded).
- Soporte de transacciones ACID y claves foráneas.
- Es medianamente escalable.
- Buena seguridad basada en usuarios/roles.
- Diferentes arquitecturas, entre ellas el Firebird incrustado (*embedded server*) que permite ejecutar aplicaciones monousuario en ordenadores sin instalar el software Firebird.
- Bases de datos de sólo lectura, para aplicaciones que corran desde dispositivos sin capacidad de escritura, como cd-roms.
- Existencia de controladores ODBC, OLEDB, JDBC, PHP, Perl, .net, etc.
- Requisitos de administración bajos, siendo considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad.
- Pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos.
- Completo lenguaje para la escritura de disparadores y procedimientos almacenados denominado PSQL.
- Capacidad de almacenar elementos BLOB (*Binary Large Objects*).

- Soporte de User-Defined Functions (UDFs).
- Versión autoejecutable, sin instalación, excelente para la creación de catálogos en CD-Rom y para crear versiones de evaluación de algunas aplicaciones.

2.5.2 LENGUAJES DE PROGRAMACION

2.5.2.1 LENGUAJE JAVA

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.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrollados por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.

4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

Para conseguir la ejecución de código remoto y el soporte de red, los programadores de Java a veces recurren a extensiones como CORBA (Common Object Request Broker Architecture), Internet Communications Engine o OSGi respectivamente.

Orientado a Objetos

La primera característica, orientado a objetos (“OO”), se refiere a un método de programación y al diseño del lenguaje. Aunque hay muchas interpretaciones para OO, una primera idea es diseñar el software de forma que los distintos tipos de datos que usen estén unidos a sus operaciones. Así, los datos y el código (funciones o métodos) se combinan en entidades llamadas objetos. Un objeto puede verse como un paquete que contiene el “comportamiento” (el código) y el “estado” (datos). El principio es separar aquello que cambia de las cosas que permanecen inalterables. Frecuentemente, cambiar una estructura de datos implica un cambio en el código que opera sobre los mismos, o viceversa. Esta separación en objetos coherentes e independientes ofrece una base más estable para el diseño de un sistema software. El objetivo es hacer que grandes proyectos sean fáciles de gestionar y manejar, mejorando como consecuencia su calidad y reduciendo el número de proyectos fallidos. Otra de las grandes promesas de la programación orientada a objetos es la creación de entidades más genéricas (objetos) que permitan la reutilización del software entre proyectos, una de las premisas fundamentales de la Ingeniería del Software. Un objeto genérico “cliente”, por ejemplo, debería en teoría tener el mismo conjunto de comportamiento en diferentes proyectos, sobre todo cuando estos coinciden en cierta medida, algo que suele suceder en las grandes organizaciones. En este sentido, los objetos podrían verse como piezas reutilizables que pueden emplearse en múltiples proyectos distintos, posibilitando así a la industria del software a construir proyectos de envergadura empleando componentes ya existentes y de comprobada calidad; conduciendo esto finalmente a una reducción drástica del tiempo de desarrollo. Podemos usar como ejemplo de objeto el aluminio. Una vez

definidos datos (peso, maleabilidad, etc.), y su “comportamiento” (soldar dos piezas, etc.), el objeto “aluminio” puede ser reutilizado en el campo de la construcción, del automóvil, de la aviación, etc.

La reutilización del software ha experimentado resultados dispares, encontrando dos dificultades principales: el diseño de objetos realmente genéricos es pobremente comprendido, y falta una metodología para la amplia comunicación de oportunidades de reutilización. Algunas comunidades de “código abierto” (open source) quieren ayudar en este problema dando medios a los desarrolladores para diseminar la información sobre el uso y versatilidad de objetos reutilizables y bibliotecas de objetos.

Independencia de la plataforma

La segunda característica, la independencia de la plataforma, significa que programas escritos en el lenguaje Java pueden ejecutarse igualmente en cualquier tipo de hardware. Éste es el significado de ser capaz de escribir un programa una vez y que pueda ejecutarse en cualquier dispositivo, tal como reza el axioma de Java, “’write once, run everywhere’”.

Para ello, se compila el código fuente escrito en lenguaje Java, para generar un código conocido como “bytecode” (específicamente Java bytecode) instrucciones máquina simplificadas específicas de la plataforma Java. Esta pieza está “a medio camino” entre el código fuente y el código máquina que entiende el dispositivo destino. El bytecode es ejecutado entonces en la máquina virtual (JVM), un programa escrito en código nativo de la plataforma destino (que es el que entiende su hardware), que interpreta y ejecuta el código. Además, se suministran bibliotecas adicionales para acceder a las características de cada dispositivo (como los gráficos, ejecución mediante hebras o threads, la interfaz de red) de forma unificada. Se debe tener presente que, aunque hay una etapa explícita de compilación, el bytecode generado es interpretado o convertido a instrucciones máquina del código nativo por el compilador JIT (Just In Time).

Hay implementaciones del compilador de Java que convierten el código fuente directamente en código objeto nativo, como GCI. Esto elimina la etapa intermedia donde se genera el bytecode, pero la salida de este tipo de compiladores sólo puede ejecutarse en un tipo de arquitectura.

La licencia sobre Java de Sun insiste que todas las implementaciones sean “compatibles”. Esto dio lugar a una disputa legal entre Microsoft y Sun, cuando éste último alegó que la implementación de Microsoft no daba soporte a las interfaces RMI y JNI además de haber añadido características “dependientes” de su plataforma. Sun demandó a Microsoft y ganó por daños y perjuicios (unos 20 millones de dólares) así como una orden judicial forzando la acatación de la licencia de Sun. Como respuesta, Microsoft no ofrece Java con su versión de sistema operativo, y en recientes versiones de Windows, su navegador Internet Explorer no admite la ejecución de applets sin un conector (o plugin) aparte. Sin embargo, Sun y otras fuentes ofrecen versiones gratuitas para distintas versiones de Windows.

Las primeras implementaciones del lenguaje usaban una máquina virtual interpretada para conseguir la portabilidad. Sin embargo, el resultado eran programas que se ejecutaban comparativamente más lentos que aquellos escritos en C o C++. Esto hizo que Java se ganase una reputación de lento en rendimiento. Las implementaciones recientes de la JVM dan lugar a programas que se ejecutan considerablemente más rápido que las versiones antiguas, empleando diversas técnicas, aunque sigue siendo mucho más lento que otros lenguajes.

La primera de estas técnicas es simplemente compilar directamente en código nativo como hacen los compiladores tradicionales, eliminando la etapa del bytecode. Esto da lugar a un gran rendimiento en la ejecución, pero tapa el camino a la portabilidad. Otra técnica, conocida como compilación JIT (Just In Time, o “compilación al vuelo”), convierte el bytecode a código nativo cuando se ejecuta la aplicación. Otras máquinas virtuales más sofisticadas usan una “recompilación dinámica” en la que la VM es capaz de analizar el comportamiento del programa en ejecución y recompila y optimiza las partes críticas. La recompilación dinámica puede lograr mayor grado

de optimización que la compilación tradicional (o estática), ya que puede basar su trabajo en el conocimiento que de primera mano tiene sobre el entorno de ejecución y el conjunto de clases cargadas en memoria. La compilación JIT y la recompilación dinámica permiten a los programas Java aprovechar la velocidad de ejecución del código nativo sin por ello perder la ventaja de la portabilidad en ambos.

La portabilidad es técnicamente difícil de lograr, y el éxito de Java en ese campo ha sido dispar. Aunque es de hecho posible escribir programas para la plataforma Java que actúen de forma correcta en múltiples plataformas de distinta arquitectura, el gran número de estas con pequeños errores o inconsistencias llevan a que a veces se parodie el eslogan de Sun, "Write once, run anywhere" como "Write once, debug everywhere" (o “Escríbelo una vez, ejecútalo en cualquier parte” por “Escríbelo una vez, depúralo en todas partes”)

El concepto de independencia de la plataforma de Java cuenta, sin embargo, con un gran éxito en las aplicaciones en el entorno del servidor, como los Servicios Web, los Servlets, los Java Beans, así como en sistemas empotrados basados en OSGi, usando entornos Java empotrados.

2.5.2.2 C Sharp

Paradigma: [Orientado a objetos](#)

Apareció en: [2001](#)

Diseñado por: [Microsoft Corporation](#)

Última versión estable: 4.0 ([12 de abril 2010](#))

Tipo de dato: Fuerte, Estático

Implementaciones: [Visual Studio .NET](#), [Mono](#)

Influido por: [Java](#), [C++](#), [Lenguaje C](#),
[Delphi](#), [Eiffel](#)

Ha influido a: [Vala](#), [Nemerle](#), [Boo](#), [F#](#)

Sistema operativo: [Multiplataforma](#)

Web: <http://www.microsoft.com/net>

C# o C# (pronunciado *si sharp* en inglés) es un [lenguaje de programación orientado a objetos](#) desarrollado y estandarizado por [Microsoft](#) como parte de su plataforma [.NET](#), que después fue aprobado como un estándar por la [ECMA](#) e [ISO](#).

Su sintaxis básica deriva de [C/C++](#) y utiliza el modelo de objetos de la plataforma.NET el cual es similar al de [Java](#) aunque incluye mejoras derivadas de otros lenguajes (entre ellos [Delphi](#)).

La creación del nombre del lenguaje, C#, proviene de dibujar dos signos positivos encima de los dos signos positivos de "C++", queriendo dar una imagen de salto evolutivo del mismo modo que ocurrió con el paso de C a C++.

C#, como parte de la plataforma.NET, está normalizado por [ECMA](#) desde diciembre de 2001 ([C# Language Specification](#) "Especificación del lenguaje C#"). El 7 de noviembre de 2005 salió la versión 2.0 del lenguaje que incluía mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. El 19 de noviembre de 2007 salió la versión 3.0 de C# destacando entre las mejoras los tipos implícitos, [tipos anónimos](#) y [LINQ](#) (*Language Integrated Query* -consulta integrada en el lenguaje).

Aunque C# forma parte de la plataforma.NET, ésta es una [interfaz de programación de aplicaciones](#) (API); mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un [compilador](#) implementado que provee el marco de [DotGNU](#) - [Mono](#) que genera programas para distintas plataformas como [Win32](#), [UNIX](#) y [Linux](#).

Clases y objetos

Varios puntos a tener en cuenta en C# con respecto a clases y objetos son los siguientes:

- Una variable de objeto de cierta clase no almacena los valores del objeto sino su referencia (al igual que Java).
- El operador de asignación no copia los valores de un objeto, sino su referencia a él (al igual que Java).
- Un constructor tiene el mismo nombre que su clase y es sintácticamente similar a un método.
- Un constructor no devuelve ningún valor.
- Al igual que los métodos, los constructores también pueden ser sobrecargados.
- Si no se especifica un constructor en una clase, se usa uno por defecto que consiste en asignar a todas las variables el valor de **0**, **null** o **false** según corresponda.
- Para crear un nuevo objeto se utiliza la siguiente sintaxis: `variable = new nombre_clase();`.
- Un destructor se declara como un constructor, aunque va precedido por un signo de tilde ~.
- Se emplea una desasignación de memoria de objetos no referenciados (recolección de basura), y cuando esto ocurre se ejecuta el destructor de dicha clase.
- El destructor de una clase no se llama cuando un objeto sale del ámbito.
- Todos los destructores se llamarán antes de que finalice un programa.
- La palabra clave **this** es un apuntador al mismo objeto en el cual se usa.
- La palabra clave **static** hace que un miembro pertenezca a una clase en vez de pertenecer a objetos de dicha clase. Se puede tener acceso a dicho miembro antes de que se cree cualquier objeto de su clase y sin referencias a un objeto.
- Un método **static** no tiene una referencia **this**.
- Un método **static** puede llamar sólo a otros métodos **static**.

- Un método **static** sólo debe tener acceso directamente a datos **static**.
- Un constructor **static** se usa para inicializar atributos que se aplican a una clase en lugar de aplicarse a una instancia.

C# permite la sobrecarga de operadores con la palabra clave **operator**

Para la determinación de requerimientos se uso el método ISAC como un apoyo, que toma en cuenta las necesidades del cliente; además de usar los diagramas de casos de uso.

3.1 LISTA DE PROBLEMAS

P1: desorden en el manejo de las historias clínicas debido a la gran cantidad.

P2: la búsqueda de la historia clínica del paciente es morosa.

P3: pérdida parcial de las historias clínicas, donde están los datos de los pacientes, debido a la trasapelación de los mismos.

P4: pérdida total de la historia clínica del paciente.

P5: elaboración de una nueva historia clínica debido a su pérdida, lo que ocasiona duplicidad de historias clínicas al encontrar las perdidas.

P6: la evolución- seguimiento y tratamiento del paciente se ve obstaculizada al no tener el documento de manera rápida y oportuna.

P7: no existe un adecuado tratamiento de la información generada en el INTRAID debido a que la administración de la misma es de forma manual.

3.2 LISTA DE GRUPOS DE INTERES

G1: Médico Psiquiatra

G2: Médico General

G3: Psicólogos

G4: Enfermeras

G5: Trabajadora Social

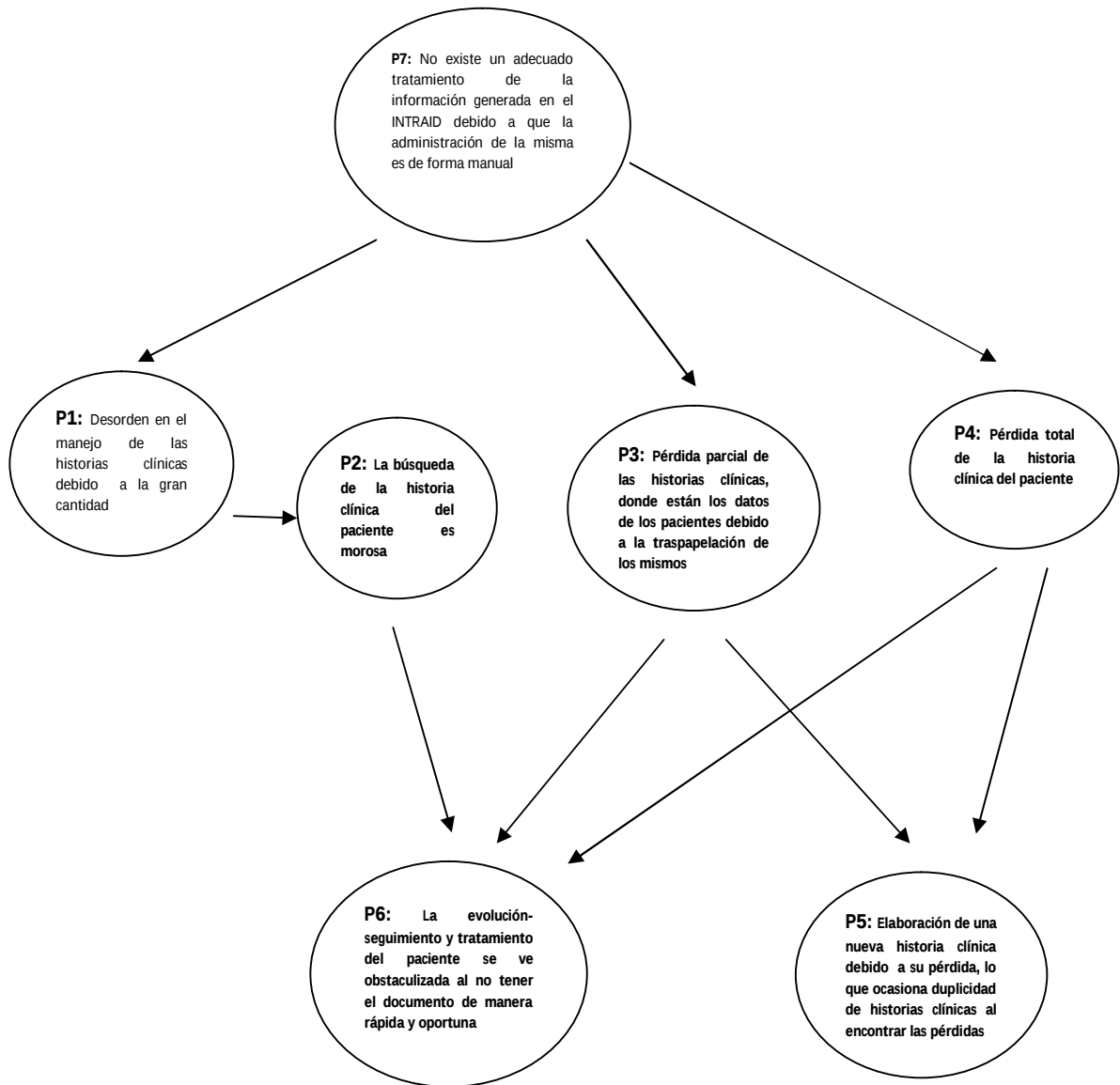
G6: Archivista

G7: Pacientes

3.3 MATRIZ DE PROBLEMAS/ GRUPOS DE INTERES

PRO\GI	G1	G2	G3	G4	G5	G6	G7
P1	X	X	X	X	X	X	
P2	X	X	X	X	X	X	X
P3	X	X	X	X	X	X	X
P4	X	X	X	X	X	X	X
P5	X	X	X	X	X	X	X
P6	X	X	X	X	X	X	X
P7	X	X	X	X	X	X	X

3.4 ANALISIS DE PROBLEMAS



3.5 CUANTIFICACION DE PROBLEMAS

PROBLEMAS	CUANTIFICACION
P1: desorden en el manejo de las historias clínicas debido a la gran cantidad.	No se puede cuantificar
P2: La búsqueda de la historia clínica del paciente es morosa.	Entre 10 a 15 minutos
P3: pérdida parcial de las historias clínicas donde están los datos de los pacientes, debido a la trasapelación de los mismos.	De 4 a 5 veces por mes
P4: pérdida total de la historia clínica del paciente.	De 3 a 4 veces por año
P5: elaboración de una nueva historia clínica debido a su pérdida, lo que ocasiona duplicidad de historias clínicas al encontrar las perdidas.	2 veces por mes
P6: la evolución- seguimiento y tratamiento del paciente se ve obstaculizada al no tener el documento de manera rápida y oportuna.	No se puede cuantificar
P7: no existe un adecuado tratamiento de la información generada en el INTRAID debido a que la administración de la misma es de forma manual.	No es posible cuantificar

3.6 MODELO DE ACTIVIDADES DEL SISTEMA ACTUAL

Para cada paciente que llega a la institución se abre una historia clínica con la siguiente información: datos personales, motivo de consulta, enfermedad actual, antecedentes personales, antecedentes laborales, antecedentes físicos, consumo de sustancias, antecedentes sexuales, área familiar, tratamientos realizados, examen físico, examen mental, examen neurológico, diagnósticos, propuesta terapéutica, evolución y resumen, genograma.

ANALISIS DE METAS

M1: optimizar el manejo de la documentación gracias a la automatización de la misma, lo que facilitaría el acceso a la información.

M2: mantener la información ordenada y clasificada para un manejo eficiente.

M3: reducir el tiempo de búsqueda de la historia clínica.

M4: lograr secuencia y sistematización del tratamiento al contar con la historia clínica oportunamente.

M5: tener una identificación para pacientes sin documentos de identidad y que en un momento debido a la patología que presentan desconocen la misma.

DEFINICION DE NECESIDADES DE CAMBIO

1.- Facilitar el manejo de las historias clínicas, permitiendo el ingreso de datos y una búsqueda fácil y rápida.

Problemas: P1 & P2 & P3 & P4 & P5 & P6

Metas: M1 & M2 & M3 & M4

2.- Tener un mayor control de los tratamientos que siguen los pacientes, al tener la información debidamente ordenada.

Problemas: P6 & P7

Metas: M4

3.- Tener un registro personal con la fotografía del paciente al momento que se está llenando su historia clínica.

Problemas: P5

Metas: M5

GENERACION DE ALTERNATIVAS DE CAMBIO

A1: automatizar el registro de las historias clínicas de los pacientes.

A2: crear y mantener una base de datos con información de historias clínicas y los tratamientos.

A3: se diseñará un sistema amigable al usuario.

A4: se contratará personal encargado solamente del manejo de las historias clínicas.

A5: se crearán planillas con hojas electrónicas para tener las historias clínicas registradas.

MODELO DE ACTIVIDADES A REALIZAR

Paquete 1: A1 & A2 & A3

Paquete 2: A4

Paquete 3: A5

Paquete 1:

Se automatizará el proceso del llenado de la historia clínica, creando la base de datos con la siguiente información: datos personales, motivo de consulta, enfermedad actual, antecedentes personales, antecedentes laborales, antecedentes físicos, consumo de sustancias, antecedentes sexuales, área familiar, tratamientos realizados, examen físico, examen mental, examen neurológico, diagnósticos, propuesta terapéutica, evolución y resumen, genograma.

3.7 CASOS DE USO

El diagrama de casos de uso representa la forma en que un actor opera con el sistema en desarrollo, además de la forma, tipo y orden de los elementos que interactúan.

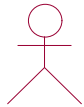
Un Caso de Uso describe un servicio provisto por un sistema, es decir, un modo específico de usarlo. El conjunto completo de Casos de Uso especifica todas las posibles maneras en las que el sistema puede ser usado, sin revelar cómo esto es implementado por el sistema.

3.7.1 Identificación y Breve Descripción de Actores



Usuario sistema

El Usuario del sistema representa a la persona que interactúa con el sistema, el cual está registrado previamente y accede al mismo mediante un Nombre de Usuario y Contraseña.



Enfermera

La enfermera es una especialización del Usuario del sistema, ya que tiene un rol dentro del sistema. Su tarea es la de administrar los pacientes y las citas de los mismos.



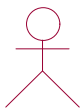
Personal médico

Personal médico, como su nombre lo indica es el médico que puede realizar una atención externa y/o interna, y hacer el respectivo llenado de la historia clínica.



Psicologo

Psicólogo también es una especialización del Usuario del sistema, el tiene la tarea de realizar el informe psicológico de cada paciente.



Psiquiatra

Psiquiatra es una especialización de personal médico y su función es la de atender una consulta externa y/o interna y registrar la historia clínica.

3.7.2 Descripción del Modelo de Casos de Uso Completo por Actor

Los actores Enfermera, Personal médico, Administrador sistema, Psicólogo, son una especialización del Usuario Sistema; además los actores Médico, Psiquiatra, son una especialización de Personal médico como se muestra en la Figura 1, heredan los procesos que realiza el Usuario del Sistema y cada actor tiene procesos propios de acuerdo al rol que le corresponde.

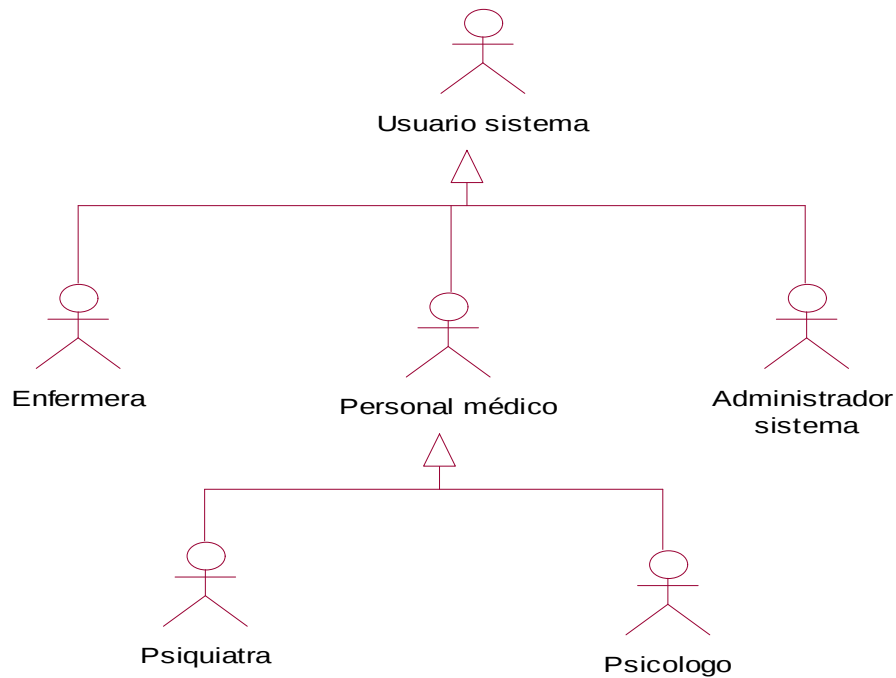


Figura 1. Herencia de actores

3.7.2.1 Casos de uso para el usuario del sistema

El usuario del sistema utiliza el caso de uso ingresar usuario y contraseña, para ingresar al sistema y habilitar sus roles dentro del mismo, como se muestra en la Figura 2.

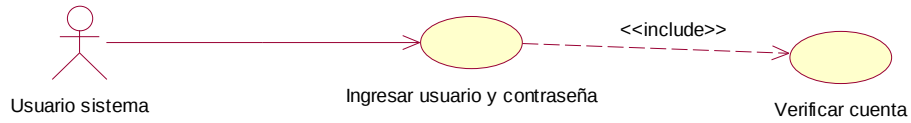


Figura 2. Caso de uso para el usuario del sistema

3.7.2.2 Casos de uso para el actor enfermera

La enfermera realizará la administración de los pacientes, a través de los casos de uso adicionar paciente, modificar paciente y eliminar paciente.

También podrá realizar el registro del comportamiento del paciente y el tratamiento que está realizando el paciente.

Este actor también tiene la tarea de administrar las citas de los pacientes, realizando los siguientes casos de uso: adicionar cita, modificar cita y eliminar cita. (Figura 3)

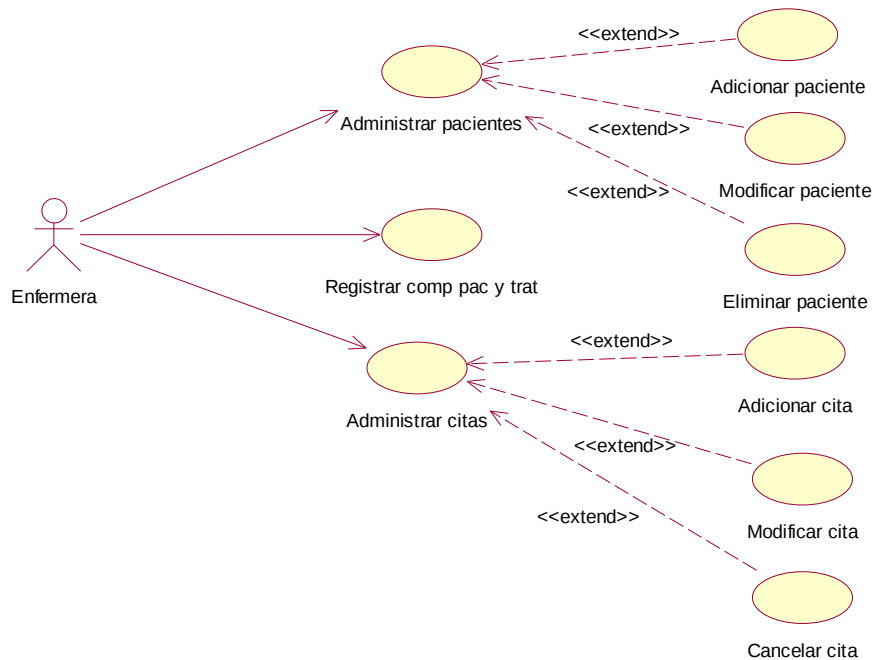


Figura 3. Caso de uso para la enfermera

3.7.2.3 Casos de uso para el personal médico

Este actor puede realizar la consulta a una historia clínica, el respectivo registro de una historia clínica, con sus variaciones registrar historia clínica en consulta externa e interna; cada una con sus pertinentes tareas adicionales. (Figura 4)

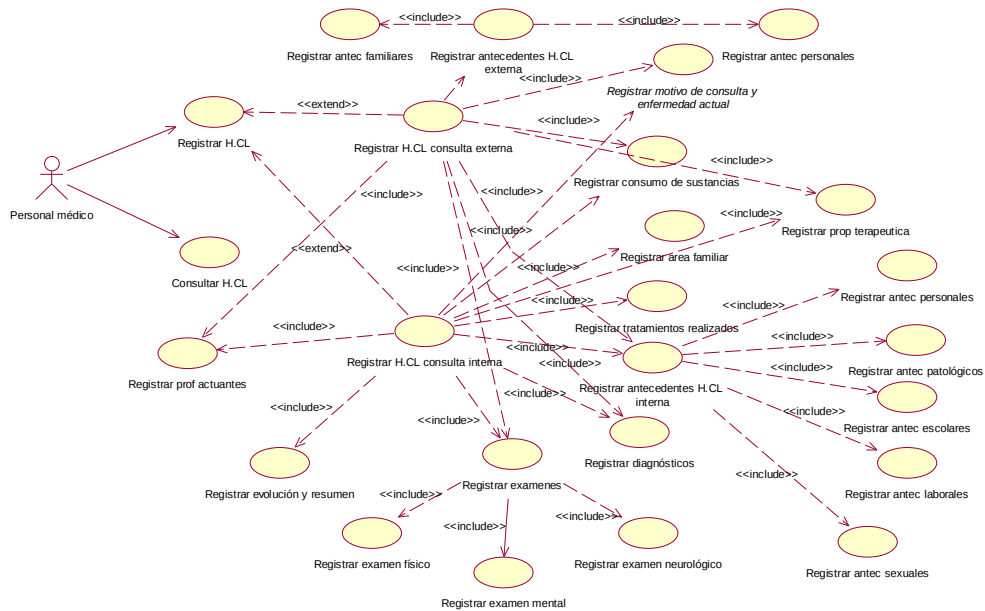


Figura 4. Caso de uso para el personal médico

3.7.2.4 Casos de uso para el psicólogo

En este caso de uso el actor psicólogo realiza el registro del informe psicológico que se le hace al paciente una vez que es atendido en consulta interna, además de realizar los reportes para las diferentes instituciones que lo requieran. (Figura 5)

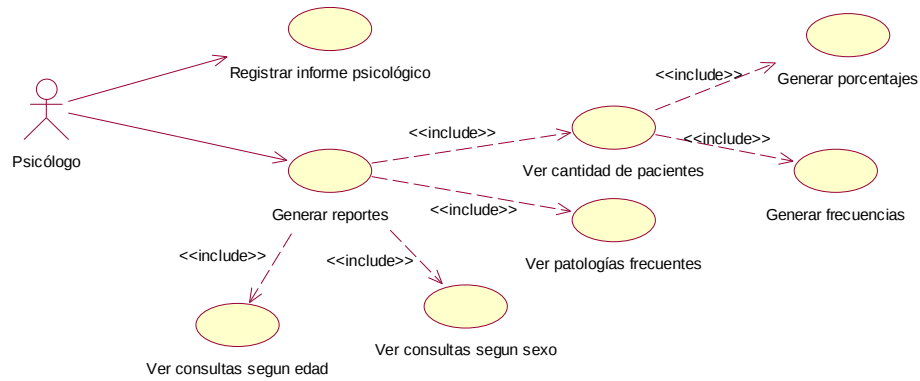


Figura 5. Caso de uso para el psicólogo

3.7.2.5 Casos de uso para el administrador del sistema

El administrador del sistema realizará la administración de usuarios a través de los casos de uso adicionar usuario, modificar usuario y eliminar usuario. También tendrá la tarea de administración de la base de datos efectuando los casos de uso de restaurar la base de datos y realizar copias de seguridad. (Figura 6)

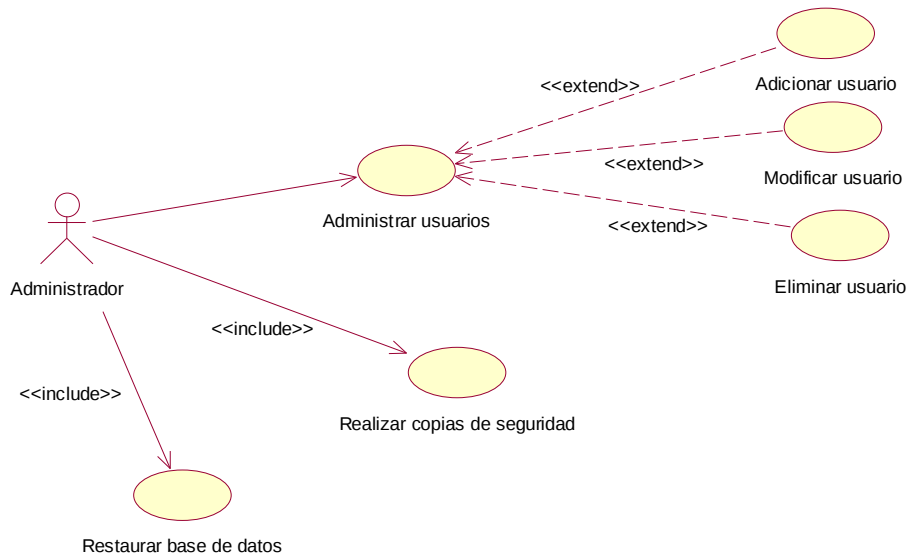


Figura 6. Caso para el administrador del sistema

3.7.3 Descripción de Casos de Uso

Nombre del Caso de Uso:	Ingresar al sistema.	
Descripción:	Con el nombre de usuario y contraseña, se permite el ingreso del mismo al sistema, habilitando sus roles y menús correspondientes.	
Actor(es):	Usuario del sistema.	
Precondiciones:	El sistema requiere el ingreso de nombre de usuario y contraseña al sistema. El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Introduce el nombre de usuario y contraseña	2.Valida los datos. 3.Verifica usuario y contraseña. 4.Muestra la pantalla principal del sistema; una vez ingresado el nombre de usuario y contraseña, muestra la pantalla del menú correspondiente a cada rol.
Flujos alternativos:	2a.Los datos están en blanco incompletos, mostrar mensaje “Datos en blanco o incompletos” Se vuelve al paso 1. 3ª.No encontrado, mostrar mensaje “Usuario no existe o contraseña incorrecta”. Vuelve al paso 1.	
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Administrar pacientes.	
Descripción:	Permite gestionar los pacientes ya registrados.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1. Elige la opción administrar pacientes del menú de la enfermera.	2.Muestra la pantalla de administrar pacientes con las opciones para elegir.
Flujos alternativos:		
Postcondiciones:		

Nombre del Caso de Uso:	Adicionar paciente.	
Descripción:	Permite registrar un nuevo paciente.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción adicionar paciente. 3.La enfermera introduce los datos del nuevo paciente. 4. Elige la opción adicionar.	2.Muestra la pantalla con los campos para ingresar los diferentes datos del paciente. 5.El sistema adiciona los datos
Flujos alternativos:	3a.Introduce incorrectamente los datos del paciente y se muestra un mensaje “los datos están incorrectos”. Se vuelve al paso 3.	
Postcondiciones:		

Nombre del Caso de Uso:	Modificar paciente.	
Descripción:	Permite modificar datos de un paciente ya registrado.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige y/o busca el paciente a modificar. 3.La enfermera introduce los datos modificados. 4.Elige la opción modificar.	2.Muestra la pantalla con los diferentes campos para modificar los datos del paciente. 5.El sistema guarda las modificaciones.
Flujos alternativos:	3a.Introduce incorrectamente los datos del paciente y se muestra un mensaje “ los datos están incorrectos”. Se vuelve al paso 3.	
Postcondiciones:	Los datos de el paciente se actualizan	

Nombre del Caso de Uso:	Eliminar paciente.	
Descripción:	Permite eliminar datos de un paciente ya registrado.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción eliminar paciente. 3.La enfermera elige cual paciente eliminar. 4.Elige el botón eliminar.	2.Se muestra la pantalla con una lista de los pacientes. 5.El sistema elimina de manera lógica los datos del paciente elegido.
Flujos alternativos:		
Postcondiciones:		

Nombre del Caso de Uso:	Registrar comportamiento de paciente y tratamiento.	
Descripción:	Permite que la enfermera realice el registro del comportamiento y el tratamiento que el paciente tiene.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción registrar comportamiento. 4.Busca y/o elige al paciente. 5.Selecciona al paciente. 7. Introduce los datos y elige el botón guardar	2. Muestra la pantalla buscar pacientes. 3. Muestra una lista de pacientes. 6. El sistema muestra una pantalla con los datos que necesita llenar y un cuadro de texto para hacer la descripción del comportamiento y tratamiento del paciente. 8.Los datos introducidos son guardados.
Flujos alternativos:	4a. No encuentra al paciente, muestra mensaje “Paciente no encontrado”. Se vuelve al paso 4.	
Postcondiciones:		

Nombre del Caso de Uso:	Administrar citas.	
Descripción:	Permite realizar el proceso de adición, modificación y eliminación de la cita de un paciente.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción administrar citas del menú enfermera. 3.Elige una de la opciones: adicionar, modificar, eliminar.	2.Muestra la pantalla administrar citas.
Flujos alternativos:		
Postcondiciones:		

Nombre del Caso de Uso:	Adicionar cita.	
Descripción:	Permite registrar una nueva cita.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción adicionar cita. 3.La enfermera introduce los datos necesarios del paciente para la cita. 4. Elige la opción adicionar.	2.Muestra la pantalla con los diferentes campos para ingresar los diferentes datos de la cita. 5.El sistema adiciona los datos
Flujos alternativos:	3a.Introduce incorrectamente los datos de la cita y se muestra un mensaje “ los datos están incorrectos”. Se vuelve al paso 3.	
Postcondiciones:		

Nombre del Caso de Uso:	Modificar cita.	
Descripción:	Permite modificar datos de la cita ya registrada del paciente.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción modificar cita. 3.La enfermera introduce los datos modificados. 4.Elige la opción modificar.	2.Muestra la pantalla con los diferentes campos para modificar los datos de la cita. 5.El sistema guarda las modificaciones.
Flujos alternativos:	3a.Introduce incorrectamente los datos de la cita y se muestra un mensaje “ los datos están incorrectos”. Se vuelve al paso 3.	
Postcondiciones:		

Nombre del Caso de Uso:	Eliminar cita.	
Descripción:	Permite eliminar datos de una cita ya registrada.	
Actor(es):	Enfermera.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción eliminar cita. 3.La enfermera elige cual paciente elimina su cita. 4.Elige el botón eliminar.	2.Se muestra en pantalla una lista con los datos de la cita de los pacientes. 5.El sistema elimina de manera lógica los datos del paciente elegido.
Flujos alternativos:		
Postcondiciones:		

Nombre del Caso de Uso:	Registrar historia clínica consulta interna.	
Descripción:	Permite realizar el registro de la historia clínica de un paciente, en consulta interna.	
Actor(es):	Personal Médico.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige registrar historia clínica consulta interna. 3.Selecciona de la lista el nombre del paciente. 5.Selecciona una opción.	2. Muestra la pantalla con una lista de pacientes con los datos básicos. 4.Muestra la pantalla datos historia clínica interna con los diferentes campos.
Flujos alternativos:		
Postcondiciones:		

Nombre del Caso de Uso:	Registrar historia clínica consulta externa.	
Descripción:	Permite realizar el registro de la historia clínica de un paciente, en consulta externa.	
Actor(es):	Personal Médico.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige registrar historia clínica consulta externa. 3.Selecciona de la lista el nombre del paciente. 5.Selecciona una opción.	2. Muestra la pantalla con una lista de pacientes con los datos básicos. 4.Muestra la pantalla datos historia clínica externa con los diferentes campos.
Flujos alternativos:		
Postcondiciones:		

Nombre del Caso de Uso:	Registrar motivo de consulta y enfermedad actual.	
Descripción:	Permite realizar el registro del motivo de la consulta y la enfermedad actual del paciente.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción consulta . 3.Introduce los datos del motivo de la consulta y la enfermedad actual. 4.Elige el botón guardar.	2.Muestra la pantalla motivo de consulta. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar consumo de sustancias.	
Descripción:	Permite el registro del consumo de sustancias del paciente.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción consumo . 3.Introduce los datos del consumo de sustancias. 4.Elige el botón guardar.	2.Muestra la pantalla registrar consumo. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar área familiar.	
Descripción:	Se realiza el registro de datos familiares del paciente.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción área familiar. 3.Introduce los datos del área familiar. 4.Elige el botón guardar.	2.Muestra la pantalla registrar área familiar. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar tratamientos realizados.	
Descripción:	Se realiza el registro de datos familiares del paciente.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción área familiar. 3.Introduce los datos del área familiar. 4.Elige el botón guardar.	2.Muestra la pantalla registrar área familiar. 5.Los datos son guardados.
Flujos alternativos:		

Nombre del Caso de Uso:	Registrar antecedentes H.CL interna.	
Descripción:	Se realiza el registro de los antecedentes personales, patológicos, escolares, laborales y sexuales del paciente en consulta interna.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción antecedentes. 3.Introduce los datos de los antecedentes. 4.Elige el botón guardar.	2. Muestra la pantalla antecedentes historia clínica interna. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar antecedentes H.CL externa.	
Descripción:	Se realiza el registro de los antecedentes personales y familiares del paciente en consulta externa.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción antecedentes. 3.Introduce los datos de los antecedentes. 4.Elige el botón guardar.	2. Muestra la pantalla antecedentes historia clínica externa. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar exámenes.	
Descripción:	Se realiza el registro de los exámenes que se le hicieron al paciente, los cuales pueden ser examen físico, mental y neurológico.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción exámenes. 3.Introduce los datos de los exámenes. 4.Elige el botón guardar.	2. Muestra la pantalla registrar exámenes. 5.Los datos son guardados.
Flujos alternativos:		

Nombre del Caso de Uso:	Registrar diagnóstico.	
Descripción:	Se realiza el registro del diagnóstico que presenta el paciente.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción diagnóstico. 3.Elige el tipo de diagnóstico e introduce los datos. 4.Elige el botón guardar.	2. Muestra la pantalla registrar diagnóstico. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar evolución y resumen.	
Descripción:	Se realiza el registro del seguimiento que se le hace al paciente en cuanto a su tratamiento.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción evolución y resumen. 3.Introduce los datos correspondientes. 4.Elige el botón guardar.	2. Muestra la pantalla registrar evolución y resumen. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar propuesta terapéutica.	
Descripción:	Permite realizar el registro de la propuesta terapéutica a la que será sometido el paciente.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción propuesta terapéutica . 3.Registra la propuesta terapéutica. 4.Elige el botón guardar.	2.Muestra la pantalla registrar propuesta terapéutica. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar profesionales actuantes.	
Descripción:	Se realiza el registro de los profesionales actuantes en el registro de la historia del paciente atendido.	
Actor(es):	Personal médico.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla datos historia, elige la opción profesionales actuantes . 3.Registra los profesionales actuantes. 4.Elige el botón guardar.	2.Muestra la pantalla registrar profesionales actuantes. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Registrar informe psicológico	
Descripción:	Permite realizar el registro del informe psicológico	
Actor(es):	Psicólogo.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En el menú del psicólogo elige la opción registrar informe psicológico . 3.Registra los datos concernientes al informe psicológico. 4.Elige el botón guardar.	2.Muestra la pantalla registrar informe psicológico. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Generar reportes.	
Descripción:	Permite generar reportes de acuerdo al tipo seleccionado.	
Actor(es):	Psicólogo.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Selecciona la opción generar reportes . 3.Selecciona un tipo de reporte.	2.Muestra la pantalla con los diferentes reportes .
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Ver cantidad de pacientes.	
Descripción:	Permite ver la cantidad de pacientes que fueron atendidos en la institución, ya sea en consulta externa o consulta interna, además de calcular porcentajes y frecuencias.	
Actor(es):	Psicólogo.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Selecciona ver cantidad de pacientes. . 3.Introduce el rango de fechas.	2.Muestra la pantalla generar reportes . 4.El sistema visualiza el reporte.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Ver patologías frecuentes.	
Descripción:	Permite ver la las patologías más frecuentes que son atendidas en la institución.	
Actor(es):	Psicólogo.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Selecciona ver patologías frecuentes. 3.Introduce el rango de fechas.	2.Muestra la pantalla generar reportes. 4.El sistema visualiza el reporte.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Ver consultas según sexo.	
Descripción:	Permite ver la cantidad de pacientes según edad que fueron atendidos en la institución, ya sea en consulta externa o consulta interna.	
Actor(es):	Psicólogo.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Selecciona ver consultas según sexo. 3.Introduce el rango de fechas.	2.Muestra la pantalla generar reportes . 4.El sistema visualiza el reporte.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Ver consultas según edad.	
Descripción:	Permite ver la cantidad de pacientes que fueron atendidos en la institución, ya sea en consulta externa o consulta interna.	
Actor(es):	Psicólogo.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Selecciona ver consultas según edad. 3.Introduce el rango de fechas.	2.Muestra la pantalla generar reportes. 4.El sistema visualiza el reporte.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Administrar usuarios.	
Descripción:	Permite realizar el proceso de adición, modificación y eliminación de un usuario.	
Actor(es):	Administrador.	
Precondiciones:	El usuario debe estar registrado en el sistema.	
Flujo básico:	Actor	Sistema
	1.En la pantalla menú administrador elige la opción administrar usuarios. 3.Elige una de las opciones: adicionar, modificar, eliminar.	2.Muestra la pantalla administrar usuarios. 5.Los datos son guardados.
Flujos alternativos:		
Postcondiciones:	Se inicia una sesión en el sistema.	

Nombre del Caso de Uso:	Adicionar usuario.	
Descripción:	Permite registrar un nuevo usuario.	
Actor(es):	Administrador.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción adicionar usuario. 3.El administrador introduce los datos necesarios del usuario. 4. Elige la opción adicionar.	2.Muestra la pantalla con los diferentes campos para ingresar los diferentes datos del usuario. 5.El sistema adiciona los datos
Flujos alternativos:	3a.Introduce incorrectamente los datos del usuario y se muestra un mensaje “ los datos están incorrectos”. Se vuelve al paso 3.	
Postcondiciones:		

Nombre del Caso de Uso:	Modificar usuario.	
Descripción:	Permite modificar datos del usuario registrado en el sistema.	
Actor(es):	Administrador.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción modificar usuario. 3.El administrador del sistema introduce los datos modificados. 4.Elige la opción modificar.	2.Muestra la pantalla con los diferentes campos para modificar los datos del usuario. 5.El sistema guarda las modificaciones.
Flujos alternativos:	3a.Introduce incorrectamente los datos del usuario y se muestra un mensaje “ los datos están incorrectos”. Se vuelve al paso 3.	
Postcondiciones:		

Nombre del Caso de Uso:	Eliminar usuario.	
Descripción:	Permite eliminar datos de un usuario ya registrado.	
Actor(es):	Administrador.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción eliminar usuario. 3.El administrador elige cual usuario eliminar. 4.Elige el botón eliminar.	2.Se muestra en pantalla una lista con los datos de los usuarios. 5.El sistema elimina de manera lógica los datos del usuario elegido.
Flujos alternativos:		
Postcondiciones:		

Nombre del Caso de Uso:	Realizar copias de seguridad.	
Descripción:	Permite que el administrador realice copias de seguridad de la base de datos.	
Actor(es):	Administrador.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción realizar copias de seguridad. 3.El administrador introduce el nombre y elige el botón generar copia.	2.Se muestra la pantalla copias de seguridad. 5.El sistema genera una copia.
Flujos alternativos:		
Postcondiciones:		

Nombre del Caso de Uso:	Restaurar base de datos.	
Descripción:	Permite que el administrador restaure la base de datos.	
Actor(es):	Administrador.	
Precondiciones:	El actor debe estar autenticado en el sistema.	
Flujo básico:	Actor	Sistema
	1.Elige la opción restaurar base de datos. 3.El administrador elige que copia restaurar y presiona el botón restaurar.	2.Se muestra la pantalla restaurar base de datos. 5.El sistema restaura la base datos.
Flujos alternativos:		

El modelo de análisis ofrece una especificación más precisa de los requisitos, los estructura de modo que facilita la comprensión, su preparación, su modificación y en general su mantenimiento, utilizando el lenguaje de los desarrolladores.

El modelo de análisis puede considerarse como una primera aproximación al modelo de diseño.

4.1 CLASES DE ANALISIS

En el modelo de análisis las clases siempre encajan en uno de tres estereotipos básicos: de interfaz, de control o de entidad. (Ver figura 7)

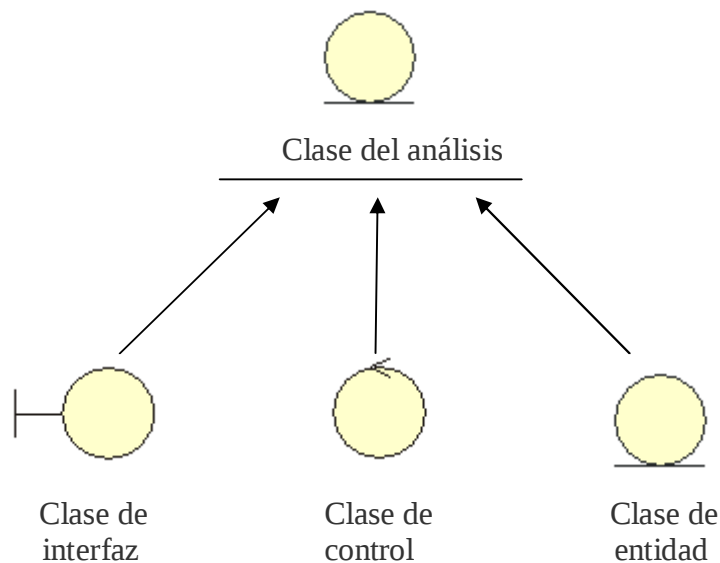
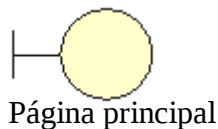


Figura 7. Estereotipos de una clase de análisis



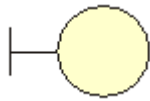
Página principal

Representa la página principal del sistema.



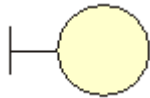
Pantalla Menú Enfermera

Representa la pantalla del menú de la enfermera.



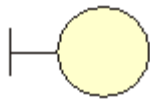
Representa la pantalla menú personal médico.

Pantalla Menú Personal Medico



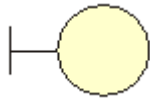
Representa la pantalla del menú del psicólogo.

Pantalla Menú Psicólogo



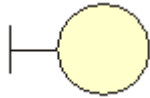
Representa la pantalla del menú del administrador del sistema.

Pantalla Menú Administrador Sistema



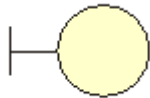
Representa la pantalla administrar pacientes, donde se elige adicionar, modificar y eliminar un paciente.

Pantalla Administrar Pacientes



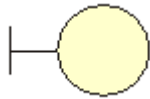
Representa la pantalla administrar citas, donde se elige adicionar, modificar y eliminar una cita.

Pantalla Administrar Citas



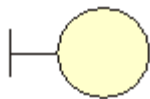
Representa la pantalla con los datos de la historia.

Pantalla Datos Historia



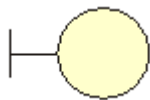
Representa la pantalla consulta donde se registra motivo de consulta y enfermedad actual.

Pantalla Consulta



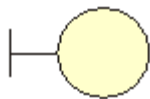
Representa la pantalla donde se registra el área familiar.

Pantalla Registrar Área Familiar



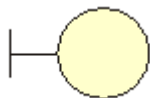
Es la pantalla donde se registra el tratamiento del paciente.

Pantalla Registrar Tratamiento



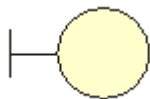
Representa la pantalla donde se registran los antecedentes de la historia clínica interna.

Pantalla Registrar Antecedentes H.CL Interna



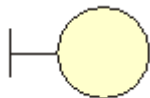
Representa la pantalla donde se registran los antecedentes de la historia clínica externa.

Pantalla Registrar Antecedentes H.CL Externa



Representa la pantalla donde se registra el diagnostico del paciente.

Pantalla Registrar Diagnostico



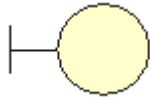
Representa la pantalla donde se registran la evolución y paciente.

Pantalla Registrar Evolución y Resumen



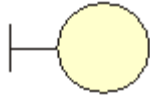
Representa la pantalla donde se registra la propuesta terapéutica.

Pantalla Registrar Propuesta Terapéutica



Representa la pantalla donde se registran los profesionales actuantes en el registro de la historia clínica del paciente.

Pantalla Registrar Profesionales Actuales



Representa la pantalla para donde los reportes del sistema.

Pantalla Generar Reportes



Personal

Representa a los usuarios que acceden al sistema



Rol Personal

Representa los roles a los que puede acceder el usuario: enfermera, personal médico, administrador sistema, psicólogo, psiquiatra, trabajadora social.



Menú

Representa el menú del usuario del sistema.



Paciente

Representa al paciente.



Persona

Representa a la clase persona en la cual está incluido el paciente con todos sus atributos.



Cita

Representa la cita a la cuál a la cuál accede el paciente.



Historia Clínica

Representa a la historia clínica del paciente.



Ficha Enfermera

Representa a la ficha de la enfermera con los datos del paciente.



Consulta

Representa la consulta del paciente.



Antecedentes

Representa los antecedentes del paciente.



Consumo de Sustancias

Representa la clase consumo de sustancias del paciente.



Lugares de Consumo

Representa los lugares de consumo de sustancias.



Obtención Tóxico

Representa la obtención del tóxico.



Antecedentes Legales Consumo

Representa los antecedentes legales del consumo de sustancias.



Área Familiar

Representa el área familiar del paciente.



Familia Biológica

Representa a la familia biológica del paciente.



Exámenes

Representa los exámenes del paciente.



Tratamiento

Representa el tratamiento del paciente.



Diagnóstico

Representa el diagnóstico del paciente.



Propuesta Terapéutica

Representa la propuesta terapéutica del paciente.



Evolución y Resumen

Representa la evolución y resumen del seguimiento al tratamiento del paciente.



Profesionales Actuantes

Representa los profesionales actuantes.



Informe Psicológico

Representa el informe psicológico del paciente.



Representa la interfaz generar porcentajes.

Generar Porcentajes



Representa la interfaz generar frecuencias.

Generar Frecuencias



Representa la interfaz generar patologías frecuentes.

Generar Patologías Frecuentes



Representa la interfaz generar consultas según edad.

Generar Consultas Según Edad



Representa la interfaz generar consultas según sexo.

Generar Consultas Según Sexo

4.2 DIAGRAMAS DE INTERACCIÓN

Los diagramas de interacción comprenden los diagramas de secuencia y los diagramas de colaboración.

4.2.1 Diagramas de Secuencia

4.2.1.1 Diagrama de secuencia para el caso de uso ingresar usuario y contraseña

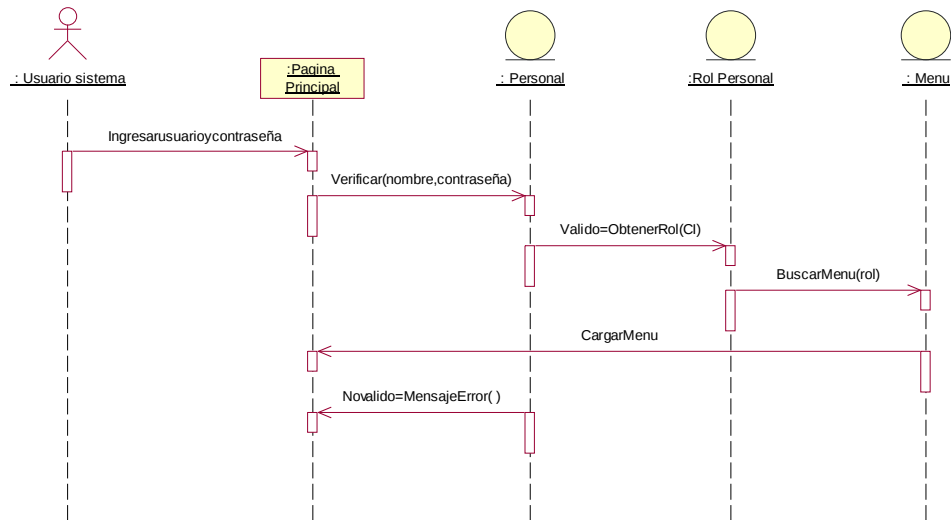


Figura 8. Diagrama de secuencia para el caso de uso ingresar usuario y contraseña

4.2.1.2 Diagrama de secuencia para el caso de uso administrar pacientes

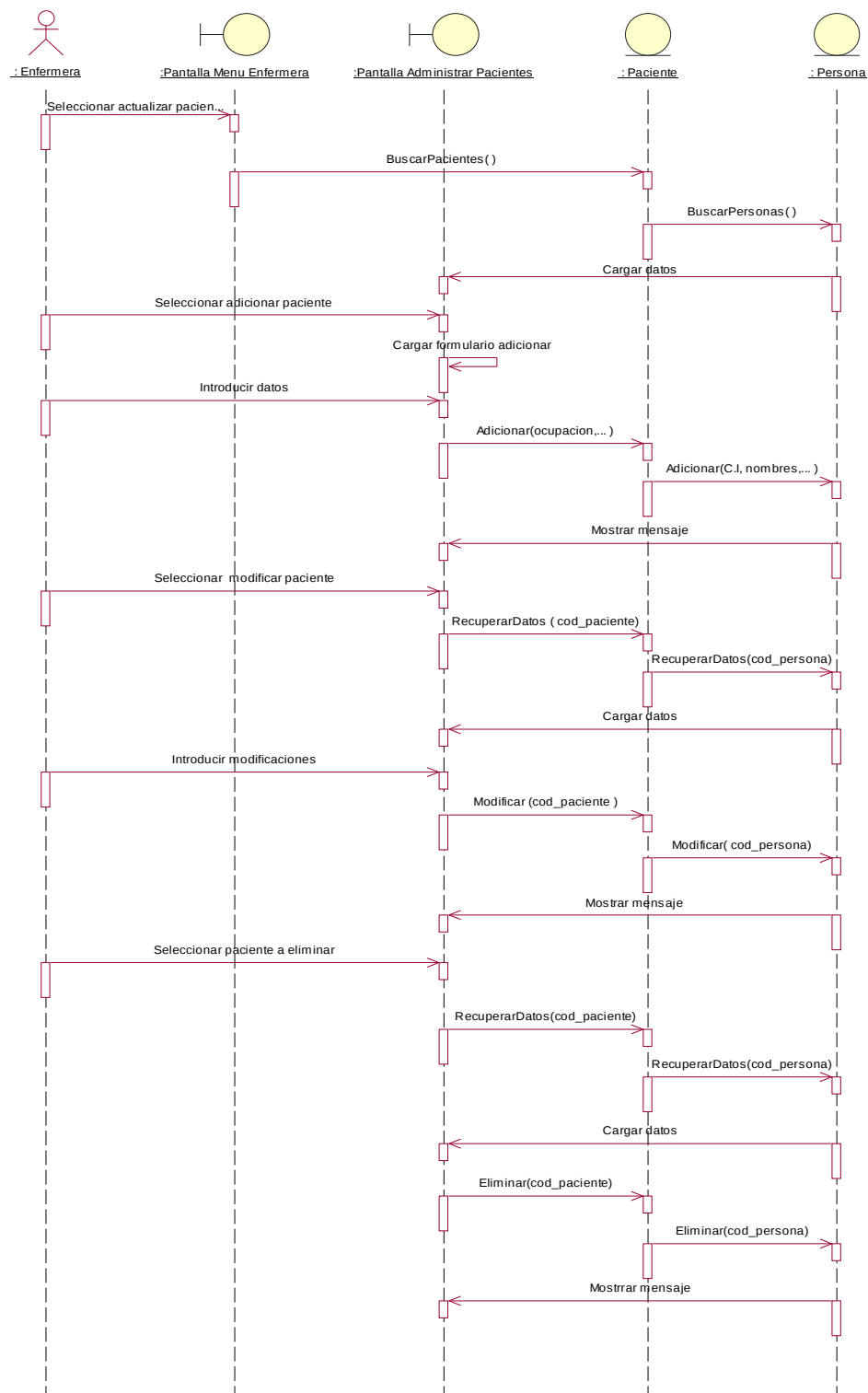


Figura 9. Diagrama de secuencia para el caso de uso administrar pacientes

4.2.1.3 Diagrama de secuencia para el caso de uso registrar comportamiento

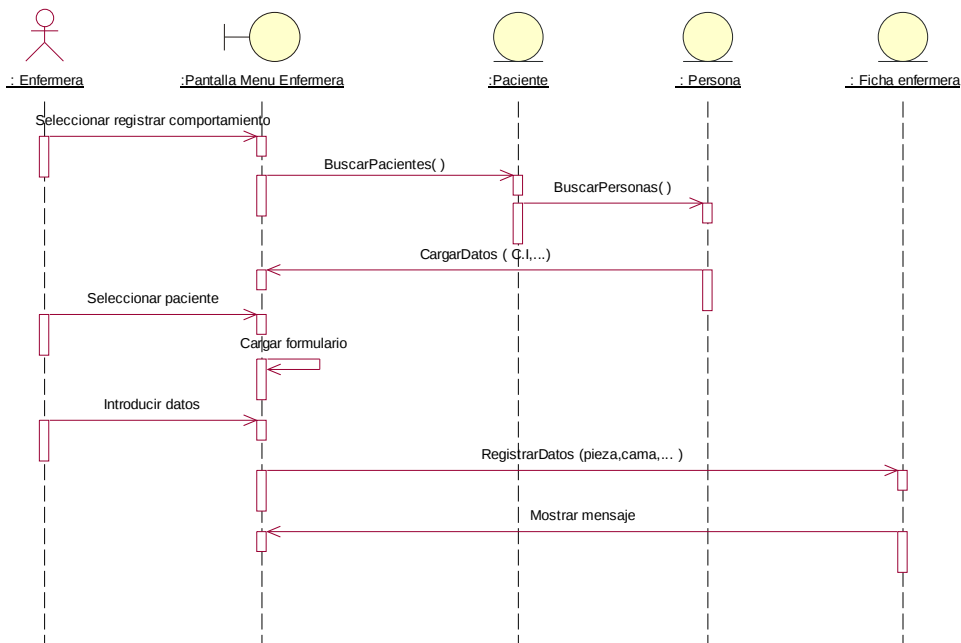


Figura 10. Diagrama de secuencia para el caso de uso registrar comportamiento

4.2.1.4. Diagrama de secuencia para el caso de uso administrar citas

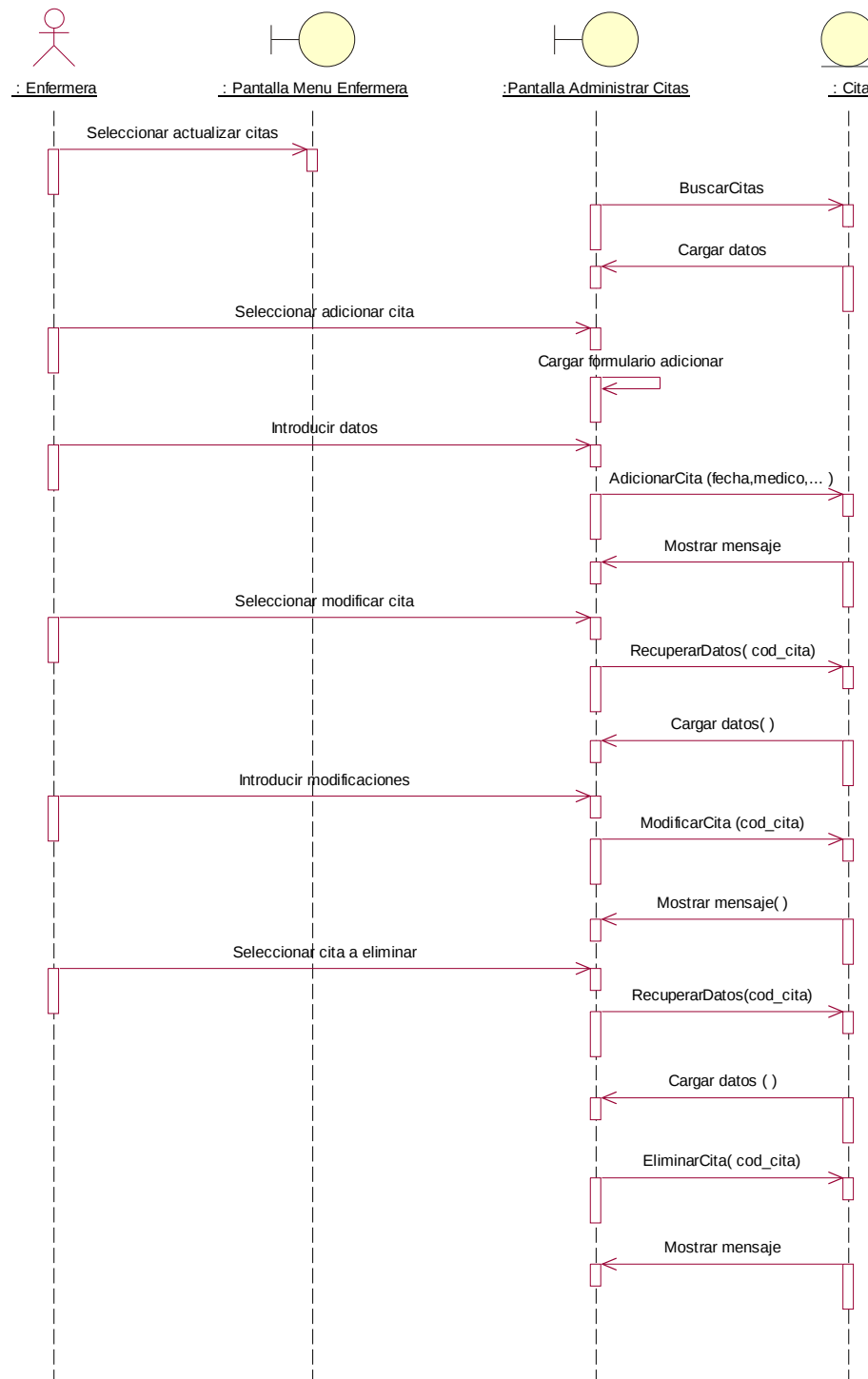


Figura 11. Diagrama de secuencia para el caso de uso administrar citas

4.2.1.5 Diagrama de secuencia para el caso de uso registrar historia clínica

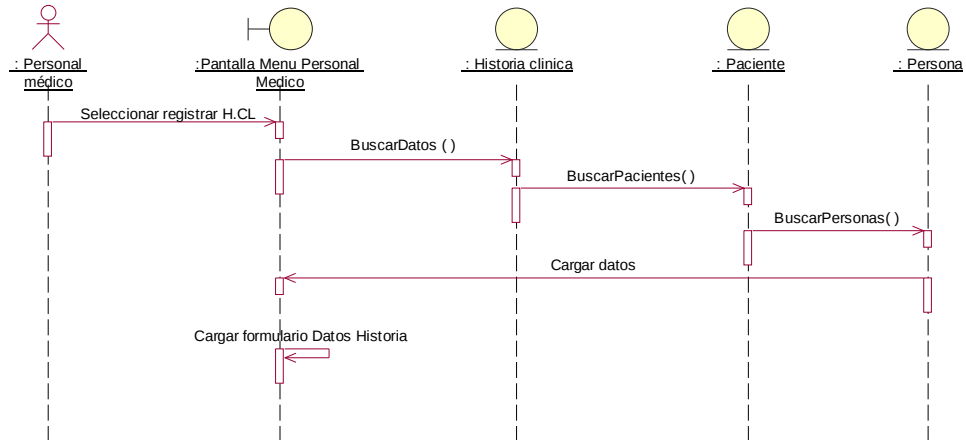


Figura 12. Diagrama de secuencia para el caso de uso registrar historia clínica

4.2.1.6 Diagrama de secuencia para el caso de uso registrar motivo de consulta y enfermedad actual

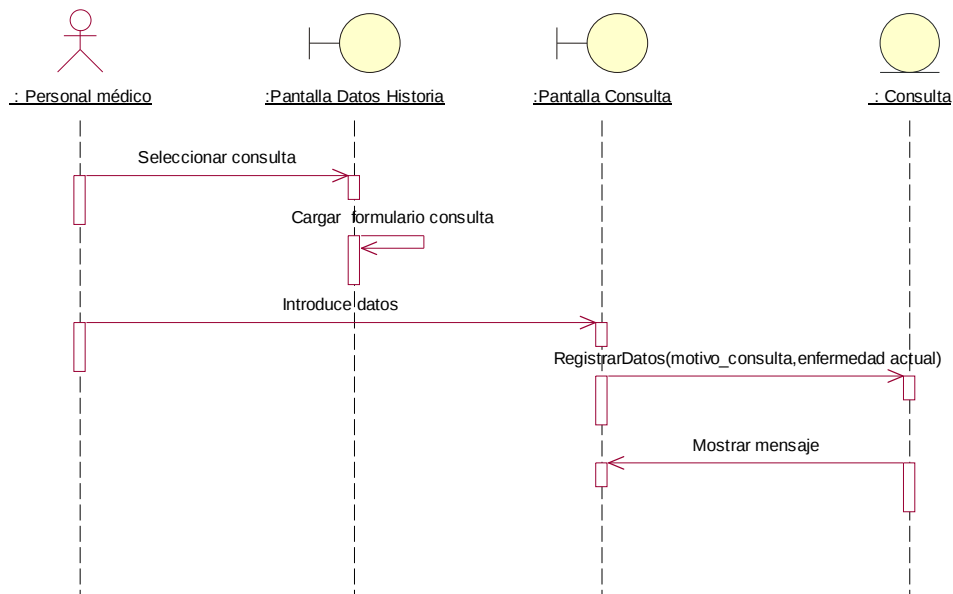


Figura 13. Diagrama de secuencia para el caso de uso registrar motivo de consulta y enfermedad actual

4.2.1.7 Diagrama de secuencia para el caso de uso registrar consumo de sustancias

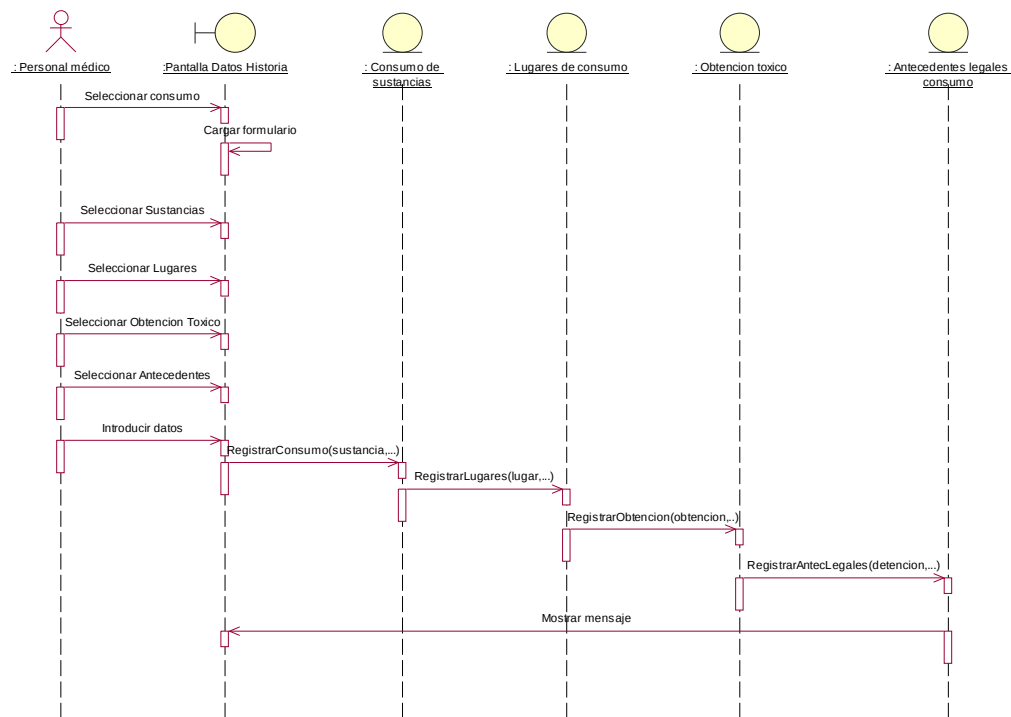


Figura 14. Diagrama de secuencia para el caso de uso registrar consumo de sustancias

4.2.1.8 Diagrama de secuencia para el caso de uso registrar área familiar

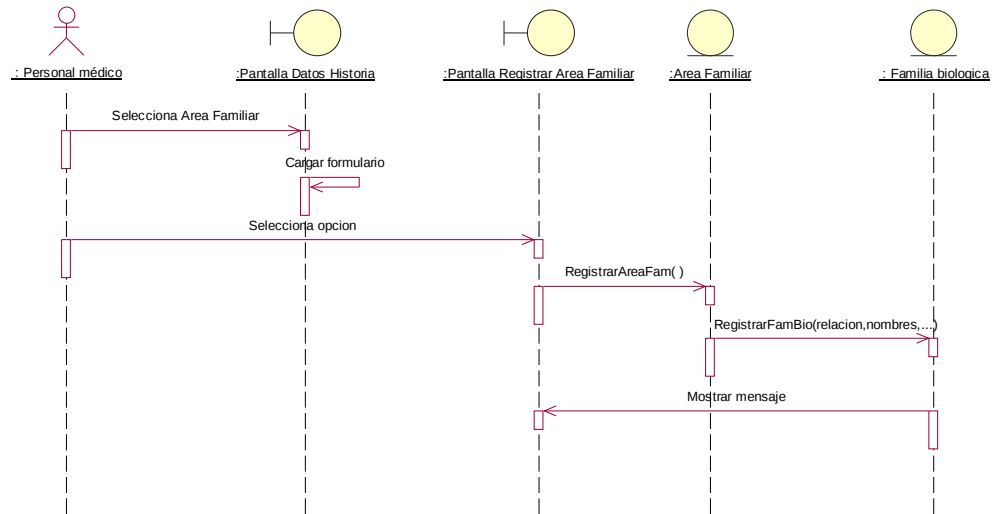


Figura 15. Diagrama de secuencia para el caso de uso registrar área familiar

4.2.1.9 Diagrama de secuencia para el caso de uso registrar tratamientos realizados

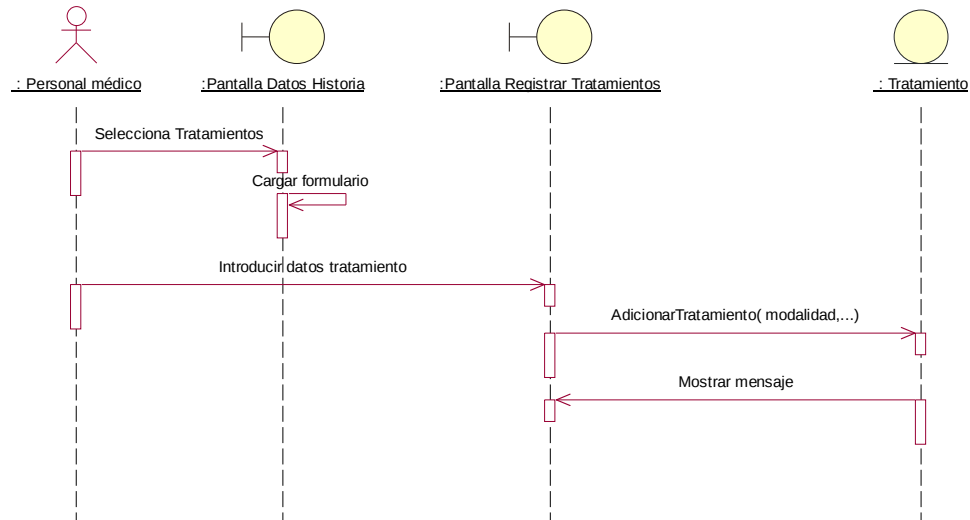


Figura 16. Diagrama de secuencia para el caso de uso registrar tratamientos realizados

4.2.1.10 Diagrama de secuencia para el caso de uso registrar antecedentes historia clínica interna

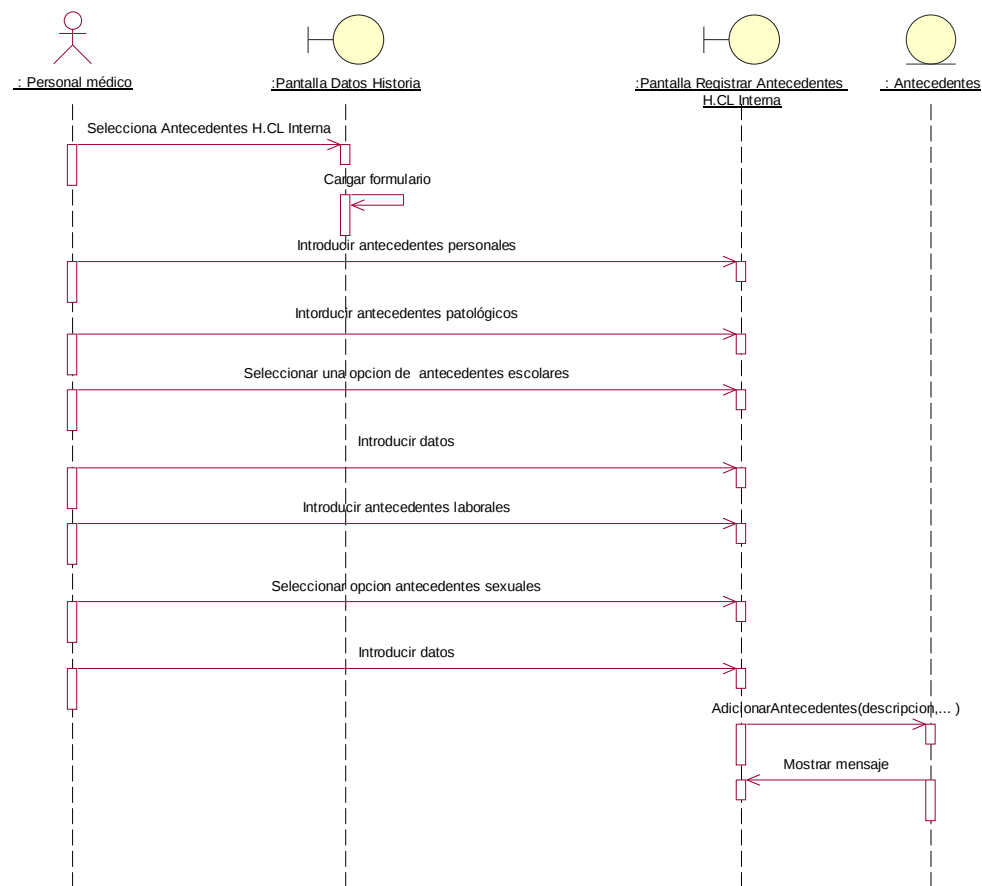


Figura 17. Diagrama de secuencia para el caso de uso registrar antecedentes historia clínica interna

4.2.1.11 Diagrama de secuencia para el caso de uso registrar antecedentes historia clínica externa

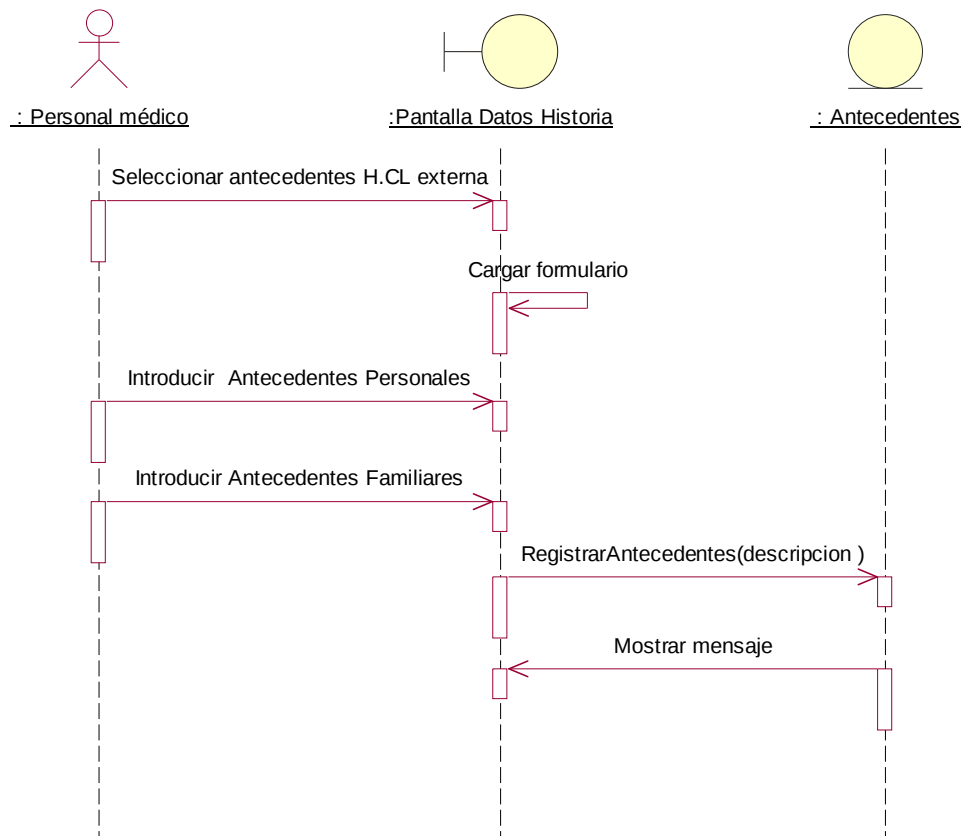


Figura 18. Diagrama de secuencia para el caso de uso registrar antecedentes historia clínica externa

4.2.1.12 Diagrama de secuencia para el caso de uso registrar diagnóstico

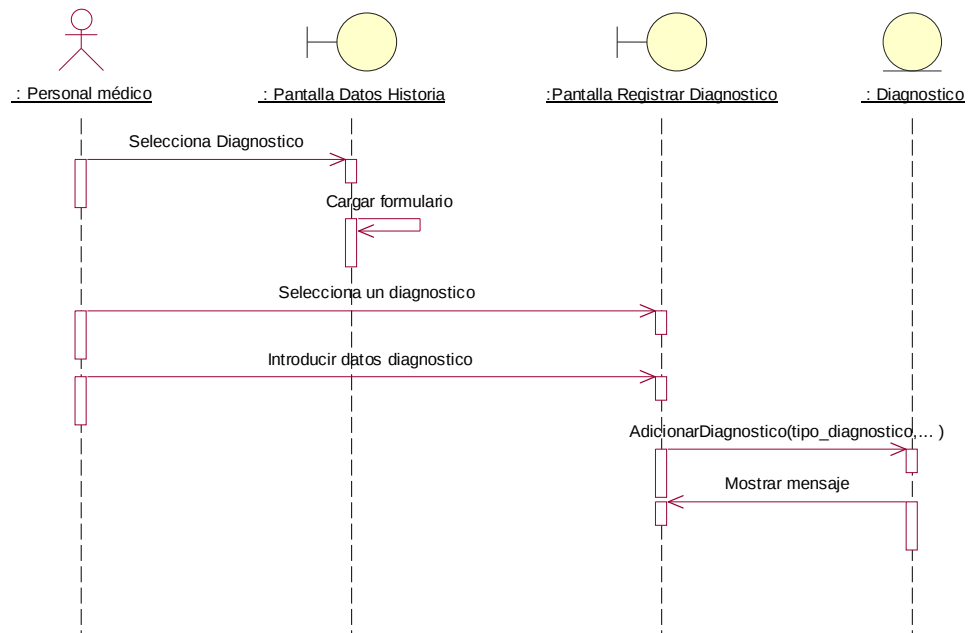


Figura 19. Diagrama de secuencia para el caso de uso registrar diagnóstico

4.2.1.13 Diagrama de secuencia para el caso de uso registrar evolución y resumen

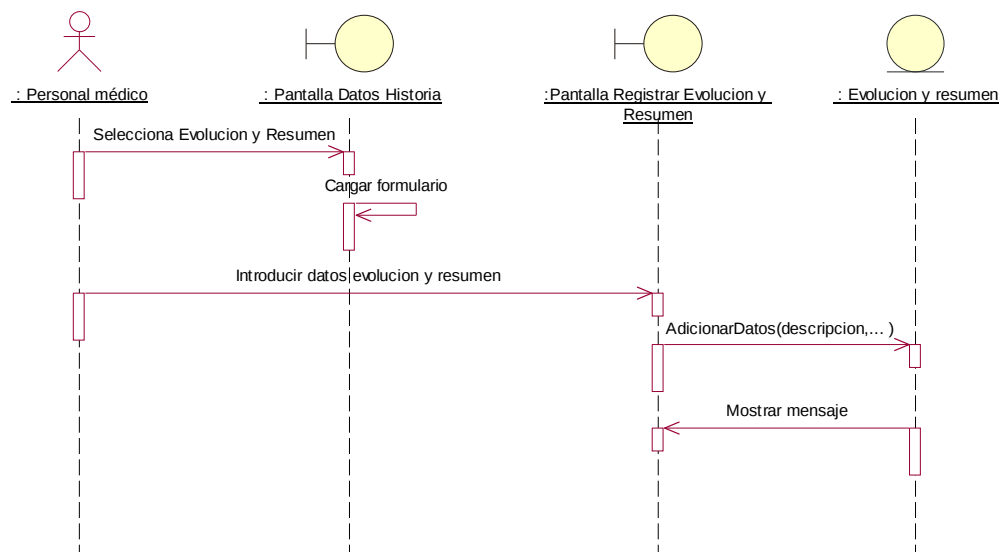


Figura 20. Diagrama de secuencia para el caso de uso registrar evolución y resumen

4.2.1.14 Diagrama de secuencia para el caso de uso registrar propuesta terapéutica

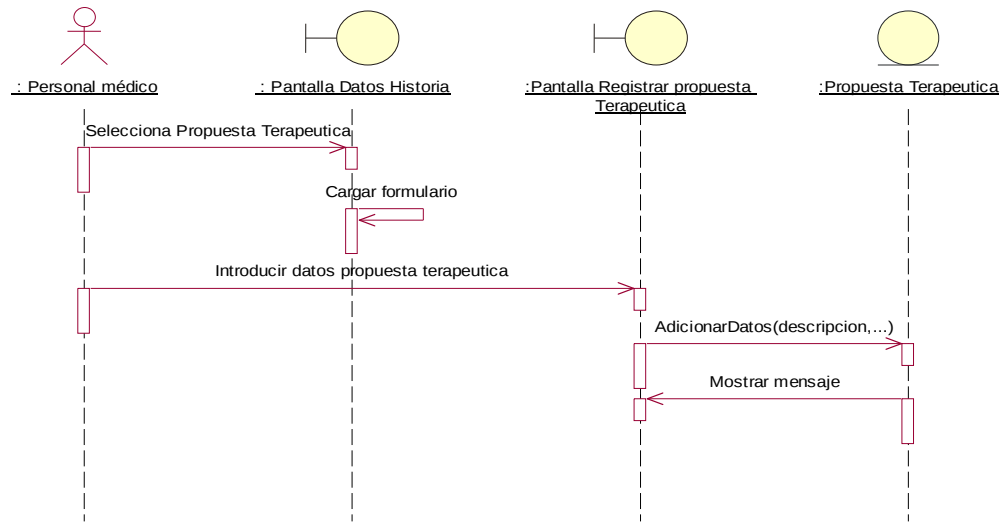


Figura 21. Diagrama de secuencia para el caso de uso registrar propuesta terapéutica

4.2.1.15 Diagrama de secuencia para el caso de uso registrar profesionales actuantes

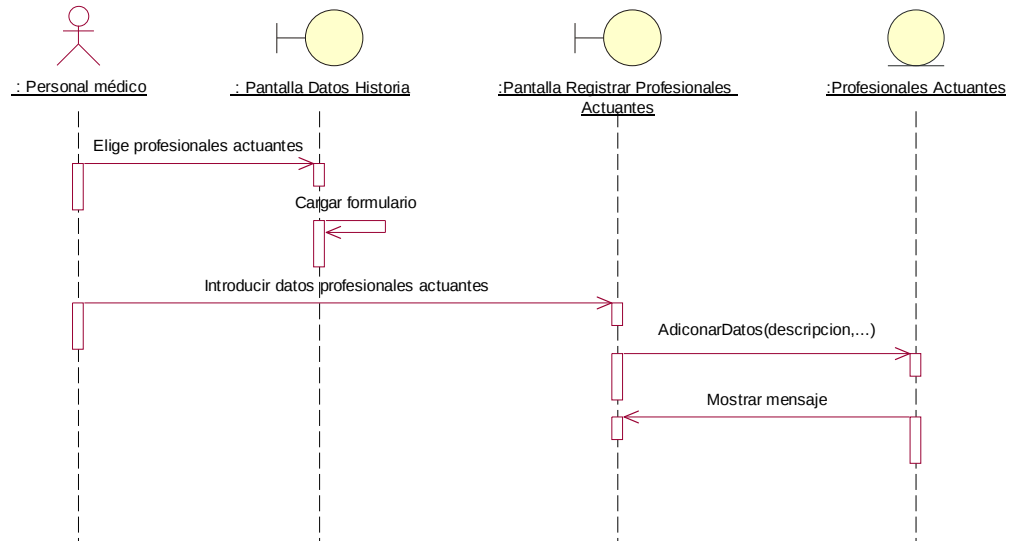


Figura 21. Diagrama de secuencia para el caso de uso registrar profesionales actuantes

4.2.1.16 Diagrama de secuencia para el caso de uso consultar historia clínica

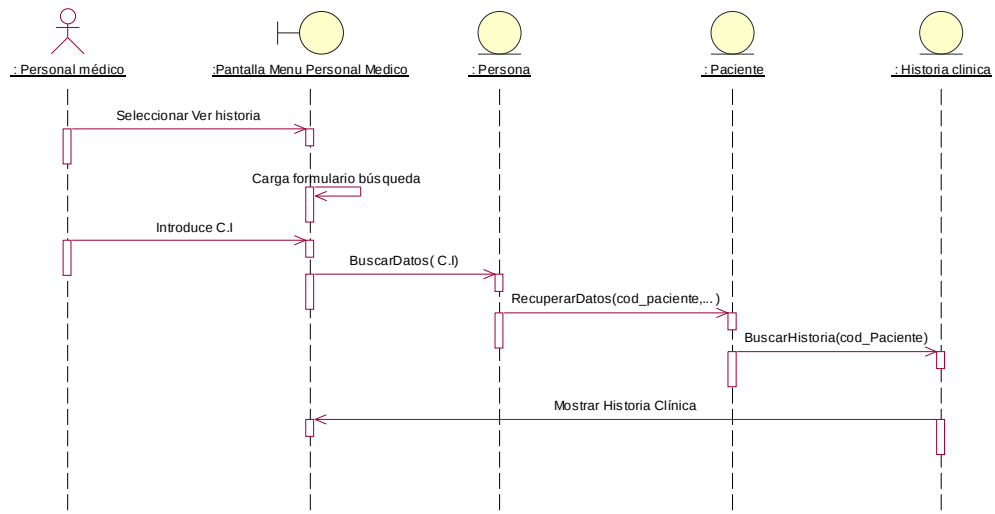


Figura 22. Diagrama de secuencia para el caso de uso consultar historia clínica

4.2.1.17 Diagrama de secuencia para el caso de uso registrar informe psicológico

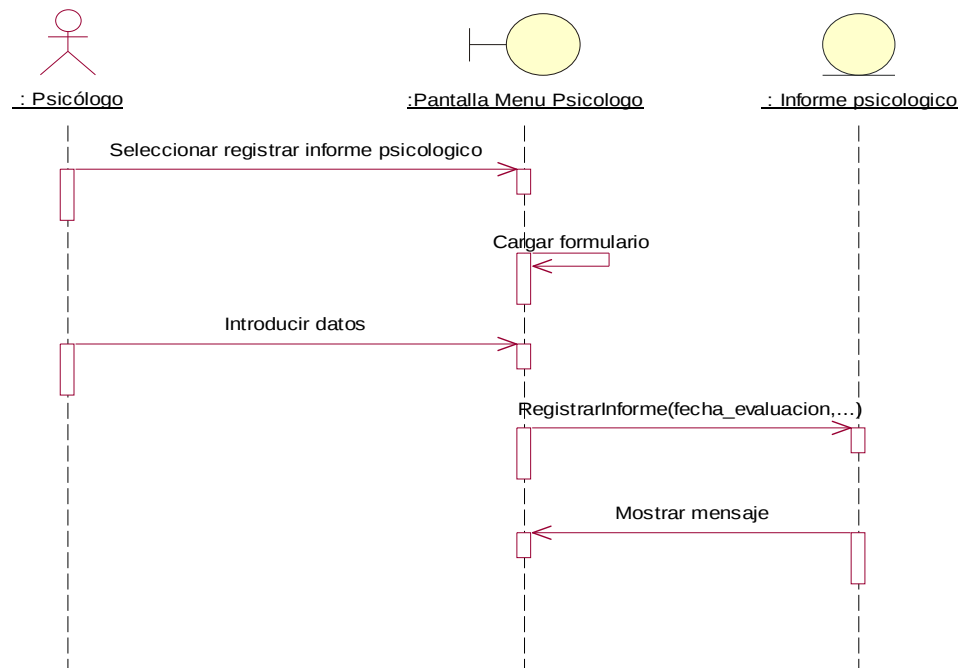


Figura 23. Diagrama de secuencia para el caso de uso registrar informe psicológico

4.2.1.18 Diagrama de secuencia para el caso de uso ver cantidad de pacientes

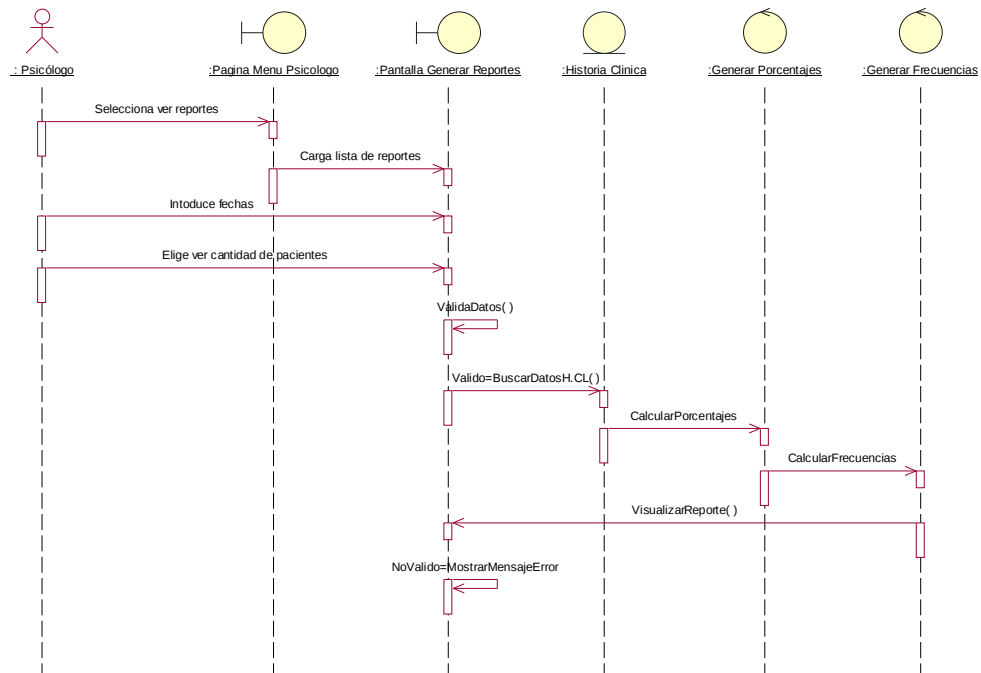


Figura 24. Diagrama de secuencia para el caso de uso ver cantidad de pacientes

4.2.1.19 Diagrama de secuencia para el caso de uso ver patologías frecuentes

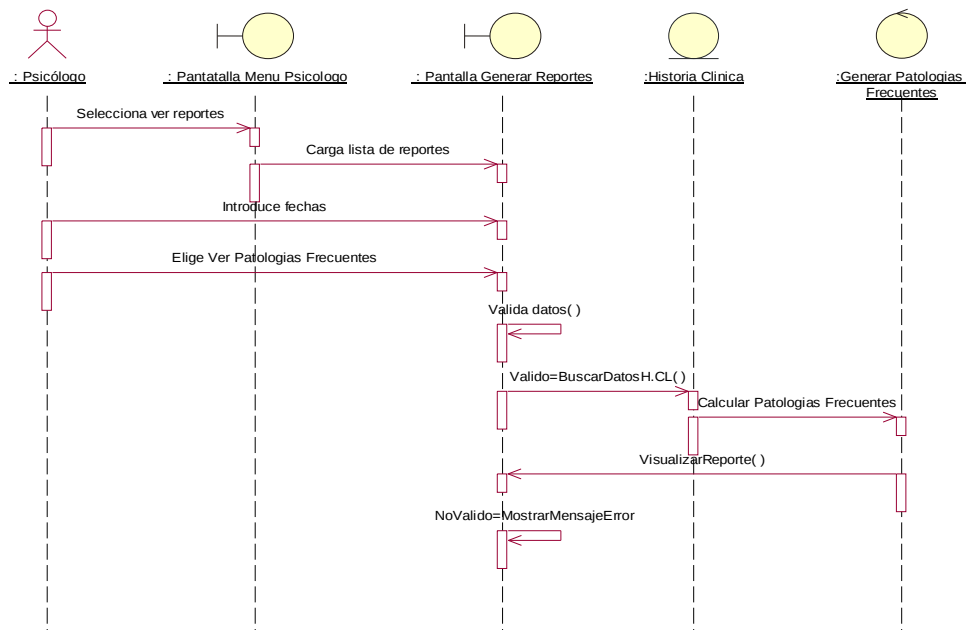


Figura 25. Diagrama de secuencia para el caso de uso ver patologías frecuentes

4.2.1.20 Diagrama de secuencia para el caso de uso ver consultas según sexo

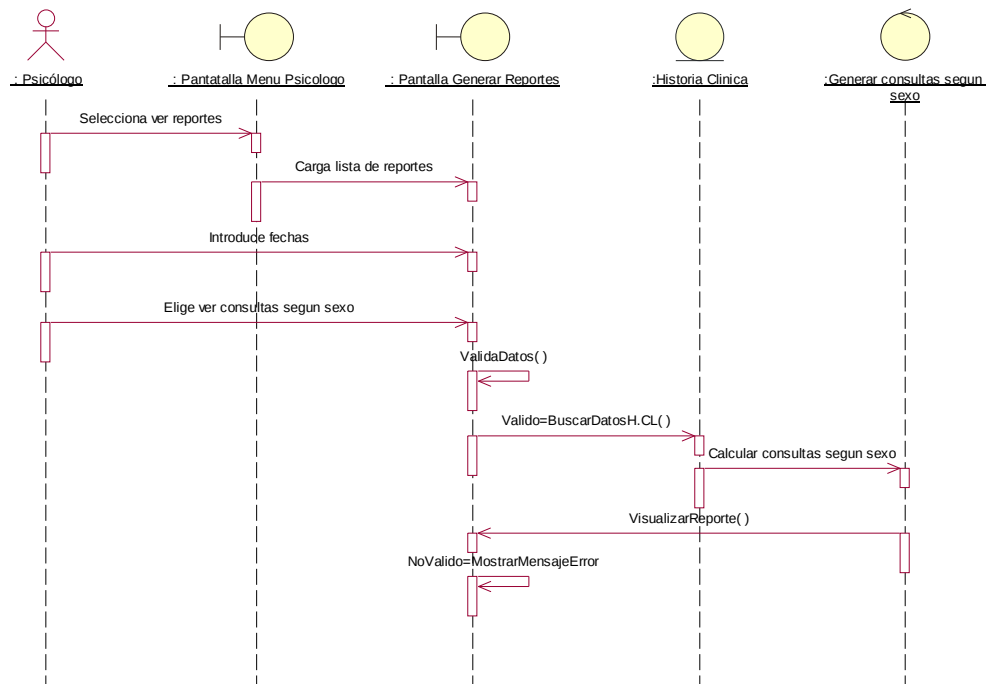


Figura 26. Diagrama de secuencia para el caso de uso ver consultas según sexo

4.2.1.21 Diagrama de secuencia para el caso de uso ver consultas según edad

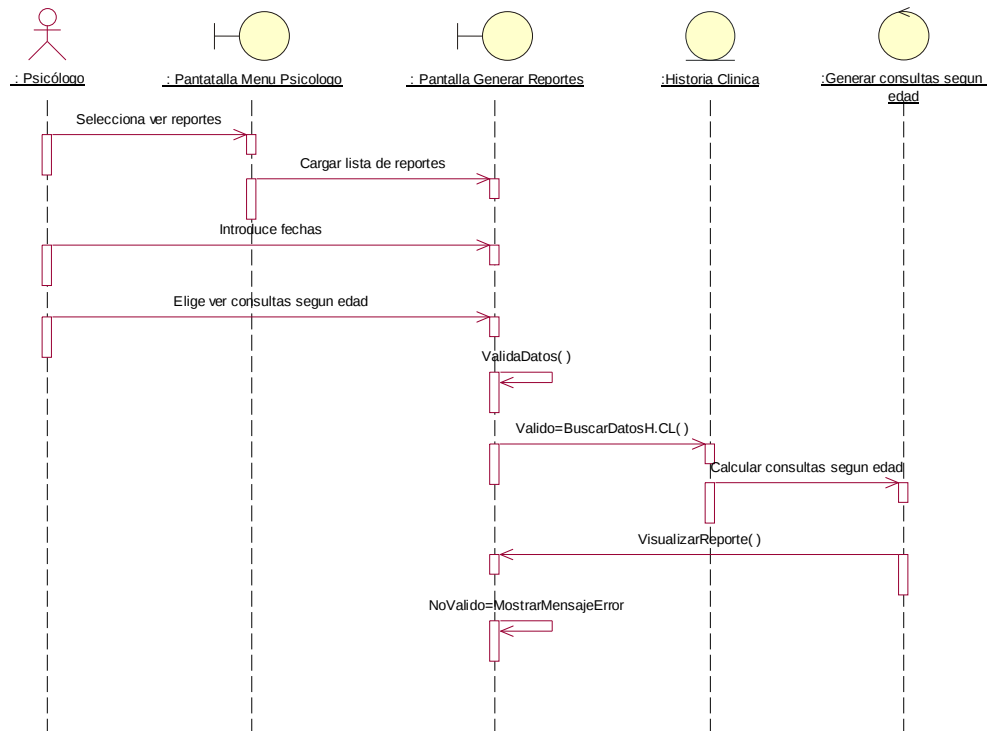


Figura 27. Diagrama de secuencia para el caso de uso ver consultas según edad

4.2.1.22 Diagrama de secuencia para el caso de uso administrar usuarios

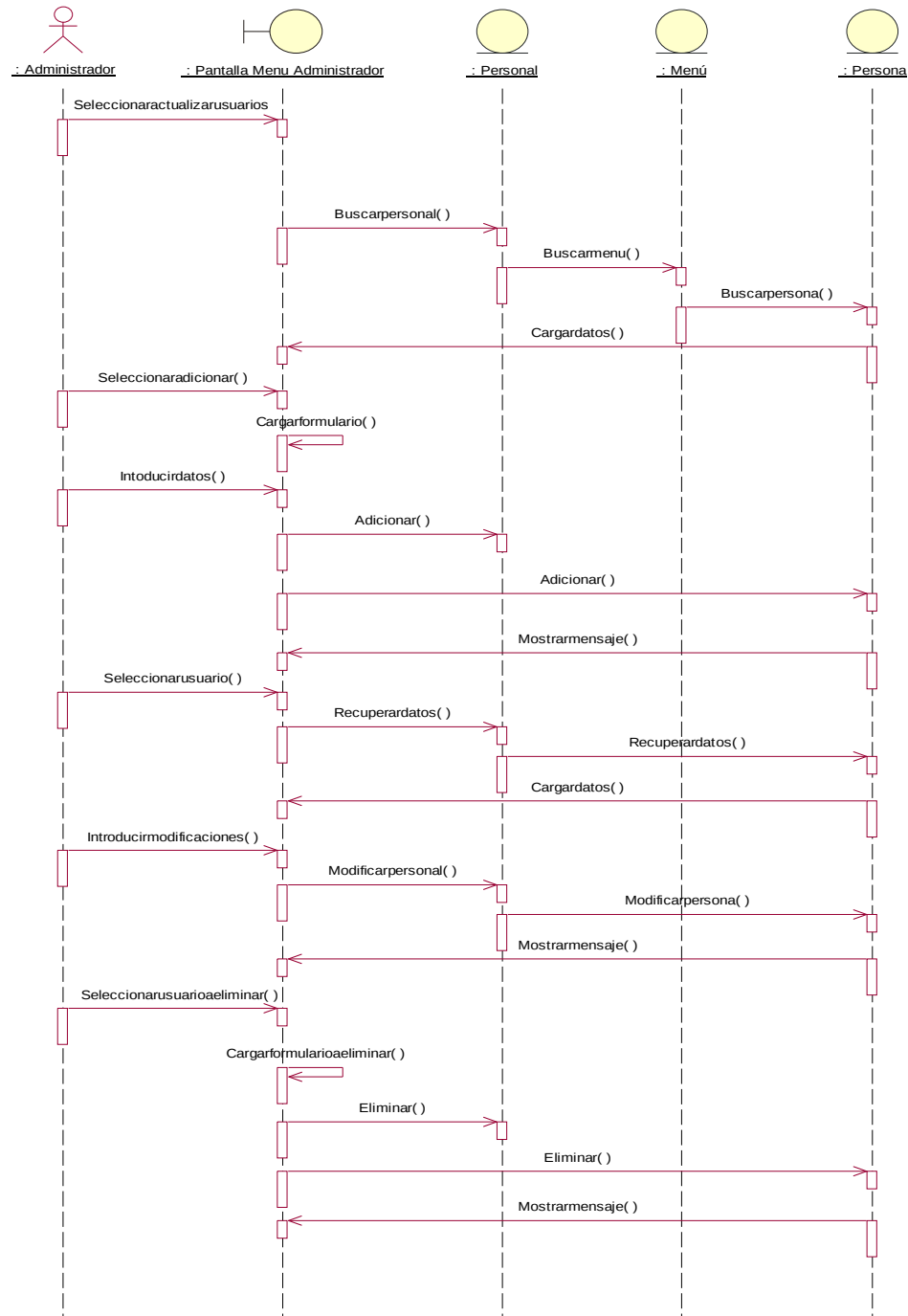


Figura 28. Diagrama de secuencia para el caso de uso administrar usuarios

4.2.1.23 Diagrama de secuencia para el caso de uso realizar copias de seguridad

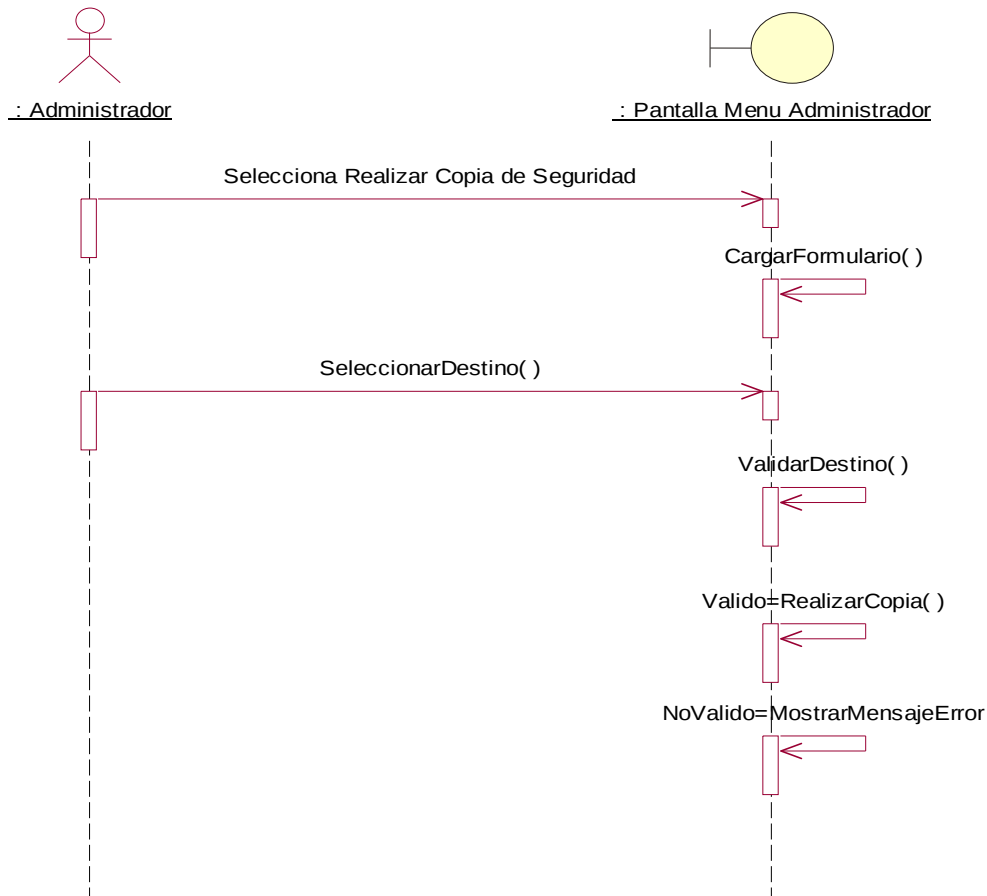


Figura 29. Diagrama de secuencia para el caso de uso realizar copias de seguridad

4.2.1.24 Diagrama de secuencia para el caso de uso restaurar base de datos

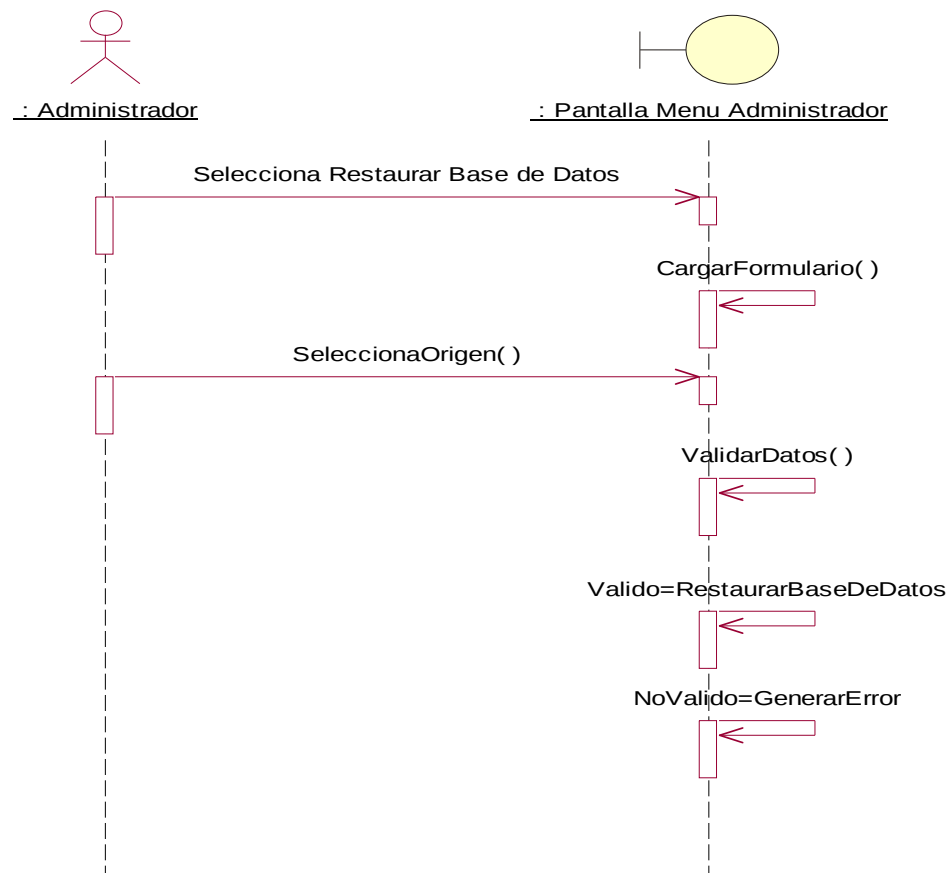


Figura 30. Diagrama de secuencia para el caso de uso restaurar base de datos

4.2.2 Diagramas de Colaboración

4.2.2.1 Diagrama de colaboración para el caso de uso ingresar usuario y contraseña

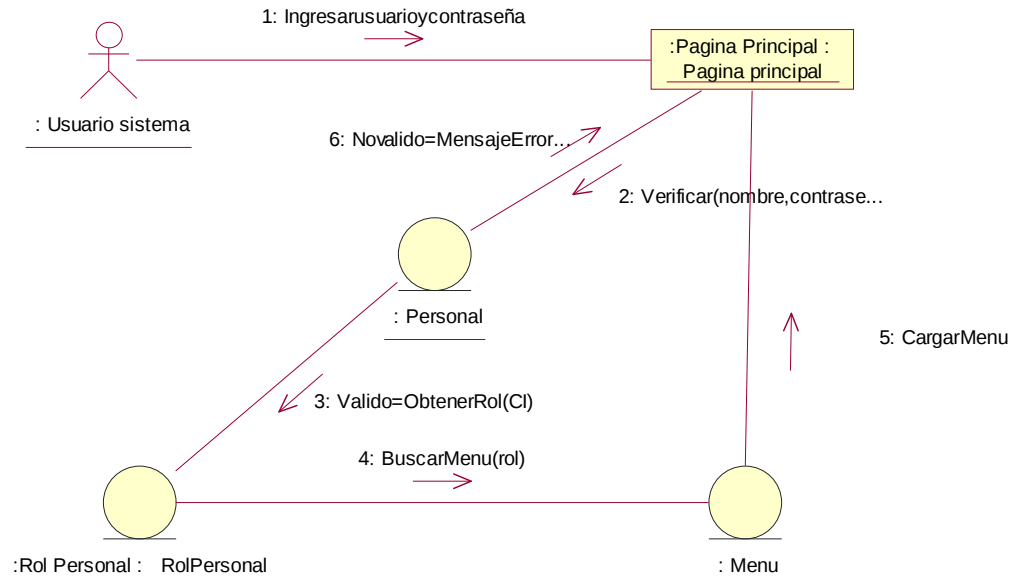


Figura 31. Diagrama de colaboración para el caso de uso ingresar usuario y contraseña

4.2.2.2 Diagrama de colaboración para el caso de uso administrar pacientes

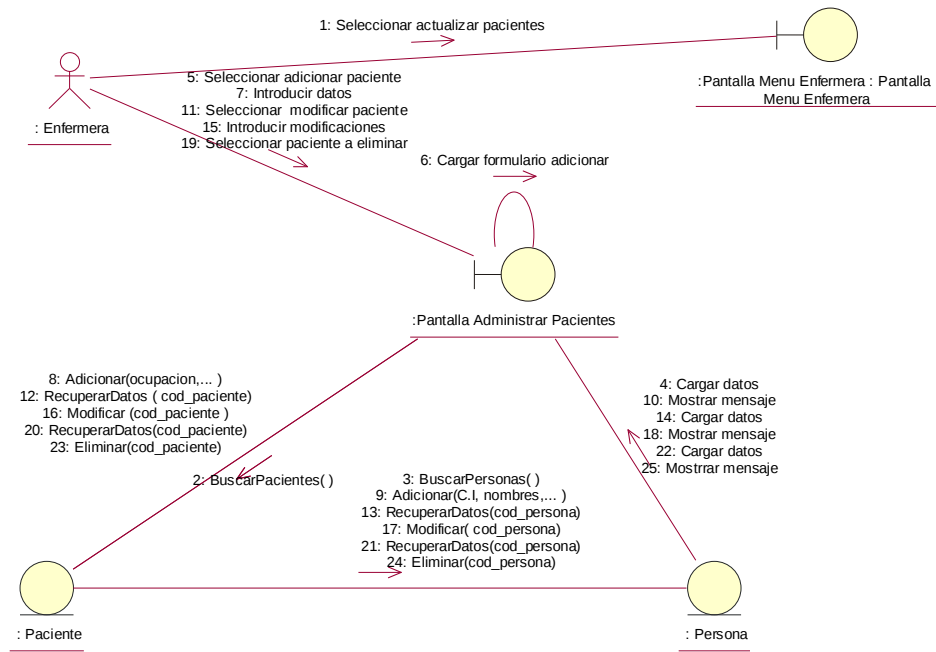


Figura 32. Diagrama de colaboración para el caso de uso administrar pacientes

4.2.2.3 Diagrama de colaboración para el caso de uso registrar comportamiento

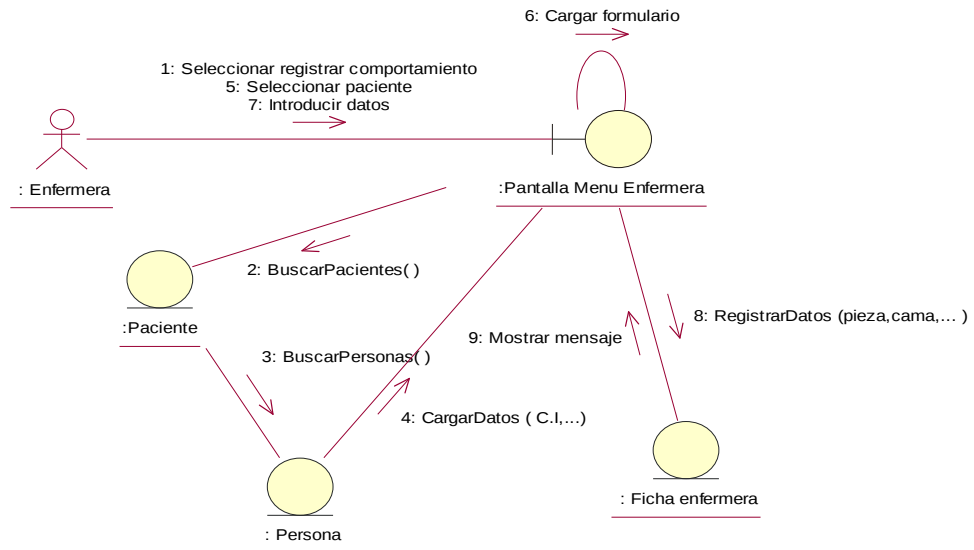


Figura 33. Diagrama de colaboración para el caso de uso registrar comportamiento

4.2.2.4 Diagrama de colaboración para el caso de uso administrar citas

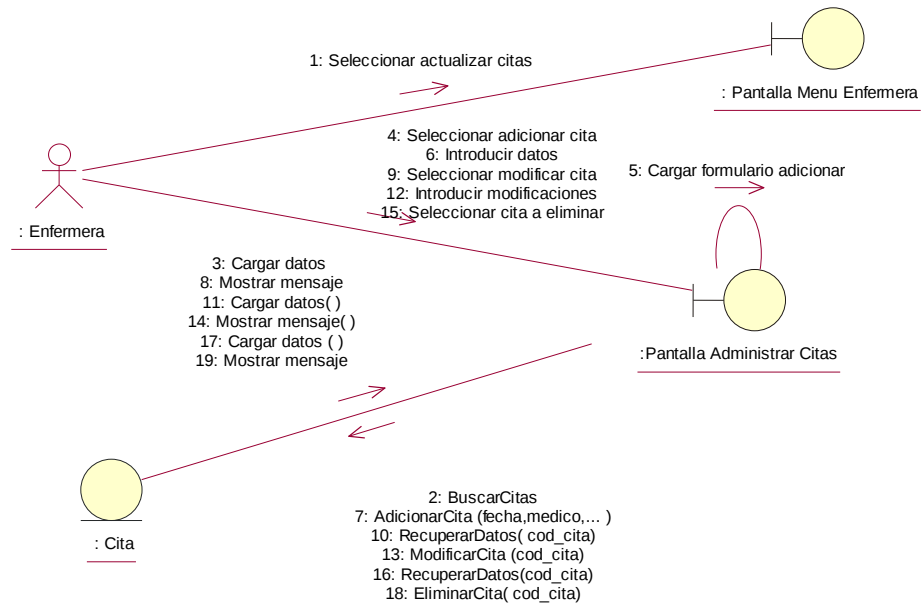


Figura 34. Diagrama de colaboración para el caso de uso administrar citas

4.2.2.5 Diagrama de colaboración para el caso de uso registrar historia clínica

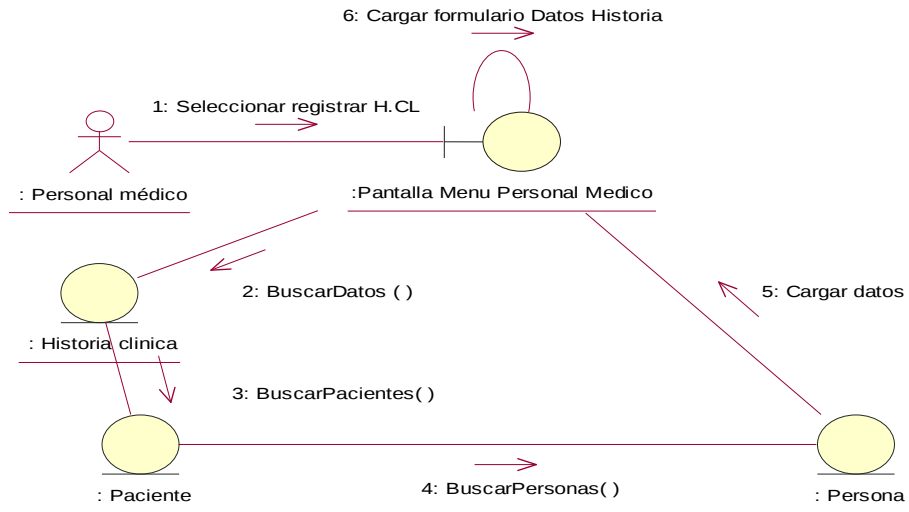


Figura 35. Diagrama de colaboración para el caso de uso registrar historia clínica

4.2.2.6 Diagrama de colaboración para el caso de uso registrar motivo de consulta y enfermedad actual

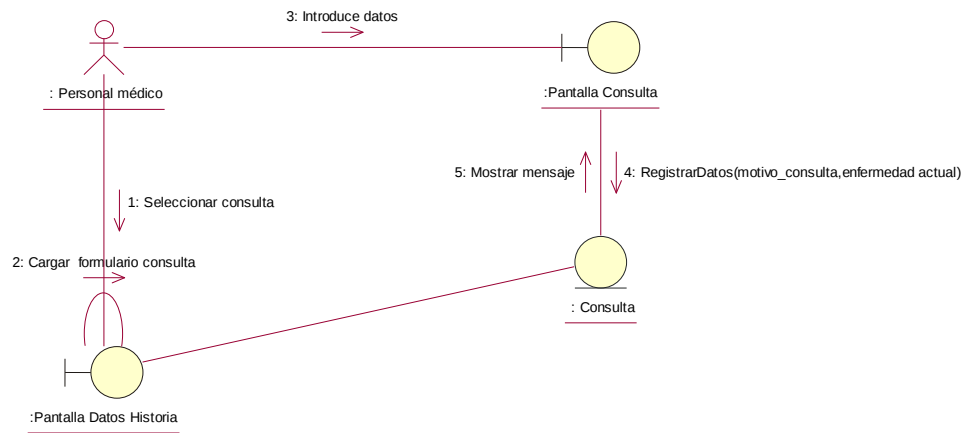


Figura 36. Diagrama de colaboración para el caso de uso registrar motivo de consulta y enfermedad actual

4.2.2.7 Diagrama de colaboración para el caso de uso registrar consumo de sustancias

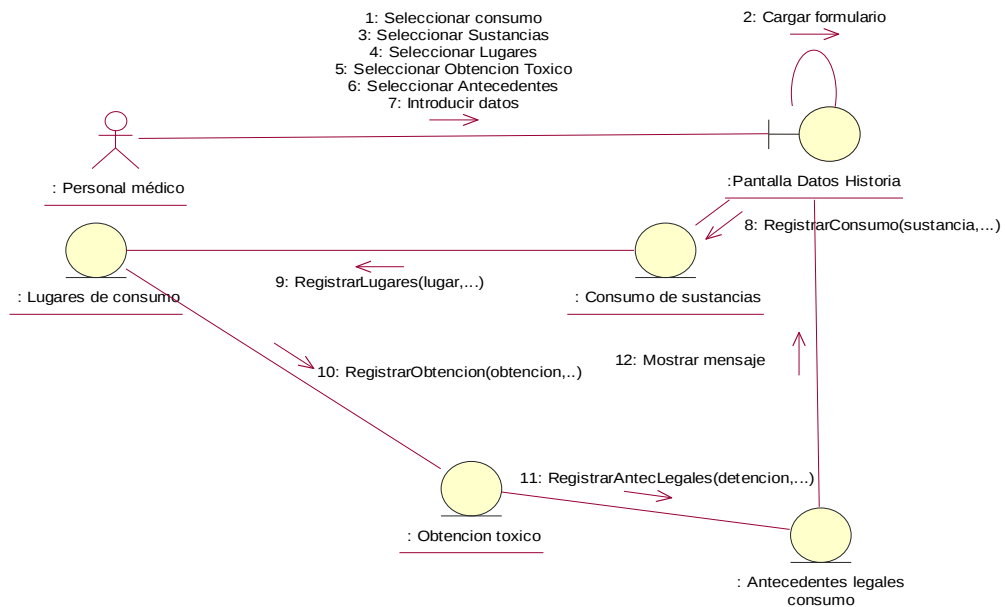


Figura 37. Diagrama de colaboración para el caso de uso registrar consumo de sustancias

4.2.2.8 Diagrama de colaboración para el caso de uso registrar área familiar

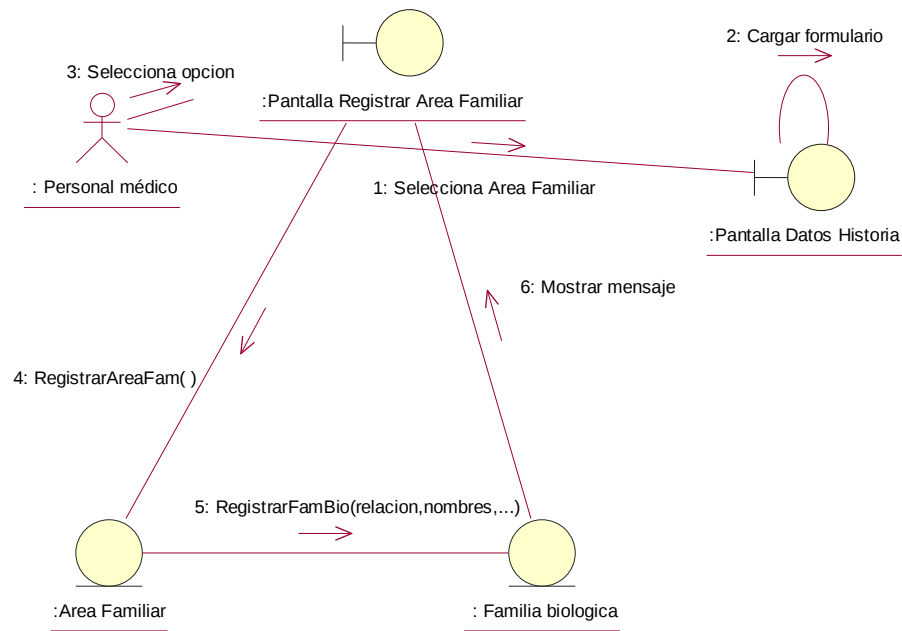


Figura 38. Diagrama de colaboración para el caso de uso registrar área familiar

4.2.2.9 Diagrama de colaboración para el caso de uso registrar tratamientos realizados

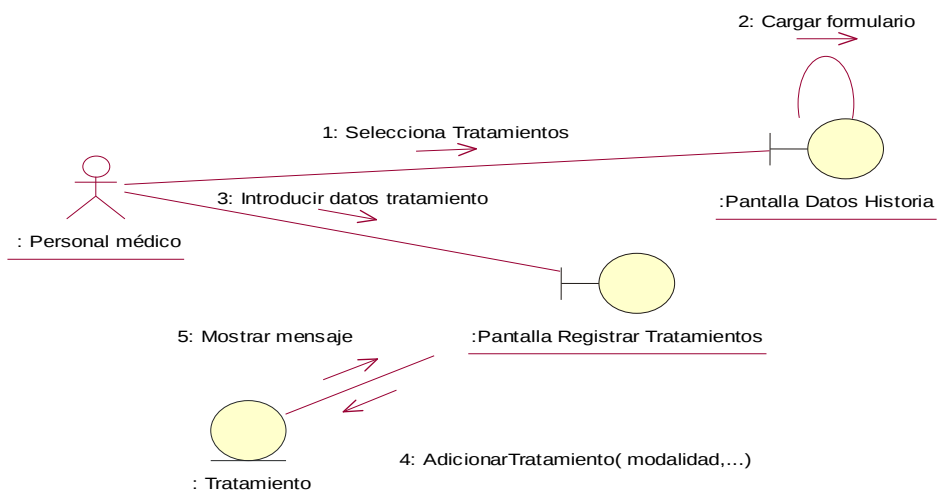


Figura 39. Diagrama de colaboración para el caso de uso registrar tratamientos realizados

4.2.2.10 Diagrama de colaboración para el caso de uso registrar antecedentes historia clínica interna

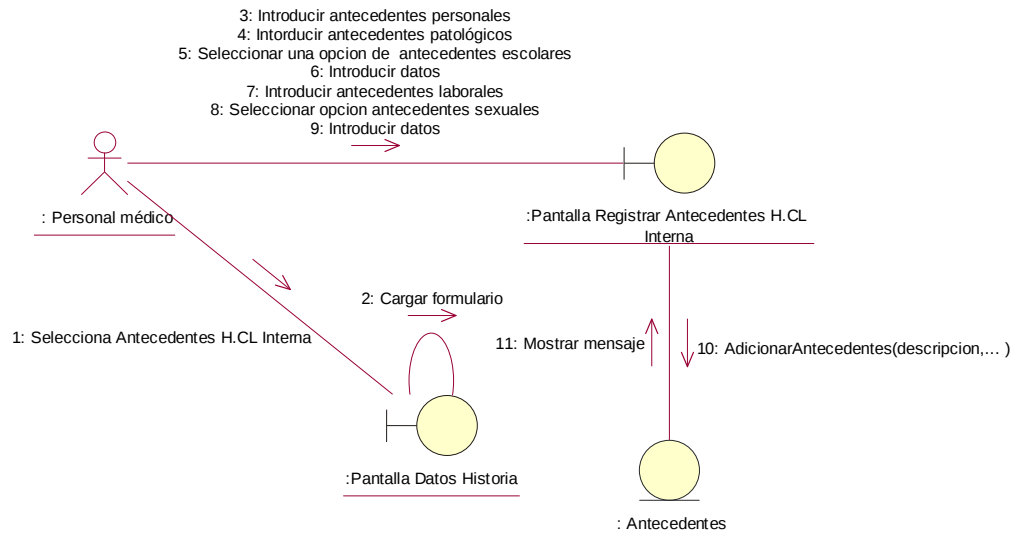


Figura 40. Diagrama de colaboración para el caso de uso registrar antecedentes historia clínica interna

4.2.2.11 Diagrama de colaboración para el caso de uso registrar antecedentes historia clínica externa

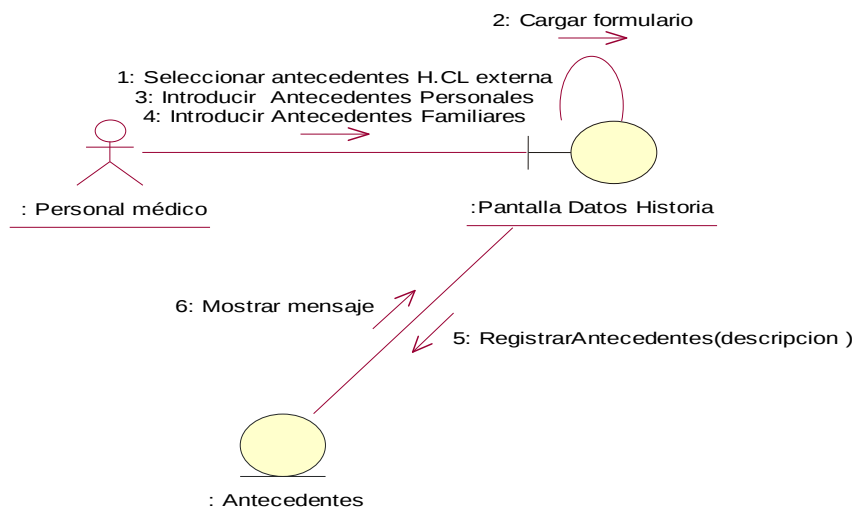


Figura 41. Diagrama de colaboración para el caso de uso registrar antecedentes historia clínica externa

4.2.2.12 Diagrama de colaboración para el caso de uso registrar exámenes

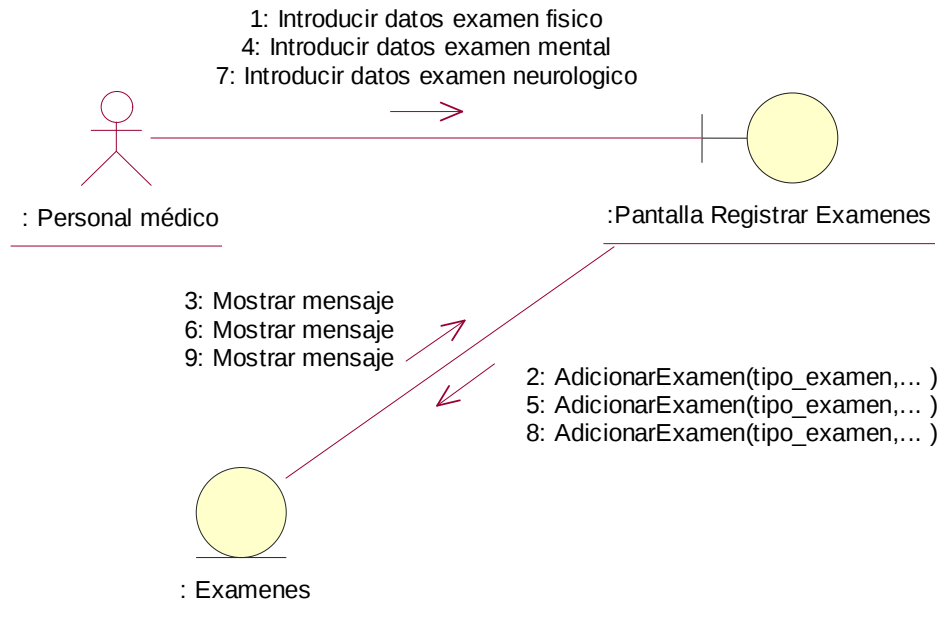


Figura 42. Diagrama de colaboración para el caso de uso registrar exámenes

4.2.2.13 Diagrama de colaboración para el caso de uso registrar diagnóstico

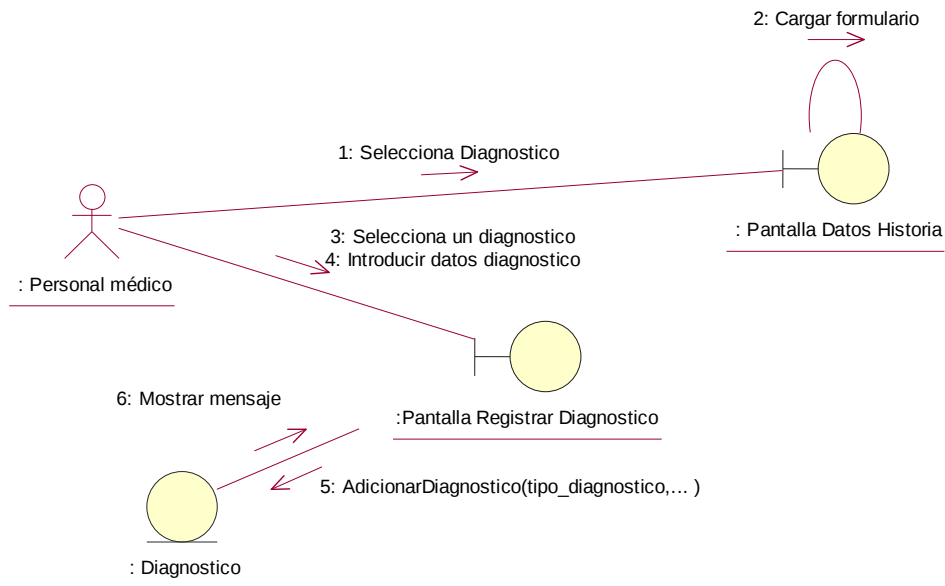


Figura 43. Diagrama de colaboración para el caso de uso registrar diagnóstico

4.2.2.14 Diagrama de colaboración para el caso de uso registrar evolución y resumen

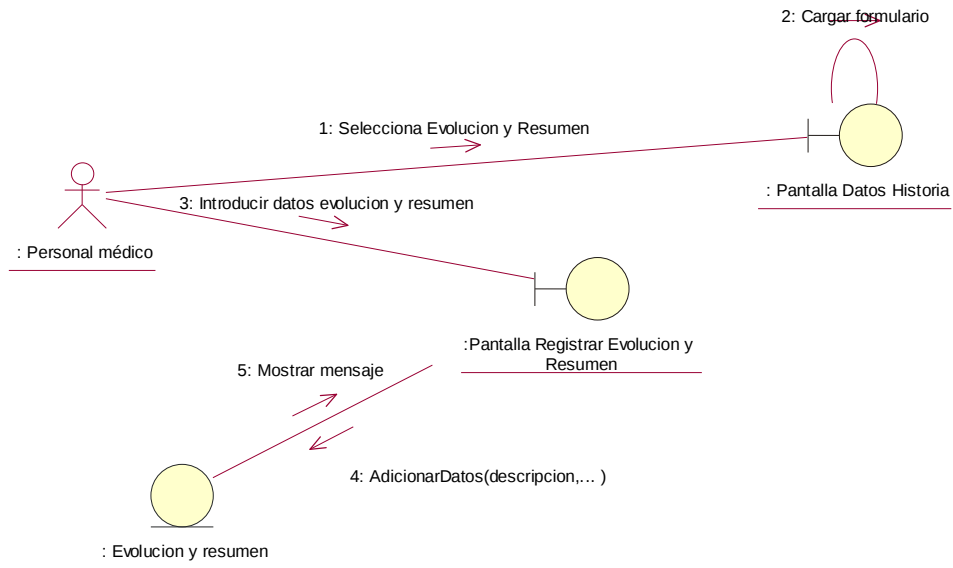


Figura 44. Diagrama de colaboración para el caso de uso registrar evolución y resumen

4.2.2.15 Diagrama de colaboración para el caso de uso registrar propuesta terapéutica

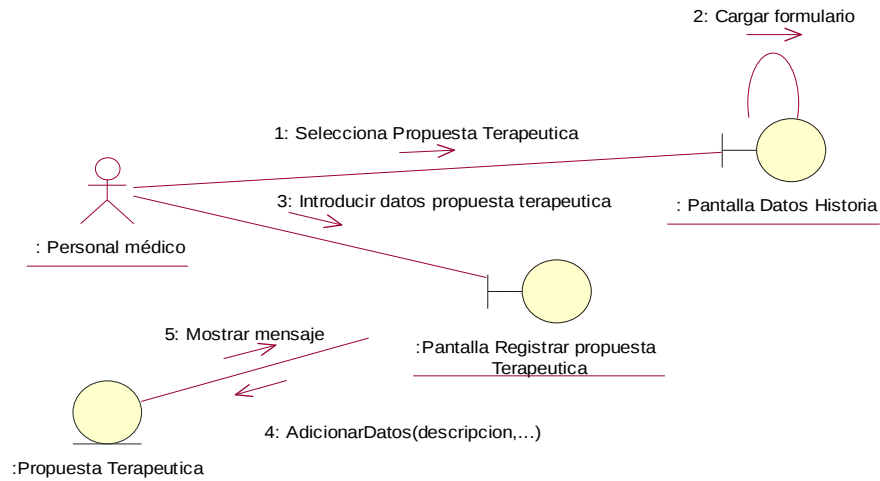


Figura 45. Diagrama de colaboración para el caso de uso registrar propuesta terapéutica

4.2.2.16 Diagrama de colaboración para el caso de uso registrar profesionales actuantes

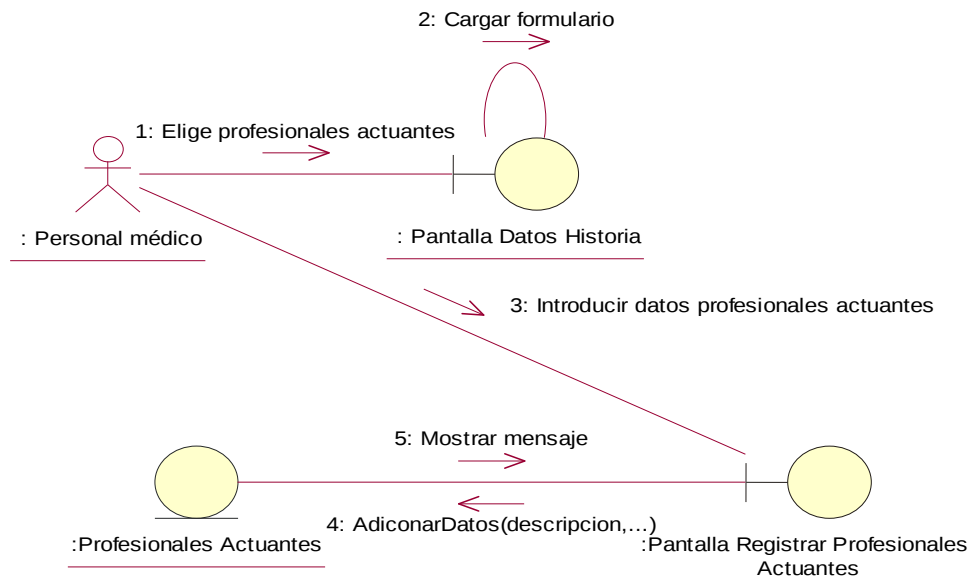


Figura 46. Diagrama de colaboración para el caso de uso registrar profesionales actuantes

4.2.2.17 Diagrama de colaboración para el caso de uso registrar consultar historia clínica

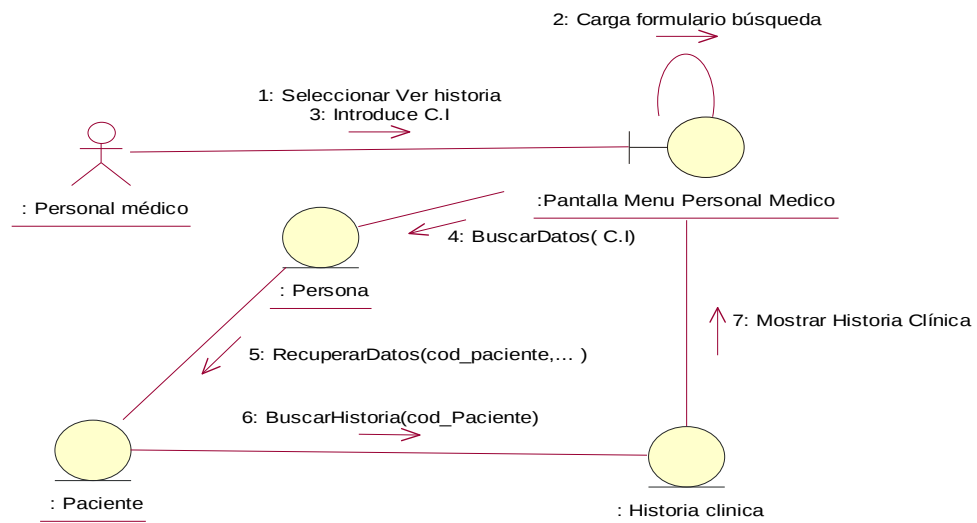


Figura 47. Diagrama de colaboración para el caso de uso consultar historia clínica

4.2.2.18 Diagrama de colaboración para el caso de uso registrar informe psicológico

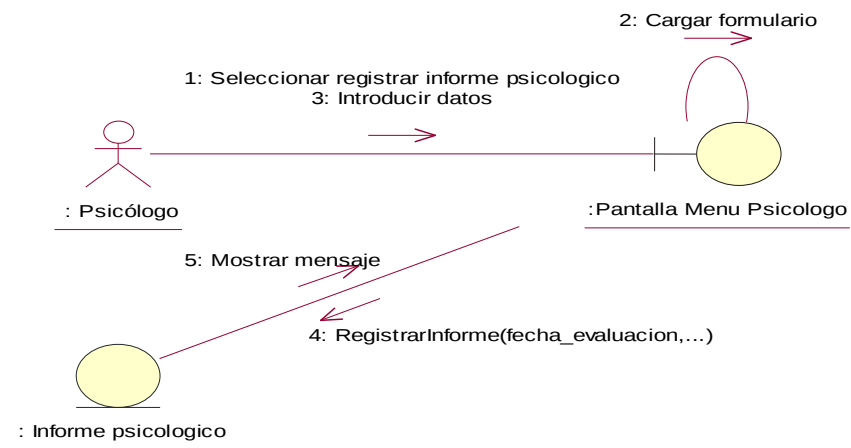


Figura 48. Diagrama de colaboración para el caso de uso registrar informe psicológico

4.2.2.19 Diagrama de colaboración para el caso de uso ver cantidad de pacientes

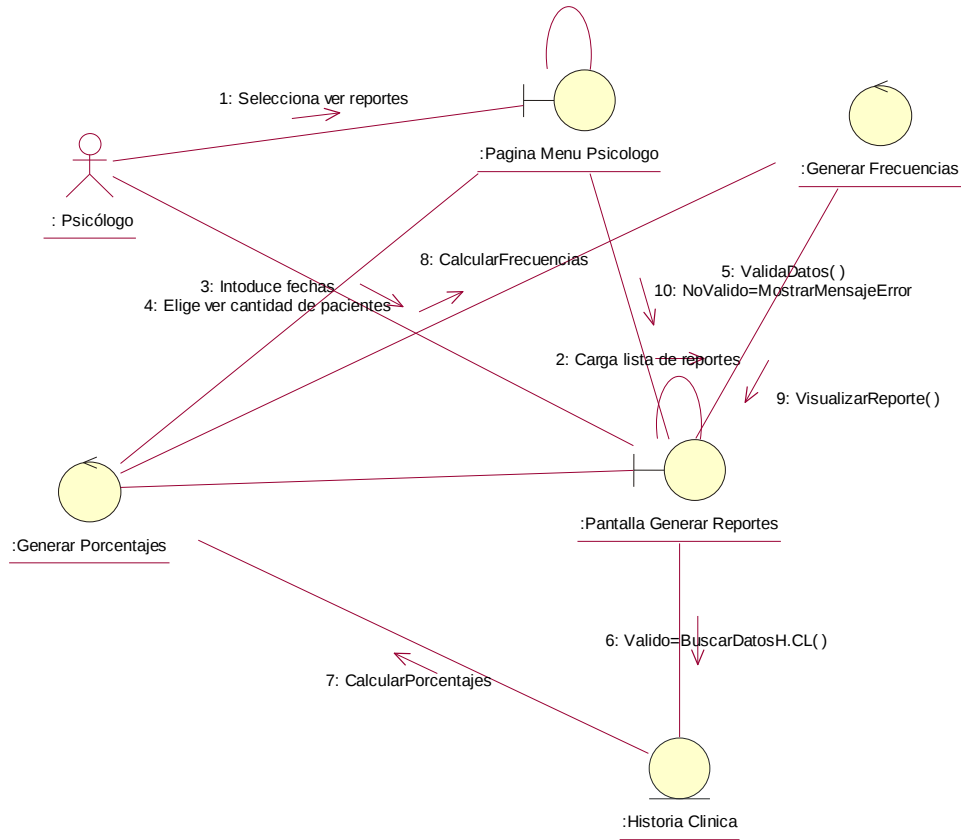


Figura 49. Diagrama de colaboración para el caso de uso ver cantidad de pacientes

4.2.2.20 Diagrama de colaboración para el caso de uso ver patologías frecuentes

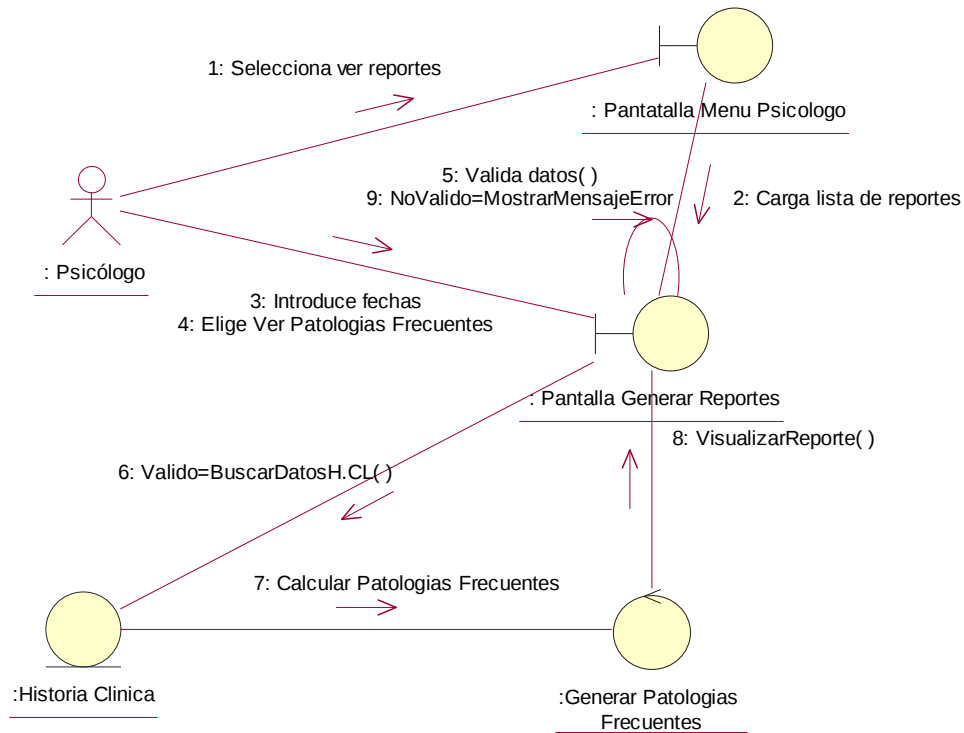


Figura 50. Diagrama de colaboración para el caso de uso ver patologías frecuentes

4.2.2.21 Diagrama de colaboración para el caso de uso ver consultas según sexo

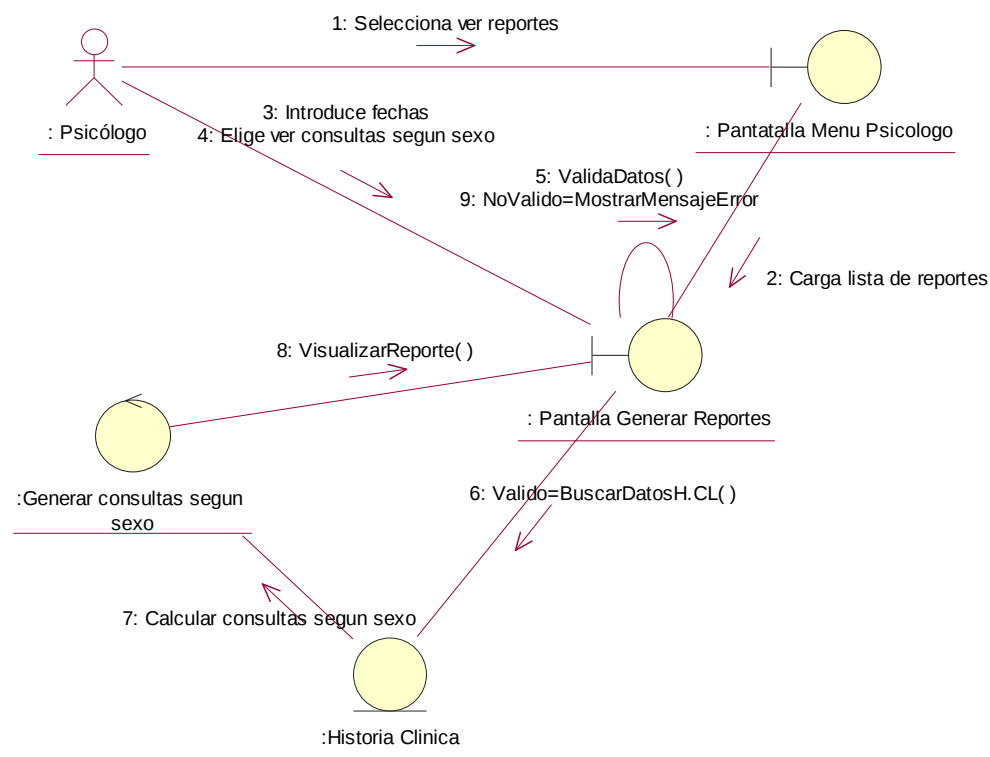


Figura 51. Diagrama de colaboración para el caso de uso ver consultas según
sexo

4.2.2.22 Diagrama de colaboración para el caso de uso ver consultas según edad

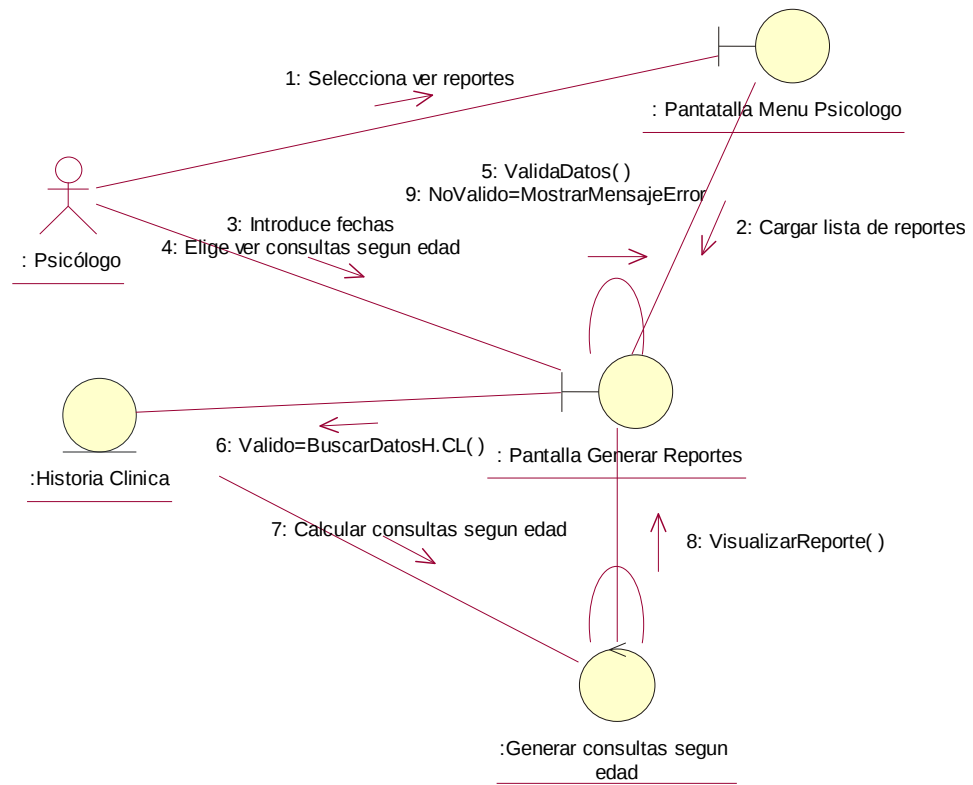


Figura 52. Diagrama de colaboración para el caso de uso ver consultas según edad

4.2.2.22 Diagrama de colaboración para el caso de uso administrar usuarios

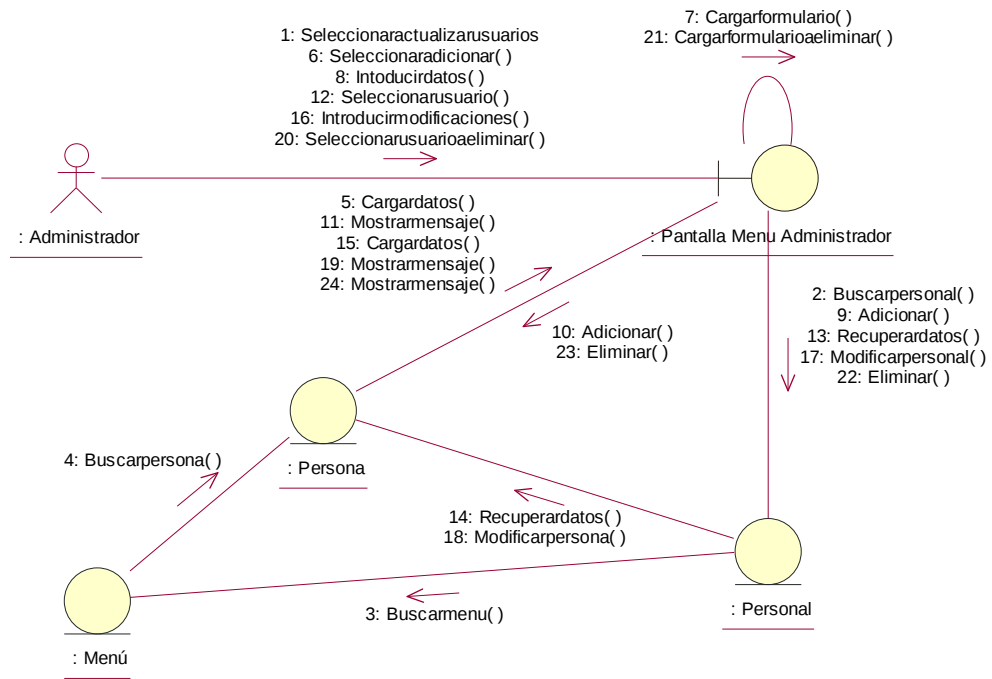


Figura 53. Diagrama de colaboración para el caso de uso administrar usuarios

4.2.2.23 Diagrama de colaboración para el caso de uso realizar copias de seguridad

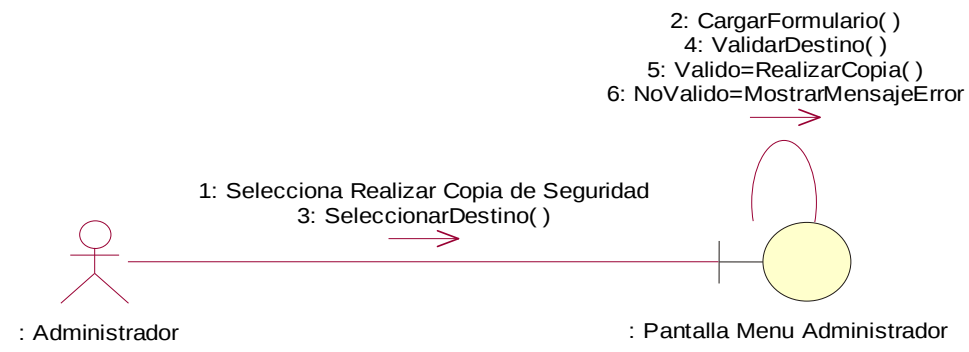


Figura 54. Diagrama de colaboración para el caso de uso realizar copias de seguridad

4.2.2.24 Diagrama de colaboración para el caso de uso restaurar base de datos

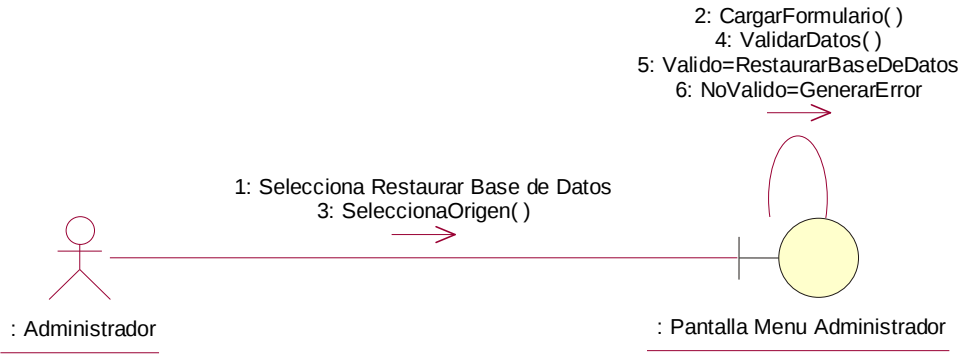


Figura 55. Diagrama de colaboración para el caso de uso restaurar base de datos

4.3 DIAGRAMAS DE ACTIVIDAD

4.3.1 Diagrama de actividad para el caso de uso ingresar usuario y contraseña

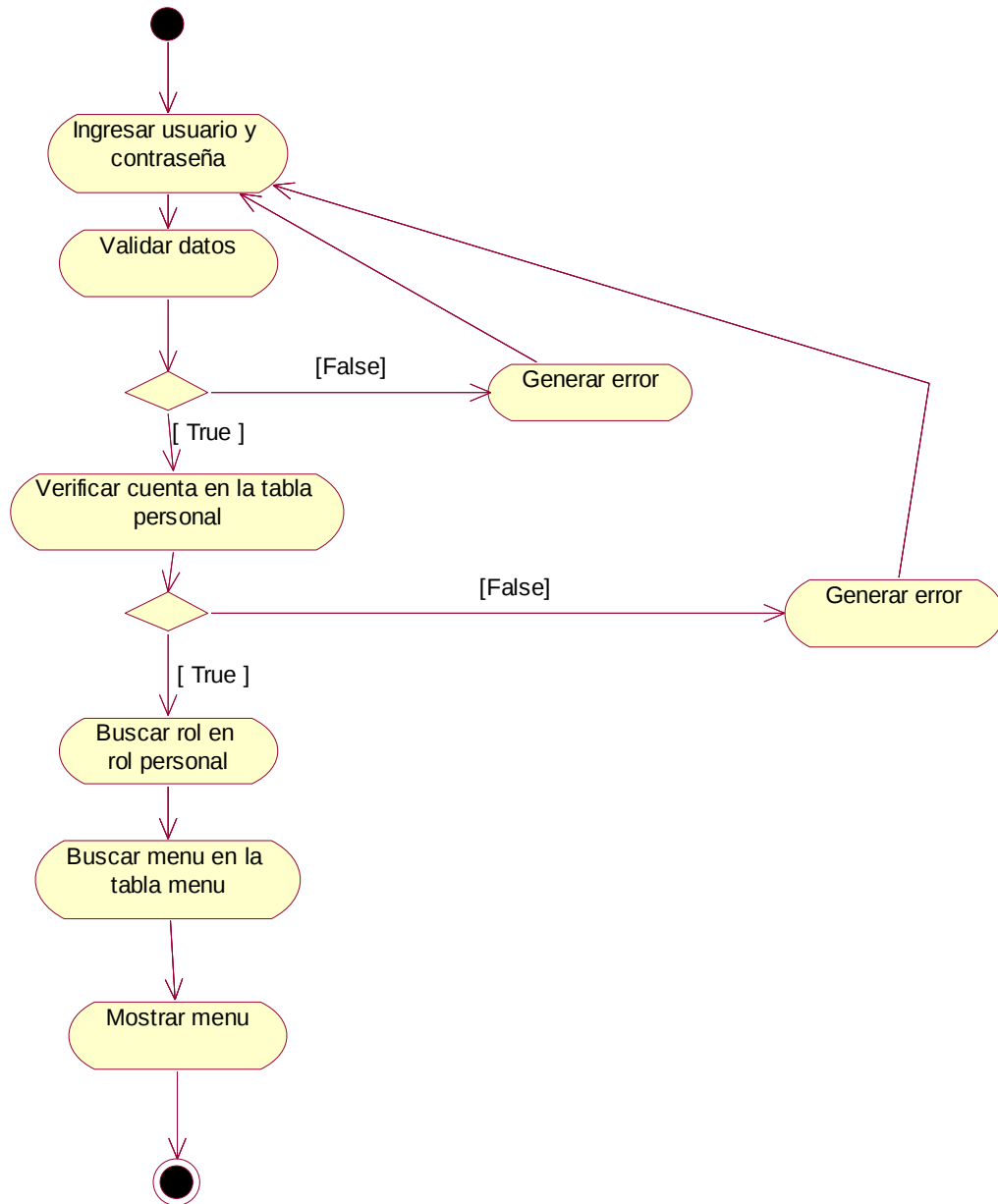


Figura 56. Diagrama de actividad para el caso de uso ingresar usuario y contraseña

4.3.2 Diagrama de actividad para el caso de uso administrar pacientes

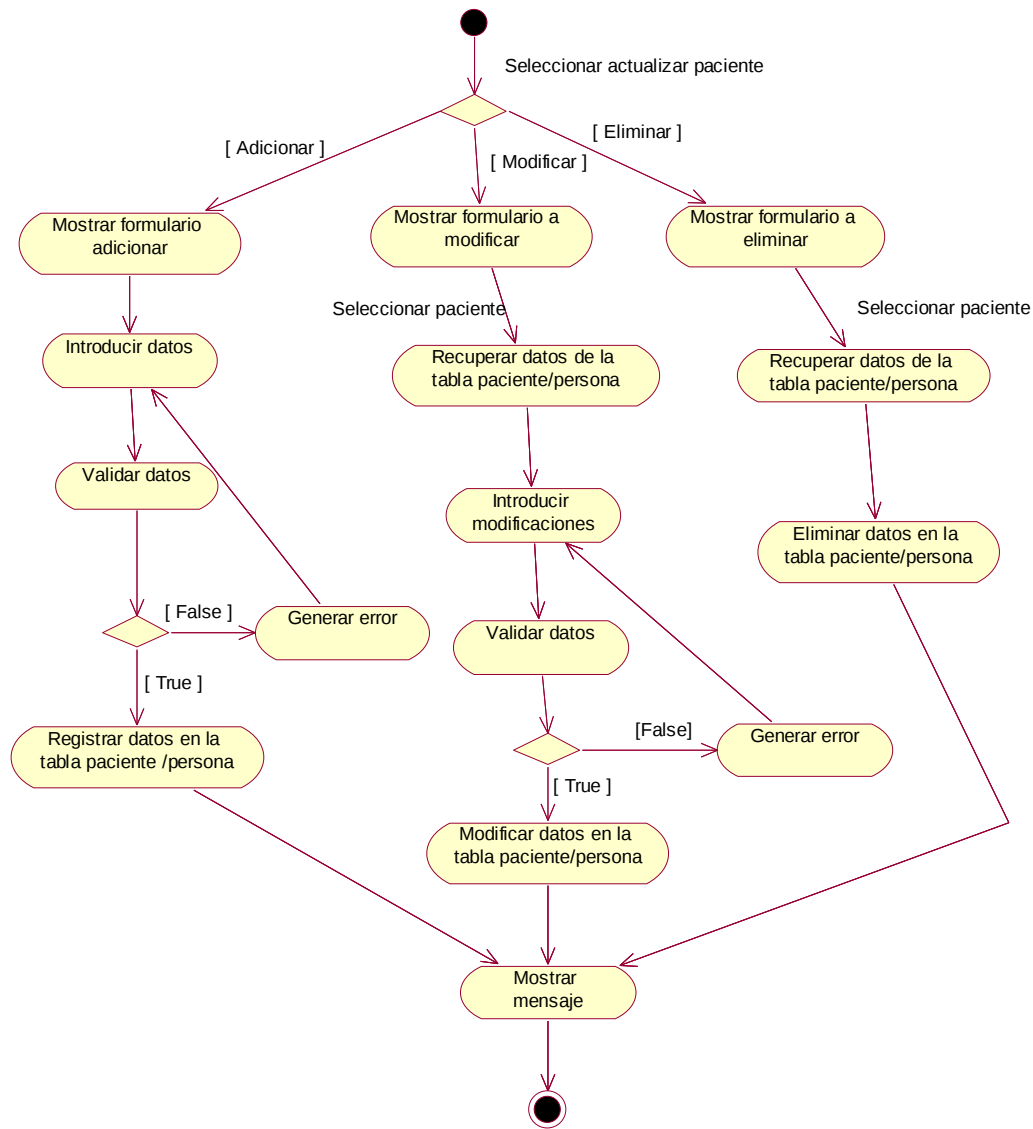


Figura 57. Diagrama de actividad para el caso de uso administrar pacientes

4.3.3 Diagrama de actividad para el caso de uso registrar comportamiento

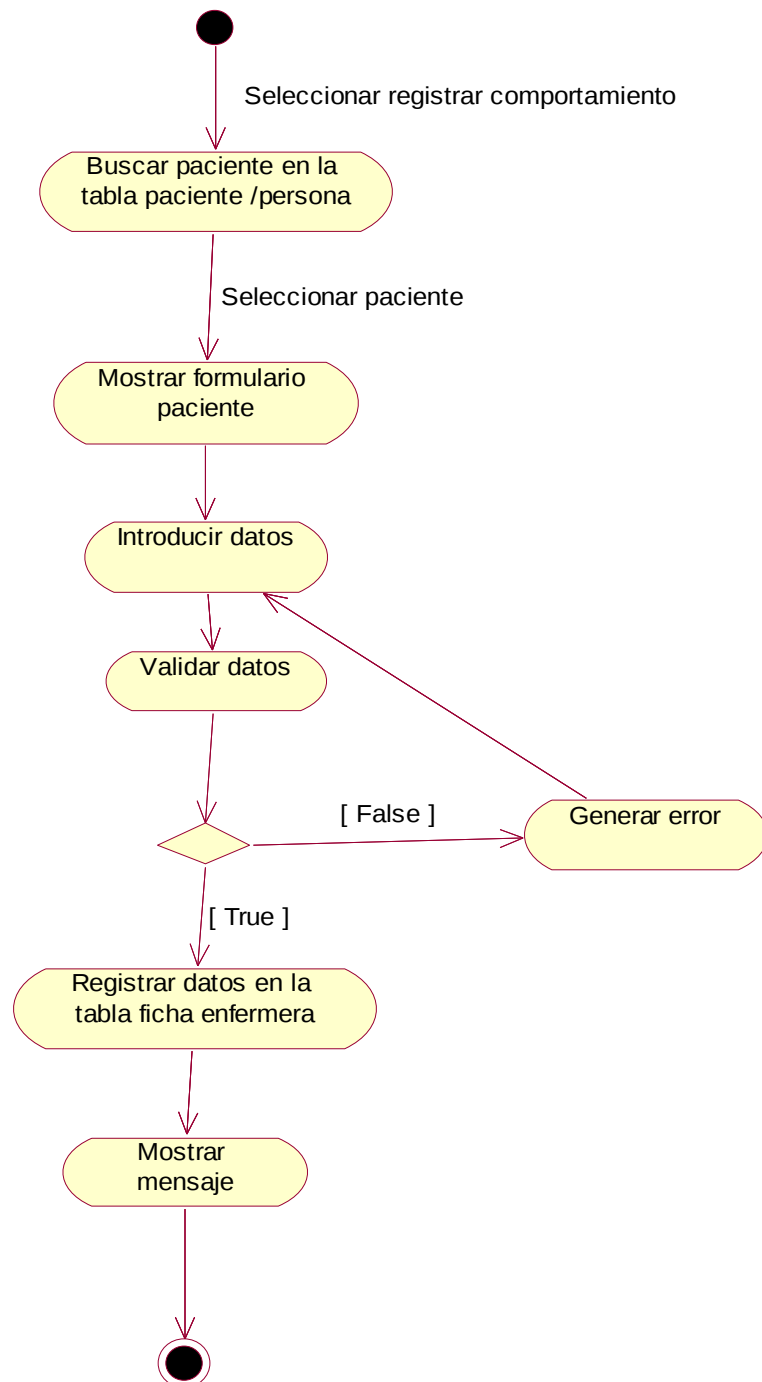


Figura 58. Diagrama de actividad para el caso de uso registrar comportamiento

4.3.4 Diagrama de actividad para el caso de uso administrar citas

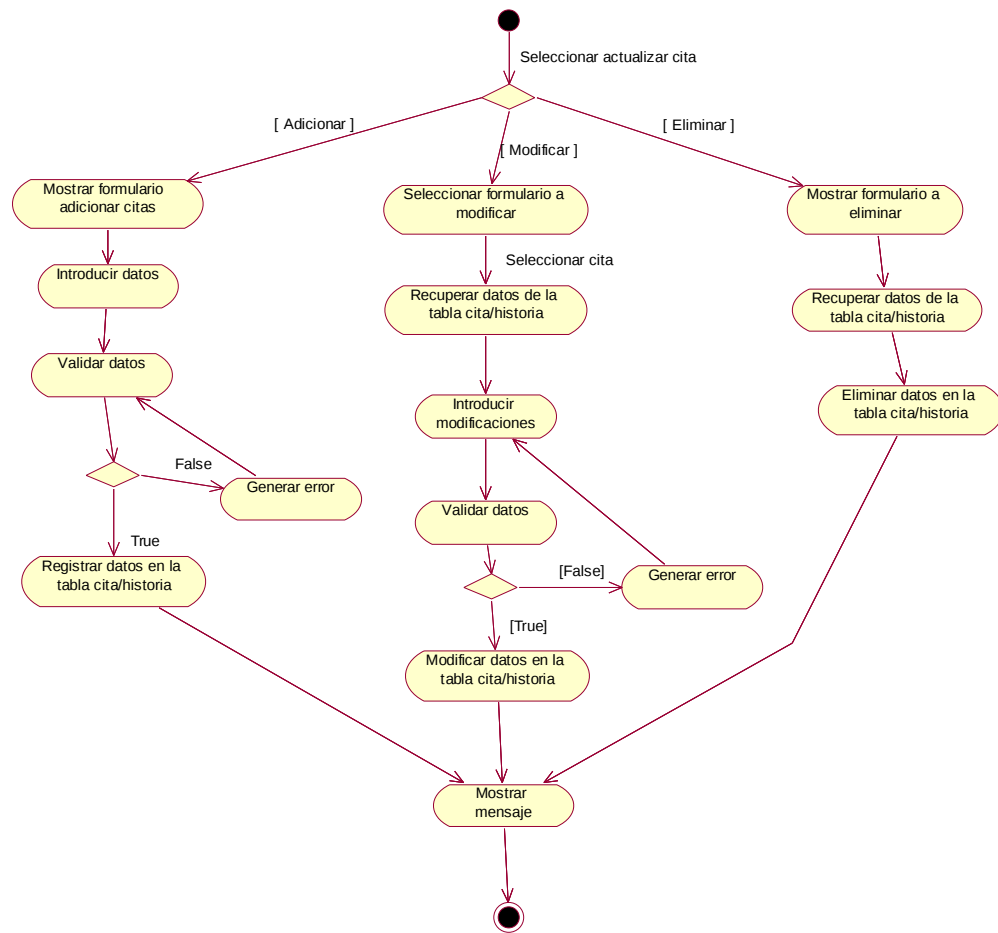


Figura 59. Diagrama de actividad para el caso de uso administrar citas

4.3.5 Diagrama de actividad para el caso de uso registrar historia clínica

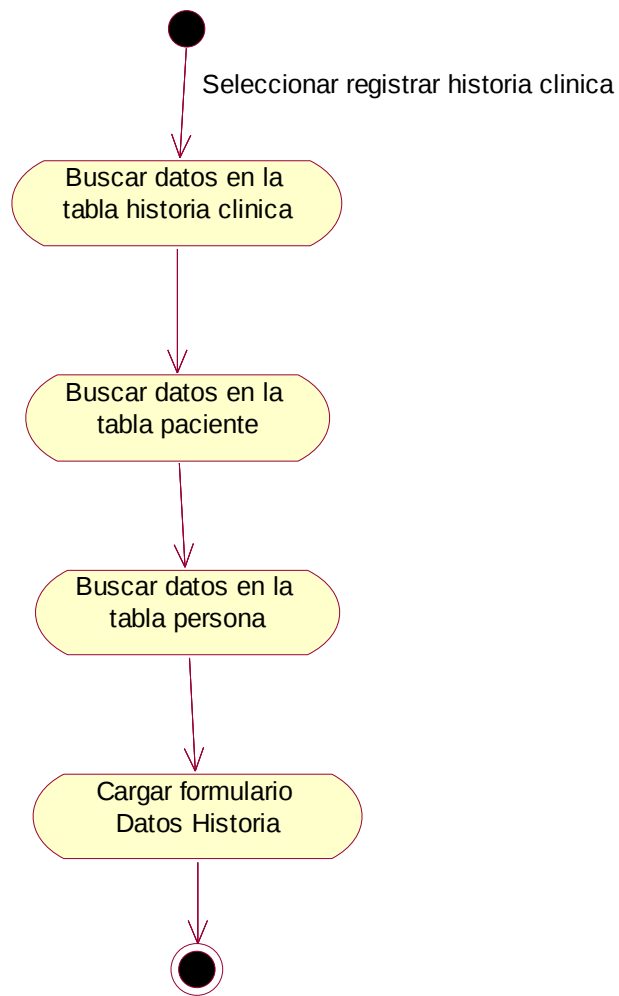


Figura 60. Diagrama de actividad para el caso de uso registrar historia clínica

4.3.6 Diagrama de actividad para el caso de uso registrar motivo de consulta y enfermedad actual

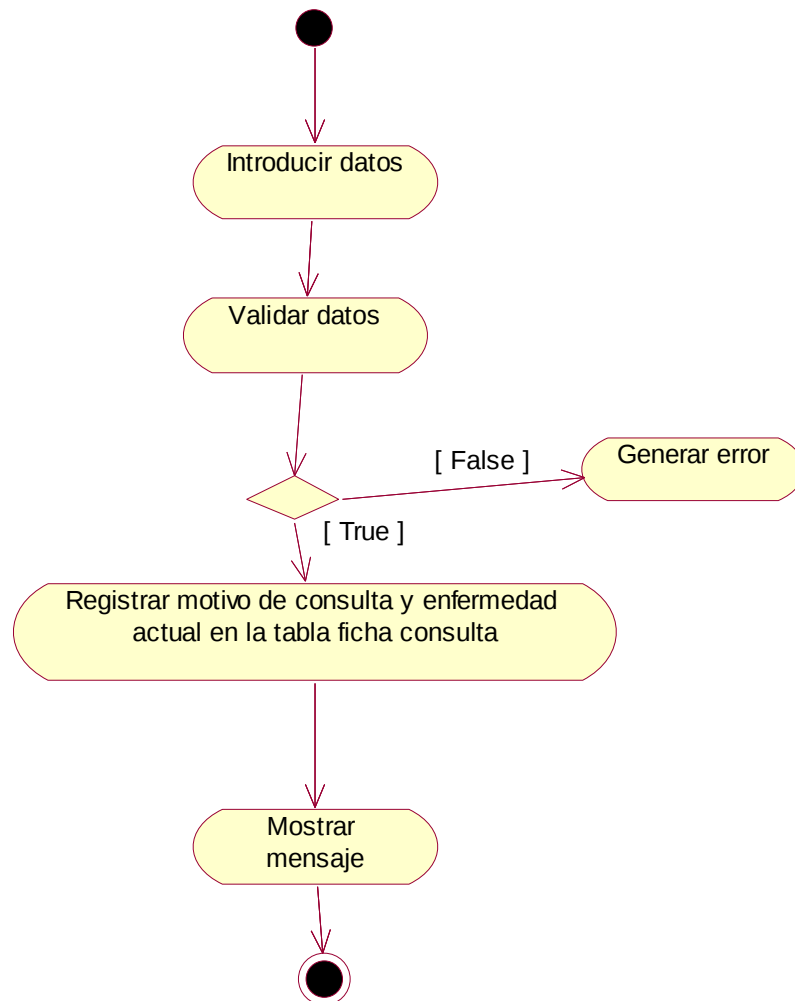


Figura 61. Diagrama de actividad para el caso de uso registrar motivo de consulta y enfermedad actual

4.3.7 Diagrama de actividad para el caso de uso registrar consumo de sustancias

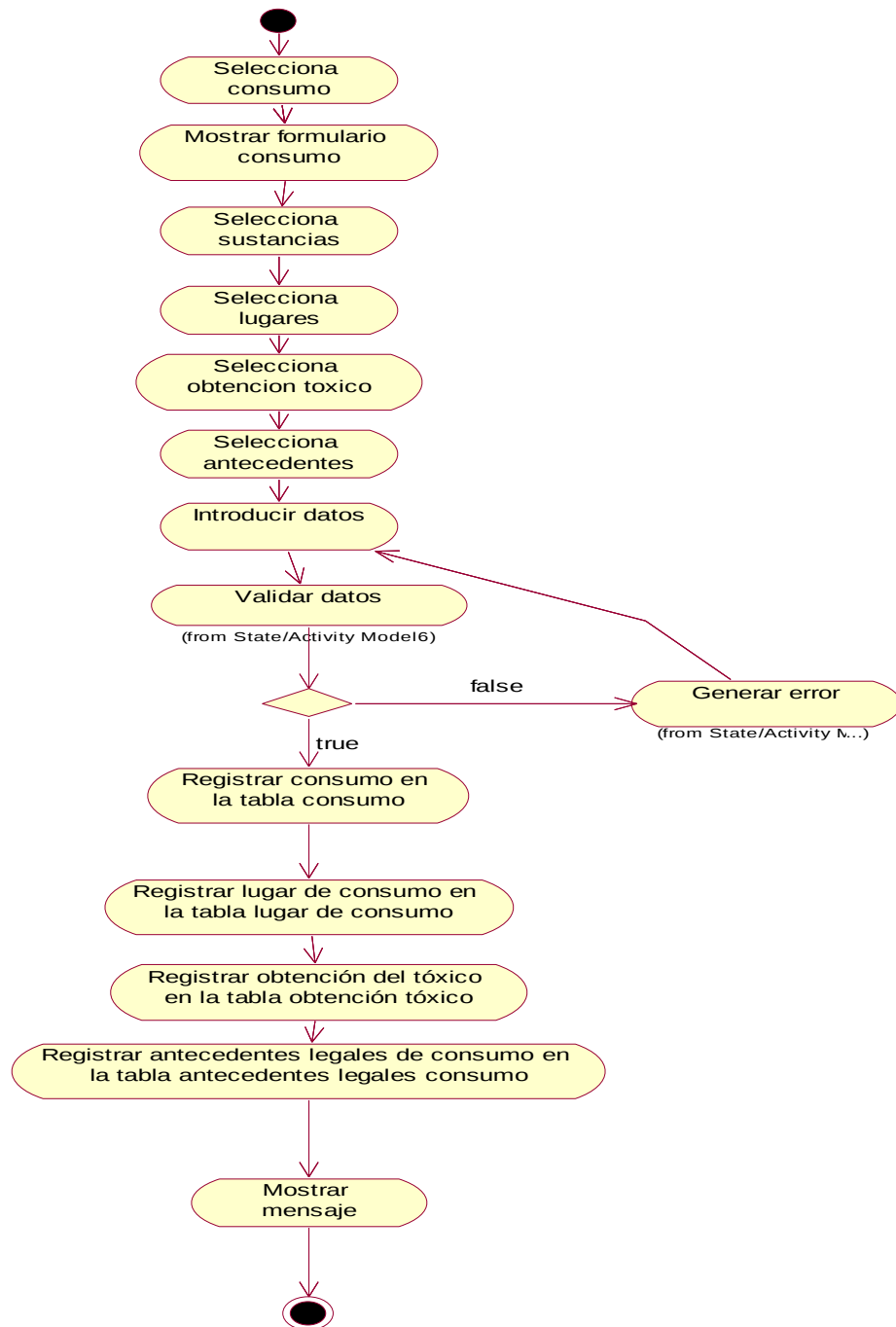


Figura 62. Diagrama de actividad para el caso de uso registrar consumo de sustancias

4.3.8 Diagrama de actividad para el caso de uso registrar área familiar

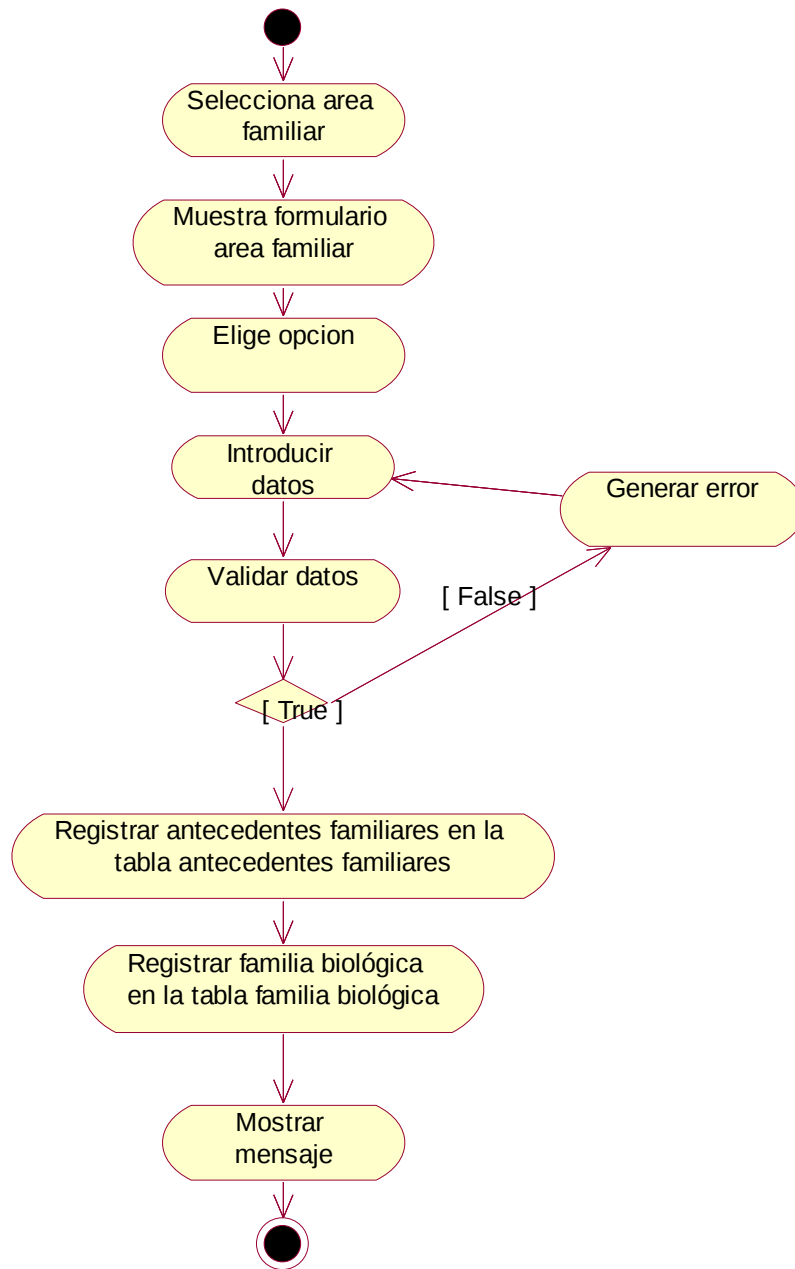


Figura 63. Diagrama de actividad para el caso de uso registrar área familiar

4.3.9 Diagrama de actividad para el caso de uso registrar tratamientos realizados

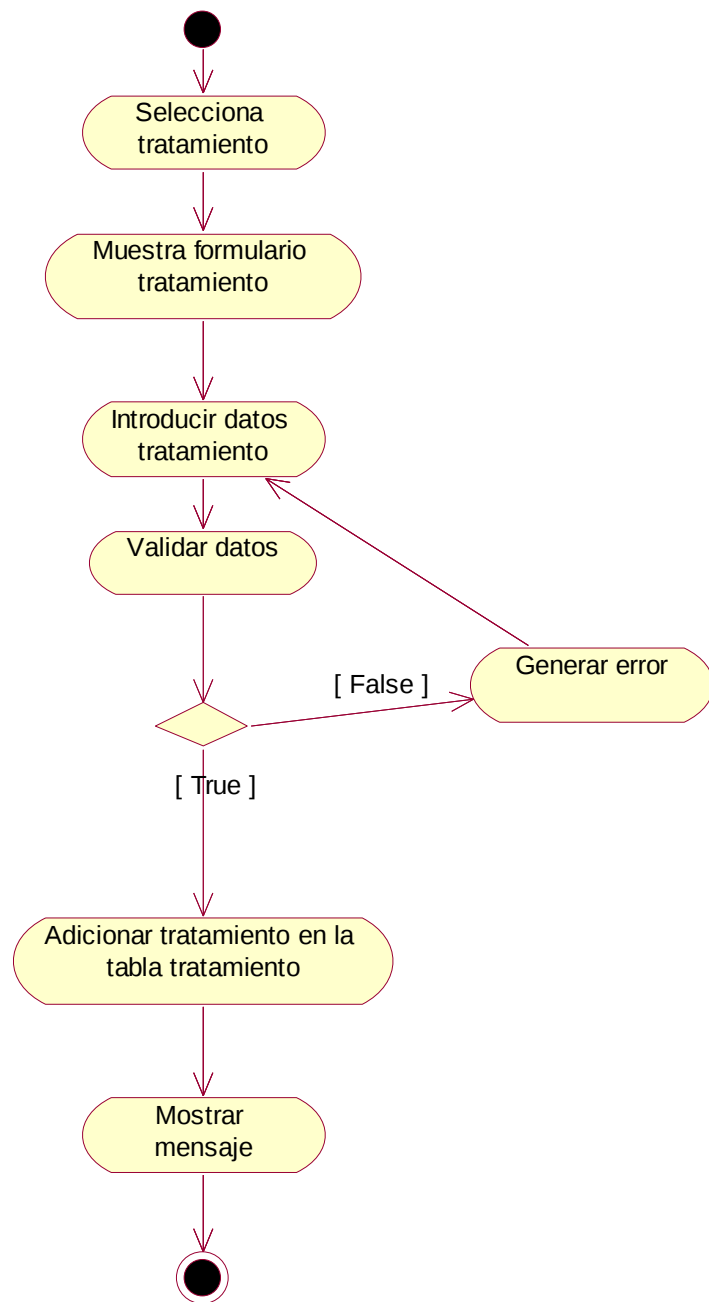


Figura 64. Diagrama de actividad para el caso de uso registrar tratamientos realizados

4.3.10 Diagrama de actividad para el caso de uso registrar antecedentes historia clínica interna

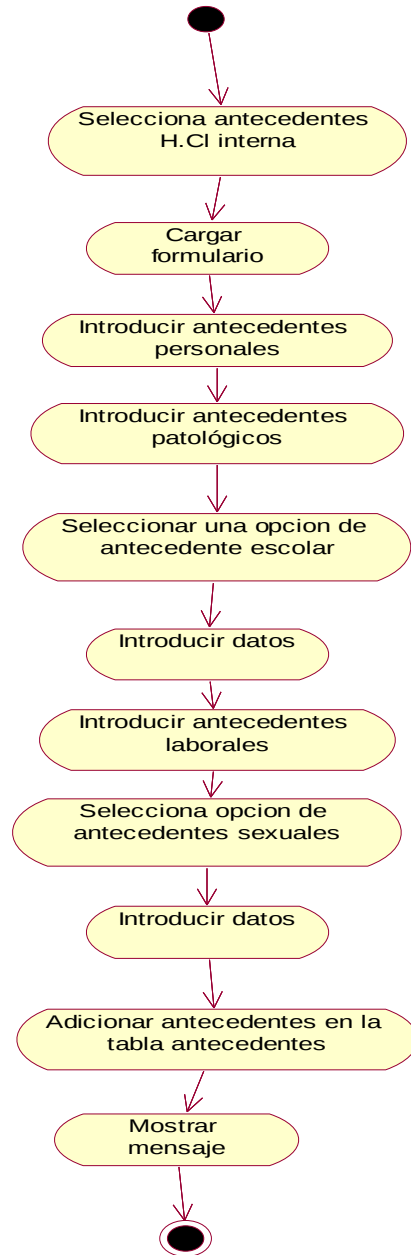


Figura 65. Diagrama de actividad para el caso de uso registrar antecedentes historia clínica interna

4.3.11 Diagrama de actividad para el caso de uso registrar antecedentes historia clínica externa

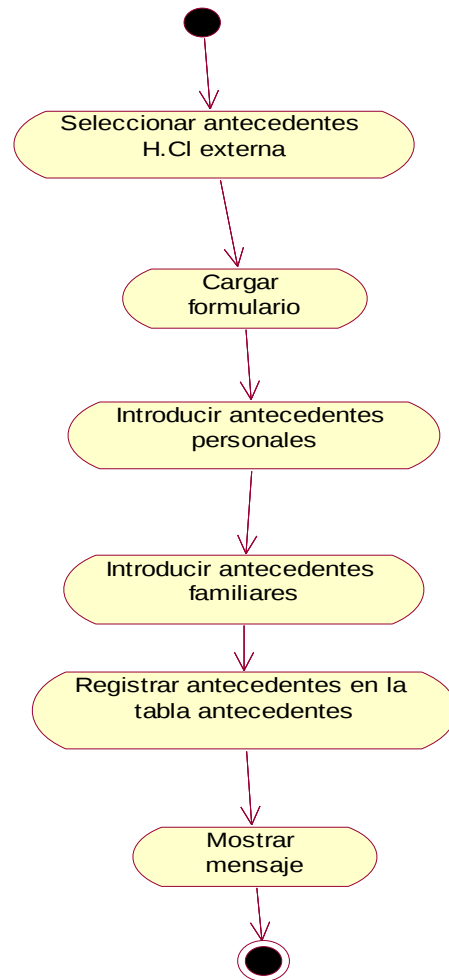


Figura 66. Diagrama de actividad para el caso de uso registrar antecedentes historia clínica externa

4.3.12 Diagrama de actividad para el caso de uso registrar exámenes

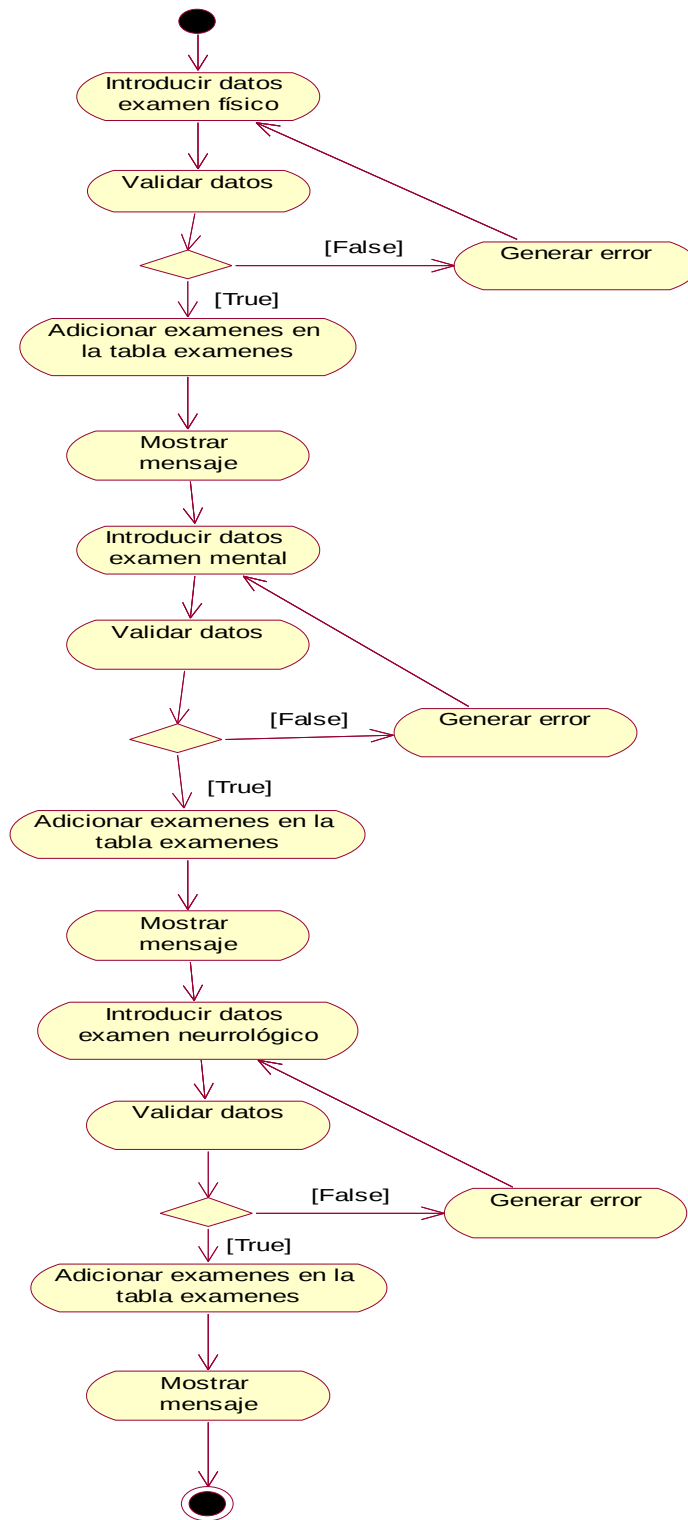


Figura 67. Diagrama de actividad para el caso de uso registrar exámenes

4.3.13 Diagrama de actividad para el caso de uso registrar diagnóstico

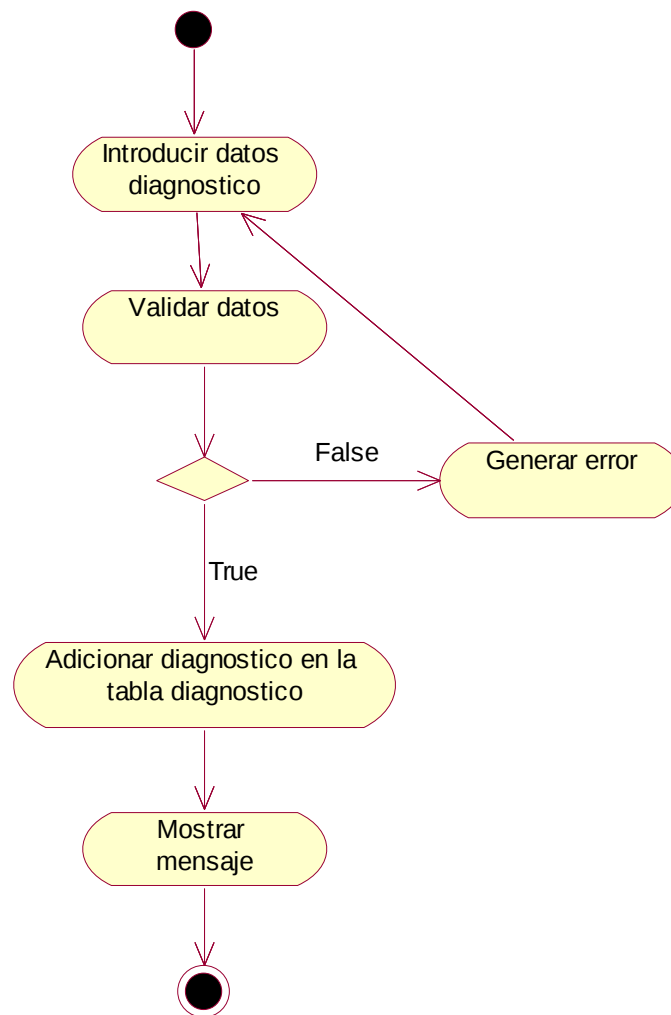


Figura 68. Diagrama de actividad para el caso de uso registrar diagnóstico

4.3.14 Diagrama de actividad para el caso de uso registrar evolución y resumen

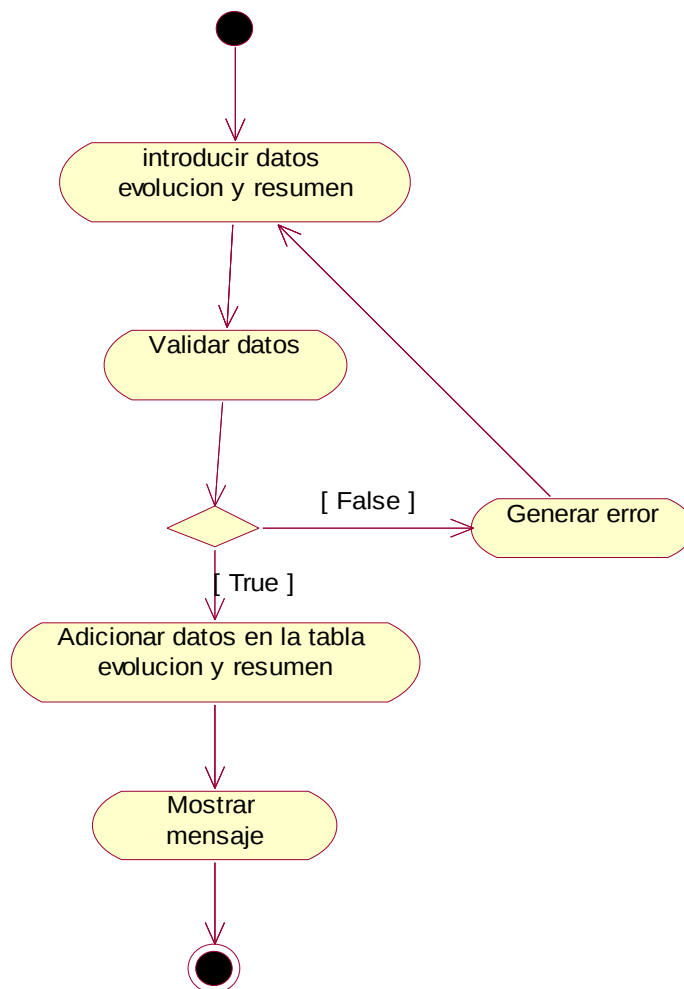


Figura 69. Diagrama de actividad para el caso de uso registrar evolución y resumen

4.3.15 Diagrama de actividad para el caso de uso registrar propuesta terapéutica

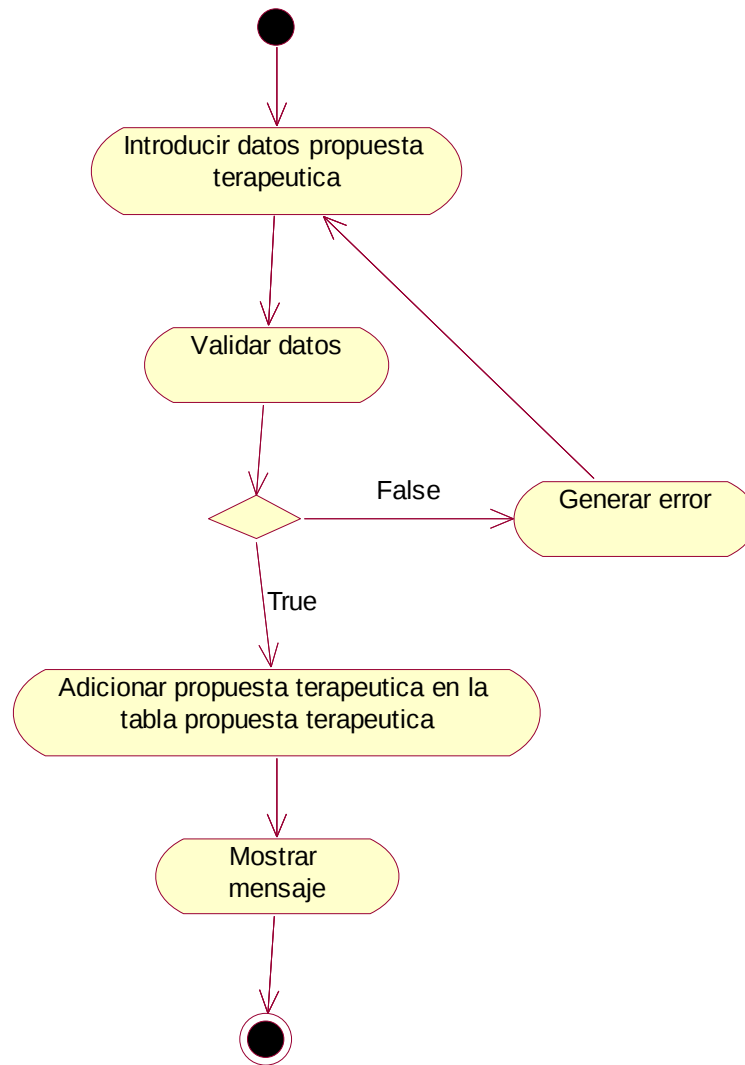


Figura 70. Diagrama de actividad para el caso de uso registrar propuesta terapéutica

4.3.16 Diagrama de actividad para el caso de uso registrar profesionales Actuantes

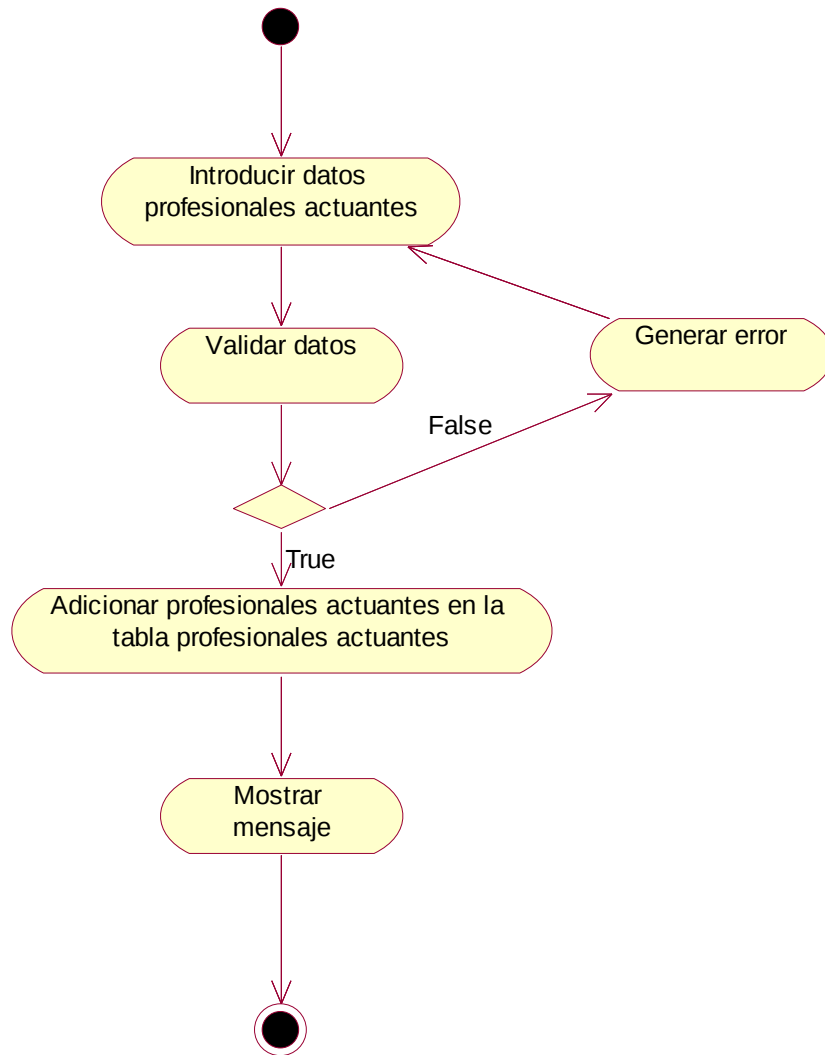


Figura 71. Diagrama de actividad para el caso de uso registrar profesionales actuantes

4.3.17 Diagrama de actividad para el caso de uso consultar historia clínica

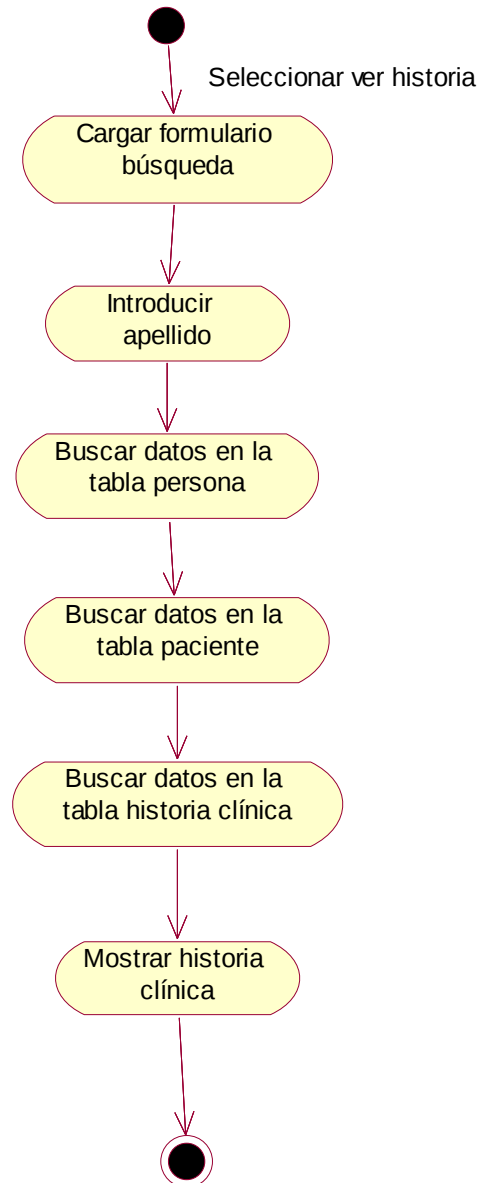


Figura 72. Diagrama de actividad para el caso de uso consultar historia clínica

4.3.18 Diagrama de actividad para el caso de uso registrar informe psicológico

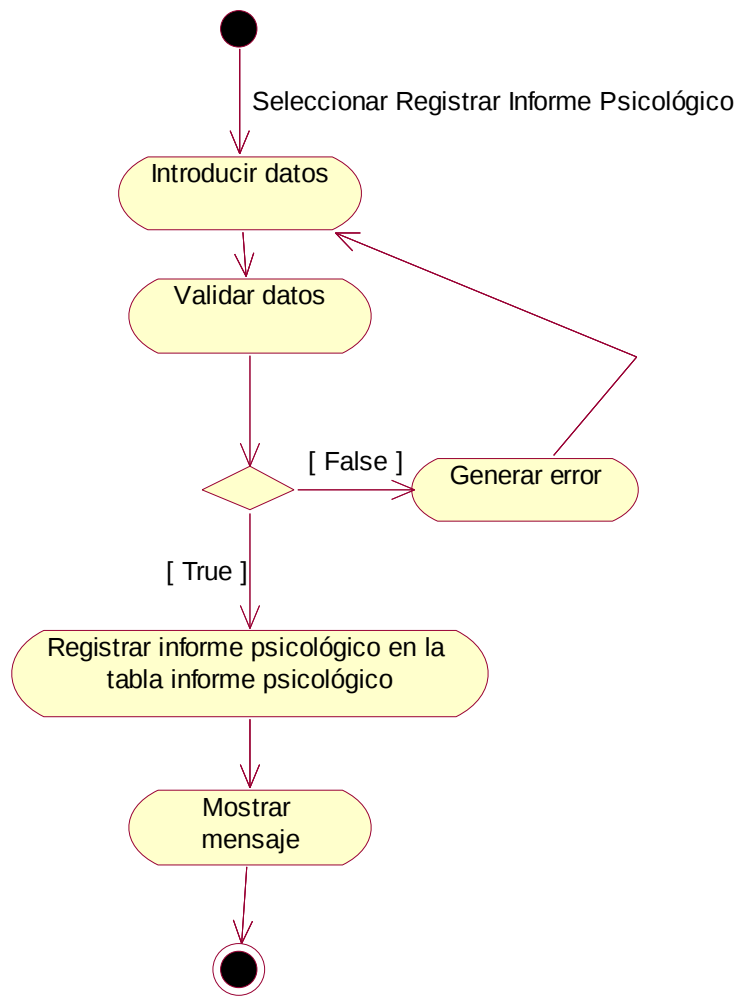


Figura 73. Diagrama de actividad para el caso de uso registrar informe psicológico

4.3.19 Diagrama de actividad para el caso de uso ver cantidad de pacientes

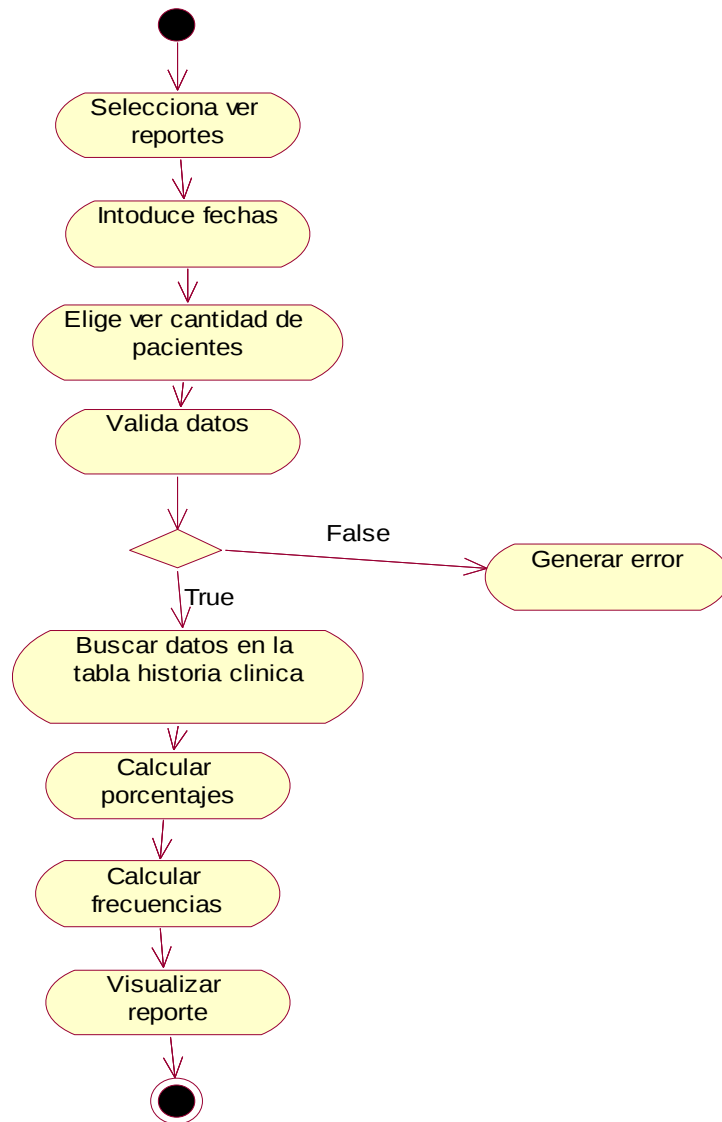


Figura 74. Diagrama de actividad para el caso de uso ver cantidad de pacientes

4.3.20 Diagrama de actividad para el caso de uso ver patologías frecuentes

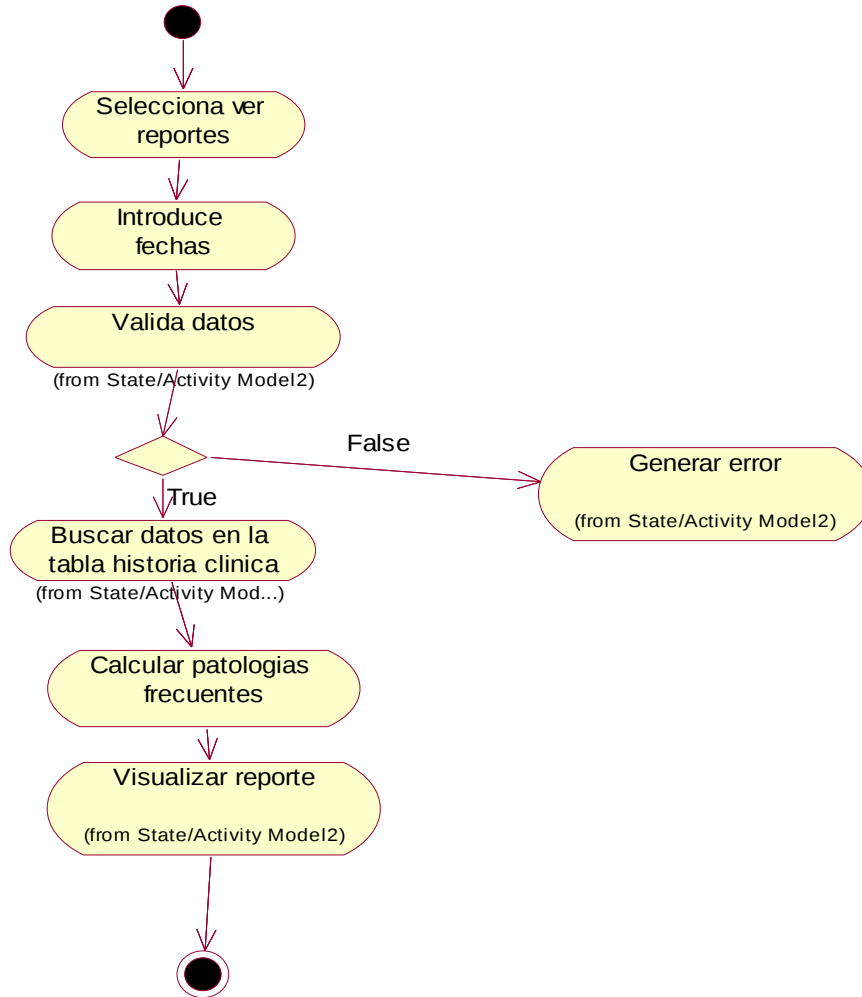


Figura 75. Diagrama de actividad para el caso de uso ver patologías frecuentes

4.3.21 Diagrama de actividad para el caso de uso ver consultas según sexo

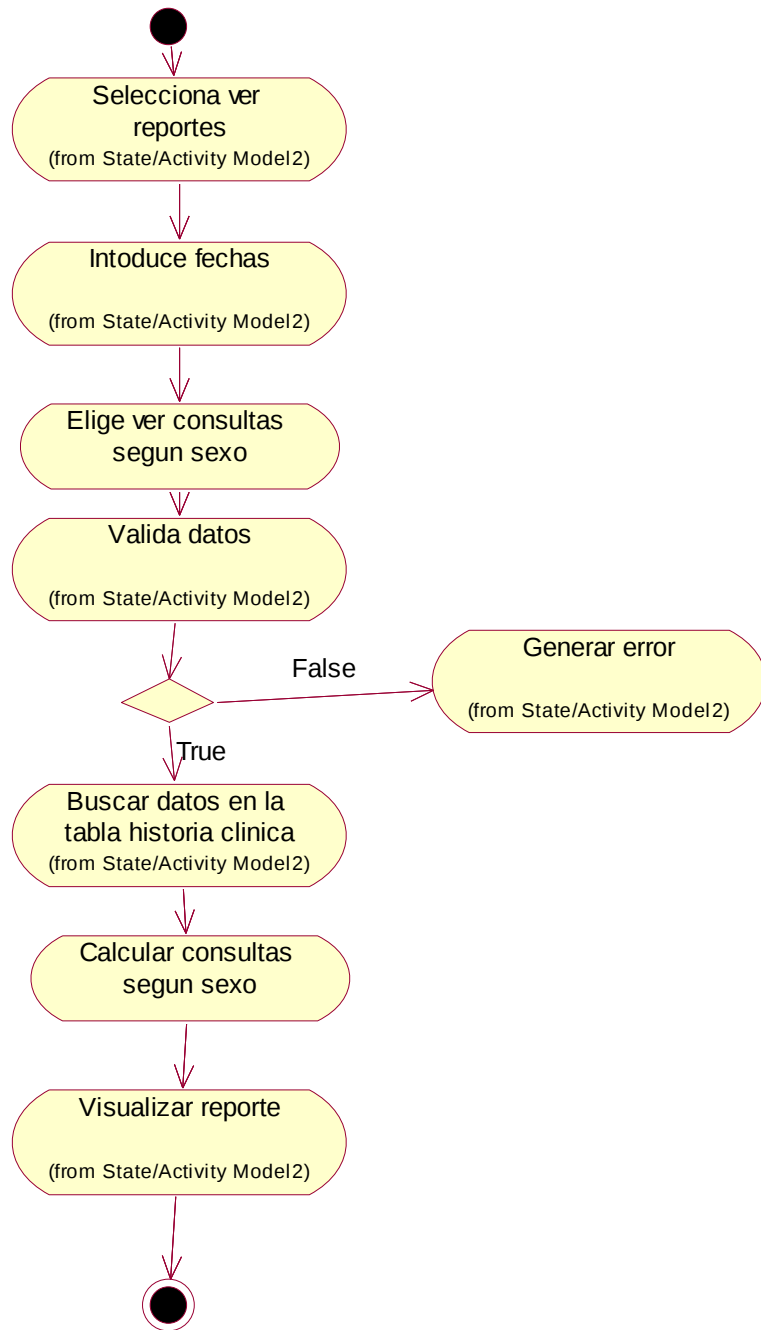


Figura 76. Diagrama de actividad para el caso de uso ver consultas según sexo

4.3.22 Diagrama de actividad para el caso de uso ver consultas según edad

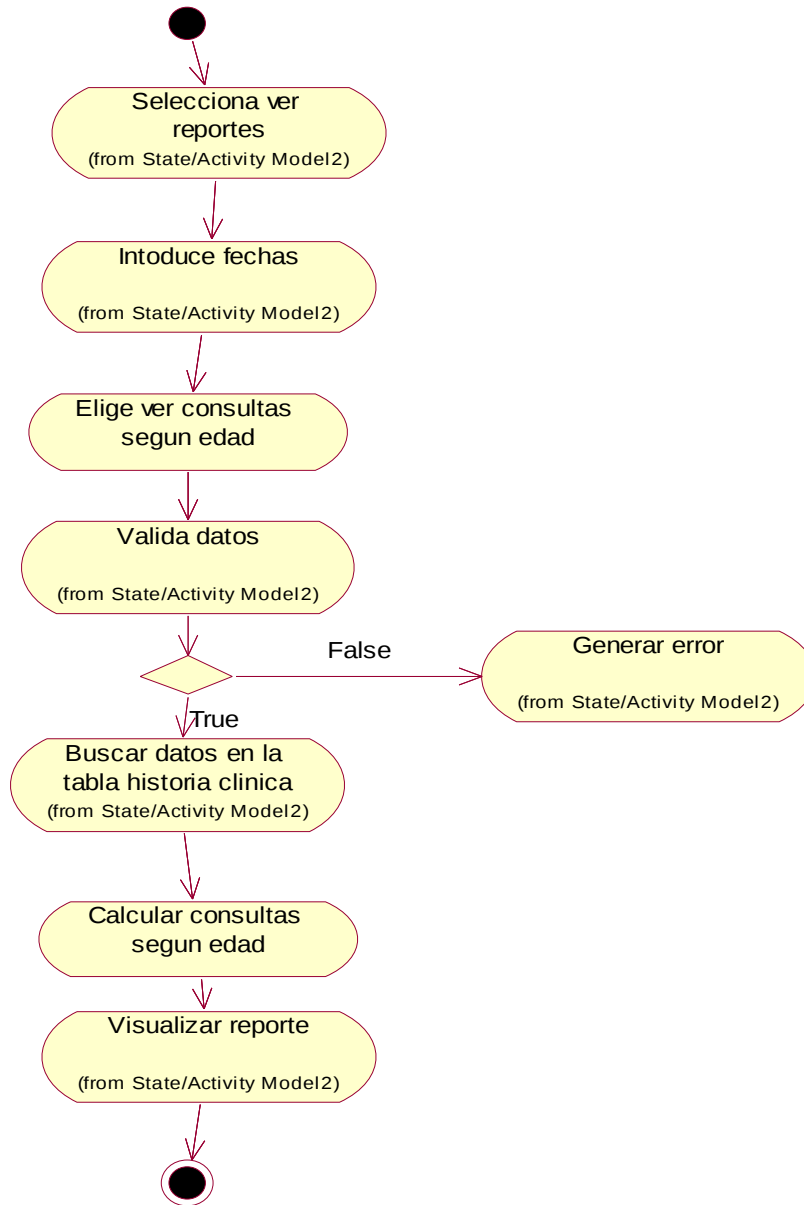


Figura 77. Diagrama de actividad para el caso de uso ver consultas según edad

4.3.23 Diagrama de actividad para el caso de uso administrar usuarios

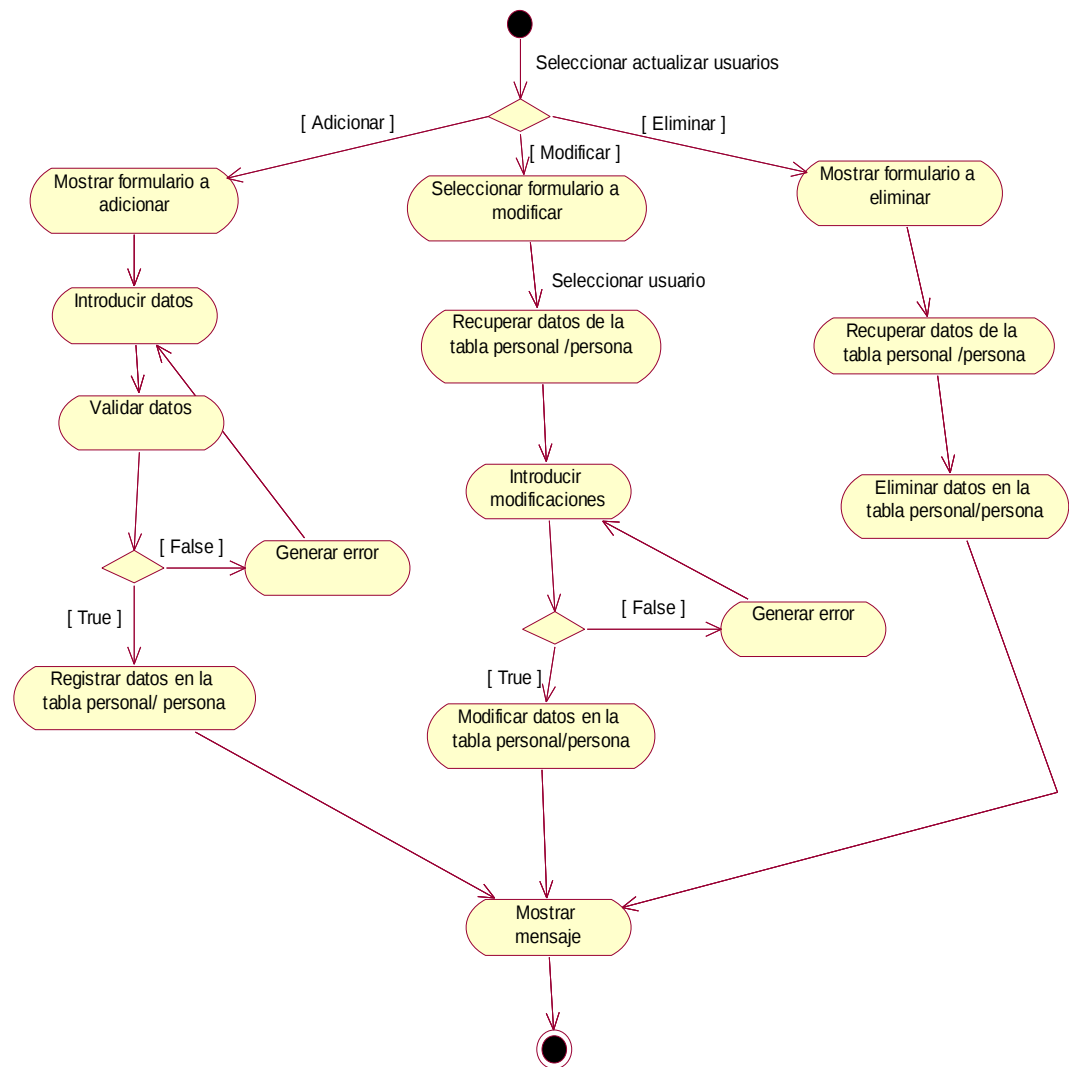


Figura 78. Diagrama de actividad para el caso de uso administrar usuarios

El principal resultado del diseño es el modelo de diseño que se esfuerza en conservar la estructura del sistema impuesta por el modelo de análisis y que sirve como esquema para la implementación.

5.1 DIAGRAMA DE CLASES

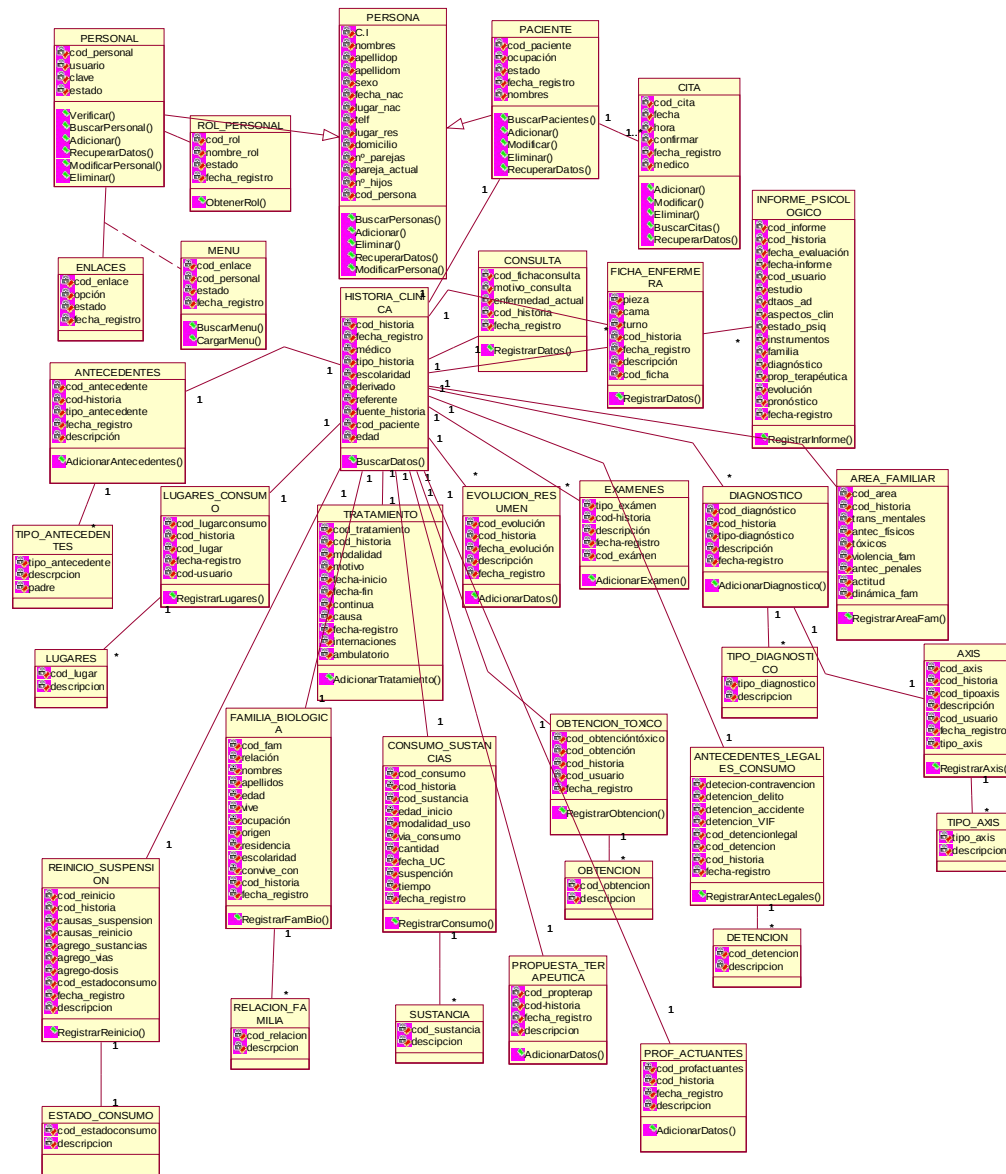


Figura 79. Diagrama de clases del sistema completo

5.2 DESCRIPCIÓN DE CLASES

Nombre de la clase: ANTECEDENTES	
Descripción: contiene datos de los antecedentes del paciente	
Atributos:	
Cod_antecedentes	Integer
Cod_historia	Integer
Tipo_antecedente	Integer
Fecha_registro	Date
Operaciones:	
AdicionarAntecedentes()	

Nombre de la clase: AREA_FAMILIAR	
Descripción: contiene datos de algunos antecedentes de la familia del paciente	
Atributos:	
Cod_area	Integer
Cod_historia	integer
Trans_mentales	Varchar(100)
Antec_físicos	Varchar(100)
Tóxicos	Varchar(100)
Violencia_familiar	Varchar(100)
Antec_penales	Varchar(100)
Actitud	Varchar(5)
Dinámica_fam	Text
Operaciones:	
RegistrarAreaFam()	

Nombre de la clase: ANTECEDENTES_ LEGALES_CONSUMO	
Descripción: guarda datos de la detención legal del paciente	
Atributos:	
Cod_anteclegal	Integer
Cod_detención	Integer
Cod_historia	Integer
Detencion_contravencion	Varchar(5)
Detencion_delito	Varchar(5)
Detencion_accidente	Varchar(5)
Detencion_VIF	Varchar(5)
Fecha_registro	Date
Operaciones:	
RegistrarAntecLegales()	

Nombre de la clase: CITA	
Descripción: contiene datos de la asignación de citas a los pacientes	
Atributos	
Cod_cita	Integer
Fecha	Date
Hora	Time
Confirmar	Varchar (50)
Fecha_registro	Date
Cod_paciente	Integer
Operaciones:	
Adicionar()	
Modificar()	
Eliminar()	
BuscarCitas()	

RecuperarDatos()											
Nombre de la clase: CONSULTA											
Descripción: contiene datos acerca de la consulta del paciente											
Atributos: <table> <tr> <td>Cod_consulta</td><td>Integer</td></tr> <tr> <td>Motivo_consulta</td><td>Varchar(500)</td></tr> <tr> <td>Enfermedad_actual</td><td>Varchar(500)</td></tr> <tr> <td>Cod_historia</td><td>Integer</td></tr> <tr> <td>Fecha_registro</td><td>Date</td></tr> </table>		Cod_consulta	Integer	Motivo_consulta	Varchar(500)	Enfermedad_actual	Varchar(500)	Cod_historia	Integer	Fecha_registro	Date
Cod_consulta	Integer										
Motivo_consulta	Varchar(500)										
Enfermedad_actual	Varchar(500)										
Cod_historia	Integer										
Fecha_registro	Date										
Operaciones: RegistrarDatos()											

Nombre de la clase: CONSUMO_SUSTANCIAS																							
Descripción: contiene datos del consumo de sustancias del paciente																							
Atributos <table> <tr> <td>Cod_consumo</td><td>Integer</td></tr> <tr> <td>Cod_historia</td><td>Integer</td></tr> <tr> <td>Cod_sustancia</td><td>Integer</td></tr> <tr> <td>Edad_inicio</td><td>Integer</td></tr> <tr> <td>Modalidad_uso</td><td>Varchar(50)</td></tr> <tr> <td>Via_consumo</td><td>Varchar(50)</td></tr> <tr> <td>Cantidad</td><td>Varchar(50)</td></tr> <tr> <td>Fecha_UC</td><td>Date</td></tr> <tr> <td>Suspension</td><td>Varchar (50)</td></tr> <tr> <td>Tiempo</td><td>Varchar (50)</td></tr> <tr> <td>Fecha_registro</td><td>Date</td></tr> </table>		Cod_consumo	Integer	Cod_historia	Integer	Cod_sustancia	Integer	Edad_inicio	Integer	Modalidad_uso	Varchar(50)	Via_consumo	Varchar(50)	Cantidad	Varchar(50)	Fecha_UC	Date	Suspension	Varchar (50)	Tiempo	Varchar (50)	Fecha_registro	Date
Cod_consumo	Integer																						
Cod_historia	Integer																						
Cod_sustancia	Integer																						
Edad_inicio	Integer																						
Modalidad_uso	Varchar(50)																						
Via_consumo	Varchar(50)																						
Cantidad	Varchar(50)																						
Fecha_UC	Date																						
Suspension	Varchar (50)																						
Tiempo	Varchar (50)																						
Fecha_registro	Date																						
Operaciones: RegistrarConsumo()																							

Nombre de la clase: DIAGNOSTICO	
Descripción: contiene datos del diagnóstico del paciente	
Atributos:	
Cod_diagnóstico	Integer
Cod_historia	Integer
Tipo_diagnóstico	Varchar(50)
Descripción	Text
Fecha_registro	Date
Operaciones:	
AdicionarDiagnostico()	

Nombre de la clase: ENLACES	
Descripción:	
Atributos:	
Cod_enlace	Integer
Opción	Varchar (200)
Estado	Char (1)
Fecha_registro	Date
Operaciones:	

Nombre de la clase: EVOLUCION_RESUMEN	
Descripción: contiene datos del seguimiento que se le hace al tratamiento	
Atributos:	
Cod_evolución	Integer
Cod_historia	Integer
Fecha_evolución	Date
Descripción	Text
Fecha_registro	Date
Operaciones:	
AdicionarDatos()	

Nombre de la clase: EXAMENES	
Descripción: almacena datos sobre los exámenes a los que es sometido el paciente	
Atributos:	
Cod_examen	Integer
Cod_historia	Integer
Descripción	Text
Fecha_registro	Date
Tipo_examen	Varchar(50)
Operaciones:	
AdicionarExamen()	

Nombre de la clase: FICHA_ENFERMERA	
Descripción: almacena los datos del paciente internado que son útiles para la enfermera	
Atributos:	
Pieza	Integer
Cama	Integer
Turno	Varchar(20)
Cod_historia	Integer
Fecha_registro	Date
Descripción	Text
Operaciones:	
Registrardatos()	

Nombre de la clase: FAMILIA_BIOLOGICA	
Descripción: contiene datos de la familia biológica del paciente	
Atributos:	
Cod_fam	Integer
Relación	Varchar(50)
Nombres	Varchar(50)
Apellidos	Varchar(50)
Edad	Integer
Vive	Varchar(5)
Ocupación	Varchar(20)
Origen	Varchar(50)
Residencia	Varchar(50)
Escolaridad	Varchar(50)
Convive_con	Text
Cod_historia	Integer
Fecha-registro	Date
Operaciones:	
RegistrarFamBio()	

Nombre de la clase: HISTORIA_CLINICA	
Descripción: contiene información clínica de los pacientes, está se crea una sola	
Atributos:	
Cod_historia	Numérico
Fecha_registro	Date
Médico	Varchar (100)
Tipo_historia	Varchar (20)
Escolaridad	Varchar (50)
Derivado	Varchar (100)
Fuente_historia	Varchar (50)
Cod_paciente	Integer
Edad	Integer
Operaciones:	
BuscarDatos()	

Nombre de la clase: INFORME_PSICOLOGICO	
Descripción: contiene datos sobre el informe psicológico del paciente	
Atributos:	
Cod_informe	Integer
Cod_-historia	Integer
Fecha_evaluación	Date
Fecha_informe	Date
Cod_usuario	Integer
Estudio	Text
Datos_ad	Text
Aspectos_clin	Text
Estado_psico	Text
Instrumentos	Text
Familia	Text
Diagnóstico	Text
Prop_terapéutica	Text
Evolución	Text
Pronóstico	Text
Fecha_registro	Date
Operaciones:	
RegistrarInforme()	

Nombre de la clase: LUGARES_CONSUMO	
Descripción: guarda información de los lugares de consumo	
Atributos:	
Cod_lugarconsumo	Integer
Cod_historia	Integer
Cod_lugar	Integer
Fecha_registro	Date
Operaciones:	
RegistrarLugares()	

Nombre de la clase: MENU	
Descripción: contiene las diferentes opciones de menú que tiene el sistema, a los que acceden los diferentes usuarios	
Atributos:	
Cod_enlace	Integer
Cod_personal	Integer
Estado	Char (1)
Fecha_registro	Date
Operaciones:	
BuscarMenu()	
CargarMenu()	

Nombre de la clase: OBTENCION_TOXICO	
Descripción: guarda información de la información del tóxico	
Atributos:	
Cod_obtencióntóxico	Integer
Cod_obtención	Integer
Cod_historia	Integer
Fecha_registro	Date
Operaciones:	
RegistrarObtencion()	

Nombre de la Clase: PERSONA	
Descripción: representa a la entidad persona con todos sus datos personales	
Atributos:	
Cod_persona	Integer
Nombres	Varchar (100)
Apellidop	Varchar (100)
Apellidom	Varchar (100)
CI	Varchar (10)
Fecha_nac	Date
Sexo	Char (1)
Lugar_de_nacimiento	Varchar (50)
Telf	Varchar (15)
Lugar_de_resid	Varchar (50)
Domicilio	Varchar (50)
N°_de_parejas	Integer
Pareja_actual	Varchar (100)
N°_de_hijos	Integer
Operaciones:	
BuscarPersonas()	
Adicionar()	
Eliminar()	
RecuperarDatos()	
ModificarPersona()	

Nombre de la clase: PACIENTE	
Descripción: representa al paciente	
Atributos:	
Cod_pac	Integer
Ocupación	Varchar (50)
Fecha_registro	Integer
Estado	Char (1)
Cod_persona	Integer
Operaciones:	
BuscarPacientes()	
Adicionar()	
Modificar()	
Eliminar()	
RecuperarDatos()	

Nombre de la clase: PROPUESTA_TERAPEUTICA	
Descripción: representa la propuesta terapéutica del paciente	
Atributos:	
Cod_propterap	Integer
Cod_historia	Integer
Fecha_registro	Date
Descripción	Text
Operaciones:	
AdicionarDatos()	

Nombre de la clase: PROFESIONALES_ACTUANTES									
Descripción: representa los profesionales que hacen el registro de la historia clinica del paciente									
Atributos: <table> <tr> <td>Cod_profactuates</td><td>Integer</td></tr> <tr> <td>Cod_historia</td><td>Integer</td></tr> <tr> <td>Fecha_registro</td><td>Date</td></tr> <tr> <td>Descripción</td><td>Text</td></tr> </table>		Cod_profactuates	Integer	Cod_historia	Integer	Fecha_registro	Date	Descripción	Text
Cod_profactuates	Integer								
Cod_historia	Integer								
Fecha_registro	Date								
Descripción	Text								
Operaciones: AdicionarDatos()									

Nombre de la clase: PERSONAL													
Descripción: contiene datos de los usuarios del sistema													
Atributos: <table> <tr> <td>Cod_personal</td><td>Integer</td></tr> <tr> <td>Usuario</td><td>Varchar (50)</td></tr> <tr> <td>Clave</td><td>Varchar (20)</td></tr> <tr> <td>Estado</td><td>Char (1)</td></tr> <tr> <td>Fecha_registro</td><td>Date</td></tr> <tr> <td>Cod_rol</td><td>Integer</td></tr> </table>		Cod_personal	Integer	Usuario	Varchar (50)	Clave	Varchar (20)	Estado	Char (1)	Fecha_registro	Date	Cod_rol	Integer
Cod_personal	Integer												
Usuario	Varchar (50)												
Clave	Varchar (20)												
Estado	Char (1)												
Fecha_registro	Date												
Cod_rol	Integer												
Operaciones: Verificar() BuscarPersonal() Adicionar() RecuperarDatos() ModificarPersonal() Eliminar()													

Nombre de la Clase: REINICIO_SUSPENSION		
Descripción: representa el reinicio y suspensión del consumo del paciente		
Atributos:		
Cod_reinicio		Integer
Cod_historia		Integer
Causas_suspension		Text
Causas_reinicio		Text
Agrego_sustancias		Text
Agrego_vias		Text
Agrego_dosis		Text
Cod_estadoconsumo		Integer
Fecha_registro		Date
Descripcion		Text
Operaciones:		
RegistrarReinicio()		

Nombre de la clase: ROL_PERSONAL		
Descripción: representa al rol que asumen los usuarios del sistema		
Atributos:		
Cod_rol	Integer	
Nombre_rol	Varchar (50)	
Estado	Char (1)	
Fecha_registro	Date	
Operaciones:		
ObtenerRol()		

Nombre de la Clase: TRATAMIENTO	
Descripción: representa el tratamiento del paciente	
Atributos:	
Cod_tratamiento	Integer
Cod_historia	Integer
Modalidad	Varchar(100)
Motivo	Varchar(100)
Fecha-inicio	Date
Fecha_fin	Date
Continua	Varchar(100)
Causa	Varchar(100)
Internaciones	Varchar(500)
Ambulatorio	Varchar(500)
Fecha_registro	Date
Operaciones:	
AdicionarTratamiento()	

5.3 DICCIONARIO DE DATOS

LP: llave primaria

LF: llave foránea

Tabla ANTECEDENTES

Atributo	Tipo	LP	LF	Descripción
Cod_antecedentes	Integer(autonumerico)	Si	No	Código del antecedente
Cod_historia	Integer	No	Si	Código de la historia clínica
Tipo_antecedente	Integer	No	Si	Tipo de antecedente
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica
Descripción	Text	No	No	Descripción del antecedente

Tabla AREA_FAMILIAR

Atributo	Tipo de dato	LP	LF	Descripción
Cod_area	Integer(autonumerico)	Si	No	Código área
Cod_historia	integer	No	Si	Código de la historia clínica
Trans_mentales	Varchar(100)	No	No	Trastornos mentales
Antec_físicos	Varchar(5)	No	No	Antecedentes físicos
Tóxicos	Varchar(100)	No	No	Consumo de tóxicos
Violencia_familiar	Varchar(100)	No	No	Violencia familiar
Antec_penales	Varchar(100)	No	No	Antecedentes penales
Actitud	Varchar(5)	No	No	Actitud de la familia
Dinámica_fam	Text	No	No	Vínculos

Tabla AXIS

Atributo	Tipo de dato	LP	LF	Descripción
Cod_axis	Integer(autonumerico)	Si	No	Código de axis
Cod_historia	Integer	No	Si	Código de la historia
Cod_tipoaxis	Integer	No	No	Código del tipo de axis
Cod_diagnostico	Integer	No	No	Código de diagnostico
Fecha_registro	Date	No	No	Fecha del registro de la historia clínica
Tipo_axis	Varchar(100)	No	Si	Código del paciente

Tabla ANTECEDENTES_LEGALES_CONSUMO

Atributo	Tipo de dato	LP	LF	Descripción
Cod_antec_legal	Integer(autonumerico)	Si	No	Código de la detención legal
Cod_detención	Integer	No	Si	Código de detención
Cod_historia	Integer	No	Si	Código de la historia clínica
Detencion_contravencion	Varchar(5)	No	No	Detención por contravención
Detencion_delito	Varchar(5)	No	No	Detención por delito
Detencion_accidente	Varchar(5)	No	No	Detención por accidente
Detencion_VIF	Varchar(5)	No	No	Detención VIF
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla CONSUMO_SUSTANCIAS

Atributo	Tipo de dato	LP	LF	Descripción
Cod_consumo	Integer(autonumerico)	Si	No	Código consumo
Cod_historia	Integer	No	Si	Código de la historia clínica
Cod_sustancia	Integer	No	Si	Código de la sustancia
Edad	Integer	No	No	Edad del paciente
Modalidad_uso	Varchar(50)	No	No	Modalidad de uso de la sustancia
Via_consumo	Varchar(50)	No	No	Vía de consumo de la sustancia
Cantidad	Varchar(50)	No	No	Cantidad de consumo
Fecha_UC	Date	No	No	Fecha del último consumo
Suspensión	Varchar(50)	No	No	Suspensión del consumo (total,parcial)
Tiempo	Varchar(50)	No	No	Tiempo (días,meses,años)
Cod_usuario	Integer	No	No	Código de usuario
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla CITA

Atributo	Tipo de dato	LP	LF	Descripción
Cod_cita	Integer(autonumerico)	Si	No	Código de la cita
Fecha	Date	No	No	Fecha de la cita
Hora	Time	No	No	Hora de la cita
Confirmar	Varchar (50)	No	No	Confirmar la cita
Fecha_registro	Date	No	No	Fecha del registro de la historia clínica
Cod_paciente	Integer	No	Si	Código del paciente

Tabla CONSULTA

Atributo	Tipo de dato	LP	LF	Descripción
Cod_consulta	Integer(Autonumerico)	Si	No	Código de la consulta
Motivo_consulta	Varchar(500)	No	No	Motivo de la consulta
Enfermedad_actual	Varchar(500)	No	No	Enfermedad actual
Cod_historia	Integer	No	No	Código de la historia del paciente
Fecha_registro	Date	No	No	Fecha del registro de la historia clínica

Tabla DIAGNOSTICO

Atributo	Tipo de dato	LP	LF	Descripción
Cod_diagnóstico	Integer(autonumerico)	Si	No	Código del diagnóstico
Cod_historia	Integer	No	Si	Código de la historia clínica
Tipo_diagnóstico	Varchar(50)	No	Si	Tipo de diagnóstico
Descripción	Text	No	No	Descripción del tipo de diagnóstico
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla DETENCION

Atributo	Tipo de dato	LP	LF	Descripción
Cod_detención	Integer	Si	No	Código de detención
Descripción	Varchar(50)	No	No	Descripción de la detención

Tabla ENLACES

Atributo	Tipo de dato	LP	LF	Descripción
Cod_enlace	Integer	Si	No	Código del enlace
Opción	Varchar (200)	No	No	Opción de menú
Estado	Char (1)	No	No	Activo(a) Inactivo(i)
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla EVOLUCION_RESUMEN

Atributo	Tipo de dato	LP	LF	Descripción
Cod_evolución	Integer(autonumerico)	Si	No	Código de evolución
Cod_historia	Integer	No	Si	Código de la historia clínica
Fecha_evolución	Date	No	No	Fecha del registro de la evolución
Descripción	Text	No	No	Descripción de la evolución del tratamiento
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla EXAMENES

Atributo	Tipo de dato	LP	LF	Descripción
Cod_examen	Integer(Autonumerico)	Si	No	Código del examen
Cod_historia	Integer	No	Si	Código de la historia clínica
Descripción	Text	No	No	Descripción del tipo de examen
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica
Tipo_examen	Varchar(50)	No	No	Tipo examen

Tabla ESTADO_CONSUMO

Atributo	Tipo de dato	LP	LF	Descripción
Cod_estadoconsumo	Integer	Si	No	Código del lugar
Descripción	Varchar(200)	No	No	Descripción del lugar

Tabla FICHA_ENFERMERA

Atributo	Tipo de dato	LP	LF	Descripción
Pieza	Integer	No	No	Número de pieza del paciente
Cama	Integer	No	No	Número de cama del paciente
Turno	Varchar(20)	No	No	Turno que hace la enfermera
Cod_historia	Integer	No	Si	Código de la historia del paciente
Fecha_registro	Date	No	No	Fecha del registro de la historia clínica
Descripción	Text	No	No	Detalle del comportamiento de paciente
Cod_ficha	Integer(autonumerico)	Si	No	Código de la ficha de la enfermera

Tabla FAMILIA_ BIOLOGICA

Atributo	Tipo de dato	LP	LF	Descripción
Cod_fam	Integer(autonumerico)	Si	No	Código de familia
Relación	Integer	No	Si	Relación con el paciente
Nombres	Varchar(50)	No	No	Nombres de los parientes
Apellidos	Varchar(50)	No	No	Apellidos de los parientes
Edad	Integer	No	No	Edad del pariente
Vive	Varchar(5)	No	No	Lugar donde vive
Ocupación	Varchar(20)	No	No	Ocupación del pariente
Origen	Varchar(50)	No	No	Lugar de origen
Residencia	Varchar(50)	No	No	Lugar donde vive
Escolaridad	Varchar(50)	No	No	Grado de escolaridad
Convive_con	Integer	No	No	Con quien vive el paciente
Cod_historia	Integer	No	Si	Código de la historia clínica
Fecha-registro	Date	No	No	Fecha de registro de la historia clínica

Tabla HISTORIA_CLINICA

Atributo	Tipo de dato	LP	LF	Descripción
Cod_historia	Numérico	Si	No	Es el código de la historia del paciente
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica
Médico	Varchar (100)	No	No	Médico que hace el registro
Tipo_historia	Varchar (20)	No	No	Tipo de la historia
Escolaridad	Varchar (50)	No	No	Escolaridad del paciente
Derivado	Varchar (100)	No	No	Médico al que ha sido derivado
Referente	Varchar (50)	No	No	
Fuente_historia	Varchar (50)	No	No	
Cod_paciente	Integer	No	Si	Código del paciente
Edad	Integer	No	No	Edad del paciente

Tabla INFORME_PSICOLOGICO

Atributo	Tipo de dato	LP	LF	Descripción
Cod_informe	Integer(autonumerico)	Si	No	Código del informe
Cod_-historia	Integer	No	Si	Código de la historia del paciente
Fecha_evaluación	Date	No	No	Fecha de la evaluación hecha al paciente
Fecha_informe	Date	No	No	Fecha en la que se hizo el informe
Cod_usuario	Integer	No	Si	Código del usuario
Estudio	Text	No	No	Motivo del estudio a hacer
Datos_ad	Text	No	No	Datos adicionales
Aspectos_clin	Text	No	No	Aspectos clínicos
Estado_psico	Text	No	No	Estado psicológico
Instrumentos	Text	No	No	Instrumentos usados
Familia	Text	No	No	Vínculo familiar
Diagnóstico	Text	No	No	Diagnóstico que presenta el paciente
Prop_terapéutica	Text	No	No	Propuesta terapéutica
Evolución	Text	No	No	La evolución del paciente
Pronóstico	Text	No	No	Pronóstico del comportamiento del paciente
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla LUGARES

Atributo	Tipo de dato	LP	LF	Descripción
Cod_lugar	Integer	Si	No	Código del lugar
Descripción	Varchar(200)	No	No	Descripción del lugar

Tabla LUGARES_CONSUMO

Atributo	Tipo de dato	LP	LF	Descripción
Cod_lugarconsumo	Integer(autonumerico)	Si	No	Código del lugar de consumo
Cod_historia	Integer	No	Si	Código de la historia clínica
Cod_lugar	Integer	No	Si	Código del lugar
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla MENU

Atributo	Tipo de dato	LP	LF	Descripción
Cod_enlace	Integer	Si	Si	Código del enlace
Cod_personal	Integer	Si	Si	Código del personal
Estado	Char (1)	No	No	Activo(a) Inactivo(i)
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla OBTENCION

Atributo	Tipo de dato	LP	LF	Descripción
Cod_obtención	Integer	Si	No	Código de obtención
Descripción	Varchar(100)	No	No	Descripción de cómo se obtuvo el tóxico

Tabla OBTENCION TOXICO

Atributo	Tipo de dato	LP	LF	Descripción
Cod_obtencióntóxico	Integer(autonumerico)	Si	No	Código de obtención del tóxico
Cod_obtención	Integer	No	Si	Código de obtención
Cod_historia	Integer	No	Si	Código de la historia clínica
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla OTROS

Atributo	Tipo de dato	LP	LF	Descripción
Cod_otros	Integer	Si	No	Código de otros
Cod_detenciónlegal	Integer	No	Si	Código de la detención legal
Otros	Varchar(100)	No	No	Otras causas de detención
Nº	Varchar(50)	No	No	Número de detenciones
Fugas	Varchar(100)	No	No	Fugas del lugar
Cod_usuario	Integer	No	Si	Código de usuario
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla PERSONA

Atributo	Tipo de dato	LP	LF	Descripción
Cod_persona	Integer(autonumerico)	Si	No	Código de la persona
Nombres	Varchar (100)	No	No	Nombres de la persona
Apellidop	Varchar (100)	No	No	Apellido paterno
Apellidom	Varchar (100)	No	No	Apellido materno
CI	Varchar (10)	No	No	Carnet de identidad del paciente
Fecha_nac	Date	No	No	Fecha de nacimiento del paciente
Sexo	Char (1)	No	No	Sexo del paciente
Lugar_de_nacimiento	Varchar (50)	No	No	Lugar de nacimiento del paciente
Telf	Varchar (15)	No	No	Número telefónico del paciente
Lugar_de_resid	Varchar (50)	No	No	Lugar de residencia del paciente
Domicilio	Varchar (50)	No	No	Domicilio del paciente
N°_de_parejas	Integer	No	No	Número de parejas del paciente
Pareja_actual	Varchar (100)	No	No	Pareja actual del paciente
N°_de_hijos	Integer	No	No	Número de hijos del paciente

Tabla PACIENTE

Atributo	Tipo de dato	LP	LF	Descripción
Cod_pac	Integer(autonumerico)	Si	No	Código del paciente
Ocupación	Varchar (50)	No	No	Ocupación del paciente
Fecha_registro	Integer	No	No	Fecha de registro del paciente
Estado	Char (1)	No	No	Activo(a) Inactivo(i)
Cod_persona	Integer	No	Si	Código de persona

Tabla PERSONAL

Atributo	Tipo de dato	LP	LF	Descripción
Cod_personal	Integer(autonumerico)	Si	No	Código del personal
Usuario	Varchar (50)	No	No	Nombre del usuario
Clave	Varchar (20)	No	No	Clave del usuario
Estado	Char (1)	No	No	Activo(a) Inactivo(i)
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica
Cod_persona	Integer	No	Si	Código de persona
Cod_rol	Integer	No	No	Código del rol

Tabla PROFESIONALES_ACTUANTES

Atributo	Tipo de dato	LP	LF	Descripción
Cod_profactuante	Integer(autonumerico)	Si	No	Código del profesional actuante
Cod-historia	Integer	No	Si	Código de la historia
Descripción	Text	No	No	Quien lleno la historia
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla PROPUESTA_TERAPEUTICA

Atributo	Tipo de dato	LP	LF	Descripción
Cod_propterapeutica	Integer(autonumerico)	Si	No	Código de la propuesta terapeutica
Cod-historia	Integer	No	Si	Código de la historia
Descripción	Text	No	No	Descripción de la propuesta terapéutica
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla ROL PERSONAL

Atributo	Tipo de dato	LP	LF	Descripción
Cod_rol	Integer	Si	No	Código del rol
Nombre_rol	Varchar (50)	No	No	Nombre del rol
Estado	Char (1)	No	No	Activo(a) Inactivo(i)
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica

Tabla REINICIO_SUSPENSION

Atributo	Tipo de dato	LP	LF	Descripción
Cod_reinicio	Integer(Autonumerico)	Si	No	Código del reinico_suspencion
Cod_historia	Integer	No	Si	Código de la historia
Causas_suspensión	Text	No	No	Causas de la suspensión
Causas_reinicio	Text	No	No	Causas del reinicio
Agregó_sustancias	Varchar(5)	No	No	Agregó sustancias a consumir
Agregó_vías	Varchar(5)	No	No	Agregó vías de consumo
Agregó_dosis	Varchar(5)	No	No	Agregó dosis a consumir
Cod_estadoconsumo	Integer	No	Si	Código del estado_consumo
Fecha_registro	Date	No	No	Fecha del registro de la historia clínica
Descripción	Text	No	No	Descripción del reinicio

Tabla RELACION_FAMILIA

Atributo	Tipo de dato	LP	LF	Descripción
Cod_relación	Integer	Si	No	Relación con el paciente
Descripción	Varchar(50)	No.	Si	Descripción de la relación

Tabla SUSTANCIA

Atributo	Tipo de dato	LP	LF	Descripción
Cod_sustancia	Integer	Si	No	Código de la sustancia
Descripción	Varchar(100)	No	No	Descripción de la sustancia

Tabla TIPO ANTECEDENTES

Atributo	Tipo de dato	LP	LF	Descripción
Tipo_antecedente	Integer	Si	No	Tipo de antecedente
Descripción	Text	No	No	Descripción del antecedente
Padre	Integer	No	No	Determina si el campo tiene padre

Tabla TRATAMIENTO

Atributo	Tipo de dato	LP	LF	Descripción
Cod_tratamiento	Integer	Si	No	Código del tratamiento
Cod_historia	Integer	No	Si	Código de la historia clínica
Modalidad	Varchar(50)	No	No	Modalidad del tratamiento
Motivo	Varchar(100)	No	No	Motivo del tratamiento
Fecha_inicio	Date	No	No	Fecha de inicio del tratamiento
Fecha_fin	Date	No	No	Fecha de finalización del tratamiento
Continua	Varchar(5)	No	No	Si o No
Causa	Varchar(100)	No	No	Causa del si o no
Fecha_registro	Date	No	No	Fecha de registro de la historia clínica
Internaciones	Varchar(500)	No	No	Porque fue internado el paciente
Ambulatorio	Varchar(500)	No	No	

Tabla TIPO DIAGNOSTICO

Atributo	Tipo de dato	LP	LF	Descripción
Tipo_diagnóstico	Varchar(50)	Si	No	Tipo de diagnóstico
Descripción	Varchar(500)	No	No	Descripción del tipo de diagnóstico

Tabla TIPO AXIS

Atributo	Tipo de dato	LP	LF	Descripción
Tipo_axis	Varchar(50)	Si	No	Tipo de axis
Descripción	Varchar(500)	No	No	Descripción del tipo de axis

5.4. DISEÑO DE INTERFACES

5.4.1 Pantalla para Conectarse

Generalidades:

Pantalla: 000

Usuario: Usuario sistema

Pantalla origen: ----

Requisitos: usuario y contraseña

custom User Interface Model

PAGINA PRINCIPAL000

Ingresar

Usuario

Contraseña

Conectar

Datos en blanco o incompletos

Usuario no existe o contraseña incorrecta



5.4.1.1 Pantalla menú enfermera

Generalidades:

Pantalla: 100

Usuario: Enfermera

Pantalla origen: 000

Requisitos: usuario y contraseña

PANTALLA 100: MENU ENFERMERA



5.4.1.2 Pantalla menú personal médico

Generalidades:

Pantalla: 200

Usuario: Medico

Pantalla origen: 000

Requisitos: usuario y contraseña



5.4.1.3 Pantalla menú psicólogo

Generalidades:

Pantalla: 300

Usuario: Psicólogo

Pantalla origen: 000

Requisitos: usuario y contraseña



5.4.1.4 Pantalla menú administrador sistema

Generalidades:

Pantalla: 400

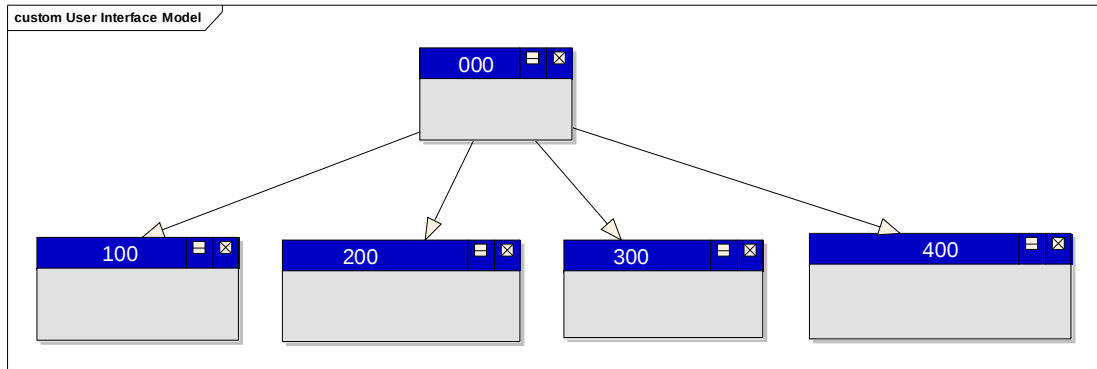
Usuario: Administrador Sistema

Pantalla origen: 000

Requisitos: usuario y contraseña



Navegacional:



5.4.2 Pantalla para administrar pacientes

Generalidades:

Pantalla: 110

Usuario: Enfermera

Pantalla origen: 100

Requisitos: ninguno

custom User Interface Model

ADMINISTRAR PACIENTES 110

Menú Inicio

Desconectar

° Adicionar Paciente

° Modificar Paciente (haga click sobre el icono correspondiente)

° Eliminar Paciente

Nombres	Apellidos	Fecha Nacimiento	OPCIONES

Diagrama de la pantalla de administración de pacientes. La pantalla tiene un título azul con el texto "ADMINISTRAR PACIENTES" y el número "110". El contenido principal es un área gris con un menú de inicio que incluye un botón "Desconectar" y tres opciones de acción: "° Adicionar Paciente", "° Modificar Paciente (haga click sobre el icono correspondiente)" y "° Eliminar Paciente". En la parte inferior, hay una tabla con cuatro columnas: "Nombres", "Apellidos", "Fecha Nacimiento" y "OPCIONES". La columna "OPCIONES" contiene un campo de texto.

5.4.2.1 Pantalla para adicionar pacientes

Generalidades:

Pantalla: 111

Usuario: Enfermera

Pantalla origen: 110

Requisitos: ninguno

custom User Interface Model

ADICIONAR PACIENTES111

Nombres		Apellido mat.		Apellido pat.	
C.I		Fecha de Nac.		Edad	
				Sexo	
Escolaridad		Estado Civil		Lugar de Nacimiento	
Ocupacion:		Telf:		Lugar de Residencia	
Domicilio:		Nº de parejas		Pareja Actual	
Nº de Hijos		Fecha		Egreso	
				Causa	
Derivado Por:					
Referente		Fuente de la Historia			

Adicionar

Cancelar

5.4.2.2.Pantalla para modificar pacientes

Generalidades:

Pantalla: 112

Usuario: Enfermera

Pantalla origen: 110

Requisitos: cod_paciente=código del paciente

custom User Interface Model

MODIFICAR PACIENTES 112

Nombres		Apellido mat.		Apellido pat.	
C.I		Fecha de Nac.		Edad	
				Sexo	
Escolaridad		Estado Civil		Lugar de Nacimiento	
Ocupacion:		Telf:		Lugar de Residencia	
Domicilio:		Nº de parejas		Pareja Actual	
Nº de Hijos		Fecha		Egreso	
				Causa	
Derivado Por:					
Referente		Fuente de la Historia			
Ingresar codigo del paciente			Modificar		Cancelar

5.4.2.3.Pantalla para eliminar pacientes

Generalidades:

Pantalla: 113

Usuario: Enfermera

Pantalla origen: 110

Requisitos: ninguno



5.4.3 Pantalla para administrar citas

Generalidades:

Pantalla: 120

Usuario: Enfermera

Pantalla origen: 100

Requisitos: ninguno

5.4.3.1 Pantalla para adicionar citas

Generalidades:

Pantalla: 121

Usuario: Enfermera

Pantalla origen: 120

Requisitos: ninguno

custom User Interface Model

ADICIONAR CITAS 121

Fecha

Hora

Nombres

Apellidos

Médico

Adicionar **Cancelar**

5.4.3.2 Pantalla para modificar citas

Generalidades:

Pantalla: 122

Usuario: Enfermera

Pantalla origen: 120

Requisitos: ninguno

custom User Interface Model

MODIFICAR CITAS

122

Fecha

Hora

Nombres

Apellidos

Médico

INTRODUCIR
CODIGO CITA

Modificar

Cancelar

5.4.3.3 Pantalla para eliminar citas

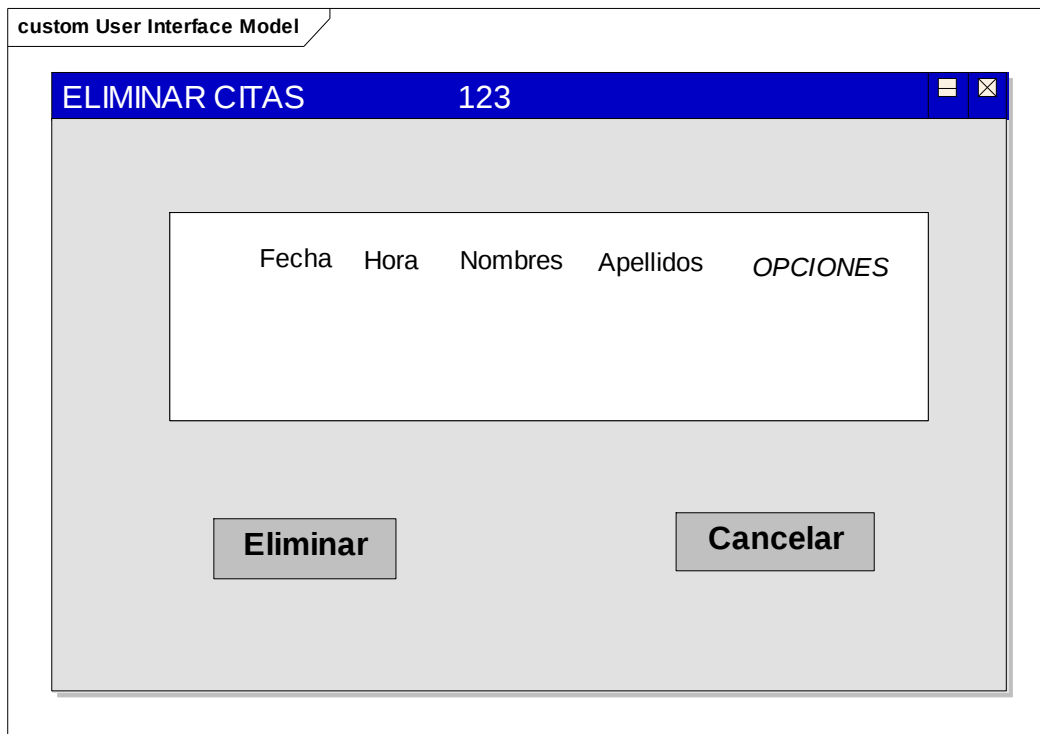
Generalidades:

Pantalla: 123

Usuario: Enfermera

Pantalla origen: 120

Requisitos: ninguno



5.4.4 Pantalla para Registrar Comportamiento de Paciente y Tratamiento

Generalidades:

Pantalla: 130

Usuario: Enfermera

Pantalla origen: 100

Requisito: ninguno

custom User Interface Model

BUSCAR PACIENTES130

Menú Inicio

Codigo Paciente

Buscar

NombresApellidosFecha Nacimiento

Paciente no Encontrado

Desconectar

Generalidades:

Pantalla: 131

Usuario: Enfermera

Pantalla origen: 130

Requisitos: ninguno

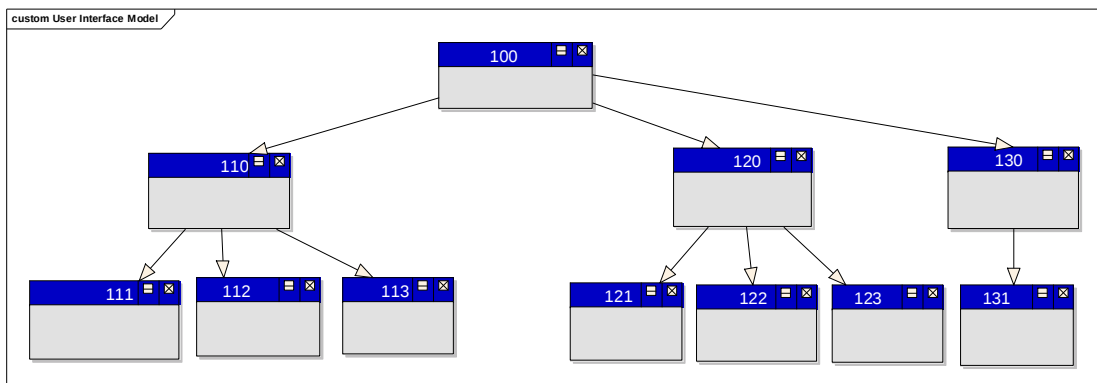
custom User Interface Model

REGISTRAR COMPORTAMIENTO 131

Pieza	
Cama	
Turno	
Fecha	
Descripción	

Guardar **Cancelar**

Navegacional para Administrar Pacientes, Administrar Citas y Registrar Comportamiento



5.4.5 Pantalla para registrar historia clínica consulta interna

Generalidades:

Pantalla: 210

Usuario: Personal Médico

Pantalla origen: 200

Requisitos: ninguno

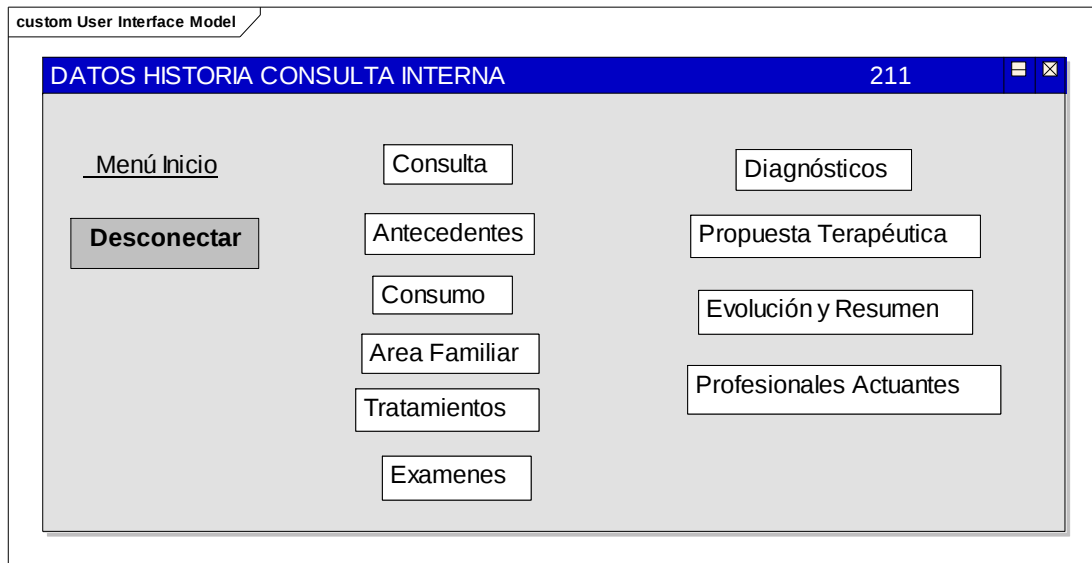
The screenshot shows a software window titled 'BUSCAR PACIENTES' with the number '210' in the top right corner. The window is part of a 'custom User Interface Model'. Inside the window, there is a 'Menú Inicio' button on the left and a 'Codigo Paciente' text box on the right. Below the 'Codigo Paciente' box is a 'Buscar' button. Underneath the 'Buscar' button is a large white rectangular area containing the labels 'Nombres', 'Apellidos', and 'Fecha Nacimiento'. At the bottom of the window, there are two buttons: 'Paciente no Encontrado' on the left and 'Desconectar' on the right.

Pantalla: 211

Usuario: Personal Médico

Pantalla origen: 210

Requisitos: código paciente



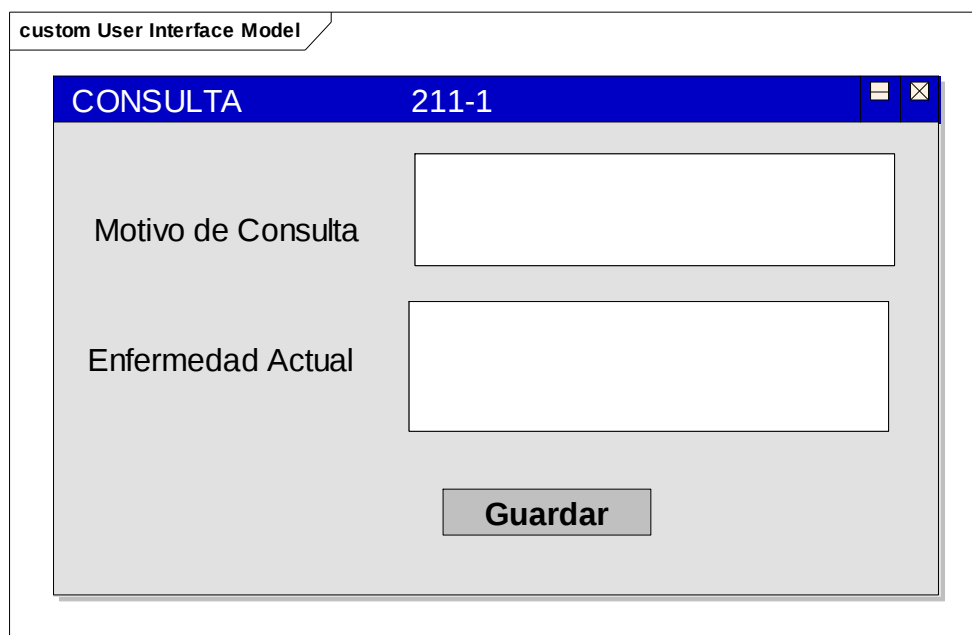
5.4.5.1 Pantalla para registrar motivo de consulta y enfermedad actual

Pantalla: 211-1

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno



5.4.5.2 Pantalla para registrar consumo

Pantalla: 211-2

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno

custom User Interface Model

REGISTRAR CONSUMO211-2

SustanciaAlcohol

Edad de inicio

Modalidad de uso

Vía de Consumo

Cantidad (ml. o gr.)

F.U.C (d, m, a)

Suspensión del Consumo

Tiempo (d, m, a)

a) Lugares de Consumo

En la calle

Trabajo

Bares Quioscos

Cancha

Reunión Amigos

Video Juegos

Escuelas

Domicilio

b) Obtención del Tóxico

Dinero Propio

Trueque Objetos Propios

Regalo

Dinero Ajeno

Trueque Objetos Ajenos

Robo-Hurto

Trueque Heterosexual

Trueque Homosexual

c) Antecedentes Legales por Consumo

Detención por Contravención

Detención por Delito

Detención por Accidente

Detención por V.I.F

Otros

Institución de menores

Nº

Fugas

d) Suspensión del Consumo

Causas

e) Reinicio del Consumo

Causas

GUARDAR

5.4.5.3 Pantalla para registrar antecedentes

Pantalla: 211-3

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno

ANTECEDENTES PERSONALES

1. Perinatales

2. Desarrollo Psicomotor

3. Personalidad Previa

ANTECEDENTES PATOLOGICOSEnfermedades
Infectocontagiosas

Otras Enfermedades

Traumatismos Intervenciones
Quirurgicas**ANTECEDENTES ESCOLARES**

a) Primaria

c) Terciaria

e) Ninguna

b) Secundaria

d) Universitaria

f) Especial

Inicio de vida escolar, rendimiento
académico

Causas de repetición o abandono

ANTECEDENTES LABORALES

Edad de Inicio

Ocupaciones

Horario de trabajo:

¿ Legalizado?

SI

NO

NC

Insalubre

SI

NO

NC

ANTECEDENTES SEXUALES

Menarca

Menopausia

UPM

Gestaciones

Partos

Cesareas

Abortos

Edad de inicio

Masturbación

Tipo de relaciones:

Heterosexual

Homosexual

Estables

Promiscuas

Trastornos Sexuales

Abuso Sexual:

NO

SI

Edad

¿ Quien?

Infecciones de Transmisión Sexual

Union Conyugal

Guardar

5.4.5.4 Pantalla para registrar exámenes

Pantalla: 211-4

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno

custom User Interface Model

REGISTRAR EXAMENES

211-4

Examen Físico

Examen Mental

Examen Neurológico

Guardar

5.4.5.5 Pantalla para registrar diagnostico

Pantalla: 211-5

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno

custom User Interface Model

REGISTRAR DIAGNOSTICO 211-5

Axis I ▼

Axis II ▼

Axis III ▼

Axis IV ▼

Axis V ▼

Guardar

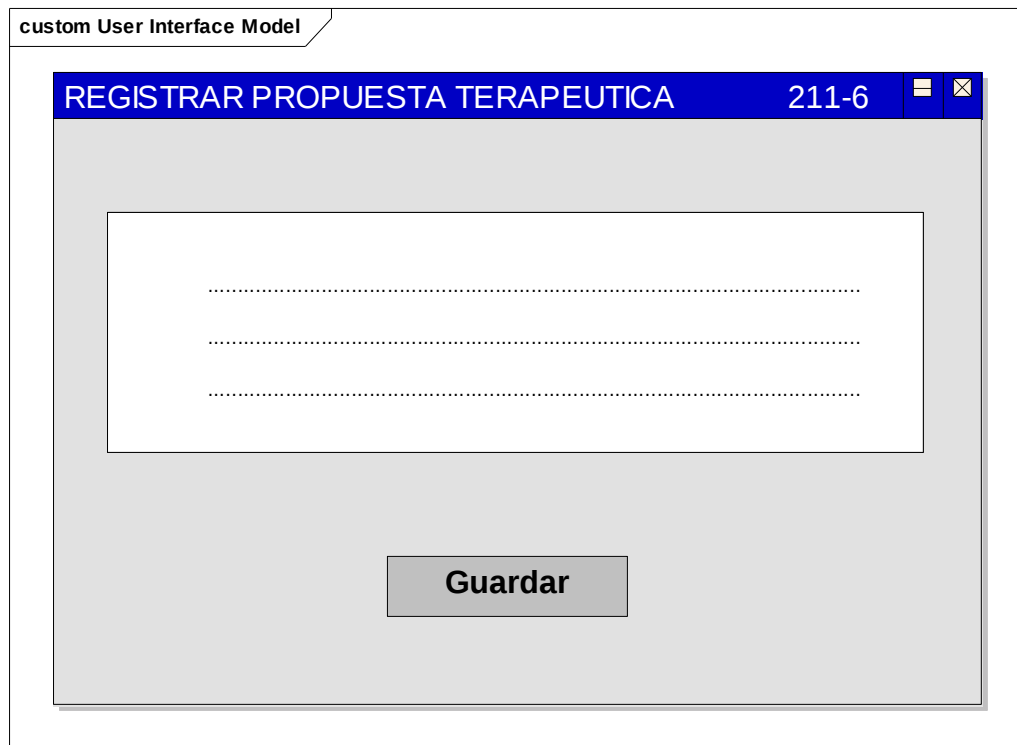
5.4.5.6 Pantalla para registrar propuesta terapéutica

Pantalla: 211-6

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno



5.4.5.7 Pantalla para registrar área familiar

Pantalla: 211-7

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno

REGISTRAR AREA FAMILIAR

211-7

1. Antecedentes de :

a) Transtornos mentales

Si

☐

No

☐

Especificar

b) Antecedentes físicos

Si

☐

No

☐

Especificar

c) Consumo de tóxicos

Si

☐

No

☐

Especificar

d) Violencia familiar

Si

☐

No

☐

Especificar

e) Antecedentes penales

Si

☐

No

☐

Especificar

2. Familia Biológica:

Relación	Nombres y Apellidos	Edad	Vive	Ocupación	Origen	Residencia	Escolaridad
Padre							
Madre							
Hermanos							

3. Convive con:

4. Actitud hacia el paciente:

Aceptación

☐

Rechazo

☐

Indiferente

☐

Otros

5. Dinámica Familiar

(vínculos)

Guardar

5.4.5.8 Pantalla para registrar tratamientos

Pantalla: 211-8

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno

custom User Interface Model

REGISTRAR TRATAMIENTOS

211-8

Fecha de Comienzo

Modalidad

Fecha de Finalización

Motivo

Continua

SI

☐

NO

☐

Causa

Internaciones

Ambulatorio

Guardar

5.4.5.9 Pantalla para registrar evolución y resumen

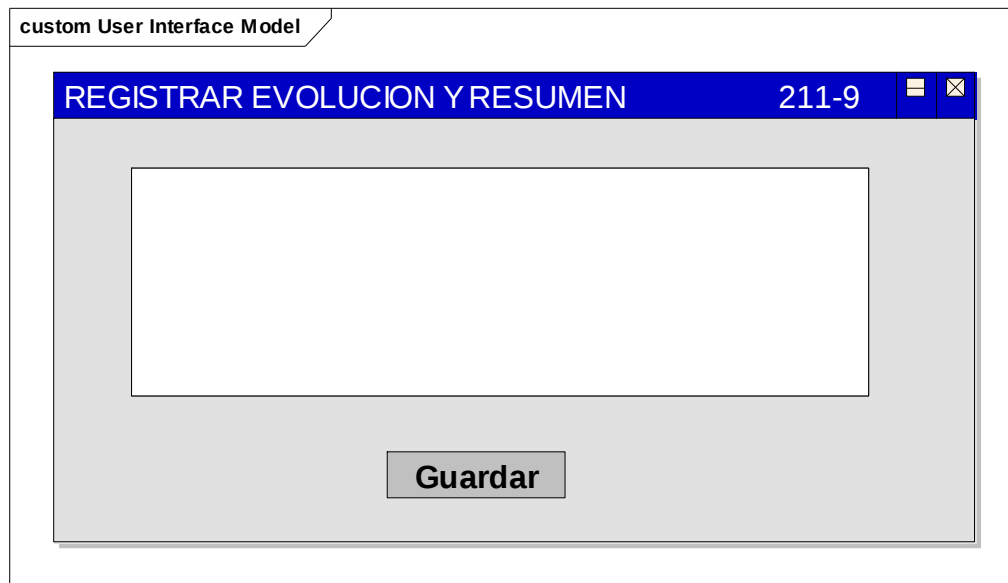
Generalidades

Pantalla: 211-9

Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno



5.4.5.10 Pantalla para registrar profesionales actuantes

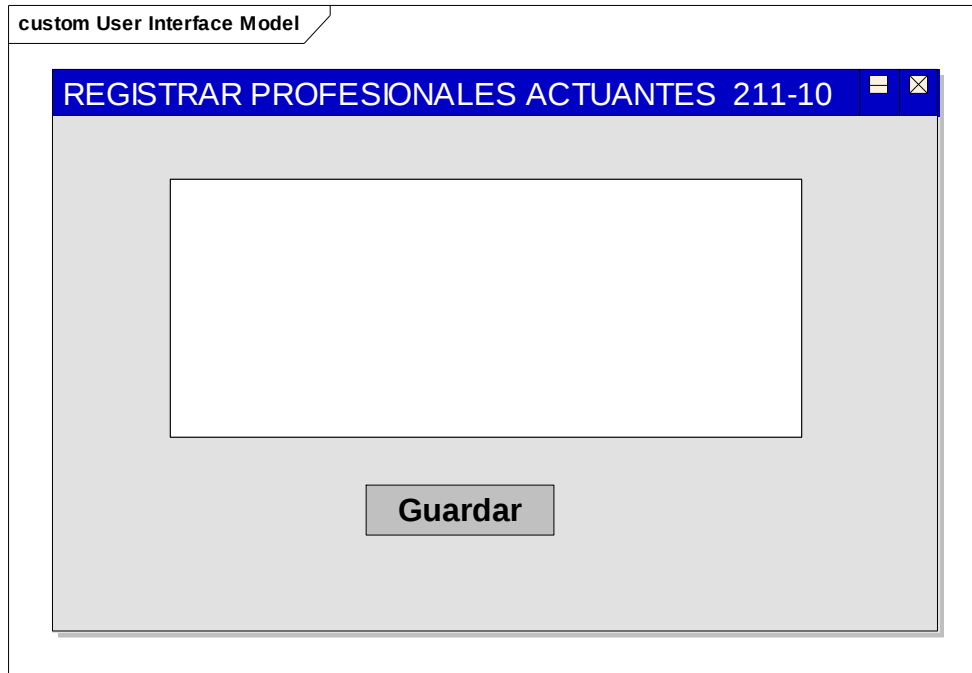
Generalidades

Pantalla: 211-10

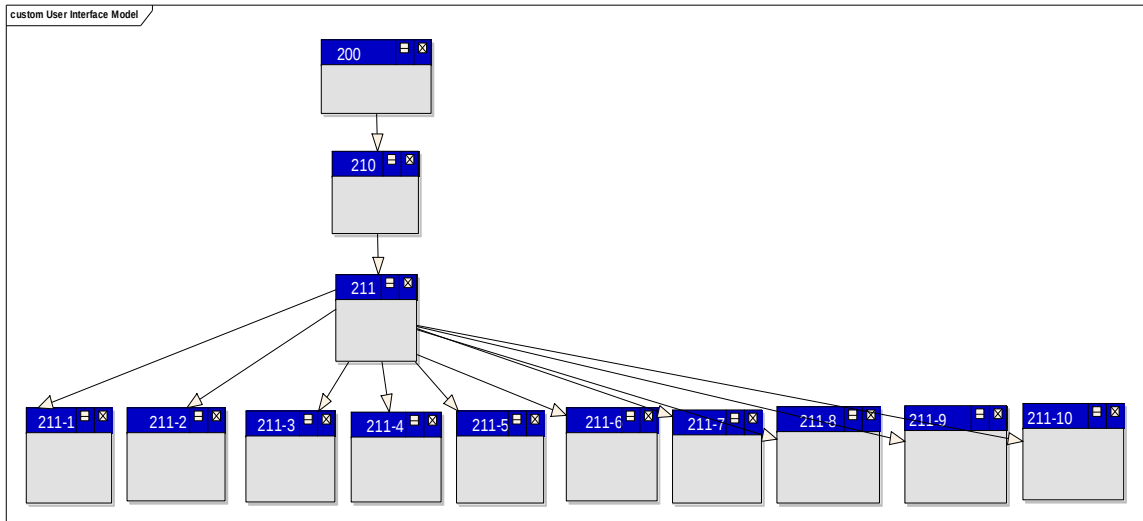
Usuario: Personal Médico

Pantalla origen: 211

Requisitos: ninguno



Navegacional:



5.4.6 Pantalla para registrar informe psicológico

Generalidades

Pantalla: 301

Usuario: Psicólogo

Pantalla origen: 300

Requisitos: ninguno

custom User Interface Model

REGISTRAR INFORME PSICOLOGICO 301

DATOS PERSONALES

MOTIVOS DE ESTUDIO

.....

5.4.7 Pantalla para reportes

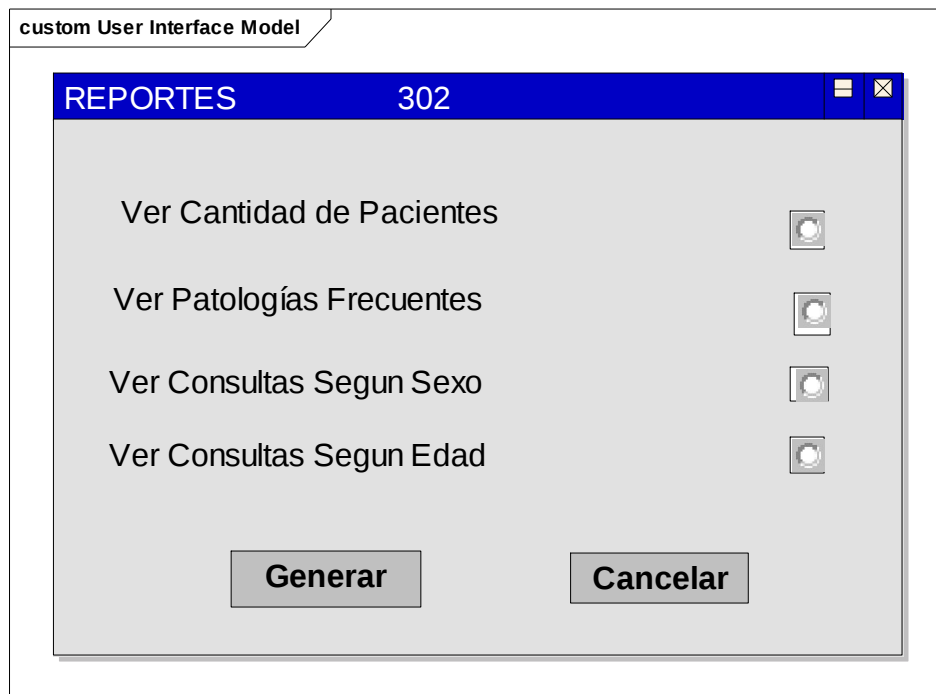
Generalidades

Pantalla: 302

Usuario: Psicólogo

Pantalla origen: 300

Requisitos: ninguno



5.4.7.1 Pantalla Reporte Cantidad de Pacientes

Generalidades:

Pantalla: 302-1

Usuario: Psicólogo

Pantalla origen: 302

Requisitos: ninguno

custom User Interface Model

REPORTE CANTIDAD DE PACIENTES 302-1



CANTIDAD DE PACIENTES

De A

Cantidad de Pacientes Internos	99
Cantidad de Pacientes Externos	99
Cantidad de Pacientes Mujeres	99
Cantidad de Pacientes Hombres	99

5.4.7.2 Pantalla Reporte Patologías Frecuentes

Generalidades:

Pantalla: 302-2

Usuario: Psicólogo

Pantalla origen: 302

Requisitos: ninguno

custom User Interface Model

REPORTE PATOLOGIAS FRECUENTES 302-2



PATOLOGIAS FRECUENTES

DE A

DIAGNOSTICO	Nº
Síndrome Dependencia Alcohol	21
Epilepsia	5
Trastorno Afectivo Organico	3
Trastorno Psicotico Organico	4

5.4.7.3 Pantalla Reporte Consultas Según Sexo

Generalidades:

Pantalla: 302-3

Usuario: Psicólogo

Pantalla origen: 302

Requisitos: ninguno

custom User Interface Model

REPORTE CONSULTAS SEGUN SEXO 302-3

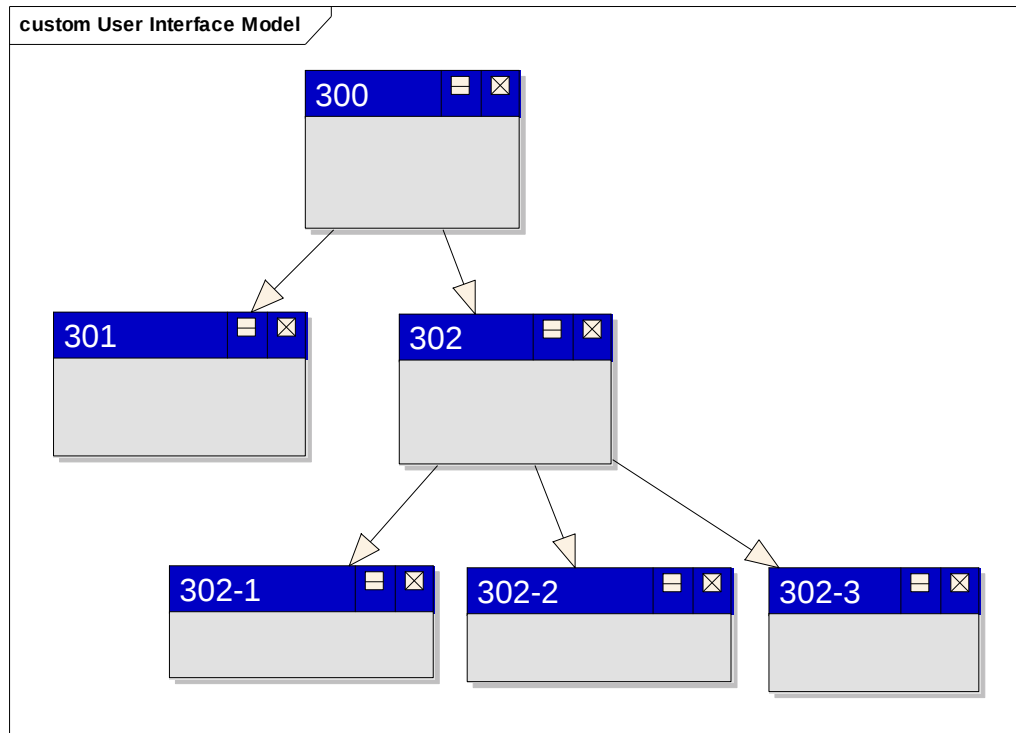


CONSULTAS SEGUN SEXO

DE A

GENERO	INTERNOS	EXTERNOS
Masculino	10	123
Femenino	3	155
Total	13	300

Navegacional:



5.4.8 Pantalla Administrar Usuarios

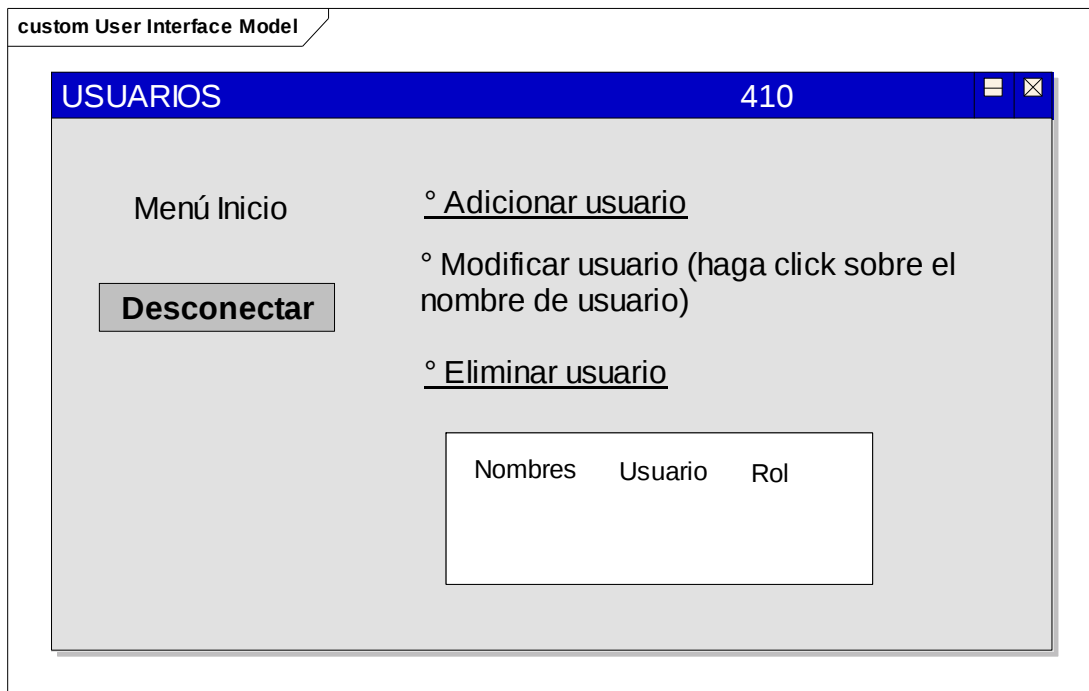
Generalidades:

Pantalla: 410

Usuario: Psicólogo

Pantalla origen: 400

Requisitos: ninguno



5.4.8.1 Pantalla Adicionar Usuario

Generalidades:

Pantalla: 411

Usuario: Psicólogo

Pantalla origen: 410

Requisitos: ninguno

custom User Interface Model

ADICIONAR USUARIO

411

C.I:

Usuario:

Nombres:

Clave:

Apellidoop:

Rol:

Enfermera

Apellidoom:

Sexo:

Dirección:

Adicionar

Cancelar

5.4.8.2 Pantalla Modificar Usuario

Generalidades:

Pantalla: 412

Usuario: Psicólogo

Pantalla origen: 410

Requisitos: ninguno

custom User Interface Model

MODIFICAR USUARIO

412

C.I:		Usuario:	
Nombres:		Clave:	
Apellidop:		Rol:	
Apellidom:			

Modificar

Cancelar

5.4.8.3 Pantalla Eliminar Usuario

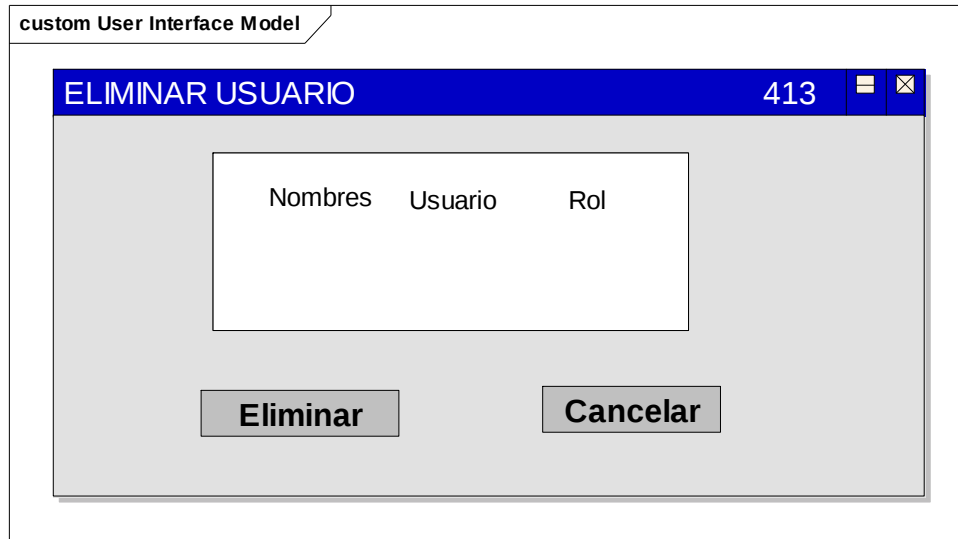
Generalidades:

Pantalla: 413

Usuario: Psicólogo

Pantalla origen: 410

Requisitos: ninguno



5.4.9 Pantalla Restaurar Base de Datos

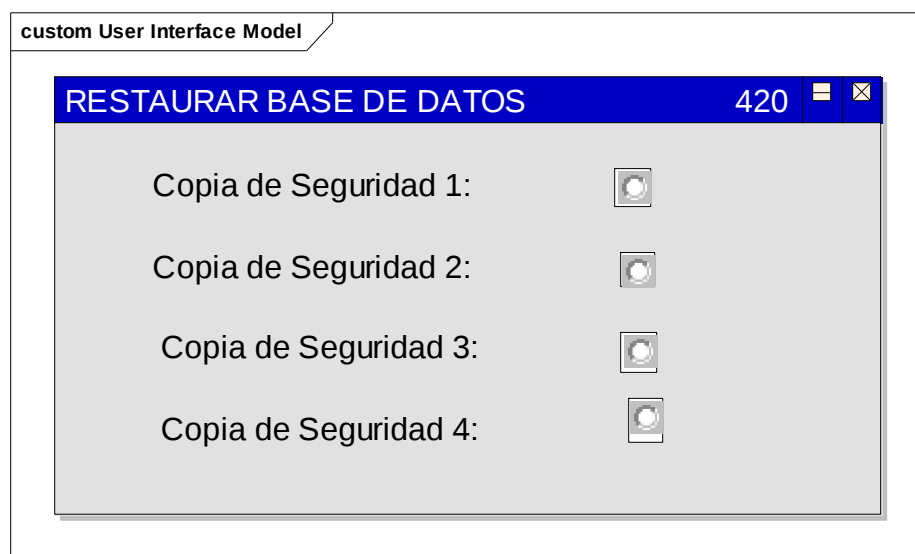
Generalidades:

Pantalla: 420

Usuario: Psicólogo

Pantalla origen: 400

Requisitos: ninguno



5.4.10 Pantalla Realizar Copias de Seguridad

Generalidades:

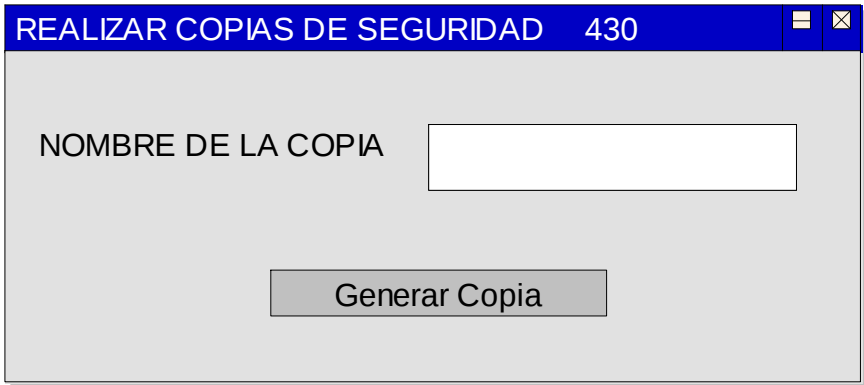
Pantalla: 430

Usuario: Psicólogo

Pantalla origen: 400

Requisitos: ninguno

custom User Interface Model

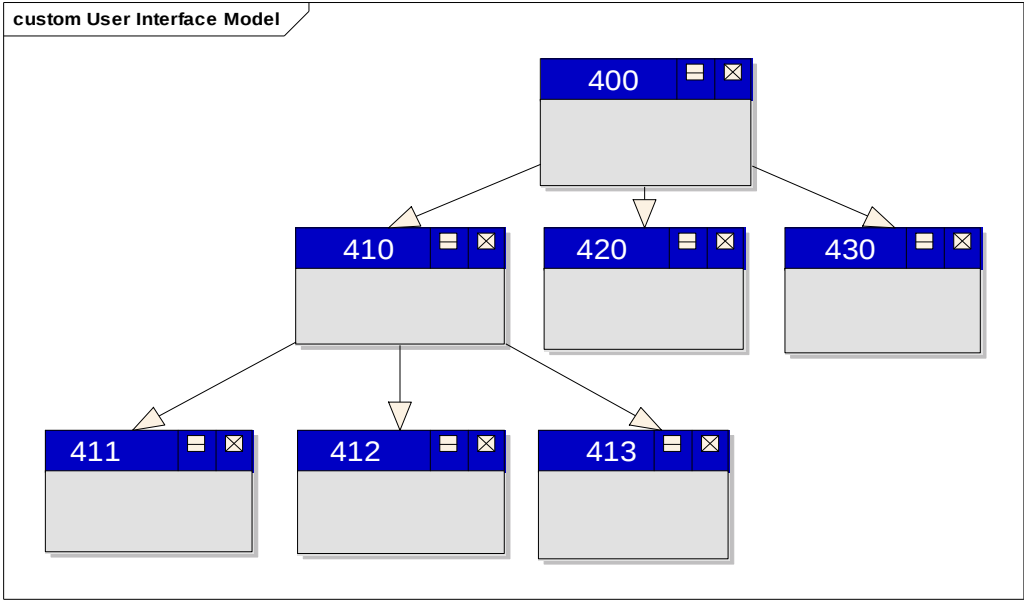


REALIZAR COPIAS DE SEGURIDAD 430

NOMBRE DE LA COPIA

Generar Copia

Navegacional:



6.1 Conclusiones

Por lo observado durante el desarrollo de este trabajo, concluimos que los sistemas manuales de información no son los más adecuados para tratar datos, cuando existe un crecimiento apreciable de los mismos, principalmente por el tiempo que demanda su procesamiento.

El “ Sistema automatizado para el registro de historias clínicas de los pacientes del Intraid ” demostrará ser una buena solución, a los problemas en la prestación de servicios de un centro médico, mejorando los tiempos de ejecución de los procesos de consulta interna y externa, permitirá que los pacientes sean atendidos de manera más rápida y se podrá llevar un mejor control de sus historias clínicas, evitando la duplicidad de trabajo, registrando los datos una sola vez, lo que reduce el porcentaje de error, inconsistencias y hacer más fiables los reportes que genera.

6.2 Recomendaciones

Al concluir este trabajo se recomienda lo siguiente:

- Que la implementación del sistema se la realice con el lenguaje de programación C Sharp, porque es un lenguaje orientado a objetos.
- Que se mantenga el diseño de la base de datos, porque permite que el sistema pueda evolucionar en el tiempo.
- Implementar el sistema en red o comenzar a trabajar en un proyecto que cubra estas necesidades.
- Realizar los esfuerzos necesarios para entrar completamente en la era de la informática, ya que hasta el momento, las tareas son realizadas casi manualmente con los consecuentes problemas que ello implica.