[Nombre del autor]

[Dirección de correo electrónico]

Descripción breve

[Dibujar su lector con un resumen de la participación. Normalmente es un breve resumen del documento.   
Cuando esté listo para agregar contenido, haga clic aquí y empiece a escribir.]

[Título del documento]

[Subtítulo del documento]

1. **Especificación del Sistema**
   1. **Introducción**

Hoy en día gran parte de los más importantes componentes de cualquier empresa es la informática. Desde la gran mayoría de los puntos de vista, esta supone un enorme avance en términos productividad, fiabilidad, calidad y desempeño.

Esto facilita enormemente cualquier actividad en la que la empresa pueda tomar parte, y solucionar complejos y aparatosos problemas, que sin estos soportes software requerirían de infinitos quebraderos de cabeza e innumerables pérdidas de tiempo y dinero.

Siendo esta informatización crucial para cualquier etapa de un negocio. Desde la creación, donde es importante para darse a conocer al mundo, hasta su etapa de madurez, donde el software es absolutamente necesario para la gestión de cualquier negocio. Siendo altamente relevante también para las tareas de marketing y publicidad.

* 1. **Descripción del problema**

La empresa solicitante, es una pequeña clínica que requería de un software para la gestión secretaría.

Como idea principal se plantea la informatización de la secretaría en orden de gestionar de forma más sencilla y eficiente, las citas, los distintos pacientes y sus respectivos tratamientos e historiales médicos. Permitiendo al secretario, añadir, consultar y modificar a placer todos estos elementos de manera cómoda y sencilla.

Para ello, nuestro producto final debe resolver todas las casuísticas anteriormente presentadas, por lo tanto, se presenta un software que mediante la interfaz de línea de comandos el secretario pueda gestionar todo.

Con la motivación de esclarecer en mayor medida el problema presentado, se presentarán una serie de subapartados que se muestran a continuación.

* + 1. **Gestión de pacientes**

Una de las partes que debíamos abordar eran los pacientes, elementos más importantes de la empresa. Para ello debíamos implementar una solución capaz de almacenar todos sus datos relevantes. Además, el sistema, permite que siempre que se desee, se pueda acceder a los datos de estos, además de cambiarlos si fuera necesario.

* + 1. **Gestión de tratamientos**

El sistema permite que siempre que un paciente esté registrado, se puedan almacenar y trabajar cómodamente con todos los tratamientos de los pacientes, pudiendo acceder a todos los datos relevantes sobre tratamientos.

* + 1. **Gestión de historial médico**

El sistema también gestionará el historial médico de cada paciente registrado.

* + 1. **Gestión de las citas**

Además de los anteriores, el sistema será capaz de gestionar las citas de los clientes, pudiendo el secretario modificarlas, añadirlas o eliminarlas fácilmente en pocos pasos.

* 1. **Solución propuesta**

Una vez establecido el problema, el siguiente paso es encontrar una solución que se definiría como un software que permitiría gestionar los pacientes, el historial médico, tratamientos y citas. Para ello usaremos el lenguaje de programación orientado a objetos C++, de manera que haremos uso de clases para abstraer el problema real y descomponerlo en partes más simples. Usando como sistema operativo LINUX.

La solución implementada presenta diversas ventajas como:

* Una mayor productividad, ya que la información es más fácil y eficiente de recuperar.
* La información se preserva, por lo que la seguridad ante estafas aumenta considerablemente.
* Permite llevar un control mayor sobre los pacientes, tratamientos, historial médico etc

1. **Especificación de requisitos.** 
   1. **Introducción**

Tras una reunión con el secretario de la clínica, abstraemos del problema presentado y la conversación establecida los siguientes elementos y requisitos.

Los requisitos funcionales y no funcionales son una parte de vital importancia a la hora de realizar cualquier software. Ya que sin ellos, no podríamos realizar un sistema que correctamente adecuado a las necesidades del cliente

## **Datos que encontraremos en la aplicación**

### **Datos del paciente**

* Nombre y apellidos.
* Dirección de la calle.
* Código postal.
* Provincia.
* Ciudad.
* DNI.

### **Tratamiento**

* Nombre del medicamento.
* Duración del tratamiento.
* Dosis diaria.
* Fecha de comienzo del tratamiento.

### **Historial médico**

* Fecha en la que se actualiza el historial.
* Síntomas.
* Pruebas realizadas al paciente.
* Tratamiento que se le impuso al paciente.

### **Citas**

* Nombre y apellidos de la persona que ha reservado la cita.
* Fecha y hora de la cita.

## **Requisitos funcionales**

1. Mostrar datos del paciente.
2. Mostrar tratamientos del paciente.
3. Mostrar historial médico del paciente.
4. Mostrar lista con pacientes existentes.
5. Mostrar citas reservadas.
6. Modificar datos del paciente.
7. Añadir nuevos pacientes.
8. Añadir nuevos tratamientos.
9. Añadir al historial médico nuevas consultas.
10. Consultar disponibilidad de citas en un día seleccionado.
11. Permitir reservar citas cuando estén disponibles.
12. Permitir modificar citas reservadas.
13. Permitir eliminar citas reservadas.
14. Mostrar menú al abrir aplicación junto a citas reservadas.
15. Permitir eliminar paciente.
16. Permitir eliminar tratamientos.

## **Requisitos no funcionales**

1. La aplicación utilizará el sistema operativo Linux.
2. Lenguaje utilizado: C++.
3. Evitar que se solapen las citas.
4. Evitar que el historial médico y los tratamientos finalizados de los pacientes sea borrado.
5. Mostrar mensaje de confirmación al eliminar cualquier dato de algún paciente.
6. Evitar que se pueda trabajar con pacientes no registrados

# **Actores, Casos de uso y priorización de los mismos.**

Para definir los casos de uso de nuestro sistema, primero, procederemos a exponer, las historias de usuario. Estas son representaciones de requisitos, utilizando un lenguaje común del usuario, haciendo de estas, una forma natural de análisis funcional.

## **Historias de usuario**

## **Añadir un paciente**

## **Nombre: AÑADIR UN PACIENTE**

ID: H01

Descripción: Como administrador quiero poder registrar un nuevo paciente, junto con sus datos personales.

Validación:

* Se debe introducir nombre y apellidos obligatoriamente.
* Se debe introducir los datos personales del mismo para que el paciente sea creado de forma correcta.
* Se añadirá el paciente recién inscrito a la base de datos de Pacientes.

## **Borrar paciente**

## **Nombre: BORRAR PACIENTE**

ID: H02

Descripción: Como administrador quiero borrar a cualquier paciente para que este no aparezca en la base de datos.

Validación:

* Se seleccionará el paciente a borrar de la base de datos de pacientes.
* Se borrará los datos del paciente, el historial y el tratamiento en curso (si el tratamiento aún no ha finalizado) del mismo.
  + 1. **Consultar datos personales**

## **Nombre: CONSULTAR DATOS PERSONALES**

ID: H03

Descripción: Como administrador quiero poder encontrar a cualquier paciente para poder consultar sus datos personales.

Validación:

* Se seleccionará el paciente a borrar de la base de datos de pacientes.
* Se puede consultar sólo los datos personales del paciente.

## **Modificar datos personales**

## **Nombre: MODIFICAR DATOS PERSONALES**

ID: H04

Descripción: Como administrador quiero poder modificar los datos de cualquier paciente en cualquier momento.

Validación:

* Se seleccionará el paciente del que se quiere modificar sus datos personales.
* Se seleccionará qué dato se quiere modificar del paciente.
  + 1. **Consultar cita**

## **Nombre: CONSULTAR CITA**

ID: H05

Descripción: Como administrador quiero poder consultar la cita de un paciente para poder saber a qué hora hay que atenderle.

Validación:

* Se puede ver las citas que hay disponibles, y reservadas para el día deseado.
  + 1. **Añadir cita**

## **Nombre: AÑADIR CITA**

ID: H06

Descripción: Como administrador quiero que cualquier paciente pueda escoger cita para que pueda ser atendido por el médico.

Validación:

* Se mostrarán las citas que están disponibles.
* Se mostrará una lista con las citas reservadas al final a modo de comprobación.
* Para que el paciente puede reservar cita deberá introducir su nombre completo, y que dicho paciente haya sido previamente registrado.
  + 1. **Borrar cita**

## **Nombre: BORRAR CITA**

ID: H07

Descripción: Como administrador quiero que cualquier paciente pueda borrar su cita para que ésta vuelva a estar disponible para otro paciente.

Validación:

* Cualquier paciente registrado podrá borrar su cita.
* La cita para que sea borrada debe haber sido reservada previamente.
* En cuanto la cita sea borrada, ésta, inmediatamente, será habilitada de nuevo.
  + 1. **Modificar cita**

## **Nombre: MODIFICAR CITA**

ID: H08

Descripción: Como administrador quiero poder modificar cualquier cita, pudiendo escoger otra cualquiera disponible.

Validación:

* Solo podrá escoger citas disponibles.
* En cuanto la cita sea modificada la anterior, será habilitada de nuevo.
  + 1. **Añadir tratamiento**

## **Nombre: AÑADIR TRATAMIENTO**

ID: H09

Descripción: Como administrador quiero añadir cualquier tratamiento a cualquier paciente.

Validación:

* Se seleccionará el paciente al quiere añadir el tratamiento de la bse de datos.
* Se introducirán los datos requeridos para que el tratamiento sea creado de forma correcta.
  + 1. **Borrar tratamiento**

## **Nombre: BORRAR TRATAMIENTO**

ID: H10

Descripción: Como administrador quiero borrar el tratamiento actual del paciente.

Validación:

* Se selecciona el paciente con su tratamiento en curso para que sea borrado.
* Una vez terminado el tratamiento este no será borrado.
* El tratamiento ya borrado se añadirá a la base de datos de Tratamientos.
  + 1. **Consultar tratamientos**

## **Nombre: CONSULTAR TRATAMIENTOS**

ID: H11

Descripción: Como administrador quiero poder consultar cualquier tratamiento ya sea en curso o finalizado.

Validación:

* Se selecciona el paciente del que se quiere consultar su tratamiento en curso.
* Se podrán consultar tratamientos ya finalizados, si el tratamiento en curso del paciente ha sido borrado anteriormente.
  + 1. **Añadir historial médico**

## **Nombre: AÑADIR HISTORIAL MÉDICO**

ID: H12

Descripción: Como administrador quiero añadir a cualquier paciente un historial.

Validación:

* Se seleccionará el paciente al que se quiere añadir un historial.
* Se introducirán los datos necesarios en el historial para que se cree correctamente.
  + 1. **Consultar historial médico**

## **Nombre: CONSULTAR HISTORIAL MÉDICO**

ID: H13

Descripción: Como administrador quiero consultar el historial médico del paciente.

Validación:

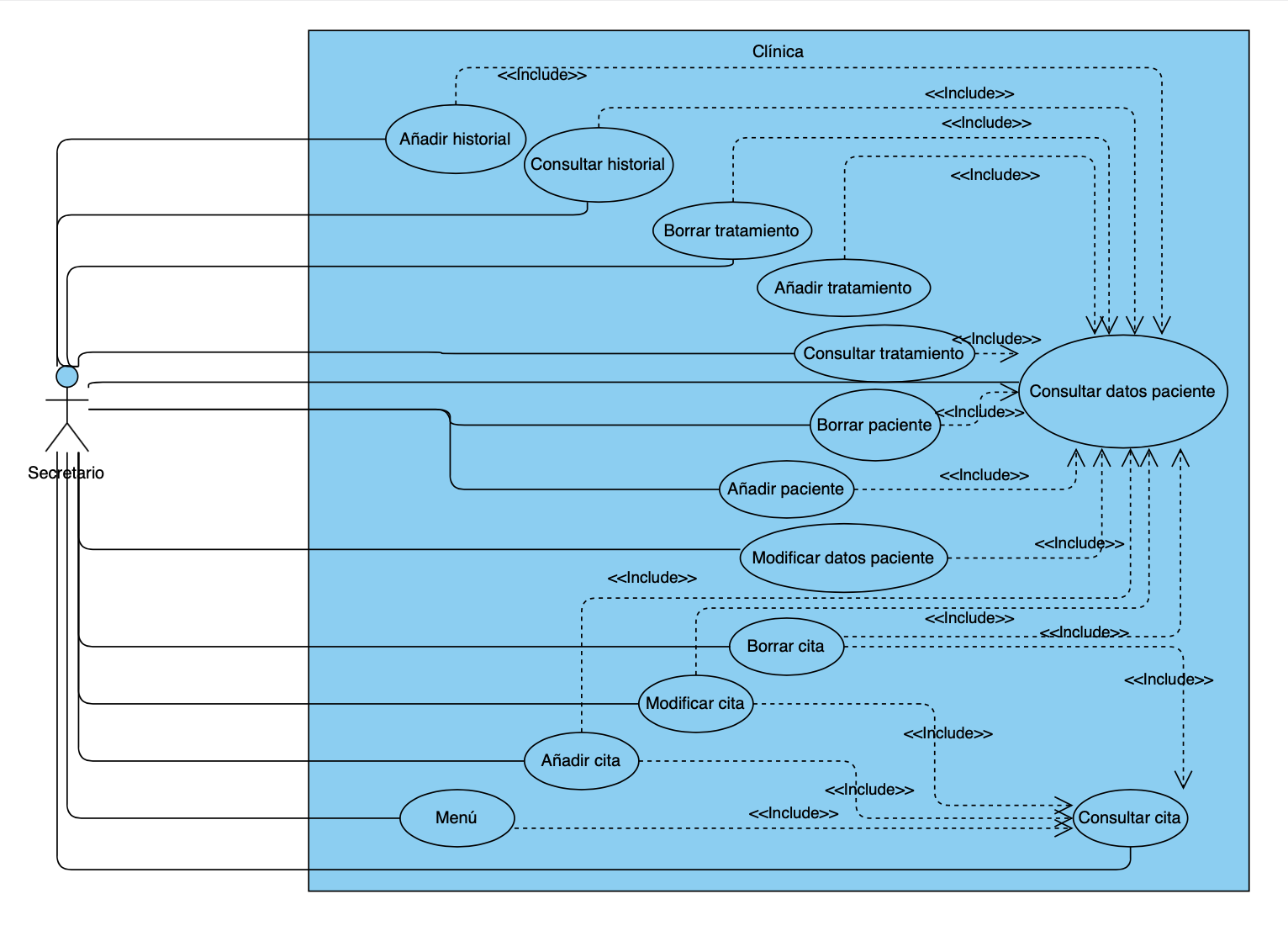
* Se seleccionará el paciente del que se quiere consultar su historial.
* Se mostrará todos los datos contenidos en el historial.
  + 1. **Menú principal**

## **Nombre: MENÚ PRINCIPAL**

ID: H14

Descripción: Como administrador quiero tener un menú para poder hacer cualquier tarea requerida.

Validación:

* Este aparecerá al iniciar la aplicación.
* Habrá diferentes menu para pacientes y citas.
* Se mostrarán las citas diarias, tanto las citas reservadas como no.

**Ilustración 1:** Diagramas de casos de uso

### **3.2 Actores**

Actores principales: Secretario, el cual tendrá acceso al software.

Actores secundarios: Pacientes, cuyos tratamiento, citas e historiales serán gestionados.

### **Especificación de los casos de uso.**

* 1. **Añadir paciente**

### **AÑADIR PACIENTE**

ID: C01

Descripción: El secretario añadirá a la base de datos un nuevo paciente.

Actores principales: secretario

Actores secundarios: Paciente

Precondiciones:

* El paciente no puede estar ya registrado.
* No puede haber dos pacientes con el mismo nombre, aunque los datos sean distintos.

Flujo principal:

1. El secretario desea añadir un nuevo paciente.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de añadir un paciente.
4. El secretario introducirá el nombre completo del paciente.
5. El secretario rellenará los campos con la información que se le pide sobre el cliente.

Postcondiciones: El cliente quedará guardado en la base de datos.

Flujos alternativos:

4.a Si al introducir el nombre del paciente al final no se quiere añadir a la base de datos, se volverá al menú principal.

5.a Si se introduce algún campo erróneo, como un código postal con 7 cifras, se pedirá de nuevo hasta que sea correcto.

* 1. **Borrar paciente**

### **BORRAR PACIENTE**

ID: C02

Descripción: El secretario borrará un paciente y todo lo referente a él de la base de datos.

Actores principales: Secretario

Precondiciones: El paciente debe estar ya registrado.

Flujo principal:

1. El secretario desea borrar un paciente de la base de datos.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de borrar un paciente.
4. El sistema mostrará una lista con todos los pacientes existentes.
5. El secretario buscará y seleccionará el paciente que desea eliminar.

Postcondiciones: El cliente dejará de existir en la base de datos y se perderán sus datos personales, historial y tratamientos en curso.

Flujos alternativos:

5.a El sistema mostrará un mensaje de confirmación.

* 1. **Consultar datos personales**

### **CONSULTAR DATOS PERSONALES**

ID: C03

Descripción: El secretario verá en el monitor la información de un paciente.

Actores principales: Secretario

Actores secundarios: Paciente

Precondiciones: El paciente debe estar ya registrado en la base de datos.

Flujo principal:

1. El secretario desea consultar los datos de un paciente.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de consultar datos personales.
4. El sistema mostrará una lista con todos los pacientes existentes.
5. El secretario buscará y seleccionará el paciente del que desea obtener información.
6. El sistema mostrará por pantalla los datos personales del paciente.
   1. **Consultar cita**

### **CONSULTAR CITA**

ID: C04

Descripción: El secretario verá una distinción entre las citas reservadas (y quien tiene dicha cita) y las disponibles para poder ser reservadas.

Actores principales: secretario

Flujo principal:

1. El secretario desea consultar las citas de un día concreto.
2. El secretario selecciona el menú de citas.
3. El secretario selecciona la opción de consultar cita.
4. El sistema preguntará por el día que quiere consultar las citas.
5. El secretario introducirá el día y el mes que quiere conocer.
6. El sistema mostrará una lista con todos las citas disponibles y otra lista con las que ya están reservadas, junto con en nombre de a quién le pertenece tal cita.

Flujos alternativos:

5.a Se validará la fecha.

* 1. **Añadir cita**

### **AÑADIR CITA**

ID: C05

Descripción: El secretario reservará una cita para un paciente.

Actores principales: secretario

Actores secundarios: Paciente

Precondiciones: El paciente debe estar registrado en la base de datos y la cita debe estar disponible.

Flujo principal:

1. El secretario desea reservar una cita para un paciente.
2. El secretario selecciona el menú de citas.
3. El secretario selecciona el menú de reserva.
4. El sistema preguntará el día que se quiere la cita.
5. El secretario introduce la fecha en la cual quiere la cita.
6. El sistema mostrará por pantalla las citas que están disponibles para ese día.
7. El secretario seleccionará la cita que quiere reservar.
8. El secretario introducirá el nombre completo de la persona a la que se le reserva la cita.
9. El sistema mostrará las citas disponibles y reservadas actualizadas

Postcondiciones: La cita reservada ya no estará disponible.

Flujos alternativos:

4.a Se validará la fecha hasta que sea válida.

7.a Validará que la cita introducida esté libre.

8.a Si el nombre no pertenece a un paciente registrado, mostrará un mensaje de error y terminará la instrucción

* 1. **Borrar cita**

### **BORRAR CITA**

ID: C06

Descripción: El secretario borrará una cita reservada para un paciente.

Actores principales: secretario

Precondiciones: La cita debe estar ya reservada.

Flujo principal:

1. El secretario desea eliminar una cita para un paciente.
2. El secretario selecciona el menú de citas.
3. El secretario selecciona la opción de borrar cita.
4. El sistema preguntará la fecha de la cita que se quiere borrar.
5. El secretario introduce el día en la que está la cita.
6. El sistema mostrará todas las citas reservadas ese día.
7. El secretario elegirá la cita que quiere borrar.
8. La cita eliminada estará disponible nuevamente

Postcondiciones: La cita pasará a estar disponible de nuevo.

Flujos alternativos:

5.a Se validará que la fecha sea válida.

7.a Se validará que la hora y el minuto sean correctos

* 1. **Actualizar historial**

### **ACTUALIZAR HISTORIAL**

ID: C07

Descripción: El secretario añadirá nueva información al historial de un paciente.

Actores principales: secretario

Flujo principal:

1. El secretario desea añadir nueva información al historial de un paciente.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de historial.
4. El secretario selecciona la opción de añadir nueva información al historial.
5. El sistema mostrará una lista con todos los pacientes existentes.
6. El secretario seleccionará el paciente al que quiere añadir información al historial.
7. El secretario introducirá la información nueva al historial.

Postcondiciones: La información se añadirá y esta no podrá ser modificada posteriormente.

Flujos alternativos:

5.a Si no hay pacientes registrados, no muestra nada y acaba la instrucción.

* 1. **Consultar historial**

### **CONSULTAR HISTORIAL**

ID: C08

Descripción: El secretario verá en el monitor el historial de un paciente.

Actores principales: Secretario

Actores secundarios: Paciente

Precondiciones: El paciente debe estar ya registrado en la base de datos.

Flujo principal:

1. El secretario desea consultar los datos de un paciente.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de historial.
4. El secretario seleccionará la opción de consultar historial.
5. El sistema mostrará una lista con todos los pacientes existentes.
6. El secretario buscará y seleccionará el paciente del que desea obtener información.
7. El sistema mostrará por pantalla todo el historial del paciente.

Flujos alternativos:

5.a Si no hay pacientes registrados, no mostrará nada y la instrucción acabará.

7.a Si no tiene nada registrado en el historial, mostrará un espacio de pantalla en blanco.

* 1. **Consultar tratamiento**

### **CONSULTAR TRATAMIENTO**

ID: C09

Descripción: El secretario verá en el monitor el tratamiento de un paciente.

Actores principales: secretario

Actores secundarios: Paciente

Precondiciones: El paciente debe estar ya registrado en la base de datos.

Flujo principal:

1. El secretario desea consultar los datos de un paciente.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de tratamiento.
4. El secretario selecciona la opción de consultar.
5. El sistema mostrará una lista con todos los pacientes existentes.
6. El secretario buscará y seleccionará el paciente del que desea obtener información.
7. El sistema mostrará por pantalla el tratamiento del paciente.

Flujos alternativos:

7.a Si no hay tratamiento, dará diferentes opciones para realizar.

* 1. **Añadir tratamiento**

### **AÑADIR TRATAMIENTO**

ID: C10

Descripción: El secretario añadirá un nuevo tratamiento a un paciente.

Actores principales: secretario

Actores secundarios: Paciente

Precondiciones: El paciente debe estar ya registrado en la base de datos.

Flujo principal:

1. El secretario desea añadir un tratamiento a un paciente.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de tratamiento.
4. El secretario selecciona la opción de añadir tratamiento.
5. El sistema mostrará una lista con todos los pacientes existentes.
6. El secretario buscará y seleccionará el paciente al que desea añadir un tratamiento.
7. El secretario escribirá la información nueva al tratamiento.

Flujos alternativos:

7.a Se validará que la fecha introducida es correcta.

* 1. **Borrar tratamiento**

### **BORRAR TRATAMIENTO**

ID: C11

Descripción: El secretario borrará el tratamiento a un paciente.

Actores principales: secretario

Actores secundarios: Paciente

Precondiciones:

1. El paciente debe estar ya registrado en la base de datos.
2. El paciente debe tener un tratamiento actualmente.

Flujo principal:

1. El secretario desea añadir un tratamiento a un paciente.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de tratamiento.
4. El secretario selecciona la opción de borrar tratamiento.
5. El sistema mostrará una lista con todos los pacientes existentes.
6. El secretario buscará y seleccionará el paciente del que desea borrar el tratamiento.
7. El sistema mostrará un mensaje de confirmación para borrar el tratamiento.
8. El sistema borrará todo el tratamiento de ese paciente.

Postcondiciones:

8.a. A pesar de haber borrado todo el tratamiento, el sistema creará una base de datos de seguridad con lo que ha borrado en caso de querer consultarse de nuevo o recuperarlo.

Flujo alternativo

7.a. El sistema mostrará un mensaje de confirmación a la hora de borrar el tratamiento.

* 1. **Modificar cita**

### **MODIFICAR CITA**

ID: C12

Descripción: El secretario modificara la cita de un paciente.

Actores principales: secretario

Actores secundarios: Paciente

Precondiciones:

1. El paciente debe estar ya registrado en la base de datos.
2. El paciente debe tener al menos una cita reservada.

Flujo principal:

1. El secretario desea modificar una cita de un paciente.
2. El secretario selecciona el menú de citas.
3. El secretario selecciona la opción de modificar citas.
4. El sistema preguntará en qué día se encuentra la cita del paciente.
5. El secretario introducirá el día.
6. El sistema mostrará las citas reservadas ese día.
7. El secretario seleccionará la cita que quiere anular.
8. El sistema preguntará en qué día quiere la nueva cita.
9. El usuario introducirá la fecha en la que quiere la nueva cita.
10. El sistema mostrará las citas disponibles ese día.
11. El secretario elegirá la nueva cita para el paciente.

Flujos alternativos:

5.a Se validará que la fecha sea válida.

9.a Se validará que la fecha sea válida.

* 1. **Modificar datos personales**

### **MODIFICAR DATOS PERSONALES**

ID: C13

Descripción: El secretario modificara los datos personales de un paciente.

Actores principales: secretario

Actores secundarios: Paciente

Precondiciones:

1. El paciente debe estar ya registrado en la base de datos.

Flujo principal:

1. El secretario desea modificar los datos de un paciente.
2. El secretario selecciona el menú de pacientes.
3. El secretario selecciona la opción de modificar datos personales.
4. El sistema mostrará una lista con todos los pacientes existentes.
5. El secretario buscará y seleccionará el paciente del que desea modificar los datos.
6. El sistema preguntará qué campo quiere modificar.
7. El secretario elegirá qué dato quiere modificar.
8. El sistema pedirá el nuevo dato.
9. El secretario introducirá la nueva información.
   1. **Menú principal**

### **MENÚ PRINCIPAL**

ID: C14

Descripción: El secretario usará el programa.

Actores principales: secretario

Flujo principal:

1. El secretario desea usar el programa.
2. El secretario abre el programa.
3. El sistema mostrará las citas reservadas y las disponibles del día en curso.
4. El sistema mostrará un menú para la elección de acciones sobre citas o pacientes.
5. El secretario elegirá qué quiere hacer.
6. **Validación de los casos de uso con los requisitos funcionales.**

La numeración utilizada para los requisitos funcionales será la misma a la asignada en el apartado 2.3 Requisitos funcionales

La numeración utilizada para los casos de uso será el ID de los mismos

RF: Requisitos funcionales (Fila superior).

CU: Casos de uso (Columna izquierda).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **RF**  **CU** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
| **C01** |  |  |  |  |  |  | X |  |  |  |  |  |  |  |  |  |
| **C02** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |  |
| **C03** | **X** |  | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C04** |  |  |  |  | **X** |  |  |  |  |  |  |  |  |  |  |  |
| **C05** |  |  |  |  |  |  |  |  |  | **X** |  |  |  |  |  |  |
| **C06** |  |  |  |  |  |  |  |  |  |  |  |  | **X** |  |  |  |
| **C07** |  |  |  |  |  |  |  |  | **X** |  |  |  |  |  |  |  |
| **C08** |  |  |  |  | **X** |  |  |  |  |  |  |  |  |  |  |  |
| **C09** |  | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C10** |  |  |  |  |  |  |  | **X** |  |  |  |  |  |  |  |  |
| **C11** |  |  | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **C12** |  |  |  |  | **X** |  |  |  |  |  | **X** |  |  |  |  |  |
| **C13** |  |  |  |  |  | **X** |  |  |  |  |  |  |  |  |  |  |
| **C14** |  |  |  |  | **X** |  |  |  |  |  |  |  |  | **X** |  |  |

**Tabla 1:** Validación de los casos de uso con los requisitos funcionales

1. **Representación, descripción y especificación de las entidades de información que manejará el sistema**
   1. **Tratamiento**

**Clase: Tratamiento**

Esta clase contiene los métodos de las operaciones sobre los tratamientos de los pacientes.

* Datos:
  + - comienzo tratamiento

Tipo: string

Esta variable contendrá la fecha de inicio del tratamiento.

* duracion\_tratamiento

Tipo: string

Esta variable contendrá la duración del tratamiento.

* dosis

Tipo: string

Esta variable contendrá la dosis del medicamento.

* nombre\_medicamento

Tipo: string

Esta variable contendrá el nombre del medicamento recetado.

* Tipo: entero

Esta variable guardará el año del sistema.

* month\_sistema

Tipo: entero

Esta variable guardará el mes del sistema.

* day\_sistema

Tipo: entero

Esta variable guardará el día del sistema.

* Métodos:
  + see\_tratamiento

Este método permitirá consultar el tratamiento en curso del paciente seleccionado.

* + add\_tratamiento

Este método actualizará y añadirá el tratamiento de un paciente.

* delete\_tratamiento

Este método borrará el tratamiento actual y no dejará ningún tratamiento.

* comprobar\_fichero

Esta función comprobará la existencia de la base de datos donde están los pacientes registrados.

* choose\_paciente

Esta función seleccionará el paciente sobre el que se quiere operar.

* + - confirmafecha

Esta función validará que una fecha sea válida.

**Clase: Historial**

Esta clase contiene los métodos de las operaciones sobre el historial de los pacientes.

* Datos:
* lista\_pacientesH\_

Tipo: Lista de string

Esta variable contendrá la lista de pacientes que hay registrados.

* Métodos:
* consultar\_historial

Este método permitirá consultar el historial médico del paciente que se desee.

* add\_historial

Este método añadirá nueva información al historial de un paciente.

* cargar\_lista\_pacientes

Este método devolverá una lista con todos los pacientes existentes en la base de datos.

**Clase: Paciente**

Esta clase contiene todos los métodos de las operaciones sobre pacientes.

* Datos:
* nombre\_

Tipo: string

Esta variable contendrá el nombre del paciente.

* apellido1\_

Tipo: string

Esta variable contendrá el primer apellido del paciente.

* apellido2\_

Tipo: string

Esta variable contendrá el segundo apellido del paciente.

* direccion

Tipo: string

Esta variable contendrá la dirección del paciente.

* codigo\_Postal

Tipo: entero

Esta variable contendrá el código postal del paciente.

* ciudad

Tipo: string

Esta variable contendrá la ciudad del paciente.

* provincia

Tipo: string

Esta variable contendrá la provincia del paciente.

* DNI

Tipo: string

Esta variable contendrá el DNI del paciente.

* nombre\_concatenar

Tipo: string

Esta variable ayudara a concatenar el nombre de un paciente.

* Métodos:
  + - comprobar\_fichero

Este método comprobará la existencia de la base de datos dónde están los pacientes registrados.

* + - add\_paciente

Este método añadirá un nuevo paciente a la base de datos.

* + - delete\_paciente

Este método borrará un paciente y toda su información de la base de datos.

* + - choose\_paciente

Este método comprobará la existencia de la base de datos donde están los pacientes registrados.

* + - modifica\_pacientes

Esta función auxiliar actualizará la base de datos tras algunas operaciones como añadir un paciente.

* + - choose\_datos\_paciente

Esta función seleccionará qué parte se quiere modificar de los datos de un paciente.

* + - consultar\_datos\_personales

Esta función mostrará los datos personales de un paciente.

* + - modificar\_datos\_paciente

Esta función modificará los datos personales de un paciente.

* + - set\_codigoPostal

Este método auxiliar ayudará a comprobar que el código postal es correcto.

**Clase: Citas**

Esta clase contiene los métodos de las operaciones de citas.

* Datos:
  + dia\_sistema

Tipo: entero

La variable guardará el día del sistema.

* + mes\_sistema

Tipo: entero

Esta variable guardará el mes del sistema.

* + anio\_sistema

Tipo: entero

La variable guardará el año del sistema.

* + hora\_reserva

Tipo: entero

La variable guardará la hora de la cita reservada.

* + minuto\_reserva

Tipo: entero

La variable guardará los minutos de la cita reservada.

* + dia\_reserva\_

Tipo: string

La variable guardará el dia que se ha reservado la cita.

* mes\_reserva\_

Tipo: string

La variable guardará el mes que se ha reservado la cita.

* anio\_reserva\_

Tipo: string

La variable guardará el año que se ha reservado la cita.

* nombre

Tipo: string

La variable guardará el nombre del paciente.

* apellidos

Tipo: string

La variable guardará los apellidos del paciente.

* lista\_

Tipo: lista de struct

La variable guardará lo que se saque de la base de datos en cuanto a las citas (libres o reservadas).

* Métodos:
  + getDia\_reserva

Este método sirve para usar una variable privada.

* + setDia\_reserva

Este método sirve para dar valor a una variable privada.

* + getMes\_reserva

Este método sirve para usar una variable privada.

* + SetMes\_reserva

Este método sirve para dar valor a una variable privada.

* + getAnio\_reserva

Esta función sirve para usar una variable privada.

* + setAnio\_reserva

Esta función sirve para dar valor a una variable privada.

* menuCitas

Esta función hará de menú entre las distintas operaciones de sobre las citas de un paciente.

* confirmafecha

Esta función validará que una fecha sea válida.

* aniadir\_citas

Esta función ayudará a reservar cita, haciendo comprobaciones y guardando los valores en las variables de reserva.

* imprimiCitasLibres

Esta función mostrará por pantalla las citas libres de un día concreto.

* imprimirCitas

Esta función mostrará por pantalla citas disponibles y reservadas de un dia concreto.

* reservaCitas

Esta función permitirá reservar una cita que este libre.

* crear\_fichero\_cl

Esta función crea el fichero de citas libres.

* consultarCitas

Esta función permitirá consultar las citas reservadas de un día concreto.

* modificar\_Cita

Esta función permitirá cancelar una cita para sustituirla por otra.

* eliminarCita

Esta función permitirá borrar una cita ya reservada.

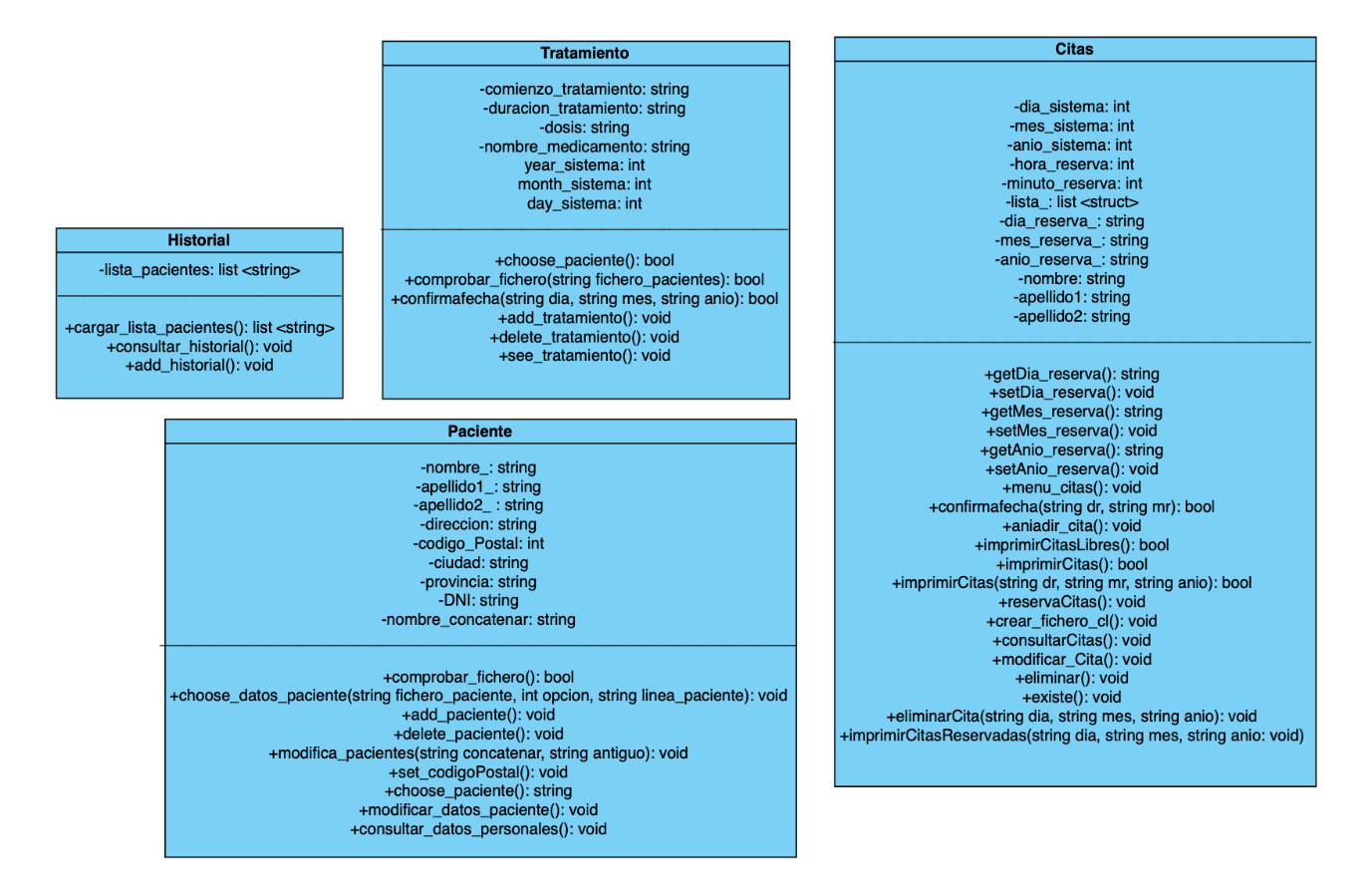
* eliminar

Esta función ayudará en el proceso de eliminar citas:

* existe

Esta función comprobará si existe algún paciente en la base de datos.

* imprimirCitasReservadas

Esta función mostrará las citas reservadas de un día en concreto.

**Ilustración 2** Modelado de las clases

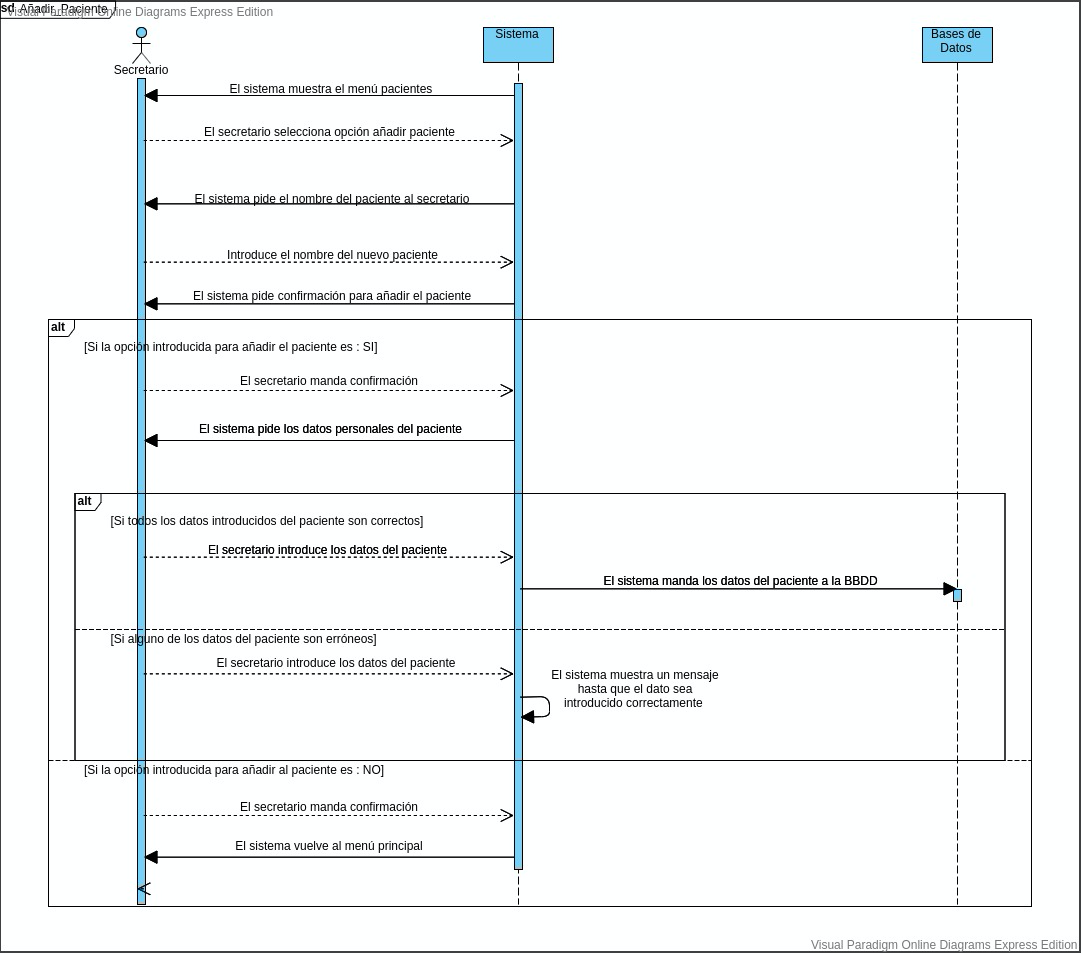
1. **Validación de las clases con los casos de uso**

**Clases: representadas por su nombre (Fila superior).**

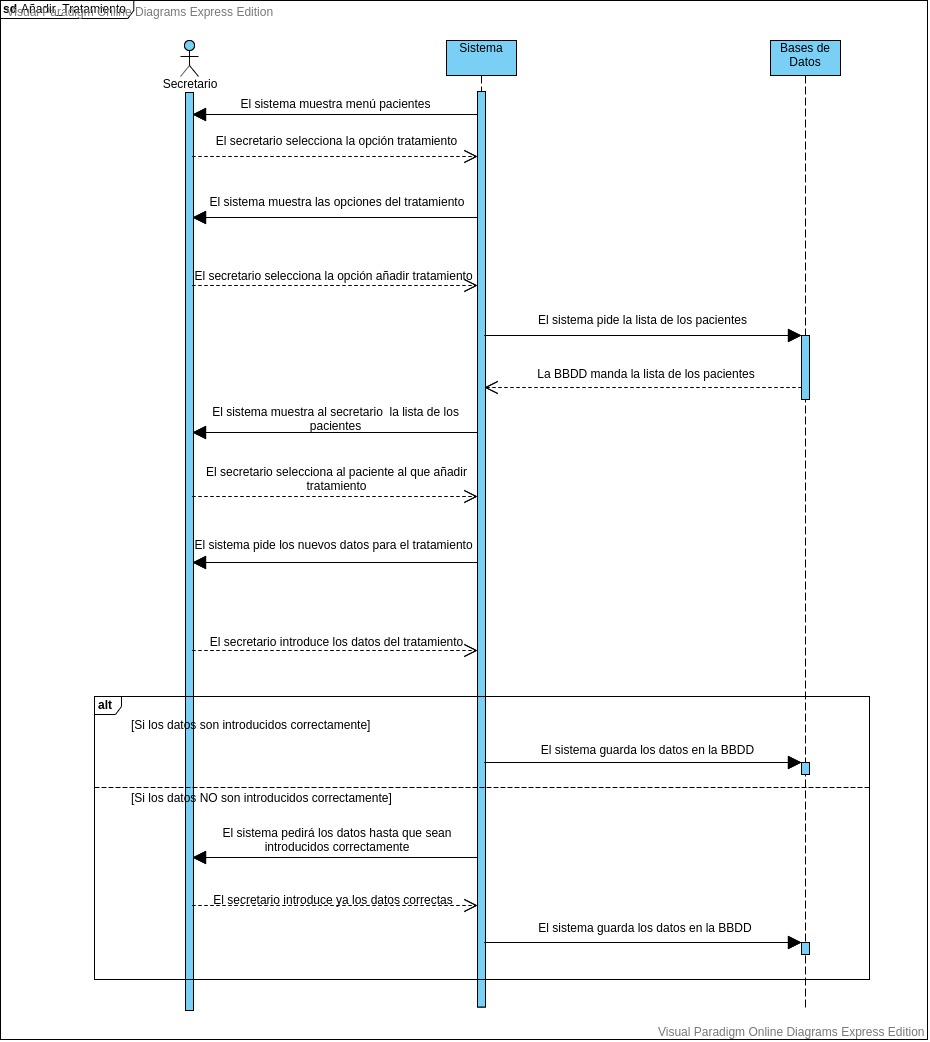
**Casos de uso (CU): representados por su id(columna izquierda)**

**Tabla 2:** Validación de las clases con los casos de uso

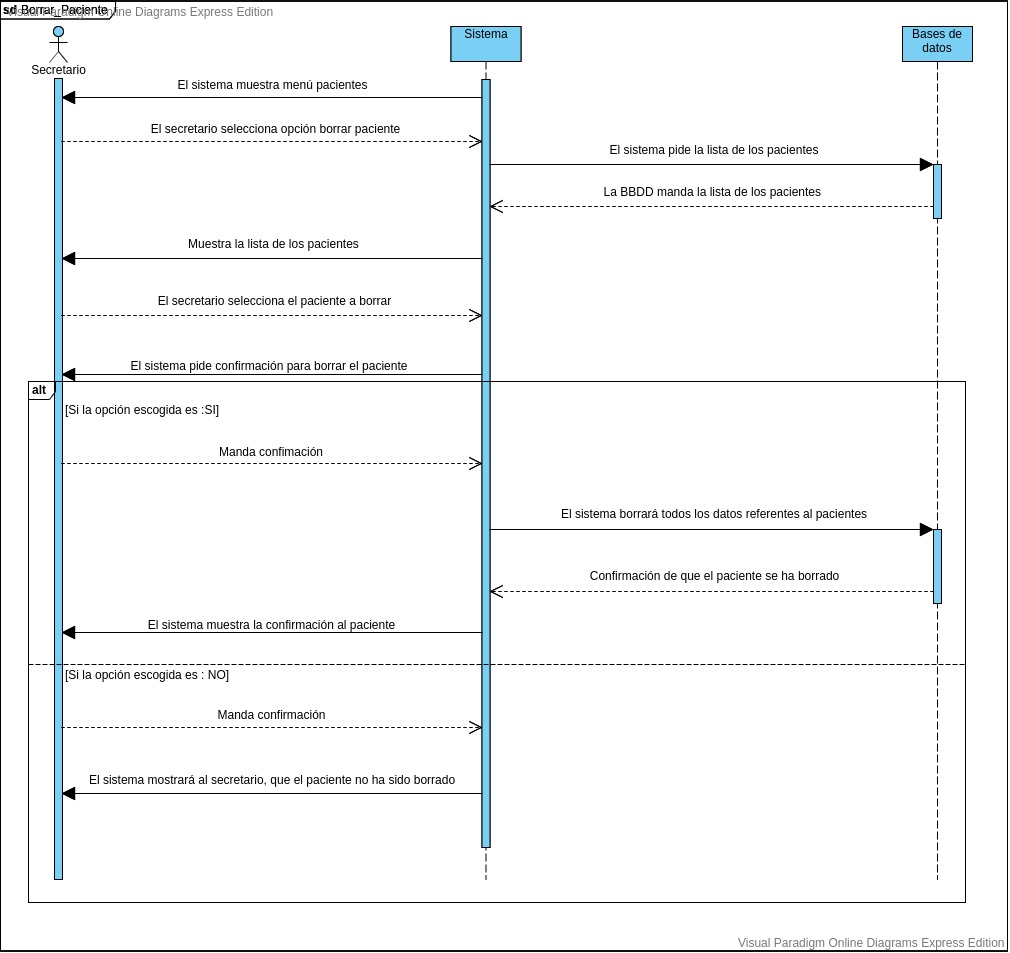
AÑADIR\_PACIENTE



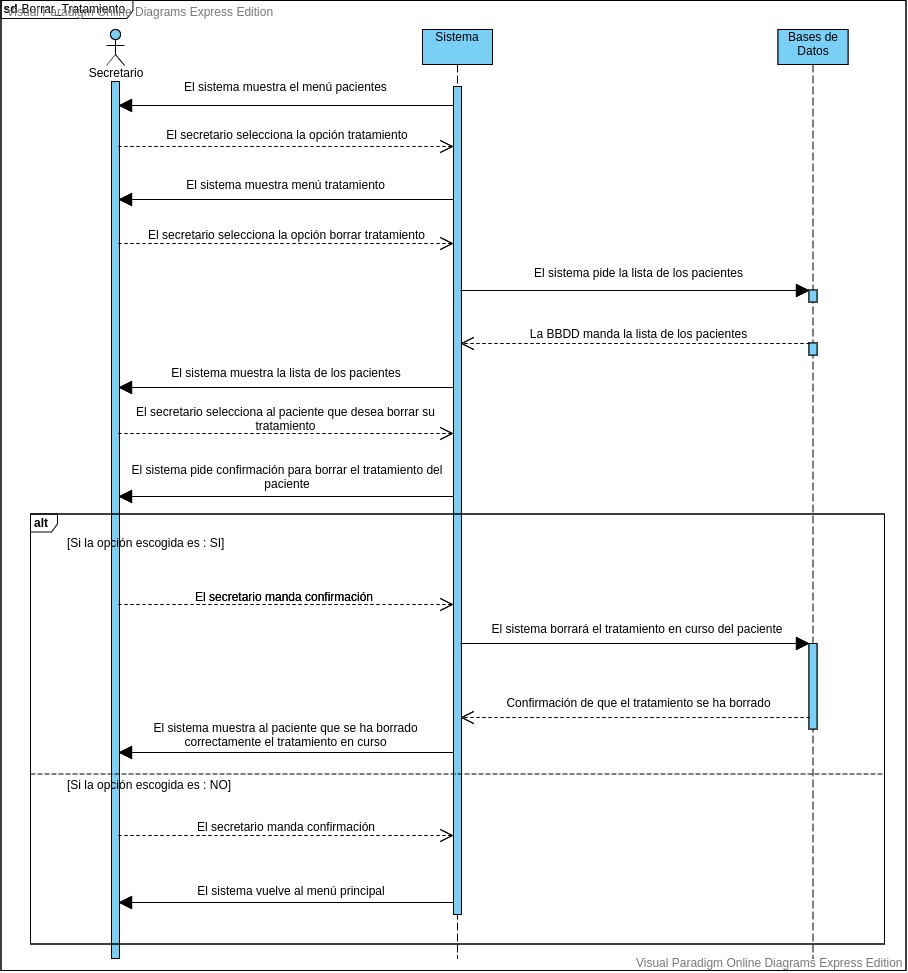
AÑADIR\_TRATAMIENTO



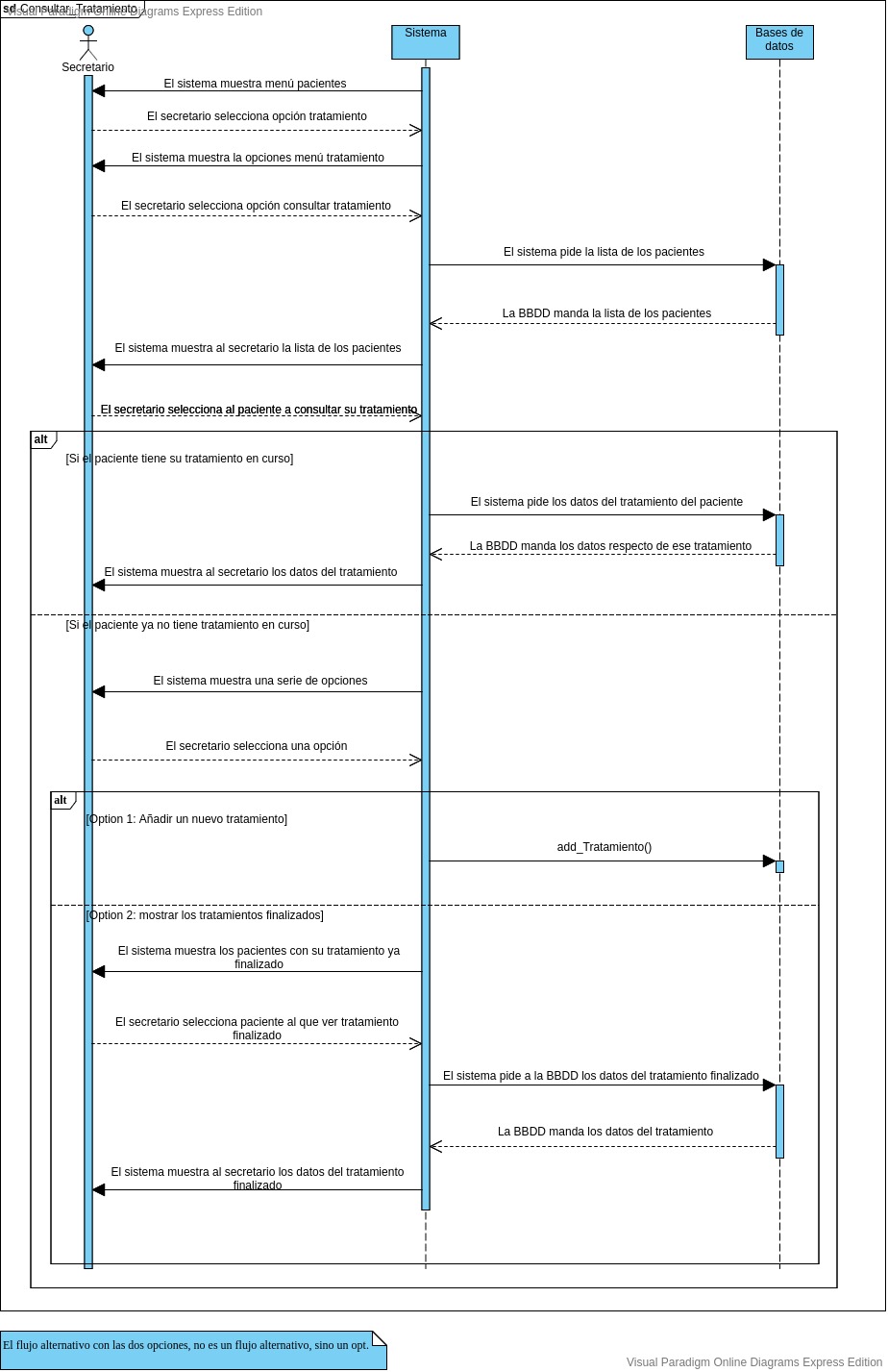
BORRAR\_PACIENTE



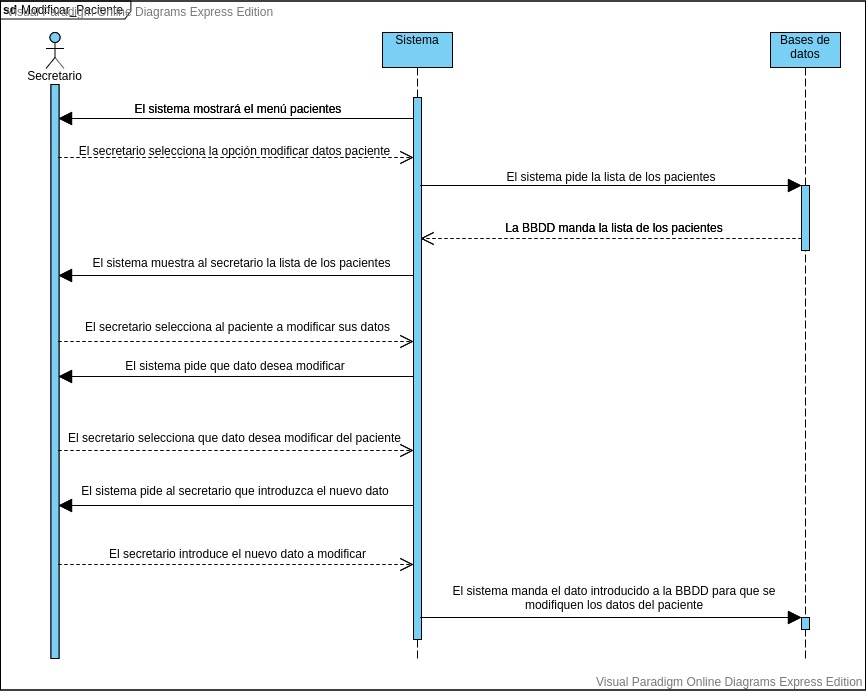
BORRAR\_TRATAMIENTO



CONSULTAR\_TRATAMIENTO



MODIFICAR\_PACIENTE



CONSULTAR\_DATOS\_PACIENTE

