

Clínica médica

SoftWave

03/09/2020

Diego José Gómez Aguilar Cristián Alberto Sánchez Salido Sergey Beryuza

A- Especificación del Sistema	3
1- Introducción	3
2- Descripción del problema	3
2.1- Organización de Pacientes	4
2.2- Organización de Tratamientos	4
2.3- Organización de las Citas	4
2.4- Organización de Historial Médico	5
3- Solución propuesta	5
B-Especificación de requisitos	5
1- Introducción	5
2- Dados que pertenecen a la aplicación.	6
2.1- Datos del Paciente	6
2.2- Datos de Tratamiento	6
2.3- Datos de Cita	6
2.4- Datos del Historial Médico	7
3- Requisitos funcionales	7
4- Requisitos no funcionales	7
C- Actores, casos de uso y priorización de los mismos	8
1- Historias de usuario	8
1.1- Añadir Paciente	8
1.2- Buscar Paciente	9
1.3- Modificar Pacientes	10
1.4- Borrar Paciente	10
1.5- Añadir Cita	11
1.6- Buscar Cita	12
1.7- Cancela Cita	12
1.8- Modificar Cita	13
1.9-Mostrar Agenda	14
1.10- Añadir un tratamiento	14
1.11- Consulta Tratamiento	15
1.12- Borrar Tratamiento	15
1.13- Modificar Tratamiento	16
1.14- Consultar Historial Médico	17
Diagrama de Casos de Uso	17
2- Actores	18
D-Especificación de los casos de Uso	18
1- Añadir Cita	18
2- Buscar Cita	19

	3- Modificar cita	21
	4- Borrar Cita	22
	5- Añadir Paciente	23
	6- Buscar Paciente	24
	7- Modificar Paciente	25
	8- Borrar Paciente	26
	9- Añadir Tratamiento	28
	10- Consultar Tratamiento	29
	11- Borrar Tratamiento	30
	12- Modificar Tratamiento	31
	13- Consultar Historial Médico	32
	14- Mostrar Agenda	34
E-	Validación de los casos de uso con los requisitos funcionales	35
	Representación, descripción y especificación de las entidades de información qu	
m	anejará el sistema Clase: Cita	36
	Clase: Paciente	36 37
	Clase: Tratamiento	38
G-	· Validación de las clases con los casos de uso	40
H-	- Diagramas de Secuencias	41
	1- Añadir Paciente	41
	2- Añadir Cita	42
	3- Añadir Tratamiento	43
	4- Buscar Paciente	44
	5- Buscar Cita	45
	6- Borrar Tratamiento	46
	7- Consultar Tratamiento	47
	8- Consultar Historial Médico	48
	9- Cancelar Cita	49
	10- Eliminar Paciente	50
	11- Modificar Paciente	51
	12- Modificar Tratamiento	52
	13- Mostrar Cita	53
	14- Mostrar Agenda	54
I-	Implementación, refinamiento y pruebas usando la metodología SCRUM con las	
	eraciones que sean necesarias.	55

A- Especificación del Sistema

1- Introducción

Aunque muchos no lo crean, la planificación y especificación de un programa antes de empezar a programarlo es fundamental para optimizar y consolidar el trabajo.

Esto facilita enormemente la creación de dicho programa, ayuda a solucionar los problemas complejos, reduce el conflicto entre los programadores,etc

Sin esta información se crearían múltiples problemas, retrasos muy grandes en el proyecto y pérdidas significativas de dinero. Desde nuestra empresa queremos evitar esto, haciendo el trabajo duro de recopilar y crear esta información.

2- Descripción del problema

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

Esta empresa quiere que este programa organiza de forma eficaz y sencilla los pacientes, las citas y los tratamientos. También se desea

que al iniciar, lo primero que se ejecute es una pequeña agenda con las citas próximas.

Para esto hemos optado por un simple software que se ejecuta por línea de comandos desde la terminas, el cual se maneja enteramente en este. Ahora profundicemos en los subapartados del problema.

2.1- Organización de Pacientes

Creemos que la gestión de pacientes es la parte más importante y fundamental del programa.

Para ellos necesitamos una solución, que será un objeto, para guardar todos los datos relacionados con ellos pero que seamos capaces de acceder a ellos siempre y cuando fuera necesario

2.2- Organización de Tratamientos

Esta gestión es similar a la del paciente, pero para que se pueda crear, manejar, borrar, etc hemos pensado que fuera obligatorio ques existiera un paciente vinculado, principalmente para que tuviera más sentido.

2.3- Organización 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.

2.4- Organización de Historial Médico

Por último, y aunque no se planteó como principal problema, creemos necesario crear un Historial Médico donde se almacenan los datos de cada pacientes como los tratamientos y las citas que les corresponde.

3- Solución propuesta

Una vez analizado el problema, y después de una reunión entre todos los participantes de este proyecto hemos decidido utilizar C++ aunque no fuera nuestra primera opción. Hemos decidido utilizar este lenguaje por muchas razones pero la principal es por la facilidad y eficiencia en la utilización de la Programación Orientada a Objetos. Por últimos, hemos decidido usar el sistema operativo LINUX aunque este mas adelante y si el cliente lo prefiere se podría portar a otros sistemas operativos.

B-Especificación de requisitos

1- Introducción

Tras la reunión entre el cliente y nuestra empresa y la reunión entre los miembros de esta, hemos decidido establecer los siguientes requisitos

Estos requisitos son tanto funcionales como no funcionales y son necesarias para saber como debe de funcionar el software que el cliente quiere.

2- Dados que pertenecen a la aplicación.

2.1- Datos del Paciente

- Nombre y Apellidos
- DNI
- Edad
- Dirección

2.2- Datos de Tratamiento

- Nombre del Tratamiento
- Duración

2.3- Datos de Cita

- DNI
- Hora
- Dia
- Mes
- Año
- Motivo

2.4- Datos del Historial Médico

Contiene todos los datos anteriores

3- Requisitos funcionales

- Añadir nuevo paciente
- Buscar paciente
- Modificar paciente
- Borrar paciente
- Añadir cita
- Busca cita
- Cancelar cita
- Modificar cita
- Mostrar agenda
- Añadir tratamiento
- Consultar tratamiento
- Borrar tratamiento
- Modificar tratamiento
- Consultar historial

4- Requisitos no funcionales

- La aplicación utilizará el sistema operativo Linux.
- Lenguaje utilizado: C++.
- Evitar que se solapen las citas.

- Evitar que el historial médico y los tratamientos finalizados de los pacientes sea borrado.
- Mostrar mensaje de confirmación al eliminar cualquier dato de algún paciente.
- Evitar que se pueda trabajar con pacientes no registrados
- Mostrar las citas al ejecutar el programa

C- Actores, casos de uso y priorización de los mismos

Antes de de comenzar con los casos de uso, es mejor especificar las historias de usuario. Estas son

representaciones de requisitos, utilizando un lenguaje común del usuario, haciendo de estas, una forma natural de análisis funcional.

1- Historias de usuario

1.1- Añadir Paciente

□ ID:01

☐ Descripción:

Como administrador quiero añadir un nuevo paciente, con todos los datos que este tiene, a la base de datos de todos los pacientes

■ Validación:

- > Se tiene que rellenar todos los campos para añadir un paciente
- > Se debe siempre poder añadir un paciente siempre y cuando se cumpla la anterior validación
- La opción añadir usuario debe de devolver al menú cuando finalize

1.2- Buscar Paciente

□ ID:02

☐ Descripción:

Como administrador quiero buscar un paciente en la base de datos y acceder a toda la información relacionada a el

☐ Validación:

- ➤ El paciente debe de estar registrado con anterioridad.
- > Este debe de mostrar sus citas,tratamientos y datos.
- ➤ El programa debe de acceder de forma indirecta a esta función sin la necesidad que el administrador seleccione una función que la necesite.
- > La opción Buscar Paciente por nombre y apellidos debe de devolver al menú cuando finalize.

1.3- Modificar Pacientes
□ ID:03
 Descripción: Como administrador quiero modificar los datos de un paciente, eliminando así la posibilidad de datos incorrectos (en la base de datos) sobre el propio paciente
 Validación: Se debe poder realizar la previa búsqueda del usuario para su posterior modificación Se debe poder modificar un paciente siempre que se cumpla la anterior validación La opción modificar usuario debe de devolver al menú cuando finalize
1.4- Borrar Paciente
□ ID:04
☐ Descripción:

Como administrador quiero eliminar los datos de un paciente, liberando así espacio para la posible incorporación de nuevos pacientes

- ☐ Validación:
- > Se debe poder realizar la previa búsqueda del usuario para su posterior eliminación
- > Debe pedirse previa confirmación para el borrado
- > Se debe poder eliminar un paciente siempre que se cumpla la anterior validación
- > La opción eliminar usuario debe de devolver al menú cuando finalize

1.5- Añadir Cita

□ ID:05

☐ Descripción:

Como administrador deseo añadir una nueva cita para la correcta organización de la agenda

☐ Validación:

- >> Se debe poder acceder a las opciones de la agenda
- > Se debe introducir datos correctos (sin posibilidad de fechas ya ocurridas)
- > La opción añadir cita debe de devolver al menú cuando finalize

1.6- Buscar Cita
□ ID:10
 Descripción: Como administrador quiero ver las citas que he tenido o voy a tener de cada paciente o de la fecha introducida
 Validación: Siempre se debe mostrar la opción de búsqueda por nombre o por fecha. Si el nombre o la fecha no existe, el programa tiene que decir el error y salirse. Se tiene que mostrar todos los datos de la cita. La opción buscar cita debe de devolver al menú cuando finalize
1.7- Cancela Cita
□ ID:11

Como administrador quiero cancelar una cita que se encuentra programada en la base de datos de citas

☐ Descripción:

- ☐ Validación:
- > Si el administrador intenta entrar en cancelar cita y no hay citas para el futuro, el programa debe de decirlo y salir.
- > No se debe de poder cancelar citas que ya han sido realizadas.
- Una vez eliminada, no debe de quedar rastro de esta en la base de datos de citas.
- > La opción cancelar cita debe de devolver al menú cuando finalize.

1.8- Modificar Cita

□ ID:13

■ Descripción:

Como administrador quiero modificar una cita ya creada anteriormente

- ☐ Validación:
- > La cita a modificar debe de existir en la base de datos.
- > Se debe de rellenar todos los datos para que se modifique la cita.
- > La cita solo se puede modificar si es para el futuro, es decir, si ya no se ha pasado la fecha de esta.
- ➤ La opción Modificar Cita debe de devolver al menú cuando finalize.

1.9-Mc	ostrar Agenda
٥	ID:07
0	Descripción: Como administrador quiero ver la agenda donde se muestra las citas que tengo
>	Validación: Esta función se ejecuta automáticamente cuando se enciende el programa. Muestra solo las citas futuras. Esta función se conecta con la base de datos de citas. La opción Mostrar Agenda debe de devolver al menú cuando finalize.
1.10- A	Añadir un tratamiento
	ID:06
0	Descripción: Como administrador quiero añadir un tratamiento a un paciente que ya está registrado en la base de datos
	Validación: El paciente debe de estar registrado con anterioridad.

- > El tratamiento debe de incluirse a la base de datos con todos los campos rellenos.
- > El tratamiento también se incluye en la base de dato del historial médico.
- > La opción Añadir un tratamiento a un paciente debe de devolver al menú cuando finalize.

1.11- Consulta Tratamiento

□ ID:09

☐ Descripción:

Como administrador deseo añadir una nueva cita para la correcta organización de la agenda

☐ Validación:

- >> Se debe poder acceder a las opciones de la agenda
- > Se debe introducir datos correctos (sin posibilidad de fechas ya ocurridas)
- > La opción añadir cita debe de devolver al menú cuando finalize.

1.12- Borrar Tratamiento

□ ID:12

Descripción:

Como administrador quiero borrar un tratamiento a un paciente que ya está registrado en la base de dato

- ☐ Validación:
- > El paciente debe de estar registrado con anterioridad.
- > El paciente debe de tener al menos un tratamiento.
- > El tratamiento debe de eliminarse sin dejar rastro en la base de datos de tratamientos con todos los campos rellenos.
- > La opción Borrar Tratamiento a un paciente debe de devolver al menú cuando finalize.

1.13- Modificar Tratamiento

□ ID:14

Descripción:

Como administrador quiero modificar un tratamiento que le corresponde a un paciente

☐ Validación:

- > El tratamiento a modificar debe de existir en la base de datos.
- > Se debe de rellenar todos los datos para que se modifique el tratamiento.
- > La opción Modificar Tratamiento debe de devolver al menú cuando finalize.

1.14- Consultar Historial Médico

□ ID:08

☐ Descripción:

Como administrador quiero consultar el historial de cada paciente introduciendo el DNI en el cual se muestre todos los datos acerca de dicha persona

■ Validación:

- > El DNI introducido debe de corresponder a un paciente registrado anteriormente.
- > Si el DNI es erróneo, el programa se lo comunica al administrador y se sale.
- > Debe de aparecer todos los datos acerca del paciente.
- > La opción Consultar Historial Médico debe de devolver al menú cuando finalize.

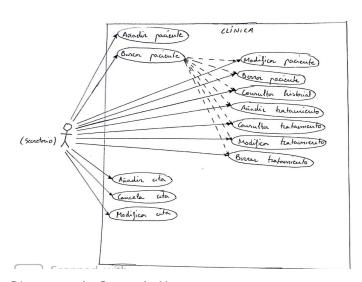


Diagrama de Casos de Uso

2- Actores

- Actores principales: Secretario, el cual tendrá acceso al software.
- Actores secundarios: Pacientes, cuyos tratamiento, citas e historiales serán gestionados.

D-Especificación de los casos de Uso

1- Añadir Cita

- ❖ ID: 05
- Descripción: Se pide por teclado la fecha, hora, nombre y DNI del paciente para la creación de una nueva cita en la agenda
- ❖ Actores principales : Secretario | | Actores secundarios : Paciente
- Precondiciones:
 - Se necesita haber entrado en las opciones de la agenda

- Flujo principal:
- 1. El Secretario desea añadir una cita a la agenda
- 2. El Secretario selecciona "AÑADIR CITA" entre las opciones de la agenda
- 3. El Sistema pide por teclado la fecha, hora y nombre del paciente así como su DNI
 - 4. El Sistema pide confirmación para la creación de dicha cita
- 5. El Sistema muestra un mensaje indicando el éxito o fracaso al añadir la cita en la agenda
 - Postcondiciones
 - Se muestran al Secretario las diferentes opciones de la agenda
 - Flujos alternativos
 - Si la fecha hora o DNI no son correctos el sistema mostrará un mensaje de error

2- Buscar Cita

- ❖ ID:10
- Descripción:Se busca una cita programada en el calendario por fecha o nombre y apellido del paciente
- ❖ Actores principales: Secretario | | Actores secundarios: Paciente

Precondiciones:

• Que exista al menos una cita programada en el calendario.

Flujo principal:

- 1. El Secretario selecciona la opcion de busqueda de cita.
- 2. El Secretario elige la opcion de busqueda por fecha o nombre y apellidos.
- a. El Secretario ha elegido fecha y se le muestra si quiere ver una fecha concreta o todas las que tiene programadas.
- b. El Secretario ha elegido nombre y apellidos y se le muestra la opción de introducir los datos y ver todas las citas para dicho paciente.
- 3. Se le muestra todos los datos de la cita y una opción para salir de dicha opción.

Postcondiciones:

• Ninguna.

Flujos alternativos:

- Se introduce un una fecha no valida, lo que hace que la opción buscar se cierra.
- Se introduce un nombre y apellidos no validos, lo que hace que la opción buscar se cierra.

3- Modificar cita

- **❖** ID: 13
- Descripción: Una vez encontrada la cita a partir de la función buscarCita, el sistema pide por teclado una fecha y hora para actualizar la cita.
- ❖ Actores principales: Secretario | | Actores secundarios: Paciente
- Precondiciones:
 - Se necesita haber encontrado la cita con la función buscarCita
- Flujo principal:
- 1. El secretario desea actualizar la fecha u hora de una cita
- 2. El secretario busca la cita gracias a la función buscarCita por nombre y apellidos o fecha
 - 3. El sistema muestra por pantalla la fecha y hora de la cita
 - 4. El sistema pide una nueva fecha y hora
- 5. El secretario introduce la nueva fecha y hora de la cita y se actualiza
 - Postcondiciones:
 - Se muestra al secretario la cita ya actualizada

- Flujos alternativos:
 - Si el paciente o la cita no existen se muestra un mensaje de error
 - Si la fecha introducida no es válida se muestra un mensaje de error

4- Borrar Cita

- ❖ ID:11
- Descripción: Tras la selección de una cita en el listado de la agenda, se pide confirmación para su consiguiente eliminación
- ❖ Actores principales : Secretario | | Actores secundarios : Paciente
- Precondiciones:
 - Se necesita haber entrado en las opciones de la agenda
- Flujo principal:
- 1. El Secretario desea cancelar la cita de un paciente
- 2. El Secretario selecciona "CANCELAR CITA" entre las opciones de la agenda
 - 3. El Sistema muestra un listado con las citas creadas
 - 4. El Secretario selecciona la cita a cancelar

- 5. El Sistema pide confirmación de la consiguiente cancelación de la cita
 - Postcondiciones
 - Se muestran al Secretario las diferentes opciones de la agenda
 - Flujos alternativos
 - Si la cita ya ha ocurrido en el tiempo, o no es posible su cancelación se muestra un mensaje de error

5- Añadir Paciente

- ❖ ID:01
- Descripción: Se añade un nuevo paciente con todos sus datos (Nombre, Apellido, Dirección, etc).
- ❖ Actores principales:Secretario | | Actores secundarios:Paciente
- Precondiciones:
 - Ninguna
- Flujo principal:
- 1. El Secretario desea añadir un nuevo paciente

- 2. El Secretario selecciona la opción de añadir nuevo paciente y se le abre una ventana
 - 3. El Secretario introduce los datos que pide la ventana
 - 4. Se añade al nuevo paciente a la base de datos del programa
 - Postcondiciones:
 - Ninguna.
 - Flujos alternativos:
 - Si no se rellena todos los datos salta un error y vuelve al paso 3

6- Buscar Paciente

- **❖** ID: 02
- Descripción: Se introduce el DNI del paciente que se busca y el sistema lo muestra.
- ❖ Actores principales: Secretario | | Actores secundarios:Paciente
- Precondiciones:
 - Ninguna

- Flujo principal:
- 1. El secretario desea consultar los datos de un paciente
- 2. El secretario abre el cuadro de diálogo de búsqueda en el menú principal
 - 3. El secretario introduce el DNI del paciente
 - 4. El sistema muestra por pantalla los datos del paciente
 - Postcondiciones:
 - Se muestran al secretario las distintas operaciones relativas al paciente
 - Flujos alternativos:
 - Si el paciente no existe, se muestra un mensaje de error.

7- Modificar Paciente

- ❖ ID: 03
- Descripción: Una vez encontrado el usuario a partir de la función búsqueda, el sistema pide por teclado el dato a modificar, así como su contenido.
- ❖ Actores principales : Secretario | | Actores secundarios : Paciente
- Precondiciones:
 - Se necesita haber llamado a la función búsqueda

- Flujo principal:
- 1. El Secretario desea modificar los datos de un paciente
- 2. El Secretario realiza la búsqueda del paciente
- 3. El Secretario abre la opción de modificación de los datos del paciente
 - 4. El Sistema hace elegir al Secretario el/los datos a modificar
 - 5. El Secretario introduce el/los datos a modificar
 - 6. El Sistema muestra por pantalla los datos modificados
 - Postcondiciones
 - Se muestran al Secretario las diferentes opciones relativas al paciente
 - Flujos alternativos
 - Si el/los datos no es/son válido/s el sistema muestra un mensaje de error;

8- Borrar Paciente

- ❖ ID: 04
- Descripción: Una vez encontrado el paciente a partir de la función búsqueda, el sistema pide confirmación para el borrado de los datos del mismo.
- ❖ Actores principales : Secretario | | Actores secundarios : Paciente

Precondiciones:

• Se necesita haber llamado a la función búsqueda

Flujo principal:

- 1. El Secretario desea borrar los datos de un paciente
- 2. El Secretario realiza la búsqueda del paciente
- 3. El Secretario abre la opción de borrado del paciente
- 4. El Sistema pide confirmación del inminente borrado
- 5. El Secretario confirma el borrado
- 6. El Sistema muestra por pantalla un mensaje donde se confirma que el borrado se ha realizado correctamente

Postcondiciones

• Se muestran al Secretario las diferentes opciones relativas al paciente

Flujos alternativos

• Si el borrado no se produce correctamente el Sistema manda un mensaje de error

9- Añadir Tratamiento

- **❖** ID: 06
- Descripción: Tras buscar un paciente podemos añadirle un tratamiento
- ❖ Actores principales: Secretario | | Actores secundarios: Paciente
- Precondiciones:
 - Buscar un paciente con la función buscarPaciente
- Flujo principal:
- 1. El secretario desea añadir un tratamiento a un paciente
- 2. El secretario busca a un paciente gracias a buscarPaciente
- 3. El secretario selecciona añadir tratamiento en el menú
- 4. El secretario introduce el tratamiento y se añade
- * Postcondiciones:
 - Se muestran al secretario los tratamientos del paciente
- Flujos alternativos:
 - Si el paciente no existe, se muestra un mensaje de error

10- Consultar Tratamiento

- ❖ ID: 09
- Descripción: Tras la búsqueda por DNI de un paciente se muestra su tratamiento
- ❖ Actores principales : Secretario | | Actores secundarios : Paciente
- Precondiciones:
 - Se necesita haber entrado en las opciones del Paciente, así como la existencia del paciente
- Flujo principal:
- 1. El Secretario debe haber realizado la búsqueda del paciente por DNI
- 2. El Secretario selecciona "CONSULTAR TRATAMIENTO" entre las opciones del Paciente
- 3. El Sistema muestra un listado con el tratamiento médico del paciente
 - Postcondiciones
 - Se necesita haber buscado al paciente y haber mostrado al Secretario las diferentes opciones del Paciente

- Flujos alternativos
 - Si el DNI es erróneo (número de dígitos incorrecto u otro fallo del tipo) mostrará un mensaje de error, así mismo si no existe tratamiento del paciente el sistema mostrará un mensaje diciendo que el paciente no ha sido añadido o no tiene tratamiento médico todavía

11- Borrar Tratamiento

- **❖** ID: 12
- Descripción: Elimina el tratamiento de un paciente buscado anteriormente.
- ❖ Actores principales: Secretario | | Actores secundarios: Paciente
- Precondiciones:
 - Debe existir un tratamiento para poder eliminarlo y buscar a un paciente con la función buscarPaciente
- Flujo principal:
- 1. El secretario desea eliminar el tratamiento de un paciente
- 2. El secretario busca al paciente que quiere eliminar el tratamiento con buscarPaciente
 - 3. El secretario selecciona en el menú eliminar tratamiento
 - 4. El sistema muestra todos los tratamientos del paciente
 - 5. El secretario introduce el tratamiento que quiere eliminar

- Postcondiciones:
 - Se muestra al secretario los tratamientos ya actualizados
- Flujos alternativos:
 - Si el paciente no existe se muestra un mensaje de error.
 - Si el paciente no tiene tratamientos para eliminar se muestra un mensaje de error
 - Si se introduce un tratamiento que no existe se muestra un mensaje de error

12- Modificar Tratamiento

- ❖ ID: 14
- Descripción: Modifica el tratamiento de un paciente buscado anteriormente.
- ❖ Actores principales: Secretario | | Actores secundarios:Paciente
- Precondiciones:
 - Debe existir un tratamiento para poder modificarlo y buscar a un paciente con la función buscarPaciente

- Flujo principal:
- 1. El secretario desea modificar el tratamiento de un paciente
- 2. El secretario busca al paciente que quiere modificarle el tratamiento con buscarPaciente
 - 3. El secretario selecciona en el menú modificar tratamiento
 - 4. El sistema muestra todos los tratamientos del paciente
 - 5. El secretario introduce el tratamiento que quiere modificar
 - 6. El secretario introduce el tratamiento ya modificado
 - Postcondiciones:
 - Se muestra al secretario los tratamientos ya actualizados
 - Flujos alternativos:
 - Si el paciente no existe se muestra un mensaje de error.
 - Si el paciente no tiene tratamientos para modificar se muestra un mensaje de error
 - Si el tratamiento no existe se muestra un mensaje de error

13- Consultar Historial Médico

- **❖** ID: 08
- Descripción: Tras la búsqueda por DNI de un paciente se muestra su historial médico

- ❖ Actores principales : Secretario | | Actores secundarios : Paciente
- Precondiciones:
 - Se necesita haber entrado en las opciones del Paciente
- Flujo principal:
- 1. El Secretario debe haber realizado la búsqueda del paciente por DNI
- 2. El Secretario selecciona "CONSULTAR HISTORIAL" entre las opciones del Paciente
 - 3. El Sistema muestra un listado con el historial médico del paciente
 - Postcondiciones
 - Se necesita haber buscado al paciente y haber mostrado al Secretario las diferentes opciones del Paciente
 - Flujos alternativos
 - Si el DNI es erróneo (número de dígitos incorrecto u otro fallo del tipo) mostrará un mensaje de error, así mismo si no existe historial del paciente el sistema mostrará un mensaje diciendo que el paciente no ha sido añadido o no tiene historial médico todavía

14- Mostrar Agenda

- **❖** ID: 07
- Descripción: Se introduce una fecha y aparecen todas las citas planeadas.
- ❖ Actores principales: Secretario | | Actores secundarios: Paciente
- Precondiciones:
 - Ninguna
- Flujo principal:
- 1. El secretario desea consultar las próximas citas
- 2. El secretario introduce la fecha en el menú principal
- 3. El sistema muestra por pantalla la hora de las próximas citas de esa fecha
 - * Postcondiciones:
 - Se muestran al secretario las distintas operaciones relativas a la modificación de una cita
 - Flujos alternativos:
 - Si se introduce una fecha ya pasada se mostrarán las citas que hubo ese día

E- Validación de los casos de uso con los requisitos funcionales

RF: Requisitos funcionales (Fila superior).

CU: Casos de uso (Columna izquierda).

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

F- Representación, descripción y especificación de las entidades de información que manejará el sistema

• Clase: Cita

Esta clase contendrá las funciones que engloban al tratado de citas. Entre ellas se encuentran funciones capaces de manejar la agenda con sus respectivas citas.

Datos

- Fecha || Cadena de caracteres || Nombre del tratamiento
- DNI_paciente || Cadena de caracteres || Dni del paciente de la cita
- Nombre_Paciente || Cadena de caracteres || Nombre y apellidos del paciente en cuestión
- Motivo || Cadena de caracteres || Especificación de la causa por la que se pide la cita

Métodos

- Añadir_Cita || Función que añade una cita en la agenda
- Buscar_Cita || Función encargada de buscar una cita en la agenda
- Cancelar_Cita || Función que elimina una cita de la agenda
- Modificar_Cita || Función que modifica una cita ya existente en la agenda
- Consultar_Historial || Función que muestra por pantalla el historial de un paciente
 - Mostrar_Agenda || Función que muestra por pantalla la agenda
 - getFecha || Función que retorna la fecha de la cita correspondiente
 - setFecha | | Función que guarda la fecha de la cita

- getDniPaciente | | Función de retorna el dni del paciente
- setDni_paciente || Función que guarda el dni del paciente
- getNombre_paciente | | Función de retorna el nombre del paciente
- setNombre_paciente | | Función que guarda el nombre del paciente
- getMotivo || Función que retorna el motivo de la cita
- setMotivo || Función que guarda el motivo de la cita

• Clase: Paciente

Esta clase contendrá las funciones que engloban al tratado de pacientes. Entre ellas se encuentran funciones que permiten manipular los elementos de los pacientes.

Datos

- DNI_paciente || Cadena de caracteres || Dni del paciente de la cita
- Nombre || Cadena de caracteres || Nombre del paciente
- Apellido || Cadena de caracteres || Apellidos del paciente
- Edad || Número entero || Edad del paciente
- Dirección || Cadena de caracteres || Dirección del paciente

Métodos

- añadir_Paciente() || Función que añade a un paciente
- buscar_paciente() || Función que busca a un paciente
- modificar_Paciente() || Función encargada de modificar el perfil de un paciente ya añadido
- borrar_Paciente() || Función encargada de eliminar el perfil de un paciente ya añadido
 - getDNI() | | Función de retorna el dni del paciente

- setDNI() || Función que guarda el dni del paciente
- getNombre() | | Función de retorna el nombre del paciente
- setNombre() || Función que guarda el nombre del paciente
- getApellidos() || Función que retorna los apellidos del paciente
- setApellido || Función que guarda los apellidos del paciente
- getEdad() || Función que retorna la edad del paciente
- setEdad() || Función que guarda la edad del paciente
- getDireccion() | | Función que retorna la dirección del paciente
- setDireccion() | | Función que guarda la dirección del paciente

• Clase: Tratamiento

Una lista que guarda los tratamientos y la duración de estos los cuales le corresponde a cada paciente.

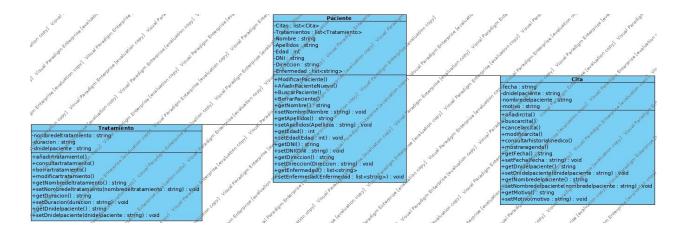
Datos

- nombredeltratamiento || Cadena de caracteres || Nombre del tratamiento
 - duración || Cadena de caracteres || Duración del tratamiento
- dnidelpaciente || Cadena de caracteres || Dni del paciente al que le corresponde el tratamiento

Método

• añadirtratamiento() || Función que añade una línea a la lista de un paciente o crea una lista de 0 si es el primer tratamiento de dicho paciente.

- consultartratemineto | | Imprime la lista de una paciente
- borrartratamiento() || Borra una línea de la lista de un paciente(por nombre del tratamiento) o borra la lista entera si es el único tratamiento que hay.
- modificartratamiento() || Modifica una línea de la lista de un paciente(por nombre del tratamiento)
- getNombredeltratamiento() : string || Recoge el nombre de un paciente en la variable nombredeltratamiento
- setNombredeltratamiento(nombredeltratamiento : string) : void | | Coge la variable nombredeltratamiento
- getDuracion() : string || Recoge la duración de un tratamiento en la variable duración
 - setDuracion(duracion : string) : void || Coge la variable duración
- getDnidelpaciente() : string || Recoge el DNI de un paciente en la variable dnidelpaciente
- setDnidelpaciente(dnidelpaciente : string) : void || Coge la variable dnidelpaciente



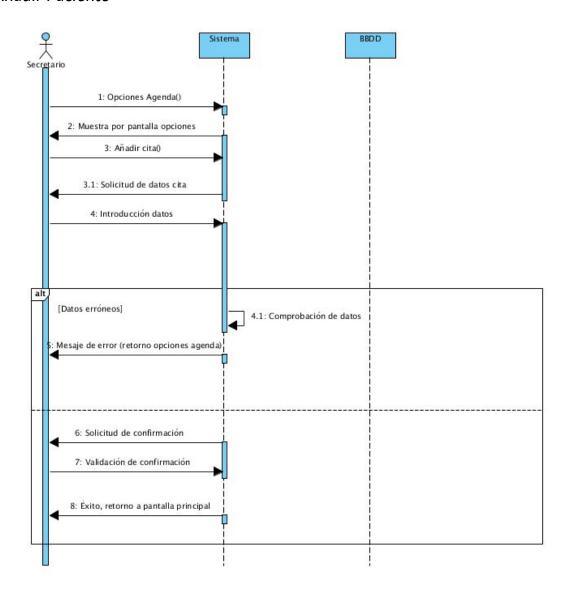
G- Validación de las clases con los casos de uso

Clases: representadas por su nombre (Fila superior). Casos de uso (CU): representados por su id(columna izquierda)

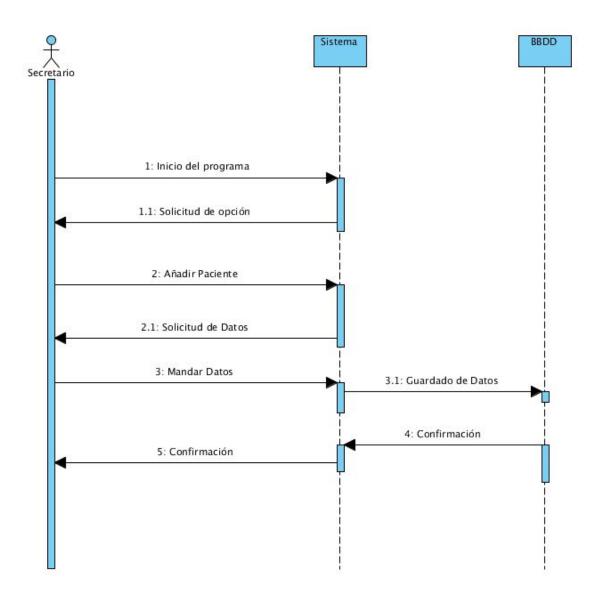
CU\CLASE	Paciente	Tratamiento	Cita
C01	X		
C02	X		
C03	X		
C04	X		
C05			X
C06		X	
C07			X
C08	X		
C09		X	
C10			X
C11			X
C12		X	
C13			X
C14		X	

H- Diagramas de Secuencias

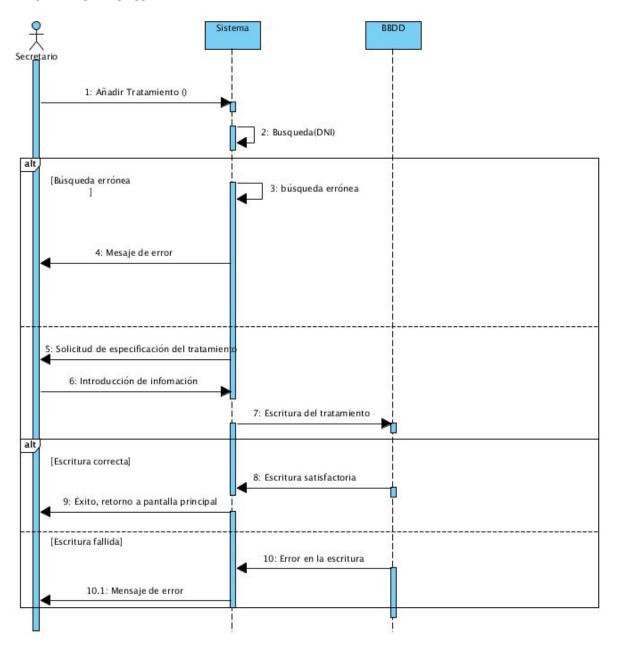
1- Añadir Paciente



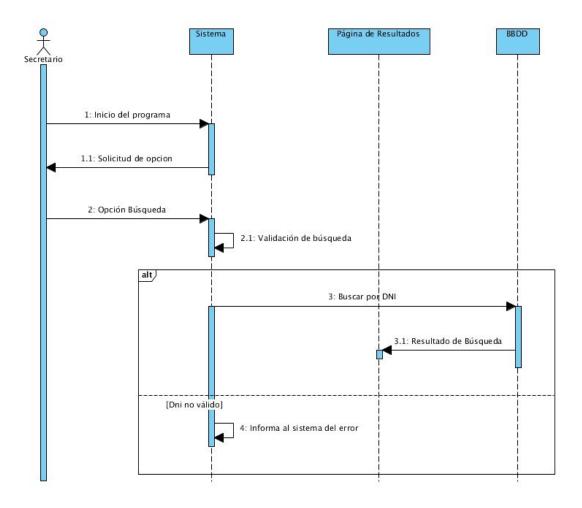
2- Añadir Cita



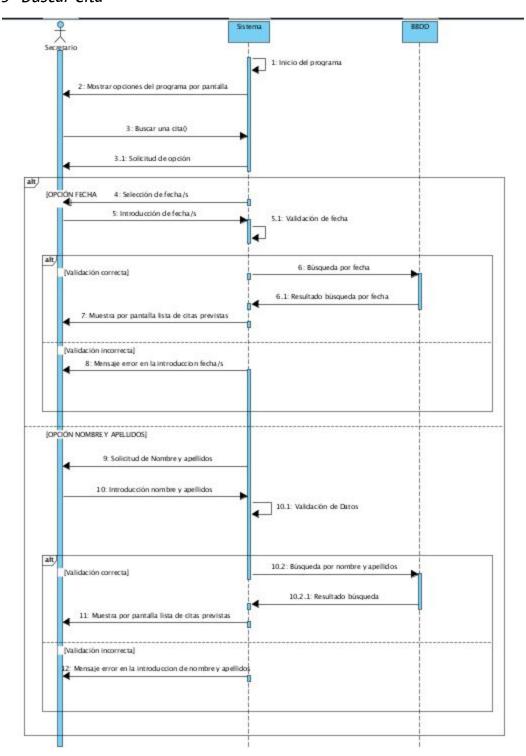
3- Añadir Tratamiento



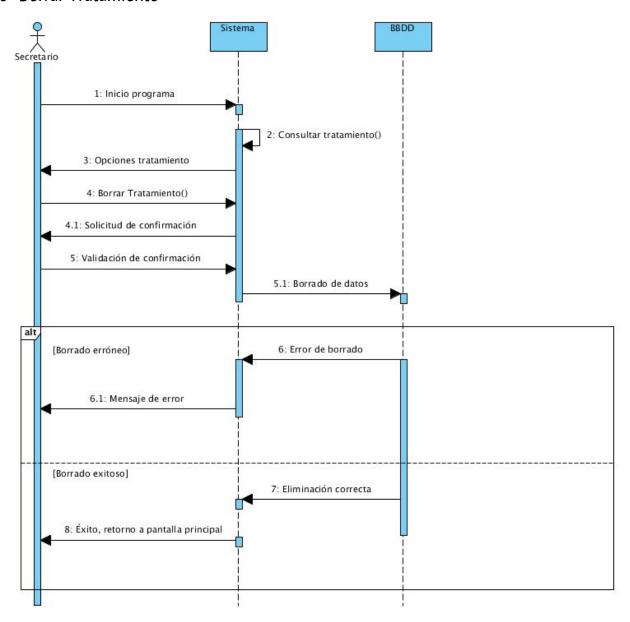
4- Buscar Paciente



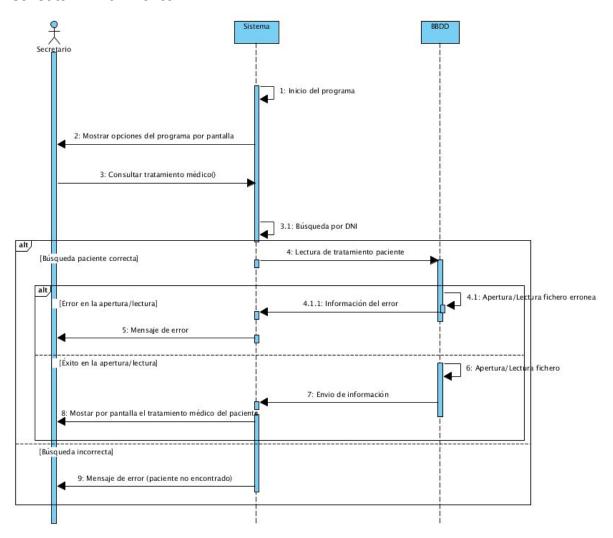
5- Buscar Cita



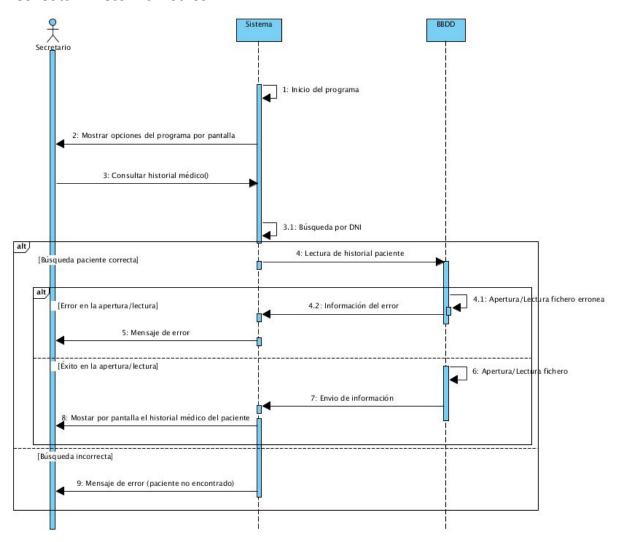
6- Borrar Tratamiento



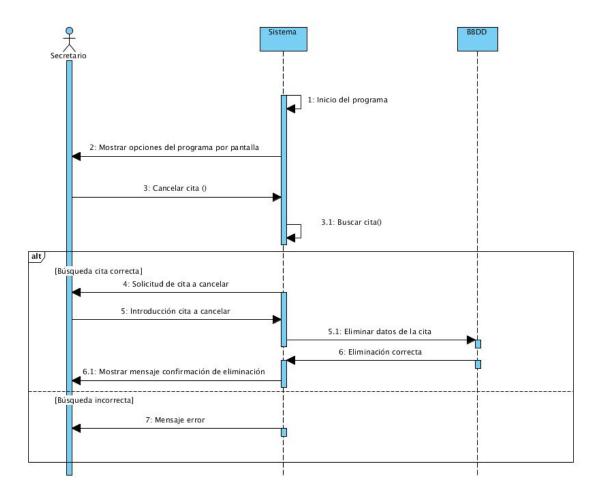
7- Consultar Tratamiento



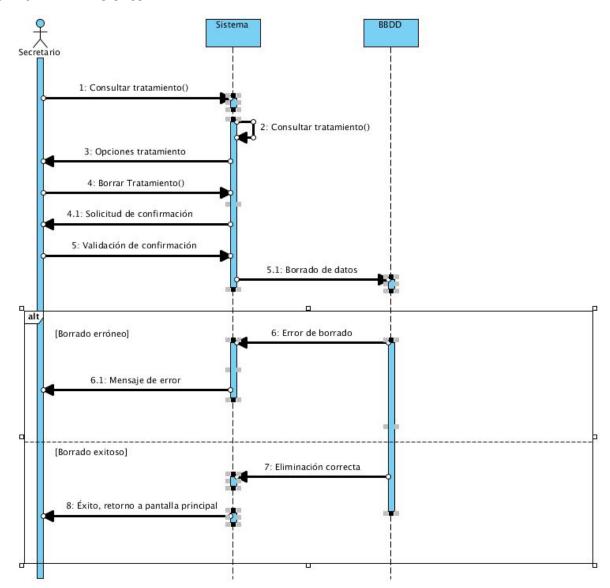
8- Consultar Historial Médico



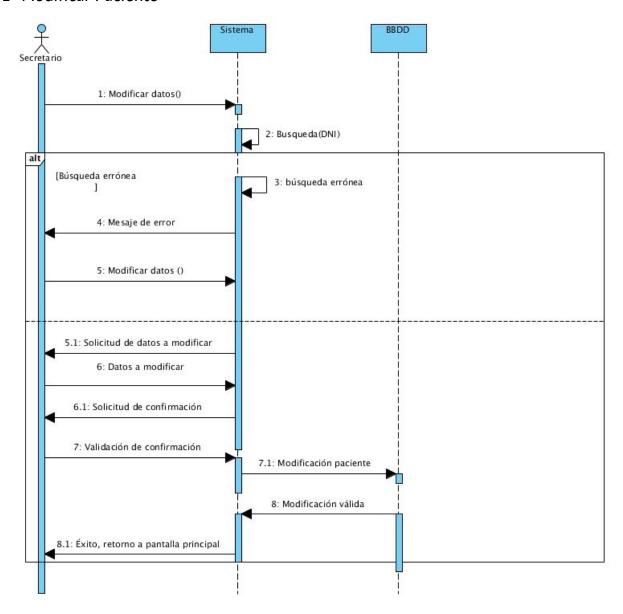
9- Cancelar Cita



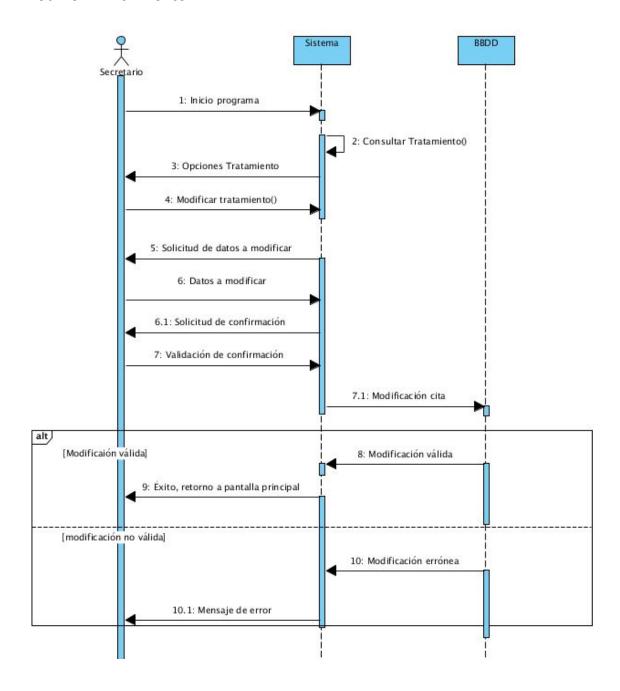
10- Eliminar Paciente



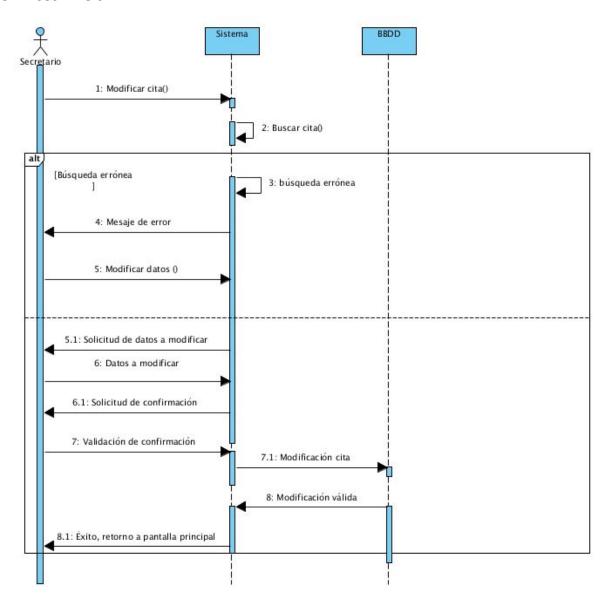
11- Modificar Paciente



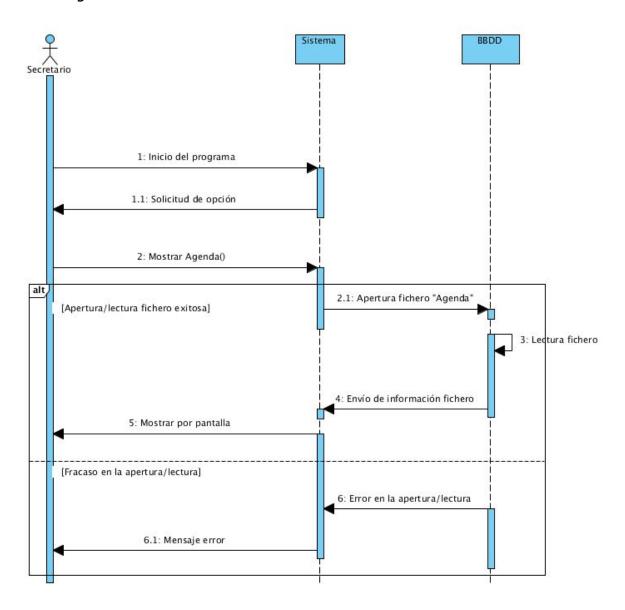
12- Modificar Tratamiento



13- Mostrar Cita



14- Mostrar Agenda



I- Implementación, refinamiento y pruebas usando la metodología SCRUM con las iteraciones que sean necesarias.

Aquí se proporcionan los enlaces a nuestro proyecto de GitHub y a nuestro SCRUM en taiga.

El código se descargará desde el enlace de GitHub y se usará según el Readme que se encuentra en GitHub, ya que se indica cómo empezar a usar el software diseñado. Las pruebas realizadas al código se han realizado tanto individualmente (función por función), como en conjunto (todo el software en conjunto), para así garantizar y asegurar el debido funcionamiento del software exigido por la empresa.

Enlace a SCRUM:

https://tree.taiga.io/project/enderfinow-sclinico/timeline

Enlace para la descarga del software:

https://github.com/DupliCraft15/Sofware_CLINICO