**Proyecto: RESGES**

**Especificación de Requisitos Software**

**(Formato IEEE Std. 830-1998)**

****

**Miembros del equipo:**

| Eduardo Regidor Serrano | Jose Antonio Almendros Corrales |
| --- | --- |
| Sergio Cuenca López | Alfonso Barriga Lucena |
| Andrés Enrique García de Arboleya Ortíz | Pablo Sánchez Redondo |
| Joel Valencia Lorente |  |

**Control de cambios**

| **Número de versión** | **Fecha** | **Autores** | **Descripción** |
| --- | --- | --- | --- |
| 1 | 4/12/2023 | Eduardo Regidor | Añadir su módulo en el 3.2 |
| 2 | 8/12/2023 | Andrés Enrique García | Añadir puntos 2.5 y 2.6 |
| 3 | 14/12/2023 | Andres Enrique García, Eduardo Regidor, Joel Valencia | Añadir 3.3, 3.6, 3.1, 2.6(modificarlo), 2.3, 2.1, 1.1, 1.2, 1.5 |
| 4 | 16/12/2023 | Andres Enrique García, Eduardo Regidor | Añadir 3.5 y editar 3.2.1 y 3.2.2 |
| 5 | 17/12/2023 | Andres Enrique García, Eduardo Regidor | Añadir 3.4, 3.7 y editar 3.2.2 |
| 6 | 18/12/2023 | Andres Enrique García, Sergio Cuenca | Añadir 3.2.3 y modificar algunos puntos |
| 7 | 19/12/2023 | Pablo Sánchez Redondo, Andres Enrique García, Eduardo Regidor | Añadir 3.2.4, y acabar los punto 2 y 1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Índice**

1 Introducción 4

1.1 Propósito 4

1.2 Alcance 4

1.3 Definiciones, acrónimos y abreviaturas 4

1.4 Referencias 4

1.5 Organización del documento 4

2 Descripción general 5

2.1 Perspectiva del producto 5

2.2 Funciones del producto 5

2.3 Características del usuario 5

2.4 Restricciones 5

2.5 Supuestos y dependencias 5

2.6 Requisitos futuros 5

3 Requisitos específicos 6

3.1 Interfaces externos 6

3.1.1 Interfaces hardware 6

3.1.2 Interfaces software 6

3.1.3 Interfaces de Comunicación 6

3.2 Requisitos funcionales 7

3.2.1 Subsistema 1 VER LAS SUBSECCIONES 7

3.2.2 Subsistema 2 7

3.2.3 Subsistema 3 . . . . . . . . . 7

3.3 Requisitos de rendimiento 8

3.4 Requisitos lógicos de la base de datos 8

3.5 Restricciones de diseño 8

3.6 Atributos del sistema software 8

3.7 Otros requisitos 8

# Introducción

## Propósito

Creación de un software de gestión que facilite a los clientes (trabajadores de residencias de ancianos) encargar, revisar y modificar la información de los residentes, desde las horas de atención hasta los cuidados específicos.

## Alcance

El nombre del proyecto será ResGes con el que pretendemos eliminar una capa de complejidad a la transmisión y revisión de información entre trabajadores y gestión al ofrecer una aplicación que genera una intuitiva red en la que los clientes pueden ver los residentes a los que atender así como sus necesidades (dietas, visitas, medicamentos…) y en caso de necesitarlo añadir observaciones que serán directamente enviadas al gestor pertinente, el cual cuenta con el acceso a la información de todos los residentes así como puede ver los horarios de los clientes permitiendo con unos pocos clics modificar el residente al que deben atender. Todo esto con el objetivo de evitar fallos de comunicación y/o errores en el beneficio de la productividad de la residencia, usuarios y los propios residentes facilitando el trabajo de los clientes.

## Definiciones, acrónimos y abreviaturas

ResGes: nombre de la aplicación.

SGBD: sistema de gestión de bases de datos.

Clientes: Los trabajadores de la residencia a la que va dirigido ResGes.

Residentes: Los ancianos de la residencia.

IEEE - Institute of Electrical and Electronics Engineers

## Referencias

Las referencias que hemos cogido para el desarrollo de este documento ha sido seguir el estándar de modelo de “SRS IEEE 830” y los apuntes de la asignatura de IS1

## 1.5 Organización del documento

1. INTRODUCCIÓN

Apartado introductorio que ofrece un rápido resumen del proyecto

1. DESCRIPCIÓN GENERAL

Descripción general de los agentes que afectan a la aplicación y sus requisitos

1. REQUISITOS ESPECÍFICOS

Especificación detallada de los requisitos destinados a diseñadores y auditores

# 2 Descripción general

Vamos a introducir las ideas principales sobre ResGes, servirá para hacerse una idea de los requisitos y funciones.

## Perspectiva del producto

ResGes se implementa gracias a la integración de un SGBD ofrecido por una empresa externa que ofrece la realización de un backup cada 3 dias, la capacidad completa dependerá del tamaño de los datos manejados por la residencia junto con Google calendar que ofrece a los usuarios la posibilidad de coordinar las tareas y horarios establecidos y modificados en el software con su calendario personal de Google.

## Funciones del producto

Podemos encontrar las funciones del producto divididas en varios módulos en los que están:

Módulo de Gestión de actividades: Se encargará de añadir actividades , eliminarlas y modificarlas además de si hay la necesidad de hacer una consulta sobre alguna en particular.

Módulo de Gestión de medicamentos: Se encargará de mostrar todos los medicamentos que están en la base de datos y además añadir o eliminar medicamentos , en la misma base de datos suma o resta existencias de un medicamento en particular.

Módulo de Gestión de los profesionales del cuidado: Se encargará de toda la gestión del cuidado hacia los residentes, podremos consultar,modificar,asignar y eliminar desde este módulo las horas de atención y hacer una visualización de las necesidades de cada residente.

Módulo de Gestión de residentes: En este módulo el cliente encontrará las herramientas para dar de alta , de baja modificar datos o mostrar los datos de cualquier residente de la residencia.

Módulo de Gestión de estancia: En este módulo el cliente encontrará las herramientas para dar de alta o de baja de estancia , también poder cambiar la duración de esa misma o un un simple cambio de estancia.

Módulo de Gestión de las patologías de los residentes: En este módulo almacenaremos todos los datos de los residentes y habrá las herramientas para poder comparar patologías en caso de necesidad de ver si hay alguna enfermedad no actualizada en los datos.

Módulo de Gestión de pagos:Con el fin de tener una gestión financiera adecuada este módulo se encarga del tema financiero donde tendrá funciones basadas en la deuda como :añadir deuda,eliminar deuda, aumentar deuda o disminuir deuda.

Módulo de Gestión de patologías:Este módulo es el encargado de modificar la base de datos donde están almacenadas las enfermedades con sus respectivas patologías.

## Características del usuario

Nuestro software cuenta con tres interfaces según el trabajador al que va dirigido: la interfaz de trabajador ofrece un acceso user friendly al asemejarse a aplicaciones que empleamos ya en nuestro día a día permitiendo su uso a cualquier persona. La del gestor mantiene su interfaz intuitiva pero se espera que la persona que lo maneje tenga conocimientos previos en aplicaciones de ofimática.Y finalmente contamos con una interfaz no obligatoria que ofrece a un personal especializado en software muchas más herramientas para la maleabilidad de la propia aplicación.

## Restricciones

* Para usar nuestro software, solo se necesitaría acceso a las bases de datos, por lo que no se necesitaría acceso a ningún otro tipo de aplicación externa.
* El lenguaje, a la hora de desarrollarlo sería java, debido a que es con el que estamos más familiarizados.
* Para tener mayor fiabilidad, los únicos que deben usar el software son los trabajadores, para evitar que alguien altere la información y proporcione alguna incorrecta.
* La intención con éste software es que los trabajadores dependan mayoritariamente de él, para tener una mejor organización, por lo tanto será importante que esté siempre en orden.
* En cuanto a la seguridad, como el software solo estará a disposición de los trabajadores, no es necesario ningún tipo de log in.

## Supuestos y dependencias

Para el correcto funcionamiento del sistema está pensado su uso en versiones de Windows superiores a XP con acceso a Internet por lo tanto hay una dependencia ya que si solo se puede disponer de un sistema operativo distinto a los mencionados anteriormente , habría que cambiar gran parte de la aplicación para poder adaptarse.

## Requisitos futuros

Las posibles mejoras para las siguientes versiones de ResGes:

Lanzar una aplicación web donde los clientes puedan entrar a ResGes desde dispositivos como móviles o tablets.

Introducir la opción de cambiar de lenguaje de la página al gusto del cliente entre: Inglés,Español,Catalán,Euskera.

Introducir una nueva funcionalidad en la que si ya tienes algún familiar en ResGes darte la posibilidad de un descuento para introducir en la residencia a más.

# Requisitos específicos

En esta sección profundizaremos sobre “ResGes” nuestra aplicación con un nivel de detalle elevado, explicando funciones, interfaz y sistema.

## Interfaces externos

*-Interfaces de usuario:*

De fácil interpretación, va dirigido al registro de usuario y a la visualización de los servicios de ResGes encontramos:

-Una interfaz de registro común en la que poner tu usuario y contraseña

-Interfaces gráficas en las que les permita ver toda la organización del centro y necesidades de cada residente dependiendo del cargo del trabajador del centro.

*-Interfaces hardware:*

Son aquellos que permiten la comunicación de elementos internos y externos por lo tanto necesitaremos:

Un buen equipamiento informático dependiendo de las necesidades necesitaremos los elementos básicos hardware de un ordenador o dispositivo móvil y una conexión capaz de poder transferir datos constantes como es el caso de módems y una red de área extensa (WAN).

*-Interfaces software:*

Al ser creado “RESGES” a partir de Java para su correcto funcionamiento necesitamos que en el dispositivo en el que se usa disponga del software Java Virtual Machine

*-Interfaces de Comunicación:*

Habrá una interfaz de comunicación con el objetivo de almacenar la información de cada centro de residencia.

## Requisitos funcionales

### Gestión de estancias

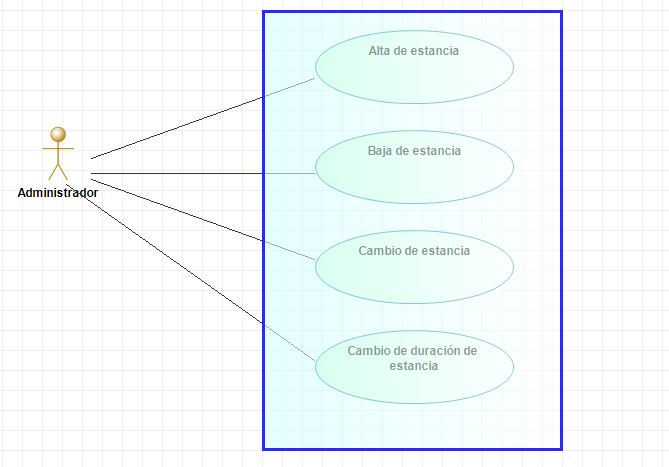
- DCU subsistema 1.1

- Diagrama Actividad 1.1 (del CU 1)

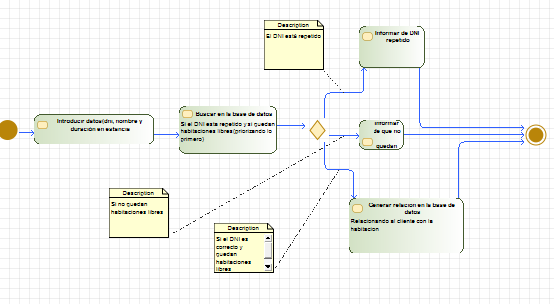
- Descripción Caso de Uso 1.1 (ver plantilla aparte)

- Descripción Resto de Casos de Uso

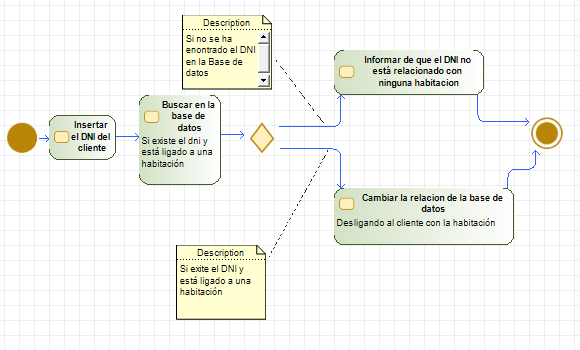
Subsistema dedicado a la gestión de las estancias



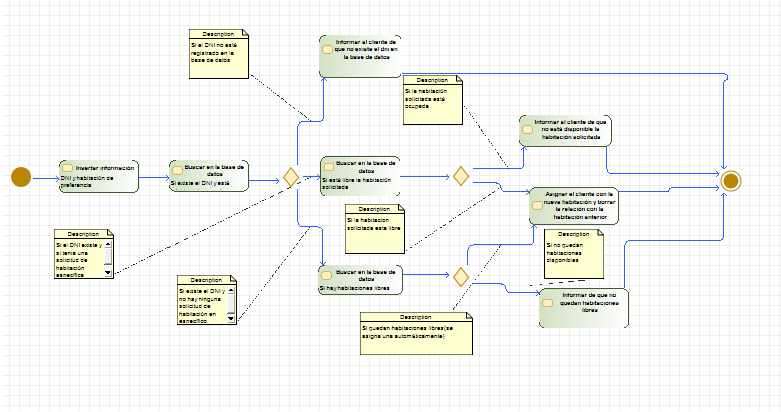
| **CU-3.2.1.1** | **Alta de estancia** | |
| --- | --- | --- |
| **Subsistema** | Gestión de estancias | |
| **Descripción** | Da de alta una habitación, es decir, le asigna la habitación al cliente | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Eduardo Regidor | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | que no haya DNI repetidos (que no esté en otra habitación). | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El administrador accede a la interfaz de dar de alta |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a o 2.b |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se inserta una relación entre habitación y cliente en la base de datos  Si fallo: No se produce ninguna relación y se vuelve a la interfaz de dar de alta, habrá que informar al cliente del mensaje que aparece | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo producido es debido a la relación del dni del usuario con otra habitación, nos devolvería al paso 1, y mostraría en el interfaz un mensaje: ”Informar al usuario de que ya dispone de otra habitación” |
| **2.b** | El fallo es debido a la ocupación de todas las habitaciones, nos devolvería al paso 1 mostrando el mensaje: “No hay habitaciones disponibles” |
| **Requisitos no funcionales** | La disponibilidad a ésta función sólo será en el horario de la residencia(en el que tiene sus puertas abiertas, y se pueda atender al cliente) | |
| **Comentarios** |  | |



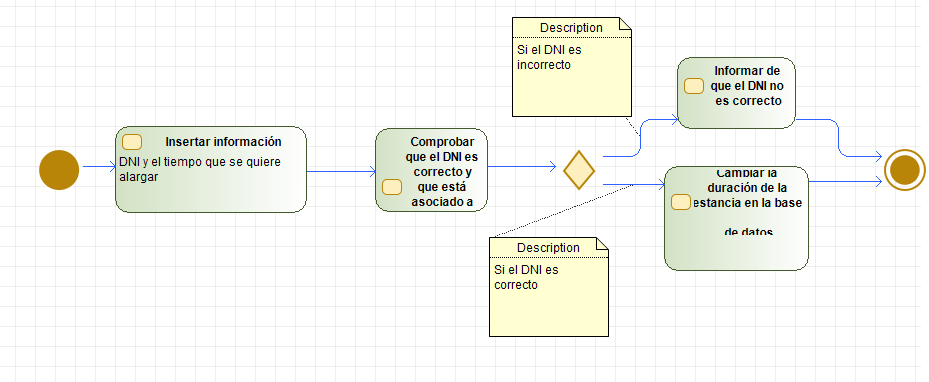
| **CU-3.2.1.2** | **Baja de estancia** | |
| --- | --- | --- |
| **Subsistema** | Gestión de estancias | |
| **Descripción** | Da de baja una habitación, es decir, desvincula la habitación y el cliente | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Eduardo Regidor | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | Que exista el DNI del cliente y que esté relacionado con una habitación | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El administrador accede a la interfaz de dar de baja |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se borra la relación entre habitación y cliente en la base de datos  Si fallo: Sigue la misma relación presente en la base de datos, se vuelve a la interfaz de dar de baja y muestra el mensaje enviado. Para poder hacerlo habría que repetir el proceso | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo producido es debido a la relación inexistente del DNI con alguna habitación, mostraría en el interfaz el mensaje: “No existe relación entre cliente y habitación” |
| **Requisitos no funcionales** | La disponibilidad a ésta función sólo será en el horario de la residencia(en el que tiene sus puertas abiertas, y se pueda atender al cliente) | |
| **Comentarios** |  | |



| **CU-3.2.1.3** | **Cambio de estancia** | |
| --- | --- | --- |
| **Subsistema** | Gestión de estancias | |
| **Descripción** | Cambia la relación con un cliente y una estancia, relacionando a éste con otra | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Eduardo Regidor | |
| **Prioridad** | Baja | |
| **Estabilidad** | Baja | |
| **Precondiciones** | Que exista el DNI del cliente y que esté relacionado con una habitación | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El administrador accede a la interfaz de dar de cambio de estancia |
| **2** | El administrador introduce los datos en la interfaz. Viendo si con la entrada viene una habitación exacta (que reflejaría el gusto del cliente de una estancia en específico, sino se le asignará una de forma automática). Si fallo 2.a, 2.b y 2.c |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se produce el cambio de relaciones correctamente  Si fallo: Sigue la misma relación presente en la base de datos, se vuelve a la interfaz de cambio de estancia y muestra el mensaje enviado. Para poder hacerlo habría que repetir el proceso | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo producido es debido a la relación inexistente del DNI con alguna habitación, mostraría en el interfaz el mensaje: “No existe relación entre cliente y habitación” |
| **2.b** | El fallo es producido debido a la ocupación de todas las habitaciones, por lo que volverá a la interfaz y devolverá el mensaje: “No quedan habitaciones disponibles” |
| **2.c** | El fallo es producido debido a la ocupación de la habitación que quería el cliente, devolverá a la interfaz de cambio de estancias y devolverá el mensaje: “Habitación ocupada, seleccione otra habitación” |
| **Requisitos no funcionales** | La disponibilidad a ésta función sólo será en el horario de la residencia(en el que tiene sus puertas abiertas, y se pueda atender al cliente) | |
| **Comentarios** |  | |



| **CU-3.2.1.4** | **Cambio de duración estancia** | |
| --- | --- | --- |
| **Subsistema** | Gestión de estancias | |
| **Descripción** | Cambia la duración de la relación entre el usuario y la estancia | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Eduardo Regidor | |
| **Prioridad** | Baja | |
| **Estabilidad** | Baja | |
| **Precondiciones** | Que exista el DNI del cliente y que esté relacionado con una habitación | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El administrador accede a la interfaz de dar de cambio de duración de estancia |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se produce el cambio correctamente  Si fallo: Sigue la misma relación presente en la base de datos, sin cambios en la duración, se vuelve a la interfaz de cambio de estancia y muestra el mensaje enviado. Para poder hacerlo habría que repetir el proceso | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo producido es debido a la relación inexistente del DNI con alguna habitación, mostraría en el interfaz el mensaje: “No existe relación entre cliente y habitación” |
| **Requisitos no funcionales** |  | |
| **Comentarios** |  | |



### Gestión de Patologías de residentes

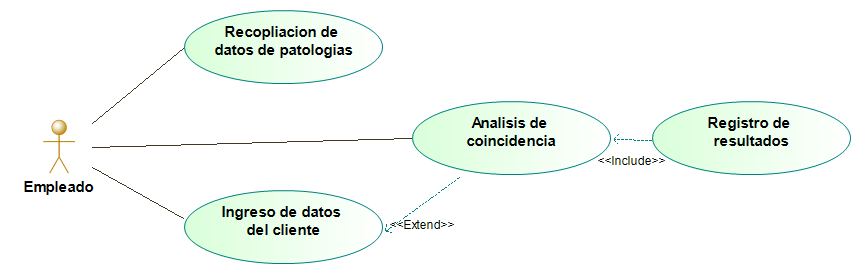
- DCU subsistema 1

- Diagrama Actividad 1 (del CU 1)

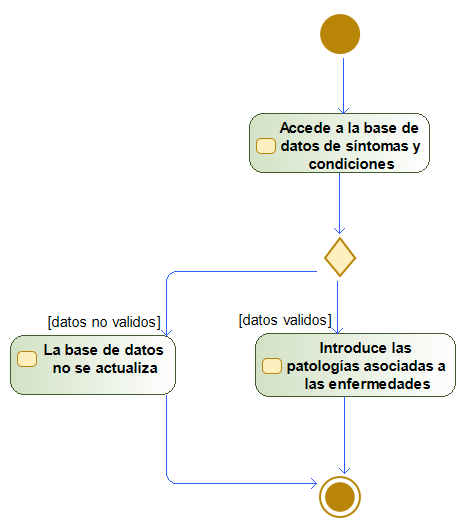
- Descripción Caso de Uso 1 (ver plantilla aparte)

- Descripción Resto de Casos de Uso

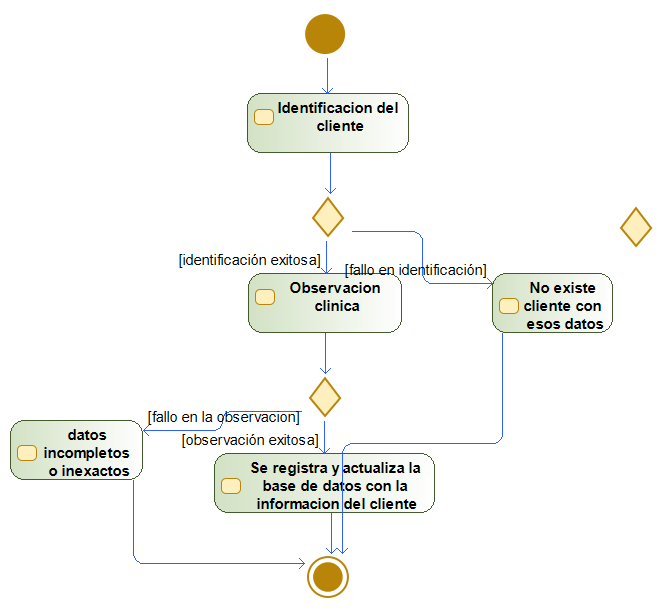
Subsistema dedicado a Gestión de Patologías de residentes.



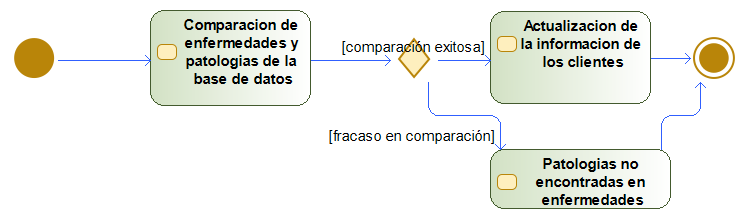
| **CU-3.2.2.1** | **Recopilación de datos de patología** | |
| --- | --- | --- |
| **Subsistema** | Gestión de patologías de clientes | |
| **Descripción** | Se recopila una lista completa de las patologías asociadas a todas las enfermedades creando una base de datos de síntomas y condiciones. | |
| **Actores** | Actor principal que invoca el Caso de Uso: *Empleado* | |
| **Descrito por** | Andrés García de Arboleya Ortíz | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | Disponibilidad de acceso a datos clínicos confiables como bases de datos médicos y guías clínicas. | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El empleado accede a la base de datos de síntomas y condiciones |
| **2** | Introduce las patologías asociadas a las enfermedades |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se actualiza la base de datos  Si fallo: La base de datos no se actualizará debido al 2.a y saldrá error | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo se produce por la introducción en la base de datos de enfermedades y sus patologías ya existentes dentro de la base. |
| **Requisitos no funcionales** | Fiabilidad de la información registrada en la base de datos | |
| **Comentarios** |  | |



| **CU-3.2.2.2** | **Ingreso de datos del cliente** | |
| --- | --- | --- |
| **Subsistema** | Gestión de patologías de clientes | |
| **Descripción** | El empleado registra las patologías que se observa del cliente | |
| **Actores** | Actor principal que invoca el Caso de Uso: *Empleado* | |
| **Descrito por** | Andrés García de Arboleya Ortíz | |
| **Prioridad** | Alta | |
| **Estabilidad** | Media | |
| **Precondiciones** | El empleado deberá estar debidamente autorizado y preparado para poder desarrollar su función | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | Se hará una identificación del cliente para su correcto seguimiento y una observación clínica prestando atención a los signos físicos o síntomas que puedan dictaminar su mejoría en la salud o si ha surgido una nueva patología |
| **2** | Se registra y actualiza la base de datos con la nueva información obtenida del cliente |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se actualiza la base de datos  Si fallo: La base de datos no se actualizará debido al 2.a y saldrá error | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo se produce por datos incompletos o inexactos |
| **Requisitos no funcionales** | Seguridad y confidencialidad | |
| **Comentarios** |  | |



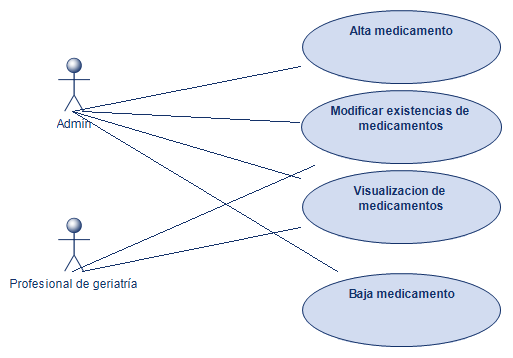
| **CU-3.2.2.3** | **Análisis de coincidencias** | |
| --- | --- | --- |
| **Subsistema** | Gestión de patologías de clientes | |
| **Descripción** | Se compara las patologías ingresadas con la base de datos previamente creada | |
| **Actores** | Actor principal que invoca el Caso de Uso: *Empleado* | |
| **Descrito por** | Andrés García de Arboleya Ortíz | |
| **Prioridad** | Alta | |
| **Estabilidad** | Baja | |
| **Precondiciones** | La actualización exitosa de la base de datos con las patologías de cada cliente | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | A partir de la base de datos se realiza las comparaciones de patologías y enfermedades que pueden coincidir con las patologías de los clientes |
|  | **2** | Una vez hecha la comparación de datos tenemos los resultados y se actualiza los datos de los clientes |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se puede hacer con éxito la comparación y se actualizan la información de los clientes.  Si fallo: No se puede hacer con éxito la comparación debido a 2.a y por lo tanto no se puede diagnosticar debidamente | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo se produce por datos incompletos o inexactos |
| **Requisitos no funcionales** | Un rendimiento eficiente en la comparación de la base de datos y sus patologías. | |
| **Comentarios** |  | |



### 

### Gestión de medicamentos

Subsistema dedicado a la gestión de medicamentos.

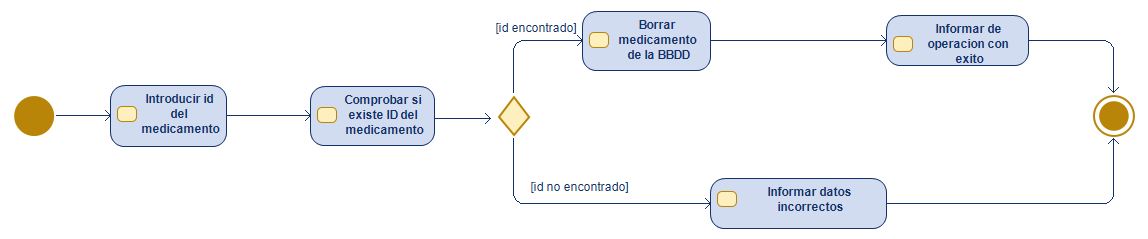


| **CU-3.2.3.1** | **Alta de medicamentos** | |
| --- | --- | --- |
| **Subsistema** | Gestión de medicamentos | |
| **Descripción** | Da de alta un medicamento, es decir, lo añade a la base de datos | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Sergio Cuenca López | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | ID no existente en la BBDD y cantidad a introducir válida | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El administrador accede a la interfaz de dar de alta |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a o 2.b |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se inserta el medicamento en la base de datos y se muestra un mensaje para indicar que la operación ha sido exitosa.  Si fallo: No se modifica ningún dato de la base de datos y se muestra un mensaje de error que indica cuál ha sido el problema por el que la alta del medicamento ha fallado. | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo es debido a que el medicamento ya está dado de alta en la BBDD. |
| **2.b** | El fallo es debido a que se ha introducido una cantidad no válida. |
| **Requisitos no funcionales** | Funcionamiento correcto de la base de datos. | |
| **Comentarios** |  | |

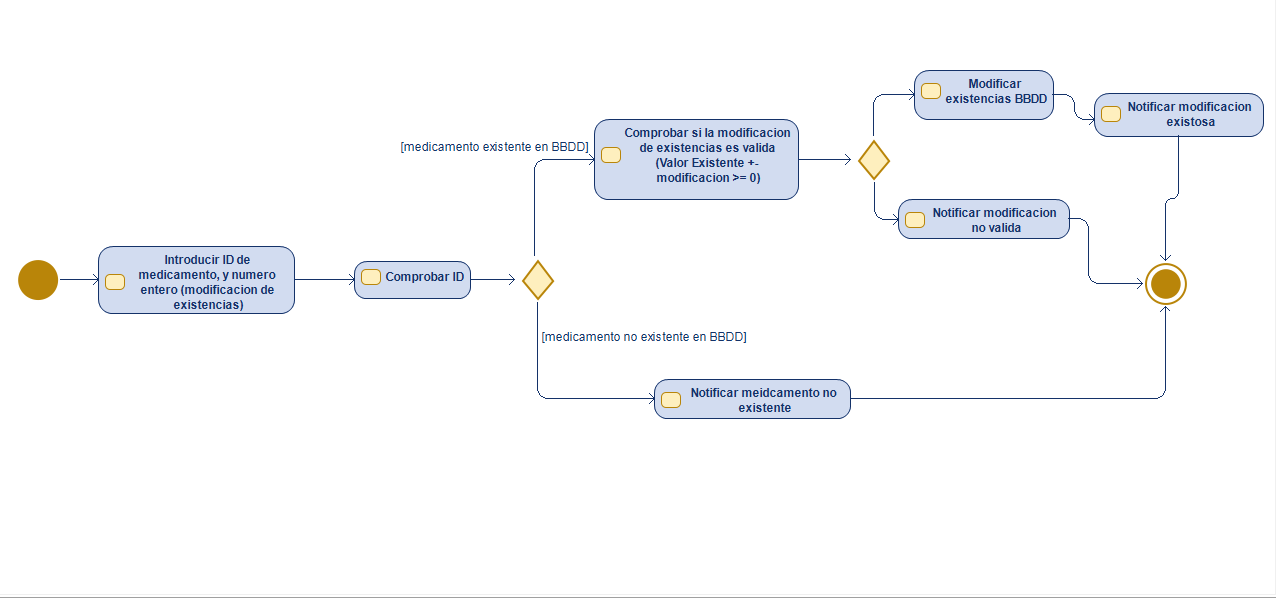
## 

| **CU-3.2.1.1** | **Baja de medicamentos** | |
| --- | --- | --- |
| **Subsistema** | Gestión de medicamentos | |
| **Descripción** | Da de baja un medicamento, es decir, lo elimina de la base de datos | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Sergio Cuenca López | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | ID existente en la BBDD | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El administrador accede a la interfaz de dar de baja |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a. |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se elimina el medicamento de la base de datos y se muestra un mensaje para indicar el éxito de esta operación  Si fallo: No se modifica ningún dato de la base de datos y se muestra un mensaje de error indicando que la ID introducida no se encuentra en la base de datos | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | El fallo es debido a que el medicamento no se ha encontrado en la BBDD, se muestra un mensaje indicándolo. |
|
| **Requisitos no funcionales** | Funcionamiento correcto de la base de datos. | |
| **Comentarios** |  | |

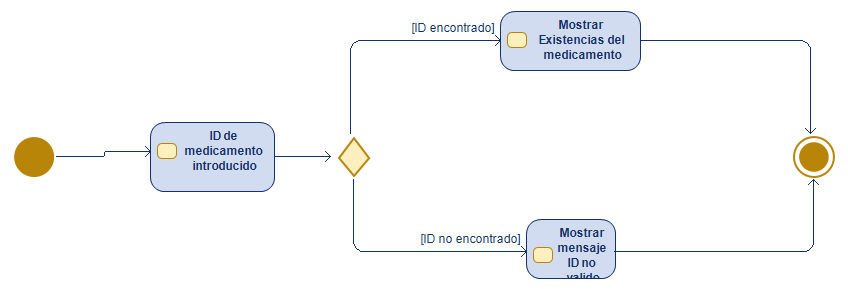
## 



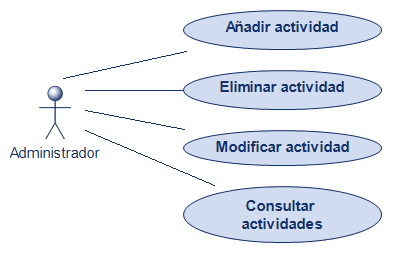
| **CU-3.2.3.2** | **Gestión de existencias de medicamentos** | |
| --- | --- | --- |
| **Subsistema** | Gestión de medicamentos | |
| **Descripción** | Cambia la cantidad de existencias de un medicamento en particular. | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador, profesional de geriatría.* | |
| **Descrito por** | Sergio Cuenca López | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | ID existente en la BBDD, cantidad válida | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El actor accede a la interfaz de gestión de existencias |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a o 2.b. |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se modifica la cantidad de el medicamento específico en la base de datos  Si fallo: No se modifica ningún dato de la base de datos y se muestra un mensaje de error indicando que la ID introducida no se encuentra en la base de datos, o que la modificación no es válida. | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | Fallo debido a que el ID introducido no se encuentra en la base de datos, se muestra una notificación por pantalla de este error. |
| **2.b** | Fallo debido a que la cantidad tras intentar actualizar era menor que cero por lo que se notifica al actor del error. |
| **Requisitos no funcionales** | Funcionamiento correcto de la base de datos. | |
| **Comentarios** |  | |



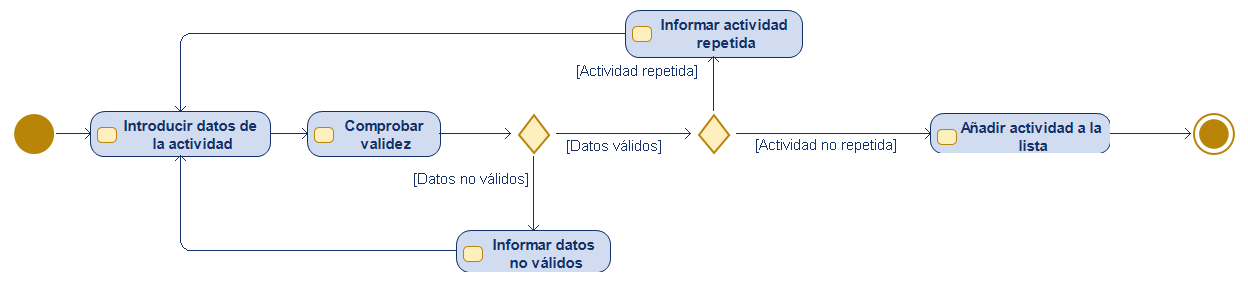
| **CU-3.2.3.3** | **Visualizar medicamento** | |
| --- | --- | --- |
| **Subsistema** | Gestión de medicamentos | |
| **Descripción** | Muestra la cantidad de existencias que hay de un medicamento en particular. | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador, profesional de geriatría.* | |
| **Descrito por** | Sergio Cuenca López | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | ID existente en la BBDD | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El actor accede a la interfaz de gestión de existencias |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a. |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se muestra la cantidad de existencias que hay de ese medicamento.  Si fallo: Se muestra un mensaje de error indicando que la ID introducida no se encuentra en la base de datos | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | Fallo debido a que el ID introducido no se encuentra en la base de datos, se muestra una notificación por pantalla de este error. |
|
| **Requisitos no funcionales** | Funcionamiento correcto de la base de datos. | |
| **Comentarios** |  | |



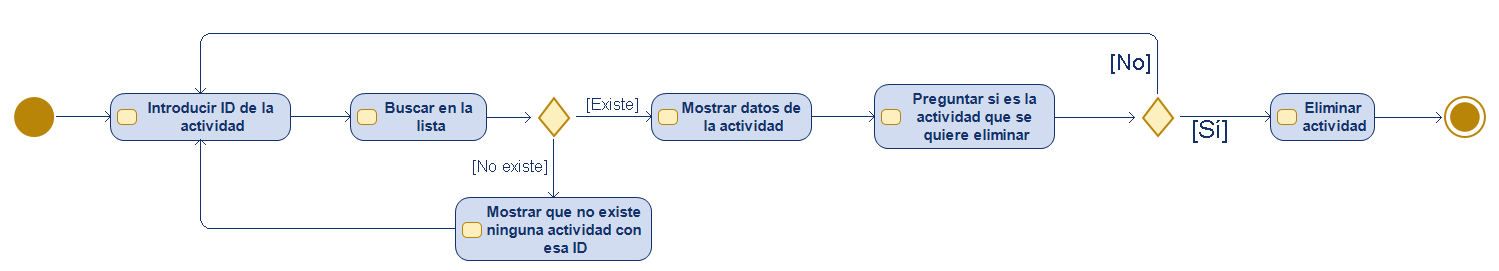
**Gestión de actividades**

****

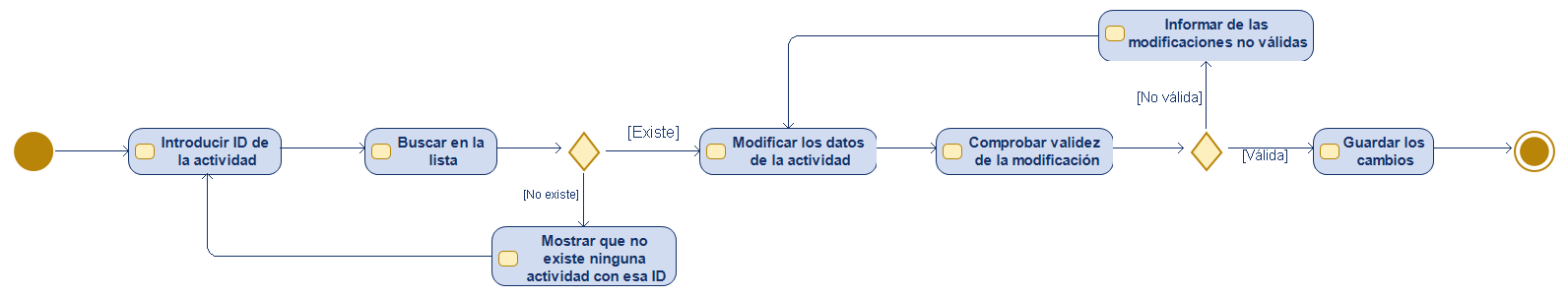
| **CU-3.2.4.1** | **Añadir actividad** | |
| --- | --- | --- |
| **Subsistema** | Gestión de actividades | |
| **Descripción** | Añade una actividad creada por el administrador a la lista de actividades | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Pablo Sánchez Redondo | |
| **Prioridad** | Baja | |
| **Estabilidad** | Alta | |
| **Precondiciones** | Fecha y hora de la actividad nueva no coincidentes con la de alguna de las actividades ya existentes y aforo igual o inferior al máximo permitido | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El actor accede a la interfaz de añadir actividad |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a. |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se añade la actividad creada a la lista de actividades.  Si fallo: Se muestra un mensaje de error indicando que los datos introducidos no son válidos o están repetidos y se vuelve al primer paso | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | Fallo debido a que los datos introducidos no son válidos (ej: la fecha introducida es anterior a la del momento en el que se realiza la gestión) o están repetidos, se muestra una notificación por pantalla de este error. |
|
| **Requisitos no funcionales** | Funcionamiento correcto de la base de datos. | |
| **Comentarios** |  | |

****

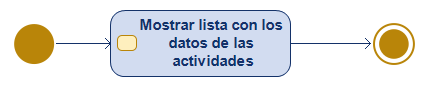
| **CU-3.2.4.2** | **Eliminar actividad** | |
| --- | --- | --- |
| **Subsistema** | Gestión de actividades | |
| **Descripción** | Elimina una actividad de la lista de actividades | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Pablo Sánchez Redondo | |
| **Prioridad** | Baja | |
| **Estabilidad** | Alta | |
| **Precondiciones** | El ID introducido debe pertenecer a alguna de las actividades de la lista | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El actor accede a la interfaz de eliminar actividad |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a. |
| **3** | Se muestra por pantalla la actividad introducida y se confirma si se quiere eliminar o no |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se elimina la actividad de la lista de actividades.  Si fallo: Se muestra un mensaje de error indicando que el ID introducido no existe en la lista y se vuelve al primer paso | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | Fallo debido a que el ID introducido no existe en la lista, se muestra una notificación por pantalla de este error. |
|
| **Requisitos no funcionales** | Funcionamiento correcto de la base de datos. | |
| **Comentarios** |  | |

****

| **CU-3.2.4.3** | **Modificar actividad** | |
| --- | --- | --- |
| **Subsistema** | Gestión de actividades | |
| **Descripción** | Modifica una actividad de la lista de actividades | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Pablo Sánchez Redondo | |
| **Prioridad** | Baja | |
| **Estabilidad** | Media | |
| **Precondiciones** | El ID introducido debe pertenecer a alguna de las actividades de la lista | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El actor accede a la interfaz de modificar actividad |
| **2** | El administrador introduce los datos en la interfaz. Si fallo 2.a. |
| **3** | El software comprueba que las modificaciones realizadas son válidas. Si fallo 2.b |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se modifica la actividad de la lista de actividades.  Si fallo: Se muestra un mensaje de error indicando que el ID introducido no existe en la lista o la modificación no es válida y se vuelve al primer paso | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
| **2.a** | Fallo debido a que el ID introducido no existe en la lista, se muestra una notificación por pantalla de este error. |
| **2.b** | Fallo debido a que las modificaciones introducidas no son válidas (ej: la fecha introducida es anterior a la del momento en el que se realiza la gestión) |
| **Requisitos no funcionales** | Funcionamiento correcto de la base de datos. | |
| **Comentarios** |  | |

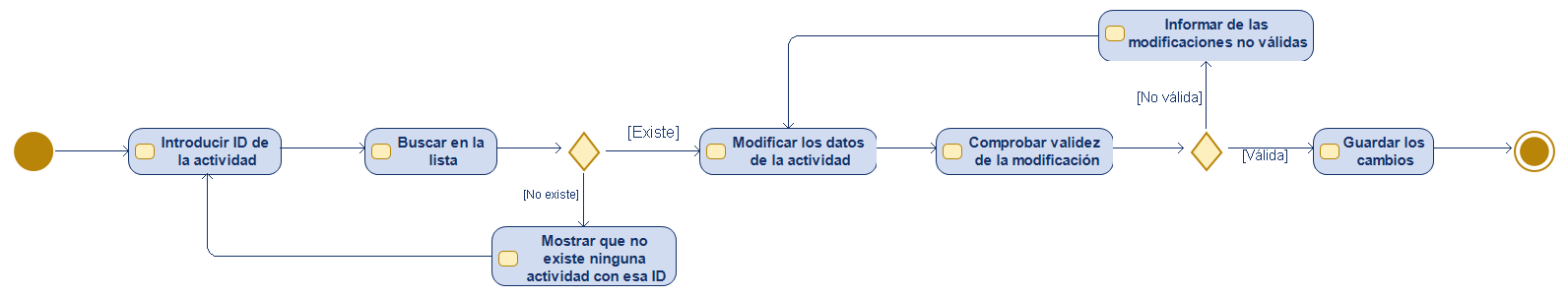
****

| **CU-3.2.4.4** | **Consultar actividades** | |
| --- | --- | --- |
| **Subsistema** | Gestión de actividades | |
| **Descripción** | Muestra por pantalla la lista de actividades | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Pablo Sánchez Redondo | |
| **Prioridad** | Baja | |
| **Estabilidad** | Alta | |
| **Precondiciones** |  | |
| **Flujo de eventos principal** | **Paso** | **Acción** |
| **1** | El actor accede a la interfaz de consultar actividades |
| **Postcondiciones**  **Éxito:**  **Fallo:** | Si éxito: Se muestra por pantalla la lista de actividades.  Si fallo: Se muestra un mensaje de error indicando que el acceso a la base de datos ha fallado | |
| **Flujo de eventos alternativo** | **Paso** | **Acción** |
|  |  |
|
| **Requisitos no funcionales** | Funcionamiento correcto de la base de datos. | |
| **Comentarios** |  | |

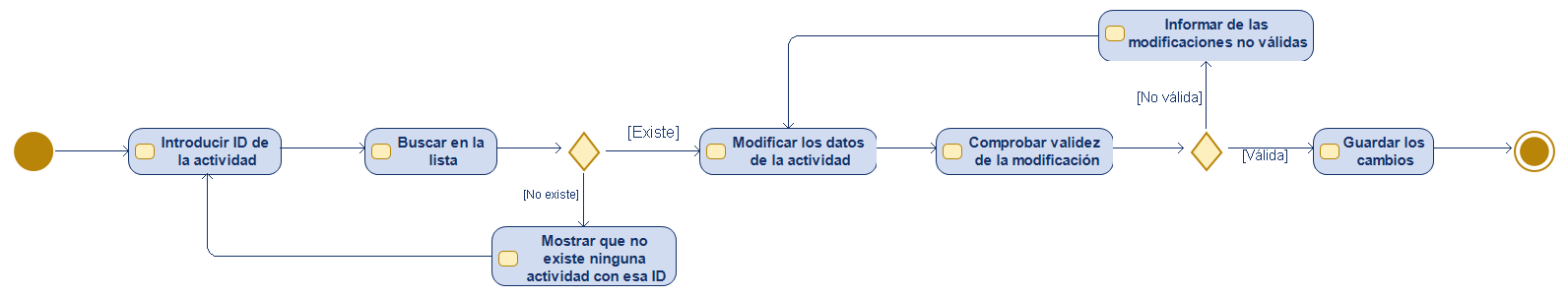


**Gestión de pagos**

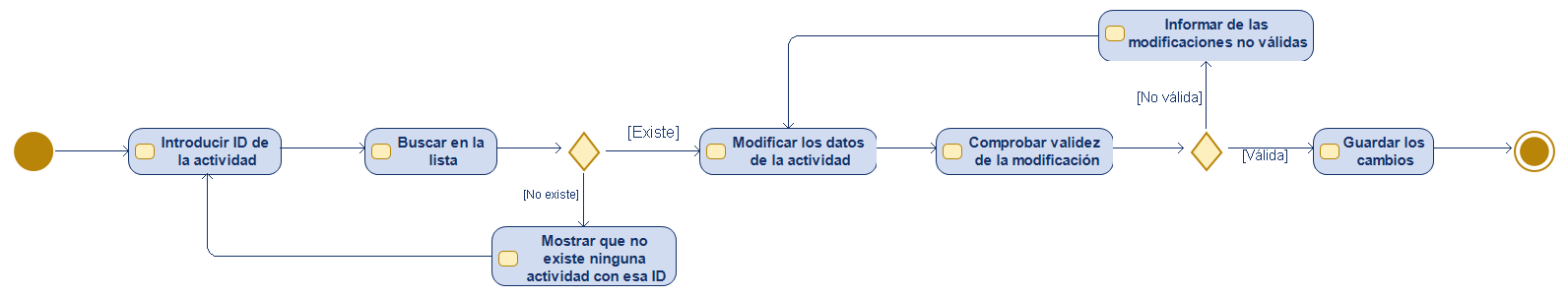
| **CU-3.2.5** | **ALTA EN LA ESTANCIA** | |
| --- | --- | --- |
| **Subsistema** | Gestión de pagos | |
| **Descripción** | Registra el pago asociado a la estancia de un cliente en una habitación. | |
| **Actores** | Actor principal que invoca el Caso de Uso: *administrador* | |
| **Descrito por** | Jose Antonio Almendros Corrales | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | No debe haber DNI repetidos (no estar en otra habitación). | |
| **Flujo de eventos principal** | **Paso** |  |
| **1** | El administrador accede a la interfaz de dar de alta |
|  | **2** | El administrador introduce los datos en la interfaz.   * **2.a:** Si hay un fallo debido a la relación del DNI del usuario con otra habitación, se muestra un mensaje y se vuelve al paso 1. * **2.b:** Si el fallo es debido a la ocupación de todas las habitaciones, se muestra un mensaje y se vuelve al paso 1. |
|  | **3** | * **Éxito:** Se inserta una relación entre habitación y cliente en la base de datos. |
|  | **4** | * **Fallo:** No se produce ninguna relación y se vuelve a la interfaz de dar de alta, informando al cliente del mensaje que aparece. |
| **Postcondiciones**  **Éxito:**  **Fallo:** | * **Éxito:** Se ha registrado el pago asociado a la estancia en la base de datos. * **Fallo:** No se ha registrado ningún pago, y se informa al usuario del mensaje correspondiente. | |
| **Flujo de eventos alternativo** | **Paso** |  |
| **2.a** | El fallo producido es debido a la relación del DNI del usuario con otra habitación, volviendo al paso 1 y mostrando en la interfaz un mensaje: "Informar al usuario de que ya dispone de otra habitación." |
|
|  | **2.b** | El fallo es debido a la ocupación de todas las habitaciones, volviendo al paso 1 y mostrando el mensaje: "No hay habitaciones disponibles." |
| **Requisitos no funcionales** | La disponibilidad de esta función está limitada al horario de la residencia, cuando tiene sus puertas abiertas y puede atender al cliente. | |
| **Comentarios** |  | |

****

| **CU-3.2.5** | **GENERAR REPORTE DE PAGOS** | |
| --- | --- | --- |
| **Subsistema** | Gestión de pagos | |
| **Descripción** | Facilita la generación de informes detallados sobre los pagos realizados en un período específico. | |
| **Actores** | *Administrador, Personal de Contabilidad* | |
| **Descrito por** | Jose Antonio Almendros Corrales | |
| **Prioridad** | Media | |
| **Estabilidad** | Alta | |
| **Precondiciones** | - | |
| **Flujo de eventos principal** | **Paso** |  |
| **1** | El personal de contabilidad accede al módulo de gestión de pagos. |
|  | **2** | Selecciona la opción "Generar Reporte de Pagos". |
|  | **3** | 3. Especifica el rango de fechas para el informe. |
|  | **4** | 4. El sistema genera un informe con los detalles de los pagos realizados durante ese período. |
| **Postcondiciones**  **Éxito:**  **Fallo:** | * **Éxito:**Se genera el informe de pagos. * **Fallo:** No se genera el informe y se informa al usuario del mensaje correspondiente. | |
| **Flujo de eventos alternativo** | **Paso** |  |
|  | Si no hay pagos registrados para el período especificado, el sistema muestra un mensaje informativo. |
|
| **Requisitos no funcionales** | - | |
| **Comentarios** |  | |

****

| **CU-3.2.5** | **GESTIONAR DEUDAS** | |
| --- | --- | --- |
| **Subsistema** | Gestión de pagos | |
| **Descripción** | Permite gestionar las deudas pendientes de los residentes y registrar pagos parciales o planes de pago. | |
| **Actores** | Administrador, Personal de Contabilidad | |
| **Descrito por** | Jose Antonio Almendros Corrales | |
| **Prioridad** | Alta | |
| **Estabilidad** | Alta | |
| **Precondiciones** | - | |
| **Flujo de eventos principal** | **Paso** |  |
| **1** | El personal de contabilidad accede al módulo de gestión de pagos. |
|  | **2** | Selecciona la opción "Gestionar Deudas". |
|  | **3** | Visualiza la lista de residentes con deudas pendientes. |
|  | **4** | Selecciona a un residente específico para ver los detalles de la deuda. |
|  | **5** | Registra el pago parcial o total de la deuda. |
| **Postcondiciones**  **Éxito:**  **Fallo:** | * **Éxito:** Se registra el pago de la deuda en la base de datos. * **Fallo:** No se registra el pago y se informa al usuario del mensaje correspondiente. | |
| **Flujo de eventos alternativo** | **Paso** |  |
|  | Si un residente no puede pagar la deuda completa, el sistema permite registrar un plan de pago. |
|
| **Requisitos no funcionales** | - | |
| **Comentarios** |  | |

****

## Requisitos de rendimiento

Debido a que el programa está destinado a residencias de una extensión pequeña, dispondremos de como mucho cuatro terminales en cada residencia, para poder trabajar con la aplicación. Destinando 2 para trabajos de cara al público(como dar de alta, que tienes que hablar con el cliente), y otras 2 para gestionar cosas internas (como la planificación de actividades, en la que no necesitamos interactuar con el cliente). Al ser tan pocas terminales, podrán tener acceso simultáneo las cuatro a la vez. El software deberá de estar conectado a la base de datos correspondiente cada vez que interactúe el administrador con la aplicación.

## Requisitos lógicos de la base de datos

Profesional geriatrico: nombre(string), DNI(string), contraseña(string)

Habitación: numero(int), ocupado(boolean)

Residentes: nombre(string), DNI(string), duración(int)

Patologías: id(int), nombre(string), descripción(string)

Medicamentos: id(string), nombre(string), cantidad(int)

Actividades: id(string), nombre(string), descripcion(string), aforo(int), horario(date)

Deudas: id(string), dni(string), cantidad(int)

|  | Profesional geriatrico | Habitación | Residentes | Patologías | Medicamentos | Actividades | Deudas |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Profesional geriatrico |  |  | 1..\* |  |  | \* |  |
| Habitación |  |  | 1..\* |  |  |  |  |
| Residentes | 1..\* | 1 |  | \* | \* | \* | 1 |
| Patologías |  |  | \* |  | 1..\* |  |  |
| Medicamentos |  |  | 1..\* | \* |  |  |  |
| Actividades | 1..\* |  | 1..\* |  |  |  |  |
| Deudas |  |  | \* |  |  |  |  |

## Restricciones de diseño

Para este proyecto hemos seguido el estándar IEEE Std. 830-1998. El lenguaje en el que pretendemos desarrollar el sistema es en java, porque dentro del equipo de desarrollo están acostumbrados a este lenguaje, específicamente orientado a objetos.

## Atributos del sistema software

El software, al estar dirigido a residencias de poca extensión, será un software básico e intuitivo, para hacer su uso más fácil, y, además, permitir un buen mantenimiento a medida de que se vaya necesitando implementar más funciones o modificarlas. A la vez deberá de ser seguro, para que gente externa no acceda al sistema

## Otros requisitos

* Todos los módulos están conectados directa, o indirectamente a una base de datos.
* La interfaz, como hemos dicho antes, debe ser intuitiva, para su fácil uso. Y el tiempo de espera mientras que se produce una acción deberá de ser de 1 segundo máximo. A su vez, los servidores tienen que estar preparados para soportar una afluencia de información moderada, debido a la pequeña extensión de las residencias a las que va dirigido el software.
* Cada vez que haya un fallo en cualquier función, te devolverá al interfaz de inicio de ésta.
* Los datos de las bases de datos sólo podrán ser modificados por los administradores.
* Si en algún momento queremos desarrollar un nuevo módulo, tomaremos éstos como referencia a la hora de estimar la duración y los costes.