



**Universidad  
Zaragoza**

**Historia Clínica  
Especificación de Requisitos**


Pablo Doñate Navarro

Adnana Dragut

Proyecto Software

Versión: 1.1

Fecha: 23/05/2022

 <b>Universidad</b> Zaragoza	<b>Historia Clínica</b> <b>Especificación de Requisitos</b>	<b>Pablo Doñate</b> <b>Adnana Dragut</b>
------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------	---------------------------------------------

## HOJA DE CONTROL

<b>Proyecto</b>	Historia Clínica		
<b>Entregable</b>	Especificación de Requisitos		
<b>Autores</b>	Pablo Doñate Navarro y Adnana Dragut		
<b>Versión/Edición</b>	1.1	<b>Fecha Versión</b>	23/05/2022
<b>Aprobado por</b>	Sin aprobación	<b>Fecha Aprobación</b>	-
		<b>Nº Total de Páginas</b>	24

## REGISTRO DE CAMBIOS

Versión	Causa del Cambio	Responsables del Cambio	Fecha del Cambio
1.0	Versión inicial.	Pablo Doñate Navarro Adnana Dragut	04/03/2022
1.1	Supervisión de riesgos	Pablo Doñate Navarro Adnana Dragut	23/05/2022

## CONTROL DE DISTRIBUCIÓN

Nombre y Apellidos
Pablo Doñate Navarro
Adnana Dragut



## TABLA DE CONTENIDOS

<b>INTRODUCCIÓN</b>	<b>5</b>
Propósito	5
Ámbito	5
Referencias	6
Descripción del documento	6
<b>DESCRIPCIÓN GENERAL</b>	<b>7</b>
Perspectiva del producto	7
Interfaces de usuario	7
<i>Interfaces software</i>	7
<i>Interfaces de comunicaciones</i>	7
<i>Modos de operación</i>	8
<i>Necesidades de infraestructura (para el alojamiento u operación)</i>	8
Funcionalidad del Producto	8
Características de los usuarios	12
Restricciones	12
<i>Regulación normativa</i>	12
<i>Operaciones paralelas</i>	12
<i>Auditoría</i>	12
<i>Lenguajes de programación</i>	13
<i>Protocolos de comunicaciones</i>	13
<i>Seguridad</i>	13
Asunciones y dependencias	13
Requisitos posibles en el futuro	13
<b>GESTIÓN DE RIESGOS</b>	<b>14</b>
<b>REQUISITOS ESPECÍFICOS</b>	<b>17</b>
Interfaces externos	17
<i>Interfaces de usuario</i>	17
<i>Interfaces hardware</i>	17
<i>Interfaces software</i>	17
Características del sistema	17
Arranque / Parada	17
Funcionalidad App Cliente	18
Funcionalidad App Servidor	20



Requisitos de prestaciones	22
Requisitos de persistencia	23
Restricciones de diseño	23
Atributos del sistema	23
<i>Fiabilidad</i>	23
<i>Disponibilidad</i>	23
<i>Seguridad</i>	23
<i>Mantenibilidad</i>	24
<i>Portabilidad</i>	24



## 1. INTRODUCCIÓN

Este documento es la Especificación de Requisitos Software (SRS) para el desarrollo de una aplicación software destinada a la gestión de la historia clínica de un hospital. Se ha estructurado basándose en las directrices dadas por el estándar IEEE Recommended Practice for Software Requirements Specifications **IEEE 830-1998**.

### 1.1. Propósito

La aplicación servirá para que los trabajadores de centros de atención primaria puedan registrar y consultar información asociada a sus pacientes.

Este software se ha considerado adecuado para que los trabajadores puedan tener toda la información en formato digital, y de este modo, las tareas de consulta, gestión y almacenamiento de los datos de los pacientes sean más rápidas y sencillas de realizar.

Las principales funciones que permitirá esta historia clínica electrónica serán las de consultar la información relativa a un paciente, registrar nuevos diagnósticos, permitir la gestión de recetas electrónicas, registrar medicamentos, registrar las vacunas administradas o registrar planes de tratamiento asociados a cada paciente y diagnóstico.

### 1.2. Ámbito

La historia clínica electrónica es un software propuesto para la realización del trabajo dirigido de la asignatura de Proyecto Software, para poder aprender a realizar la gestión de un proyecto amplio de desarrollo software, así como para comprender y saber aplicar las diferentes fases del ciclo de vida de una aplicación informática.

Este proyecto servirá para poder sentar las bases del desarrollo de software, permitiendo participar en todas las fases del mismo. Es por ello por lo que los objetivos que se persiguen con la realización de este proyecto son los siguientes:

- Conocer y aplicar las fases de gestión de un proyecto, que incluye la definición de las actividades a llevar a cabo, la planificación y calendarización del proyecto, así como la realización de un análisis de riesgos.
- Conocimiento y aplicación del modelo RUP.
- Llevar a cabo las cuatro fases básicas de desarrollo software: análisis, diseño, implementación y pruebas.
- Poder calcular los costes derivados de la puesta en marcha de un proyecto informático.



### 1.3. Referencias

#### Fuentes bibliográficas:

[1] <https://www.boe.es/eli/es/l/2002/11/14/41/con>

#### Documentos:

- i. IEEE Recommended Practice for Software Requirements Specifications," in *IEEE Std 830-1998*, vol., no., pp.1-40, 20 oct. 1998
- ii. ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirement's engineering," in *ISO/IEC/IEEE 29148:2018(E)*, vol., no., pp.1-104, 30 nov. 2018

### 1.4. Descripción del documento

En el apartado 2 del documento, se hace una descripción genérica de la funcionalidad del software "Historia Clínica". Para ello se presenta un diagrama UML de casos de uso que muestra los actores que interactúan con el sistema, así como su funcionalidad principal organizada en base a una serie de escenarios. También se identifican los requisitos genéricos (requisitos de usuario), las características de los posibles tipos de usuarios y las restricciones que deberán considerarse.

Seguido, en el capítulo 3 se va a desarrollar el análisis de riesgos, con el objetivo de identificar los posibles candidatos, y por tanto evaluar su impacto en el proyecto, siendo un riesgo la probabilidad de que una circunstancia adversa ocurra.



## 2. DESCRIPCIÓN GENERAL

### 2.1. *Perspectiva del producto*

El proyecto “Historia Clínica”, se considera un producto totalmente autónomo, ya que no se va a necesitar de ningún otro servicio para funcionar.

Añadir que este software está muy relacionado con el ya implementado en los hospitales, pero como es un proyecto para el desarrollo de una práctica, éste no va a representar la misma funcionalidad que los actuales.

#### 2.1.1. Interfaces de usuario

- ❖ La aplicación únicamente deberá de considerar la interfaz asociada a un ordenador.
- ❖ Las interfaces serán descritas con mayor detalle en el apartado de prototipado de documento de Diseño.

#### 2.1.2. Interfaces software

- ❖ Como se ha indicado en el apartado 2.1, la aplicación no dispondrá de la información proveniente de fuentes externas, motivo por el cual no se considera necesaria ninguna interfaz software.

#### 2.1.3. Interfaces de comunicaciones

- ❖ La interfaz de comunicación entre el servidor de bases de datos y la aplicación, se podrá realizar con cualquier API con acceso a datos.
- ❖ El protocolo de comunicación Cliente / Servidor que se va a utilizar será el habitual para este tipo de sistemas.



#### 2.1.4. Modos de operación

❖ La aplicación podrá ser utilizada por dos tipos de usuarios, ambos con acceso mediante una interfaz de escritorio:

- El usuario administrador.
- El usuario estándar, asociado al personal sanitario.

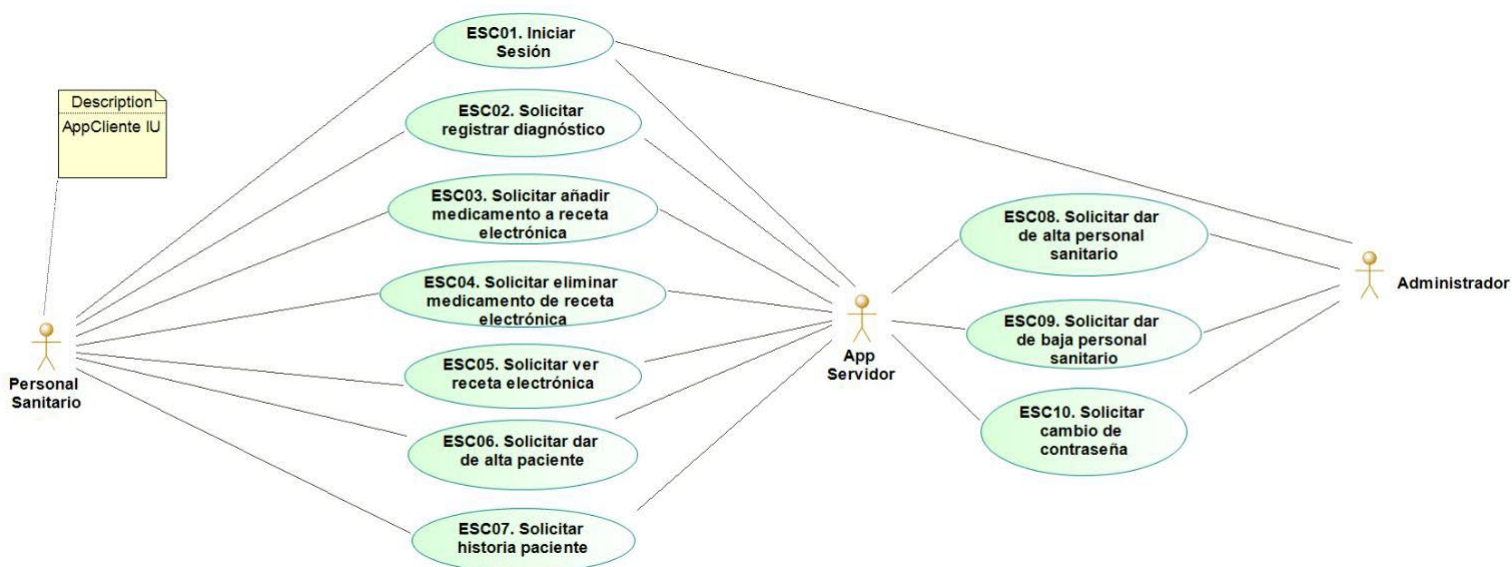
#### 2.1.5. Necesidades de infraestructura (para el alojamiento u operación)

- ❖ Servidor de Base de Datos.
- ❖ Clúster de servidores, con el objetivo de ofrecer velocidad y sobre todo alta disponibilidad ante fallos.

### 2.2. Funcionalidad del Producto

En este apartado se mostrarán varios diagramas de CU, asociados a la funcionalidad que proporciona el software a sus actores, es decir, la funcionalidad refejada en los casos de uso de los siguientes diagramas:

#### APP Cliente







En el caso del actor “Personal Sanitario”, las funcionalidades que podrá realizar, y que serán llevadas a cabo por el actor “AppServidor”, son las siguientes:

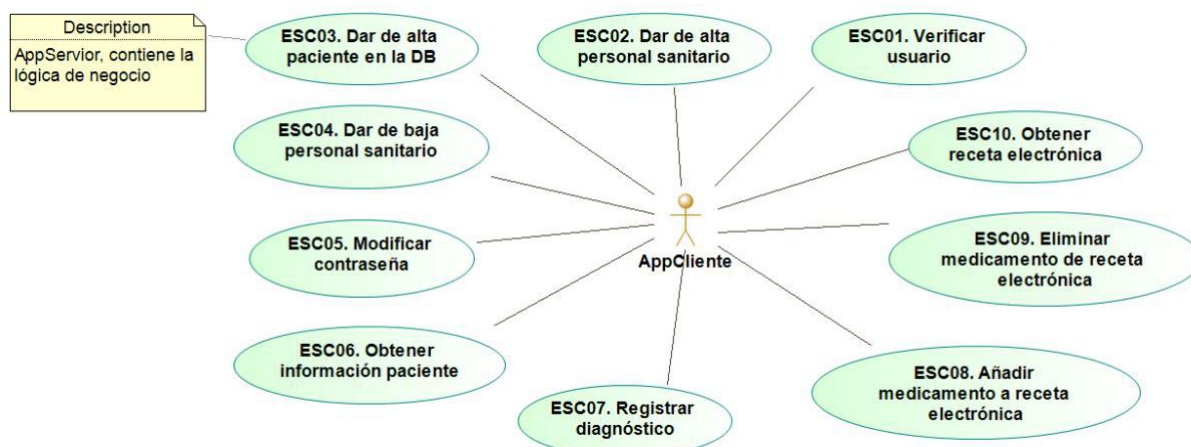
- “Iniciar Sesión”: el personal sanitario previamente registrado en el sistema podrá acceder al mismo, mediante el envío de sus datos de inicio de sesión a la App Servidor para que éste pueda verificarlos.
- “Solicitar registrar diagnóstico”: el personal sanitario tendrá la capacidad de registrar un diagnóstico asociado a un paciente, enviando la solicitud correspondiente a la App Servidor.
- “Solicitar añadir medicamento a receta electrónica”: el personal sanitario podrá prescribir de forma electrónica cualquier medicamento, añadiéndolo a la receta de un determinado paciente, mediante el envío de dicha solicitud a la App Servidor.
- “Solicitar eliminar medicamento de receta electrónica”: el personal sanitario podrá eliminar de la receta electrónica de un paciente cualquier medicamento que considere oportuno, mediante el envío de dicha solicitud a la App Servidor.
- “Solicitar ver receta electrónica”: el personal sanitario podrá solicitar ver la medicación asociada a una receta electrónica, mediante el envío de dicha solicitud a la App Servidor.
- “Solicitar dar de alta paciente”: el personal sanitario podrá solicitar añadir un nuevo paciente a la DB del sistema, enviando dicha solicitud a la App Servidor.
- “Solicitar historia paciente”: el personal sanitario podrá obtener toda la información de un paciente almacenada en el sistema, mediante el envío de dicha solicitud a la App Servidor.

En el caso del actor “Administrador”, las funcionalidades que podrá realizar, y que serán llevadas a cabo por el actor “AppServidor”, son las siguientes:

- “Iniciar Sesión”: el administrador podrá acceder al sistema, mediante el envío de sus datos de inicio de sesión a la App Servidor para que éste pueda verificarlos.
- “Solicitar dar de alta personal sanitario”: el administrador podrá introducir a nuevos miembros del personal sanitario, enviando la solicitud correspondiente a la App Servidor.

- “Solicitar dar de baja personal sanitario”: el administrador podrá eliminar miembros del personal sanitario, previamente registrados, enviando la solicitud correspondiente a la App Servidor.
- “Solicitar cambio de contraseña”: el administrador podrá solicitar el cambio de la contraseña de un miembro del personal sanitario, enviando la solicitud correspondiente a la App Servidor.

## APP Servidor



Las funcionalidades del servidor que podrán solicitar los actores administrador y personal sanitario, a través de la interfaz de usuario, son las siguientes:

- “Verificar usuario”: el servidor comprobará la existencia de un usuario realizando la consulta necesaria a la DB, como respuesta a la llegada de la solicitud correspondiente al inicio de sesión.
- “Dar de alta personal sanitario”: el servidor añadirá un nuevo miembro del personal sanitario a la DB, como respuesta a la llegada de la petición correspondiente a solicitar dar de alta personal sanitario.
- “Dar de alta paciente en la DB”: el servidor añadirá un nuevo paciente a la DB, como respuesta a la llegada de la petición correspondiente a solicitar dar de alta paciente.



- “Dar de baja personal sanitario”: el servidor eliminará a un miembro del personal sanitario existente en la DB, como respuesta a la llegada de la petición correspondiente a solicitar dar de baja personal sanitario.
- “Modificar contraseña”: el servidor modificará la contraseña de un miembro del personal sanitario, como respuesta a la llegada de la petición correspondiente a solicitar cambio de contraseña.
- “Obtener información paciente”: el servidor devolverá la información de un determinado paciente, como respuesta a la llegada de la petición correspondiente a solicitar historia paciente.
- “Registrar diagnóstico”: el servidor añadirá un nuevo diagnóstico asociado a un paciente en la DB, como respuesta a la llegada de la petición correspondiente a solicitar registrar diagnóstico.
- “Añadir medicamento a receta electrónica”: el servidor añadirá un nuevo medicamento asociado a la receta electrónica de un paciente en la DB, como respuesta a la llegada de la petición correspondiente a solicitar añadir medicamento a receta electrónica.
- “Eliminar medicamento de receta electrónica”: el servidor eliminará un medicamento asociado a la receta electrónica de un paciente de la DB, como respuesta a la llegada de la petición correspondiente a solicitar eliminar medicamento de receta electrónica.
- “Obtener receta electrónica”: el servidor devolverá la receta electrónica de un paciente, como respuesta a la llegada de la petición correspondiente a solicitar ver receta electrónica.



### 2.3. Características de los usuarios

En esta aplicación, pueden considerarse 2 tipos de usuarios:

- ❖ Administrador:
  - Puede gestionar al resto de usuarios: darles de alta, cambiar sus contraseñas y darles de baja.
- ❖ Usuario (personal sanitario):
  - Puede visualizar toda la historia clínica registrada de un paciente.
  - Puede gestionar la receta electrónica del paciente.
  - Puede dar de alta pacientes en la DB.

### 2.4. Restricciones

#### 2.4.1. Regulación normativa

Los datos relativos a la salud, están considerados como información sensible, por lo que deben de estar especialmente protegidos. Por ello, este producto está obligado a cumplir la Ley de confidencialidad del paciente, que está estipulada en la Ley 41/2002 <sup>[1]</sup> de autonomía del paciente, la cual señala lo siguiente: *“Toda persona tiene derecho a que se respete el carácter confidencial de los datos referentes a su salud, y a que nadie pueda acceder a ellos sin previa autorización amparada en la Ley”*.

#### 2.4.2. Operaciones paralelas

Como se ha mencionado anteriormente, al trabajar con tantos datos, que pueden ser accedidos por varios clientes simultáneamente, es necesario de un clúster de servidores para repartir la carga de trabajo, y por lo tanto, acelerar ese acceso continuo a datos.

#### 2.4.3. Auditoría

Con el objetivo de satisfacer los requisitos de las múltiples jurisdicciones regulatorias, será necesario la realización de auditorías del sistema de gestión de calidad, que pueden ser realizadas por MDSAP, el programa de Auditoría Unificado de Productos Sanitarios.

<https://www.imdrf.org/working-groups/medical-device-single-audit-program-mdsap>



#### **2.4.4. Lenguajes de programación**

Aunque aún está por determinar, el lenguaje de programación que se va a utilizar va a ser o Python o Java.

En cuanto a las consultas que se van a realizar a la DB, será en lenguaje SQL.

#### **2.4.5. Protocolos de comunicaciones**

El protocolo de comunicación que se va a utilizar puede ser el habitual para este tipo de sistemas, que es TCP con MySQL.

El protocolo de comunicación con la DB, al igual que en el apartado anterior, está aún por determinar, en función del lenguaje de programación.

#### **2.4.6. Seguridad**

Para garantizar la seguridad en la aplicación, el acceso a ésta será únicamente posible si un administrador nos ha dado de alta. De esta forma, se controla el acceso no autorizado.

### **2.5. *Asunciones y dependencias***

- ❖ Los equipos en donde sea desplegada la aplicación deben contar con un mínimo de recursos para el correcto funcionamiento.
- ❖ La empresa debe contar con una persona con conocimientos suficientes para llevar a cabo la administración de la aplicación.

### **2.6. *Requisitos posibles en el futuro***

Incluir funcionalidad adicional, que como se ha comentado anteriormente, no lleva integrada la misma funcionalidad que el software que existe en la actualidad, ya que es un proyecto de la asignatura “Proyecto Software”.

La funcionalidad que podría ser integrada en un futuro, sería la de dar acceso al sistema a los pacientes, de forma que éstos puedan consultar su historia clínica, o citologías.



### 3. GESTIÓN DE RIESGOS

En este apartado, se va a realizar el análisis de riesgos, es decir, se va a identificar los posibles candidatos, y se va a evaluar su posible impacto en el proyecto. Para ello, se van a seguir las siguientes 4 etapas:

#### 3.1. Identificación

En esta etapa, se va a realizar el descubrimiento de riesgos, mediante la técnica de Tormenta de ideas.

Los riesgos encontrados son los siguientes:

- Riesgo 1. La aparición de nuevos requisitos en etapas posteriores que cubran las necesidades del proyecto.

Este riesgo, es del tipo “Riesgo de Requerimientos”, y pertenece a la categoría de “Riesgos del proyecto”, ya que afecta a la temporización del mismo.

- Riesgo 2. Que se produzca una rotación del personal durante el desarrollo del proyecto.

Este riesgo, es del tipo “Riesgo de personal”, y pertenece a la categoría de “Riesgos del proyecto”, ya que afecta a la temporización del mismo.

- Riesgo 3. Que se produzca un cambio del lenguaje de programación durante la fase de implementación del proyecto (por desconocimiento del mismo, o por dificultad técnica).

Este riesgo, es del tipo “Riesgo técnico”, y pertenece a la categoría de “Riesgos del proyecto”, ya que afecta a la temporización del mismo.

- Riesgo 4. Calendarización inadecuada para poder llevar a cabo con éxito las distintas fases del proyecto.

Este riesgo, es del tipo “Riesgo de estimación”, y pertenece a la categoría de “Riesgos del proyecto”, ya que afecta a la temporización del mismo.



- Riesgo 5. Elección inadecuada de la base de datos, que provoca un mal rendimiento de la aplicación.

Este riesgo, es del tipo “Riesgo de tecnología”, y pertenece a las categorías de “Riesgos del proyecto”, ya que afecta a la temporización del mismo, y de “Riesgo de producto”, ya que también afecta al rendimiento de la app.

### 3.2. *Análisis*

En esta etapa, se va a considerar para cada riesgo identificado, su probabilidad, y el efecto asociado al mismo:

Riesgo	Probabilidad	Efecto
Riesgo 1	85% (Muy alta)	Serio
Riesgo 2	20% (Baja)	Catastrófico
Riesgo 3	55% (Alta)	Catastrófico
Riesgo 4	80% (Muy alta)	Tolerable
Riesgo 5	15% (Baja)	Serio

### 3.3. *Planificación*

En esta etapa, se va a decidir para cada riesgo contemplado, la estrategia a seguir:

- Riesgo 1: La estrategia que se va a seguir es la de Prevención, con el objetivo de reducir la probabilidad de que ocurra. Para ello, se podría realizar un estudio más exhaustivo de los requisitos (ampliando la duración de la fase de análisis) para evitar su aparición en el futuro.
- Riesgo 2: La estrategia que se va a seguir es la de Minimización, con el objetivo de reducir el impacto adverso. Para ello, se podría reorganizar el equipo de tal forma que cada miembro del equipo conozca el trabajo de los demás.
- Riesgo 3: La estrategia que se va a seguir es la de Prevención, con el objetivo de reducir la probabilidad de que ocurra. Para ello, se podría realizar un estudio más exhaustivo de los lenguajes en base a proyectos similares.



- Riesgo 4: La estrategia que se va a seguir es la de Minimización, con el objetivo de reducir el impacto adverso. Para ello, se podría dejar días libres cerca de tareas críticas con el fin de que, si una tarea se prolonga, pueda ocupar esos días libres y, por tanto, afecte en menor medida a la calendarización del proyecto.
- Riesgo 5: La estrategia que se va a seguir es la de Prevención, con el objetivo de reducir la probabilidad de que ocurra. Para ello, se podría realizar un estudio más exhaustivo de las actuales BD en proyectos similares.

### 3.4. Supervisión

En este apartado se comentará la evolución de cada uno de los riesgos detectados en el apartado 3.1, y si la estrategia planteada ha funcionado para evitar o minimizar los efectos de dicho riesgo. También se comentarán si se ha llegado a identificar otros tipos de riesgos no detectados en esta fase del ciclo de vida del desarrollo de nuestro software.

A continuación, se va a proceder a analizar cada uno de los riesgos anteriores, identificando para cada uno de ellos si se ha producido, y en tal caso, si la estrategia planteada anteriormente ha sido la correcta:

- Riesgo 1: Este riesgo no se ha llegado a producir, ya que no se han descubierto nuevos requisitos importantes que afecten el resto del proyecto. Es cierto que se ha considerado y se ha añadido nueva funcionalidad, sin embargo, esta funcionalidad no supone ningún cambio ni en el diseño ni en el análisis inicial del proyecto.
- Riesgo 2: Este riesgo no se ha producido, ya que todos los miembros del proyecto han mantenido los mismos roles desde el principio hasta el final del proyecto. Además, no ha habido ninguna incorporación nueva.
- Riesgo 3: Este riesgo no se ha producido. Hemos comenzado con lenguaje Java y hemos terminado con el mismo.
- Riesgo 4: Este riesgo se ha producido al comienzo del desarrollo del proyecto, ya que, debido a la falta de experiencia, la calendarización inicial no ha sido adecuada y muchas de las tareas se habían retrasado. Para solucionarlo se ha seguido la estrategia planteada en el apartado 3.3, concretamente se ha creado una nueva versión de la planificación con un mayor número de días libres próximos a las tareas que se han considerado críticas.
- Riesgo 5: Este riesgo no se ha producido. Se ha elegido una BD en MySQL, con la que no hemos tenido problemas en ningún momento en aspectos de rendimiento.





## 4. REQUISITOS ESPECÍFICOS

### 4.1. Interfaces externos

#### 4.1.1. Interfaces de usuario

REQ 1. El aspecto de la aplicación debe ser consistente e intuitivo en todas sus páginas.

REQ 2. En caso de que se produzca un error, la aplicación deberá notificar al usuario del mismo.

REQ 3. La aplicación deberá solicitar la información de acceso del usuario cada vez que éste acceda a la misma.

#### 4.1.2. Interfaces hardware

La aplicación no va a necesitar de ningún dispositivo hardware adicional, por lo que no cabe considerar ningún requisito en este aspecto.

#### 4.1.3. Interfaces software

REQ 4. La aplicación únicamente estará disponible para equipos que utilizan SO Windows.

### 4.2. Características del sistema

#### 4.2.1. Arranque / Parada

ESC01. Arranque

- Propósito

El escenario de arranque es habitual (y necesario) en todos los sistemas informáticos.

Contiene la serie de acciones que deben llevarse a cabo cuando se inicia la ejecución de la aplicación (proceso de arranque/boot), de forma previa a que dicha aplicación esté en condiciones de ser utilizada por sus usuarios.



- RQs asociados (RQs Funcionales)

RQ 5. En la fase de arranque, el sistema deberá comprobar que son accesibles los ficheros de configuración definidos para la aplicación, y que su formato es adecuado. Si esto no es así, deberá abortarse el arranque y emitir un mensaje de error.

RQ 6. En la fase de arranque, deberá comprobarse la existencia del usuario "admin". Si éste no existe, deberá crearse con una contraseña predefinida, que podrá obtenerse de un fichero de configuración.

#### ESC02. Parada

- Propósito

El escenario de parada también es necesario en toda aplicación informática. En él, la aplicación ya habrá dejado de estar operativa para sus usuarios, pero todavía deberán ejecutarse algunas acciones antes de que el sistema operativo pueda considerarla como finalizada.

- RQs asociados (RQs Funcionales)

RQ 7. En la fase de parada, el sistema deberá guardar todos los datos útiles generados desde el último arranque del programa.

RQ 8. Antes de que el programa finalice, el sistema deberá cerrar todas las comunicaciones que existan entre el programa y, los usuarios y servidores.

#### 4.2.2. Funcionalidad App Cliente

##### Requisitos → ESC01. Iniciar Sesión

RQ 9. La aplicación deberá de solicitar el inicio de sesión en todos los intentos de acceder a la aplicación, independientemente de que el usuario haya cerrado o no sesión en su acceso anterior.

RQ 10. En el proceso de autenticación, el usuario deberá introducir una dirección de correo válida, y dada de alta en el sistema previamente.

RQ 11. En el proceso de inicio de sesión, se dará la posibilidad al usuario de recuperar su contraseña si se le ha olvidado.



Requisitos → ESC02. Solicitar registrar diagnóstico

RQ 12. La aplicación deberá solicitar los datos de identificación del paciente.

RQ 13. La aplicación deberá permitir al usuario introducir la descripción asociada al diagnóstico de un paciente.

RQ 14. La aplicación deberá informar al usuario en el caso en el que los datos de identificación de un paciente sean incorrectos.

Requisitos → ESC03. Solicitar añadir medicamento a receta electrónica

RQ 15. La aplicación deberá mostrar al usuario una lista completa con los medicamentos disponibles.

RQ 16. La aplicación deberá permitir al usuario seleccionar el medicamento que desea introducir en la receta electrónica de un paciente.

Requisitos → ESC04. Solicitar eliminar medicamento de receta electrónica

RQ 17. La aplicación deberá permitir al usuario seleccionar el medicamento que desea eliminar de la receta electrónica de un paciente.

Requisitos → ESC05. Solicitar ver receta electrónica

RQ 18. La aplicación deberá permitir al usuario visualizar los medicamentos que posee un paciente en su receta electrónica.

Requisitos → ESC06. Solicitar dar de alta un paciente

RQ 19. La aplicación deberá permitir añadir un nuevo paciente al sistema.

RQ 20. La aplicación deberá solicitar al usuario la siguiente información del nuevo paciente: identificador (número de la tarjeta de la seguridad social), nombre y apellidos, edad, peso, altura y alergias.

Requisitos → ESC07. Solicitar historia paciente

RQ 21. La aplicación deberá permitir visualizar toda la información existente asociada a un paciente.



Requisitos → ESC08. Solicitar dar de alta personal sanitario

RQ 22. La aplicación deberá permitir añadir un nuevo usuario al sistema.

RQ 23. Esta funcionalidad deberá estar restringida al administrador, que será el encargado de darles de alta.

RQ 24. El administrador deberá introducir la siguiente información del nuevo usuario: Nombre y apellidos, DNI, número de teléfono, correo electrónico y contraseña.

Requisitos → ESC09. Solicitar dar de baja personal sanitario

RQ 25. Esta funcionalidad deberá estar restringida al administrador, que será el encargado de darles de baja.

RQ 26. La aplicación deberá permitir eliminar un usuario del sistema.

Requisitos → ESC10. Solicitar cambio de contraseña

RQ 27. Esta funcionalidad deberá estar restringida al administrador, que será el encargado de cambiarles las contraseña.

RQ 28. La aplicación deberá permitir eliminar un usuario del sistema.

RQ 29. La aplicación deberá enviar un correo electrónico de confirmación en el momento en el que el administrador haya realizado el cambio de la contraseña de un usuario.

RQ 30. La nueva contraseña, no podrá ser una cadena vacía y deberá contener mínimo 8 caracteres, incluyendo mínimo una mayúscula, una minúscula, un número y un símbolo.

RQ 31. La nueva contraseña, no podrá ser nunca igual a la contraseña actual ni a ninguna contraseña anterior que tuviera el usuario.

#### **4.2.3. Funcionalidad App Servidor**

Requisitos → ESC01. Verificar usuario

RQ 32. La AppServidor deberá de responder a la solicitud enviada por la AppCliente, a la hora de permitir el inicio de sesión de un usuario, comprobando que los datos introducidos por el usuario existen en su base de datos y, por lo tanto, permitiendo su acceso a la app.



RQ 33. En el caso del administrador la AppServidor deberá de reconocer, en base a sus credenciales, de que se trata de un usuario privilegiado, para poder concederle los permisos adecuados.

Requisitos → ESC02. Dar de alta personal sanitario

RQ 34. La AppServidor deberá de responder a las solicitudes de dar de alta un usuario, realizando la operación CRUD correspondiente y, por lo tanto, añadiendo a ese nuevo usuario a su BD.

Requisitos → ESC03. Dar de alta paciente en la DB

RQ 35. La AppServidor deberá de responder a las solicitudes de dar de alta un paciente, realizando la operación CRUD correspondiente y, por lo tanto, añadiendo a ese nuevo paciente a su BD.

Requisitos → ESC04. Dar de baja personal sanitario

RQ 36. La AppServidor deberá de responder a las solicitudes de dar de baja un usuario, realizando la operación CRUD correspondiente y, por lo tanto, eliminando a ese usuario de su BD.

Requisitos → ESC05. Modificar contraseña

RQ 37. La AppServidor deberá de responder a las solicitudes de modificar la contraseña de un usuario, realizando la operación CRUD correspondiente y, por lo tanto, modificando el campo asociado a la contraseña de un usuario.

Requisitos → ESC06. Obtener información paciente

RQ 38. La AppServidor deberá de responder a las solicitudes de modificar la contraseña de un usuario, realizando la operación CRUD correspondiente y, por lo tanto, devolviendo la información de un paciente almacenada en la BD.

Requisitos → ESC07. Registrar diagnóstico

RQ 39. La AppServidor deberá de responder a las solicitudes de registrar diagnóstico, realizando la operación CRUD correspondiente y, por lo tanto, añadiendo dicha información en la BD del sistema.



Requisitos → ESC08. Añadir medicamento a receta electrónica

RQ 40. La AppServidor deberá de responder a las solicitudes de añadir medicamento a receta electrónica, realizando la operación CRUD correspondiente y, por lo tanto, añadiendo dicha información en la BD del sistema.

Requisitos → ESC09. Eliminar medicamento de receta electrónica

RQ 41. La AppServidor deberá de responder a las solicitudes de eliminar medicamento de receta electrónica, realizando la operación CRUD correspondiente y, por lo tanto, eliminando dicho medicamento de la receta electrónica almacenada en la BD del sistema.

Requisitos → ESC10. Obtener receta electrónica

RQ 42. La AppServidor deberá de responder a las solicitudes de obtener receta electrónica, realizando la operación CRUD correspondiente y, por lo tanto, devolviendo dicha información de la BD del sistema.

### **4.3. Requisitos de prestaciones**

La enumeración de los siguientes requisitos no funcionales tiene que ver con la carga que se espera que tenga que soportar el sistema, como pueden ser número de usuarios simultáneos, número de terminales, número de transacciones por segundo...

RQs asociados:

RQ 43. La aplicación deberá dar soporte a todos los usuarios que quieran acceder al sistema.

RQ 44. Todas las páginas deberán de tener un tiempo mínimo de respuesta igual o inferior a 0.1 segundos.

RQ 45. La aplicación deberá de poder realizar con éxito, al menos 20 millones de solicitudes por segundo a su base de datos.



#### **4.4. Requisitos de persistencia**

Los requisitos asociados a este apartado serán incorporados en próximas fases de desarrollo.

#### **4.5. Restricciones de diseño**

RQs asociados:

RQ 46. El aspecto de la aplicación debe ser consistente en todas sus páginas, además de amigable e intuitivo hacia el usuario.

#### **4.6. Atributos del sistema**

##### **4.6.1. Fiabilidad**

RQs asociados:

RQ 47. Toda la información otorgada por los usuarios se manipulará únicamente con fines corporativos y de manera limpia.

RQ 48. El tiempo medio entre fallos superará las 150 horas.

##### **4.6.2. Disponibilidad**

RQs asociados:

RQ 49. La aplicación debe estar disponible en todo momento.

##### **4.6.3. Seguridad**

RQs asociados:

RQ 50. El protocolo o librería usado para manejar la seguridad en la sesión del usuario debe ser lo suficientemente confiable.

La información sensible, como contraseñas debe manipular bajo algún nivel de encriptación o cifrado.



RQ 51. Por el diseño de la aplicación, el único que tendrá acceso a la información de los usuarios almacenada en la base de datos, será el administrador, evitando así cualquier problema de privacidad.

#### **4.6.4. Mantenibilidad**

RQs asociados:

RQ 52. El sistema debe ser capaz de manejar toda la información recolectada a través del tiempo con fiidez.

RQ 53. La aplicación deberá de contar con soporte técnico que permita solventar cualquier incidencia producida en el sistema.

#### **4.6.5. Portabilidad**

RQs asociados:

RQ 54. En la elaboración del sistema, se ha tenido en cuenta las distintas plataformas utilizables y se ha intentado no utilizar nada concreto de ningún sistema para conseguir una mayor compatibilidad con todas las plataformas posibles.