



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

GRADO EN INGENIERÍA INFORMÁTICA

**TECNOLOGÍA ESPECÍFICA DE
TECNOLOGÍAS DELA INFORMACIÓN**

TRABAJO FIN DE GRADO

**ONCOSUP: Sistema de Gestión de Datos de
Pacientes Supervivientes a Largo Plazo de un
Cáncer Infantil**

María López Carrasco

Julio, 2018

**ONCOSUP: SISTEMA DE GESTIÓN DE DATOS DE PACIENTES
SUPERVIVIENTES A LARGO PLAZO DE UN CÁNCER INFANTIL**



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

Tecnologías y Sistemas de Información

**TECNOLOGÍA ESPECÍFICA DE
TECNOLOGÍAS DELA INFORMACIÓN**

TRABAJO FIN DE GRADO

**ONCOSUP: Sistema de Gestión de Datos de
Pacientes Supervivientes a Largo Plazo de un
Cáncer Infantil**

Autor: María López Carrasco

Director: Macario Polo Usaola

Director: José Felipe Lozano Gijón

Julio, 2018

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

Resumen

Las mejoras de la medicina en las últimas décadas son muchísimas y la cantidad de enfermedades a las que, poco a poco, se les ha ido ganando terreno, se incrementa casi a diario.

El cáncer es una de esas enfermedades contra las que se está luchando y cada día supone un avance en su tratamiento. La Doctora Blanca Herrero comenzó a registrar todos los datos relacionados con la enfermedad de sus pacientes, que habían sobrevivido a un cáncer infantil, para así, poder realizar un seguimiento de su avance y estudio con el fin de poder detectar sus necesidades y problemas de forma precoz.

El registro de estos datos, que se hace manualmente, supone una tarea lenta y tediosa. La Fundación Ramón Areces en colaboración con la Fundación Oncohematología Infantil, deciden regalar la aplicación ONCOSUP al Hospital Infantil Universitario Niño Jesús y permitir así a la Dra. Herrero registrar los datos de los pacientes al mismo tiempo que pasan la consulta y exportarlos después fácilmente, agilizando su trabajo enormemente.

Este trabajo documenta la unión de la autora a un equipo de trabajo en Informática El Corte Inglés, con el fin de desarrollar la aplicación ONCOSUP, usando tecnologías punteras para tratar de desarrollar una aplicación web completa en un tiempo récord.

El resultado de este trabajo supone un cambio en el estudio oncológico del cáncer infantil, y sienta las bases para la futura adición de funcionalidades de ONCOSUP o su aplicación a otros hospitales, permitiendo así, que estudios similares puedan comenzar.

Abstract

Improvements in medicine in last decades are many and the amount of deseases which, step by step, we have been gaining ground, increases almost daily.

Cancer is one of those deseases that are still being fought and each year is an advance in its treatment. Doctor Blanca Herrero started to register all data related to the desease of her patients, who had survived childhood cancer, in order to be able to track the progress and study their needs as early as possible.

The registration of these patients is a slow and tedious task Ramón Areces Foundation in colaboration with Child Oncohematology Fundation, have decided to give the ONCOSUP application to Hospital Infantil Universitario Niño Jesús as a gift and allow Dr. Herrero to register her patients data at the same time she pass consultation and export them easily, making her work a faster task.

This work documents the union of the author to a work team at Infomática El Corte Inglés, in order to develop the functionalities of ONCOSUP, using new technologies to develop an application in a record time.

The result of this work ir a change in the oncological study of childhood cancer, and lays the foundations for the future addition of functionalities or its extension to other hospitals, allowing similar studies to begin.

Agradecimientos

María

A mis padres, que cada vez que tropicé, estuvieron ahí para ayudar a levantarme.

Índice general

Resumen	III
Abstract	V
Agradecimientos	VII
Índice general	XI
Índice de cuadros	XV
Índice de figuras	XVII
Índice de listados	XIX
1. Introducción	1
1.1. Estructura del documento	2
2. Objetivos	3
2.1. Objetivo general	3
2.2. Objetivos específicos	3
2.2.1. Objetivo 1: Estudio Preconsulta	3
2.2.2. Objetivo 2: Consulta de Seguimiento	4
2.2.3. Objetivo 3: Informes	4
2.2.4. Objetivo 4: Administración de Usuarios y Administración de Protocolos	4
2.2.5. Objetivo 5: Exportación de datos	4
2.2.6. Objetivos personales	4
3. Antecedentes	7
3.1. Comunicación en aplicaciones Web	7
3.2. JHipster	8
3.3. Angular 5	11

0. ÍNDICE GENERAL

3.4. Otras tecnologías	12
4. Método de trabajo	13
4.1. SCRUM	13
4.2. Roles	13
4.2.1. Product Owner	13
4.2.2. Equipo de desarrollo	14
4.2.3. Scrum Master	14
4.3. Flujo de trabajo o Workflow	14
4.3.1. Sprints	15
4.3.2. Sprint planning	15
4.3.3. Daily scrum	15
4.3.4. Sprint review	16
4.3.5. Retrospectiva	16
4.3.6. Otros flujos	16
4.4. Artefactos	17
4.4.1. Product Backlog	17
4.4.2. Sprint Backlog	17
4.4.3. Incremento del producto	18
4.4.4. Otros artefactos	18
4.5. Valores de Scrum	18
4.6. Planificación de ONCOSUP	19
5. Resultados	21
5.1. Inception	21
5.1.1. Visión.	21
5.1.2. Comunidad de Proyecto.	22
5.1.3. Restricciones y riesgos.	23
5.1.4. Story Map.	26
5.1.5. Volcado en JIRA y Confluence.	27
5.2. Incorporación a IECISA	29
5.3. Sprint 0	31
5.4. Sprint 1	32
5.4.1. Sprint Planning	32
5.4.2. Desarrollo del sprint	34
5.4.3. Sprint Review	41

5.4.4. Retrospectiva	41
5.5. Sprint 2	43
5.5.1. Sprint Planning	43
5.5.2. Desarrollo del sprint	44
5.5.3. Sprint Review	47
5.5.4. Retrospectiva	47
5.6. Sprint 3	49
5.6.1. Sprint Planning	49
5.6.2. Desarrollo del sprint	49
5.6.3. Sprint Review	52
5.6.4. Retrospectiva	53
5.7. Sprint 4	54
5.7.1. Sprint Planning	54
5.7.2. Desarrollo del sprint	56
5.7.3. Sprint Review	58
5.7.4. Retrospectiva	59
5.8. Sprint 5	60
5.8.1. Sprint Planning	60
5.8.2. Desarrollo del sprint	61
5.8.3. Sprint Review	64
5.8.4. Retrospectiva	64
5.9. Sprint 6	66
5.9.1. Sprint Planning	66
5.9.2. Desarrollo del sprint	67
5.9.3. Sprint Review	71
5.9.4. Retrospectiva	71
6. Conclusiones	73
6.1. SCRUM	74
6.2. JHipster	74
6.3. Objetivos cumplidos	76
6.4. Objetivos personales	77
6.5. Un estudio de valor añadido: Valoración de SCRUM	77
A. Estructura de la aplicación final	87
A.1. Estructura de entidades de ONCOSUP	87

0. ÍNDICE GENERAL	
A.2. JDL de ONCOSUP	88
B. Generación de una aplicación con JHipster	107
Referencias	115

Índice de cuadros

2.1. Competencias de Tecnologías de la Información	5
4.1. Planificación para ONCOSUP	19
5.1. Detalle de la comunidad de proyecto	23
5.2. Historias planeadas Sprint 1	33
5.3. Historias planeadas Sprint 2	43
5.4. Historias planeadas Sprint 3	49
5.5. Historias planeadas Sprint 4	55
5.6. Historias planeadas Sprint 5	61
5.7. Historias planeadas Sprint 6	67

Índice de figuras

3.1. Esquema de comunicación HTTP	8
3.2. Ejemplo de JDL	8
3.3. Cliente para el ejemplo de Jefes y Proyectos	9
3.4. Servidor para el ejemplo de Jefes y Proyectos	10
3.5. Base de Datos para el ejemplo de Jefes y Proyectos	11
3.6. Proyecto Angular	11
3.7. Otras tecnologías que se usarán para el desarrollo del proyecto	12
5.1. Resultado de la fase de visión	22
5.2. Comunidad de proyecto	23
5.3. Restricciones y Trade offs	24
5.4. Matriz de riesgos	25
5.5. Ejemplo de una Historia de Usuario	26
5.6. Product Backlog	27
5.7. Índice de la documentación del proyecto	28
5.8. Matriz de riesgos en Confluence	29
5.9. Arquitectura de la aplicación	31
5.10. HU002 - Alta enfermedad	34
5.11. Detalle de un paciente	36
5.12. Crear o editar un paciente	37
5.13. Predicción del avance del proyecto	48
5.14. Modulo de Auditoría	51
5.15. Listado de pacientes una vez eliminados los campos innecesarios	53
5.16. Previsión del avance del proyecto	54
5.17. Creación de una incidencia en JIRA	57
5.18. Burn Down Chart del Sprint 4	58
5.19. Predicción del avance del proyecto	59
5.20. Previsión de sprint restantes	60

0. ÍNDICE DE FIGURAS

5.21. Diálogo de un ítem sin maquetar	63
5.22. Diálogo de un ítem maquetado	64
5.23. Diálogo de un item maquetado al desplegar un grupo de campos	65
5.24. Burn Down Chart. Sprint 5	65
5.25. Previsión de sprint restantes	66
5.26. Previsualización del Informe	69
5.27. Vista de impresión del informe	70
6.1. Recuento de líneas de código al generar el proyecto	75
6.2. Recuento de líneas de código al finalizar el proyecto	75
6.3. Resultados del equipo en la primera parte del cuestionario	79
6.4. Resultados del equipo en la segunda parte del cuestionario	80
6.5. Resultados de la Product Owner en la primera parte del cuestionario	81
6.6. Resultados de la Product Owner en la segunda parte del cuestionario	82
6.7. Comparativa de los resultados del primer y el sexto sprint	83
A.1. Enlace a la imagen con la estructura de las entidades de ONCOSUP	87
B.1. Creación de la carpeta para el proyecto	107
B.2. Inicio de JHipster	108
B.3. Selección del tipo de aplicación	108
B.4. Elección del tipo de autenticación	109
B.5. Configuración del tipo de Base de Datos	109
B.6. Selección acerca de la abstracción de caché	109
B.7. Versiones de Angular a elegir	110
B.8. Configuración del idioma principal y secundarios	110
B.9. Herramientas de testing disponibles	110
B.10. Resumen de la configuración para la aplicación	111
B.11. Resultado de la generación	111
B.12. Estructura de entidades usada para la aplicación	112
B.13. Importación del JDL	112
B.14. Resultado de la generación de entidades	113
B.15. Página de inicio de la aplicación	113

Índice de listados

5.1.	Asociación de una hoja de estilos a un componente	34
5.2.	Estilos generales para páginas de detalle	35
5.3.	Botón añadido al detalle del paciente	36
5.4.	Código dónde se definen las rutas para el diálogo del diagnóstico	38
5.5.	Paso del <i>id</i> al pop-up del diagnóstico	38
5.6.	Archivo .model del diagnóstico	39
5.7.	Código del pop-up del diagnóstico	39
5.8.	Código añadido al DTO de diagnóstico	40
5.9.	Asignación de un diagnóstico al paciente	40
5.10.	ComboBox dependientes	45
5.11.	Implementación onComunidadActual(paciente.comunidadActual)	45
5.12.	Cambios necesarios en el DTO para mostrar el diagnóstico específico	46
5.13.	Línea añadida al PacienteMapper.java	46
5.14.	adición de un span con el diagnóstico específico	46
5.15.	Cambios en ServiceImpl para añadir auditoría	50
5.16.	Cambios en Resource para añadir auditoría	51
5.17.	Cambios en DiagnósticoServiceImpl para almacenar el año del diagnóstico en el paciente	52
5.18.	Añadir nuevas entidades y sus relaciones en el JDL	56
5.19.	Control de campos que se muestran según el <i>check box</i> marcado	62
5.20.	Implementación del método print()	68
5.21.	Estilos específicos de la impresión	68
A.1.	JDL correspondiente a todas las entidades de la aplicación	88

Capítulo 1

Introducción

UN paciente de cáncer es considerado superviviente a largo plazo cuando han pasado cinco años desde su diagnóstico y tratamiento. Los distintos avances en medicina han generado un aumento en los supervivientes de cáncer.

Es habitual que los supervivientes a largo plazo desarrollen algún tipo de afección tardía relacionada con el tratamiento oncológico, que puede, en ocasiones, requerir tratamiento. De esta forma, el servicio de Oncología Pediátrica del Hospital Infantil Universitario Niño Jesús, en colaboración con la Fundación Ramón Areces y la Fundación Oncohematología Infantil firman un acuerdo de colaboración para la creación del primer registro de supervivientes a largo plazo de cáncer infantil en España [not18]. El registro de pacientes supervivientes surge con el objetivo de poder realizar un seguimiento de los pacientes, con el fin de poder tener en cuenta su enfermedad oncológica y las posibles complicaciones.

La encargada de la construcción de este registro es la Doctora Blanca Herrero, que tras un tiempo trabajando en él, detectó la necesidad de agilizar el trabajo disponiendo de un sistema informático específico que permitiera la adquisición y procesamiento de los datos.

Hasta el momento, la recopilación de la información se llevaba a cabo manualmente, en papeles manuscritos, de los que posteriormente se pasaba la información a una hoja de cálculo. Todo este trabajo resulta lento y ralentiza el registro de los datos y, en consecuencia, retrasa la obtención de datos en beneficio de los pacientes.

De esta forma se plantea a Informática El Corte Inglés, en adelante IECISA, desarrollar un sistema informático con los requisitos deseados por el servicio de Oncología Pediátrica. Este trabajo es el resultado de la participación de la autora en el desarrollo de este proyecto.

Así pues, en este documento se describe ONCOSUP, una aplicación web para el manejo de datos de pacientes supervivientes a largo plazo de cáncer infantil, que permitirá entregar al paciente un informe con toda la información referente a su seguimiento y recomendaciones para su tratamiento, además de permitir la exportación sencilla de datos para su posterior estudio con una herramienta de análisis estadístico.

1. INTRODUCCIÓN

Estructura del documento

El presente documento sigue la estructura expuesta a continuación:

Capítulo 2: Objetivos

Se definen los objetivos del proyecto.

Capítulo 3: Antecedentes

Se explica el contexto en el que surge la necesidad de ONCOSUP y se explican las diferentes herramientas que se usan en su desarrollo.

Capítulo 4: Método de trabajo

Desarrolla el método de trabajo seguido durante el desarrollo del proyecto.

Capítulo 5: Resultados

Expone todo el desarrollo del proyecto, haciendo hincapié en el trabajo realizado por la autora.

Capítulo 6: Conclusiones

Reflexión de la autora de lo acontecido durante el desarrollo del proyecto, tanto del trabajo en IECISA, como de la experiencia del FORTE.

Capítulo 2

Objetivos

El capítulo se centra en exponer cuáles son los objetivos del proyecto, desde el objetivo principal y más general que se pretende que cumpla la aplicación, pasando por los 5 objetivos principales que componen la totalidad del proyecto. Se hará también una breve exposición de cuáles son los entornos de los que se dispone para la realización de la aplicación.

Cada uno de los objetivos planteados para el proyecto supone un incremento funcional que, al ser completado, aportará valor al cliente.

Objetivo general

El objetivo principal de ONCOSUP es el desarrollo de una aplicación web para el Hospital Infantil Universitario Niño Jesús, que permitirá registrar datos sobre pacientes que han sobrevivido a un cáncer infantil. Dicho registro servirá para que los datos puedan ser exportados a una hoja de cálculo, permitiendo así su importación a una herramienta estadística con la que poder realizar estudios sobre los datos registrados y permitirá generar un informe para cada paciente que podrá entregar a su médico de cabecera con el fin de informarles sobre todos los tratamientos y complicaciones que ha llevado el paciente por la enfermedad.

Objetivos específicos

Esta sección expondrá, en orden de prioridad, los cinco objetivos que debe cumplir la aplicación, que de completarse, supondrán la realización íntegra del proyecto.

Objetivo 1: Estudio Preconsulta

El primer objetivo a completar es permitir el registro de todos los datos relacionados con el estudio previo a una consulta, esto es, todos los datos que deben existir antes de que el paciente vaya a la consulta de la doctora. Por tanto, lo principal es permitir el registro del paciente: nombre y apellidos, datos de contacto, de residencia, etcétera, además para cada paciente registrado, se debe poder asociar un diagnóstico, un tratamiento e indicar si ha tenido recaídas y la información acerca de éstas.

Los fármacos y protocolos deben existir previamente en la aplicación, lo que implica que, a

2. OBJETIVOS

pesar de pertenecer a subtareas del objetivo 4, de menor prioridad, será obligatorio desarrollar parte de ese objetivo para que el primero esté completo. Por tanto, como se verá en el capítulo 5 algunas de las tareas del cuarto objetivo se planificarán para los primeros sprints.

Objetivo 2: Consulta de Seguimiento

El segundo objetivo es quizás el más amplio de los cinco que se plantean. Se trata de la implementación de todo aquello relacionado con la consulta, permitiendo a la doctora poder registrar, mientras pasa consulta a sus pacientes, todos los cambios y avances de su enfermedad.

Debido a que la cantidad de información que se debe poder almacenar durante la consulta es enorme, además de la complejidad de algunas de las tareas necesarias para completar el objetivo, el objetivo 2 es el más importante y prácticamente el corazón de la aplicación. Sin acabarlo, no tendría sentido terminar los siguientes, ya que es imprescindible para poder exportar después toda la información.

Objetivo 3: Informes

Todo aquello referente a la creación e impresión de informes. La doctora debe poder acceder a una vista previa del informe que podrá modificar según las necesidades del paciente. Una vez esté conforme con los datos del informe, podrá imprimirla fácilmente.

Objetivo 4: Administración de Usuarios y Administración de Protocolos

En la administración se pueden diferenciar dos partes, la primera relacionada con la administración de usuarios y auditoría de la aplicación y la segunda con la administración de los datos médicos de la aplicación.

La administración de datos médicos abarca toda la administración de datos sobre protocolos y recomendaciones estándar, que podrán ser seleccionadas, si es necesario, para un paciente y es necesario completarla para terminar el objetivo 1.

Objetivo 5: Exportación de datos

Incluye las tareas relacionadas con la selección de información y su exportación final. Deberá existir un filtro que permita seleccionar qué datos se desea exportar.

Objetivos personales

Se exponen, a continuación, otros objetivos independientes de los específicos del proyecto.

Cumplir con las competencias de la intensificación

En el anteproyecto de este documento se justificó qué competencias de la intensificación de Tecnologías de la Información cubriría la participación de la autora en ONCOSUP. Las

competencias y su justificación son las que muestra el cuadro 2.1.

Cuadro 2.1: Competencias de Tecnologías de la Información

Competencia	Justificación
Capacidad para comprender el entorno de una organización y sus necesidades en el ámbito de las tecnologías de la información y las comunicaciones.	Al desarrollarse el TFG en el marco de un FORTE y en las instalaciones de IECISA, la integración de la autora con la empresa (organización, métodos de trabajo, etcétera) será completa. Igualmente, el TFG se llevará a cabo utilizando SCRUM con una gran iteración con el cliente, lo que permitirá conocer de primera mano sus necesidades reales.
Capacidad para seleccionar, diseñar, desplegar, integrar, evaluar, construir, gestionar, explotar y mantener las tecnologías de hardware, software y redes, dentro de los parámetros de coste y calidad adecuados.	El proyecto está acotado en cuanto a tiempo y coste, estando el alcance bien delineado desde la fase de <i>Inception</i> . Por tanto, las variaciones en éstas u otras variables han de estar bien controladas para evitar sobrecostes y retrasos
Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas.	Se aplicará SCRUM como marco de trabajo, que, a través de una serie de roles, eventos, artefactos y unas reglas que los relacionan, permite la gestión de proyectos donde la comunicación y la cooperación son base fundamental.
Capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse. Capacidad para seleccionar, desplegar, integrar y gestionar sistemas de información que satisfagan las necesidades de la organización, con los criterios de coste y calidad identificados.	Al aplicar SCRUM se definen unos riesgos que se clasifican según su impacto potencial. Se definen, además, planes de contingencia para los riesgos más significativos, existiendo también una política de seguimiento y monitorización del impacto medio.
Capacidad de concebir sistemas, aplicaciones y servicios basados en tecnologías de red, incluyendo: Internet, web, comercio electrónico, multimedia, servicios interactivos y computación móvil.	El proyecto se implementará en la forma de una aplicación web accesible desde la intranet del hospital.

2. OBJETIVOS

Capacidad para comprender, aplicar y gestionar la garantía y seguridad de los sistemas informáticos.

El proyecto gestionará información médica absolutamente confidencial, por lo que se establecerán todos los mecanismos necesarios para garantizar la privacidad y la seguridad en el acceso a la información.

Uniéndose a un equipo de trabajo en IECISA, se pretende que la autora cubra las competencias expuestas más arriba.

Ampliar conocimientos

El último objetivo es que la autora tenga mayor conocimiento tanto, de las herramientas nuevas que vaya a usar durante el desarrollo de la aplicación, como el de las que ya conocía antes de entrar en el FORTE. En el capítulo 6 se expondrán los avances en este área comparando los conocimientos antes de comenzar el proyecto con los conocimientos al acabarlo.

Capítulo 3

Antecedentes

Tal y como se ha comentado en capítulos anteriores, este proyecto trata de automatizar la adquisición de datos para su posterior exportación a hojas de cálculo que poder importar a herramientas estadísticas, además de permitir generar informes para los pacientes con toda la información relevante que el doctor considere.

Hasta la puesta en marcha de ONCOSUP, todas las tareas se realizaban de forma manual, requiriendo que la Dra. Herrero introdujera la información recopilada en la consulta ella misma en una hoja de cálculo. Esta hoja de cálculo estaba compuesta por más de 200 campos, que debían ser comprobados y rellenados con la información pertinente. Esta tarea no sólo era lenta, sino que además propiciaba el error humano al tener que manejar tal cantidad de datos manualmente.

El proyecto nace con importantes restricciones de plazo y presupuesto. Además, se da la circunstancia de que IECISA quería probar el desempeño de algunos frameworks de desarrollo rápido de aplicaciones web como JHipster[jhi18]. Por tanto, esta iniciativa unida a las restricciones del proyecto desembocó en la elección de JHipster como herramienta para el desarrollo de la aplicación.

La planificación y gestión del proyecto se hará según el marco de trabajo SCRUM, que está muy implantado en el Centro de Ingeniería del Software (CIS) en el “Espacio Calatrava”, centro de IECISA en Miguelturra.

A continuación, se hará un repaso por las técnicas y tecnologías más relevantes empleadas en el desarrollo de ONCOSUP.

Comunicación en aplicaciones Web

La forma más habitual de comunicación en aplicaciones Web es a través del protocolo HTTP (Hypertext Transfer Protocol). Funciona siguiendo una estructura *request-response* (petición-respuesta), en el que un usuario establece comunicación con el cliente a través de un *User Agent* desde el que se enviarán las peticiones y al que llegarán las respuestas. El ejemplo más típico de ésto son los navegadores web (*User Agent*) con los que el usuario se establece comunicación con los sitios web (servidores) solicitando acceso a un sitio web, y el servidor le devolverá el código de la página.

3. ANTECEDENTES

Además este protocolo permite que cliente y servidor sean independientes de la tecnología del otro, ya que solo define cómo debe ser la comunicación, pero no hay restricciones respecto a las tecnologías a usar. La figura 3.1 muestra un esquema de la comunicación por HTTP.

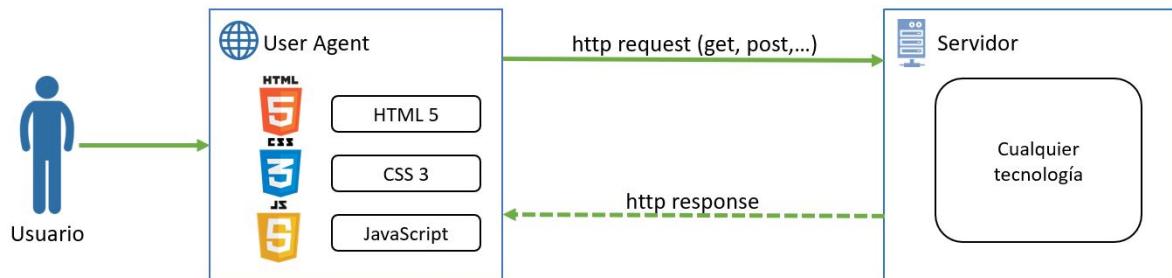


Figura 3.1: Esquema de comunicación HTTP

Otra forma habitual de comunicación entre cliente y servidor son los *websockets* [web18]; no obstante, se omite su descripción ya que no se han utilizado para el desarrollo del proyecto.

JHipster

JHipster se trata de un framework de desarrollo rápido de aplicaciones web que, a partir de un fichero en el que se describe la estructura de datos del sistema, el JDL (JHipster Domain Language) [jdl18b], un motor de generación de código produce el código del cliente, el del servidor y el de la base de datos, generando una aplicación completamente funcional.

La figura 3.2 muestra un ejemplo sencillo de un código JDL que describe una aplicación en la que un proyecto tiene un jefe de proyecto, visto en JDL-Studio [jdl18a]. Esta herramienta permite escribir el JDL, mostrando errores que pueda contener el código y generando en tiempo real un diagrama con la estructura de la aplicación.

En el código le indicamos las entidades que habrá en la aplicación y sus atributos, las relaciones que hay entre ellos y además se le puede indicar si se desea que las entidades tengan paginación o si queremos o no que se generen los DTO (Data Transfer Object) o los *ServiceImpl*,

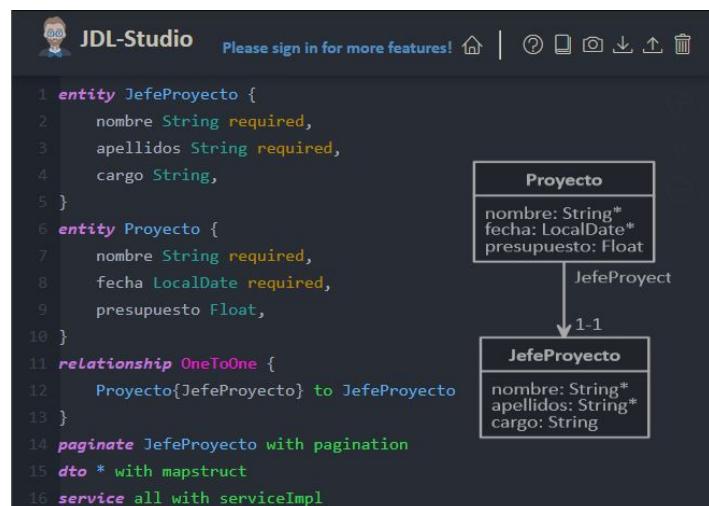


Figura 3.2: Ejemplo de JDL

dónde está el negocio de la aplicación.

JDL-Studio ofrece además la posibilidad de gestionar los diferentes JDL con los que haremos trabajado guardándolos en nuestra cuenta.

Por otro lado, permite exportar el código que hemos escrito o obtener una imagen del diagrama generado que se podrá usar para la documentación del proyecto, por ejemplo, como es el caso de este trabajo A.

Toda la documentación sobre el lenguaje está disponible en la misma herramienta de edición, siendo posible tener en la misma pantalla nuestro código, el diagrama de éste y la documentación para consultarla mientras escribimos.

A partir del código del JDL, JHipster genera:

- Para la parte del cliente:

Todos los archivos que necesita Angular 5 (sección 3.3). Esto supone que generará el cliente siguiendo el patrón MVC (Modelo Vista Controlador) dentro del cliente que permitirá tratar los datos de manera similar a como lo haría una aplicación de escritorio. Se generarán para el Jefe y el Proyecto todos los HTML necesarios, con su detalle y diálogos, además del CSS necesario para su maquetación.

Además, se generan diferentes archivos en *TypeScript* para los diferentes componentes y sus servicios, que serán los encargados de la comunicación con el lado servidor.

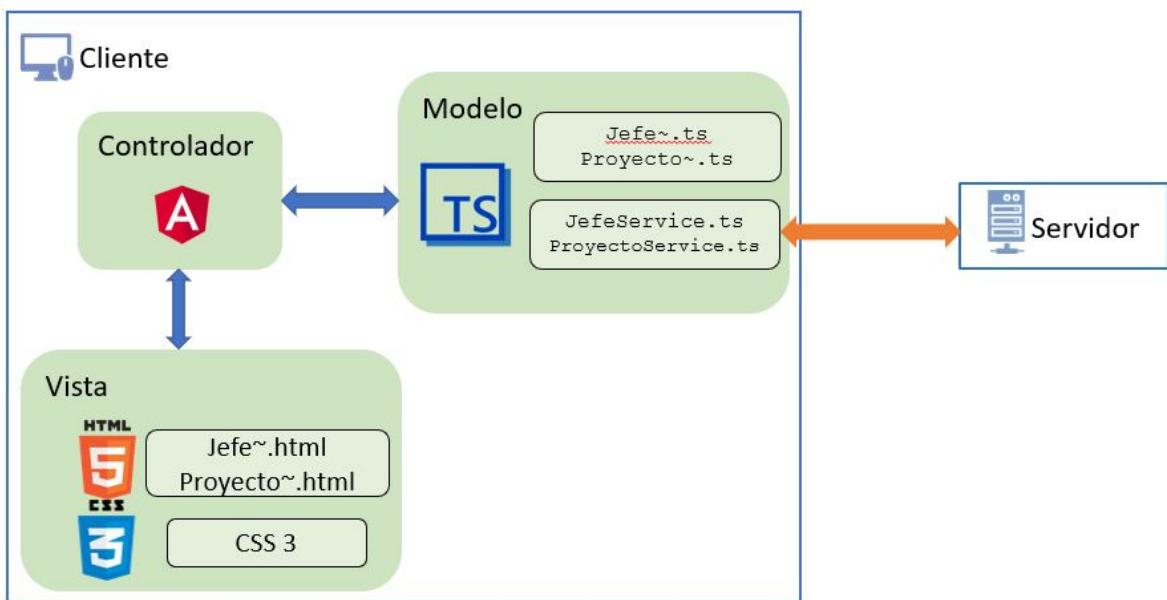


Figura 3.3: Cliente para el ejemplo de Jefes y Proyectos

3. ANTECEDENTES

- Para el servidor:

Genera un MVC, los Resource que se encargan de la comunicación con el cliente, los Services en los que estará el negocio de la aplicación, y los repository que serán los que realicen operaciones sobre la base de datos.

Los DTO se encargan de transmitir información entre *Services* y *Resources*, y las clases para cada entidad lo harán con los *Services* y los *Repository*.

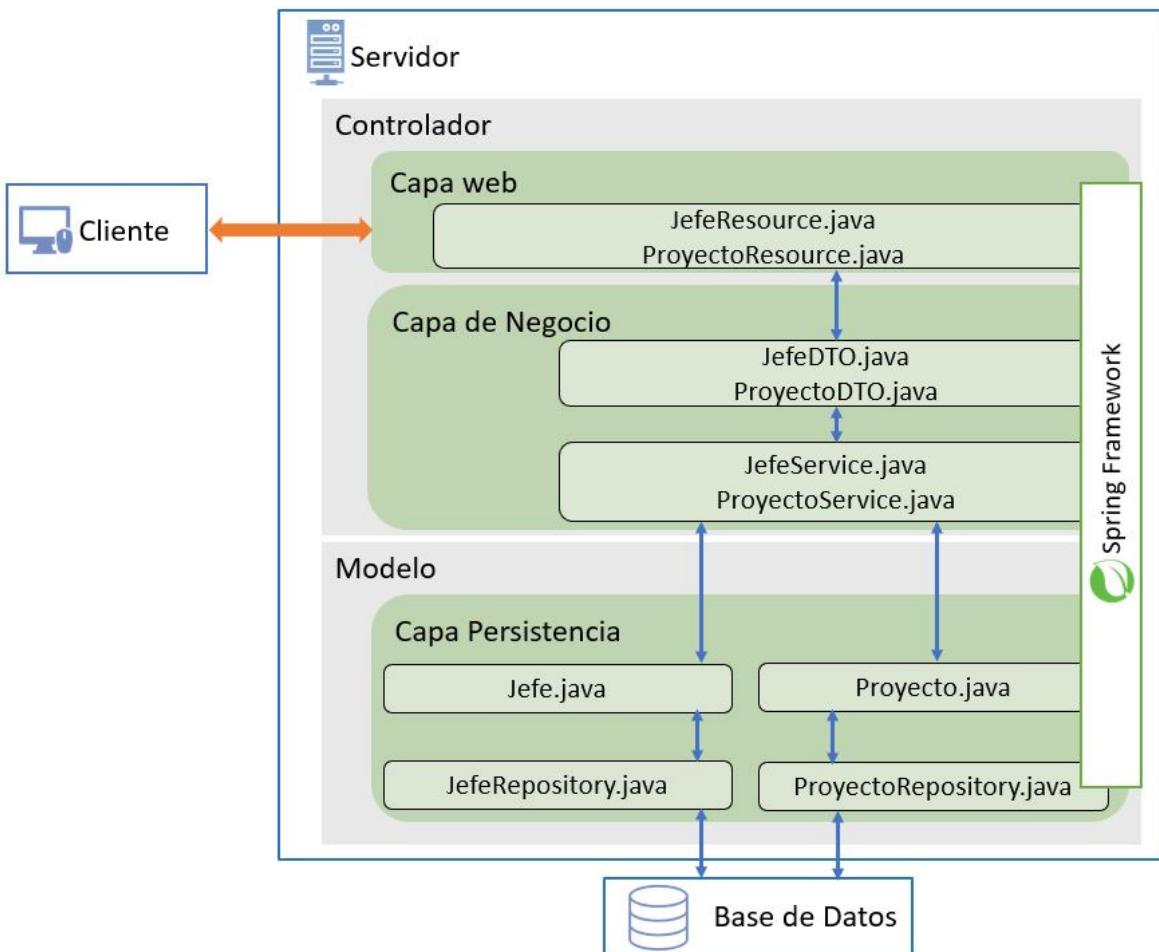


Figura 3.4: Servidor para el ejemplo de Jefes y Proyectos

- Finalmente, para la base de datos:

Cada clase es anotada con etiquetas como `@Table` o `@Column`, que servirá para generar un archivo `xml` para cada entidad con su estructura en base de datos y otro con las relaciones entre entidades. Después, *Liquibase*, que es una librería de gestión y aplicación de cambios en bases de datos [liq18], se encarga de traducirlo al formato soportado por nuestra base de datos y crear o añadir los cambios en ella.

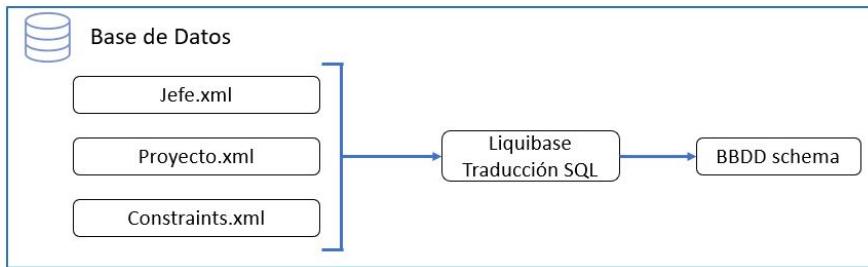


Figura 3.5: Base de Datos para el ejemplo de Jefes y Proyectos

Angular 5

Angular (Figura 3.6) se trata de un framework de desarrollo de aplicaciones web desarrollado por Google [ang18], la versión 5 será la usada en ONCOSUP. El lenguaje usado es *TypeScript*, en el que se generan los diferentes componentes, servicios o enrutadores. Angular trata de facilitar el desarrollo de aplicaciones web implementando un MVC en el front-end, que además facilita las pruebas.

Angular es una tecnología con un despegue relativamente nuevo, pero su primera versión, AngularJS, salió a la luz en el año 2010. Esta versión estaba limitada al uso de JavaScript, y su uso se fue popularizándose en el 2012 hasta que apareciera una nueva versión del framework. Angular fue, de alguna forma, una manera de impulsar el cambio de Apache y PHP a JavaScript tanto en el front-end como en el back-end.

En septiembre de 2016 se lanza la nueva versión de AngularJS, en la que pierde el JS y se empiezan a numerar, saliendo a la luz Angular 2. La segunda versión de Angular fue reescrita casi por completo, conservando conceptos de funcionalidad. El paso de AngularJS a Angular 2, supuso un descontento en la comunidad que había estado desarrollando sus aplicaciones con AngularJS, ya que ahora tendrían que volver a empezar sus proyectos o migrarlos de alguna manera a Angular 2. A pesar de esto, el cambio a la nueva versión supuso una mejora, ya que Google pretendía llevar una gestión de versiones que permitiera ir añadiendo funcionalidad, conservando la compatibilidad entre versiones.

Alrededor de seis meses después, en marzo de 2017, el equipo de Google lanza Angular 4, con diferencias no muy significativas respecto a la versión anterior, pero que conservaba retrocompatibilidad con la anterior. Esta nueva versión, al llegar tan rápido, causó revuelo en la comunidad, haciendo pensar a los usuarios del framework que ocurriría lo mismo que con el salto de Angular JS a Angular 2. Afortunadamente no fue así y la popularidad de la



Figura 3.6: Proyecto Angular

3. ANTECEDENTES

herramienta fue aumentando.

En noviembre de ese mismo año se publica Angular 5, versión que se usará para el desarrollo de ONCOSUP, que añadiría mejoras de rendimiento, junto con alguna funcionalidad nueva.

Durante el desarrollo del proyecto, en marzo de 2018, saldrá a la luz la nueva versión, Angular 6. Y se estima que Angular 7, será lanzado entre septiembre y noviembre del mismo año.

Como curiosidad, la versión Angular 3 nunca salió a la luz y se saltó directamente a la versión 4 debido a un desalineamiento con la numeración de las versiones de uno de sus paquetes que iban por su versión 3.

Otras tecnologías

Algunas tecnologías que se usarán en el proyecto, pero en las que no se profundizará tanto, son (figura 3):

- MariaDB [mar18] - Se trata de una evolución de MySQL en la que han estado trabajando algunos de los desarrolladores originales de MySQL.
- SpringBoot [spr18a] - Herramienta de generación de código, que JHipster usa y añade nuevos elementos para crear una aplicación más completa que la que SpringBoot ofrece.
- Spring Security [spr18b] - Toda la seguridad de la aplicación será gestionada por esta herramienta del framework de Spring. Los desarrolladores tan sólo tendrán que hacer pequeñas modificaciones para que se ajuste a las necesidades de la aplicación.
- Apache Tomcat [tom18] - La generación de la aplicación incluye un servidor Tomcat embebido.



Figura 3.7: Otras tecnologías que se usarán para el desarrollo del proyecto

Capítulo 4

Método de trabajo

ESTE capítulo se centra en describir el método de trabajo que se seguirá en el desarrollo del proyecto, sus diferentes eventos y artefactos. Se verá también cuál es la planificación y las distintas fases en función del marco de trabajo elegido.

SCRUM

SCRUM se trata de un marco de trabajo, o *framework*, ágil, iterativo e incremental con especial énfasis en el desarrollo software, su principal fin es la gestión del desarrollo del producto. Este marco de trabajo es el que se practica en IECISA. Es común ver que a veces se refiere a SCRUM como una metodología ágil, pero como bien explica Gunther Verheyen en este post[not13], SCRUM se trata de un marco de trabajo ya que no hay reglas que seguir, sino que se trata de un acercamiento heurístico.

Está pensado para equipos de trabajo y fomenta que se autogestionen y se organicen ellos mismos. Es deseable que los miembros del equipo trabajen en el mismo lugar, pero de no ser posible, se pueden realizar colaboraciones periódicas online. El equipo se reparte el trabajo dividiéndolo en tareas que pueden ser completadas en iteraciones de tiempo limitado a las que se llaman *sprints*.

Además, en el “SCRUM Team” o equipo SCRUM, existen diferentes roles según el papel de cada integrante y su función en el equipo.

Se realizan diferentes tipos de reuniones para llevar un control del seguimiento, ya sean diarias o por *sprint*.

Roles

SCRUM define tres roles principales que idealmente trabajan juntos para entregar un producto potencialmente usable al final de cada *sprint*.

Product Owner

El *Product Owner* o propietario del producto representa a los *stakeholders* o interesados del producto y al cliente. Es el responsable de asegurar que el equipo aporta valor a la empresa, definiendo correctamente el producto en términos del cliente, historias de usuario, que se

4. MÉTODO DE TRABAJO

añadirán al *Product Backlog* y después priorizará en función de su importancia y dependencias con otras historias. Debe asegurarse de que el *Product Backlog* es visible, transparente y claro.

Una responsabilidad principal del *Product Owner* es la comunicación con el equipo y *stakeholders*. Su habilidad para transmitir prioridades y empatizar tanto con los miembros del equipo como con los *stakeholders* es vital para que el desarrollo del producto siga el camino correcto. Debe saber dar la información necesaria a cada participante, ya que dar más de la necesaria puede provocar, por ejemplo, la pérdida de interés en el producto de un *stakeholder*.

Equipo de desarrollo

El equipo de desarrollo es el encargado de entregar incrementos potencialmente usables del producto en cada sprint. Suele estar compuesto de entre tres y nueve miembros que llevan a cabo las tareas requeridas en cada incremento del producto.

Los equipos de desarrollo son autoorganizados, aunque esto no implica que no pueda haber interacción con otros miembros de fuera del equipo con el fin de tomar decisiones y llevar la gestión del desarrollo.

Scrum Master

El *Scrum Master* es responsable de facilitar al equipo el cumplir las metas propuestas para el producto durante su proceso de desarrollo. No se trata de un líder de equipo o encargado de gestión del proyecto, sino que su misión es aconsejar e informar al equipo de los problemas o beneficios de sus acciones para que lo tengan en cuenta a la hora de tomar decisiones.

Su principal actividad es asegurarse de que el equipo sigue el marco de trabajo y asegurarse de que se siguen los procesos que define SCRUM, además de animar al equipo a ser proactivos y mejorar continuamente. Algunas de sus tareas son:

- Es responsable de ayudar al *Product Owner* a mantener el *Product Backlog* para que el trabajo sea comprendido por el equipo y puedan completarlos de forma rápida y sin malentendidos.
- Debe ayudar al equipo a definir el concepto de “completado” (*Definition of Done*) para el producto. En esta tarea se puede tener ayuda de algunos *stakeholders* clave en el producto.
- Aconsejar al equipo con el fin de entregar un producto con la máxima calidad posible.

Flujo de trabajo o Workflow

El marco de trabajo define una serie de eventos y reuniones que sirven para guiar el flujo de trabajo durante el desarrollo del producto.

Sprints

Un sprint es la unidad básica de desarrollo. Es el esfuerzo, hecho por el equipo, en un período de tiempo fijo, normalmente entre dos y cuatro semanas, en el que se completa cierto número de tareas y se entrega un incremento del producto.

Todos los sprint comienzan siempre con un *Sprint Planning*, en el que se define el *Sprint Backlog*, se selecciona el trabajo que se tratará de completar y se establece un objetivo o meta para el sprint.

Al final del sprint se realiza la *Sprint Review* y la sesión de Retrospectiva, dónde se revisa el trabajo completado con el *Product Owner* y los *stakeholders* y se identifican aspectos a mejorar de cara al siguiente sprint.

Sprint planning

Se trata de la primera reunión que se realiza en cada sprint. En el *Sprint Planning* se acuerda cuál será el objetivo del sprint, que se tendrá en cuenta si fuese necesario en la toma de decisiones durante el sprint.

El equipo propone los elementos del *Product Backlog* que, consideran, podrán completar durante el sprint. Los elementos del *Sprint Backlog* se dividen en tareas más pequeñas que simplifican el trabajo necesario para completar una historia de usuario. Si se considera que el esfuerzo requerido por una historia es demasiado alto, en esta reunión es cuando se divide en más historias para que el esfuerzo sea equilibrado.

La duración recomendada de esta reunión es de unas dos horas por cada semana del sprint.

Daily scrum

La “daily” se realiza cada día, una reunión de corta duración que se hace de pie y a la que asisten todos los miembros del equipo. Se realizan de forma puntual, siempre a la misma hora y en el mismo lugar y suelen tener una duración de unos 15 minutos.

De forma general, cada participante debe tratar de responder a tres puntos:

- Qué completó el día anterior que contribuye al objetivo del sprint.
- Qué planea completar hoy y qué aporta al objetivo del sprint.
- Si identificó algún problema. Esto debe ser registrado por el *Scrum Master* en los riesgos del proyecto y asignado a un encargado para su gestión.

Como resultado de esta reunión, todo el equipo es consciente de en qué está trabajando cada uno y puede además ayudar a algún compañero si lo necesita.

4. MÉTODO DE TRABAJO

Sprint review

En la *Sprint Review* se hace un repaso de cuál ha sido el trabajo completado y el que ha quedado por terminar. Se hace un breve estudio, con gráficas que ofrezcan las herramientas de gestión que tengan disponibles, en el que se compara la velocidad obtenida con la de otros sprint, o se ve una predicción de fecha de entrega. Esta información será tomada en cuenta en el próximo *Sprint Planning* para decidir el trabajo que se planeará.

Se muestra al *Product Owner* y *stakeholders* el incremento conseguido en el sprint, y se valora cómo ha sido el desarrollo del sprint.

La duración recomendada de esta reunión para un sprint de dos semanas es de una hora por semana.

Retrospectiva

Esta sesión consiste en que el equipo haga una reflexión sobre lo ocurrido durante el sprint con el objetivo de identificar tanto cosas buenas y que han agradoado al equipo como problemas para la mejora continua del desarrollo.

Existen distintas dinámicas que el *Scrum Master* propone al equipo para tratar de extraer la mayor cantidad de información posible. Estas actividades suelen tratar de responder a las preguntas: ¿Qué ha ido bien durante el sprint? y ¿qué podríamos mejorar de cara al siguiente?, cuyas respuestas se debaten entre los integrantes del equipo.

La duración recomendada es de una hora por cada semana del sprint.

Otros flujos

Existen otros eventos o actividades que se pueden realizar de forma habitual, pero que no se consideran clave en SCRUM.

Inception

La *Inception* se trata de todas las actividades que se realizan antes de comenzar el desarrollo del proyecto, es decir, su descubrimiento, ideación y definición. Es considerada la parte más importante de un proyecto, ya que definir bien qué producto se está buscando y hacer que cada uno de los integrantes del equipo tengan la misma visión de éste es clave para asegurar que el proyecto sale adelante [Ras10].

De manera general, en la *Inception* se busca responder a las siguientes preguntas [Ras18].

- 1. ¿Por qué estamos aquí?.
- 2. Crea un Elevator Pitch.
- 3. Diseña una caja del producto o Product Box.
- 4. Crea una NOT List.
- 5. Conoce a tus vecinos.
- 6. Muestra la solución.
- 7. Pregunta qué nos quita el sueño.

- 8. ¿Qué tamaño tiene el proyecto?.
- 9. Sé claro sobre lo que vas a dar.
- 10. ¿Cuánto tiempo nos va a llevar?.

Para responder a estas preguntas, en IECISA se llevan a cabo diferentes fases de las que se hablará en la sección 5.1.

Refinamiento

Consiste en una revisión del *Product Backlog* en el que junto con el *Product Owner* el equipo se asegura de que las historias de usuario están correctamente definidas y de que el equipo ha comprendido correctamente qué es necesario hacer para completarlas. También se verifica que las historias, hasta el momento, están correctamente priorizadas.

Artefactos

A continuación, se describen una serie de artefactos o conceptos comunes en SCRUM.

Product Backlog

Es una lista ordenada de requerimientos, características, errores, que el equipo define; cualquier elemento que sea necesario para el desarrollo correcto del producto.

El *Product Owner* prioriza los elementos del *Product Backlog* teniendo en cuenta los riesgos, el valor de negocio, las dependencias, el tamaño y la fecha para que deben ser completados.

Las historias de usuario deben ser visibles por cualquier miembro del equipo y suelen escribirse en un lenguaje que cualquiera puede entender, pero sólo el *Product Owner* puede decidir si se hacen cambios sobre la definición de las historias.

Mientras que el *Product Owner* se encarga de los elementos que componen el *Product Backlog*, el equipo de desarrollo se encarga de estimar las horas de trabajo o los puntos de historia, de tal forma que se pueda saber cuánto esfuerzo requiere cada historia.

Sprint Backlog

Se trata de la lista de trabajo que el equipo se ha propuesto completar durante el sprint.

La lista se construye tomando los elementos de la parte más alta del *Product Backlog* hasta que el equipo cree que el trabajo añadido es suficiente para el sprint. En este momento el equipo de desarrollo debe tener en cuenta su desempeño en los sprints pasados para seleccionar una cantidad de trabajo que sean capaces de alcanzar.

Las distintas historias que componen el *Sprint Backlog* no se asignan a ningún componente del equipo en concreto, sino que cada integrante selecciona las que cree que podrá completar. Con esto se trata de promover la autoorganización del equipo y la motivación. Es frecuente

4. MÉTODO DE TRABAJO

usar una tabla para hacer el seguimiento del proceso de desarrollo de las tareas del sprint, como se verá en el capítulo 5, que se divide en tres partes: trabajo por hacer, en proceso y completado.

Incremento del producto

Consiste en la suma de todos los elementos del *Product Backlog* que han sido completados durante un sprint, integrado con el trabajo completado de los anteriores. Un incremento debe ser completado al final de cada sprint.

Otros artefactos

Se describen a continuación otros artefactos que también tienen presencia en el marco de trabajo.

Definition of Done

Son unos criterios de aceptación que determinan cuándo un elemento del *Product Backlog* se puede considerar completado. Estos criterios pueden ser distintos y tener variaciones de unos equipos a otros, pero siempre deben ser consistentes dentro del mismo equipo.

Velocidad

Es una forma de medir el esfuerzo total que un equipo es capaz de hacer en un sprint. El valor se consigue evaluando el trabajo completado en el último sprint.

También es común hablar de velocidad media o velocidad media de los últimos sprint.

Pico o Spike

Período de tiempo usado para informarse sobre un concepto o crear un prototipo simple. En el caso de ONCOSUP, veremos que el Sprint 0 (sección 5.3) se trata de un spike.

Valores de Scrum

SCRUM define cinco valores: compromiso, coraje, foco, franqueza y respeto.

Estos valores se consideran clave para el uso exitoso del marco de trabajo como se puede leer en *La Guia de SCRUM* [SS17] “El uso exitoso de SCRUM depende de que las personas lleguen a ser más virtuosas en la convivencia con estos cinco valores. Las personas se comprometen de manera individual a alcanzar las metas del Equipo Scrum. Los miembros del Equipo Scrum tienen coraje para hacer bien las cosas y para trabajar en los problemas difíciles. Todos se enfocan en el trabajo del Sprint y en las metas del Equipo Scrum. El Equipo Scrum y sus interesados acuerdan estar abiertos a todo el trabajo y a los desafíos que se les presenten al realizar su trabajo. Los miembros del Equipo Scrum se respetan entre sí para ser personas capaces e independientes”.

Planificación de ONCOSUP

A la hora de hacer la estimación del trabajo que supondrá todo un proyecto y poder hacer así su planificación, en IECISA se usan distintas herramientas propias de la empresa.

En primer lugar y de forma general, en casi cualquier proyecto se usa una herramienta que estima los *puntos de historia estándar* o PHE y las horas que llevará completarlos. En esta herramienta se van añadiendo todas las historias de usuario, y para cada una de ellas se rellenan diferentes campos como pueden ser el tipo, la complejidad o la iteración en que se va a completar. Con estos datos, la herramienta proporciona una cantidad de PHE estimados para cada historia de usuario. Tras calcular los PHE, se indica una velocidad, que se basa en la velocidad obtenida en proyectos anteriores de características similares, junto con otros datos como pueden ser si se usan entornos o la metodología del cliente; con esta información y los puntos estimados la herramienta da una estimación de horas necesarias para el proyecto.

Como es la primera vez que en IECISA se usa JHipster, el equipo no está seguro de si esta herramienta proporcionará una estimación fiable. En este tipo de situación se recurre una hoja de cálculo en la que alguien experimentado se encarga de especificar para cada historia el número de horas de trabajo que considera que llevará completarla. Una vez llenada la hoja con las horas que llevará cada historia, se trata de dividir las horas obtenidas entre las horas requeridas para cada sprint.

Con la primera herramienta, y usando una velocidad de 10 puntos, que es la que suelen llevar proyectos similares en Java, la herramienta estima 216 PHE y más de 3.500 horas de trabajo, cantidad de horas que el equipo no considera fiable. Se decide usar la estimación por experiencia, en la que salen unas 850 horas de trabajo, número mucho más realista y que se puede dividir en los cinco sprints para los que da el presupuesto.

Cuadro 4.1: Planificación para ONCOSUP

Sprint	Horas estimadas	Puntos de historia (PHE)
1	169	42.8
2	166	43.1
3	168	41.9
4	169	47.8
5	169	39.9
		215.6

Con los datos que ha proporcionado la hoja de estimación, el trabajo se reparte en 5 sprints como muestra el cuadro 4.6. El número de horas por sprint excede las 160 que deberían ser, pero para cinco sprints es lo más ajustado que se puede planificar.

Tras esta estimación, y hacer una revisión del presupuesto, se concluye que el proyecto podría tener seis sprints en lugar de cinco, por lo que, teniendo en cuenta la estimación inicial y considerando la opinión del equipo de desarrollo, se decide que lo ideal es que para cada

4. MÉTODO DE TRABAJO

sprint se planifiquen unos 35 puntos y completar el proyecto en seis sprints. Esto permitirá ajustar mejor las 160 horas por sprint, y tener cierto margen si el alcance del proyecto se amplía en el transcurso de los sprints.

Capítulo 5

Resultados

En este capítulo se describirá el proceso de desarrollo de la aplicación según el método de trabajo presentado en el capítulo 4. Para cada fase del proyecto se detallan los elementos con los que se ha trabajado, los resultados, así como un breve análisis de cada sprint.

Antes de que la autora se uniera al equipo dentro de IECISA, el proyecto ya había pasado por las fases previas al inicio del desarrollo, la *Inception* y un Sprint 0 o inicial. Ya que asistir a estos eventos fue imposible, se proporcionó a la autora toda la documentación disponible, con el fin de que tuviera tanta información como fuese posible y pudiera estar al tanto de cuál era el objetivo del proyecto.

Inception

La *Inception* es una sesión cuyo objetivo es conseguir que todas las partes involucradas en el proyecto tengan una misma visión del proyecto y una misma idea. Para ello se ponen en común las expectativas de cada uno de los involucrados, se exponen los riesgos del proyecto y se intentan reducir las incertidumbres.

Durante el desarrollo de la *Inception* se realizan una serie de talleres para dejar bien definidas las respuestas a los diez puntos vistos en el capítulo 4, en la sección 4.3.6. En IECISA se sigue como referencia lo expuesto en las secciones II, III y VI de "*The Agile Smurai*".

En estos talleres se fomenta la participación y son muy visuales. La *Inception* de ONCO-SUP se dividió en cinco fases que se describen a continuación:

Visión.

Es necesario que todo el equipo comprenda por qué estamos aquí y cuál es el fin de este proyecto. Que el equipo tenga claro ésto le permitirá mejorar la toma de decisiones respecto a los posibles cambios o complicaciones que puedan ir apareciendo, realizar su trabajo mejor y además les permite encontrar soluciones mejores ya que todos conocen a la perfección qué se quiere hacer en el proyecto.

5. RESULTADOS

Tras haberse presentado y conocido todos los miembros del equipo, se expone la idea del proyecto, y se enuncia la *Frase del Capitán* (Figura 5.1). Esta frase describe de la forma más clara posible el propósito del *Product Owner* con respecto al producto que se va a desarrollar. La *Frase del Capitán* facilita al equipo la toma de decisiones respecto a posibles cuestiones que aparezcan en el futuro en lo referente al desarrollo del proyecto, por lo que es importante tenerla siempre en mente para no olvidar el objetivo del proyecto.

En esta primera fase se formula también un *Elevator Pitch* que especifica qué va a ser la aplicación (Figura 5.1). En este proyecto, el *Elevator Pitch* fue el siguiente

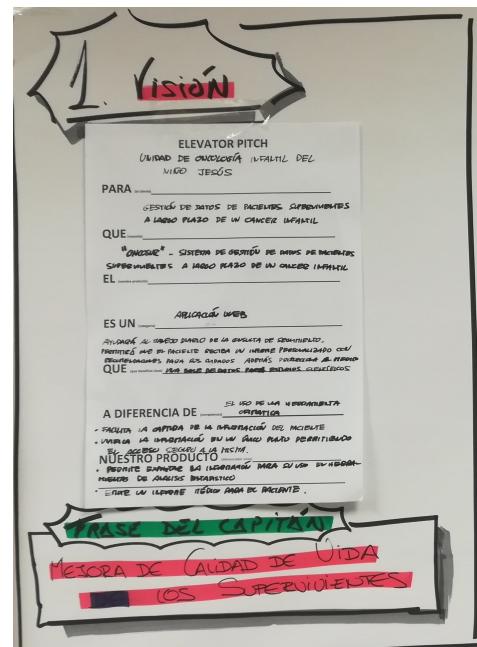


Figura 5.1: Resultado de la fase de visión

“

Para la unidad de oncología infantil del Hospital Infantil Universitario Niño Jesús, que necesita una gestión de datos de pacientes supervivientes a largo plazo de un cáncer infantil, “ONCOSUP” - Sistema de gestión de datos de pacientes supervivientes a largo plazo de un cáncer infantil; es una aplicación web que ayudará al manejo diario de la consulta de seguimiento, permitirá que el paciente reciba un informe personalizado con recomendaciones para sus cuidados y además proporciona al médico una base de datos para estudios científicos. A diferencia de la herramienta ofimática actual, nuestro producto facilita la captura de la información del paciente, unifica la información en un único punto permitiendo el acceso seguro a la misma, permite la exportación de la información para su uso en una herramienta de análisis estadístico y emite un informe médico para el paciente.”

Comunidad de Proyecto.

En líneas generales, la comunidad de proyecto suele ser más grande de lo que se piensa en un principio, por lo que es necesario tratar de identificar a los distintos participantes y definir sus roles en el proyecto lo antes posible, para de iniciar las distintas relaciones entre ellos.

Se optó por usar un panel en el que se pondrían pósits que identificasen a los participantes y sus roles (Figura 5.2)

En la figura 5.2 vemos que la comunidad de proyecto difiere ligeramente de la especificada en el cuadro 5.1. Esto se debe a que al comienzo del proyecto, la Dra. Herrero no creía poder dedicar el tiempo necesario al papel de *Product Owner*, pero finalmente trató de organizar su agenda para poder dedicar algunos días ya que ella es la primera interesada en la aplicación, es quien sabe qué necesita de ella y probablemente será la primera usuaria. Por otra parte, el Director de Proyecto de la UCLM y la autora, aparecen en el listado de participantes y no en el panel que se hizo en la *Inception*; la autora se une al proyecto al comienzo del sprint 1, por tanto no se contaba con la participación de ninguno de ellos cuando se realizó esta especificación.

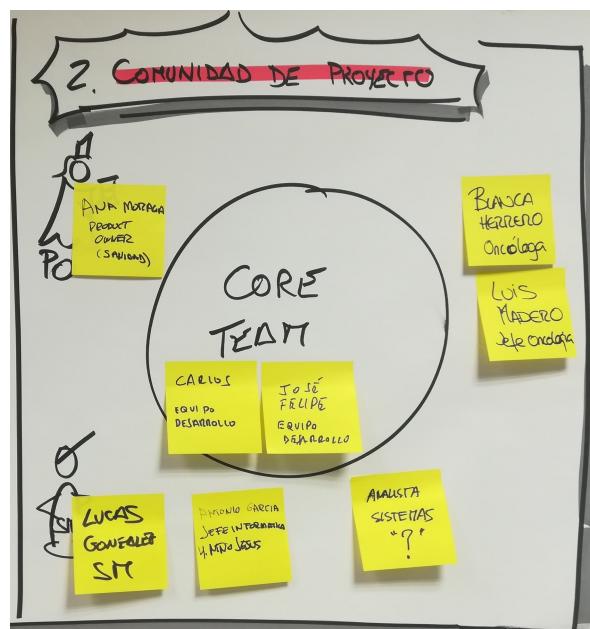


Figura 5.2: Comunidad de proyecto

Cuadro 5.1: Detalle de la comunidad de proyecto

Comunidad de proyecto	
Stakeholders	- Carlos Navas (IECISA). - Macario Polo Usaola (Director TFG UCLM).
Product owner	- Blanca Herrero (Oncóloga HNJ)
Equipo de desarrollo	- Carlos Romero Martín-Duarte (IECISA). - José Felipe Lozano Gijón (IECISA). - María López Carrasco (Autora)
SCRUM Master	- Lucas González (IECISA).

Restricciones y riesgos.

Identificar las restricciones del proyecto es un punto muy importante de la especificación que se realiza en la *Inception*, debido a que una vez identificados los elementos más restrictivos, se les asignará una importancia en la escala de *Trade Offs*.

También se definen unos riesgos, que son ordenados en una matriz (Figura 5.4). Esto sirve para que el equipo sea consciente del impacto que puede generar si alguno de estos riesgos se llegara a producir. Cuanto más a la derecha se coloca un riesgo mayor impacto puede generar si se produce. Por otro lado, cuanta mayor sea la probabilidad de que suceda, se situará más arriba en la matriz. Se trata de una forma muy visual y fácil de comprender para ordenar los distintos riesgos que pueden afectar al proyecto. Para cada riesgo se definen unas acciones de contingencia, es decir, qué hacer en caso de que ocurra alguno de los riesgos definidos.

5. RESULTADOS

Restricciones

Como se puede observar en la figura 5.3, en la dinámica destinada a esta tarea se deja un espacio lo suficientemente grande en el que se van escribiendo las restricciones que el equipo va identificando. Las restricciones son todos aquellos aspectos que limitan el proyecto. Para ONCOSUP se identificaron:

- Fecha. La fecha de ONCOSUP es muy restrictiva, pues el cliente quiere que el proyecto esté terminado para el mes de mayo, lo que deja tres meses para su desarrollo.
- Presupuesto. El presupuesto para el desarrollo es reducido, lo que se traduce en menos personas trabajando en el desarrollo y poca flexibilidad para extender la fecha de entrega.
- Tecnología. El equipo remitió un documento con las tecnologías a utilizar. Es necesario que el hospital apruebe las tecnologías propuestas por el equipo.
- Requisitos no funcionales (NFR). Es necesario saber qué tipo de restricciones no funcionales habrá por parte del hospital (navegadores, resolución de los equipos, etcétera)
- Calidad. Se desea que el proyecto tenga la mejor calidad posible, pero será algo que habrá que ir gestionando debido a otras restricciones como la fecha o el presupuesto.
- Rendimiento. El rendimiento debe ser tan bueno como sea posible, ya que requerirá accesos frecuentes a base de datos.
- Riesgos. Los riesgos son elementos que el equipo identifica como potenciales situaciones que podrían bloquear el desarrollo.



Figura 5.3: Restricciones y Trade offs

Riesgos

En lo referente a los riesgos, se identificó la siguiente lista:

- R1 - No disponer de la información de seguridad a tiempo. Además de los aspectos generales de la Ley Orgánica de Protección de Datos [lop18], el equipo de desarrollo

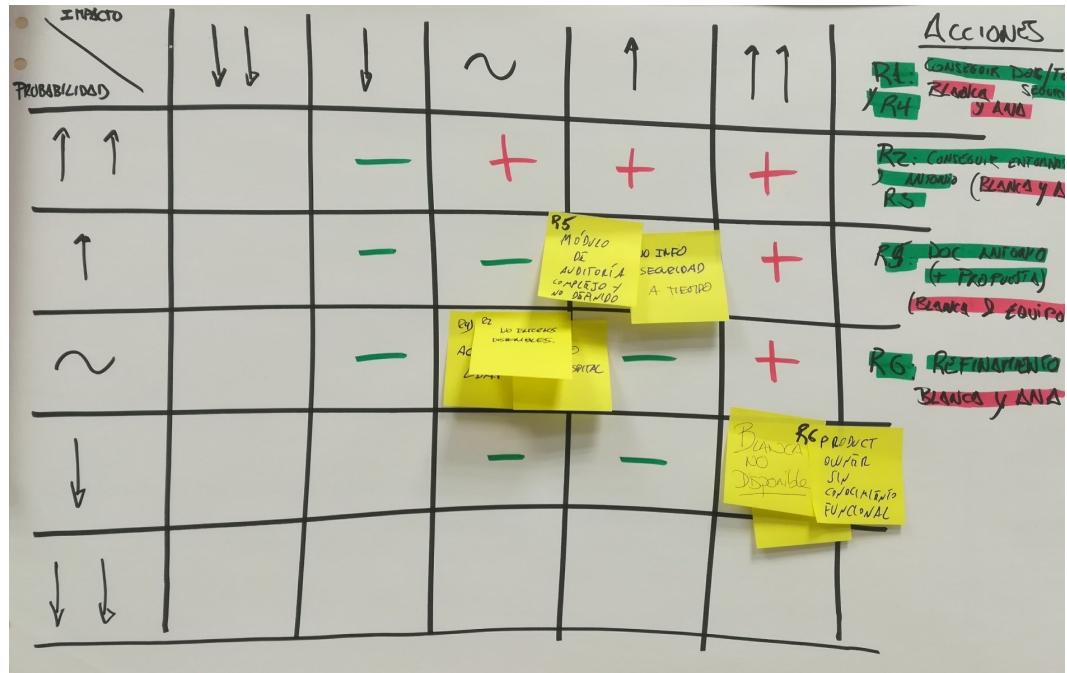


Figura 5.4: Matriz de riesgos

necesita saber si hay algún tipo de normativa específica a seguir respecto a la seguridad de los datos con los que trata la aplicación.

- R2 - Entornos no disponibles. Para la realización de pruebas, es necesario saber cuál será el entorno que proporcionará el hospital. Igualmente para el despliegue en producción.
- R3 - Acceso al hospital. Es necesario poder acceder al entorno del hospital, para ello desde el hospital deben informar sobre cómo acceder a través de VPN.
- R4 - No se dispone de información sobre el acceso a LDAP [ld18]. Es necesario conocer cómo acceder al LDAP, ya que los usuarios de ONCOSUP, serán usuarios del LDAP.
- R5 - Módulo de Auditoría complejo y no definido. El Módulo de Auditoría es complejo por la naturaleza del proyecto y no está definido qué datos deben guardarse para ser auditados.
- R6 - Product Owner sin conocimiento funcional. La Dra. Herrero no va a tener mucha disponibilidad durante el proyecto y otra persona debe ejercer de *Product Owner* teniendo menos conocimientos del dominio de la aplicación que la doctora, que es la usuaria final.

Una vez identificados los riesgos que hasta el momento pueden afectar al proyecto, el equipo los coloca en la matriz, de manera que de un vistazo se pueda ver cuán importantes es cada uno (Figura 5.4). En el caso de que a lo largo del desarrollo del proyecto se identifiquen

5. RESULTADOS

nuevos riesgos, se registrarán y se colocarán en la matriz de la misma forma.

Trade offs

En la parte inferior de la figura 5.3 vemos cómo se han organizado los *Trade offs* o *palancas*. Para el desarrollo de ONCOSUP lo menos relevante es la calidad, mientras que lo más restrictivo y menos tolerante a cambios será el presupuesto.

A cada elemento se le da un valor de relevancia a la hora de tomar decisiones, colocando los más importantes en la parte derecha y los menos importantes en la izquierda. No se puede asignar el mismo valor dos veces, con lo que se busca que el cliente piense bien qué elementos desea priorizar. Esta escala será la referencia a tomar si durante el desarrollo del proyecto se diera algún conflicto y fuese necesario priorizar un elemento en lugar de otro.

Story Map.

Teniendo ya claro el objetivo de la aplicación y sus limitaciones, la siguiente fase consiste en definir el *Product Backlog*.

El equipo trata de definir los *Flujos de negocio*, que son una forma de englobar diferentes historias que estarán relacionadas entre sí en una gran categoría. Cada una de las historias de usuario que se generan es asignada a uno de estos *Flujos de negocio*, permitiendo al equipo tener una visión más organizada de las diferentes partes que compondrán la aplicación.

El *Scrum Master* ayuda al *Product Owner* y al equipo de desarrollo a definir las historias de usuario dándoles unas directrices sobre su extensión, estructura, los criterios de aceptación, etcétera. Así se escriben estas historias en pósits (Figura 5.5), que luego se colocan en el panel del *Story Map*, dónde se ponen según la priorización que el *Product Owner* considera.

Además de las 26 historias de usuario que se han definido en esta etapa, se seleccionan las historias que compondrán el *Producto Mínimo Viable* (MVP), que en este caso son las que están por encima de la línea roja (Figura 5.6). Cuando esas historias hayan sido completadas, se obtendrá una versión mínima que aportará valor al cliente, que en este caso se corresponde con la épica de *Estudio preconsulta y Administración* casi en su totalidad. El motivo de haber seleccionado estas historias como MVP es que cuando estén completadas, permitirán a la doctora comenzar a introducir prácticamente todos los datos que necesita antes de pasar las consultas, de manera que podría ir preparando la base de datos con la información que considere.

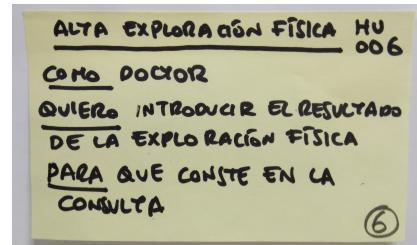


Figura 5.5: Ejemplo de una Historia de Usuario

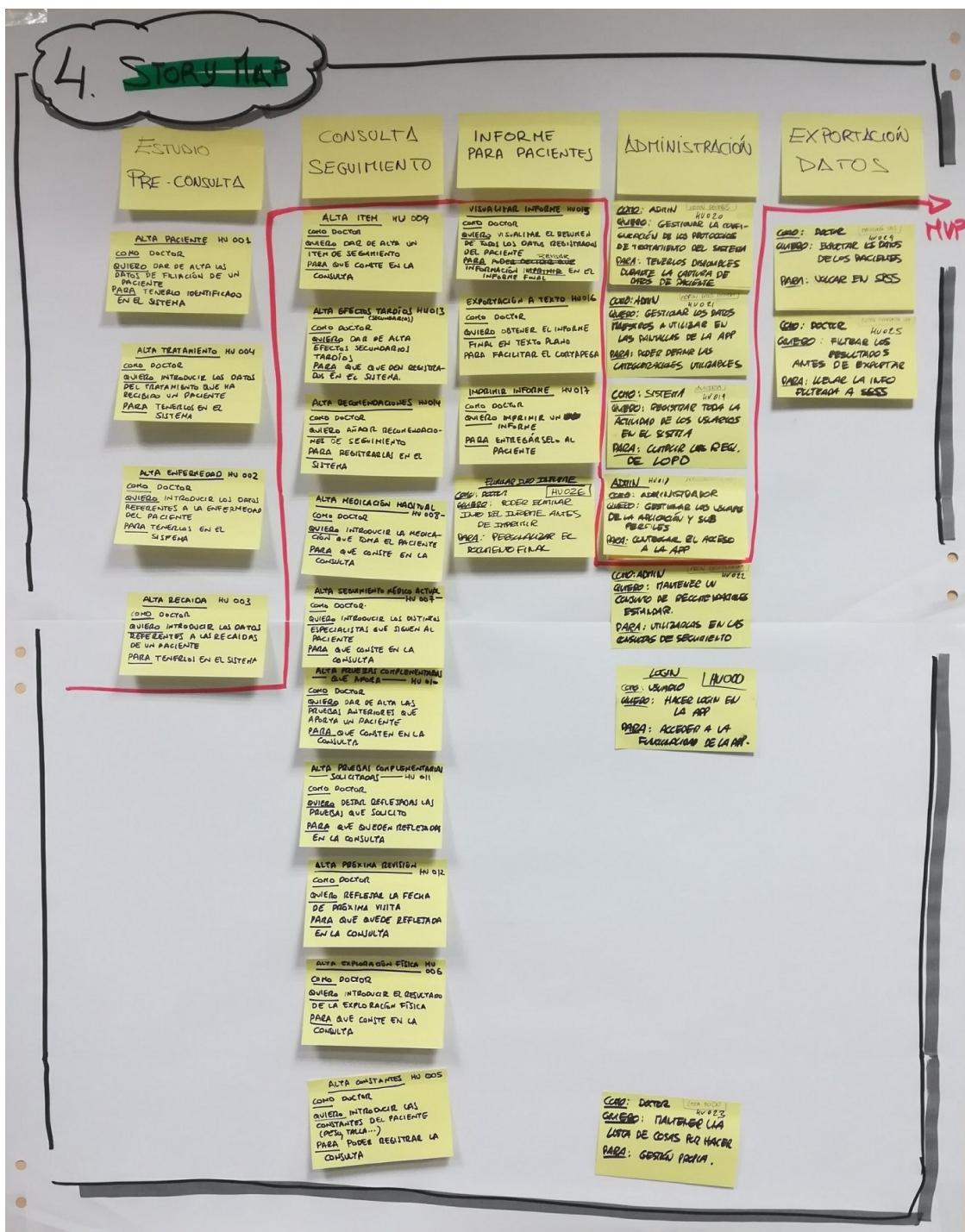


Figura 5.6: Product Backlog

Volcado en JIRA y Confluence.

Toda la información obtenida en estos talleres se registra para permitir al equipo gestionar todo el proceso de desarrollo del proyecto. Así, al acabar la *Inception*, el equipo registra todo en *Confluence* para documentar el desarrollo del proyecto, y en *JIRA* para su gestión del desarrollo.

5. RESULTADOS

The screenshot shows the Confluence interface. At the top, there's a navigation bar with icons for 'Espacios' and 'Personas'. Below the navigation bar is a sidebar titled 'ONCOSUP' which includes links for 'Páginas', 'Blog', 'Calendarios', 'ACCESES DIRECTOS DE ESPACIO', 'Retrospectivas', 'ÁRBOL DE PÁGINAS' (with sub-links 1. Seguimiento, 2. Requisitos, 3. Herramientas y Entornos, 4. Arquitectura/Diseño, 5. Testing, 6. Entregas, and Retrospectivas), and a star icon.

Figura 5.7: Indice de la documentación del proyecto

Confluence es una *wiki* abierta a todo el equipo que organiza la información de forma arborescente (Figura 5.7).

A la autora le ha parecido de especial interés la sección "Lecciones aprendidas", en la que los miembros del equipo anotan diferentes problemas que han surgido durante el proyecto y cómo solucionarlos, permitiendo encontrar solución rápida a problemas recurrentes o que no son fáciles de resolver. Además, los usuarios reciben notificaciones cuando se crea una nueva entrada en un espacio que siguen en *Confluence*, facilitando que todos los miembros del equipo estén al tanto de las actualizaciones que se hagan.

Al inicio del proyecto y haciendo uso de una gran cantidad de funcionalidades que *JIRA* pone a disposición de sus usuarios se registran todos los artefactos que se han generado durante el desarrollo de la *Inception*. Es posible registrar las historias de usuario, los riesgos del proyecto, los compromisos, tareas y subtareas relacionadas con el proyecto, lo que

ayuda a la gestión de este, permitiendo tener toda la información importante en un mismo espacio.

Además, como tanto *JIRA* como *Confluence* y *BitBucket* son todas herramientas de *Atlassian*, es posible, enlazar referencias de diferentes elementos creados en una herramienta a otro elemento creado en otra distinta. Por ejemplo, en el espacio de *Confluence* de ONCOSUP, en el apartado de compromisos, hay enlaces a los elementos creados en *JIRA*, dónde están especificados, o al subir código al repositorio (*commit*) podemos referenciar la tarea con la que corresponde la codificación que se va a subir al repositorio.

Por otra parte, en la sección 5.1.3 se hablaba de la matriz de riesgos, esta matriz no es algo que se hace en papel durante la *Inception* sin más, sino que la propia herramienta *JIRA* (ver Figura 5.8), con la información que se aporta al registrar un riesgo, genera esa matriz. Esta matriz se va actualizando según los riesgos se van resolviendo o aparecen nuevos, siendo lo ideal que al final del proyecto la matriz esté vacía. La matriz de riesgos se puede ver tanto desde *JIRA* como desde el espacio en *Confluence*, ya que ambas herramientas se sincronizan, en este caso la matriz que se muestra es la que se puede ver desde *Confluence*. También se aprecia que la matriz que muestra la figura 5.8 se corresponde con la figura 5.4 de la sección 5.1.3.

	Impacto				
Probabilidad	Muy Bajo	Bajo	Moderado	Alto	Muy Alto
Muy Alta	0 tiques	1 tique	0 tiques	0 tiques	0 tiques
Alta	0 tiques	0 tiques	1 tique	<u>2 tiques</u>	0 tiques
Moderada	0 tiques	0 tiques	2 tiques	0 tiques	0 tiques
Baja	0 tiques	0 tiques	0 tiques	0 tiques	1 tique
Muy Baja	0 tiques	0 tiques	0 tiques	0 tiques	0 tiques

Figura 5.8: Matriz de riesgos en Confluence

Incorporación a IECISA

La autora se incorpora a la actividad de la empresa el día 5 de febrero de 2018, coincidiendo precisamente con el comienzo del primer sprint.

Justo la primera actividad que se realizó al incorporarse la autora, y sin apenas tiempo para aclaraciones, fue el *Sprint planning* del Sprint 1 que se detalla en la sección 5.4. Tras este primer contacto con el proyecto se informó a la autora de todo lo ocurrido en el sprint 0 (ver sección 5.3) y en la *Inception* (ver sección 5.1), además de proporcionársele toda la documentación registrada en *Confluence* y la información del proyecto en *JIRA* con el fin de que durante los primeros días, y hasta que llegasen los equipos de trabajo, pudiera empaparse tanto como fuese posible sobre el proyecto.

Durante los primeros días, tanto a la autora como al resto de alumnos de FORTE en la empresa, se les dio una breve formación sobre SCRUM. La gran mayoría de los proyectos en IECISA usan este marco de trabajo para la gestión de sus proyectos, por lo que consideran muy importante conocerlo desde el primer día. Además, se animó a todos los FORTE a “trastear” con las herramientas (*JIRA* y *Confluence* principalmente) durante estos primeros días de aclimatación a la empresa.

De los artefactos producidos en la *Inception* y elementos que componen el proyecto, resultan de especial interés para la autora los que se describen en el listado a continuación:

5. RESULTADOS

- El equipo - Conocer a Carlos y José Felipe como compañeros de trabajo y al resto de la comunidad del proyecto es de especial relevancia en la incorporación al proyecto.
- Historias de Usuario - A la autora le habría resultado interesante poder asistir a la *Inception* y haber vivido cómo se generan estos artefactos y cómo todo el equipo trabaja para definir cada funcionalidad de la aplicación.
- MVP - Es también de interés cómo se decide el número mínimo de historias que aporta valor al cliente.
- Tecnologías - Por el poco presupuesto y el tiempo reducido se utiliza JHipster para la generación del esqueleto o arquitectura de la aplicación. Esta herramienta utiliza muchas otras para esta generación como, por ejemplo, Angular5, Spring Boot, Apache Tomcat, etcétera.

Después de ponerse al día con el proyecto y las herramientas que se utilizan en la empresa, la autora dedicó tiempo al estudio y aprendizaje de las diferentes tecnologías que se usan en el proyecto. En su front-end es dónde se incorpora la autora. La aplicación utiliza un patrón de diseño MVC que, pese a ser contenido que se estudia durante el desarrollo de la carrera, se trata de una arquitectura de aplicación con el que la autora apenas ha tenido experiencia; de esta forma, otra de las actividades que la autora hizo durante los primeros días, esta vez ya de forma individual, fue refrescar conocimientos y afianzar algunos nuevos respecto al MVC.

JHipster, herramienta principal que genera código para distintas herramientas. De todas las que usa quizá las más importantes y de las que menos conocimientos tiene la autora son Angular5 y Spring Boot. Spring Boot se trata de la herramienta principal que usa JHipster para generar la arquitectura de nuestra aplicación, esta herramienta se utiliza de forma simple, indicándole las características de nuestro proyecto nos generará la estructura básica de la aplicación. Además, JHipster se vale de Angular5 para generar el front-end de nuestra aplicación. De la misma forma que con el MVC, se dedicó parte de estos días al aprendizaje de Angular5 como framework de desarrollo front-end y el lenguaje *TypeScript* muy similar a *JavaScript*.

En figura 5.9, a continuación, se puede ver cómo es la arquitectura de la aplicación. Vemos la arquitectura habitual de un patrón MVC, siendo el cliente la vista, y estando en el servidor el controlador y el modelo que se comunica con la base de datos. Todos los elementos del servidor usan el framework de Spring para comunicarse entre ellos.

Se puede apreciar que Angular 5, nos genera un cliente basado en el patrón MVC también.

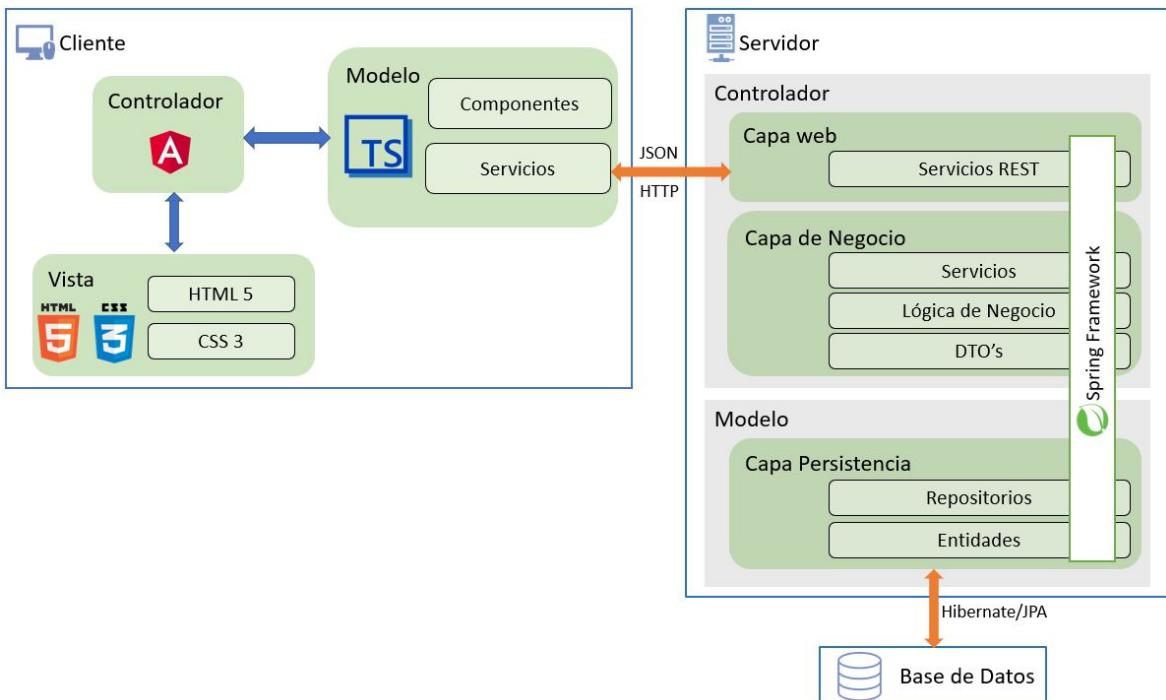


Figura 5.9: Arquitectura de la aplicación

Sprint 0

El fin de este sprint inicial es decidir qué herramientas se van a utilizar para el desarrollo del proyecto. Para ello, se proponen diferentes opciones y se realizan distintas pruebas. En función de los resultados obtenidos se toma la decisión de cuál es la más adecuada.

El presupuesto de ONCOSUP es reducido, lo que se traduce en un equipo de desarrollo pequeño y un tiempo limitado para completar el proyecto. El equipo al que se une la autora es el autodenominado “Equipo Negro”, compuesto únicamente por dos miembros: Carlos Romero Martín-Duarte y José Felipe Lozano Gijón.

Debido a las limitaciones del proyecto el equipo decidió probar una herramienta relativamente nueva: JHipster. Se trata de una plataforma de desarrollo para generar, desarrollar y desplegar aplicaciones web con Spring Boot y Angular5. Se tomó la decisión de probar JHipster con la idea de poder generar el esqueleto de la aplicación que se quería desarrollar, para después moldearla según las necesidades del proyecto, y así ser capaces de cubrir el alcance del proyecto a pesar de ser únicamente dos miembros en el equipo.

Como resultado de las pruebas de este sprint, se confirma que JHipster es la herramienta que se usará para el desarrollo. Si bien la herramienta debería facilitar al equipo conseguir desarrollar la aplicación en menos tiempo del que es habitual para aplicaciones de este tipo, se tiene en cuenta que será necesario que los miembros de equipo de desarrollo aprendan Angular5, framework que hasta el momento ninguno había usado, lo que puede ralentizar el desarrollo al inicio del mismo, pero se espera que el equipo vaya adquiriendo el conocimien-

5. RESULTADOS

to suficiente como para acelerar el ritmo y ser capaces de llegar a cumplir con el proyecto en la fecha acordada. También se asumen ciertas restricciones en favor de conseguir tener el proyecto dentro del tiempo y presupuesto acordado, por ejemplo, la personalización de estilos será limitada, JHipster ya crea una aplicación relativamente maquetada y responsive, que se modifica en la menor medida posible; hay decisiones de diseño que es necesario descartar adaptándonos a lo que JHipster nos genera, esto supone cambiar la idea de cómo será la navegación (que en un principio iba a basarse en pestanas) requiriendo aprender cómo JHipster, junto con Angular5, gestiona el acceso, creación y edición de los diferentes elementos que componen la aplicación y ser capaces de modificarlo para que la aplicación se comporte como se desea.

En este sprint también se decide qué software se usa para la base de datos. Inicialmente se pretendía usar MySQL, pero sería necesario pagar una licencia si nuestra aplicación no es *open source* [mys18]; se decide entonces, usar MariaDB que nos ofrece más opciones de almacenamiento que MySQL y operaciones más rápidas [mdb18].

SCRUM no define dentro de su marco de trabajo un sprint 0, pero fomenta el agilismo¹, lo que nos permite tomar decisiones sobre cómo afrontar ciertos riesgos en función de las necesidades del proyecto. Es por esto que se ha realizado este primer sprint, que, aunque no forma parte del desarrollo del mismo, sí que será muy útil de cara a que el equipo tome la mejor decisión según las necesidades del producto.

Sprint 1

El primer sprint de ONCOSUP comienza el día 5 de febrero, como ya se ha mencionado al inicio de la sección 5.2. En el capítulo 4 se habla de que la duración de los sprints es de dos semanas, por tanto este primer sprint finaliza el día 16 de febrero.

La primera actividad que se realiza al comenzar cualquier sprint es el *Sprint Planning*, al que la autora se une inmediatamente el día de la incorporación a IECISA.

Sprint Planning

Unos minutos antes de comenzar el *Sprint Planning*, los miembros del equipo de desarrollo y el *Scrum Master* le explicaron a la autora de qué trataba el proyecto para que pudiera comprender lo que se iba a hacer en esta reunión. Una vez aclarado lo que se iba a hacer, el equipo se conecta a través de videoconferencia con la *Product Owner*, se abre JIRA para ver las historias del proyecto y la pantalla se comparte con la *Product Owner* y se proyecta para que todos los participantes puedan ver la herramienta.

Hay que decidir qué historias serán las que se abordarán en este sprint, en la *Inception* ya se habían organizado las historias por orden de prioridad, así que ahora, tras haber hecho

¹Aunque la traducción correcta de *agile* sería agilidad, agilismo es la traducción que se usa en desarrollo de software para denominar esta corriente. <https://es.wikipedia.org/wiki/Agilismo>

una estimación del tiempo que llevará completar cada historia, se seleccionan tantas historias como el equipo cree que será capaz de completar de las de mayor prioridad. Además de esta selección de historias se hace un *refinamiento*, que consiste en revisar la especificación inicial de la historia que se introdujo al registrarla en *JIRA*, y que el *Product Owner* aclare, cambie o concrete más las características de ésta o las posibles dudas que el equipo de desarrollo pueda tener a la hora de abordar la historia. Tras unas horas de reunión, finalmente se planificaron y refinaron las siguientes historias para el Sprint 1 (ver cuadro 5.2).

Cuadro 5.2: Historias planeadas Sprint 1

Identificador	Resumen	Puntos historia
HU001	Alta Paciente	11
HU002	Alta Enfermedad	6
HU020A	Administrar Protocolos	12
HU020B	Administrar Fármacos	8

- HU001 -Alta Paciente. El objetivo de esta historia es poder dar de alta a un paciente en la base de datos. Es necesario introducir valores como su nombre, nacionalidad u hospital en el que se le trató. Una vez creado un paciente debe ser posible asociarle un diagnóstico.
- HU002 - Alta Enfermedad. Se necesita poder dar de alta enfermedades, en concreto los diagnósticos. Un diagnóstico tiene campos como fecha, patología, localización, etcétera. Estos diagnósticos deben poder ser enlazados a pacientes existentes en la base de datos.
- HU020A - Administrar Protocolos. Es necesario poder crear protocolos, es decir, poder definir cómo se trata la dolencia de un paciente.
- HU020B - Administrar Fármacos. Se desea poder dar de alta diferentes fármacos para poder asignarlos en el futuro a la hora de crear tratamientos.

A la hora de refinar las historias se refleja lo especificado en la reunión en cada una de las historias en *JIRA*. La figura 5.10 muestra un recorte de *JIRA* en dónde se puede ver cómo queda reflejada toda la información referente a una historia de usuario; bajo la descripción de la historia se concretan los criterios de aceptación de forma más o menos específica, dependiendo habitualmente, de la complejidad de la historia. Además, para cada historia de usuario se crean unas sub-tareas que también quedan reflejadas en *JIRA*. Estas subtareas sirven para facilitar el trabajo dividiéndolo en otras más sencillas, y para además llevar un mejor seguimiento del avance del proyecto; si no hubiera sub-tareas, una historia no se podría cerrar hasta haberla completado en su totalidad, en cambio con esta división, se pueden cerrar sub-tareas y de esta forma poder visualizar cuánto queda por completar de cada historia de usuario.

5. RESULTADOS

Figura 5.10: HU002 - Alta enfermedad

Desarrollo del sprint

Después de esta primera toma de contacto con los eventos de SCRUM, se empieza a valorar qué tipo de tareas podría comenzar a hacer la autora. Se decidió comenzar por tareas sencillas para, poco a poco, ir asignándole tareas más complejas como se verá en próximos sprints. Para el sprint 1 se le asignaron dos tareas: encargarse de la maquetación de las diferentes pantallas de la aplicación, y hacer el enlace entre un paciente y su diagnóstico, es decir asociar un diagnóstico existente en la base de datos a un paciente.

En cuanto a la primera tarea, la autora se encargó de averiguar cómo enlazar un archivo de estilos a un componente, y después generó unos estilos que en general y con algunas excepciones, se usarán en el futuro para todas las pantallas de la aplicación.

Para asociar un archivo de estilos a un componente es necesario ir a su archivo, y definir el nombre del archivo, tal y como se ve en el ejemplo del listado 5.1. Importante mencionar que es necesario que el archivo exista, a diferencia de en HTML que, si no se encuentra el archivo, simplemente no se cargan los estilos; en este caso la aplicación no compilaría.

```

1 // ... Code above ...
2
3 @Component ({
4   selector: 'jhi-paciente-detail',
5   templateUrl: './paciente-detail.component.html',
6   styleUrls: [
7     'paciente.css'
8   ]
9 })
10
11 // ... Code below .../

```

Listado 5.1: Asociación de una hoja de estilos a un componente

Respecto a los estilos, el listado 5.2 muestra un ejemplo de cuáles son los estilos que se han creado para la web. La información del detalle de cada entidad se ha organizado de tal forma que los elementos relacionados están en un mismo "grupo", al contenedor de los elementos

que forman parte de un grupo se le aplica la clase `.block` y los atributos más importantes son `display: flex` y `flex-wrap: wrap`. Estos atributos le indican al navegador que si ese contenedor tiene más elementos de los que caben en la pantalla horizontalmente, debe pasarlo a la línea de abajo en cuánto uno no quepa, permitiendo añadir cuántos elementos queramos a un contenedor. Otra clase importante es `.item`, que se le asigna a cada elemento dentro de un grupo (tanto título como valor). Con esta clase el navegador distribuye los elementos dentro de un grupo dándole un ancho del 25 % a cada uno, de esta forma tendremos 4 elementos en cada línea de un grupo.

La clase `.item-diagnóstico` es un caso especial, el valor de este campo es un texto que puede ser de cierto tamaño, como no hay más elementos a su derecha, se ha decidido darle más anchura a ese elemento para que el texto se distribuya por toda la página aprovechando el espacio. Esto se repetirá en otras ocasiones en el futuro, debido a que, en diferentes pantallas, hay diferentes casos en los que es necesario redistribuir los elementos para aprovechar el espacio de forma más eficiente.

Desde este punto en adelante estos serán los estilos que se aplicarán a cada pantalla de detalle, con ligeras modificaciones como es la de la descripción del diagnóstico.

```

1 .block{
2     width:100%;
3     display:flex;
4     flex-wrap:wrap;
5     justify-content: flex-start;
6     margin-bottom:15px;
7 }
8 .item{
9     width:25%;
10    margin-bottom:10px;
11 }
12 .item-título{
13     font-weight: bold;
14 }
15 .item-diagnóstico{
16     width:75%;
17 }
18 .half-block{
19     display:flex;
20     justify-content: space-between;
21 }
22 .half-item{
23     width: 49.5%;
24 }
```

Listado 5.2: Estilos generales para páginas de detalle

En la parte final de los estilos aparecen las clases `.half-block` y `half-item` que son estilos específicos para los componentes de diálogo. Con `justify-content: space-between` el navegador tratará de disponer los elementos de un contenedor separándolos tanto como pueda, para evitar que el navegador ponga un espacio demasiado grande, a cada elemento del bloque se

5. RESULTADOS

le asigna un ancho de casi el 50 %, de tal forma que el navegador ponga solo un pequeño espacio entre los elementos.

Las figuras 5.11 y 5.12 muestran el resultado de estos estilos. Como se ha comentado en varias ocasiones JHipster genera una aplicación prácticamente funcional, así que la cantidad de estilos necesaria para maquetar los elementos no es elevada. El trabajo de la autora fue redistribuir los elementos para aumentar la usabilidad de la aplicación y hacer que fuese más *user friendly*.

The screenshot shows a patient profile page for 'Pepe Martín'. At the top, there's a header with the logo 'Hospital Infantil Universitario Niño Jesús', navigation links for 'Inicio', 'Registro', 'Administración', and 'Cuenta', and a search bar. Below the header, the patient's name 'Pepe Martín' and ID '9678619849684651498' are displayed. The main content area contains a table with patient information:

Nombre Pepe	Apellidos Martín	Fecha Nacimiento 8 mar. 2018	Sexo Varón
Email pepa@mail.com	Teléfono 655655655	Nacionalidad Española	Etnia Caucásico
Número Seguridad Social 654981968465198	Tarjeta Sanitaria 684981984984	Cipa 654987498749	Número de Historia Clínica 9678619849684651498
Hospital HGM	Diagnóstico Tumor Sólido - Diagnóstico específico de prueba		
Comunidad Diagnóstico Comunidad Valenciana	Provincia Diagnóstico Alicante	Comunidad Actual Castilla-La Mancha	Provincia Actual Ciudad Real

At the bottom, there are buttons for 'Consultas' (Consultations), 'Volver' (Back), and 'Editar' (Edit).

Figura 5.11: Detalle de un paciente

La segunda tarea de la que tuvo que encargarse la autora fue de enlazar un paciente con su diagnóstico a través de un *id*. Esta tarea, que en un principio puede parecer sencilla de resolver con un sencillo script y añadiendo una variable en el paciente, con Angular5 no es tan sencillo. Si bien es cierto que Angular5 facilita muchas tareas, también es necesario aprender bien cómo funciona y habituarse a su uso.

Para realizar esta tarea no es necesario realizar cambios muy grandes, pero sí es necesario hacerlos en bastantes archivos. Para explicar esto se hará un recorrido desde el html del componente, hasta el guardado del *id* en base de datos.

```
1 <!-- More code above -->
2
3 <button *ngIf="!paciente.diagnosticoId" class="btn btn-primary btn-sm jh-create-
  entity create-diagnostico" [routerLink]="/" { outlets: { popup: 'diagnostico-new/' + paciente.id } }>
4   <span class="fa fa-plus"></span>
5   <span class="hidden-sm-down" jhiTranslate="oncosupApp.diagnostico.home.
    createLabel">Create new Diagnostico</span>
6 </button>
7
8 <!-- More code below -->
```

Crear o editar Paciente ×

Nombre	Apellidos
<input type="text"/>	<input type="text"/>
Fecha Nacimiento	Sexo
<input type="text"/>	<input type="text"/> ▼
Número de Historia Clínica	Número Seguridad Social
<input type="text"/>	<input type="text"/>
Tarjeta Sanitaria	Cipa
<input type="text"/>	<input type="text"/>
Teléfono	Email
<input type="text"/>	<input type="text"/>
Nacionalidad	Etnia
<input type="text"/> ▼	<input type="text"/> ▼

Figura 5.12: Crear o editar un paciente

Listado 5.3: Botón añadido al detalle del paciente

En el html del detalle del paciente se añade un botón para la creación (Listado 5.3) y edición de un diagnóstico. A través del atributo *routerlink* se indica qué elemento debe abrirse al hacer clic en el botón; en este caso *diagnostico-new* abrirá el *pop-up* para crear o editar el diagnóstico. A este *routerlink* se le ha añadido el *id* del paciente para, al mismo tiempo que se hace una consulta para crear el diagnóstico, hacer otra consulta que seleccione el paciente con dicho *id* y le asigne el diagnóstico que acabamos de crear.

Este cambio no se puede hacer sin más, existe un archivo en el que se definen las distintas rutas a través de las que se puede abrir un componente, es necesario indicar este cambio también en ese archivo como muestra la figura 5.4, en la línea 5, dónde se ha añadido el *id* del paciente a la ruta.

5. RESULTADOS

```
1 // ... More code above ... //
2
3 export const diagnosticoPopupRoute: Routes = [
4     {
5         path: 'diagnostico-new/:pacienteId',
6         component: DiagnosticoPopupComponent,
7
8 // ... More code below ... //
```

Listado 5.4: Código dónde se definen las rutas para el diálogo del diagnóstico

Además, como último cambio en lo referente a las rutas y el paso de parámetros a través de éstas, necesitamos indicar al diálogo de edición del diagnóstico, que al abrir el *pop-up* deberá pasarle como parámetro el *id* que recibe en la ruta. En la parte superior del listado 5.5 vemos que se hace una primera comprobación que verifica si se está recibiendo un *id* (en este caso se trata del *id* del diagnóstico), esta comprobación es para saber si debemos crear un diagnóstico nuevo o modificar uno ya existente. En ambos casos se ha añadido el parámetro ***paciente_id*** (líneas 7 y 9), con el fin de tener siempre identificado el paciente al que corresponde el diagnóstico.

```
1 // ... More code above ... //
2
3 ngOnInit() {
4     this.routeSub = this.route.params.subscribe((params) => {
5
6         if ( params['id'] ) {
7             this.diagnosticoPopupService.open(DiagnosticoDialogComponent as
8                 Component, params['pacienteId'], params['id']);
9         } else {
10             this.diagnosticoPopupService.open(DiagnosticoDialogComponent as
11                 Component, params['pacienteId']);
12         }
13     });
14 }
15 // ... More code below ... //
```

Listado 5.5: Paso del *id* al pop-up del diagnóstico

Tras esto se está enviando el *id* del paciente en la ruta al abrir el diálogo de creación y edición de un diagnóstico, pero es necesario indicar que ahora tenemos una variable que contendrá el valor de este *id*.

Esto se soluciona en tres pasos, primero es necesario ir al archivo *.model* correspondiente, en el que se especifican todos los atributos que va a tener una entidad, y añadir ahí la variable ***paciente_id*** que se va a recibir, de la forma en que se puede ver en la línea 17 de la figura 5.6 a continuación.

```

1 // ... More code above ... //
2
3 export class Diagnostico implements BaseEntity {
4     constructor(
5         public id?: number,
6         public fechaDiagnostico?: any,
7         public patologiaOncologica?: PatologiaOncologica,
8         public diagnosticoEspecifico?: string,
9         public estadioLeucemia?: EstadioLeucemia,
10        public estadioTumorSolido?: EstadioTumorSolido,
11        public estadioOtros?: string,
12        public localizacion?: string,
13        public recaida?: boolean,
14        public recaidas?: BaseEntity[],
15        public cie10Id?: number,
16        public cie10SecundarioId?: number,
17        public pacienteId?: number,
18    ) {
19        this.recaida = false;
20    }
21 }
22
23 // ... More code below ... //

```

Listado 5.6: Archivo .model del diagnóstico

En el listado 5.6 se ha añadido un parámetro a la función *open*, por tanto es necesario indicar en su declaración que recibirá este parámetro. La figura 5.7 muestra en la línea 3 cómo se ha añadido esto a la declaración de la función. Además, en la parte en la que se guarda un diagnóstico al crearlo, se ha modificado de tal manera que ahora creamos el diagnóstico antes de guardarla, así podemos asignarlo al paciente correspondiente antes de almacenar el diagnóstico creado, a diferencia de cómo se estaba guardando antes (comentario línea 8).

```

1 // ... More code above ... //
2
3     open(component: Component, pacienteId: number, id?: number | any): Promise<
4         NgbModalRef> {
5
6     // ... More code between ... //
7
8     setTimeout(() => {
9         // this.ngbModalRef = this.diagnosticoModalRef(component, new
10            Diagnostico());
11         const diagnostico: Diagnostico = new Diagnostico();
12         diagnostico.pacienteId = pacienteId;
13         this.ngbModalRef = this.diagnosticoModalRef(component, diagnostico);
14         resolve(this.ngbModalRef);
15     }, 0);
16
17 // ... More code below ... //

```

Listado 5.7: Código del pop-up del diagnóstico

5. RESULTADOS

En este punto sólo queda hacer la llamada para que el servidor efectúe las operaciones necesarias sobre la base de datos, para el paso de datos entre cliente y servidor, Angular5 utiliza un patrón denominado *Data Transfer Object*(DTO) u objeto de transferencia de datos, que permite hacer de intermediario entre el cliente y el servidor, almacenando todos los cambios que se van realizando y permitiendo efectuar menos llamadas entre cliente y servidor. Así lo último que falta antes de poder guardar el *id* del diagnóstico junto a su paciente es añadir el *id* al DTO de diagnóstico. La figura 5.8 muestra el código añadido a este archivo, que se corresponde con la declaración de la variable y sus métodos de consulta y modificación.

```
1 // ... More code above ... //
2
3     private Long pacienteId;
4 // .. More code between ... //
5
6     public Long getPacienteId() {
7         return pacienteId;
8     }
9
10    public void setPacienteId(Long pacienteId) {
11        this.pacienteId = pacienteId;
12    }
13
14 // ... More code below ... //
```

Listado 5.8: Código añadido al DTO de diagnóstico

Tras todos los cambios explicados más arriba, ahora se puede ir al archivo *DiagnosticoServiceImpl.java*, en el que se añaden, además de las importaciones pertinentes, las líneas 10, 11, 12 del listado 5.9, que finalmente, tras guardar el diagnóstico, busca en la base de datos el paciente correspondiente al *id* recibido, y le asigna el diagnóstico creado.

```
1 // ... More code above ... //
2
3     public DiagnosticoDTO save(DiagnosticoDTO diagnosticoDTO) {
4         log.debug("Request to save Diagnostico : {}", diagnosticoDTO);
5         Diagnostico diagnostico = diagnosticoMapper.toEntity(diagnosticoDTO);
6         diagnostico = diagnosticoRepository.save(diagnostico);
7         DiagnosticoDTO result = diagnosticoMapper.toDto(diagnostico);
8         diagnosticoSearchRepository.save(diagnostico);
9
10        Paciente paciente = pacienteRepository.findOne(diagnosticoDTO.
11            getPacienteId());
12        paciente.setDiagnostico(diagnostico);
13        pacienteRepository.saveAndFlush(paciente);
14
15        return result;
16    }
17 // ... More code below ... //
```

Listado 5.9: Asignación de un diagnóstico al paciente

Para el primer sprint se hizo un refinamiento en el *Sprint planning*, pero para el resto de

sprints lo habitual es aproximadamente a la mitad del sprint anterior se haga el refinamiento de las historias que podrían ser planeadas en el siguiente sprint, de esta manera en la segunda semana del primer sprint se hizo el refinamiento de las posibles historias que se planearían para el segundo. Esta rutina se repite a lo largo de todos los sprints.

Sprint Review

Finalizadas las dos semanas de duración del sprint, se hace la sesión de *Sprint Review*. En esta reunión, a la que asiste todo el equipo, se hace un repaso por las historias que se habían planeado, para que tanto el equipo como el *Product Owner* tengan fresco lo qué debería estar hecho a esta altura del sprint. Después se le explica al *Product Owner* qué historias se han completado y cuáles no, y las razones por las que no ha sido posible completarlas, lo que se tendrá en cuenta en el siguiente *Sprint Planning* para ajustar mejor la carga de trabajo que el equipo puede completar. Al finalizar el sprint 1 se completaron todas las historias de usuario a falta de la “HU020A - Administrar Protocolos” debido a que era una tarea muy grande, que, a pesar de tener múltiples subtareas completadas, al no estarlo todas, no se pudo cerrar.

Tras este breve repaso sobre las historias, se le enseña al *Product Owner* el resultado del sprint. Como se explicaba en el *Sprint Planning* al principio de la sección, estas reuniones se hacen a través de videoconferencia y compartiendo la pantalla de tal forma que todos puedan verla, de este modo se le puede hacer una demostración al *Product Owner* de lo avanzado en la aplicación, haciendo un recorrido de cómo sería su uso, al menos hasta el punto en el que está el desarrollo. Afortunadamente, a pesar de no haber sido posible completar todas las historias planificadas, la *Product Owner* queda muy contenta con lo que ha podido ver en este *Sprint Review*, esta opinión es muy bien recibida por el equipo de desarrollo y se refleja en la *Retrospectiva* a continuación.

Retrospectiva

En la sesión de *retrospectiva* el *Scrum Master* selecciona una dinámica para que el equipo la lleve a cabo en esta sesión. En la pared se pondrán pósits, divididos en 3 categorías, lo que ha gustado, lo que no ha gustado y lo que se ha echado en falta. Cada miembro del equipo escribe en pósits y organiza según estas categorías sus sensaciones respecto al desarrollo del sprint. Cuando todos los miembros han anotado todo lo que querían, se explica el porqué de cada uno de sus pósits y se debate con el equipo la razón de haber puesto ese pósit en la pared. Esta sesión se trata de resaltar lo que se ha hecho bien y detectar los elementos que se pueden mejorar para enriquecer la experiencia del equipo en el desarrollo del proyecto.

En este sprint, algunas de las cosas se han destacado en la *retrospectiva* han sido:

- Cooperación con la *Product Owner*. Según las experiencias del equipo en otros proyectos, no es habitual que el *Product Owner* tenga una actitud tan positiva y facilite

5. RESULTADOS

tanto el trabajo al equipo de desarrollo.

- Control de JHipster. A lo largo del sprint han ido surgiendo diferentes problemas que, poco a poco, el equipo de desarrollo ha ido resolviendo. Esto será de mucha utilidad a la hora de enfrentarse a problemas nuevos con la herramienta en el futuro ya que el equipo empieza a aprender cómo funciona JHipster.
- El cambio a Angular5. Aunque tanto para la autora como para los integrantes del equipo de desarrollo Angular era algo totalmente nuevo y difícil de aprender en un principio, valoran de forma muy positiva las posibilidades que ofrece.
- “Tropezones positivos”. A pesar de haber tenido problemas y no haber sido capaces de terminar todas las historias del sprint, se valora mucho lo aprendido en estos “tropezones” de cara al futuro.
- El JDL (JHipster Domain Language) es una herramienta muy potente que permite definir tanto entidades como relaciones entre éstas, ahorrando mucho trabajo. En contraposición, un pequeño cambio en una entidad provoca que al volver a importar el JDL ésta se regenere por completo, habiendo perdido todos los cambios hechos. Este problema será algo con lo que tratar en cada sprint, ya que las diferentes entidades se van definiendo poco a poco y no tenemos toda la información de cada una de ellas desde el principio.
- El equipo proporcionado a la autora era antiguo y poco potente, lo que ha ralentizado su trabajo durante el sprint.
- Historias demasiado grandes. Algunas historias tienen una carga de trabajo demasiado grande, como ha ocurrido con la historia “HU020A - Administrar Protocolos”. JIRA no permite cerrar una historia si no se han completado todas sus tareas, por lo que tuvo que quedarse sin cerrar a pesar de haber completado gran parte de ella.

De todo lo hablado durante la *retrospectiva*, se acuerda tratar de conseguir un mejor equipo para la autora, y además se realiza un acuerdo de Trabajo; el equipo decide que todas aquellas historias de usuario que tengan más de diez puntos de historias serán divididas para simplificarlas en historias más pequeñas y fáciles de alcanzar.

Finalmente, a pesar de no haber conseguido acabar con todas las historias, tras el *Sprint Review* y la *retrospectiva* el equipo cierra el sprint con buen sabor de boca y buenas sensaciones para el siguiente.

Sprint 2

Dos semanas después, tras la finalización del primer sprint, comienza el sprint 2, el día 19 de febrero y finaliza el día 2 de marzo.

Sprint Planning

Como se ha mencionado en varias ocasiones, la primera actividad que se realiza al comenzar un sprint es el *Sprint Planning*. Durante el sprint anterior se realizó la sesión de refinamiento, en la que se concretaron los criterios de aceptación de las historias candidatas a ser planeadas para el segundo sprint. Con ese trabajo adelantado durante el primer sprint, esta reunión tiene menos duración que la del sprint 1.

Los miembros del equipo se reúnen, como es habitual, a través de videoconferencia con la *Product Owner* para decidir qué historias son las que son capaces de aborda de cara a este sprint. Tras debatir cuáles son las historias que se tratarán de completar, y aplicar el acuerdo de trabajo acordado al final del sprint anterior, el *Backlog* del sprint 2 se muestra en el cuadro 5.3.

Cuadro 5.3: Historias planeadas Sprint 2

Identificador	Resumen	Puntos historia
HU020A	Administrar Protocolos	12
HU003	Alta Recaída	10
HU004A	Alta Tratamiento:Resumen Tratamiento	1
HU004B	Alta Tratamiento:Quimioterapia	3
HU004C	Alta Tratamiento: Radioterapia	2
HU004D	Alta Tratamiento: Cirugía	2
HU004E	Aldta Tratamiento: Trasplante	10
HU004F	Alta Tratamiento: Otros tratamientos	2

La historia *Alta Tratamiento* tenía 20 puntos historia, debido a que se acordó dividir las historias de más de 10 puntos en historias más pequeñas, se puede ver en el cuadro 5.3, cómo el equipo la ha dividido en 6 historias más pequeñas, siendo únicamente una de ellas superior a los 3 puntos. El equipo es consciente de que no debería haber quedado ninguna historia con tantos puntos como *Alta tratamiento: Trasplante*, pero tras debatir cuál sería la mejor forma de hacer la división, se llegó a la conclusión de que esa historia no permitía dividirla en otras más pequeñas.

- HU002A - Administrar Protocolos. Es necesario poder crear protocolos, es decir, poder definir cómo se trata la dolencia de un paciente. Esta tarea quedó por terminar en el sprint anterior.
- HU003 - Alta Recaída. Permite registrar una o más recaídas para un paciente.
- HU004 Alta Tratamiento. Engloba todos los elementos que conforman un tratamiento y que es necesario poder crear antes de pasar la consulta con el paciente. Se divide en:

5. RESULTADOS

- HU004A - Alta Tratamiento: Resumen Tratamiento. Registrar un resumen de cuál es el tratamiento.
- HU004B - Alta Tratamiento: Quimioterapia. Dar de alta una quimioterapia a la que se le podrá asignar un fármaco de los que hayan sido creados previamente en la aplicación.
- HU004C - Alta tratamiento: Radioterapia. Registra la información relacionada con una radioterapia.
- HU004D - Alta tratamiento: Cirugía. Permite registrar una cirugía e indicar entre otros valores una amputación o no.
- HU004E - Alta tratamiento: Trasplante. Dar de alta un trasplante para un paciente.
- HU004F - Alta tratamiento: Otros tratamientos. Se trata de una opción que permite indicar si el paciente ha tenido algún tratamiento que no se corresponda con ninguno de los anteriores.

Al igual que se vio en la sección 5.4.2, también se especifican en *JIRA* los criterios de aceptación para considerar que una tarea esté completada. En relación con los criterios de aceptación de una historia, cabe mencionar la *Definición de hecho* o *Definition of done*, que se trata de una lista con ciertas condiciones que todas las historias deben cumplir para considerarlas completadas. El *Definition of done* suele comenzar siendo una lista pequeña que va aumentando con el desarrollo del proyecto, de manera que a medida que el equipo detecta nuevas características que las historias deberían cumplir, las van añadiendo. En el sprint 2 las condiciones eran solo tres: las historias deben estar codificadas, probadas y maquetadas.

Desarrollo del sprint

Dado que la *Product Owner* estuvo muy contenta con la poca maquetación que se le pudo mostrar en la Sprint Review del sprint 1 (ver sección 5.4.3), la autora siguió encargándose de ésta durante el resto de sprints. A parte de las tareas de maquetación, se le fueron dando a la autora tareas cada vez más complejas, aumentando su dificultad con cada sprint.

Una de las nuevas tareas que se le encargó a la autora era investigar e implementar *combo-Box* dependientes; esto es, que en función del valor seleccionado en un *comboBox* aparezcan unos valores u otros en otro *comboBox*. Tras un poco de investigación sobre cómo permite Angular 5 hacer ésto la solución que se aplicó se explica a continuación.

Primeramente en el código HTML del componente (listado 5.10) es necesario añadir un par de cambios, en el primer *comboBox* se añade una llamada al método “onComunidadActual(paciente.comunidadActualId)” cada vez que el valor seleccionado del combo cambia. En el segundo combo cambiamos la lista de la que se extraen los datos en el ”*ngFor” a “provinciaPaisActual”.

```

1  <!-- More code above -->
2  <div class="half-block">
3      <div class="form-group half-item">
4          <label class="form-control-label" jhiTranslate="oncosupApp.paciente.
5              comunidadActual" for="field_comunidadActual">Comunidad Actual</label>
6          <select class="form-control" id="field_comunidadActual" name="
7              comunidadActual" (change)="onComunidadActual(paciente.
8                  comunidadActualId)" [(ngModel)]="paciente.comunidadActualId" >
9              <option [ngValue]="null"></option>
10             <option [ngValue]="comunidadAutonomaOption.id" *ngFor="let
11                 comunidadAutonomaOption of comunidadesAutonomas; trackBy:
12                     trackComunidadAutonomaById">{{comunidadAutonomaOption.nombre}}</
13                     option>
14             </select>
15         </div>
16         <div class="form-group half-item">
17             <label class="form-control-label" jhiTranslate="oncosupApp.paciente.
18                 provinciaActual" for="field_provinciaActual">Provincia Actual</label>
19             <select class="form-control" id="field_provinciaActual" name="
20                 provinciaActual" [(ngModel)]="paciente.provinciaActualId" >
21                 <option [ngValue]="null"></option>
22                 <option [ngValue]="provinciaPaisOption.id" *ngFor="let
23                     provinciaPaisOption of provinciasPaisActual; trackBy:
24                         trackProvinciaPaisById">{{provinciaPaisOption.nombre}}</option>
25             </select>
26         </div>
27     </div>
28     <!-- More code above -->

```

Listado 5.10: ComboBox dependientes

Ahora solamente queda cambiar el contenido de la lista “provinciaPaisActual” en función del valor elegido. Para ello implementamos en el archivo *ts* del componente la lógica del método “onComunidadActual(paciente.comunidadActualId)” (listado 5.11). El funcionamiento del método es sencillo, se inicializa la lista para que no contenga ningún valor, después se recorre la lista de provincias y se añaden a la lista únicamente las que tienen un *id* que coincide con el de la comunidad seleccionada.

```

1 // ... More code above ... //
2 onComunidadActual(selectedValue) {
3     this.provinciaPaisActual = [];
4     for (let i = 0; i < this.provinciapais.length; i++) {
5         if (this.provinciapais[i].comunidadAutonomaId === selectedValue) {
6             this.provinciaPaisActual.push(this.provinciapais[i]);
7         }
8     }
9 }
10 // ... More code below ... //

```

Listado 5.11: Implementación onComunidadActual(paciente.comunidadActual)

La otra tarea que se le encargó a la autora fue que en el detalle del paciente apareciera el diagnóstico específico junto con el diagnóstico del paciente. El procedimiento para solucionar esto es similar a lo visto en el sprint 1. Primero se añade la variable y sus *getter* y *setter*

5. RESULTADOS

en el archivo PacienteDTO.java (listado 5.12).

```
1 // ... More code above ... //
2
3 private String diagnosticoEspecifico;
4
5 // ... More code between ... //
6
7 public String getDiagnosticoEspecifico() {
8     return diagnosticoEspecifico;
9 }
10
11 public void setDiagnosticoEspecifico(String diagnosticoEspecifico) {
12     this.diagnosticoEspecifico = diagnosticoEspecifico;
13 }
14
15 // ... More code below ... //
```

Listado 5.12: Cambios necesarios en el DTO para mostrar el diagnóstico específico

Se mapea la variable en el *PacienteMapper.java* para poder tener el valor en el DTO (listado 5.13).

```
1 // ... More code above ... //
2
3 @Mapping(source = "diagnostico.diagnosticoEspecifico", target = "diagnosticoEspecifico")
4
5 // ... More code below ... //
```

Listado 5.13: Línea añadida al PacienteMapper.java

Finalmente falta añadir este campo en el html como se puede ver en la línea 9 del listado 5.14.

```
1 <!-- More code above -->
2
3 <div class="item-diagnostico">
4     <span class="item-titulo" jhiTranslate="oncosupApp.paciente.diagnostico">
5         Diagnostico</span><br>
6     <div *ngIf="paciente.diagnosticoId">
7         <a [routerLink]="/diagnostico', paciente.diagnosticoId]">
8             <span jhiTranslate="{{'oncosupApp.PatologiaOncologica.' + paciente.
9                 diagnosticoNombre}}}></span>
10            </a>
11            <span> - {{paciente.diagnosticoEspecifico}}</span>
12        </div>
13        <!-- More code between -->
14    </div>
15
16 <!-- More code below -->
```

Listado 5.14: adición de un span con el diagnóstico específico

A pesar de que esta tarea fue sencilla, hubo algunos problemas para hacerla funcionar.

Una vez realizados los cambios descritos más arriba, al compilar y levantar la aplicación no aparecía el diagnóstico específico en el detalle. Tras algo de investigación se detectó que el archivo *PacienteMapperImpl.java*, que se genera con cada compilación, no se estaba sobrescribiendo, por lo que hubo que eliminarlo manualmente del directorio del proyecto para que se generase de nuevo al compilar la aplicación.

Sprint Review

Nuevamente tras las dos semanas de duración del sprint, todo el equipo se reúne para mostrar a la *Product Owner* los avances conseguidos.

Al finalizar el sprint anterior hubo que añadir unos cambios a petición de la *Product Owner*. Esto supuso un cambio de alcance en el segundo sprint que según *JIRA* se correspondía con un 5 % del alcance del sprint 2.

La productividad del equipo aumenta también, algo que es habitual en los primeros sprints y que se suele estabilizar a partir del tercero. En el sprint 2 se consiguieron 34 puntos, 9 más que en el sprint anterior, en el que se completaron 25. A pesar de este aumento de velocidad del equipo de desarrollo en este sprint también queda una historia que no se ha podido completar, “HU003 - Alta Recaída”.

A pesar de que, de nuevo hay una historia sin terminar, *JIRA* ya comienza a proporcionar análisis y resultados que resultan interesantes respecto al avance del proyecto. La figura 5.13 muestra la predicción que ofrece la herramienta en lo referente a las fechas. Como se puede apreciar en la imagen, nos da tres fechas de finalización del proyecto, una fecha de finalización del proyecto, y además una optimista y una pesimista. Dado que el proyecto está estimado para seis sprints, se espera acabar el 5 de mayo, por lo que el aumento de velocidad del equipo parece acercarlos a esa fecha.

Retrospectiva

En esta retrospectiva se trataron distintos temas. El más claro es que el equipo fue demasiado optimista a la hora de prever el trabajo que llevarían las diferentes historias relacionadas con el tratamiento, lo que supuso que el trabajo real necesario para completarlas fuese mayor que el estimado. En adelante el equipo será más cauteloso a la hora de hacer estas estimaciones para evitar que vuelvan a quedar historias sin terminar.

Además el equipo está muy contento con el desempeño de la *Product Owner*, que valora el trabajo del equipo y da muy buena realimentación, lo que tiene un impacto positivo de cara al próximo sprint.

A pesar de no haber sido capaces de completar todo el alcance del sprint, el equipo está muy contento con la velocidad conseguida, y esperan ser capaces de aumentarla de cara al tercero.

5. RESULTADOS

Análisis de Versión 1.0

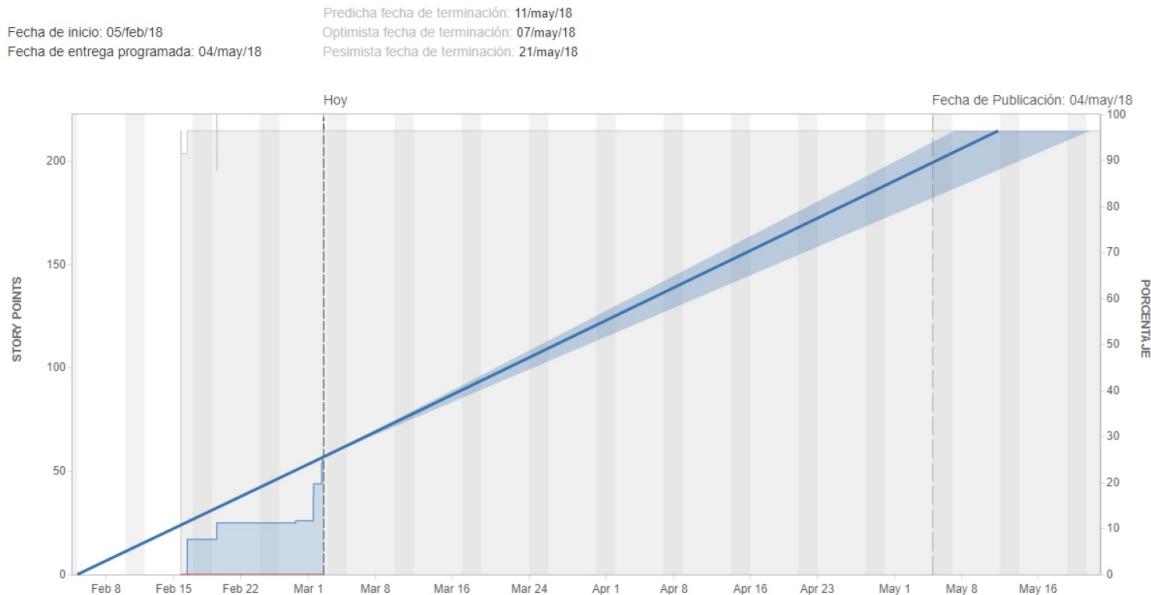


Figura 5.13: Predicción del avance del proyecto

La autora está cada vez más integrada en el equipo y los miembros valoran su aportación al proyecto. Consideran además, que su participación ha sido buena de cara a la *Product Owner* que ha valorado positivamente ver pantallas maquetadas y con la información más clara.

Otro elemento que motiva al equipo es que cada vez se sienten más seguros con la arquitectura que proporciona JHipster y se sienten más cómodos trabajando con Angular5. Esto es importante de cara a los últimos sprints, en los que habrá que hacer más investigación para historias como las de crear el informe o exportar los datos a SPSS.

Por otro lado, el factor más negativo y que obliga al equipo a tener que rehacer trabajo cada sprint, son las limitaciones del JDL. Teniendo claros todos los elementos de la aplicación desde el principio y las diferentes relaciones entre ellos, el JDL es una forma muy sencilla y útil de generar gran parte del código. El problema que detecta el equipo es que, debido a la esencia de SCRUM, el proyecto debe hacerse en incrementos y pueden ir surgiendo cambios en el mismo, lo que obliga al equipo a generar un nuevo JDL al principio de cada sprint; esto supone volver a generar la aplicación y que cada entidad que haya sido modificada vea sobrescritos todos sus archivos.

Sprint 3

El sprint 3 da comienzo el día 5 de marzo y finaliza el día 16 de marzo.

Sprint Planning

En esta ocasión, al igual que en el sprint anterior, quedó una historia sin terminar, *HU003 - Alta Recaída*, así que esta historia será planificada para este sprint. El cuadro 5.4, a continuación, muestra las historias planificadas para este sprint.

Cuadro 5.4: Historias planeadas Sprint 3

Identificador	Resumen	Puntos historia
HU003	Alta Recaída	10
HU007	Alta Seguimiento Médico Actual	3
HU008	Alta Medicación Habitual	6
HU018	Administrador Usuarios	13
HU019	Auditoría	5

- HU003 - Alta Recaída. Permite registrar una o más recaídas para un paciente. Esta tarea quedó por terminar en el sprint anterior.
- HU007 - Alta Seguimiento Médico Actual. Da la posibilidad de dejar constancia en el registro de qué especialistas siguen de forma habitual a un paciente.
- HU008 - Alta medicación Habitual. Se podrá relacionar ciertos fármacos con el tratamiento habitual de un paciente.
- HU018 - Administrador Usuarios. Es necesario que la aplicación gestione diferentes roles para usuario, permitiendo acceder a diferentes funcionalidades dependiendo del rol que tenga asignado el usuario.
- HU019 - Auditoría. Es necesario que quede un registro de las diferentes operaciones que se hagan en la aplicación.

Desarrollo del sprint

En este sprint la autora se encargó de una historia completa, *HU019 - Auditoría*. Para abordar esta historia se hicieron dos propuestas, crear una entidad Auditoría en la que se creará un nuevo registro por cada operación realizada, o, la que finalmente se escogió, aprovechar el módulo de Auditoría que incluye la propia aplicación.

En el módulo de auditoría solamente se registran las autenticaciones, por tanto, era necesario añadir cada una de las operaciones que se realicen sobre pacientes con: la operación realizada, el usuario y la IP de origen de la operación. Para cada entidad dentro de paciente de la que sea necesario registrar la actividad se realizan las siguientes modificaciones. Se usa la entidad Quimioterapia como ejemplo de los pasos a seguir.

5. RESULTADOS

En su archivo *ServiceImpl.java* se importa el repositorio de la auditoría y se indica que la clase extiende del servicio Auditoría. Se crean unas constantes con la información que se mostrará para cada operación. Se añade al constructor y se inicializa. Por último, se indica que el método *save* recibirá también una variable *IP* y en función de si el guardado se trata de una actualización o una creación se crea la entrada de auditoría como creación o edición (5.15). También se añade esto para el método *delete*.

```
1 // ... More code above ... //
2
3 import com.iecisa.oncosup.repository.PersistenceAuditEventRepository;
4
5 // ... More code between ... //
6
7 public class QuimioterapiaServiceImpl extends AuditoriaService implements
8     QuimioterapiaService {
9
10    // ... More code between ... //
11
12    private static final String CREAR_QUIMIOTERAPIA = "CREAR QUIMIOTERAPIA";
13    private static final String EDITAR_QUIMIOTERAPIA = "EDITAR QUIMIOTERAPIA";
14    private static final String ELIMINAR_QUIMIOTERAPIA = "ELIMINAR QUIMIOTERAPIA"
15    ;
16
17
18    public QuimioterapiaServiceImpl(QuimioterapiaRepository
19        quimioterapiaRepository, QuimioterapiaMapper quimioterapiaMapper,
20        QuimioterapiaSearchRepository quimioterapiaSearchRepository,
21        TratamientoRepository tratamientoRepository, RamaRepository
22        ramaRepository, FarmacoQuimioRepository farmacoQuimioRepository,
23        FarmacoAsociadoRepository farmacoAsociadoRepository, AuditoriaRepository
24        auditoriaRepository)
25        FarmacoAsociadoRepository farmacoAsociadoRepository,
26        PersistenceAuditEventRepository persistenceAuditEventRepository)
27        {
28            super(persistenceAuditEventRepository);
29            // ... More code between ... //
30
31        }
32        // ... More code between ... //
33
34    public QuimioterapiaDTO save(QuimioterapiaDTO quimioterapiaDTO, boolean
35        update, String ipAddress) {
36
37        // ... More code between ... //
38        if (update) {
39            createAuditoria(EDITAR_QUIMIOTERAPIA, quimioterapia.getId(),
40                ipAddress);
41        } else {
42            createAuditoria(CREAR_QUIMIOTERAPIA, quimioterapia.getId(), ipAddress
43                );
44        }
45        return result;
46    }
47
48    // ... More code below ... //
49 }
```

Listado 5.15: Cambios en *ServiceImpl* para añadir auditoría

Ahora en el archivo Resource correspondiente, se importa el paquete HttpServletRequest para poder pasar la request a los métodos create, update y delete y extraer la ip desde la que se realiza la operación. El listado 5.16 muestra ésto en el método *deleteQuimioterapia*, el procedimiento es similar para el resto de métodos.

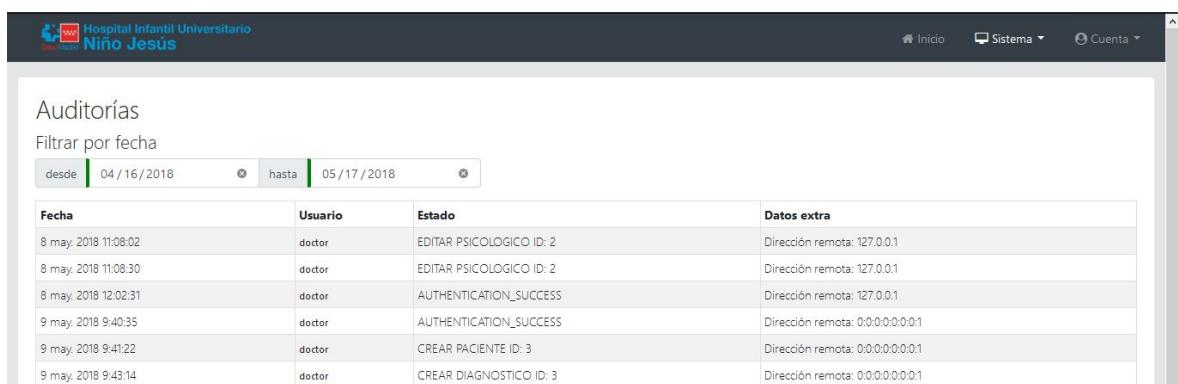
```

1 // ... More code above ... //
2
3 import javax.servlet.http.HttpServletRequest;
4
5 // ... More code between ... //
6
7 /**
8 * DELETE /quimioterapias/:id : delete the "id" quimioterapia.
9 *
10 * @param id the id of the quimioterapiaDTO to delete
11 * @param request the request
12 * @return the ResponseEntity with status 200 (OK)
13 */
14 @DeleteMapping("/quimioterapias/{id}")
15 @Timed
16 public ResponseEntity<Void> deleteQuimioterapia(@PathVariable Long id,
17     HttpServletRequest request) {
18     log.debug("REST request to delete Quimioterapia : {}", id);
19     quimioterapiaService.delete(id, request.getRemoteAddr());
20     return ResponseEntity.ok().headers(HeaderUtil.createEntityDeletionAlert(
21         ENTITY_NAME, id.toString())).build();
22 }
23 // ... More code below ... //

```

Listado 5.16: Cambios en Resource para añadir auditoría

Una vez añadidos estos cambios, si se accede a la auditoría desde la aplicación, se podrá ver un listado como el que se muestra en la figura 5.14 con todas las operaciones realizadas.



The screenshot shows a web application interface for auditing. At the top, there is a header with the hospital's logo and navigation links for 'Inicio', 'Sistema', and 'Cuenta'. Below the header, the title 'Auditorías' is displayed. There is a search bar labeled 'Filtrar por fecha' with fields for 'desde' (04/16/2018) and 'hasta' (05/17/2018). The main content area is a table titled 'Auditorías' with the following columns: Fecha, Usuario, Estado, and Datos extra. The table contains the following data:

Fecha	Usuario	Estado	Datos extra
8 may. 2018 11:08:02	doctor	EDITAR PSICOLOGICO ID: 2	Dirección remota: 127.0.0.1
8 may. 2018 11:08:30	doctor	EDITAR PSICOLOGICO ID: 2	Dirección remota: 127.0.0.1
8 may. 2018 12:02:31	doctor	AUTHENTICATION_SUCCESS	Dirección remota: 127.0.0.1
9 may. 2018 9:40:35	doctor	AUTHENTICATION_SUCCESS	Dirección remota: 0:0:0:0:0:1
9 may. 2018 9:41:22	doctor	CREAR PACIENTE ID: 3	Dirección remota: 0:0:0:0:0:1
9 may. 2018 9:43:14	doctor	CREAR DIAGNOSTICO ID: 3	Dirección remota: 0:0:0:0:0:1

Figura 5.14: Modulo de Auditoría

Por otra parte, la doctora solicitó poder filtrar y ordenar a los pacientes por año de diagnóstico, lo que implicaba añadir en el listado de pacientes una variable nueva con el año del diagnóstico, ya que los pacientes tienen un diagnóstico completo y no se almacena la fecha

5. RESULTADOS

como un valor aparte. Para ello se añadió la nueva variable a la entidad paciente en el JDL, lo que genera automáticamente tanto la variable en el paciente y sus *getter* y *setter*.

Al crear o editar un diagnóstico se requiere que la variable que se acaba de crear se actualice con la fecha del diagnóstico. Esto se hace recuperando el paciente por su diagnóstico y asignándole el año del diagnóstico como se muestra en el listado 5.17.

```
1 // ... More code above ... //
2
3 public DiagnosticoDTO save(DiagnosticoDTO diagnosticoDTO, boolean update) {
4     log.debug("Request to save Diagnostico : {}", diagnosticoDTO);
5     Diagnostico diagnostico = diagnosticoMapper.toEntity(diagnosticoDTO);
6     diagnostico = diagnosticoRepository.save(diagnostico);
7     DiagnosticoDTO result = diagnosticoMapper.toDto(diagnostico);
8     diagnosticoSearchRepository.save(diagnostico);
9
10    Paciente paciente = pacienteRepository.findOne(diagnosticoDTO.getpacienteId
11        ());
11    paciente.setAnnoDiagnostico(Integer.toString(diagnostico.getFechaDiagnostico
12        ().getYear())));
12
13    if (update) {
14        createAuditoria(EDITAR_DIAGNOSTICO, diagnostico.getId(), ipAddress);
15    } else {
16        createAuditoria(CREAR_DIAGNOSTICO, diagnostico.getId(), ipAddress);
17        paciente.setDiagnostico(diagnostico);
18    }
19    pacienteRepository.saveAndFlush(paciente);
20    return result;
21 }
22
23 // ... More code below ... //
```

Listado 5.17: Cambios en DiagnósticoServiceImpl para almacenar el año del diagnóstico en el paciente

Además, al resolver esta petición de la doctora, también se redujo el número de columnas mostradas en el listado de pacientes dejando únicamente las que la doctora consideraba relevantes para identificar pacientes de forma rápida, quedando como muestra la figura 5.15

Sprint Review

En este sprint el equipo finalmente consigue cumplir con todos los puntos planificados, aumentando la productividad en 3 puntos, con una velocidad de 37 puntos de historia. Además, se ha completado la versión MVP, que se correspondía con el estudio preconsulta.

Otro punto que destacar de este sprint es que el equipo ha conseguido hacer una instalación de la aplicación en un entorno de pruebas, pudiendo comprobar así que no hay problemas con la instalación o con el funcionamiento de la aplicación lanzada directamente desde su war (Web Application Resource).

La retroalimentación de la *Product Owner* acerca de los avances en la aplicación siguen

Nombre	Apellidos	Número de Historia Clínica	Teléfono	Año Diagnóstico	
Pepe	Martín	6498418949454	654654984648	2009	Vista Editar Eliminar
Rodolfo	Langostino	1651651651651	065555444	2018	Vista Editar Eliminar
Chicho	Terremoto	3654968464198498	6985483217	2023	Vista Editar Eliminar

Mostrando 1 - 3 de 3 artículos.

Figura 5.15: Listado de pacientes una vez eliminados los campos innecesarios

siendo muy positivos, lo que mantiene al equipo motivado y comprometido.

Se comienzan a gestionar también algunos riesgos. Es necesario que se proporcione al equipo acceso a la red del hospital a través de VPN para poder hacer una instalación en el entorno de certificación y que la oncóloga pueda ir probando la aplicación para encontrar defectos. Este punto es importante, ya que, si no se detectan errores ahora, es posible que aparezcan al final del proyecto, cuando sea complicado gestionarlos.

Por otra parte, *JIRA* a partir del tercer sprint ya empieza a darnos información y algunos gráficos interesantes. La figura 5.16 muestra la estimación del avance por sprint que proporciona la herramienta. Según *JIRA*, a este ritmo de trabajo, la duración del proyecto será de cinco sprints. A pesar de esta previsión el equipo cuenta con que es muy posible que aparezcan más cambios de alcance y se desarrolle seis sprints en total.

Retrospectiva

En este sprint el equipo ha completado el 100 % de los puntos planificados, y esto se refleja en la retrospectiva. Todo el equipo está muy satisfecho con la velocidad de este sprint, que les ha permitido, no solo completar todas las historias, sino también, probar la aplicación y buscar errores en ella. Esto último es muy importante debido a que, a causa del poco presupuesto, no hay equipo de testing en este proyecto; tener tiempo para comprobar que todo funciona como es debido, aunque sea solo haciendo pruebas manuales, ha sido muy útil para encontrar errores y pulir la aplicación.

Se destaca como punto positivo la capacidad que ha tenido el equipo para solucionar los problemas que han ido surgiendo, lo que refleja que es un equipo y que es capaz de autogestionarse.

De nuevo sale a relucir lo contento que está todo el equipo con el desempeño de la *Product Owner*, que este sprint ha estado más centrada que nunca.

5. RESULTADOS

Trabajo por hacer

SUPERV: ONCOSUP 1.0 ▾

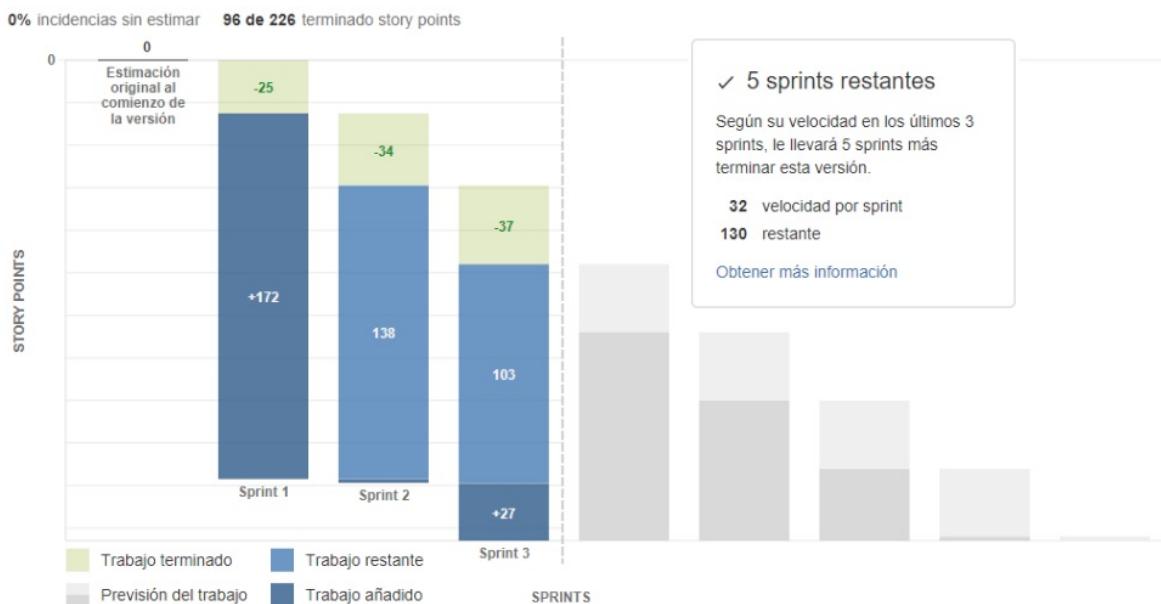


Figura 5.16: Previsión del avance del proyecto

Sin embargo, el JDL y sus limitaciones siguen saliendo como punto negativo del sprint.

Además, el equipo está descontento porque las historias crecen en alcance en los refinamientos, pero esto no queda reflejado después en *JIRA*.

Sprint 4

El sprint 4 comienza el 19 de marzo, y finaliza el 6 de abril. Este sprint dura una semana más para compensar los días de vacaciones que tiene el equipo por Semana Santa.

Sprint Planning

Para este sprint, y tras hablar con la *Product Owner* el equipo considera imprescindible dividir la historia *HU009 - Alta Ítem* en tantas historias como ítems se puedan dar de alta en la aplicación. La estimación de estas historias se hace teniendo en cuenta que gran parte del código será reutilizado. Es por esto que la mayoría de los ítems tienen muy pocos puntos historia.

Prácticamente la totalidad de las historias planificadas (ver cuadro 5.5), forman parte de la consulta y, a pesar de ser una cantidad considerable, una vez acabado el sprint, apenas se habrá completado la mitad de la funcionalidad de la consulta.

- HU005 - Alta Constantes. Permite registrar valores constantes de un paciente como, por ejemplo: el peso, la altura, índice de masa corporal, etcétera.
- HU006 - Alta Exploración física. Da la posibilidad de dejar constancia de cómo ha ido

Cuadro 5.5: Historias planeadas Sprint 4

Identificador	Resumen	Puntos historia
HU005	Alta Constantes	3
HU006	Alta Exploración Física	6
HU009A	Alta Ítem Cardio	2
HU009B	Alta Ítem Neumo	1
HU009C	Alta Ítem Crecimiento	1
HU009D	Alta Ítem Peso	1
HU009E	Alta Ítem Diabetes	1
HU009F	Alta Ítem Tiroides	1
HU009G	Alta Ítem Maduración sexual	1
HU009H	Alta Ítem Nefrourológico	1
HU009I	Alta Ítem Digestivo	1
HU009J	Alta Ítem Hígado	1
HU009K	Alta Ítem Piel	1
HU009L	Alta Ítem ORL	1
HU009M	Alta Ítem Dientes-boca	1
HU009N	Alta Ítem Ojos	1
HU009Ñ	Alta Ítem Hematológico	1
HU009O	Alta Ítem Vacunaciones	2
HU009P	Alta Ítem Autoinmunidad	1
HU009Q	Alta Ítem Neurológico	1
HU010	Alta Pruebas Complementarias que aporta	6
HU011	Alta Pruebas Complementarias solicitadas	6
HU012	Alta Próxima Revisión	2

la exploración física que la oncóloga hace al paciente durante la consulta.

- HU009 - Alta Ítem. Esta historia se divide en tantas historias como ítems hay en la aplicación. Debido a la gran cantidad de historias que han resultado de esta división, se comentan solamente las más relevantes:
 - HU009A - Ítem Cardio. Permite registrar si el paciente tiene una complicación cardiológica. Esta historia tiene más puntos que el resto debido a que es la primera y, de la que se reutilizará la mayor parte del código.
 - HU009(B-Q) excepto HU00O. Al igual que con el anterior, permiten registrar complicaciones sobre el ítem en cuestión además de otros campos relevantes. Estas historias tienen únicamente un punto debido a que se reutiliza el código generado para el primer ítem.
 - HU009O - Ítem Vacunaciones. Este ítem tiene 2 puntos debido a que contiene más campos que los anteriores y más comprobaciones que los demás.
- HU010 - Alta Pruebas Complementarias que aporta. Permite al doctor registrar cualquier tipo de prueba que el paciente le aporte para el registro.

5. RESULTADOS

- HU011 - Alta Pruebas Complementarias solicitadas. Si se solicitan pruebas por parte del oncólogo, aquí podrá registrar los resultados de estas.
- HU012 - Alta Próxima Revisión. Permite reflejar la fecha de la próxima revisión del paciente, para que conste como una cita y aparezca en el informe, que se hará en próximos sprints.

Desarrollo del sprint

En este sprint se animó a la autora a aprender cómo funciona el JDL añadiendo ella misma los ítems como entidades nuevas.

Primeramente, se indican cuáles serán las nuevas entidades identificándolas con la palabra clave *entity*. Se añaden los atributos que tendrá cada entidad como muestra el ejemplo del listado 5.18 para el ítem cardio. Esta operación se realiza con cada uno de los ítems de la consulta.

También se usa la palabra clave *relationship* para indicar las relaciones *OneToOne* (ejemplo: una consulta tiene un ítem cardio) y *OneToMany* (ejemplo: una consulta tiene una o varias pruebas aportadas)

```
1 // ... More code above ... //
2
3 entity Cardio {
4     fechaUltimaRevision LocalDate required,
5     complicacion Boolean required,
6     desComplicacion String,
7     tratamiento Boolean required,
8     desTratamiento String,
9     fechaInicioTratamiento LocalDate
10 }
11
12 // ... More code between... //
13 relationship OneToOne {
14     // ... More code between ... //
15     Consulta{cardio} to Cardio,
16     // ...
17     Consulta{neurologico} to Neurologico
18 }
19
20 // ... More code between... //
21
22 relationship OneToMany {
23     // ... More code between ... //
24     Consulta{pruebaAportada} to PruebaAportada{consulta},
25     Consulta{pruebaSolicitada} to PruebaSolicitada{consulta}
26 }
27
28 // ... More code below ... //
```

Listado 5.18: Añadir nuevas entidades y sus relaciones en el JDL

De la mano de la tarea anterior, se enseñó a la autora como crear historias y tareas en *JIRA* para que ella misma pudiera encargarse de crearlas para cada uno de los ítems y comenzar a

aprender más sobre la herramienta.

La creación de historias en *JIRA* es sencilla. Estando en el espacio del proyecto, al hacer clic en la opción crear incidencia aparece una ventada de diálogo (figura 5.17) en la que podremos definir, entre otros parámetros, el tipo de incidencia (en este caso “story”), su nombre, el sprint al que pertenece, quién es el responsable, etcétera. Una vez creadas las historias, a través de un procedimiento similar, podemos acceder a cada una de ellas y añadirles las tareas, en las que además se añade la estimación de tiempo que requerirá llevarlas a cabo.

Figura 5.17: Creación de una incidencia en JIRA

Debido a que en este sprint se terminó el trabajo estimado antes de lo previsto, se decidió añadir una historia más al trabajo del sprint, *HU013 - Alta efecto tardío*. El trabajo de esta

5. RESULTADOS

tarea se repartió entre los miembros del equipo, tocándole a la autora maquetar todas las pantallas y eliminar los botones de edición, creación y eliminación. La eliminación de los botones se debe a que un efecto tardío se crea solamente cuando en un ítem se marca la casilla de “complicación”, siendo un elemento que no es posible crear de otra manera.

La maquetación se realizó de misma manera que se explicó en la sección 5.4.2 y respecto a los botones, simplemente se eliminó el código HTML correspondiente tanto de la pantalla del listado como de la pantalla de detalle.

Sprint Review

Al finalizar el sprint 4, el equipo no sólo ha completado todos los puntos planificados, sino que han sido capaces de aumentar el alcance del sprint añadiendo una historia más, *HU013 - Alta efecto tardío* de 11 puntos historia. Los puntos totales completados en el sprint 4 son 54. La adición de esta historia en el sprint supone un aumento del 26 % del alcance, por lo que la productividad de este sprint ha sido muy buena.

Un buen gráfico que ofrece *JIRA* para ver esto reflejado es el *Burn Down Chart*, en el que se puede ver como el equipo va cerrando puntos y se van completando las distintas historias. Por gráficos como este es importante que las historias tengan un tamaño razonable, de manera que el equipo las pueda ir cerrando durante el desarrollo del sprint en lugar de todas de golpe al final. Esto facilita el seguimiento del desarrollo del sprint.

En la figura 5.18 se puede observar que el equipo ha ido completando historias a buen ritmo durante todo el sprint, y que el día 2 de abril decidieron ampliar el alcance añadiendo una historia más. También se refleja en los días 3 y 5 de abril que se reabrieron dos historias para solucionar errores que se detectaron mientras se hacían pruebas ágiles en la aplicación.

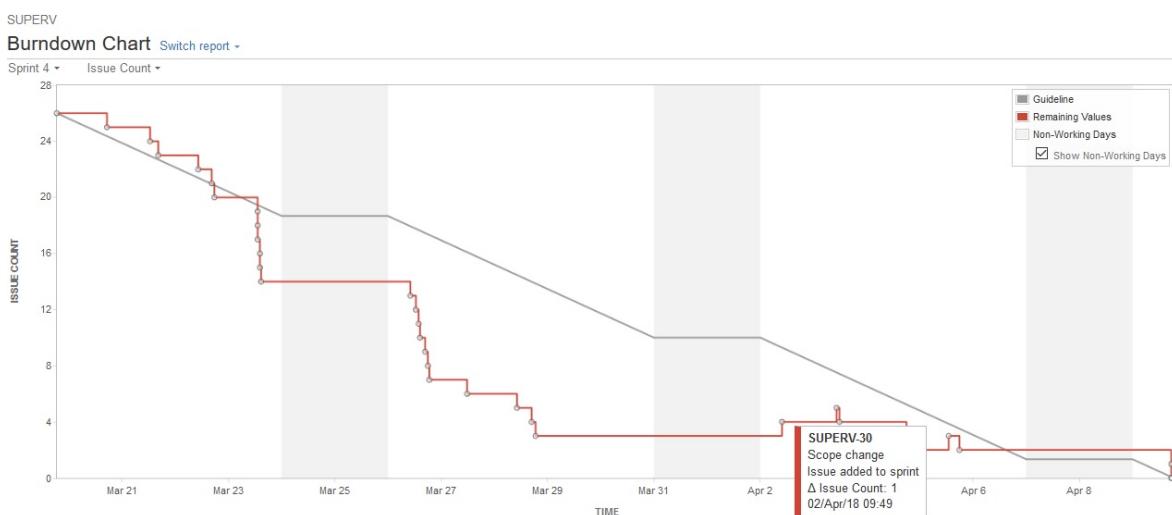


Figura 5.18: Burn Down Chart del Sprint 4

En este punto del proyecto el equipo aún no ha sido capaz de solucionar el problema de

acceso a través de VPN al entorno del hospital, por lo que se propone mitigar este riesgo realizando el próximo *Sprint Review* en el propio Hospital, pudiendo así hacer la instalación *in situ* y permitiendo así que la oncóloga pueda probar la aplicación y dar de alta defectos.

En la figura 5.19, se muestra cómo la velocidad en este último sprint ha impactado positivamente en la fecha estimada de entrega, estando ahora mucho más acotada la fecha real prevista.

Análisis de Versión 1.0

Fecha de inicio: 05/feb/18
Fecha de entrega programada: 04/may/18

Predicha fecha de terminación: 07/may/18
Optimista fecha de terminación: 03/may/18
Pesimista fecha de terminación: 09/may/18

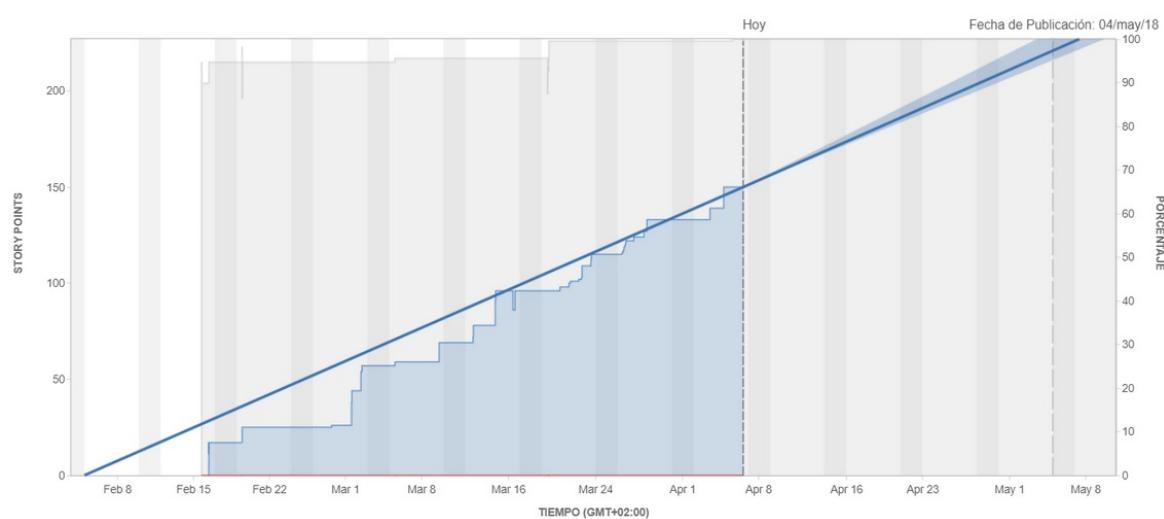


Figura 5.19: Predicción del avance del proyecto

Por otra parte, la figura 5.20 muestra que *JIRA* considera que el proyecto se seguirá completando en seis sprints. Desafortunadamente, el equipo considera que esta previsión no es fiable, ya que la reutilización de código en los ítems ha permitido alcanzar mucho más rápido y completar muchos más puntos de los esperado. Por tanto, prefieren ser prudentes y seguir contando con un sexto sprint para cerrar el proyecto.

Retrospectiva

En este sprint el equipo en general está contento con la velocidad obtenida, pero son conscientes de que, debido a la reutilización de código en los ítems y a que debido a los días de vacaciones el sprint ha durado tres semanas, la velocidad no ha sido realista. De cualquier forma, están contentos con haber sido capaces de ampliar el alcance del sprint y reducir el trabajo restante para los dos próximos sprint.

El equipo sigue notando la falta de un tester y un plan de pruebas que les permita controlar que las diferentes funcionalidades de la aplicación funcionan correctamente. La velocidad de este sprint les ha permitido hacer ellos mismo pruebas exploratorias y corregir algunos defectos, pero les preocupa la alta probabilidad de que haya otros errores que puedan hacer

5. RESULTADOS

Trabajo por hacer

SUPERV: ONCOSUP 1.0 ▾

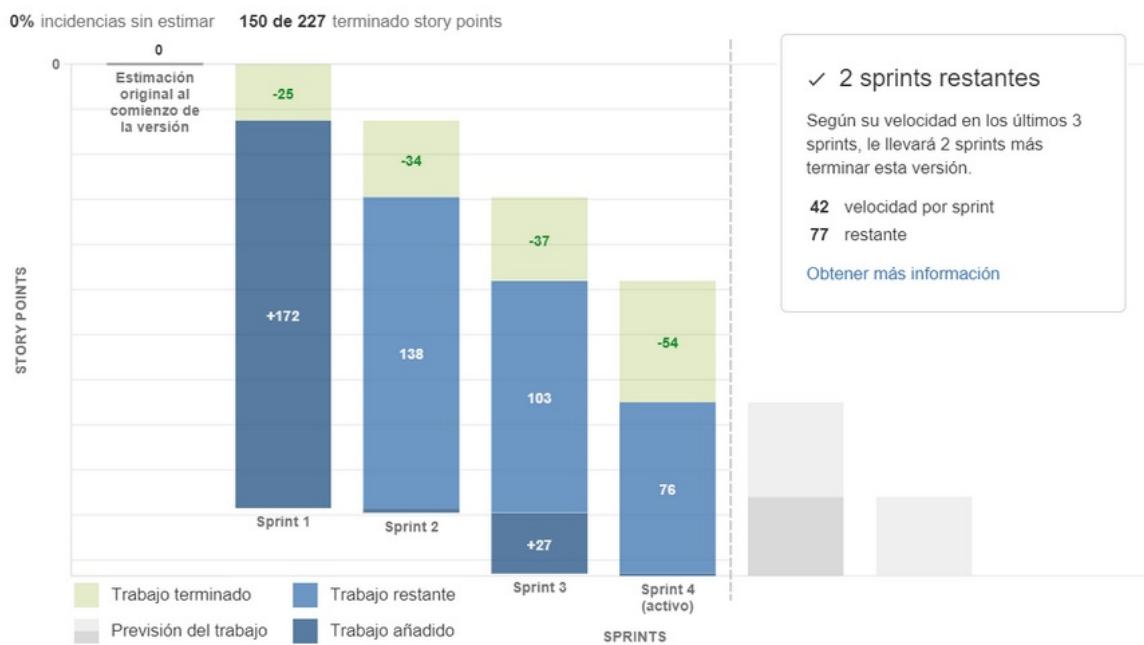


Figura 5.20: Previsión de sprint restantes

que la aplicación no funcione como es debido.

Por otra parte, en el *Sprint Review* la *Product Owner* ha detectado que los efectos tardíos no se generan como ella quiso hacer entender al equipo, lo que implicará rehacer eso en cada ítem completado y asegurarse de que los nuevos se creen correctamente. El equipo detecta en este suceso un mal refinado de las historias del sprint que implica volver a codificar trabajo ya completado. La autora sugiere como medida para evitar que esto ocurra tratar de llevar bocetos o *mockups* a los refinamientos para tratar de evitar confusión entre lo que el *Product Owner* quiere transmitir y lo que el equipo entiende. Esto permitiría al *Product Owner* hacerse una mejor idea de cómo será la aplicación viendo una simulación de cuál sería el funcionamiento de ésta y detectar errores antes de que se codifique. Todo el equipo considera que es una buena idea y se decide dar de alta una medida a tomar en *JIRA* para que quede reflejado.

Sprint 5

La actividad correspondiente al sprint 5 comienza el día 9 de abril y acaba el día 20 del mismo mes.

Sprint Planning

En este sprint se planifican, entre otras, todos los ítems restantes a falta de los que no han sido refinados aún. Además la *Product Owner* ha decidido dividir la historia del ítem

“musculoesquelético” en dos distintas, quedando dos historias nuevas: *HU009U1 - Alta Ítem Muscular* y *HU009U2 - Alta Ítem Esquelético*.

Cuadro 5.6: Historias planeadas Sprint 5

Identificador	Resumen	Puntos historia
HU009R	Alta Ítem Neuropsicológico	1
HU009U1	Alta Ítem Muscular	1
HU009U2	Alta Ítem Esquelético	1
HU009V	Alta Ítem Capacidad Física	1
HU009W	Alta Ítem Sexualidad	1
HU009Y	Alta Ítem Fertilidad	1
HU014	Alta Recomendaciones	3
HU015	Visualizar Informe	11
HU022	Administrar Recomendaciones	6
HU026	Eliminar Información del informe	10

- HU009(R-Y) - Alta distintos Ítems. Al igual que se explicó en el sprint anterior (Sección 5.7.1) deben permitir dar de alta diferentes ítems. Aunque con pequeñas diferencias en cuanto a sus atributos, los ítems son muy similares y también habrá bastante reutilización de código.
- HU014 - Alta Recomendaciones. Dará posibilidad de seleccionar una o varias recomendaciones de las ya creadas para asignarlas a un paciente en una consulta. Además, permitirá introducir una recomendación personalizada si es necesario.
- HU015 - Visualizar Informe. Permitirá visualizar una vista previa del informe que se le entregará al paciente.
- HU022 - Administrar Recomendaciones - El doctor podrá dar de alta diferentes recomendaciones para poder asignarlas más tarde a la consulta de un paciente.
- HU026 - Eliminar Información del informe. El doctor podrá abrir un diálogo de edición que le permitirá eliminar o añadir la información que considere oportuna para el informe de un paciente.

Desarrollo del sprint

En este sprint se aprovecha que la autora ha ido adquiriendo conocimientos suficientes para encargarle todos los ítems, de los que esta vez se encargará ella sola. Esto implica que tendrá que encargarse de añadir cada nuevo ítem al JDL, importarlo a la aplicación, y realizar todos los cambios necesarios; esto incluye, añadir todo lo relacionado con la auditoría, añadirlo a la consulta, añadir sus métodos de creación, edición y eliminación a ésta, y realizar además todos los ajustes necesarios en su *detail* y su *dialog*.

Debido a que algunas de estas tareas ya se han explicado en secciones anteriores, ésta se centrará en hablar de la maquetación de los diálogos. Aunque ya se habló algo sobre

5. RESULTADOS

maquetación en la sección 5.4.2, se tratará de explicar en detalle cómo se decidió organizar la información de los diálogos tratando de priorizar la usabilidad y la experiencia del usuario.

En primer lugar, la figura 5.21 muestra cómo se ve la ventana de diálogo tal y como la genera JHipster. A pesar de que JHipster genera una aplicación totalmente maquetada, se puede apreciar que hay algunas cosas que son muy mejorables. Los *check box* están separados de su etiqueta, lo cual puede confundir al usuario. La cantidad de campos que aparecen es demasiada, esto además de poder abrumar la vista del usuario, obliga a tener que desplazar la pantalla para poder ver todos los campos.

Al maquetar estas pantallas se trató de simplificar la información y agruparla de forma que fuese más fácil de comprender e identificar para el usuario. Para ello, se aprovecharon los *check box* para controlar qué información se mostraría en caso de marcarlos. En este ejemplo en concreto, los grupos son: complicación, tratamiento y deporte habitual; quedando el campo alteración como un *check box* independiente. La manera en que se hace esto con Angular5 es haciendo uso de `*ngIf` al que indicaremos que muestre los campos cuando su *check box* correspondiente esté marcado como se aprecia en el listado 5.19.

```
1  <!-- More code above -->
2  <div class="modal-body">
3      <div class="form-group">
4          <input type="checkbox" class="check" name="complicacion" id="field_complicacion" [(ngModel)]="capacidadFisica.complicacion" />
5          <label class="form-control-label" jhiTranslate="oncosupApp.capacidadFisica.complicacion" for="field_complicacion">Complicacion</label>
6      </div>
7      <div class="form-group" *ngIf="capacidadFisica.complicacion">
8          <label class="form-control-label" jhiTranslate="oncosupApp.capacidadFisica.desComplicacion" for="field_desComplicacion">Des Complicacion</label> <input type="text" class="form-control" name="desComplicacion" id="field_desComplicacion" [(ngModel)]="capacidadFisica.desComplicacion" />
9      </div>
10     <div class="form-group" *ngIf="capacidadFisica.complicacion">
11         <label class="form-control-label" jhiTranslate="oncosupApp.capacidadFisica.fechaInicioComplicacion" for="field_fechaInicioComplicacion">Fecha Inicio Complicacion</label>
12         <div class="input-group">
13             <input id="field_fechaInicioComplicacion" type="text" class="form-control" name="fechaInicioComplicacion" ngbDatepicker # fechaInicioComplicacionDp="ngbDatepicker" [(ngModel)]="capacidadFisica.fechaInicioComplicacion" [minDate]="minDate" required /> <span class="input-group-append">
14                 <button type="button" class="btn btn-secondary" (click)="fechaInicioComplicacionDp.toggle()">
15                     <i class="fa fa-calendar"></i>
16                 </button>
17             </span>
18         </div>
19     </div>
20 </div>
21 </div>
```

```
23 | <hr *ngIf="capacidadFisica.complicacion">  
24 |  
25 |   <!-- More code below-->
```

Listado 5.19: Control de campos que se muestran según el *check box* marcado

Crear o editar Capacidad Física

Complicación

Descripción Complicación

Fecha Inicio Complicación

Tratamiento

Descripción Tratamiento

Fecha Inicio Tratamiento

Alteración

Deporte Habitual

Frecuencia

Número Horas Semana

Tipo Actividad Física

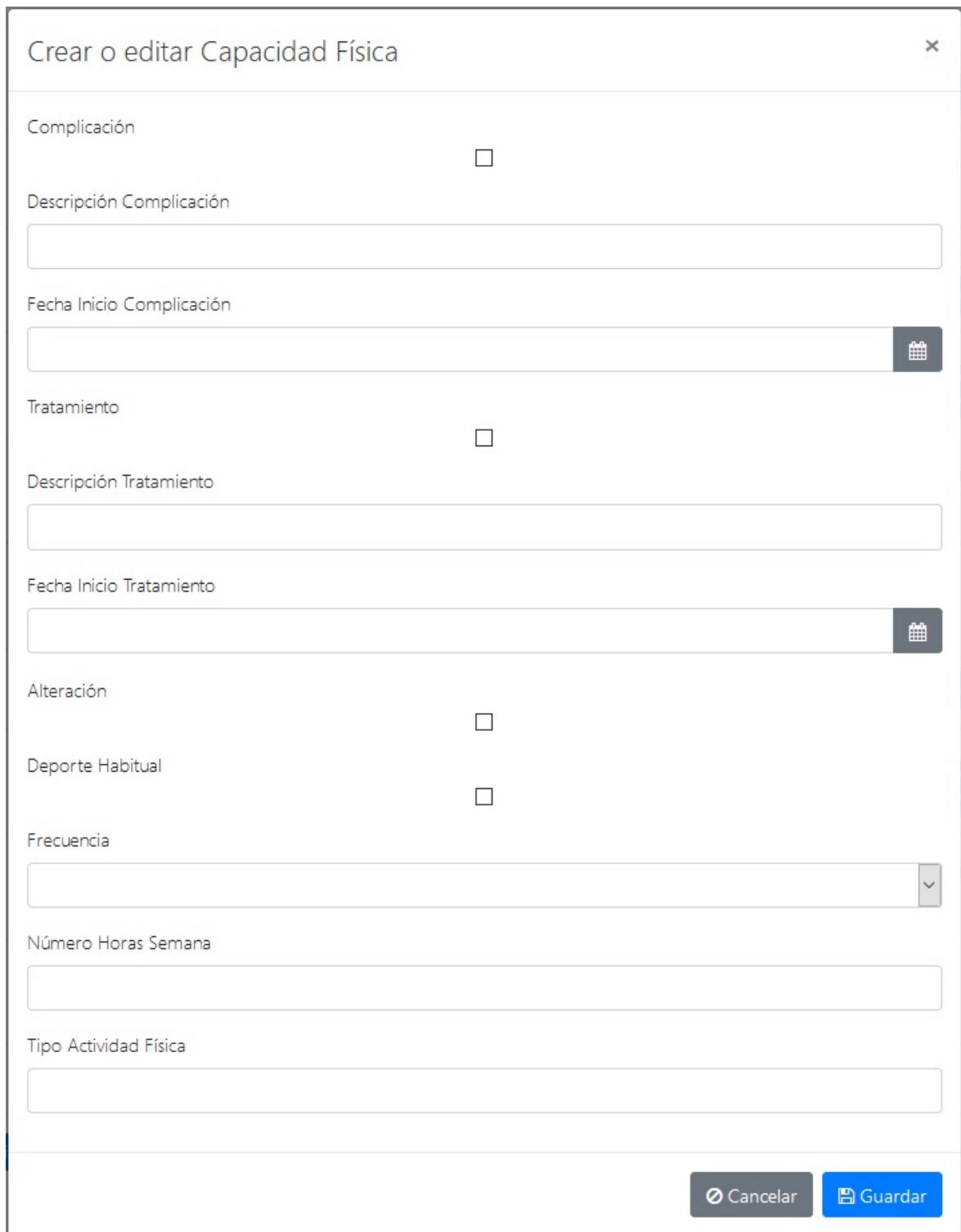


Figura 5.21: Diálogo de un ítem sin maquetar

5. RESULTADOS

La apariencia del diálogo tras los cambios es la que muestra la figura 5.22. Los campos que se muestran se han reducido a cuatro, que se desplegarán si el usuario marca alguno. Esta presentación es más legible para el usuario.



Figura 5.22: Diálogo de un ítem maquetado

Cuando el usuario marque un *checkbox* se desplegarán los campos como muestra la figura 5.23, además, al desplegar un grupo de campos, se añade también un separador para que sea más claro dónde acaban.

Sprint Review

El resultado de este sprint ha sido bueno debido de que el equipo fue precavido y no se dejó llevar por la alta velocidad obtenida en sprints anteriores, eligiendo menos historias debido a su mayor complejidad. Se ha completado toda la funcionalidad planeada para este sprint, pasando al siguiente sprint tan solo unas mejoras menores. Esto puede verse reflejado en la figura 5.24, dónde se aprecia que al acabar el sprint todavía quedaban algunos puntos por cerrar. Es interesante ver cómo, comparado con la figura mostrada en la sección 5.7.3, el equipo ha ido mucho más apurado en este sprint yendo por encima de la línea de progreso ideal que marca *JIRA*.

Consultando la previsión de sprint restantes (figura 5.25) se puede ver que *JIRA* estima un sprint más para completar todos los puntos de historia restantes, coincidiendo con las previsiones iniciales del equipo.

Retrospectiva

Nuevamente el equipo está muy contento con la reacción de la *Product Owner* respecto a las soluciones que han aportado. Esto se debe a que, a la hora de presentar el informe, hubo un debate sobre cómo sería mejor hacerlo, y se llegó a la conclusión de que lo ideal es

Crear o editar Capacidad Física

x

 Complicación

Descripción Complicación

Fecha Inicio Complicación

 Tratamiento Alteración Deporte Habitual Cancelar Guardar

Figura 5.23: Diálogo de un item maquetado al desplegar un grupo de campos

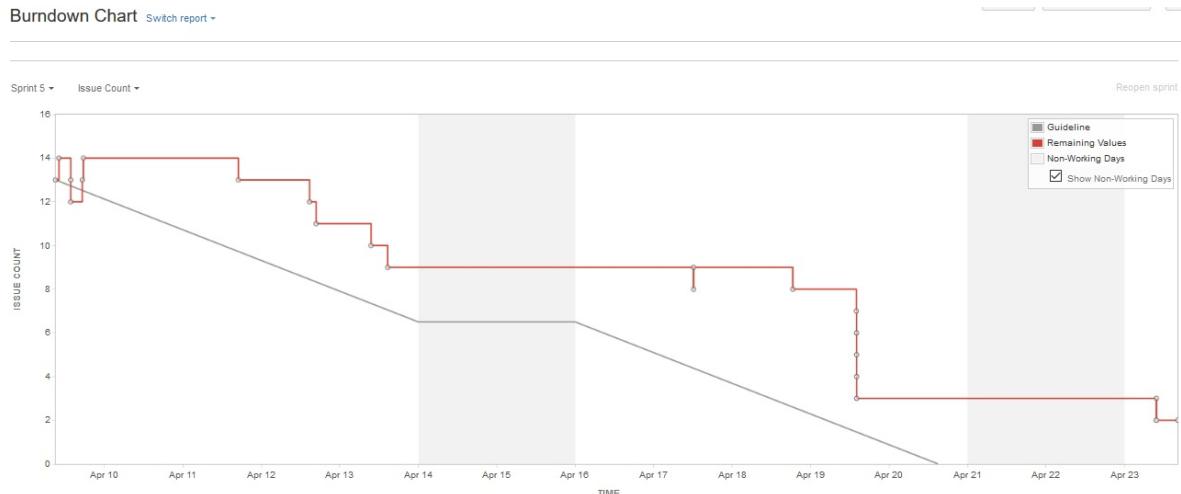


Figura 5.24: Burn Down Chart. Sprint 5

continuar con la misma tónica de toda la aplicación, el informe sería una entidad, su detalle serviría como previsualización y su diálogo serviría para añadir o quitar información. Tan solo sería necesario añadir un botón para la impresión, tarea que se acometerá en el próximo sprint.

Algo con lo que el equipo está muy contento es con el trabajo de Carlos, miembro del

5. RESULTADOS

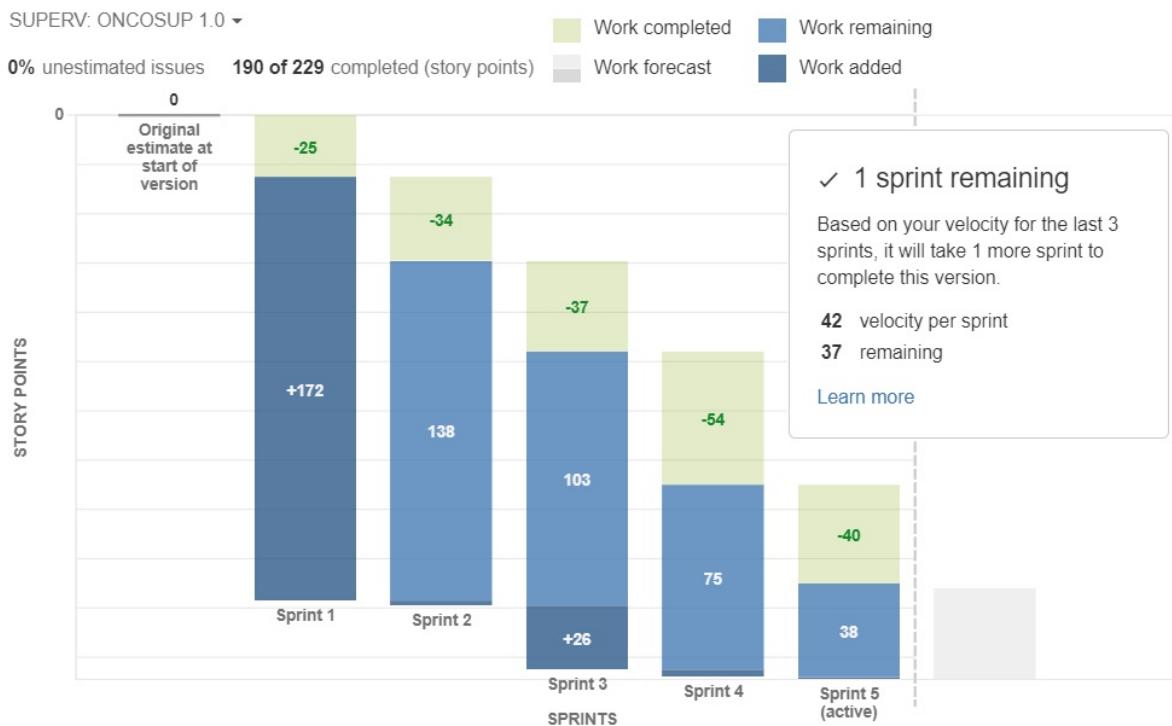


Figura 5.25: Previsión de sprint restantes

equipo que se ha encargado de preparar todo lo relacionado con la preparación de la instalación. Por otra parte, el equipo está satisfecho con haber podido realizar la instalación, a pesar de haberlo hecho de forma local en el ordenador de trabajo de la Dra. Herrero, ya que ahora será posible que la doctora pruebe la aplicación y les reporte los errores de funcionamiento que detecte.

Además, el equipo se da cuenta de lo grande que es la aplicación y les agrada ver que a pesar de ser pocos miembros están siendo capaces de sacarla adelante.

Por otro lado, el equipo sigue preocupado por el hecho de que no haya un plan de pruebas, y los errores que la doctora notifique solo podrán ser corregidos durante el siguiente sprint. Por esta misma razón al equipo le gustaría que hubiera posibilidad de ampliar el presupuesto del proyecto para poder acabar la aplicación.

Finalmente, el Hospital aún no ha dado acceso a la VPN, y la instalación de versiones en el entorno de certificación del Hospital no va a poder hacerse.

Sprint 6

El sexto y último sprint comienza el día 23 de abril y acaba el día 4 de mayo.

Sprint Planning

En este último sprint se planean todas las historias de usuario restantes en el *Product Backlog*.

El cuadro 5.7 muestra un total de puntos historia menor al que se podía ver en la figura 5.25, en la sección 5.8.3, esto se debe a que debido a que el equipo no considera que fuese capaz de completar toda la funcionalidad, se decidió eliminar una historia que la *Product Owner* no consideraba vital para la aplicación.

Cuadro 5.7: Historias planeadas Sprint 6

Identificador	Resumen	Puntos historia
HU000	Login	7
HU009S	Alta Ítem Psicológico	3
HU009T	Alta Ítem Social	3
HU009U	Alta Ítem Información	1
HU009V	Alta Ítem Segundo Tumor	1
HU017	Imprimir Informe	7
HU024	Exportación a SPSS	5
HU025	Filtro Exportación SPSS	3

- HU000 - Login. Implementar una autenticación de usuarios que esté integrada con el LDAP del hospital.
- HU009(S-V) - Alta distintos Ítems. Se debe permitir dar de alta los diferentes ítems. En este caso el ítem psicológico y el ítem social tienen tres puntos debido a que la complejidad de éstos es mayor que la del resto.
- HU017 - Imprimir Informe. Permitirá imprimir el informe.
- HU024 - Exportación a SPSS - Se trata de una opción que permite exportar toda la información de la base de datos de la aplicación a una hoja de cálculo que permitirá a la Dra. Herrero importar la información la herramienta SPSS.
- HU025 - Filtro Exportación SPSS. Un filtro que permitirá elegir qué datos se desea exportar.

Desarrollo del sprint

En este último sprint la autora tuvo que encargarse, además de los ítems restantes, de implementar la parte del informe referente a la impresión y algunos aspectos estéticos sobre la visualización e impresión de éste.

Para la impresión se decidió, por temas de tiempo, recurrir a la opción que al equipo le pareció más sencilla y que ofrecía un buen resultado: utilizar la herramienta de impresión del navegador. Esto permite que el navegador haga una impresión de la página usando los estilos que se han añadido en el .css, con lo que sólo es necesario maquetar la pantalla una vez y no dos, una para la impresión y otra para la visualización. Hacer uso de esta herramienta es sencillo, al hacer clic en el botón imprimir que se ha añadido al final de la previsualización del informe, se hace una llamada al método *print()* en el que se invocará la impresión del navegador (listado 5.20).

5. RESULTADOS

```
1 // ... More code above ... //
2
3 print() {
4     this.originalTitle = window.document.title;
5     this.curURL = window.location.href;
6     history.replaceState(history.state, "", "/");
7     window.print();
8     history.replaceState(history.state, "", this.curURL);
9     window.document.title = this.originalTitle;
10    }
11 }
```

Listado 5.20: Implementación del método print()

Lo único que se necesita es que no imprima ni los botones de la parte baja de la página, ni el *header* ni el *footer*. Para ello, en el archivo .css del informe (listado 5.21), se hace uso de la etiqueta *@media print* en la que se indica qué debe ocurrir cuando se va a imprimir la página. Se crea una clase que no se va a mostrar en la pantalla de impresión y que se aplicará al contenedor de los botones y además se indica que no se muestren tampoco ni el *header* ni el *footer*. La etiqueta *@media screen* sirve para crear estilos que se aplican solo cuando se está en la pantalla del navegador. En la pantalla se desea que se muestre el nombre del paciente, pero al imprimir tan solo se muestra un título.

```
1 /* ... More styles above ... */
2
3 @media print{
4     .doNotPrint{display:none;}
5     #Header, #Footer {display:none;}
6 }
7
8 @media screen{
9     .doNotShow{display:none}
10}
```

Listado 5.21: Estilos específicos de la impresión

El resultado de estos estilos se puede ver en las figuras a continuación. La figura 5.26 muestra cómo se ve la pantalla del informe antes de imprimir, mientras que la figura 5.27 muestra el informe tal y como se verá al imprimirlo.

 Hospital Infantil Universitario
Niño Jesús

[Inicio](#) [Registro](#) [Administración](#) [Cuenta](#)

Informe - Chicho Terremoto

Fecha Consulta 11 may. 2018	Nombre Completo Chicho Terremoto	
Fecha Nacimiento 4 may. 2018	Sexo Varón	Número de Historia Clínica 3654968464198498

Diagnóstico
Fecha Diagnóstico: 11/05/2023. Patología Oncológica: Tumor sólido. Cie 10: II16 - Tumores (neoplasias) in situ. Cie 10 Secundario: D02 - Carcinoma in situ del sistema respiratorio y del oído medio.

Protocolo Estandarizado
No

Fecha Inicio Tratamiento 19 may. 2018	Fecha Fin Tratamiento 15 may. 2020
---	--

Seguimiento Habitual
Sí

Especialidades
Especialidad: Neumo, motivo: dagadfg, frecuencia: adfgadfdf, centro sanitario: gsgdsgfds, fecha de inicio: 13/05/2011.

Tratamiento Habitual
No

Cardio
Sí

Complicación: Sí, descripción de la complicación: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, , fecha de inicio de la complicación: 13/05/2018. Seguimiento: Sí, fecha de última revisión: 03/05/2018. Tratamiento: Sí, descripción del tratamiento: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, , fecha de inicio del tratamiento: 06/05/2018.

Maduración Sexual
Sí

Complicación: Sí, descripción de la complicación: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, , fecha de inicio de la complicación: 18/05/2018. Seguimiento: Sí, Tratamiento: No.

Hígado
Sí

Complicación: Sí, descripción de la complicación: sgfdgdsdfgagfda, fecha de inicio de la complicación: 17/05/2018. Seguimiento: Sí, fecha de última revisión: 18/05/2005. Tratamiento: No.

Neurológico
Sí

Complicación: Sí, descripción de la complicación: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, , fecha de inicio de la complicación: 10/05/2018. Seguimiento: Sí, fecha de última revisión: 10/05/2018. Tratamiento: Sí, descripción del tratamiento: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, , fecha de inicio del tratamiento: 11/05/2018.

Capacidad Física
Sí

Complicación: No, Tratamiento: Sí, descripción del tratamiento: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, , Alteración: No. Deporte Habitual: Sí, Frecuencia: Más de tres días por semana. Número Horas Semana: 6. Tipo Actividad Física: Lorem ipsum dolor sit amet, consectetur adipiscing elit..

Sexualidad
Sí

Complicación: No, Tratamiento: No, Pareja: Sí, Relaciones Sexuales: Sí, Problemas Relaciones Sexuales: Sí, Descripción Problemas Relaciones Sexuales: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida..

Fertilidad
Sí

Complicación: No, Seguimiento: No, Tratamiento: Sí, descripción del tratamiento: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, , fecha de inicio del tratamiento: 24/05/2018. Uso Técnicas Preservación: Sí, Técnicas Preservación: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, , Deseo Hijos: Sí, Preocupación: Sí, Número Hijos: 0, Uso Técnicas Reproducción: Sí, Técnicas Reproducción: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, .

Social
Sí

Nivel de Estudios: Universitaria, ¿Repitió curso? Sí, ¿Cuándo repitió? Después de la enfermedad. Situación Laboral: Activo. Residencia: Vive solo/Independiente. Prestación Enfermedad: Sí, Descripción Prestación Enfermedad: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet, .

Neumo
No

Endocrino Crecimiento No	Endocrino Peso No	Endocrino Diabetes No
------------------------------------	-----------------------------	---------------------------------

Endocrino Tiroides
No

Nefrourológico No	Digestivo No	Piel No
-----------------------------	------------------------	-------------------

ORL
No

Dientes-Boca No	Ojos No	Hematológico No
---------------------------	-------------------	---------------------------

Vacunas
No

Autoinmunidad No	Neuropsicológico No	Muscular No
----------------------------	-------------------------------	-----------------------

Esquelético
No

Drogas No	Segundo Tumor No	Psicológico No
---------------------	----------------------------	--------------------------

Información
No

Observaciones
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam luctus neque a malesuada sodales. Fusce dignissim risus eu ipsum dictum, maximus cursus lectus gravida. Pellentesque a purus tempus, tincidunt quam id, lacinia massa. Cras et cras amet.

[Volver](#) [Editar](#) [Imprimir](#)

Figura 5.26: Previsualización del Informe

5. RESULTADOS

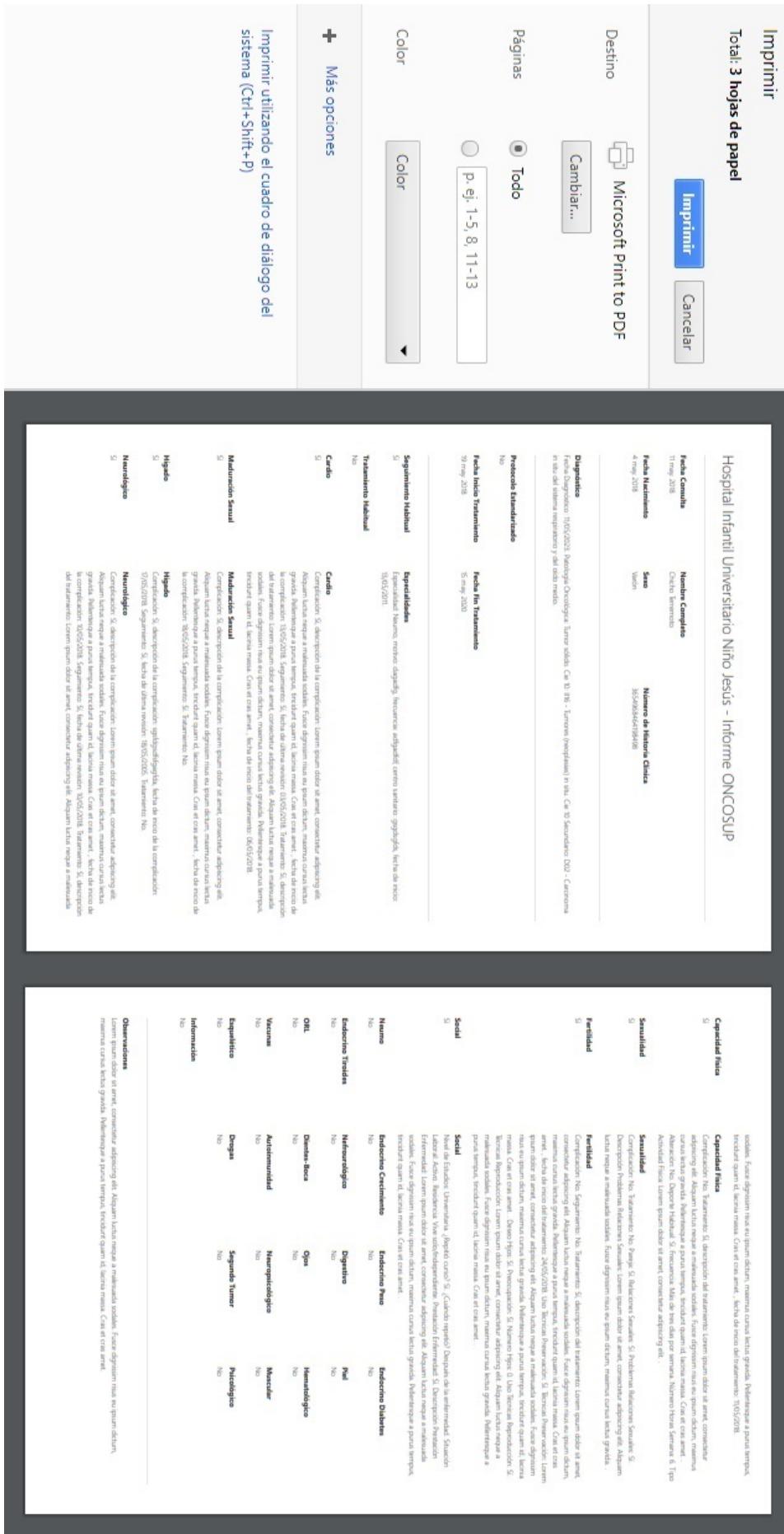


Figura 5.27: Vista de impresión del informe

Sprint Review

En el sprint final del proyecto el equipo no ha podido completar todos los puntos planificados. El equipo no ha recibido la información necesaria para la integración de la aplicación con el LDAP, de manera que la historia *HU000 - Login* ha quedado sin completar por causas ajenas al equipo.

Por otra parte, *JIRA* indica que los cambios añadidos por la Dra. Herrero sobre algunas funcionalidades ya implementadas y la complejidad añadida a los ítems restantes han supuesto un incremento del alcance de un 14 %, alcance que sí se ha llegado a completar.

Si bien es cierto que la productividad del equipo ha sido menor que en los sprints anteriores, se debe a que al ser una versión final se ha dedicado más tiempo a corrección de defectos y por otra parte, el sprint 3 duró una semana más debido a las vacaciones. Esto, sumado a que una de las tareas planificadas ha sido imposible de completar, es la causa de este descenso en la productividad del equipo.

Algunos riesgos de los mencionados en la sección 5.1.3, como son: *R2 - Entornos no disponibles*, *R3 - Acceso al hospital* y *R4 - No se dispone de información sobre el acceso a LDAP* han acabado materializándose a pesar de los esfuerzos del equipo en mitigarlos, lo que ha impactado de forma negativa en el producto final.

Todo esto, unido a las restricciones económicas del proyecto ha provocado que el equipo no haya conseguido hacer entrega de la versión 1.0 tal y como estaba previsto, quedando alcance y funcionalidades que, para ser completadas, deberían ser acometidas en futuras fases o versiones de la aplicación.

Retrospectiva

Con el desarrollo del proyecto ya acabado, el equipo se reúne para hacer la última retrospectiva. Esta retrospectiva no solo es sobre el sprint 6, sino que se analiza todo el proyecto y se trata de extraer un balance global.

Comenzando por lo positivo, todo el equipo está orgulloso de haber sido capaces de finalizar el proyecto a pesar del poco tiempo del que se disponía. Están muy contentos de que el primer proyecto hecho con JHipster haya salido tan bien, habiendo podido completar toda la funcionalidad en tan poco tiempo. Les gusta haber podido aprender tecnologías nuevas como Angular 5. Agradecen a la Dra. Herrero su participación como *Product Owner* que, consideran, ha sido impecable.

En lo referente a las cosas que se deberían haber hecho mejor, les gustaría que hubiera habido mayor implicación por parte del departamento de informática del hospital, ya que ha supuesto la materialización de algunos de los riesgos detectados en la *Inception*. Han echado en falta mayor apoyo institucional que diera visibilidad al proyecto. Las fuertes restricciones de presupuesto no han permitido dar ni siquiera un período de soporte para ONCOSUP.

5. RESULTADOS

Por último, el equipo expone que desearía respecto a este y otros proyectos basándose en lo vivido durante el desarrollo. Desearían que hubiera una segunda fase para ONCOSUP, para poder refinar la aplicación y añadir toda la funcionalidad que tuvo que descartarse en la *Inception*. ONCOSUP ha obligado al equipo a valorar los recursos que tienen y creen que es algo que deberían tener en cuenta en próximos proyectos. El proyecto y la problemática que trataba ha sido una gran motivación para todos los miembros del equipo, y les gustaría que la motivación sea la misma en futuros proyectos. Esperan que ONCOSUP salga adelante, que cumpla el objetivo con el que fue concebida y no acabe olvidada en un cajón, sino que se expanda su uso y ayude a mejorar las vidas de los supervivientes y a salvar otras.

Capítulo 6

Conclusiones

COMO conclusión general el resultado del proyecto no solo ha sido bueno, sino que teniendo en cuenta las fuertes restricciones de presupuesto, las limitaciones debido a la falta de información y conocimientos de algunas de las herramientas, ONCOSUP ha sido realmente un éxito. JHipster, a pesar de las complicaciones al inicio del proyecto por el desconocimiento de la herramienta, ha sido de grandísima ayuda, facilitando y agilizando el trabajo de forma excepcional. El inicio del proyecto fue algo lento, por la necesidad de aprender el funcionamiento de JHipster y Angular 5, herramientas con las que ninguno de los integrantes del equipo había trabajado antes; pero las investigaciones acerca de las herramientas, la ayuda entre integrantes del equipo y en ocasiones el ensayo y error, han hecho que JHipster haya dado los resultados esperados, y que el equipo realmente valore este tipo de herramientas que están en auge, mejoran cada día y nos obligan a mantenernos actualizados.

Por otro lado, desde el punto de vista personal de la autora, la experiencia haciendo el FORTE ha sido muy enriquecedora. Le ha permitido aprender nuevas tecnologías y formas de trabajar de las que en la carrera solo había oído hablar, además de adentrarse en el mundo laboral dentro del ámbito de la informática. Aprender cómo es el flujo de trabajo dentro de una empresa, en un proyecto real es algo que será muy útil de cara al futuro, ayudando a los recién graduados a perder ese “miedo” que da buscar el primer trabajo y hacer las primeras entrevistas, además de aportar una experiencia extra antes de entrar definitivamente en el mundo laboral.

En cuanto a IECISA, la acogida fue muy buena y la autora se sintió casi desde el principio como una más del equipo, gracias al buen ambiente de trabajo y entre compañeros. No sentirse fuera de lugar animó a la autora a entrar en la dinámica de trabajo fácilmente y querer participar desde el primer momento.

En resumen, la realización de este Trabajo de Fin de Grado con el convenio FORTE le ha parecido a la autora una oportunidad única que ofrece una experiencia totalmente distinta a lo vivido durante el grado que además muy probablemente abrirá puertas al alumno que participe en él.

6. CONCLUSIONES

SCRUM

Desde el punto de vista del marco de trabajo seguido en ONCOSUP, SCRUM, el resultado ha sido muy satisfactorio para la autora y todo el equipo implicado en el desarrollo. SCRUM fomenta el trabajo en equipo, la interacción, la comunicación, la proactividad, entre muchas otras características, todas ellas hacen que, si los miembros del equipo participan, el ambiente dentro del proyecto sea muy bueno y los integrantes tengan ganas de trabajar y sacarlo adelante.

Desde el punto de vista de la autora el trabajo con SCRUM ha dado muy buenos resultados. A pesar de que algunos de los riesgos que se planificaron al principio se materializaron, el equipo tuvo un plan de mitigación que se llevó a cabo, asegurándose de haber hecho todo lo que estaba en su mano para evitarlos. En lo referente a los problemas que ha habido a lo largo del desarrollo, es reconfortante saber que alguno son bastante comunes y que en general las soluciones son similares a las que se han dado en ONCOSUP, como se puede leer en el capítulo “CÓMO HACEMOS RETROSPECTIVAS DE SPRINT” de *SCRUM y XP desde las trincheras* [Kni07], en el que se habla de los problemas más habituales que se detectan en las retrospectivas.

Además el marco de trabajo captó la atención de la autora lo suficiente como para animarse a hacer su propia aportación y realizar un pequeño estudio acerca de la confianza, la visión y distintos puntos de SCRUM, desde el punto de vista de la comunidad de proyecto. Los resultados de este análisis se pueden ver en la sección 6.5.

JHipster

En cuanto a JHipster, los resultados han sido incluso mejores que los esperados. Se ha conseguido cubrir todo el alcance del proyecto en el tiempo que se había previsto.

JHipster proporciona una arquitectura con una aplicación lista para usar, que requiere por parte del equipo de desarrollo implementar los cambios necesarios para que cumpla con los requisitos del cliente. Esto permite al equipo centrarse en desarrollar la funcionalidad, ahorrando mucho trabajo.

Por otro lado si se consulta el número de horas que se han dedicado al proyecto, se obtiene una cifra de unas 1.240 horas, que comparado con las 840 que se estimaron al inicio del proyecto puede parecer mucho, pero la diferencia no es tan grande teniendo en cuenta que: es la primera vez que se trabaja con la herramienta y la estimación se basaba en proyectos similares, pero no iguales; el equipo tuvo que dedicar tiempo a estudiar y aprender; y que en esas horas están también las que registró la autora, con cuya participación no se contaba al inicio del proyecto.

La autora ha querido realizar un pequeño análisis de la herramienta haciendo una comparativa de la aplicación recién generada, y la aplicación tal y como ha quedado tras trabajar

en ella. Se ha decidido realizar un conteo de líneas para poder ver cuánto ha crecido la aplicación en su desarrollo. Para realizar este recuento se ha usado la herramienta *Cloc* [clo18], un proyecto cuyo fin es ayudar a realizar este tipo de análisis, permitiendo saber el total de líneas blancas, comentadas y de código. Permite entre otras opciones además mostrarlas ordenadas por lenguajes, contar solamente archivos con cierta extensión o excluir directorios o archivos según un patrón.

La figura 6.1 muestra un conteo de las líneas que genera JHipster para ONCOSUP, mientras que, la figura 6.1 refleja el recuento de líneas una vez implementada toda la funcionalidad que se entregó al final del proyecto.

Language	files	blank	comment	code
Java	727	15996	11929	63188
TypeScript	1142	9597	1998	58313
HTML	301	452	31	21433
JSON	289	135	0	11660
XML	92	1125	422	4310
Maven	1	10	77	881
YAML	8	38	161	355
CSS	7	59	53	274
Bourne Shell	1	30	65	138
DOS Batch	1	32	0	113
Markdown	1	64	0	103
JavaScript	1	21	20	40
Dockerfile	1	4	0	9
SUM:	2572	27563	14756	160817

Figura 6.1: Recuento de líneas de código al generar el proyecto

Language	files	blank	comment	code
Java	732	16944	13773	70183
TypeScript	1142	9891	1998	60264
JSON	237	76	0	26937
HTML	301	819	203	20902
XML	92	1125	422	4310
Maven	1	9	77	891
CSS	29	70	53	847
YAML	8	38	161	359
Bourne Shell	1	30	65	138
DOS Batch	1	32	0	113
Markdown	1	64	0	101
JavaScript	1	21	20	40
Dockerfile	1	4	0	9
SUM:	2547	29123	16772	185094

Figura 6.2: Recuento de líneas de código al finalizar el proyecto

6. CONCLUSIONES

Analizando los números se puede observar que la mayor cantidad de líneas que ha tenido que añadir el equipo han sido en Java, unas 7.000 líneas, lo que tiene sentido, teniendo en cuenta que ha habido que adaptar el negocio de la aplicación a lo que se necesitaba para ONCOSUP. Se han añadido unas 2.000 líneas en TypeScript, lo cual es razonable ya que los archivos escritos en este lenguaje son los que se encargan del tratamiento de los datos en el cliente. Debido a que ha habido que retocar la maquetación se puede observar un incremento de archivos *css* y, en consecuencia, un aumento de líneas.

Es curioso que en los archivos HTML lo que se ha hecho ha sido reducir el número de líneas, esto se debe a que en la mayoría de casos se han eliminado campos o botones innecesarios, y la nueva maquetación de las pantallas requería menos líneas.

En el cómputo global, la aplicación final tiene un total de 185.094 líneas, 24.277 más que las 160.817 generadas inicialmente. Lo que deja claro que a pesar de que el equipo ha trabajado mucho y ha añadido mucha funcionalidad al equipo, JHipster ha ayudado mucho a que el proyecto salga adelante. Un proyecto de esta magnitud no habría sido posible sin toda la ayuda que JHipster ha dado al equipo.

Objetivos cumplidos

A pesar de lo ajustado que se ha ido de tiempo, finalmente se pudieron completar los cinco objetivos específicos del proyecto, completando así el objetivo general. Estos objetivos son:

- Objetivo 1: Estudio Preconsulta. Se comienza a trabajar en este objetivo desde el primer sprint, y a pesar de haberse planeado finalizarlo para la finalización del sprint 2, fue necesario alargar su desarrollo un sprint más, habiendo sido completado al acabar el sprint 3.
- Objetivo 2: Consulta de Seguimiento. Cuando el primer objetivo estuvo prácticamente acabado se comenzó a trabajar en el segundo, es decir durante el tercer sprint. La cantidad de elementos que conformaban la consulta resultó en que el equipo tuvo que trabajar en este objetivo hasta el último día del proyecto, habiéndolo acabado durante el sexto sprint.
- Objetivo 3: Informes. Una vez la consulta estuvo lo suficientemente avanzada, se inició el trabajo en el objetivo 3. Esto fue en el sprint 5, en el que se trabajó en el filtro, y se finalizó con la impresión en el sprint 6.
- Objetivo 4: Administración de Usuarios y Administración de Protocolos. Como se explicó en la sección 2.2, era necesario comenzar a trabajar en este objetivo para poder cerrar el primero, así que en el primer, segundo y tercer sprint se completó gran parte de las tareas necesarias para cumplir con el objetivo, quedando parado su avance hasta el sexto sprint, donde finalmente se completó.

- Objetivo 5: Exportación de datos. Para exportar los datos introducidos en la aplicación era imprescindible saber con certeza que la estructura de los datos no iba a cambiarse. Como se ha trabajado de forma ágil se han tenido que hacer múltiples modificaciones durante el desarrollo, así que este objetivo se decidió abordarlo en el sexto sprint, cuando se sabía con seguridad que no habría más cambios.

Además, es importante mencionar que al acabar el sprint 3 ya se había acabado el MVP. El avance y cumplimiento de los objetivos ha sido rápido y organizado, lo que ha permitido completarlos todos y no tener que dejar nada sin abordar.

Objetivos personales

La autora cree que se han cubierto de sobra sus objetivos personales. Como ya se justificó en el capítulo 2 y con lo redactado en el capítulo 5 queda más que claro durante la participación de la autora en ONCOSUP se han tocado diferentes aspectos de esas competencias.

En cuanto a la ampliación de conocimientos, la autora está más que satisfecha. Sus conocimientos respecto a herramientas como JHipster, Angular 5, *JIRA*, *Confluence* o SCRUM, eran mínimos. Aunque está claro que aún hay margen para seguir aprendiendo y ampliando conocimientos, la estancia en IECISA ha sentado una buena base de conocimientos que permiten a la autora desenvolverse con más facilidad, tanto con las herramientas mencionadas, como en el ambiente de trabajo en la empresa.

Un estudio de valor añadido: Valoración de SCRUM

Durante el desarrollo de ONCOSUP la autora tomó la iniciativa de pasar unos breves cuestionarios al final de cada sprint a los participantes del proyecto, con el fin de ver qué tan bien funciona SCRUM y cómo ayuda a cambiar, para bien o para mal, la opinión de los involucrados en el proyecto.

La herramienta utilizada para la realización y gestión de las encuestas fueron los Formularios de Google [for18], que permite crear, modificar o incluso duplicar formularios de forma rápida y sencilla. También, muestra un breve análisis de las respuestas, incluyendo la posibilidad de exportar los resultados en una hoja de cálculo, lo que facilitará mucho la recopilación y procesado de los datos.

Se quiso valorar por un lado las sensaciones respecto al proyecto durante su desarrollo y por otra parte se quiso medir en la medida de lo posible cómo el equipo considera que está desempeñando en lo referente a los valores de SCRUM. De esta forma, se les pide que respondan a las siguientes preguntas valorando del 1 al 5, siendo 1 muy poco y 5 mucho:

- Sensaciones respecto al proyecto
 - ¿Cuál es tu grado de confianza relativo al éxito del proyecto?
 - ¿Cuál es tu grado de satisfacción respecto al avance del proyecto en este sprint?

6. CONCLUSIONES

- El sentimiento de “ownership” (Pertenencia, implicación...) se corresponde con el nivel de entrega e inversión personal en el proyecto y puede variar por muchos factores a lo largo del proyecto. ¿Cómo valoras tu sentimiento de “ownership” tras este sprint?

■ Valores de SCRUM

- Compromiso. Estar comprometidos con el equipo y el objetivo del sprint, de tal forma que al acabar el trabajo se comprueba si es posible ayudar al resto del equipo a alcanzar el objetivo.
- Foco. Mantener los ojos en el objetivo del sprint y no dejarse distraer por preferencias personales o el CEO (*Chief Executive Officer*, el máximo responsable) si eso hace peligrar el objetivo del sprint.
- Franqueza. Ser transparentes acerca de cómo lo está haciendo el equipo y ser abierto en lo referente a fallos o dificultades que puedan aparecer.
- Respeto. Compartir el conocimiento con el equipo y aceptar la profesionalidad del resto de componentes.
- Coraje. Ser un profesional que admite errores y se mueve más allá de su propio ego para cambiar la dirección con el fin de resolver problemas complejos.

La figura 6.3 muestra los resultados por sprint de las sensaciones del equipo. Se puede apreciar que desde el principio y de forma general las sensaciones eran buenas, comenzando con valores relativamente altos. Hay dos elementos claros en las tres gráficas: todas tienen una tendencia ascendente casi todo el desarrollo, y las tres sufren una caída al final de éste.

Era de esperar que la valoración de los integrantes del equipo fuese aumentando con el paso de los sprints. Si nos remitimos a lo documentados en el capítulo 5, resulta fácil comprender a qué se debe la bajada hacia el final del proyecto; algunos de los riesgos identificados antes de comenzar con el desarrollo se materializaron, lo que junto a la incertidumbre de si sería posible abarcar todo el alcance del proyecto y la presión de la fecha de finalización cada vez más cerca, probablemente repercutieron en la visión del equipo.

Cabe mencionar que la caída no es muy drástica, pero lo ideal sería mantener todas las puntuaciones en el 5, el valor más alto. Una mejor gestión de riesgos o un presupuesto menos restrictivo, muy probablemente se habría traducido en puntuaciones más altas hacia el final del proyecto.

Respecto a la valoración del equipo de los valores de SCRUM (figura 6.4), de nuevo se aprecia una esperada tendencia ascendente. En general, todos los números ascienden o se mantienen en valores altos, acabando todos más altos al acabar el proyecto que al empezarlos. La excepción es el compromiso, que tiene una pequeña subida, para al final mantenerse en su valor inicial. Se dio la opción de añadir comentarios referentes a las respuestas en los

UN ESTUDIO DE VALOR AÑADIDO: VALORACIÓN DE SCRUM

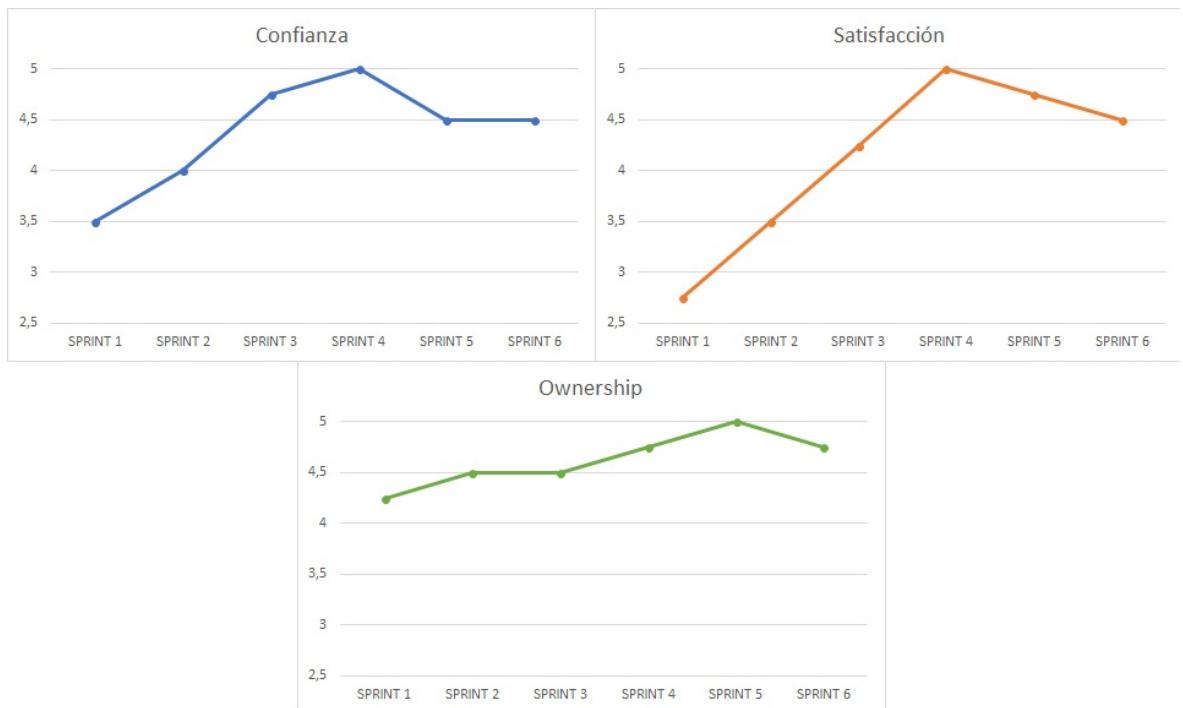


Figura 6.3: Resultados del equipo en la primera parte del cuestionario

formularios de los que se puede extraer que quizás la presión o responsabilidades en otros proyectos pudo afectar al compromiso del equipo, explicando que no haya aumentado como el resto de valores.

6. CONCLUSIONES

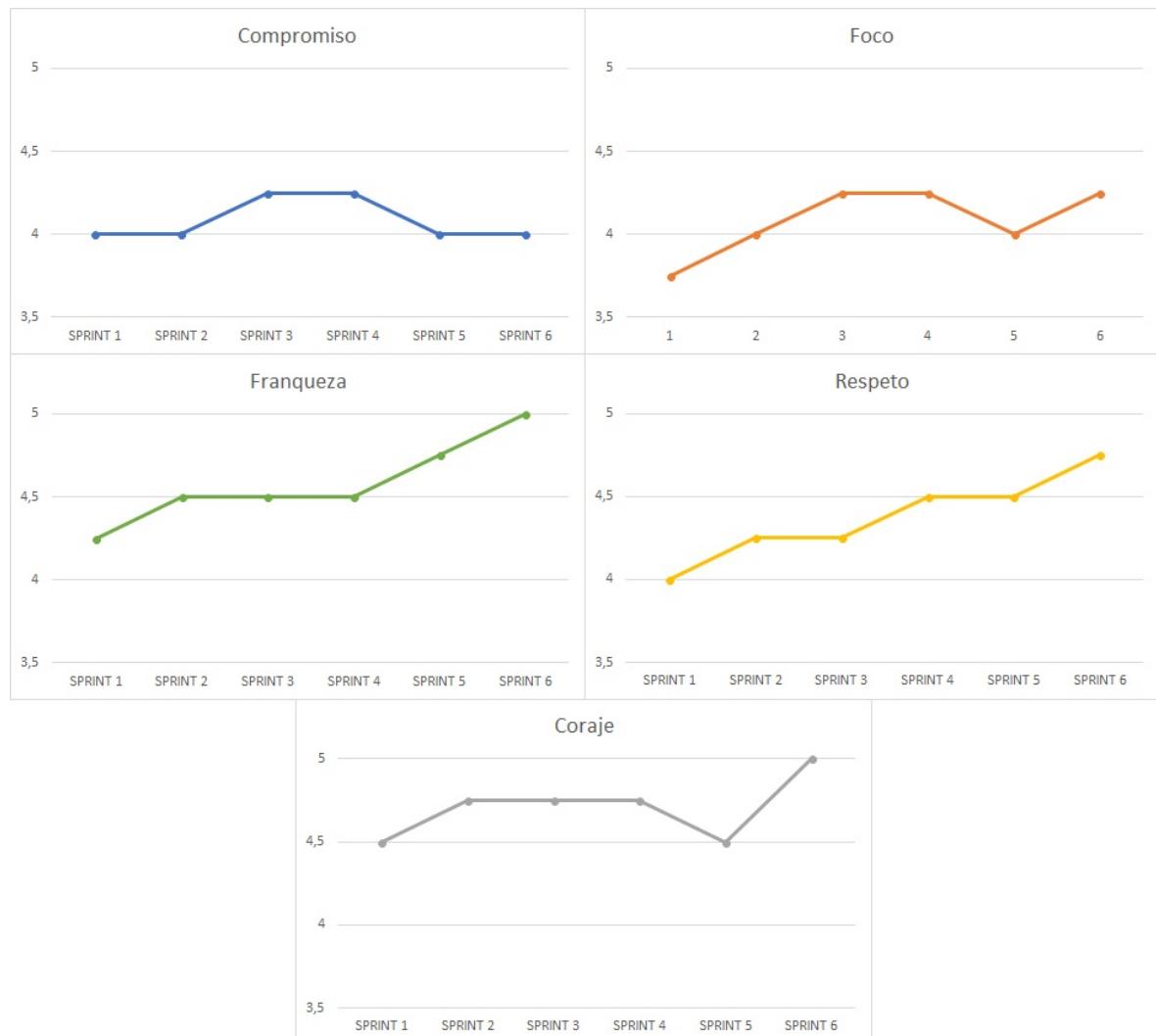


Figura 6.4: Resultados del equipo en la segunda parte del cuestionario

El balance general es un ascenso, pero se puede apreciar una caída en el coraje y el foco en el sprint 4, que podría ser resultado de la falta de información respecto a los items durante ese sprint.

Aunque SCRUM se centra en el equipo y lo considera la principal unidad en un proyecto, se considera interesante en este caso analizar las respuestas de la *Product Owner*, ya que no solo era su primer contacto con SCRUM sino que aceptó la responsabilidad sin tener ningún tipo de formación sobre el rol y a sabiendas de que quizás no podría dedicar todo el tiempo necesario.

En la figura 6.5 se puede ver que las sensaciones son similares a las del equipo, siendo la bajada únicamente en el sexto sprint, lo que puede deberse a que algunos de los riesgos que se dieron durante los últimos sprints no le afectaban directamente mientras que otros en el sexto sprint sí lo hacían, por ejemplo, la instalación fue algo que durante el último sprint hizo peligrar que pudiera acabar usando la aplicación.

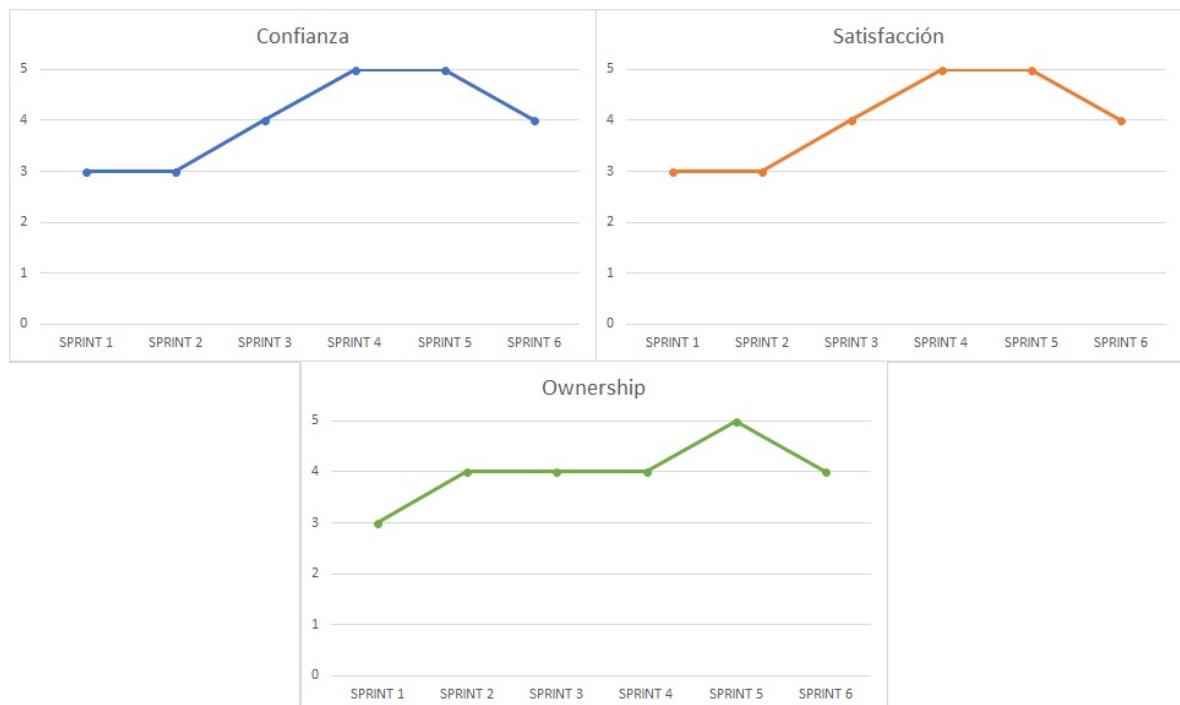


Figura 6.5: Resultados de la Product Owner en la primera parte del cuestionario

En cuanto a los valores de SCRUM (figura 6.6), lo más llamativo es el valor bajo y estable durante todo el proyecto del compromiso, algo de lo que el equipo era consciente, ya que, tal y como se explicó en la sección 5.1, no era posible asegurar que la *Product Owner* tuviera el tiempo necesario que requería el rol. Este es uno de los riesgos de tener un *Product Owner* que no puede dedicarse al completo a su rol. El respeto también ha sido estable durante todos los sprints, pero con valores más altos.

6. CONCLUSIONES

Lo más relevante de estas gráficas es la caída en el sprint 4 del foco, que quizá debido a sus dificultades para estar totalmente comprometida con el proyecto, provocaron que en ese sprint no fuese capaz de proporcionar al equipo toda la información necesaria para completar el trabajo del sprint. Lo realmente interesante, es como esta caída en el foco del *Product Owner* en el sprint 4, se tradujo en una caída general en el foco del equipo en el sprint 5, como se puede ver al comparar con la figura 6.4.

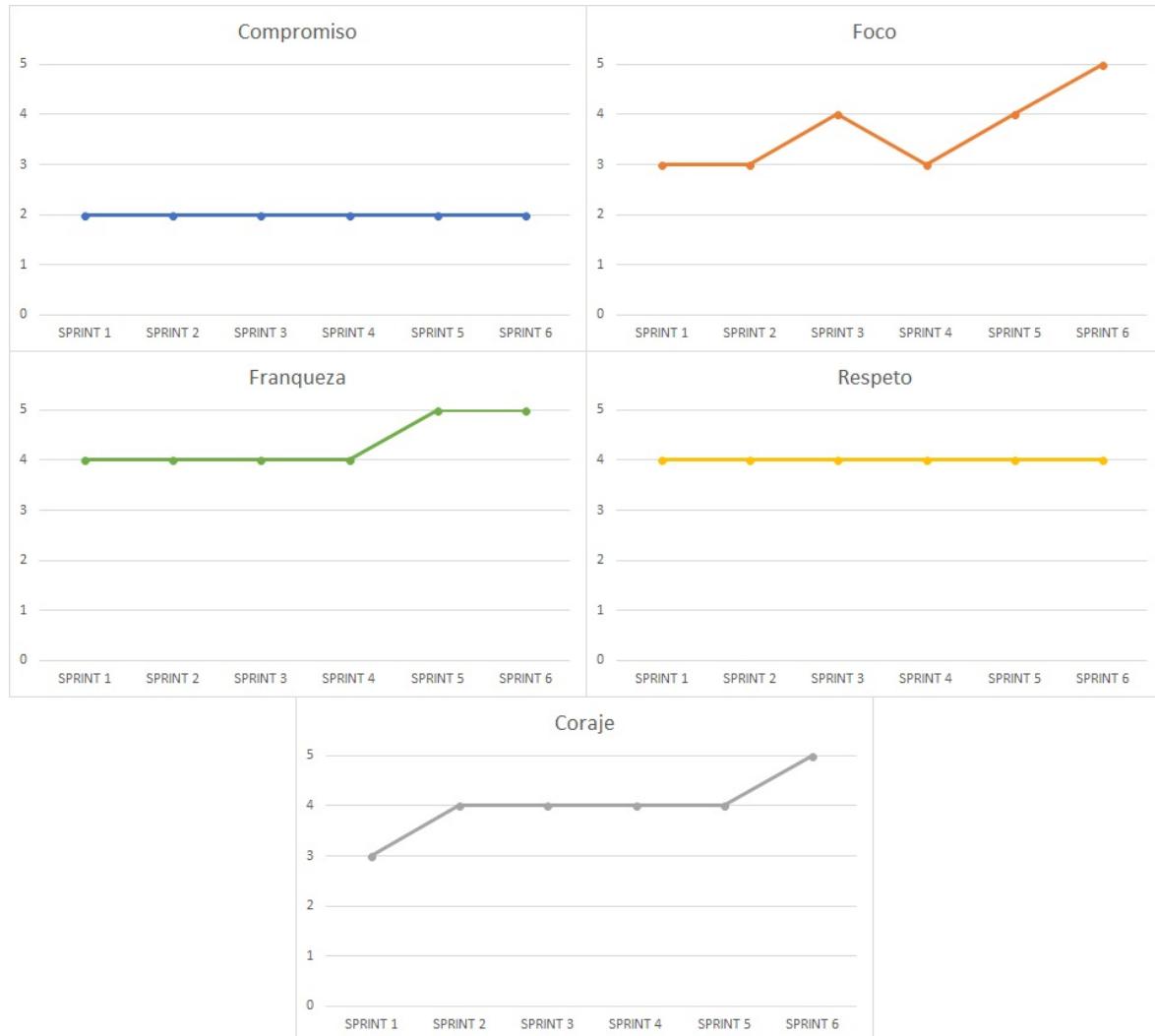


Figura 6.6: Resultados de la Product Owner en la segunda parte del cuestionario

Finalmente, las gráficas de la figura 6.7 son una propuesta sobre cómo mostrar la información recopilada en los cuestionarios de tal forma que permita ver de forma rápida cual ha sido la evolución del equipo desde el inicio hasta el final del proyecto.

En líneas generales, se puede decir que SCRUM es un marco de trabajo que funciona realmente bien. Los resultados en general han sido muy buenos, y las opiniones de los miembros del equipo también. Como opinión personal, la autora considera que la implantación de formularios de este tipo en proyectos de desarrollo, puede ser una herramienta muy buena como

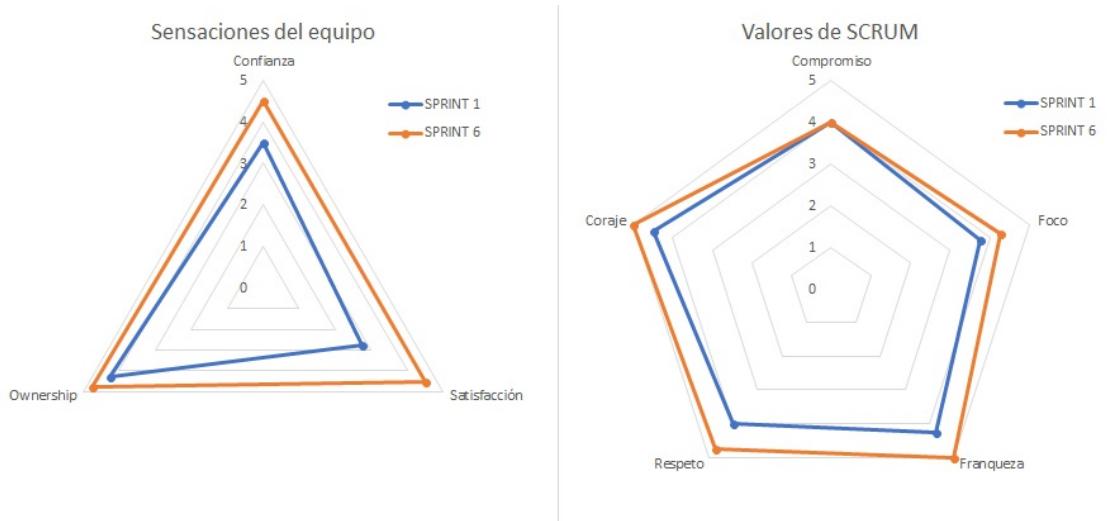


Figura 6.7: Comparativa de los resultados del primer y el sexto sprint

soporte para retrospectivas, como herramienta de facilitación para *Scrum Master* para divulgar y promover el cambio cultural a través de la organización poniendo énfasis en este tipo de elementos intangibles y subjetivos que fuera de la agilidad no son fáciles de medir. Podría incluso servir para medir el estado y evolución del equipo de forma similar o complementaria al Modelo de Tuckman [Smi18].

ANEXOS

Anexo A

Estructura de la aplicación final

EN este anexo se proporciona el código del JDL de ONCOSUP una vez finalizado su desarrollo. Además, se adjunta una imagen que muestra un esquema de la estructura de la aplicación.

Estructura de entidades de ONCOSUP

Debido al tamaño de la aplicación, la imagen es demasiado grande para mostrarla en una sola página. Por esta razón, se adjunta un CD en la versión física de este tomo con dicha imagen para su consulta.

Se proporciona, además para su consulta en la versión digital un enlace a la imagen y un código QR (figura A.1) para su acceso desde dispositivos móviles.



Figura A.1: Enlace a la imagen con la estructura de las entidades de ONCOSUP

<https://drive.google.com/open?id=1lKQUtffj-JNOvRDxbMoLG8FVTUozmGj0>

JDL de ONCOSUP

```
1  entity Paciente {
2      nombre String required,
3      apellidos String required,
4      fechaNacimiento LocalDate required,
5      sexo Sexo required,
6      nhc String maxlength(38) pattern(/[0-9]+/),
7      nss String maxlength(20),
8      tarjetaSanitaria String maxlength(20),
9      cipa String maxlength(20),
10     telefono String,
11     email String,
12     annoDiagnostico String maxlength(4)
13 }
14
15 enum Sexo {
16     VARON, MUJER
17 }
18
19 entity Nacionalidad {
20     nombre String required
21 }
22
23 entity Etnia {
24     nombre String required
25 }
26
27 entity Hospital {
28     nombre String required
29 }
30
31 entity ComunidadAutonoma {
32     nombre String required
33 }
34
35 entity ProvinciaPais {
36     nombre String required
37 }
38
39 entity Diagnostico {
40     fechaDiagnostico LocalDate required,
41     patologiaOncologica PatologiaOncologica required,
42     diagnosticoEspecifico String,
43     estadioLeucemia EstadioLeucemia,
44     estadioTumorSolido EstadioTumorSolido,
45     estadioOtros String,
46     localizacion String,
47     recaida Boolean required
48 }
49
50 enum PatologiaOncologica {
51     LEUCEMIA, TUMOR_SOLIDO, OTROS
52 }
53
54 enum EstadioLeucemia {
55     SIN_AFECTACION_AL_SISTEMA_NERVIOSO_CENTRAL,
56     CON_AFECTACION_AL_SISTEMA_NERVIOSO_CENTRAL
57 }
58
59 enum EstadioTumorSolido {
60     LOCALIZADO, METASTASICO
```

```

60    }
61
62 entity Cie10 {
63     codigo String required,
64     nombre String required
65 }
66
67 entity Cie10Secundario {
68     codigo String required,
69     nombre String required
70 }
71
72 entity Recaida {
73     fechaRecaida LocalDate required,
74     patologiaOncologica PatologiaOncologica required,
75     tiempoLeucemia TiempoLeucemia,
76     localizacionLeucemia LocalizacionLeucemia,
77     localizacionTumorSolido LocalizacionTumorSolido
78 }
79
80 enum TiempoLeucemia {
81     TARDIA, PRECOZ
82 }
83
84 enum LocalizacionLeucemia {
85     MEDULAR, EXTRAMEDULAR, COMBINADA
86 }
87
88 enum LocalizacionTumorSolido {
89     LOCAL, METASTASICO
90 }
91
92 entity Protocolo {
93     identificador String required,
94     nombre String required,
95     fechaInicio LocalDate required,
96     fechaFin LocalDate required
97 }
98
99 entity Rama {
100     identificador String required,
101     nombre String required
102 }
103
104 entity TratamientoRecomendado {
105     identificador String required,
106     quimioterapia Boolean required,
107     radioterapia Boolean required,
108     cirugia Boolean required,
109     trasplante Boolean required,
110     otrosTratamientos Boolean required
111 }
112
113 entity FarmacoAsociado {
114     dosis String required
115 }
116
117 entity Farmaco {
118     nombre String required,
119     unidad Unidad required,
120     nombreOtraUnidad String,
121     viaAdministracion ViaAdministracion required

```

```

122     }
123
124     enum Unidad {
125         MG_M2, MG_KG, UL_M2, OTRA_UNIDAD
126     }
127
128     enum ViaAdministracion {
129         INTRAMUSCULAR, INTRATECAL, INTRAVENOSA, ORAL, SUBCUTANEO
130     }
131
132     entity Tratamiento {
133         protocoloEstandarizado Boolean required,
134         fechaInicio LocalDate required,
135         fechaFin LocalDate required,
136         recQuimioterapia Boolean required,
137         recRadioterapia Boolean required,
138         recCirugia Boolean required,
139         recTrasplante Boolean required,
140         recOtrosTratamientos Boolean required,
141         complicaciones Boolean required,
142         complicacion String,
143         dispositivos TextBlob
144     }
145
146     entity Quimioterapia {
147         lugarAdministracion String,
148         fechaInicio LocalDate required,
149         fechaFin LocalDate required
150     }
151
152     entity FarmacoQuimio {
153         dosis String required
154     }
155
156     entity Radioterapia {
157         lugarTratamiento String,
158         fechaInicio LocalDate required,
159         fechaFin LocalDate required,
160         modalidadTratamiento ModalidadTratamiento,
161         localizacion String,
162         craneoespinal Boolean,
163         dosisTotal String,
164         fraccionado Boolean,
165         boost Boolean,
166         organosRiesgo String
167     }
168
169     enum ModalidadTratamiento {
170         FOTONES, PROTONES
171     }
172
173     entity Cirugia {
174         fechaCirugia LocalDate,
175         lugar String,
176         tipoIntervencion String,
177         amputacion Boolean,
178         protesis Boolean
179     }
180
181     entity Trasplante {
182         fechaTrasplante LocalDate,
183         segundoTrasplante Boolean,

```

```

184     motivoSegundoTrasplante MotivoSegundoTrasplante,
185     motivosSegundoTrasplante String,
186     statusPrevioLeucemia StatusPrevioLeucemia,
187     statusPrevioTumor StatusPrevioTumor,
188     tipoTrasplante TipoTrasplante,
189     tipoDonante TipoDonante,
190     tipoAcondicionamiento TipoAcondicionamiento,
191     usoRadioterapia Boolean,
192     eichAgudo Boolean,
193     eichAgudoGrado GradoEich,
194     eichAgudoResuelto Boolean,
195     eichCronico Boolean,
196     eichCronicoGrado GradoEich,
197     eichCronicoResuelto Boolean,
198     enfermedadVenoclusiva Boolean,
199     microangiopatiaTrombotica Boolean,
200     infecciones Boolean,
201     otrasComplicaciones String
202 }
203
204 enum StatusPrevioLeucemia {
205     PRIMERA_REMISION_COMPLETA, SEGUNDA_REMISION_COMPLETA,
206         TERCERA_REMISION_COMPLETA_O_MAS, NO_REMISION_COMPLETA_O_PROGRESION
207 }
208
209 enum StatusPrevioTumor {
210     REMISION_COMPLETA, REMISION_PARCIAL, PROGRESION
211 }
212
213 enum TipoTrasplante {
214     AUTOLOGO, ALOGENICO
215 }
216
217 enum TipoDonante {
218     EMPARENTADO_HLA_IDENTICO, EMPARENTADO_HLA_NO_IDENTICO,
219         NO_EMPARENTADO_HLA_IDENTICO, NO_EMPARENTADO_HLA_NO_IDENTICO
220 }
221
222 enum TipoAcondicionamiento {
223     MIELOABLATIVO, INTENSIDAD_REDUCIDA
224 }
225
226 enum GradoEich {
227     UNO, DOS, TRES, CUATRO
228 }
229
230 enum MotivoSegundoTrasplante {
231     RECAIDA, FALLO_INJERTO, OTROS
232 }
233
234 entity FarmacoTrasplante {
235     dosis String required
236 }
237
238 entity FarmacoInmunosupresor {
239     dosis String required,
240     tiempo String
241 }
242
243 entity OtroTratamiento {
244     inmunoterapia Boolean,
245     inmunoterapiaFecha LocalDate,

```

```

244     inmunoterapiaDesc String,
245     tratamiento131Mibg Boolean,
246     tratamiento131MibgFecha LocalDate,
247     ablacionTiroideAI131 Boolean,
248     ablacionTiroideAI131Fecha LocalDate,
249     otrosTratamientos String,
250     otrosTratamientosFecha LocalDate
251 }
252
253 entity Consulta {
254     fechaConsulta LocalDate required,
255     edad String maxlength(3),
256     exitus Boolean required,
257     seguimientoHabitual Boolean required,
258     tratamientoHabitual Boolean required,
259     isCardio Boolean required,
260     isNeumo Boolean required,
261     isEndocrinoCrecimiento Boolean required,
262     isEndocrinoPeso Boolean required,
263     isEndocrinoDiabetes Boolean required,
264     isEndocrinoTiroides Boolean required,
265     isMaduracionSexual Boolean required,
266     isNefrourologico Boolean required,
267     isDigestivo Boolean required,
268     isHigado Boolean required,
269     isPiel Boolean required,
270     isORL Boolean required,
271     isDientesBoca Boolean required,
272     isOjos Boolean required,
273     isHematologico Boolean required,
274     isVacunas Boolean required,
275     isAutoinmunidad Boolean required,
276     isNeurologico Boolean required,
277     isNeuropsicologico Boolean required,
278     isMuscular Boolean required,
279     isEsqueletico Boolean required,
280     isCapacidadFisica Boolean required,
281     isSexualidad Boolean required,
282     isFertilidad Boolean required,
283     isPsicologico Boolean required,
284     isSocial Boolean required,
285     isDrogas Boolean required,
286     isInformacion Boolean required,
287     isSegundoTumor Boolean required,
288     observaciones TextBlob,
289     proxRevision String
290 }
291
292 entity Especialidad {
293     nombreEspecialidad NombreEspecialidad required,
294     motivo String,
295     frecuencia String,
296     centroSanitario String,
297     fechaInicio LocalDate
298 }
299
300 enum NombreEspecialidad {
301     CARDIO, NEUMO, ENDOCRINO_CRECIMIENTO, ENDOCRINO_PESO, ENDOCRINO_DIABETES,
302         ENDOCRINO_TIROIDES, MADURACION_SEXUAL, NEFRUROLOGICO, DIGESTIVO, HIGADO,
303             PIEL, ORL, DIENTES_BOCA,
304             OJOS, HEMATOLOGICO, VACUNAS, AUTOINMUNIDAD, NEUROLOGICO, NEUROPSICOLOGICO,
305                 MUSCULAR, ESQUELETICO, CAPACIDAD_FISICA, SEXUALIDAD, FERTILIDAD,

```

```

    PSICOLOGICO, SOCIAL, DROGAS
303 }
304
305 entity TratamientoHabitual {
306     farmaco String required,
307     fechaInicio LocalDate
308 }
309
310 entity Constante {
311     escalaActividad EscalaActividad required,
312     temperatura String,
313     tas String,
314     tad String,
315     fc String,
316     fr String,
317     saturacionOxigeno String,
318     peso String,
319     talla String,
320     imc String,
321     superficieCorporal String
322 }
323
324 enum EscalaActividad {
325     KARNOFSKY, LANSKY
326 }
327
328 entity EscalaKarnofsky {
329     codigo String required,
330     nombre String required
331 }
332
333 entity EscalaLansky {
334     codigo String required,
335     nombre String required
336 }
337
338 entity ExploracionFisica {
339     resultadoExploracion ResultadoPrueba required,
340     observaciones String maxlength(1000),
341     estadioTanner EstadioTanner
342 }
343
344 enum ResultadoPrueba {
345     NORMAL, ALTERADA
346 }
347
348 enum EstadioTanner {
349     I, II, III, IV, V
350 }
351
352 entity Cardio {
353     complicacion Boolean,
354     desComplicacion String,
355     fechaInicioComplicacion LocalDate,
356     seguimiento Boolean,
357     fechaUltimaRevision LocalDate,
358     tratamiento Boolean,
359     desTratamiento String,
360     fechaInicioTratamiento LocalDate
361 }
362
363 entity Neumo {

```

```

364     complicacion Boolean,
365     desComplicacion String,
366     fechaInicioComplicacion LocalDate,
367     seguimiento Boolean,
368     fechaUltimaRevision LocalDate,
369     tratamiento Boolean,
370     desTratamiento String,
371     fechaInicioTratamiento LocalDate
372 }
373
374 entity EndocrinoCrecimiento {
375     complicacion Boolean,
376     desComplicacion String,
377     fechaInicioComplicacion LocalDate,
378     seguimiento Boolean,
379     fechaUltimaRevision LocalDate,
380     tallaMadre String,
381     tallaPadre String,
382     tallaDiana String,
383     tratamiento Boolean,
384     desTratamiento String,
385     fechaInicioTratamiento LocalDate
386 }
387
388 entity EndocrinoPeso {
389     complicacion Boolean,
390     desComplicacion String,
391     fechaInicioComplicacion LocalDate,
392     seguimiento Boolean,
393     fechaUltimaRevision LocalDate,
394     tratamiento Boolean,
395     desTratamiento String,
396     fechaInicioTratamiento LocalDate,
397     dietaAdelgazar Boolean
398 }
399
400 entity EndocrinoDiabetes {
401     complicacion Boolean,
402     desComplicacion String,
403     fechaInicioComplicacion LocalDate,
404     seguimiento Boolean,
405     fechaUltimaRevision LocalDate,
406     tratamiento Boolean,
407     desTratamiento String,
408     fechaInicioTratamiento LocalDate
409 }
410
411 entity EndocrinoTiroides {
412     complicacion Boolean,
413     desComplicacion String,
414     fechaInicioComplicacion LocalDate,
415     seguimiento Boolean,
416     fechaUltimaRevision LocalDate,
417     tratamiento Boolean,
418     desTratamiento String,
419     fechaInicioTratamiento LocalDate
420 }
421
422 entity MaduracionSexual {
423     complicacion Boolean,
424     desComplicacion String,
425     fechaInicioComplicacion LocalDate,

```

```

426     seguimiento Boolean,
427     fechaUltimaRevision LocalDate,
428     tratamiento Boolean,
429     desTratamiento String,
430     fechaInicioTratamiento LocalDate
431 }
432
433 entity Nefrourologico {
434     complicacion Boolean,
435     desComplicacion String,
436     fechaInicioComplicacion LocalDate,
437     seguimiento Boolean,
438     fechaUltimaRevision LocalDate,
439     tratamiento Boolean,
440     desTratamiento String,
441     fechaInicioTratamiento LocalDate
442 }
443
444 entity Digestivo {
445     complicacion Boolean,
446     desComplicacion String,
447     fechaInicioComplicacion LocalDate,
448     seguimiento Boolean,
449     fechaUltimaRevision LocalDate,
450     tratamiento Boolean,
451     desTratamiento String,
452     fechaInicioTratamiento LocalDate
453 }
454
455 entity Higado {
456     complicacion Boolean,
457     desComplicacion String,
458     fechaInicioComplicacion LocalDate,
459     seguimiento Boolean,
460     fechaUltimaRevision LocalDate,
461     tratamiento Boolean,
462     desTratamiento String,
463     fechaInicioTratamiento LocalDate
464 }
465
466 entity Piel {
467     complicacion Boolean,
468     desComplicacion String,
469     fechaInicioComplicacion LocalDate,
470     seguimiento Boolean,
471     fechaUltimaRevision LocalDate,
472     tratamiento Boolean,
473     desTratamiento String,
474     fechaInicioTratamiento LocalDate
475 }
476
477 entity ORL {
478     complicacion Boolean,
479     desComplicacion String,
480     fechaInicioComplicacion LocalDate,
481     seguimiento Boolean,
482     fechaUltimaRevision LocalDate,
483     tratamiento Boolean,
484     desTratamiento String,
485     fechaInicioTratamiento LocalDate
486 }
487

```

```

488 entity DientesBoca {
489     complicacion Boolean,
490     desComplicacion String,
491     fechaInicioComplicacion LocalDate,
492     seguimiento Boolean,
493     fechaUltimaRevision LocalDate,
494     tratamiento Boolean,
495     desTratamiento String,
496     fechaInicioTratamiento LocalDate
497 }
498
499 entity Ojos {
500     complicacion Boolean,
501     desComplicacion String,
502     fechaInicioComplicacion LocalDate,
503     seguimiento Boolean,
504     fechaUltimaRevision LocalDate,
505     tratamiento Boolean,
506     desTratamiento String,
507     fechaInicioTratamiento LocalDate
508 }
509
510 entity Hematologico {
511     complicacion Boolean,
512     desComplicacion String,
513     fechaInicioComplicacion LocalDate,
514     seguimiento Boolean,
515     fechaUltimaRevision LocalDate,
516     tratamiento Boolean,
517     desTratamiento String,
518     fechaInicioTratamiento LocalDate
519 }
520
521 entity Vacunas {
522     complicacion Boolean,
523     desComplicacion String,
524     fechaInicioComplicacion LocalDate,
525     seguimiento Boolean,
526     fechaUltimaRevision LocalDate,
527     tratamiento Boolean,
528     desTratamiento String,
529     fechaInicioTratamiento LocalDate,
530     revacunacion Boolean,
531     vacunaAnualGripe Boolean,
532     vacunaBexero Boolean,
533     otrasVacunas String
534 }
535
536 entity Autoinmunidad {
537     complicacion Boolean,
538     desComplicacion String,
539     fechaInicioComplicacion LocalDate,
540     seguimiento Boolean,
541     fechaUltimaRevision LocalDate,
542     tratamiento Boolean,
543     desTratamiento String,
544     fechaInicioTratamiento LocalDate
545 }
546
547 entity Neurologico {
548     complicacion Boolean,
549     desComplicacion String,

```

```

550     fechaInicioComplicacion LocalDate,
551     seguimiento Boolean,
552     fechaUltimaRevision LocalDate,
553     tratamiento Boolean,
554     desTratamiento String,
555     fechaInicioTratamiento LocalDate
556 }
557
558 entity Neuropsicologico {
559     complicacion Boolean,
560     desComplicacion String,
561     fechaInicioComplicacion LocalDate,
562     seguimiento Boolean,
563     fechaUltimaRevision LocalDate,
564     tratamiento Boolean,
565     desTratamiento String,
566     fechaInicioTratamiento LocalDate
567 }
568
569 entity Muscular {
570     complicacion Boolean,
571     desComplicacion String,
572     fechaInicioComplicacion LocalDate,
573     seguimiento Boolean,
574     fechaUltimaRevision LocalDate,
575     tratamiento Boolean,
576     desTratamiento String,
577     fechaInicioTratamiento LocalDate
578 }
579
580 entity Esqueletico {
581     complicacion Boolean,
582     desComplicacion String,
583     fechaInicioComplicacion LocalDate,
584     seguimiento Boolean,
585     fechaUltimaRevision LocalDate,
586     tratamiento Boolean,
587     desTratamiento String,
588     fechaInicioTratamiento LocalDate,
589     osteoporosis Boolean,
590     fracturasPatologicas Boolean
591 }
592
593 entity CapacidadFisica {
594     complicacion Boolean,
595     desComplicacion String,
596     fechaInicioComplicacion LocalDate,
597     tratamiento Boolean,
598     desTratamiento String,
599     fechaInicioTratamiento LocalDate,
600     alteracion Boolean,
601     deporteHabitual Boolean,
602     frecuencia Frecuencia,
603     numeroHorasSemana String,
604     tipoActividadFisica String maxlength(1000)
605 }
606
607 enum Frecuencia {
608     DIARIO, MAS_TRES_DIAS_SEMANA, MENOS_TRES_DIAS_SEMANA, ESPORADICO
609 }
610
611 entity Sexualidad {

```

```

612     complicacion Boolean,
613     desComplicacion String,
614     fechaInicioComplicacion LocalDate,
615     tratamiento Boolean,
616     desTratamiento String,
617     fechaInicioTratamiento LocalDate,
618     pareja Boolean,
619     relacionesSexuales Boolean,
620     isProblemasRelaciones Boolean,
621     problemasRelaciones String maxlength(500)
622 }
623
624 entity Fertilidad {
625     complicacion Boolean,
626     desComplicacion String,
627     fechaInicioComplicacion LocalDate,
628     seguimiento Boolean,
629     fechaUltimaRevision LocalDate,
630     tratamiento Boolean,
631     desTratamiento String,
632     fechaInicioTratamiento LocalDate,
633     isTecnicasPreservacion Boolean,
634     tecnicasPreservacion String maxlength(500),
635     deseoHijos Boolean,
636     preocupacion Boolean,
637     numeroHijos String,
638     isTecnicasReproduccion Boolean,
639     tecnicasReproduccion String maxlength(500)
640 }
641
642 entity Psicologico {
643     seguimiento Boolean,
644     tipoSeguimiento TipoSeguimientoPsicologico,
645     complicacion Boolean,
646     fechaComplicacion LocalDate,
647     diagnostico String,
648     fechaInicioTratamiento LocalDate,
649     fechaFinTratamiento LocalDate,
650     testPsicologico Boolean,
651     apoyoEmocional Long,
652     apoyoInstrumental Long,
653     interaccionSocialPos Long,
654     apoyoSocialEfectivo Long,
655     indiceGlobalApoyoSocial Long,
656     rosenberg Long,
657     sf36r Long,
658     inforEnfermedad String maxlength(500),
659     infoTestMedicos String maxlength(500),
660     InfoTratamiento String maxlength(500),
661     infoOtrosServicios String maxlength(500),
662     scoreGlobal String
663 }
664
665 enum TipoSeguimientoPsicologico {
666     PSICOLOGO, TRABAJADOR_SOCIAL, AMBOS
667 }
668
669 entity Social {
670     nivelEstudios NivelEstudios,
671     curso String,
672     repetirCurso Boolean,
673     repetirCursoMomento RepetirCursoMomento,

```

```

674     situacionLaboral SituacionLaboral,
675     residencia Residencia,
676     perstacionEnfermedad Boolean,
677     descPrestacionEnfermedad TextBlob
678 }
679
680 enum NivelEstudios {
681     PRIMARIA, SECUNDARIA, BACHILLERATO, FORMACION_PROFESIONAL, UNIVERSIDAD,
682     POSTGRADO, SIN_ESTUDIOS
683 }
684
685 enum RepetirCursoMomento {
686     ANTES, DURANTE, DESPUES
687 }
688
689 enum SituacionLaboral {
690     ESTUDIANTE, ACTIVO, DESEMPLEADO, BAJA_LABORAL, OTROS
691 }
692
693 enum Residencia {
694     INDEPENDIENTE, PADRES, COMPAÑEROS, PAREJA, OTROS
695 }
696
697 entity Drogas {
698     complicacion Boolean,
699     desComplicacion String,
700     fechaInicioComplicacion LocalDate,
701     seguimiento Boolean,
702     fechaUltimaRevision LocalDate,
703     tratamiento Boolean,
704     desTratamiento String,
705     fechaInicioTratamiento LocalDate,
706     tabaco Boolean,
707     numCigarrillosDia String,
708     fechaInicioTabaco LocalDate,
709     alcohol Boolean,
710     frecuenciaAlcohol Frecuencia,
711     numeroCopasDia String,
712     numeroBorrachoMes String,
713     isOtrasDrogas Boolean,
714     otrasDrogas String maxlength(1000)
715     frecuenciaOtrasDrogas Frecuencia,
716     isDependenciaFarmacos Boolean,
717     dependenciaFarmacos String maxlength(1000)
718     fechaInicioFarmacos LocalDate
719 }
720
721 entity Informacion {
722     infoEnfermedad Info,
723     infoEfectosSecundarios Info,
724     tipoInformacion TipoInformacion,
725     consultaUtil Boolean,
726     nivelAnsiedad NivelAnsiedad
727 }
728
729 enum Info {
730     SI, NO, NO_RECUEDE
731 }
732 enum TipoInformacion {
733     TOTAL, PARCIAL
734 }
735 enum NivelAnsiedad {

```

```

735     UNO, DOS, TRES, CUATRO, CINCO, SEIS, SIETE, OCHO, NUEVE, DIEZ
736 }
737
738 entity SegundoTumor {
739     patologiaOncologica PatologiaOncologica required,
740     diagnosticoEspecifico String,
741     fechaDiagnostico LocalDate,
742     tratamiento TextBlob
743 }
744
745 entity PruebaAportada {
746     tipoPrueba TipoPrueba required,
747     resultadoPrueba ResultadoPrueba,
748     observaciones String maxlength(500)
749 }
750
751 enum TipoPrueba {
752     ANALITICA, PRUEBA_IMAGEN, INTERCONSULTAS, OTRAS
753 }
754
755 entity PruebaSolicitada {
756     tipoPrueba TipoPrueba required,
757     observaciones String maxlength(1000)
758 }
759
760 entity Informe {
761     fechaConsulta LocalDate required,
762     nombreCompleto String required,
763     fechaNacimiento LocalDate required,
764     sexo Sexo required,
765     nhc String maxlength(38) pattern(/[0-9]+/),
766     diagnostico TextBlob,
767     protocoloEstandarizado Boolean required,
768     protocolo String,
769     rama String,
770     fechaInicioTratamiento LocalDate,
771     fechaFinTratamiento LocalDate,
772     recQuimioterapia Boolean required,
773     recRadioterapia Boolean required,
774     recCirugia Boolean required,
775     recTrasplante Boolean required,
776     recOtrosTratamientos Boolean required,
777     complicaciones Boolean required,
778     complicacion String,
779     dispositivos TextBlob,
780     quimioterapia TextBlob,
781     radioterapia TextBlob,
782     cirugia TextBlob,
783     trasplantes TextBlob,
784     otrosTratamientos TextBlob,
785     recaida Boolean required,
786     recaidal TextBlob,
787     protocoloEstandarizado1 Boolean required,
788     protocolo1 String,
789     ramal String,
790     fechaInicioTratamiento1 LocalDate,
791     fechaFinTratamiento1 LocalDate,
792     recQuimioterapia1 Boolean required,
793     recRadioterapia1 Boolean required,
794     recCirugia1 Boolean required,
795     recTrasplante1 Boolean required,
796     recOtrosTratamientos1 Boolean required,

```

```
797 complicaciones1 Boolean required,
798 complicacion1 String,
799 dispositivos1 TextBlob,
800 quimioterapial TextBlob,
801 radioterapial TextBlob,
802 cirugial TextBlob,
803 trasplantes1 TextBlob,
804 otrosTratamientos1 TextBlob,
805 recaida2 TextBlob,
806 protocoloEstandarizado2 Boolean required,
807 protocolo2 String,
808 rama2 String,
809 fechaInicioTratamiento2 LocalDate,
810 fechaFinTratamiento2 LocalDate,
811 recQuimioterapia2 Boolean required,
812 recRadioterapia2 Boolean required,
813 recCirugia2 Boolean required,
814 recTrasplante2 Boolean required,
815 recOtrosTratamientos2 Boolean required,
816 complicaciones2 Boolean required,
817 complicacion2 String,
818 dispositivos2 TextBlob,
819 quimioterapia2 TextBlob,
820 radioterapia2 TextBlob,
821 cirugia2 TextBlob,
822 trasplantes2 TextBlob,
823 otrosTratamientos2 TextBlob,
824 recaida3 TextBlob,
825 protocoloEstandarizado3 Boolean required,
826 protocolo3 String,
827 rama3 String,
828 fechaInicioTratamiento3 LocalDate,
829 fechaFinTratamiento3 LocalDate,
830 recQuimioterapia3 Boolean required,
831 recRadioterapia3 Boolean required,
832 recCirugia3 Boolean required,
833 recTrasplante3 Boolean required,
834 recOtrosTratamientos3 Boolean required,
835 complicaciones3 Boolean required,
836 complicacion3 String,
837 dispositivos3 TextBlob,
838 quimioterapia3 TextBlob,
839 radioterapia3 TextBlob,
840 cirugia3 TextBlob,
841 trasplantes3 TextBlob,
842 otrosTratamientos3 TextBlob,
843 recaida4 TextBlob,
844 protocoloEstandarizado4 Boolean required,
845 protocolo4 String,
846 rama4 String,
847 fechaInicioTratamiento4 LocalDate,
848 fechaFinTratamiento4 LocalDate,
849 recQuimioterapia4 Boolean required,
850 recRadioterapia4 Boolean required,
851 recCirugia4 Boolean required,
852 recTrasplante4 Boolean required,
853 recOtrosTratamientos4 Boolean required,
854 complicaciones4 Boolean required,
855 complicacion4 String,
856 dispositivos4 TextBlob,
857 quimioterapia4 TextBlob,
858 radioterapia4 TextBlob,
```

```
859 cirugia4 TextBlob,  
860 trasplantes4 TextBlob,  
861 otrosTratamientos4 TextBlob,  
862 seguimientoHabitual Boolean required,  
863 especialidades TextBlob,  
864 tratamientoHabitual Boolean required,  
865 tratamientosHabituales TextBlob,  
866 constantes TextBlob,  
867 exploracionFisica TextBlob,  
868 isCardio Boolean required,  
869 itemCardio TextBlob,  
870 isNeumo Boolean required,  
871 itemNeumo TextBlob,  
872 isEndocrinoCrecimiento Boolean required,  
873 itemEndocrinoCrecimiento TextBlob,  
874 isEndocrinoPeso Boolean required,  
875 itemEndocrinoPeso TextBlob,  
876 isEndocrinoDiabetes Boolean required,  
877 itemEndocrinoDiabetes TextBlob,  
878 isEndocrinoTiroides Boolean required,  
879 itemEndocrinoTiroides TextBlob,  
880 isMaduracionSexual Boolean required,  
881 itemMaduracionSexual TextBlob,  
882 isNefrourologico Boolean required,  
883 itemNefrourologico TextBlob,  
884 isDigestivo Boolean required,  
885 itemDigestivo TextBlob,  
886 isHigado Boolean required,  
887 itemHigado TextBlob,  
888 isPiel Boolean required,  
889 itemPiel TextBlob,  
890 isORL Boolean required,  
891 itemORL TextBlob,  
892 isDientesBoca Boolean required,  
893 itemDientesBoca TextBlob,  
894 isOjos Boolean required,  
895 itemOjos TextBlob,  
896 isHematologico Boolean required,  
897 itemHematologico TextBlob,  
898 isVacunas Boolean required,  
899 itemVacunas TextBlob,  
900 isAutoinmunidad Boolean required,  
901 itemAutoinmunidad TextBlob,  
902 isNeurologico Boolean required,  
903 itemNeurologico TextBlob,  
904 isNeuropsicologico Boolean required,  
905 itemNeuropsicologico TextBlob,  
906 isMuscular Boolean required,  
907 itemMuscular TextBlob,  
908 isEsqueletico Boolean required,  
909 itemEsqueletico TextBlob,  
910 isCapacidadFisica Boolean required,  
911 itemCapacidadFisica TextBlob,  
912 isSexualidad Boolean required,  
913 itemSexualidad TextBlob,  
914 isFertilidad Boolean required,  
915 itemFertilidad TextBlob,  
916 isPsicologico Boolean required,  
917 itemPsicologico TextBlob,  
918 isSocial Boolean required,  
919 itemSocial TextBlob,  
920 isDrogas Boolean required,
```

```

921     itemDrogas TextBlob,
922     isInformacion Boolean required,
923     itemInformacion TextBlob,
924     isSegundoTumor Boolean required,
925     itemSegundoTumor TextBlob,
926     observaciones TextBlob,
927     pruebasAportadas TextBlob,
928     pruebasSolicitadas TextBlob,
929     recomendaciones TextBlob,
930     otrasRecomendaciones String maxlength(1000),
931     proxRevision String
932 }
933
934 entity EfectoTardio {
935     fecha LocalDate required,
936     tipoItem String required,
937     subTipoItem String,
938     complicacion String,
939     tratamiento String,
940     itemId Long
941 }
942
943 entity Recomendacion {
944     nombreRecomendacion String required,
945     descripcion String maxlength(500)
946 }
947
948 entity RecomendacionConsulta {
949     otrasRecomendaciones String maxlength(1000)
950 }
951
952 entity ExportacionSpss {
953     fechaExportacion LocalDate required,
954     annoDiagnostico String maxlength(4),
955     sexo Sexo,
956     patologiaOncologica PatologiaOncologica,
957     quimioterapia Boolean,
958     radioterapia Boolean,
959     cirugia Boolean,
960     trasplante Boolean,
961     otrosTratamientos Boolean
962 }
963
964 relationship OneToOne {
965     Paciente{diagnostico} to Diagnostico,
966     Rama{tratamientoRecomendado} to TratamientoRecomendado,
967     Diagnostico{tratamiento} to Tratamiento,
968     Recaida{tratamiento} to Tratamiento,
969     Tratamiento{quimioterapia} to Quimioterapia,
970     Tratamiento{radioterapia} to Radioterapia,
971     Tratamiento{otroTratamiento} to OtroTratamiento,
972     Consulta{constante} to Constante,
973     Consulta{exploracionFisica} to ExploracionFisica,
974     Consulta{cardio} to Cardio,
975     Consulta{neumo} to Neumo,
976     Consulta{endocrinoCrecimiento} to EndocrinoCrecimiento,
977     Consulta{endocrinoPeso} to EndocrinoPeso,
978     Consulta{endocrinoDiabetes} to EndocrinoDiabetes,
979     Consulta{endocrinoTiroides} to EndocrinoTiroides,
980     Consulta{maduracionSexual} to MaduracionSexual,
981     Consulta{nefrourologico} to Nefrourologico,
982     Consulta{digestivo} to Digestivo,

```

```

983     Consulta{higado} to Higado,
984     Consulta{piel} to Piel,
985     Consulta{orl} to ORL,
986     Consulta{dientesBoca} to DientesBoca,
987     Consulta{ojos} to Ojos,
988     Consulta{hematologico} to Hematologico,
989     Consulta{vacunas} to Vacunas,
990     Consulta{autoinmunidad} to Autoinmunidad,
991     Consulta{neurologico} to Neurologico,
992     Consulta{neuropsicologico} to Neuropsicologico,
993     Consulta{muscular} to Muscular,
994     Consulta{esqueletico} to Esqueletico,
995     Consulta{capacidadFisica} to CapacidadFisica,
996     Consulta{sexualidad} to Sexualidad,
997     Consulta{fertilidad} to Fertilidad,
998     Consulta{psicologico} to Psicologico,
999     Consulta{social} to Social,
1000    Consulta{drogas} to Drogas,
1001    Consulta{informacion} to Informacion,
1002    Consulta{segundoTumor} to SegundoTumor,
1003    Consulta{recomendacionConsulta} to RecomendacionConsulta{consulta},
1004    Consulta{informe} to Informe
1005 }
1006
1007 relationship OneToMany {
1008     ComunidadAutonoma{provinciaPais} to ProvinciaPais{comunidadAutonoma required
1009         },
1010     Cie10{cie10Secundario} to Cie10Secundario{cie10(codigo) required},
1011     Nacionalidad{paciente} to Paciente{nacionalidad},
1012     Etnia{paciente} to Paciente{etnia},
1013     ComunidadAutonoma{pacienteDiagnostico} to Paciente{comunidadDiagnostico},
1014     ProvinciaPais{pacienteDiagnostico} to Paciente{provinciaDiagnostico},
1015     ComunidadAutonoma{pacienteActual} to Paciente{comunidadActual},
1016     ProvinciaPais{pacienteActual} to Paciente{provinciaActual},
1017     Hospital{paciente} to Paciente{hospital},
1018     Cie10{diagnostico} to Diagnostico{cie10(codigo) required},
1019     Cie10Secundario{diagnostico} to Diagnostico{cie10Secundario(codigo) required
1020         },
1021     Diagnostico{recaida} to Recaida{diagnostico required},
1022     Protocolo{rama} to Rama{protocolo(identificador) required},
1023     TratamientoRecomendado{farmacoAsociado} to FarmacoAsociado{
1024         tratamientoRecomendado(identificador) required},
1025     Farmaco{farmacoAsociado required} to FarmacoAsociado{farmaco(nombre) required
1026         },
1027     Protocolo{tratamiento} to Tratamiento{protocolo},
1028     Rama{tratamiento} to Tratamiento{rama},
1029     Quimioterapia{farmacoQuimio} to FarmacoQuimio{quimioterapia required},
1030     Farmaco{farmacoQuimio required} to FarmacoQuimio{farmaco(nombre) required},
1031     Tratamiento{cirugia} to Cirugia{tratamiento required},
1032     Tratamiento{trasplante} to Trasplante{tratamiento required},
1033     Trasplante{farmacoTrasplante} to FarmacoTrasplante{trasplante required},
1034     Farmaco{farmacoTrasplante required} to FarmacoTrasplante{farmaco(nombre)
1035         required},
1036     Trasplante{farmacoInmunosupresor} to FarmacoInmunosupresor{trasplante
1037         required},
1038     Farmaco{farmacoInmunosupresor required} to FarmacoInmunosupresor{farmaco(
1039         nombre) required},
1040     Paciente{consulta} to Consulta{paciente},
1041     Consulta{especialidad} to Especialidad{consulta},
1042     Consulta{tratamientoHabitual} to TratamientoHabitual{consulta},
1043     Consulta{pruebaAportada} to PruebaAportada{consulta},
1044     Consulta{pruebaSolicitada} to PruebaSolicitada{consulta},

```

```

1038     Paciente{efectoTardio} to EfectoTardio{paciente},
1039     EscalaKarnofsky{constante} to Constante{escalaKarnofsky(codigo)},
1040     EscalaLansky{constante} to Constante{escalaLansky(codigo) }
1041 }
1042
1043 relationship ManyToMany {
1044     RecomendacionConsulta{recomendacion(nombreRecomendacion)} to Recomendacion{
1045         recomendacionConsulta}
1046 }
1047 paginate Paciente, Protocolo, Farmaco, EfectoTardio, Recomendacion with
1048     pagination
1049 dto * with mapstruct
1050
1051 service all with serviceImpl

```

Listado A.1: JDL correspondiente a todas las entidades de la aplicación

Anexo B

Generación de una aplicación con JHipster

ESTE anexo será una guía que explicará cómo generar con JHipster una aplicación de características similares a ONCOSUP. Aunque no se exploren, se podrán ver las diferentes opciones de generación que ofrece la herramienta.

Antes de comenzar la generación de la aplicación es necesario poner a punto el entorno de trabajo. Se comenzará por ir a la web de Oracle y comprobar que la versión de java instalada es la 8 o superior (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>).

A continuación, se debe acceder al sitio de *Node.js* y descargar la última versión estable (<https://nodejs.org/en/>). No es necesario cambiar nada en la configuración de instalación.

Es necesario instalar *Yarn*, un gestor de dependencias (<https://yarnpkg.com/en/docs/install#windows-stable>).

Finalmente, se debe abrir una consola de comandos para poder realizar la instalación de JHipster con el comando: \$ yarn global add generator-jhipster.

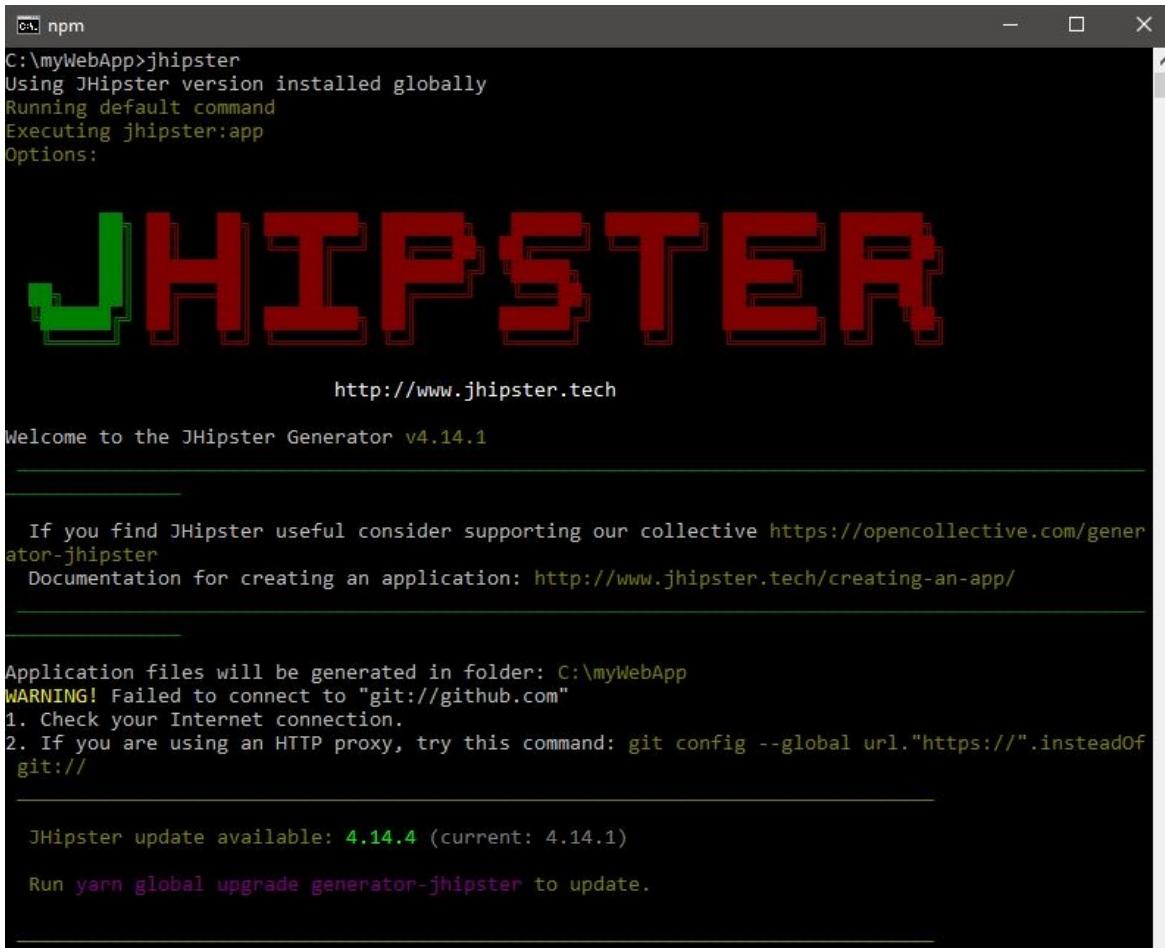
Con estos pasos previos el entorno está listo para generar la primera aplicación con JHipster. Para ello, en una consola de comandos, se crea la carpeta en la que se desea generar la aplicación y se accede a ella (figura B.1).



```
C:\>mkdir myWebApp
C:\>cd myWebApp
C:\myWebApp>
```

Figura B.1: Creación de la carpeta para el proyecto

Una vez se está ubicado en la carpeta en la que se desea generar la aplicación, se ejecuta el comando: \$ jhipster. Se podrá ver como se inicia la generación (figura B.2).



The screenshot shows a terminal window titled 'npm' with the command 'C:\myWebApp>jhipster' entered. The output shows the JHipster logo and version information:

```
C:\myWebApp>jhipster
Using JHipster version installed globally
Running default command
Executing jhipster:app
Options:
```

Below the command line, the JHipster logo is displayed in large red letters with a green 'J'. The URL <http://www.jhipster.tech> is shown below the logo.

The welcome message for the JHipster Generator v4.14.1 follows:

```
Welcome to the JHipster Generator v4.14.1
```

Support and documentation links are provided:

```
If you find JHipster useful consider supporting our collective https://opencollective.com/generator-jhipster
Documentation for creating an application: http://www.jhipster.tech/creating-an-app/
```

A note about application file generation and a git warning:

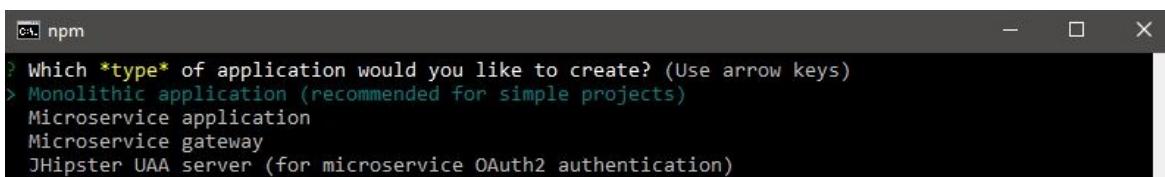
```
Application files will be generated in folder: C:\myWebApp
WARNING! Failed to connect to "git://github.com"
1. Check your Internet connection.
2. If you are using an HTTP proxy, try this command: git config --global url."https://".insteadOf
git://
```

Information about updates and how to run the upgrade command:

```
JHipster update available: 4.14.4 (current: 4.14.1)
Run yarn global upgrade generator-jhipster to update.
```

Figura B.2: Inicio de JHipster

En este momento será necesario responder a diferentes preguntas para realizar la configuración de la aplicación. La primera pregunta es acerca del tipo de aplicación que se desea generar, en el caso de ONCOSUP, se escogió una aplicación monolítica.



The screenshot shows a terminal window titled 'npm' with the question 'Which *type* of application would you like to create? (Use arrow keys)' followed by a list of options:

```
? Which *type* of application would you like to create? (Use arrow keys)
> Monolithic application (recommended for simple projects)
  Microservice application
  Microservice gateway
  JHipster UAA server (for microservice OAuth2 authentication)
```

Figura B.3: Selección del tipo de aplicación

Tras elegir el tipo de aplicación, se solicitarán la carpeta en la que queremos generarla (la carpeta actual si no se especifica nada) y el paquete por defecto de Java. A la de pregunta de si se desea utilizar *JHipster Registry*, en el caso de ONCOSUP, se responde que no. Respecto al tipo de autenticación JHipster ofrece varias opciones, en este ejemplo se elige HTTM Session Authentication, que es la que usa *Spring Security* por defecto. (Figura B.4)

```

c:\ npm
? What is the base name of your application? myWebApp
? What is your default Java package name? com.uclm.mywebapp
? Do you want to use the JHipster Registry to configure, monitor and scale your application? No
? Which *type* of authentication would you like to use? (Use arrow keys)
> JWT authentication (stateless, with a token)
  OAuth 2.0 / OIDC Authentication (stateful, works with Keycloak and Okta)
  HTTP Session Authentication (stateful, default Spring Security mechanism)

```

Figura B.4: Elección del tipo de autenticación

A continuación, se solicitará información acerca de la base de datos que usará la aplicación (figura B.5). En este caso se elige la primera opción, MySQL, ya que se usará MariaDB, pero hay otras opciones como MongoDB o Cassandra que también pueden seleccionarse.

```

c:\ npm
? Which *type* of database would you like to use?
> SQL (H2, MySQL, MariaDB, PostgreSQL, Oracle, MSSQL)
  MongoDB
  Cassandra
  [BETA] Couchbase

```

Figura B.5: Configuración del tipo de Base de Datos

Se seleccionará MariaDB tanto para producción como para desarrollo. Cuando se pregunte si se quiere usar la abstracción de caché de Spring, se responderá que no, ya que no queremos deshabilitar Hibernate (figura B.6).

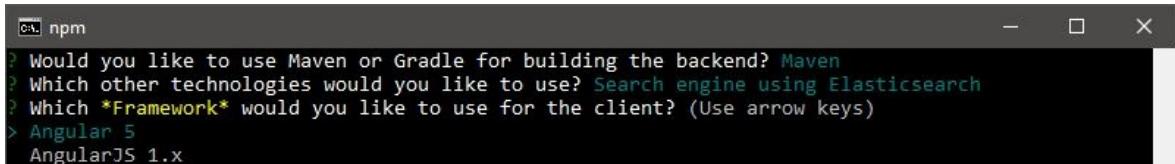
```

c:\ npm
? Which *type* of database would you like to use? SQL (H2, MySQL, MariaDB, PostgreSQL, Oracle, MSSQL)
? Which *production* database would you like to use? MariaDB
? Which *development* database would you like to use? MariaDB
? Do you want to use the Spring cache abstraction?
  Yes, with the Ehcache implementation (local cache, for a single node)
  Yes, with the Hazelcast implementation (distributed cache, for multiple nodes)
  [BETA] Yes, with the Infinispan (hybrid cache, for multiple nodes)
> No (when using an SQL database, this will also disable the Hibernate L2 cache)

```

Figura B.6: Selección acerca de la abstracción de caché

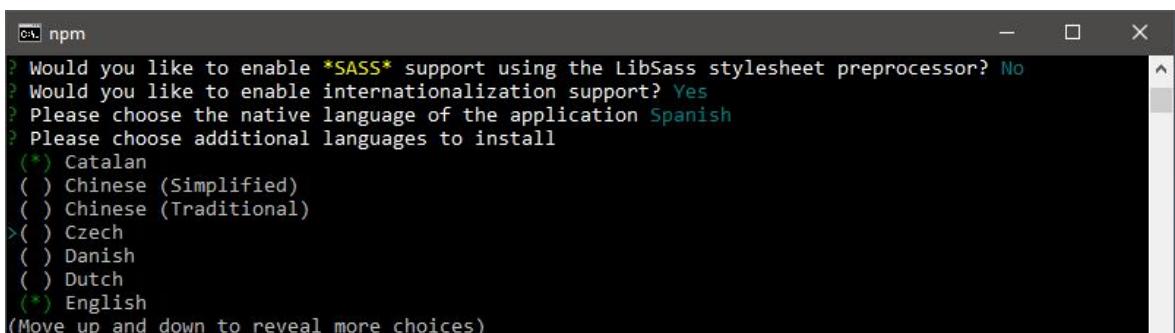
Para la compilación del proyecto en el back-end se usará Maven y respecto a otras tecnologías se seleccionará el motor de búsqueda con ElasticSeach. Para ONCOSUP no hacía falta ninguna otra tecnología, pero algunas opciones interesantes permiten añadir un login social, para acceder usando cuentas de Google, Facebook o Twitter o añadir un servicio de mensajes asíncronos. Como framework se seleccionará Angular5, aunque hay soporte para todas las versiones de AngularJS.(figura B.7)



```
npm
? Would you like to use Maven or Gradle for building the backend? Maven
? Which other technologies would you like to use? Search engine using Elasticsearch
? Which *Framework* would you like to use for the client? (Use arrow keys)
> Angular 5
  AngularJS 1.x
```

Figura B.7: Versiones de Angular a elegir

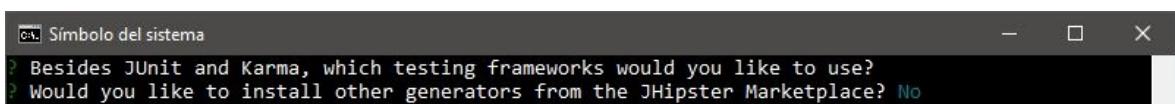
Se elegirá que no a la pregunta sobre activar soporte para SASS. En cuanto a la internacionalización, en este caso, si queremos tenerla. Como idioma nativo seleccionamos español, y como idiomas adicionales catalán e inglés. La internacionalización permite añadir tantos idiomas como se desee de la lista. (figura B.8)



```
npm
? Would you like to enable *SASS* support using the LibSass stylesheet preprocessor? No
? Would you like to enable internationalization support? Yes
? Please choose the native language of the application Spanish
? Please choose additional languages to install
(*) Catalan
( ) Chinese (Simplified)
( ) Chinese (Traditional)
>( ) Czech
( ) Danish
( ) Dutch
(*) English
(Move up and down to reveal more choices)
```

Figura B.8: Configuración del idioma principal y secundarios

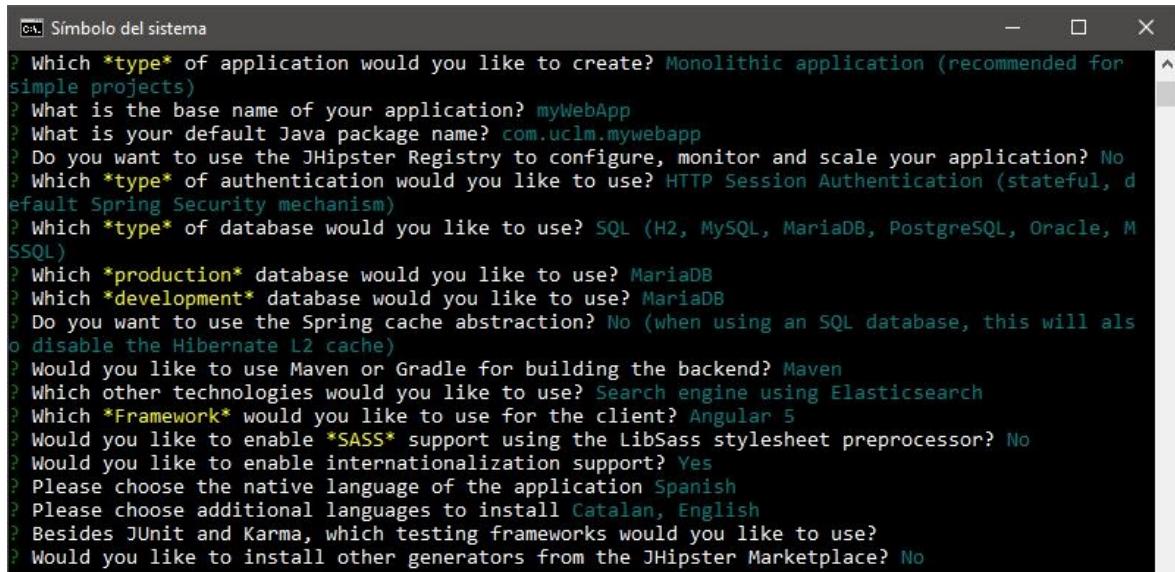
En el caso de este proyecto el presupuesto no permite tener equipo de testing, así no se seleccionará ninguna herramienta adicional de testing. Y por último responderemos que no a instalar otros generadores de la tienda de JHipster. (figura B.9)



```
Symbolo del sistema
? Besides JUnit and Karma, which testing frameworks would you like to use?
? Would you like to install other generators from the JHipster Marketplace? No
```

Figura B.9: Herramientas de testing disponibles

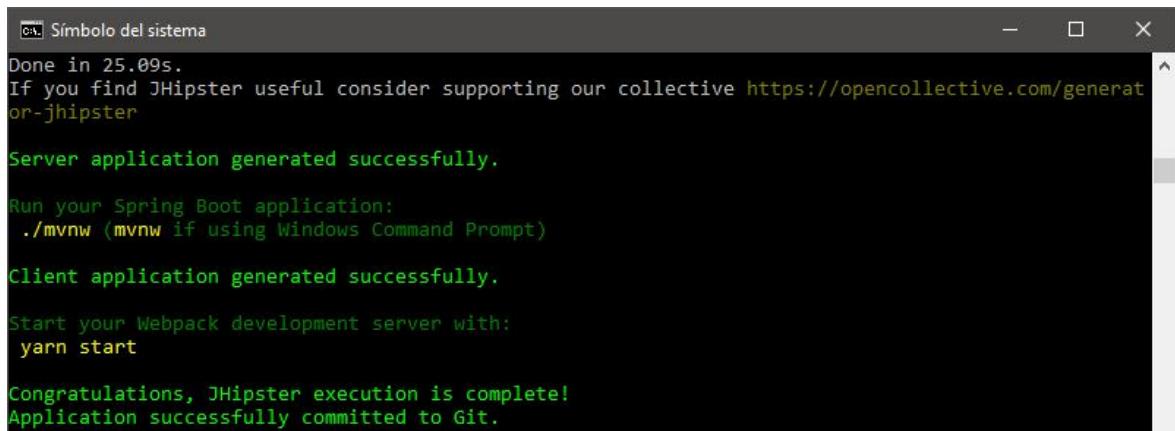
La figura B.10, a continuación, muestra un resumen de la configuración seleccionada para ONCOSUP.



```
? Which *type* of application would you like to create? Monolithic application (recommended for simple projects)
? What is the base name of your application? myWebApp
? What is your default Java package name? com.uclm.mywebapp
? Do you want to use the JHipster Registry to configure, monitor and scale your application? No
? Which *type* of authentication would you like to use? HTTP Session Authentication (stateful, default Spring Security mechanism)
? Which *type* of database would you like to use? SQL (H2, MySQL, MariaDB, PostgreSQL, Oracle, MySQL)
? Which *production* database would you like to use? MariaDB
? Which *development* database would you like to use? MariaDB
? Do you want to use the Spring cache abstraction? No (when using an SQL database, this will also disable the Hibernate L2 cache)
? Would you like to use Maven or Gradle for building the backend? Maven
? Which other technologies would you like to use? Search engine using Elasticsearch
? Which *Framework* would you like to use for the client? Angular 5
? Would you like to enable *SASS* support using the LibSass stylesheet preprocessor? No
? Would you like to enable internationalization support? Yes
? Please choose the native language of the application Spanish
? Please choose additional languages to install Catalan, English
? Besides JUnit and Karma, which testing frameworks would you like to use?
? Would you like to install other generators from the JHipster Marketplace? No
```

Figura B.10: Resumen de la configuración para la aplicación

Con esto, se habrá completado la configuración de la aplicación y se mostrará en pantalla algo similar a lo que se ve a continuación: la instalación de los lenguajes, instalación y resolución de dependencias de paquetes. Finalmente, tras todas las instalaciones que JHipster necesita hacer, mostrará que ha generado de forma exitosa, tanto el cliente como el servidor(figura B.11).



```
Done in 25.09s.
If you find JHipster useful consider supporting our collective https://opencollective.com/generator-jhipster

Server application generated successfully.

Run your Spring Boot application:
./mvnw (mvnw if using Windows Command Prompt)

Client application generated successfully.

Start your Webpack development server with:
yarn start

Congratulations, JHipster execution is complete!
Application successfully committed to Git.
```

Figura B.11: Resultado de la generación

La aplicación ya está generada, pero no hay una estructura de entidades definida, es necesario importar un JDL. En los capítulos 3 y 5, se trata acerca del JDL, cómo crearlo y algunas de sus características más relevantes. Para este ejemplo se usará un JDL de ejemplo

que se puede encontrar en la página de JHipster. La estructura de entidades es la que muestra la figura B.12

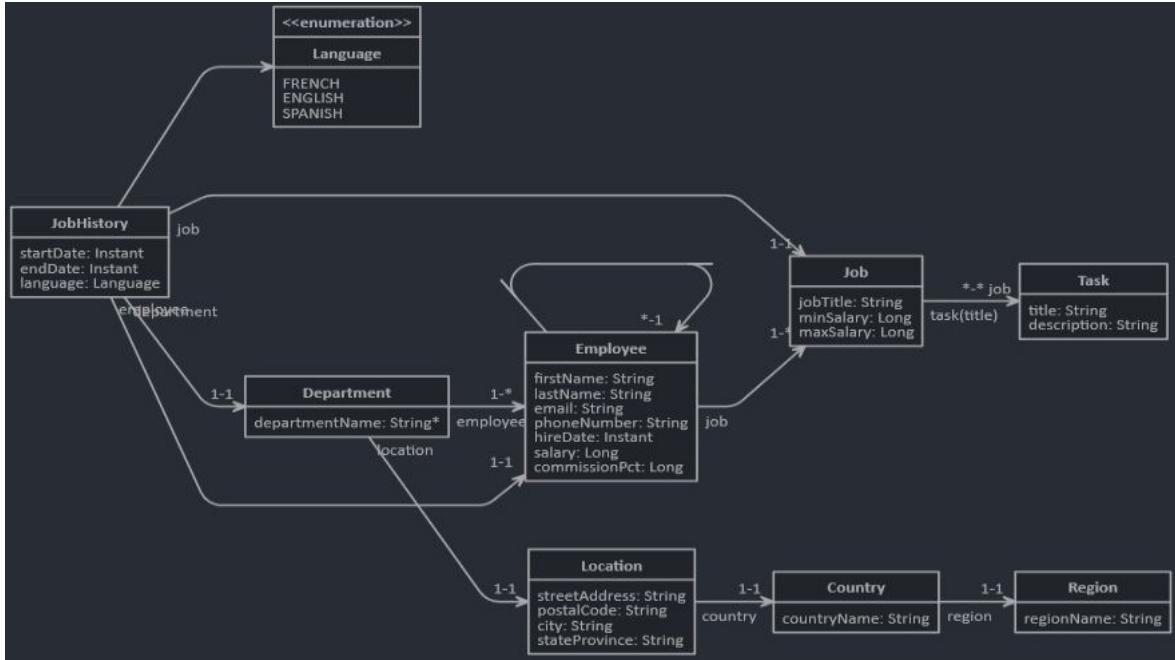


Figura B.12: Estructura de entidades usada para la aplicación

Tras haber creado nuestro JDL, ya sea en un editor de texto o en la herramienta que proporciona JHipster, JDL-Studio, es necesario importarlo en nuestra aplicación recién generada. Para ello, en la carpeta de nuestro proyecto, donde también hemos dejado nuestro JDL, ejecutamos el comando: `$ jhipster import-jdl tuJDL.jh`.

En la consola se irán mostrando las entidades creadas, y en caso de que haya sobrescripciones, como en el caso de los elementos que ya estaban generados, se preguntará si se desea sobrescribir (figura B.13). Para este ejemplo elegimos sobrescribir, pero al realizar modificaciones, es buena opción pensar qué archivos conviene sobrescribir y cuáles no.

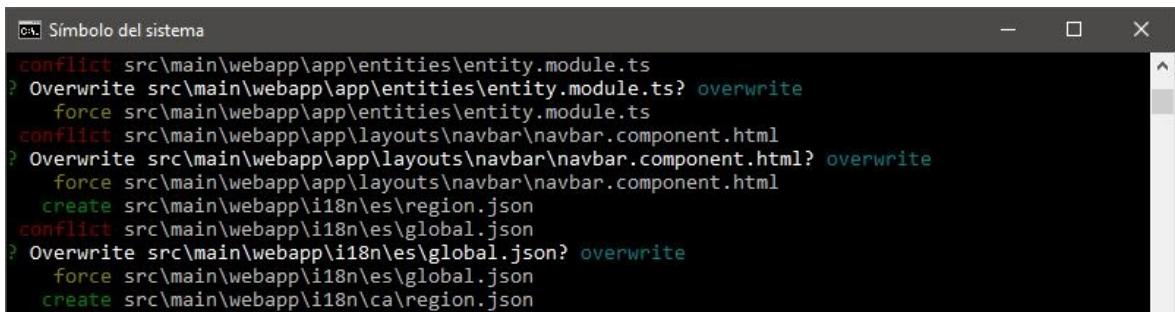
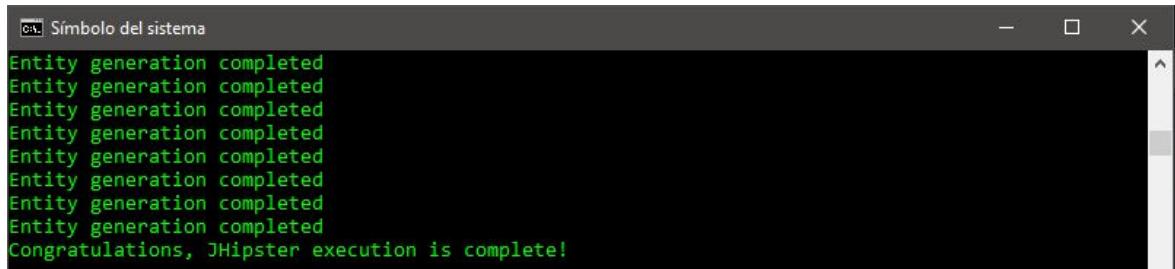


Figura B.13: Importación del JDL

Una vez se ha respondido a todas las preguntas acerca de sobreescrituras, podremos ver algo similar a lo que muestra la figura B.14, si todo ha ido correctamente.



```
Símbolo del sistema
Entity generation completed
Congratulations, JHipster execution is complete!
```

Figura B.14: Resultado de la generación de entidades

La aplicación ya es funcional, se puede compilar y acceder a ella a través del puerto 8080. Para compilar y lanzar la aplicación, tan solo es necesario ejecutar el comando `$ mvnw` en la ubicación del proyecto. Si hemos configurado la base de datos podremos autenticarnos y navegar por la aplicación, en caso contrario, tan sólo podremos verla página de inicio (figura B.15).



Figura B.15: Página de inicio de la aplicación

Referencias

- [ang18] Angular. One framework. Mobile and desktop, último acceso en 2018. url: <https://angular.io/>.
- [clo18] GitHub, Cloc, último acceso en 2018. url: <https://github.com/AlDanial/cloc>.
- [for18] Formularios de Google, último acceso en 2018. url: <http://google.es/intl/es/forms/about/>.
- [jdl18a] JDL-Studio, último acceso en 2018. url: <https://www.start.jhipster.tech/jdl-studio/>.
- [jdl18b] Jhipster Domain Language (JDL), último acceso en 2018. url: <https://www.jhipster.tech/jdl/>.
- [jhi18] Greetings, Java Hipster. JHipster, último acceso en 2018. url: <https://www.jhipster.tech>.
- [Kni07] Henrik Kniberg. *Scrum and XP from the Trenches*. C4Media, 2007.
- [lda18] RFC 2251 - Lightweight Directory Access Protocol (v3), último acceso en 2018. url: <http://www.faqs.org/rfcs/rfc2251.html>.
- [liq18] Liquibase official Site, último acceso en 2018. url: <https://www.liquibase.org/>.
- [lop18] Reglamento de la LOPD, último acceso en 2018. url: https://www.agpd.es/portalwebAGPD/canaldocumentacion/informes_juridicos/reglamento_lopd/index-ides-idphp.php.
- [mar18] The MariaDB Foundation – Supporting continuity and open collaboration in the MariaDB ecosystem, último acceso en 2018. url: <https://mariadb.org/>.
- [mdb18] MariaDB versus MySQL - Features, último acceso en 2018. url: <https://mariadb.com/kb/en/library/mariadb-vs-mysql-features/>.

- [mys18] Commercial License for OEMs, ISVs and VARs, último acceso en 2018. url: <https://www.mysql.com/about/legal/licensing/oem/>.
- [not13] SCRUM: Framework not methodology, 2013. url: <https://guntherverheyen.com/2013/03/21/scrum-framework-not-methodology/>.
- [not18] La Fundación Ramón Areces impulsa el primer registro de supervivientes a largo plazo de cáncer infantil en España, 2018. url: <http://www.fundacionareces.es/fundacionareces/portal.do?TR=C&IDR=2024>.
- [Ras10] Jonathan Rasmusson. *The Agile Samurai*. Pragmatic Bookshelf, 2010.
- [Ras18] Jonathan Rasmusson. The Agile Inception Deck, último acceso en 2018. url: <https://agilewarrior.wordpress.com/2010/11/06/the-agile-inception-deck/>.
- [Smi18] Mark K. Smith. Bruce W. Tuckman – forming, storming norming and performing in groups, último acceso en 2018. url: <http://infed.org/mobi/bruce-w-tuckman-forming-storming-norming-and-performing-in-groups/>.
- [spr18a] Spring Boot, último acceso en 2018. url: <https://spring.io/projects/spring-boot>.
- [spr18b] Spring Security, último acceso en 2018. url: <https://projects.spring.io/spring-security/>.
- [SS17] Ken Schwaber y Jeff Sutherland. *The Scrum Guide*. 2017.
- [tom18] Apache Tomcat, último acceso en 2018. url: <http://tomcat.apache.org/>.
- [web18] Mozilla Developer Network, WebSockets, último acceso en 2018. url: <https://developer.mozilla.org/es/docs/WebSockets-840092-dup>.

Este documento fue editado y tipografiado con L^AT_EX empleando
la clase **esi-tfg** (versión 0.20180620) que se puede encontrar en:
<https://bitbucket.org/arco-group/esi-tfg>

[respeta esta atribución al autor]

