

Proyecto Fin de Grado
Grado en Ingeniería de Tecnologías de
Telecomunicación

Servicio web para el acceso a información sanitaria
de pacientes renales

Autor: Miguel Pedrera Jiménez

Tutor: Laura M. Roa Romero

Dep. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Servicio web para el acceso a información sanitaria de pacientes renales

Autor:

Miguel Pedrera Jiménez

Tutor:

Laura M. Roa Romero

Catedrática de Universidad

Dep. de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2016

Trabajo Fin de Grado: Servicio web para el acceso a información sanitaria de pacientes renales

Autor: Miguel Pedrera Jiménez

Tutor: Laura M. Roa Romero

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2016

El Secretario del Tribunal

*A Loli y Adrián, siempre seréis
parte de mí.*

Agradecimientos

Con este trabajo se cierra una etapa de mi vida, y no me gustaría hacerlo sin acordarme y dar las gracias a todas esas personas que me han acompañado a lo largo de este duro camino, convirtiéndolo en un agradable paseo.

En primer lugar, a mis padres, Carmen y Miguel, y a mi hermana Marta, por darme todo lo que han tenido y más. Por ser un ejemplo en el que fijarme cada día, sin el cuál no estaría ahora mismo escribiendo estas líneas.

A mi familia, los que están y los que no, a mis abuelos, mis tíos, mis primos... porque no hay nada más importante que la familia, y cuando algún miembro ya no está, todo cambia para siempre.

A mis amigos de siempre, Pablo, Jose, Natalia, Lorena, Jorge y Víctor; por demostrarme cada día que la amistad no tiene fecha de caducidad.

A los festivaleros, Sito, Fati, Curro, Marta y Trini; porque la música se vuelve especialmente valiosa cuando te hace descubrir personas como ellos.

A todas las personas que he conocido durante mi etapa en la escuela, en especial a Soriano, Fran, Richy, Marchal, Dani, Jairo, Bole, Ana, Franxi, Ale y Caye, porque estos años sin ellos no habrían sido lo mismo.

A mis ex-compañeros de Everis, en especial a Jorge y Miguel, por enseñarme que de unas prácticas no sólo te llevas conocimientos, sino también grandes amigos.

A los profesores, por todo lo que me han enseñado, en particular a Laura María Roa, por confiar en mí para este proyecto, y a Jorge Calvillo, por la inestimable atención y ayuda prestada durante la realización del mismo.

A todos, muchas gracias.

Miguel Pedrera Jiménez

Sevilla, 2016

Resumen

La e-Salud alude a la práctica de cuidados sanitarios apoyada en tecnologías de la información y las comunicaciones, introduciendo dichas tecnologías en la vida de los pacientes, y haciendo que participen de forma activa en el cuidado de su salud.

Con un número cada vez mayor de sistemas de e-Salud trabajando conjuntamente, se hace necesario que interoperen e intercambien información con el fin de mejorar la atención a los pacientes, reducir costes y hacer sostenibles los complejos sistemas sanitarios.

Hacer que sistemas independientes interoperen no es una tarea sencilla pues dichos sistemas pueden trabajar internamente de maneras muy distintas. Por ello se hace necesario que, de cara a comunicarse con otros sistemas, se utilice un *lenguaje* común que todos entiendan. Ese lenguaje común puede ser convenido entre cada par de sistemas (protocolo propietario) o adoptado desde un estándar de interoperabilidad en salud.

Este trabajo tiene por objeto, el diseño y desarrollo de un servicio web, siguiendo la especificación Fast Healthcare Interoperability Resources (FHIR) de la organización Health Level 7 (HL7), para el acceso a información sanitaria almacenada en la base de datos de un sistema de e-Salud para monitorización remota de pacientes renales. Aunque el dominio de aplicación del servicio es la patología renal, los resultados son extensibles a cualquier otro sistema que gestione información de pacientes crónicos.

Previo al diseño del servicio web, se ha llevado a cabo un profundo estudio del estándar, así como de su estado actual y los principales sistemas que lo utilizan o planean utilizar, con el objetivo de enmarcar nuestro trabajo en el contexto actual.

Abstract

e-Health alludes to healthcare supported in TICs, introducing these technologies into patient's lifes, and making them participate in an active way in their own health care.

With a growing number of e-Salud systems working all together, it is increasingly necessary that they are able to interoperate and share information, in order to improve patient's care, reduce costs and help the complex sanitary systems to be sustainable.

Getting independent systems into working together it's not an easy task, because those systems could internally work in so many different ways. It is because of that sort of things that we need a common language for the systems to communicate, a language understandable by all of them. That common language could be agreed between each pair of systems (proprietary protocol) or adopted from an interoperability standard in health.

The aim of this work is the design and development of a web service, following the Fast Healthcare Interoperability Resources (FHIR) specification from the HealthLevel 7(HL7) organization, for access the healthcare information stored in an e-Salud system for remote monitoring of renal patients. Although the service application domain is renal pathology, results could be extended to any other system used to manage chronicle patient's information.

Previous to the design of web service, it has conducted a thorough study of the standard, its current status and major systems that use or plan to use, in order to frame our work in the current context.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xvii
Índice de Figuras	xix
Notación	xxi
1 Introducción	1
1.1. <i>Motivación y Objetivos</i>	1
1.2. <i>Metodología de trabajo</i>	3
1.3. <i>Plan de trabajo</i>	3
2 Estado del arte	5
2.1. <i>La Enfermedad Renal Crónica</i>	5
2.1.1. <i>Diálisis Peritoneal, beneficios y problemática</i>	6
2.2. <i>El sistema e-Nefro</i>	6
2.3. <i>Interoperabilidad en Salud</i>	7
2.3.1. <i>El estándar HL7 FHIR</i>	7
2.4. <i>Material y método</i>	14
3 Resultados	15
3.1. <i>Revisión literaria sobre la especificación HL7 FHIR</i>	15
3.1.1. <i>Metodología de búsqueda</i>	15
3.1.2. <i>Resultados</i>	16
3.1.3. <i>Discusión</i>	18
3.1.4. <i>Conclusiones</i>	19
3.2. <i>Identificación y modelado de los objetos de información de e-Nefro</i>	20
3.2.1. <i>Identificación de los objetos de información</i>	20
3.2.2. <i>Modelado de los objetos de información</i>	22
3.3. <i>Identificación y modelado de los Recursos FHIR</i>	36
3.3.1. <i>Identificación de los Recursos FHIR</i>	37
3.3.2. <i>Modelado de los Recursos FHIR</i>	38
3.4. <i>Definición de los métodos de acceso a la información</i>	47
3.4.1. <i>Forma de acceso a la información</i>	48
3.4.2. <i>Lista de métodos de acceso a los recursos FHIR</i>	48
3.5. <i>Implementación del servicio web</i>	58
3.5.1. <i>Diseño del servicio web</i>	58
3.5.2. <i>Desarrollo del servicio web</i>	59
3.5.3. <i>Validación del servicio web</i>	67
4 Conclusiones y líneas futuras	85
4.1. <i>Líneas futuras</i>	86
5 Referencias	87

Índice de Tablas

Tabla 3-1. Objeto Paciente	26
Tabla 3-2. Objeto Personal sanitario	27
Tabla 3-3. Objeto Organización	28
Tabla 3-4. Objeto Cita	29
Tabla 3-5. Objeto Actividad Toma de medicación	30
Tabla 3-6. Objeto Registro Toma de medicación	31
Tabla 3-7. Objeto Actividad Medida de variable	31
Tabla 3-8. Objeto Registro Medida de variable	32
Tabla 3-9. Objeto Actividad Toma de imagen	33
Tabla 3-10. Objeto Registro Toma de imagen	34
Tabla 3-11. Objeto Incidencia	34
Tabla 3-12. Objeto Plan de cuidados	35
Tabla 3-13. Objeto Periodo de seguimiento	35
Tabla 3-14. Lista de recursos FHIR	37
Tabla 3-15. Recurso Patient	38
Tabla 3-16. Recurso Practitioner	38
Tabla 3-17. Recurso Organization	39
Tabla 3-18. Recurso Appointment	39
Tabla 3-19. Recurso Encounter	40
Tabla 3-20. Recurso Location	41
Tabla 3-21. Recurso MedicationOrder	41
Tabla 3-22. Recurso MedicationStatement	42
Tabla 3-23. Recurso ProcedureRequest	43
Tabla 3-24. Recurso Observation	44
Tabla 3-25. Recurso DetectedIssue	45
Tabla 3-26. Recurso CarePlan	46
Tabla 3-27. Recurso EpisodeOfCare	47
Tabla 3-28. Recurso Condition	47
Tabla 3-29. Método para el recurso Patient (I)	48
Tabla 3-30. Método para el recurso Patient (II)	49
Tabla 3-31. Método para el recurso Practitioner	49

Tabla 3-32. Método para el recurso Appointment (I)	50
Tabla 3-33. Método para el recurso Appointment (II)	50
Tabla 3-34. Método para el recurso Appointment (III)	50
Tabla 3-35. Método para el recurso Appointment (IV)	51
Tabla 3-36. Método para el recurso Encounter (I)	51
Tabla 3-37. Método para el recurso Encounter (II)	51
Tabla 3-38. Método para el recurso Encounter (III)	52
Tabla 3-39. Método para el recurso Encounter (IV)	52
Tabla 3-40. Método para el recurso MedicationOrder (I)	53
Tabla 3-41. Método para el recurso MedicationOrder (II)	53
Tabla 3-42. Método para el recurso MedicationStatement (I)	54
Tabla 3-43. Método para el recurso MedicationStatement (II)	54
Tabla 3-44. Método para el recurso ProcedureRequest (I)	54
Tabla 3-45. Método para el recurso ProcedureRequest (II)	55
Tabla 3-46. Método para el recurso Observation (I)	55
Tabla 3-47. Método para el recurso Observation (II)	56
Tabla 3-48. Método para el recurso DetectedIssue (I)	56
Tabla 3-49. Método para el recurso DetectedIssue (II)	56
Tabla 3-50. Método para el recurso CarePlan	57
Tabla 3-51. Método para el recurso EpisodeOfCare	57

Índice de Figuras

Figura 1-1. Funcionalidad del servicio web	2
Figura 1-2. Metodología incremental	3
Figura 2-1. Estructura del Recurso FHIR	9
Figura 2-2. Tipos de datos en FHIR	10
Figura 3-1. Diagrama de flujo de la selección de artículos	16
Figura 3-2. Distribución temporal de los artículos sobre FHIR	16
Figura 3-3. Clasificación en función del área médica donde se aplica FHIR	17
Figura 3-4. Clasificación según el tipo de sistema donde se aplica FHIR	18
Figura 3-5. Funcionalidad del Paciente/Cuidador en e-Nefro	20
Figura 3-6. Funcionalidad del Profesional en e-Nefro	21
Figura 3-7. Modelo de datos de e-Nefro: Información personal de usuario	23
Figura 3-8. Modelo de datos de e-Nefro: Organización del profesional	23
Figura 3-9. Modelo de datos de e-Nefro: Cargo y especialidad del profesional	24
Figura 3-10. Modelo de datos de e-Nefro: Actividades	24
Figura 3-11. Modelo de datos de e-Nefro: Registros	25
Figura 3-12. Modelo de datos de e-Nefro: Relación profesional-paciente	25
Figura 3-13. Modelo de datos de e-Nefro: Relación cuidador-paciente	26
Figura 3-14. Relaciones entre objetos de información	36
Figura 3-15. Referencias en el recurso Practitioner	39
Figura 3-16. Referencias en el recurso Appointment	40
Figura 3-17. Referencias en el recurso Encounter	41
Figura 3-18. Referencias en el recurso MedicationOrder	42
Figura 3-19. Referencias en el recurso MedicationStatement	43
Figura 3-20. Referencias en el recurso ProcedureRequest	44
Figura 3-21. Referencias en el recurso Observation	45
Figura 3-22. Referencias en el recurso DetectedIssue	45
Figura 3-23. Referencias en el recurso CarePlan	46
Figura 3-24. Referencias en el recurso EpisodeOfCare	47
Figura 3-25. Sintaxis GET/POST de la operación Search	48
Figura 3-26. Diagrama de secuencia de la funcionalidad del servicio web	58
Figura 3-27. Diagrama de paquetes de la estructura del servicio web	59
Figura 3-28- Paquete servlet	60

Figura 3-29. Paquete narrative	61
Figura 3-30. Paquete bean	62
Figura 3-31. Paquete SQL	63
Figura 3-32. Paquete model	64
Figura 3-33. Paquete provider	65
Figura 3-34. Paquete confíg	67
Figura 3-35. Módulo de validación: Página principal	68
Figura 3-36. Módulo de validación: Página de recurso	68
Figura 3-37. Módulo de validación: Página de resultado	69
Figura 3-38. Petición/respuesta consulta de un paciente	70
Figura 3-39. Elemento narrativo de recurso Patient	71
Figura 3-40. Petición/Respuesta consulta de un paciente inexistete.	72
Figura 3-41. Petición/Respuesta consulta de pacientes a cargo de un profesional	72
Figura 3-42. Petición/Respuesta consulta de un profesional	73
Figura 3-43. Elemento narrativo de recurso Practitioner	75
Figura 3-44. Petición/Respuesta consulta de un profesional inexistente	75
Figura 3-45. Petición/Respuesta consulta de las citas de un profesional	76
Figura 3-46. Petición/Respuesta consulta de los encuentros de un profesional	78
Figura 3-47. Petición/Respuesta acceso a método no implementado	80
Figura 3-48. Cliente FHIR Android: Interfaz	81
Figura 3-49. Cliente FHIR Android: Muestra de los pacientes obtenidos	82
Figura 3-50. Cliente FHIR Android: Notificación de no infomación	83

Notación

FHIR	Fast Health Interoperability Resources
CDA	Clinical Document Architecture
HL7	Health Level Seven
ERC	Enfermedad Renal Crónica
TSR	Tratamiento Sustitutivo Renal
TX	Trasplante Renal
HD	Hemodiálisis
DP	Diálisis Peritoneal
DSTU	Draft Standard for Trial Use
URI	Uniform Resource Identifier
API	Application Programming Interface
CASE	Computer Aided Software Engineering
UML	Unified Modeling Language
POM	Project Object Model
URI	Uniform Resource Identifier

1 INTRODUCCIÓN

“El conocimiento es poder. El conocimiento compartido es poder multiplicado.”

Robert Noyce

En este primer capítulo, se exponen los motivos y objetivos de este trabajo, y cómo se ha llevado a cabo dicha tarea, explicando y justificando la metodología de trabajo escogida y desglosando el plan de trabajo que se ha diseñado y llevado a cabo.

1.1. Motivación y Objetivos

La e-Salud alude a la práctica de cuidados sanitarios apoyada en tecnologías de la información y las comunicaciones, introduciendo dichas tecnologías en la vida de los pacientes, y haciendo que participen de forma activa en el cuidado de su salud. Se busca hacer de la atención sanitaria un proceso cada vez más eficiente.

Con un gran número de sistemas de e-Salud trabajando conjuntamente, el intercambio de información entre ellos es un pilar fundamental en los procesos de atención a los pacientes, gestión y generación de conocimiento clínico, reducción de costes y sostenibilidad del sistema sanitario. No obstante, con frecuencia se desarrollan sistemas de información sanitarios que no contemplan ese intercambio de información con otros, lo que se traduce en una asistencia fragmentada y lo que se han venido a llamar las islas de información.

El interés de este proyecto surge de la necesidad de hacer accesible la información almacenada en la base de datos de un sistema de e-Salud para el tratamiento de pacientes renales crónicos (sistema e-Nefro) a otros sistemas. Entre la información almacenada en dicha base de datos se puede encontrar: tratamientos seguidos por los pacientes, citas concertadas, registros e incidencias de las actividades que han realizado en el tratamiento de su enfermedad. Si se dejara esta información aislada en el ámbito de e-Nefro, estaríamos perdiendo parte del potencial del sistema, y estaríamos lejos de lograr una implementación eficiente para el tratamiento de la enfermedad renal crónica.

Se busca, por tanto, una integración de e-Nefro con el resto de sistemas para así hacer posible que toda la información que genera el sistema pueda ser extraída y utilizada por otros. Para ello se diseñará y desarrollará un servicio web siguiendo la reciente especificación FHIR de HL7 para el intercambio de información sanitaria. En la Figura 1-1 se muestra el esquema que pretendemos alcanzar, donde nuestro servicio web hará de intermediario entre la base de datos de e-Nefro y el resto de clientes FHIR externos.

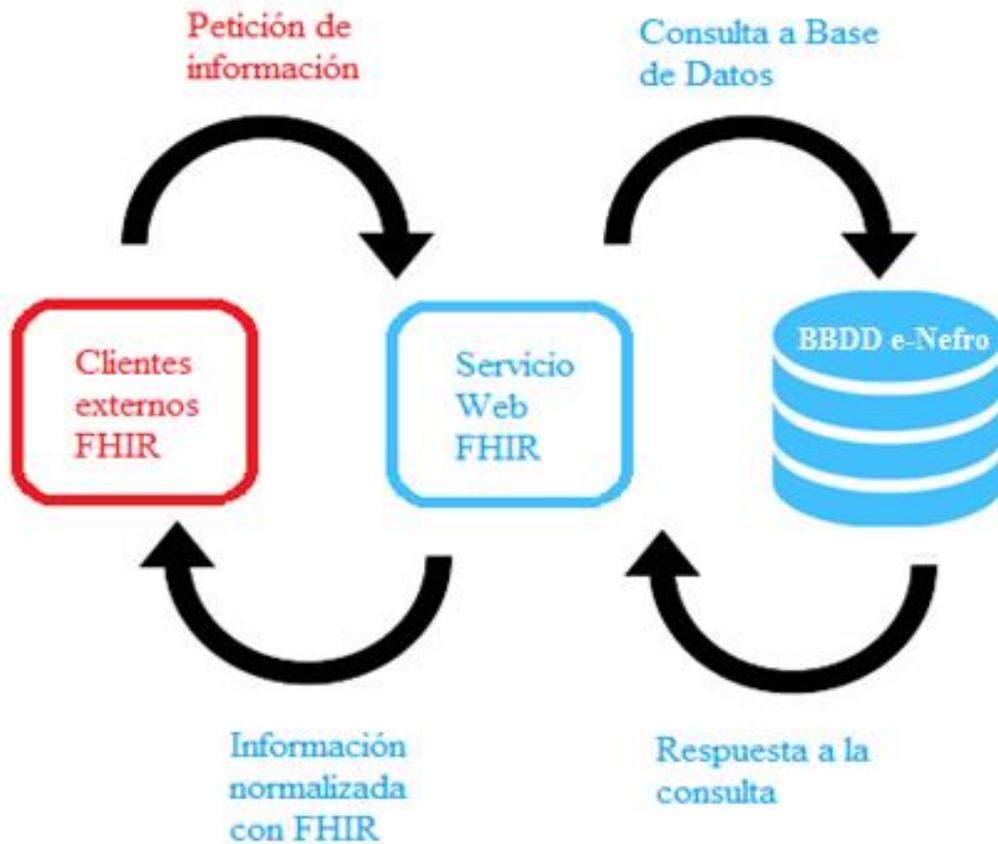


Figura 1-1. Funcionalidad del servicio web

Frente a otros estándares, y como se describe en el apartado 2.3.1, se ha escogido FHIR porque está llamado a ser una de las especificaciones de referencia para el intercambio de información en la próxima generación de la e-Salud pues busca reducir de forma importante los tiempos y, por consiguiente, los costes de los proyectos de integración, al ser diseñado para ser fácil de aprender, desarrollar e implementar, a la vez que combina las características más interesantes de los estándares actualmente en uso.

En definitiva, podemos resumir los objetivos del proyecto en los siguientes puntos:

- Estudio del estándar HL7 FHIR y su aplicación en sistemas de e-Salud para el tratamiento de enfermedades crónicas.
- Diseño e implementación de un servicio web que permita el acceso a la información de la base de datos de e-Nefro a cualquier otro sistema que adopte el estándar anteriormente mencionado.
- Extracción de conclusiones y valoración de la utilización del estándar para sistemas de este tipo.

1.2. Metodología de trabajo

Para la realización del trabajo se va a llevar a cabo una metodología iterativa y creciente, más conocida como metodología incremental. Esta metodología es muy empleada en el desarrollo ágil de software donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

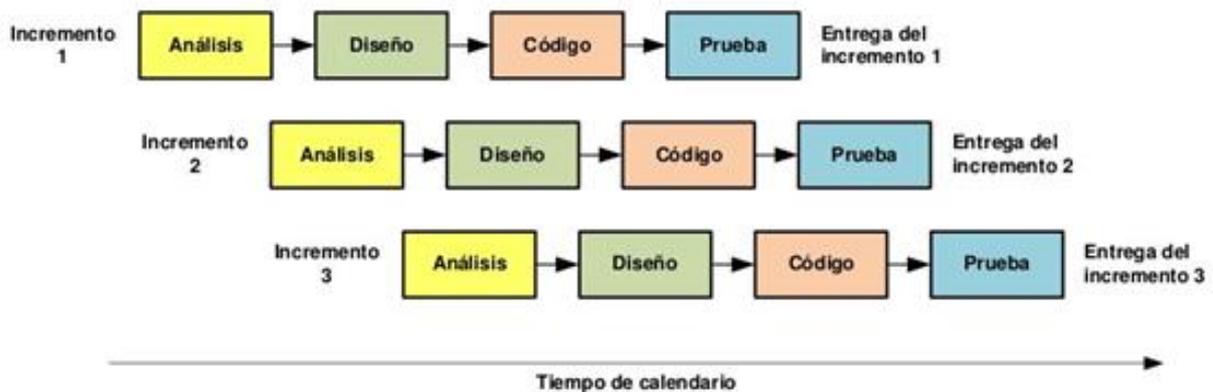


Figura 1-2. Metodología incremental

Esta metodología de trabajo difiere del tradicional desarrollo en cascada, en el que llevaríamos a cabo las tareas en el orden lógico de realización. De este modo, una vez estudiado el problema médico y técnico a resolver, se realizarán las tareas de diseño, implementación y pruebas, de manera incremental.

Con esto se pretende, por un lado, obtener beneficio de lo aprendido a lo largo de cada iteración y aplicarlo en el siguiente, es decir, retroalimentación, y, por otro lado, detectar, gracias a las comprobaciones y pruebas a cada bloque, los errores lo antes posible, y poder solucionarlos con la menor repercusión posible en el trabajo realizado hasta el momento.

1.3. Plan de trabajo

Un proyecto de este tipo conlleva una serie de tareas para su comprensión y realización:

1. Estudio y comprensión de la enfermedad renal crónica, así como de sus principales tratamientos domiciliarios: prediálisis, diálisis peritoneal y tratamiento conservador.
2. Estudio y comprensión del problema técnico a resolver con nuestro sistema: la necesidad de interoperabilidad entre aplicaciones sanitarias, y cómo se logra.
3. Estudio y comprensión del estándar de comunicación en salud FHIR y su API, y la aplicación de éste en un sistema de e-Salud para pacientes crónicos.
4. Identificación de los objetos de información existentes en la aplicación, tras el estudio de la aplicación e-Nefro y su base de datos.

Las tareas que vienen a continuación, seguirán una metodología iterativa y creciente, llevándose a cabo en bloque para cada objeto de información identificado en el punto anterior.

5. Modelado de los objetos de información, identificando sus atributos y relacionándolos con la base de datos de la aplicación.
6. Establecer la correspondencia entre los objetos de información definidos y los recursos ofrecidos por FHIR y sus campos.

7. Solución técnica para la creación de los objetos de información definidos anteriormente, esto es, creación de las clases que los definen y los métodos para la extracción de información de la base de datos.
8. Diseño de los métodos de consulta que se ofrecerán al resto de sistemas para el acceso a la información.
9. Implementación de los métodos para la solicitud de los recursos FHIR con la información clínica, extraída donde se lleva a cabo el mapeo de la información obtenida a los recursos FHIR, tal como los habíamos establecido anteriormente.
10. Implementación de los puntos anteriores en el servicio web.
11. Evaluación y pruebas.

2 ESTADO DEL ARTE

“Nunca consideres el estudio como una obligación, sino como una oportunidad de penetrar en el maravilloso mundo del saber.”

Albert Einstein

Es necesario hablar del contexto sanitario y técnico en el que se enmarca el presente trabajo. En primer lugar, se va a ver en qué consiste la enfermedad renal crónica, qué terapias existen actualmente para ella y la problemática que conllevan, centrándonos en la prediálisis y la diálisis peritoneal.

También se explicará con mayor profundidad qué es y cómo funciona el sistema e-Nefro, ya mencionado anteriormente, y qué solución ofrece a los pacientes renales crónicos.

Por último, se tratará el tema de la interoperabilidad en el ámbito sanitario y el estándar HL7 FHIR.

2.1. La Enfermedad Renal Crónica

La Enfermedad Renal Crónica (ERC) es una patología consistente en la pérdida de las funciones vitales que desempeñan los riñones [1]. Si este deterioro de la función renal es importante pasa a denominarse Enfermedad Renal Crónica Avanzada (ERCA) y no es posible seguir tratándola de forma conservadora (medicación y medidas higiénico-dietéticas), siendo necesario pasar a lo que se conoce como Tratamiento Sustitutivo Renal (TSR).

Cada año casi 6.000 nuevos pacientes ven progresar su ERC hasta la necesidad de recibir uno de los tres tipos de TSR en nuestro país [2].

Dados los costes sanitarios, sociales y personales del TSR, lo convierten en un problema sanitario de especial relevancia. El TSR tiene tres modalidades: Trasplante renal (TX), Hemodiálisis (HD) y Diálisis Peritoneal (DP), pudiendo ser tratado inicialmente con una de ellas y sucesivamente con las otras, en beneficio del paciente [1].

- Trasplante Renal (TX), consiste en el trasplante de un riñón sano en un paciente con una enfermedad renal avanzada. Cada año se realizan alrededor de 2500 trasplantes renales en nuestro país, y hay algo más de 4000 pacientes en lista de espera. Por motivos clínicos, sólo un 48% de los pacientes de ERC pueden acceder a un riñón [2].
- Hemodiálisis (HD), la sangre del paciente se extrae y depura empleando una máquina que la filtra y se la devuelve al paciente, para lo cual tiene que desplazarse a un hospital o centro especializado tres veces por semana como mínimo. La mayoría de los pacientes, en torno al 90%, en diálisis usan esta modalidad, en parte, debido a que la situación estructural y organizativa actual favorece la realización de esta técnica [2].
- Diálisis Peritoneal (DP), esta modalidad de tratamiento es domiciliaria. La eliminación de sustancias tóxicas se realiza a través de la membrana peritoneal del paciente, y se puede realizar de forma manual o de manera automatizada, durante la noche, mientras se duerme. El paciente solo acude al hospital

para revisión cada uno o dos meses. Pese a estas ventajas, un escaso porcentaje, en torno al 10%, de pacientes usan esta modalidad [2], debido a que no está suficientemente desarrollada en nuestro país (aspectos organizativos, respaldo de especialistas y capacitación de los facultativos) y a que no se realiza una adecuada información al paciente sobre su enfermedad y las opciones de tratamiento.

2.1.1. Diálisis Peritoneal, beneficios y problemática

Este tratamiento tiene especial relevancia en nuestro trabajo, pues el sistema e-Nefro ha sido diseñado para cubrir las necesidades de los pacientes en prediálisis y diálisis peritoneal.

Como hemos dicho anteriormente, la DP, a diferencia de la HD, es una modalidad de tratamiento domiciliaria, esto es, el paciente realiza su tratamiento, o la mayor parte de éste, desde su propio hogar.

Al margen de los aspectos médicos, la DP presenta una serie de ventajas y problemática derivadas de ser una técnica domiciliaria.

Podemos hablar de dos grandes beneficios, tanto para paciente como para el sistema sanitario:

- No altera en demasía la vida laboral y familiar de los pacientes, lo que repercute en una mayor independencia y en una mejor calidad de vida [3].
- Reduce costes y libera infraestructura y personal hospitalario [1] [2].

Por estas ventajas, podemos considerar la DP una técnica muy recomendable para pacientes que quieran mantenerse laboralmente activos, así como niños y ancianos que tengan buen soporte familiar.

No obstante, este tipo de técnicas llevan implícita una serie de inconvenientes que hay que tener muy en cuenta para proporcionar un buen servicio al paciente [3]:

- La menor supervisión médica con respecto a los pacientes tratados con HD.
- Inseguridad del paciente, provocada por su responsabilidad e implicación en el tratamiento.

Con el sistema e-Nefro, explicado a continuación, se intenta poner solución a esta problemática, acortando la brecha existente entre paciente y profesional sanitario.

2.2. El sistema e-Nefro

En Nefrología, la aplicación de las TIC puede dotar de capacidades como el control remoto de pacientes en diálisis, la monitorización de las sesiones de diálisis para la detección temprana de problemas, o la transmisión de datos y mensajes desde el hogar.

El proyecto e-Nefro es un esfuerzo multicéntrico entre la Universidad de Sevilla y diversos hospitales nacionales [4]. En dicho proyecto se investiga sobre la mejor forma de optimizar los cuidados en el hogar de los pacientes en tratamiento conservador, prediálisis y diálisis peritoneal, con la intención de potenciar los autocuidados, la seguridad del paciente en su domicilio, y el seguimiento supervisado y continuo de dichos pacientes.

Para ello, se ha diseñado una arquitectura modular adaptable para la teleasistencia integral de pacientes nefrológicos (extensible a cualquier otra especialidad y escenario clínico), donde se pone en práctica una metodología de aplicación TIC en sanidad que prima la implicación en su diseño y desarrollo a los usuarios (profesionales y pacientes), con el fin de que se adaptes a sus necesidades reales.

Con e-Nefro, se busca que los pacientes se sientan más seguros con su implicación directa en el tratamiento. Para ello, el sistema les da la capacidad de:

- Monitorizar y transmitir automáticamente sus condiciones sanitarias.
- Estar en contacto con los médicos, para notificar situaciones de emergencia.
- Recibir información sobre el tratamiento de acuerdo a sus preferencias.

Por su parte, los profesionales sanitarios también cuentan con una serie de funcionalidades, que ayudan a dar una mejor atención a los pacientes:

- El acceso a información útil, con una visualización de ésta ajustada a las necesidades y preferencias de cada usuario, para evaluar la evolución de los pacientes.
- La capacidad de acceder a registros de los pacientes completos y fiables.
- El acceso a los datos del usuario asistido actualizados en tiempo real en cualquier momento.
- La recepción de eventos críticos, sin ser inundados con intervenciones/alarmas no críticas.

Todos estos requisitos tienen un punto común, se basan en la generación y recepción de información sanitaria.

El sistema e-Nefro, se apoya en una base de datos relacional implementada con tecnología Postgresql, para almacenar la información de manera persistente, y acceder a ella de manera estructurada cuando sea necesario. En ella se almacena la información referente a las personas implicadas en el tratamiento, pacientes, profesionales y cuidadores; así como de las organizaciones que intervienen en él; el plan de cuidados y las actividades que debe realizar el paciente en su domicilio; los registros e incidencias generados por éste en el seguimiento de su tratamiento y los mensajes entre paciente y profesionales.

Toda esta información almacenada, base del sistema e-Nefro, es lo que se quiere hacer accesible con el servicio web diseñado al resto de sistemas de salud. En el Capítulo 3 se analizará el modelo de datos de e-Nefro, para la identificación de los objetos de información de interés para otros sistemas.

2.3. Interoperabilidad en Salud

La interoperabilidad es la habilidad de un conjunto de sistemas para trabajar juntos intercambiando información, conocimiento y servicios.

La necesidad de sistemas interoperables en salud es evidente, pues se requiere un constante intercambio de información entre los profesionales de la salud, entre el personal sanitario y los pacientes, entre instituciones prestadoras de servicios, entre sistemas auxiliares con sistemas de registros clínicos electrónicos, etc. [5].

Dichos sistemas pueden haber sido desarrollados por organizaciones diferentes e implementados con distintas tecnologías. De cara a la interoperabilidad, los sistemas son cajas negras en los que no importa cómo trabajen internamente, lo importante es que la información que desean comunicar pueda ser entendida y utilizada por el resto.

En consecuencia, la comunicación entre dichos sistemas requiere que se comparta un marco de referencia común que permita la interacción y coordinación entre ellos. Uno de los mecanismos para establecer ese marco común es a través de estándares de interoperabilidad, los cuales ofrecen uniformidad en la denominación de los componentes del sistema de salud ya sean personas, diagnósticos o procedimientos, y la forma de ser transmitidos, para garantizar que los datos estén disponibles y tengan significado clínico y administrativo.

Hay una gran cantidad de estándares para la comunicación en salud. En función de qué tipo de interoperabilidad se busque será interesante usar uno u otro. En lo que respecta al presente trabajo, es interesante mencionar tres estándares de HL7 que anteceden a FHIR y de los cuáles ha heredado sus características más interesantes, estos son, HL7 V2 [6], HL7 V3 [7] y CDA [8].

2.3.1. El estándar HL7 FHIR

El estándar FHIR [9], *Fast Healthcare Interoperability Resources*, es la última especificación de estándares HL7 para el intercambio electrónico de información en salud, cuyo surgimiento obedece a las dificultades de implementación de HL7 V3, a la necesidad de transición de HL7 V2 y los retos que suponen las nuevas tendencias como salud móvil (mobile Health o m-Health), intercambio de registros clínicos personales (Personal Health Record, PHR), historia clínica electrónica (Electronic Health Record, EHR), las comunicaciones en la nube o el Internet de las cosas (Internet of Things, IoT). Está en fase DSTU (*Draft Standard for Trial Use*), siendo la DSTU 2 la versión oficial actual.

FHIR ha sido diseñado desde el punto de vista de las necesidades de las implantaciones y con la idea de que sean tan sencillas como sea posible. Por su naturaleza y diseño, es fácil de aprender, de desarrollar y de implementar, consiguiendo reducir de forma notable los tiempos y, por consiguiente, el coste final de los proyectos de integración. Además, se ha diseñado para que pueda convivir e integrarse correctamente con sistemas actuales empleando versiones antiguas de las especificaciones HL7, lo que a priori supone una ventaja para su incorporación a sistemas ya operativos.

El estándar está diseñado específicamente para la web. La información se intercambia en base a estructuras XML o JSON que utilizan un protocolo REST basado en HTTP, en contraposición a los servicios basados en SOAP que se pueden encontrar en la mayoría de sistemas de este tipo. La API RESTful define una serie de operaciones a realizar sobre los recursos, donde cobrará especial relevancia la interacción *Search*, utilizada para la búsqueda de recursos siguiendo algún criterio [10].

Hay que tener en cuenta que FHIR no es un protocolo de seguridad, ni define ninguna función relacionada con ella. Sin embargo, FHIR sí define los protocolos de intercambio y modelos de contenido que necesitan ser utilizado para un intercambio de información seguro. Sin entrar en demasiados detalles, el estándar propone una serie de medidas de seguridad [11]:

- Comunicaciones cifradas mediante TLS/SSL, por ejemplo, utilizando el protocolo HTTPS.
- Sistemas de identificación y autenticación de usuarios.
- Sistemas de control de acceso a las operaciones del servidor, mediante etiquetas de seguridad.
- Sistemas de generación de *logs*, con los que conseguir un registro de las acciones llevadas a cabo en el servidor, y detectar intrusos.
- Uso de firmas digitales.
- Cuidado en el uso de archivos adjuntos y elementos narrativos.

Para comprender la especificación FHIR, es necesario familiarizarse con términos como Recursos, Elementos Narrativos y Extensiones; además de tener conocimientos previos en los conceptos de Mensajes" y Documentos" electrónicos ya empleados en las especificaciones HL7 V2 y V3, y CDA.

2.3.1.1. Recursos FHIR: tipos de datos, extensiones y elementos narrativos

FHIR se basa en lo que denominan Recursos [12]. Un recurso es la unidad básica de interoperabilidad, la unidad más pequeña que tiene sentido intercambiar. Los recursos son representaciones de conceptos del mundo sanitario, y administrativo, como, por ejemplo, paciente, profesional sanitario, procedimiento o cita.

Los recursos tienen una serie de elementos comunes (Figura 2-1) [13]:

- Un identificador a través de la cual puede ser registrado, localizado y recuperado.
- El contenido, formado por una serie de campos de información, definidos en la estructura del propio recurso.
- Un mecanismo de extensión que permite a los implementadores añadir nuevas propiedades de manera sencilla.
- Un componente denominado elemento narrativo, que permite una visión legible de los datos almacenados en el recurso.

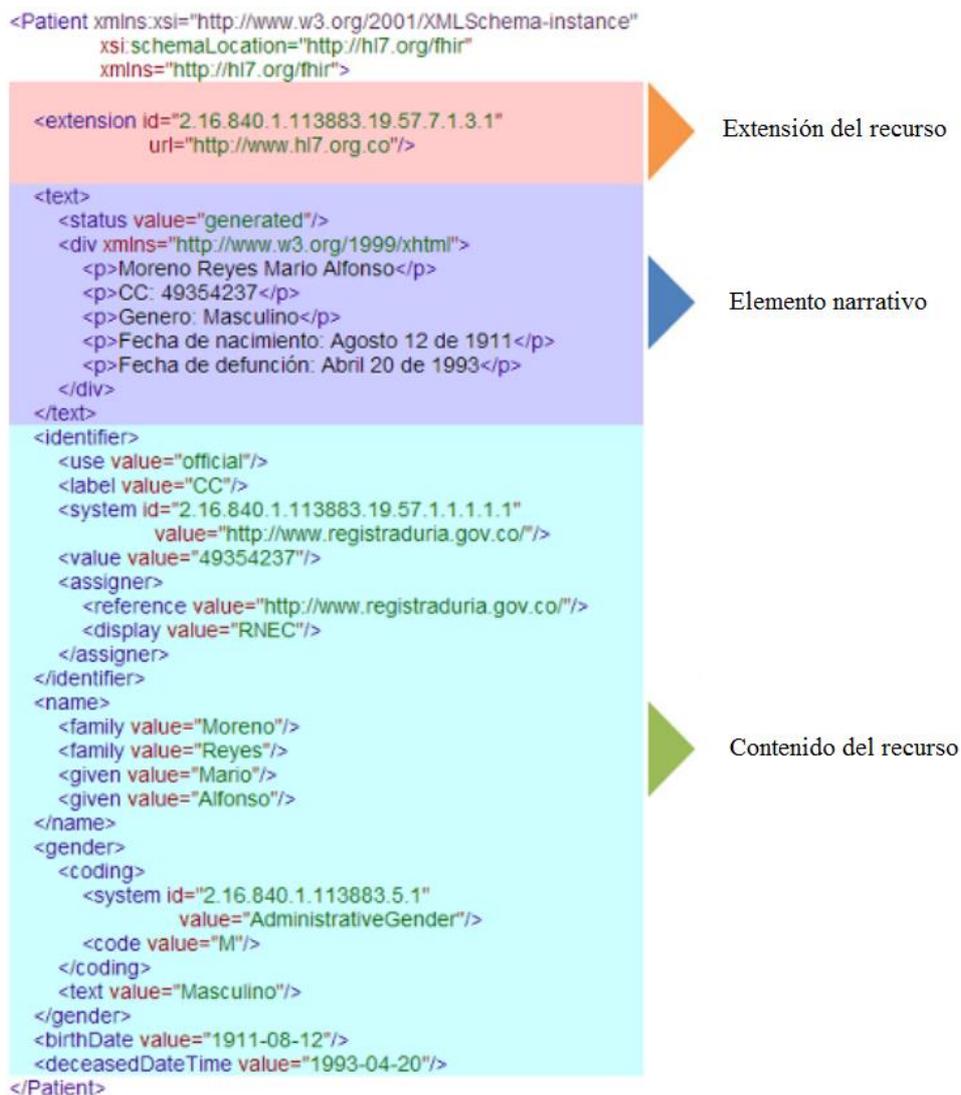


Figura 2-1. Estructura del Recurso FHIR

En primer lugar, todos los recursos necesitan un identificador, que es asignado por el servidor. Este identificador es único dentro del espacio de todos los recursos del mismo tipo en el servidor. Una vez asignado, el identificador no cambia nunca.

La información va almacenada en los campos del propio recurso. Estos campos dependen del tipo de recurso que estemos manejando, por ejemplo, en el recurso del tipo Paciente, hablaríamos de campos como *Nombre* o *Fecha de nacimiento*, mientras que en un recurso Cita, tendríamos campos como *Localización* o *Fecha de inicio*.

Estos campos están definidos en la estructura del propio recurso, y tienen asignado un tipo de datos concreto [14]. FHIR define una serie de tipos de datos, simples o primitivos, como String o Boolean, y Complejos, como CodeableConcept o Period, formados a su vez por tipos de datos más simples.

En la Figura 2-2 podemos ver todos los tipos de datos definidos en FHIR.

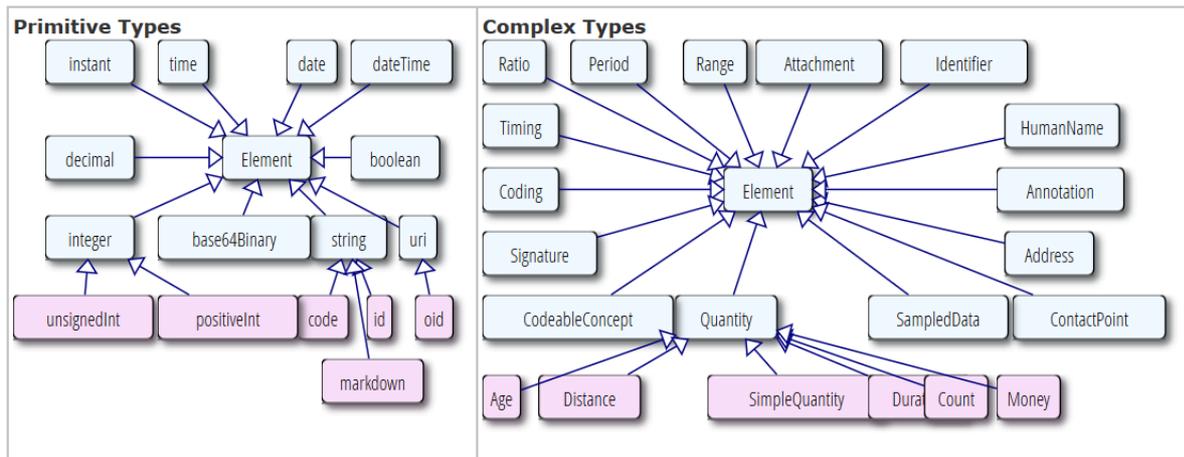


Figura 2-2. Tipos de datos en FHIR

Algunos atributos, que siguen la nomenclatura atributo[x], pueden ser implementados con diferentes tipos datos, en función de lo que necesitemos de ellos (Figura 2-3).



Figura 2-3. Tipo de dato atributo[x]

Un tipo de dato especial es el llamado *Reference* (Figura 2-4) [15], que nos proporciona un mecanismo muy útil a la hora de estructurar la información. Con él, es posible contener recursos dentro de otros. Esto nos permite combinar recursos y poder plasmar la información sanitaria en toda su complejidad. Por ejemplo, podríamos tener un recurso de tipo *Practitioner*, y en su campo *ManagingOrganization* referenciar un recurso de tipo *Organization*, con toda la información relativa a la organización a la que pertenece el profesional.

Existen dos tipos de referencias:

- Internas, referencias a recursos empaquetados dentro del recurso principal.
- Externas, referencias a recursos almacenados externamente a través de su URI.

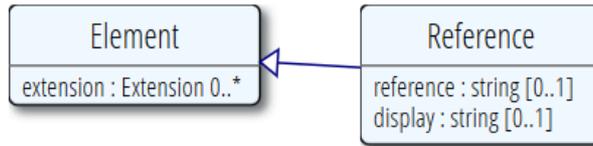


Figura 2-4. Diagrama de referencia a recursos

El campo *reference* contiene la dirección del recurso contenido, pudiendo ser una URL o un identificador, mientras que el campo *display* sirve para identificar a lo que se está haciendo referencia, sin entrar en detalle.

En nuestro servicio utilizaremos el primer tipo de referencia. En la Figura 2-5 se muestra un ejemplo.

```

<Composition xmlns="http://hl7.org/fhir">
  <extension>...</extension>
  <text>...</text>
  <contained>
    <Organization>
      <id value="org1"/>
      <!-- whatever information is available -->
    </Organization>
  </contained>
  <information>
    <!-- other attributes -->
    <custodian>
      <reference value="#org1" />
    </custodian>
    <!-- other attributes -->
  </information>
</Composition>

```

Figura 2-5. Ejemplo de referencia a recursos

Tal como especifica el estándar, existen algunas limitaciones en los recursos contenidos: no deben tener extensiones, recursos contenidos ni elemento narrativo.

Como hemos dicho anteriormente, los recursos son representaciones de conceptos básicos del mundo sanitario y administrativo. A veces, es necesario utilizar FHIR en contextos locales o sistemas tan específicos, que estos recursos definidos se nos quedan cortos a la hora de representar nuestra información. Ante este problema, FHIR nos da la solución en forma de Extensiones [16]. Una extensión no es más que añadir nuevos campos de información a un recurso ya definido.

Todas las extensiones requieren la publicación de una definición formal, mediante un recurso tipo *StructureDefinition* (Figura 2-6). Siempre que se intercambian los recursos que contienen las extensiones, las definiciones de las extensiones estarán a disposición de todas las partes que comparten los recursos.

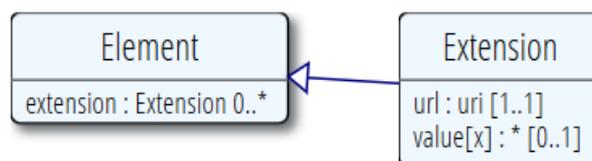


Figura 2-6. Diagrama de extensión de recursos

Cada extensión contiene un URI, en el campo *url*, que hace referencia a la fuente de la definición, mientras que el campo *value* contiene la información como tal. El campo *value* puede ser cualquiera de los tipos de datos vistos anteriormente, incluido el tipo Reference.

En la Figura 2-7 se puede ver un ejemplo de recurso Patient extendido.

```

<Patient>
  <extension url="http://hl7.org/fhir/StructureDefinition/patient-clinicalTrial" >
    <extension url="NCT" >
      <valueString value="123456789" />
    </extension>
    <extension url="period" >
      <valuePeriod>
        <start value="2009-03-14" />
      </valuePeriod>
    </extension>
    <extension url="reason" >
      <valueCodeableConcept>
        <coding>
          <system value="http://acme.org/codes/general" />
          <code value="tt14j" />
        </coding>
      </valueCodeableConcept>
    </extension>
  </extension>
  <!-- other data for patient -->
</Patient>

```

Figura 2-7. Ejemplo de extensión de recursos

Por último, hay que hablar de los elementos narrativos [17], cuya finalidad es facilitar la legibilidad humana del contenido. Para ello se utiliza un fragmento XHTML, generando el resumen de los datos, en forma no estructurada (Figura 2-8).

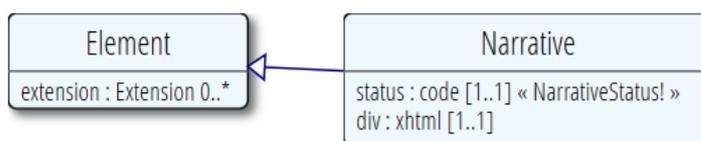


Figura 2-8. Diagrama de elemento narrativo

El contenido del elemento *div* es el fragmento de XHTML que únicamente contendrá elementos básicos de HTML. Esto es para asegurar que el contenido del elemento narrativo está íntegramente dentro del recurso y que no hay contenido activo. Dicho contenido introduciría problemas de seguridad y por tanto se prohíbe su uso.

Podemos ver un ejemplo de elemento narrativo en la Figura 2-9.

```
<narrative>
  <div xmlns="http://www.w3.org/1999/xhtml">This is a simple
    example with only plain text</div>
</narrative>

<narrative>
<div xmlns="http://www.w3.org/1999/xhtml">
  <p>
    This is an <i>example</i> with some <b>xhtml</b> formatting.
  </p>
</div>
</narrative>
```

Figura 2-9. Ejemplo de elemento narrativo

En la Figura 2-10, se puede ver el resultado práctico del elemento narrativo de un recurso que contiene la información de un profesional médico.

Result Narrative	Alberto NÚÑEZ GARCÍA	
	Identifier	57722647F
	Teléfono	698454545
	Email	alberto@us.es
	Cargo	MEDICO
	Especialidad	Especialista Nefrología
	Organización	Hospital de prueba

Figura 2-10. Resultado de elemento narrativo

En el apartado correspondiente a la implementación del servicio web, se verá cómo se generan estos elementos, y de qué forma pueden los clientes utilizarlos.

2.3.1.2. Mensajes y Documentos

Aunque, debido al tipo de servicio web que se va a diseñar, no se han utilizado Mensajes y Documentos, es conveniente explicarlos brevemente para ver el potencial de FHIR.

Los recursos pueden agruparse en forma de mensajes, al igual que se hacía en las versiones 2 y 3 de HL7. En la mensajería FHIR, cuando ocurre un evento que desencadena la necesidad de un intercambio de información, se genera y envía un mensaje desde una aplicación de origen a una aplicación de destino, de forma similar a los mensajes y eventos disparadores en HL7 V3.

Así mismo, también se pueden utilizar para construir documentos clínicos. Los documentos FHIR, al igual que en CDA, tienen una serie de características (persistencia, custodia, potencial de autenticación, contexto, integridad y legibilidad humana), que los convierten en una representación de la realidad de un evento clínico.

2.4. Material y método

Para concluir el estado del arte, se hablará de las principales herramientas que se han utilizado para las tareas de diseño e implementación del servicio web.

- **MagicDraw** [18]. Esta herramienta CASE (*Computer Aided Software Engineering*), compatible con el estándar UML, ha sido utilizada en las tareas de modelado necesarias para el diseño del servicio web, esto es, modelado de los objetos de información y las relaciones existentes entre ellos, y de la funcionalidad y estructura del servicio web.
- **Eclipse Mars** [19]. Para el desarrollo del servicio web, se ha utilizado esta plataforma de desarrollo en la plataforma Java, junto a los plugins de **Spring** [20] e **Hibernate** [21], muy utilizados en el desarrollo de aplicaciones de este tipo.
- **Apache Maven** [22]. Maven es una herramienta CASE de gestión de proyectos de software y comprensión, basada en el concepto de un *Project Object Model* (POM). Se ha utilizado Maven para describir el proyecto software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Su principal característica es que está listo para usar en red. El motor incluido en su núcleo puede dinámicamente descargar plugins de un repositorio que provee acceso a muchas versiones de diferentes proyectos Open Source en Java, de Apache y otras organizaciones y desarrolladores.
- **Apache Tomcat 8.0** [23]. Se utiliza este contenedor web con soporte de servlets y JSPs, para el despliegue de nuestro servicio web, a través de Eclipse.
- **PostgreSQL** [24]. Sistema de gestión de bases de datos relacional orientado a objetos y de código libre. La base de datos de e-Nefro está implementada bajo este sistema, por lo que se ha utilizado tanto en la fase de diseño, para realizar consultas y estudiar la base de datos, como en la fase de implementación, para extraer y manejar la información, incorporando la librería correspondiente en Eclipse.
- **Android Studio** [25]. Herramienta principal para el desarrollo de aplicaciones Android. Este IDE desarrollado por la misma compañía propietaria del sistema operativo Android, lo que aporta numerosas ventajas adicionales a la hora de escribir aplicaciones para su plataforma. En nuestro caso se ha utilizado para la implementación de un cliente Android FHIR que utilice nuestro servicio web.
- **HAPI FHIR** [26]. Para la implementación de la especificación HL7 FHIR en nuestro servicio web, se utiliza esta API de código abierto, completamente compatible con aplicaciones Java. Entre los distintos usos que se proponen para la librería, es especialmente interesante para nuestro proyecto el de emplearla en una aplicación que permita a aplicaciones externas acceder a nuestra información (Figura 2-11).

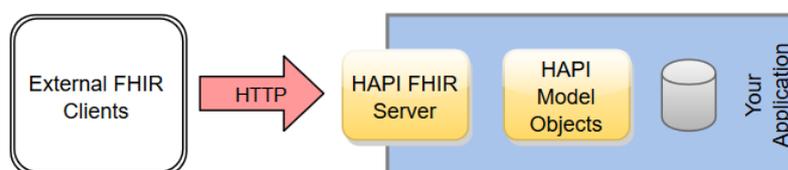


Figura 2-11. Funcionalidad HAPI FHIR

3 RESULTADOS

“Los científicos estudian el mundo tal como es; los ingenieros crean el mundo que nunca ha sido.”

Theodore von Kármán

En este tercer capítulo, se muestran los resultados obtenidos en la realización del trabajo. En primer lugar, se ha realizado una revisión literaria sobre la situación actual de FHIR, centrada en determinar las principales áreas y sistemas donde se está utilizando dicha especificación.

Tras ello, se han realizado las tareas que abarcan el diseño del servicio web: identificación y modelado de objetos de información de e-Nefro, identificación y modelado de los recursos FHIR a intercambiar y diseño de los métodos de acceso a la información.

Por último, se ha llevado a cabo la implementación del servicio web, utilizando para ello una API de código abierto de FHIR en Java: HAPI FHIR.

3.1. Revisión literaria sobre la especificación HL7 FHIR

Previo al diseño del servicio, se ha realizado un estudio de la situación actual del estándar a través de una profunda revisión literaria. Con este análisis, se busca conocer en qué áreas médicas se está aplicando el estándar, así como cuáles son los principales sistemas que están planteando su implementación, para poder enmarcar nuestro servicio en el panorama actual.

A continuación, se expone la metodología de búsqueda utilizada para recopilar la información, los resultados obtenidos del estudio y su discusión, y las principales conclusiones que podemos extraer de ello.

3.1.1. Metodología de búsqueda

Para la elaboración de esta revisión bibliográfica, se han realizado búsquedas de artículos relevantes, atendiendo al número de citas en otros artículos, en lengua inglesa utilizando las bases de datos de Scopus, ScienceDirect y Google Scholar, desde el año 2013, fecha de publicación de FHIR, hasta abril de 2016 (documentos en prensa incluidos). El conjunto de artículos se obtuvo recogiendo aquellos que incluían "FHIR" o "Fast Health Interoperability Resource" en el título, resumen o como palabras clave, dando ambas búsquedas prácticamente el mismo resultado. Se excluyeron de dicha lista artículos de opinión y documentos no catalogados como artículos.

La Figura 3-1 muestra el proceso de selección para el conjunto de artículos, en la que, tras agrupar todos los artículos de las diferentes bases de datos, y excluir duplicados, se hizo una revisión del resumen y, posteriormente, del contenido, para excluir aquellos artículos en los que FHIR no era el objeto del trabajo, o no se presentaba ninguna implementación que fuera a utilizarlo.

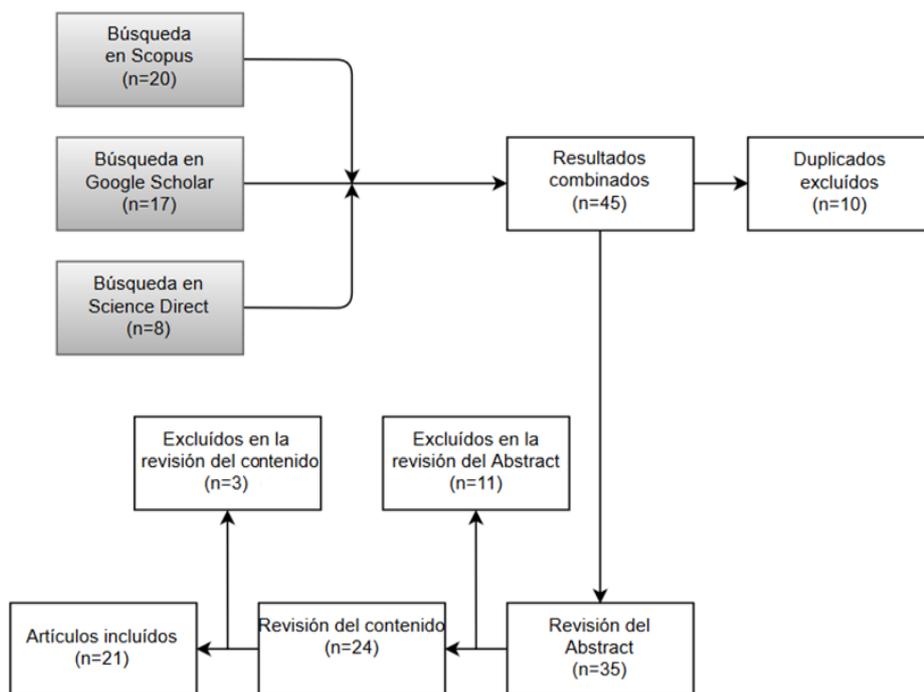


Figura 3-1. Diagrama de flujo de la selección de artículos

3.1.2. Resultados

La búsqueda en las bases de datos mencionadas dio un total de 45 artículos donde se mencionaba la especificación FHIR. Tras combinar los resultados y eliminar los duplicados, nos quedamos con 35 artículos, de los cuáles tras hacer una revisión del resumen y contenido, sólo 21 de ellos cumplían los requisitos del proceso de selección.

Como se puede ver en la Figura 3-2, a partir del conjunto de artículos seleccionado, se extrajeron datos como su fecha de publicación, interesante para estudiar la tendencia que seguirán, y la importancia de FHIR, en el futuro.

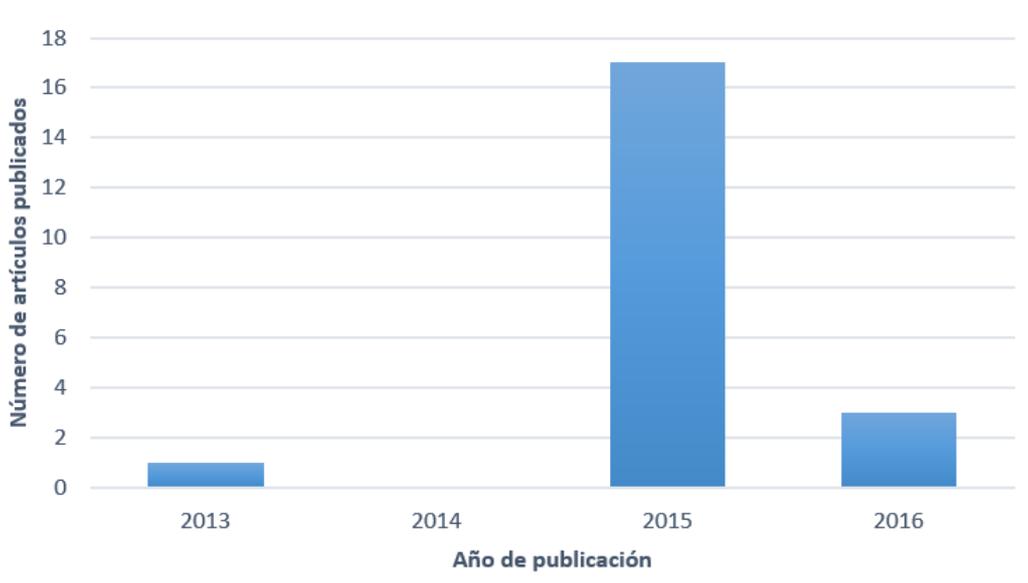


Figura 3-2. Distribución temporal de los artículos sobre FHIR

Podemos observar que el número de artículos publicados sobre FHIR va en aumento. Tras la publicación del artículo de presentación en 2013, en 2014 no se publicaron artículos donde la jugase un papel relevante. Ya en 2015, con la especificación más asentada, nos encontramos con un total de 17 artículos publicados sobre dicha especificación, y sólo en los cuatro primeros meses de 2016 se publicaron otros tres artículos.

Los artículos seleccionados también han sido clasificados en función de en qué área de la medicina se va a aplicar el sistema de e-Salud para el que se propone el uso de FHIR (Figura 3-3). Hay que mencionar que 5 de los 21 artículos seleccionados no tratan sobre un sistema donde aplicarlo, sino sobre aspectos teórico-técnicos de la especificación como tal, por lo que no han sido incluidos en la siguiente clasificación, que se realiza sobre los 16 artículos restantes.

El 41% (6) de los artículos tratan sobre la utilización de FHIR en sistemas de información sanitaria general, con el propósito de que se integre en el conjunto de sistemas e intercambie información con éstos. El 23% (4) proponen utilizar FHIR en sistemas para la monitorización de pacientes crónicos, donde se registrarán sus variables que serán accesibles por otros sistemas, aunque no especifican a qué tipo de pacientes concretos irán dirigidos. Profundizando algo más en este tipo de sistemas, el 6% de los artículos habla de aplicar FHIR en un sistema de lo que se conoce como *Healthy Ageing* [27], mientras que otro 6% (1) va dirigidos a pacientes de Diabetes mellitus gestacional [28]. Por último, un 25% (4) trata sobre sistemas de ayuda a la decisión clínica y de generación de conocimiento, mediante análisis de datos genómicos y la aplicación de reglas de decisión.

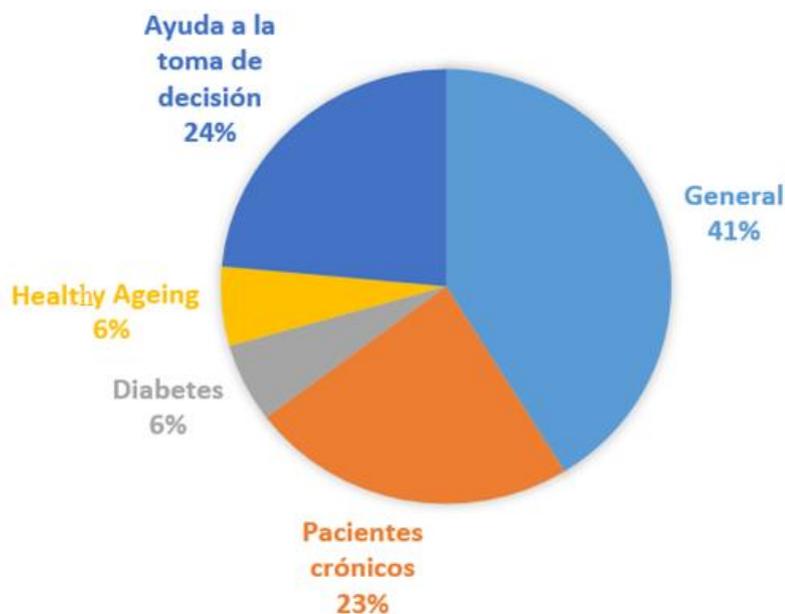


Figura 3-3. Clasificación en función del área médica donde se aplica FHIR

Otra clasificación interesante a la hora de analizar la tendencia de los sistemas que utilizan FHIR, es la de en qué tipo de sistemas se plantea su uso (Figura 3-4). En este aspecto, encontramos:

- 3 artículos son meramente informativos sobre FHIR. Entre ellos se encuentra el artículo de presentación [29], y 2 artículos que tratan sobre la colaboración médico-paciente [30] y la toma de decisiones clínicas [31], y el aporte de la especificación a estas tareas.
- 4 artículos realizan un estudio teórico de FHIR, los recursos que ofrece, tipos de datos [32-34] y la capacidad para utilizarlo con un sistema basado en arquetipos. [35]
- 4 artículos [36-39] plantean el uso de FHIR en el envío y captación de registros en sistemas de monitorización para pacientes crónicos y hogares asistidos, entre ellos el sistema CyPhyS+ [39], para para personas de edad avanzada, rama conocida como *healthy ageing* [27].

- 4 artículos [40-43] que aplican FHIR en sistemas de ayuda a la decisión clínica y de generación de conocimiento, mediante análisis de datos genómicos y la aplicación reglas de decisión.
- 3 artículos [44-46] sobre el uso de FHIR en aplicaciones móviles, lo que se conoce como m-Salud [47], una de las principales tendencias de la e-Salud.
- 2 artículos [48-49] plantean el uso de FHIR en sistemas EHR, para integrarlos con el resto de sistemas. Concretamente, se plantea su uso en OpenMRS [50].
- Un artículo [51] sobre el uso de FHIR en un sistema de validación de documentos clínicos.

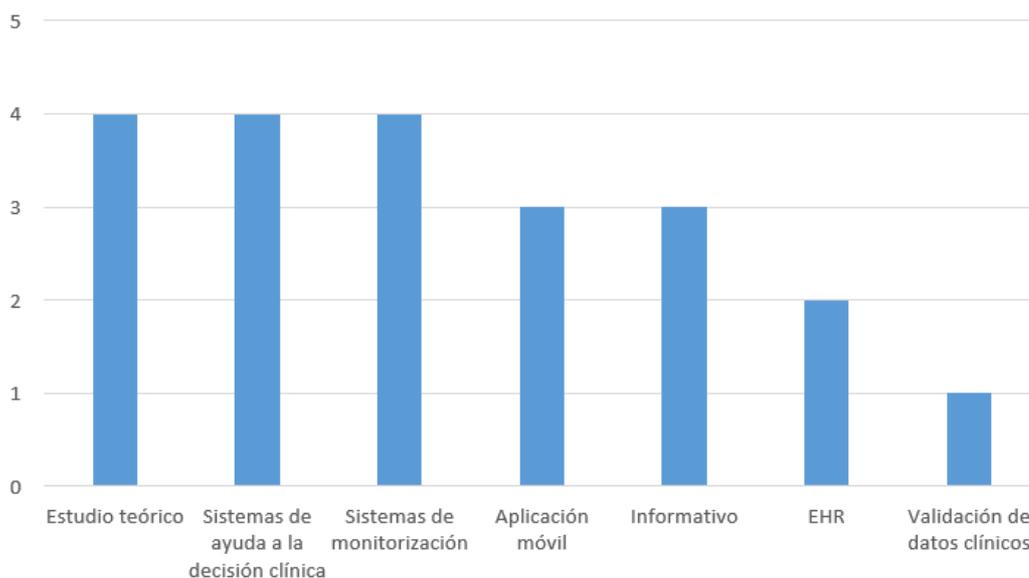


Figura 3-4. Clasificación según el tipo de sistema donde se aplica FHIR

3.1.3. Discusión

En la introducción se estableció el objetivo de la revisión: estudiar la situación actual de la especificación FHIR, y valorar su uso frente a otras propuestas. Se seleccionaron un total de 21 artículos donde FHIR jugaba un papel relevante.

Viendo la evolución de los artículos desde el año 2013 hasta la actualidad (Figura 3-2) podemos ver una tendencia claramente creciente en el número de artículos publicados. En 2013, año en el que se lanzó la especificación, tenemos el artículo [29] en el que fue presentado por HL7 y en el que se describen sus principales características. Durante 2014 no hubo artículos relevantes, pues FHIR estaba recién lanzado, y en una fase muy baja de desarrollo, por lo que se necesitaba tiempo para estudiarlo en profundidad y ver cómo evolucionaba. Ya durante 2015 y los cuatro primeros meses de 2016, con la especificación más asentada, encontramos 17 y 4 artículos, respectivamente, donde se trata su uso en diferentes sistemas de actual relevancia. Esta evolución nos hace pensar que, en los próximos años, con FHIR más maduro y estudiado, el número de sistemas que lo utilizarán aumentará considerablemente.

Analizando la Figura 3-3, donde se han clasificado los sistemas que se integrarán con FHIR, en función el área de la medicina a la que pertenecen, podemos observar que su uso no se limita a un área concreta, sino que vemos implementaciones en tres grandes áreas: sistemas de información clínica general, donde se registran las historias clínicas de los pacientes, sistemas para pacientes crónicos y en sistemas para la toma de decisiones médicas. Esto es posible debido a que FHIR no nos limita su uso a un área concreta, sino que ofrece una serie de recursos básicos que son empleados en todas las áreas médicas, tales como Paciente o Procedimiento, a la vez que permite la extensión de éstos para adecuarlos a un contexto específico. He aquí una de las características que hacen tan interesante a FHIR, su flexibilidad.

A nivel técnico, FHIR se considera [44] una especificación emergente que aprovecha elementos positivos de sus antepasados HL7 v2 y v3, y CDA, abordando sus limitaciones. Estos estándares han jugado un papel importante en garantizar la interoperabilidad de la asistencia sanitaria, pero que debido a las nuevas tendencias en e-Salud, como las aplicaciones móviles, o los sistemas de EHR, se hacen demasiado complejos para una integración del sistema rápida y poco costosa.

Por un lado, FHIR se basa en el estilo arquitectónico REST, siendo [38] adecuado para dispositivos ligeros y con poca capacidad de procesamiento, como los terminales móviles. También [38] hay que tener en cuenta que son dispositivos portátiles, donde la batería es un factor fundamental a tener en cuenta, y utilizar una tecnología que la ahorre al máximo es una gran ventaja. Aquí vemos una de los principales campos donde FHIR puede ser especialmente interesante, los sistemas de m-Salud. Atendiendo a la clasificación por el tipo de sistema donde se plantea utilizar FHIR (Figura 3-4), podemos observar que las aplicaciones móviles juegan un papel importante en la tendencia del estándar, así como los sistemas de monitorización, que también cuentan con elementos de batería.

La interoperabilidad es esencial para hacer frente a las limitaciones causadas por la aplicación *ad hoc* de los sistemas de información e historia clínica. FHIR [48] es particularmente relevante en este tipo de sistemas, una de las bases de la atención médica moderna.

Otro de los puntos que hace la especificación tan interesante, es que su diseño se ha hecho pensando en crear un estándar de interoperabilidad lo más sencillo posible de implementar, reduciendo los costes de los proyectos de integración. Es por ello particularmente relevante para OpenMRS [50], un sistema de registros médicos de código abierto ampliamente utilizado en todas las economías emergentes. Debido a la naturaleza del proyecto, es especialmente interesante el principio de ahorro de costes bajo los que la especificación ha sido diseñada. FHIR tiene el potencial de permitir a OpenMRS alejarse de una API propia para la integración con otros sistemas y definir los planes a largo plazo para la transición hacia una API basada en FHIR que reduce en gran medida la curva de aprendizaje para los desarrolladores y reducir costes [48] [49].

Otra de las líneas de investigación abierta, es la combinación de FHIR con arquetipos. El uso de arquetipos aportaría una serie de ventajas como ser capaz de verificar si un recurso es válido dentro del modelo de referencia, aunque aún hay cuestiones a resolver cuando se trata de arquetipos, sobre todo debido a la separación entre la definición narrativa y la definición formal de los recursos [35].

Actualmente, se está trabajando en el estudio de FHIR, con el objetivo de hacer que funcione junto a otros estándares y mecanismos de interoperabilidad. Esto es relativamente sencillo de conseguir con las especificaciones previas de HL7, HL7 V2 y V3 y CDA [52], pues FHIR se ha diseñado con esa idea. Conseguir esta integración con otros estándares, como por ejemplo Arden Syntax [41], requiere un esfuerzo mayor.

En comparación con estas especificaciones HL7 previas, FHIR ha implementado sus características más interesantes como la generación de mensajes y documentos clínicos, además de reducir la complejidad. Es especialmente interesante, como ya se ha comentado, [38] en sistemas con batería, donde es preferible el uso de la arquitectura REST que implementa FHIR, frente a SOAP, implementada, por ejemplo, por HL7 V2, uno de los estándares actualmente más asentados. Así mismo [38], se ha demostrado que hay una disminución significativa en el tráfico de datos cuando se basa en una arquitectura REST en combinación con HL7 FHIR. Otra ventaja es que el recurso FHIR proporciona un nivel de auto descriptivo, en comparación con el enfoque HL7v2 y v3, siendo mucho más intuitivo de manejar, ahorrando tiempo de aprendizaje a los desarrolladores.

Desde este punto de vista, resulta interesante dar el paso a FHIR, un estándar más flexible, ligero, eficiente y fácil de implementar, siempre teniendo en cuenta los costes asociados a adaptar los sistemas actuales a este nuevo estándar.

3.1.4. Conclusiones

FHIR está llamado a ser una de las especificaciones de referencia para el intercambio de información en la próxima generación de la e-Salud. Cada vez cobra más peso en los artículos sobre sistemas de e-Salud actuales, en detrimento de otras especificaciones como HL7 V2 y V3, y CDA.

Debido a cómo se ha concebido su diseño, hace que sea fácil y poco costoso de implementar, y al utilizar tecnología REST, es especialmente interesante en aparatos portátiles y con poca capacidad de procesamiento,

como los terminales móviles. Hemos visto que su uso no está limitado a ningún área médica, así como tipo de tecnología, pudiendo ser utilizado tanto en sistemas de historia clínica, de monitorización de pacientes crónicos o de ayuda al a toma de decisión.

Como todo nuevo estándar que se quiera introducir, es necesario hacer un esfuerzo para adaptar los sistemas actuales a él, especialmente en la combinación con estándares actualmente en uso y en entornos de trabajo específicos.

La situación actual de FHIR está condicionada por su corta vida. En la actualidad, gran parte del esfuerzo va destinado a analizarlo en profundidad y mejorarlo, así como hacer que trabaje junto a otros estándares y mecanismos de interoperabilidad. Aquí es donde radica su principal limitación, técnicamente es una especificación que combina lo mejor de sus antepasados con la sencillez, pero debido a su corta vida, y que aún está continua evolución, faltan unos años hasta que los sistemas de e-Salud lo implementen como estándar de interoperabilidad de referencia. Todo apunta a que en los próximos años se llevará a cabo esta transición de los sistemas actuales hacia FHIR.

3.2. Identificación y modelado de los objetos de información de e-Nefro

El primer paso en el diseño del servicio web es identificar qué información se va a ofrecer al resto de sistemas. Esta tarea consta de dos etapas. En primer lugar, es necesario estudiar el sistema e-Nefro, sus funcionalidades e interacción con el usuario, y extraer los objetos de información que tienen interés exportar fuera del sistema.

En segundo lugar, ya con los objetos de información identificados, es necesario estudiar la base de datos del sistema, así como su estructura, y modelar estos objetos, identificando los campos que han de tener y su relación con las tablas de la base de datos del sistema.

3.2.1. Identificación de los objetos de información

Como hemos introducido, en primer lugar, es necesario hacer un estudio de la funcionalidad del sistema e-Nefro, ya explicado en el estado del arte. Para dicho estudio se ha utilizado su manual de usuario, analizando cómo interactúan pacientes y profesionales con la aplicación. Se van a exponer las principales conclusiones sacadas de dicho estudio, así como los objetos identificados.

En primer lugar, e-Nefro establece los tipos de usuario Administrador, Jefe de servicio, Responsable, Profesional médico y Paciente/Cuidador.

En nuestro caso, para identificar los objetos de información, nos vamos a centrar en los dos últimos, pues son los usuarios que tienen importancia médica en el sistema.

Los pacientes en e-Nefro son personas afectadas por una enfermedad renal crónica. Se establece para ellos un periodo de seguimiento y un plan de cuidados para el tratamiento de su enfermedad. Es posible que un paciente cuente con un cuidador, otra persona que interactuará con el Sistema E-Nefro en nombre del paciente.

En la figura se puede ver cómo sería el menú de la aplicación para un usuario de este tipo.

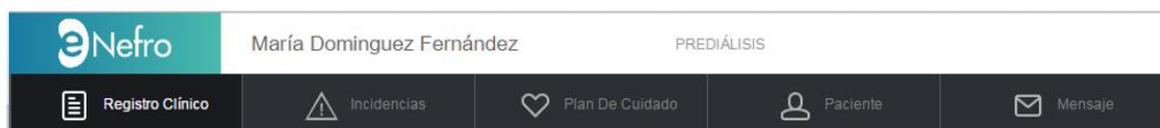


Figura 3-5. Funcionalidad del Paciente/Cuidador en e-Nefro

- **Registro Clínico:** desde esta opción, el paciente podrá consultar, editar o registrar la información clínica necesaria para que su médico pueda realizar el seguimiento de su tratamiento.
- **Incidencias:** pulsando en esta opción del menú, el usuario podrá registrar una incidencia y recibir una recomendación automática.

- **Plan de Cuidados:** el paciente podrá consultar el plan de cuidados que su profesional médico le ha diseñado. El plan de cuidados está formado por actividades, que pueden ser citas, tomas de medicamento, medida de variables fisiológicas o toma de imágenes.
- **Paciente/Cuidador:** aquí podrán ser consultados los datos personales del paciente/cuidador. También podrá solicitar la modificación de estos datos.
- **Mensaje:** el paciente podrá consultar los mensajes que le han sido enviados y contestar a los que reciba. Sin embargo, no podrá comenzar de motu proprio la comunicación con los miembros del equipo médico a través de esta vía.

Por su parte, los profesionales médicos, que forman parte de una organización sanitaria, también juegan un papel fundamental en el sistema. Se pretende que, través de e-Nefro, el profesional pueda conocer toda la información de sus pacientes y disponga de un apartado de análisis de datos que le permita estudiar las respuestas que dan a los distintos planes de cuidado.

En la figura se puede ver cómo sería el menú de la aplicación para un usuario de este tipo.



Figura 3-6. Funcionalidad del Profesional en e-Nefro

- **Pacientes:** muestra el listado de los pacientes que corresponde al profesional médico. A partir de aquí se podrán consultar todos los datos del mismo.
- **Citas:** se muestra el formulario para dar de alta una nueva cita y en la zona inferior, se muestran las citas ya programadas.
- **Incidencias:** en este apartado el profesional podrá consultar todas las incidencias que sus pacientes han dado de alta.
- **Mensajes:** muestra un gestor de mensajería. El profesional podrá escribir a cualquiera de sus pacientes. Además, podrá responder los mensajes que reciba de otros usuarios del sistema.
- **Análisis de datos:** en este apartado el profesional podrá realizar diversos análisis de los datos almacenados sobre sus pacientes.

Con todo ello, podemos identificar una serie de objetos de información en el sistema e-Nefro:

- Paciente
- Cuidador
- Profesional sanitario
- Organización
- Periodo de seguimiento
- Plan de cuidados
 - Cita
 - Toma de medicamento
 - Medida de variable fisiológica
 - Toma de Imágenes

- Registro Clínico
 - Cita
 - Toma de medicamento
 - Medida de variable fisiológica
 - Toma de Imágenes
- Incidencia
- Mensaje

Un punto a tener en cuenta, es que estos objetos no solo tienen que existir, sino tener sentido fuera del contexto de e-Nefro. Es por ello que se descarta el objeto 'Mensaje' para el modelado e inclusión en el servicio, pues la información que contiene solo tiene sentido dentro del sistema.

En el siguiente apartado, se desarrollarán estos objetos de información, identificando sus campos, y relacionándolos con la base de datos de la aplicación.

3.2.2. Modelado de los objetos de información

Una vez identificados los objetos de información que intercambiaremos con el resto de sistemas, hay que modelarlos, indicando los atributos que lo forman e identificándolos con campos de la base de datos de donde recogeremos la información.

Para ello, lo primero será hacer un análisis de la base de datos del Sistema y su estructura, para ver qué información contiene y cómo está organizada. Una vez entendido su funcionamiento, procederemos al modelado de los objetos.

En estos objetos es donde se guardará la información obtenida de las consultas a la base de datos para, posteriormente, modelarla en los recursos FHIR. Por ello, como métodos únicamente tendrán los llamados *Getters & Setters*, para obtener o modificar, respectivamente, estos atributos con la información.

3.2.2.1. Estudio y análisis del modelo de datos de e-Nefro

Para el análisis se ha empleado la documentación de la base de datos de e-Nefro, suministrada por el Grupo de Ingeniería Biomédica, así como consultas puntuales a una base de datos real, con el objetivo de determinar qué información estaba almacenada en ciertos campos donde la documentación no era del todo clara.

Como se ha dicho, e-Nefro, como sistema de registros clínicos, se sustenta en una base de datos relacional implementada con tecnología PostgreSQL. La base de datos cuenta con un total de 85 tablas y 416 columnas. Debido a esta extensión, no se va a entrar en demasiada profundidad en el estudio realizado, únicamente nos centraremos en las tablas cuya información van a aparecer en nuestros objetos.

La información de los usuarios de la aplicación se almacena en la tabla `e_usuario` (Figura 3-7). Es información propia del usuario e-Nefro en cuestión y no su información personal. Esta información se almacena en la tabla `e_person`, relacionada con la primera a través del atributo `usua_persona`. Las tablas `e_valor_dominio` y `e_tipoidentificacion`, se utilizan para almacenar códigos y sus descripciones, generales y de identificación respectivamente, que se relacionan con las tablas principales de la misma forma que antes.

Vemos que la tabla `e_usuario`, además de a `e_person`, hace muchas referencias a otras tablas. Por importancia, hay que mencionar la referencia a la tabla `e_organizacionelement` (Figura 3-8) que contiene la información referente a la organización sanitaria a la que pertenece el usuario.

Y a la tabla `e_staffmember` (Figura 3-9), donde se almacena qué cargo y especialidad ocupa el usuario, profesional en este caso, en la organización.

Close relationships within one two degrees of separation:

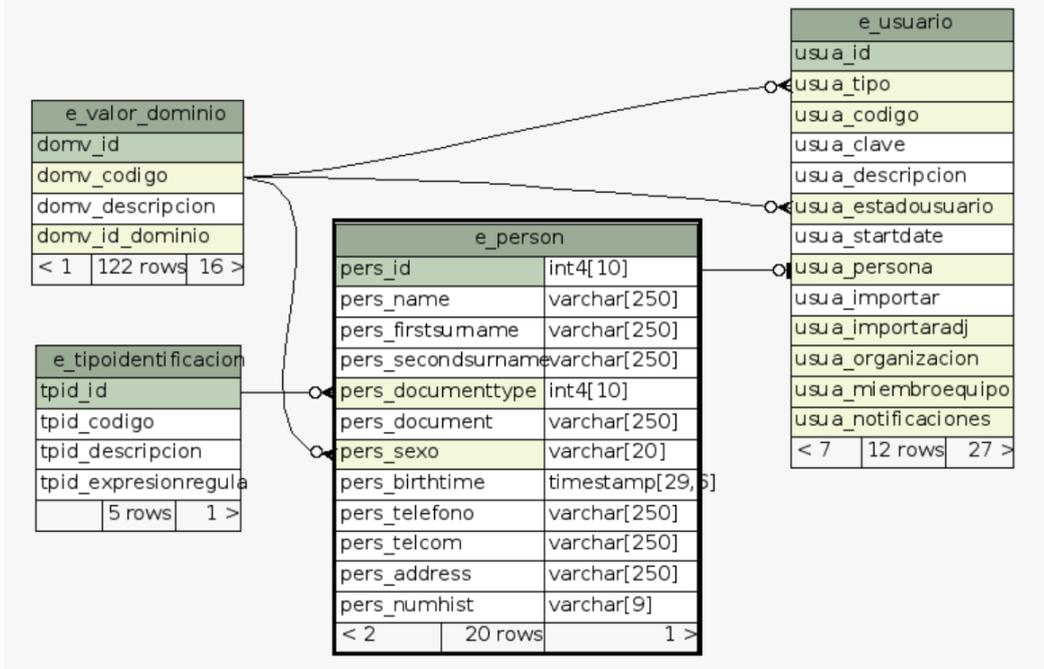


Figura 3-7. Modelo de datos de e-Nefro: Información personal de usuario

Close relationships within one two degrees of separation:

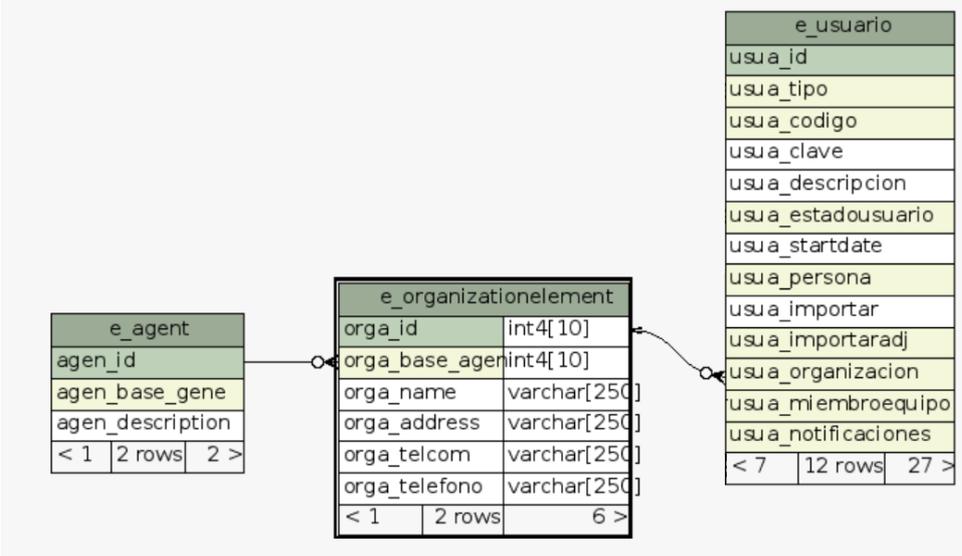


Figura 3-8. Modelo de datos de e-Nefro: Organización del profesional

Como se explicó con anterioridad, cada usuario dispone de un plan de cuidados, formado por actividades, y en su seguimiento se generan registros e incidencias. Vamos a ver la estructura que sigue la información referente a una actividad Cita, siendo ésta extensible al resto de actividades.

La información del plan seguido por un usuario se almacena en e_plan. Esta tabla se relaciona con la tabla e_usuario, explicada anteriormente, y con la tabla e_appointmentactivity, tabla principal de una actividad de tipo Cita, y donde se almacena la información referente a ella. Común a todas las tablas de actividades, existe la tabla e_activity, que almacena información común que poseen todas las actividades de un plan. Por último, los registros que produce la realización de la actividad Cita son almacenados en e_appointmentregistry, también relacionada con la tabla principal de la Cita. Se pueden ver estas relaciones en la Figura 3-10.

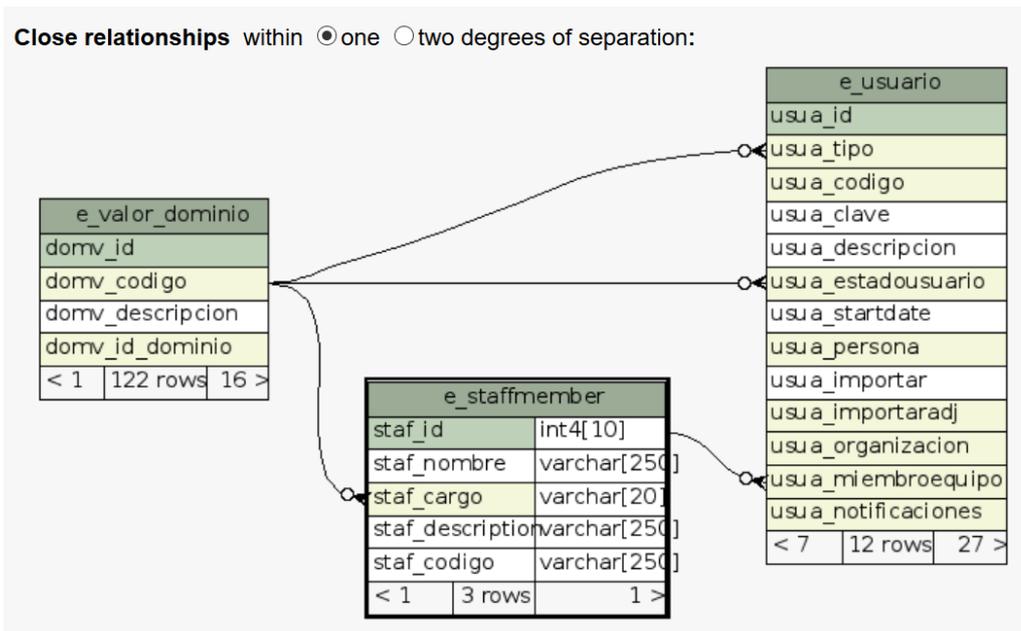


Figura 3-9. Modelo de datos de e-Nefro: Cargo y especialidad del profesional

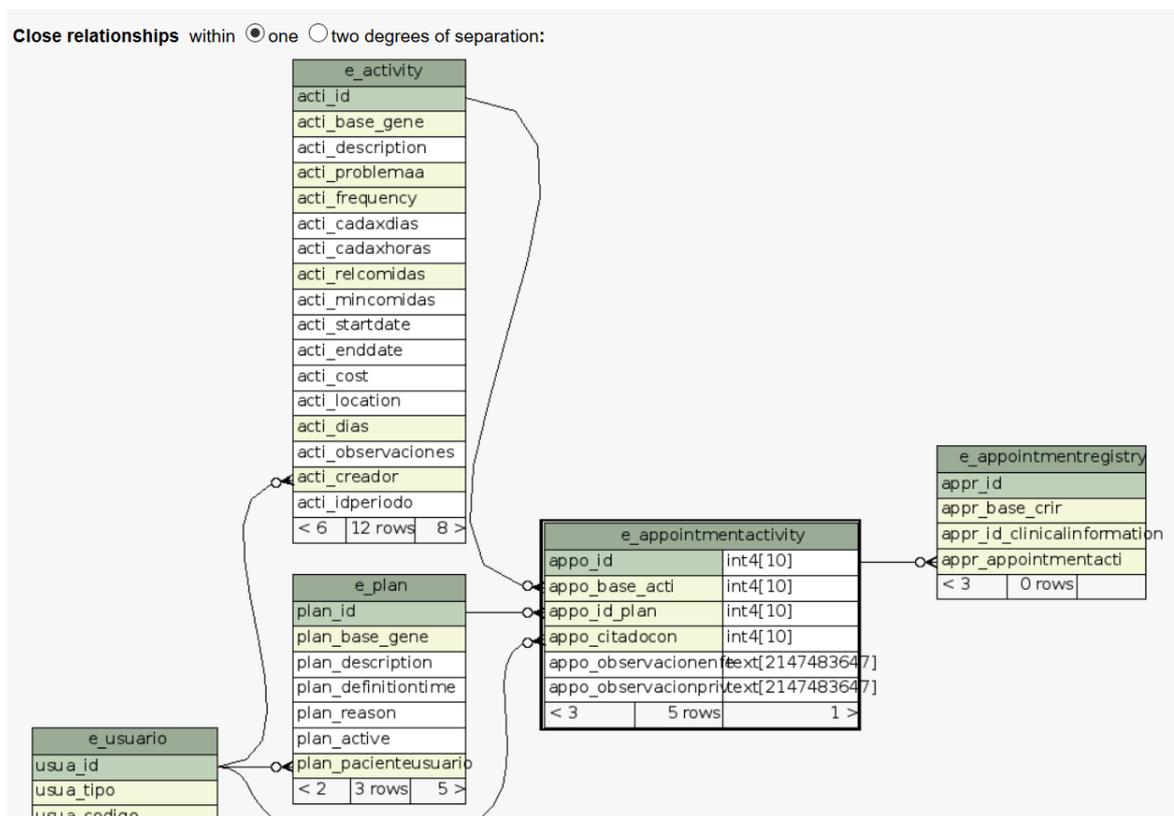


Figura 3-10. Modelo de datos de e-Nefro: Actividades

La tabla para los registros generados, además de con e_appointmentactivity, se relaciona con otras dos tablas, e_clinicalregistry, donde se almacena información común a todas las tablas de tipo registros, y e_clinicalinformation, donde se almacena información clínica como el paciente que ha generado el registro (Figura 3-11).

Por último, aunque ya se ha visto que las tablas están relacionadas entre sí, hay tablas específicas utilizadas para relacionar elementos de información y cuyo nombre empieza de la forma e_rel_X.

Es especialmente interesante la tabla e_rel_profmedi (Figura 3-12) pues relaciona a pacientes y profesionales, haciendo que podamos obtener todos los pacientes que tiene un profesional a su cargo.

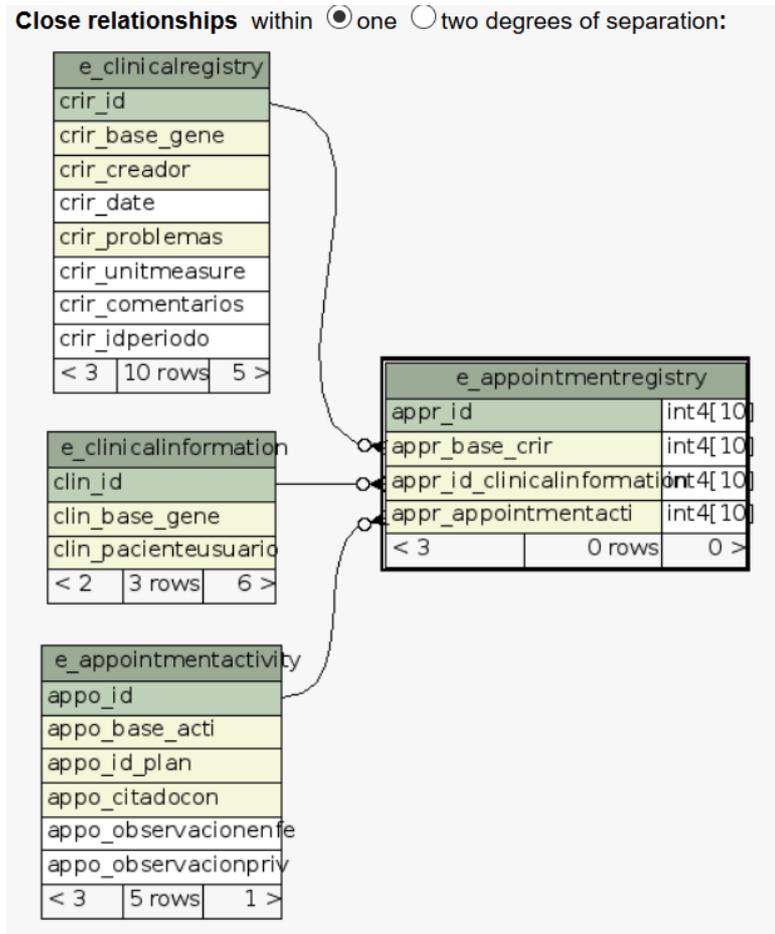


Figura 3-11. Modelo de datos de e-Nefro: Registros

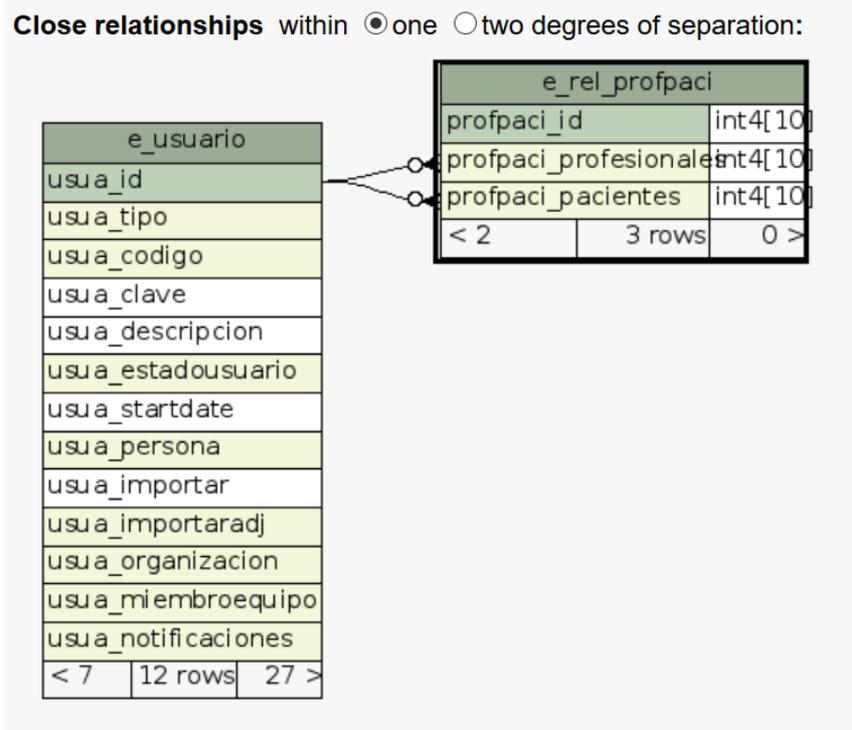


Figura 3-12. Modelo de datos de e-Nefro: Relación profesional-paciente

Del mismo modo, la tabla e_rel_espepaci (Figura 3-13) relaciona paciente y cuidador.

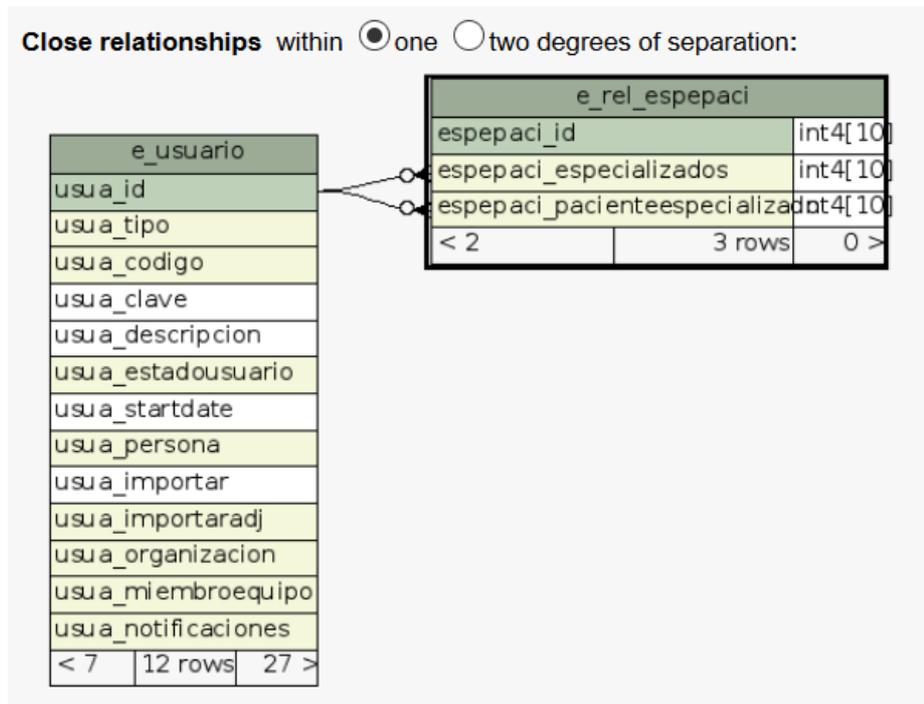


Figura 3-13. Modelo de datos de e-Nefro: Relación cuidador-paciente

Con esto, ya tenemos identificada la información existente en la base de datos. En los siguientes apartados, procederemos al modelado de los objetos de información definidos.

3.2.2.2. Objeto Paciente

En la Tabla 3-1 se muestra el objeto que contiene la información referente a un usuario de tipo paciente. Se identifican y describen sus atributos y dónde está almacenada cada elemento de información en la base de datos. También se añade la representación del objeto modelado.

Tabla 3-1. Objeto Paciente

Atributo	Relación con BD	Descripción	Objeto modelado
idPaciente	pers_id	Identificador del objeto	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Paciente</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idPaciente : int -estado : String -numHist : String -nombre : String -primerApellido : String -segundoApellido : String -tipoDocumento : String -documento : String -sexo : String -fechaNacimiento : date -telefono : String -telecom : String -direccion : String <p style="text-align: center;"><i>operations</i></p> <p style="text-align: center;">+Getters & Setters()</p> </div>
estado	usua_estadousuario	Estado del usuario en la aplicación	
numHist	pers_numhist	Número de historia clínica del paciente	
nombre	pers_name	Nombre del paciente	
primerApellido	pers_firstsurname	Primer apellido del paciente	
segundoApellido	pers_secondsurname	Segundo apellido del paciente	
tipoDocumento	tpid_codigo	Tipo de documento que identifica al paciente	

Atributo	Relación con BD	Descripción	Objeto modelado
documento	pers_document	Documento de identificación del paciente	
sexo	pers_sexo	Sexo del paciente	
fechaNacimiento	pers_birthtime	Fecha de nacimiento del paciente	
telefono	pers_telefono	Teléfono del paciente	
telecom	pers_telcom	Correo electrónico del paciente	
direccion	pers_address	Dirección del paciente	

3.2.2.3. Objeto Personal sanitario

Objeto que contiene la información almacenada en base de datos sobre un usuario del tipo Profesional o Cuidador.

Se emplea una única clase para ambos tipos de usuarios, ya que la información a extraer es la misma, y posteriormente veremos que FHIR tiene un único recurso definido tanto para profesionales sanitarios, como para cuidadores particulares, por lo que es más eficiente utilizar una misma clase.

Tabla 3-2. Objeto Personal sanitario

Atributo	Relación con BD	Descripción	Objeto modelado
idSanitario	pers_id	Identificador del objeto	<div style="border: 1px solid black; padding: 10px; background-color: #fff9c4;"> <p style="text-align: center;">Personal sanitario</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idSanitario : int -idOrganizacion : int -estado : String -tipo : String -nombre : String -primerApellido : String -segundoApellido : String -tpoDocumento : String -documento : String -sexo : String -fechaNacimiento : String -telefono : String -telecom : String -cargo : String -especialidad : String <p style="text-align: center;"><i>operations</i></p> <p>+Getters & Setters()</p> </div>
idOrganizacion	orga_id	Identificador de la organización del sanitario	
estado	usua_estadousuario	Estado del usuario en la aplicación	
tipo	usua_tipo	Tipo de personal sanitario	
nombre	pers_name	Nombre del sanitario	
primerApellido	pers_firstsurname	Primer apellido del sanitario	
segundoApellido	pers_secondsurname	Segundo apellido del sanitario	
tipoDocumento	tpid_codigo	Tipo de documento que identifica al sanitario	
documento	pers_document	Documento de identificación del sanitario	
sexo	pers_sexo	Sexo del profesional	
fechaNacimiento	pers_birthtime	Fecha de nacimiento del profesional	

Atributo	Relación con BD	Descripción	Objeto modelado
telefono	pers_telefono	Teléfono del profesional	
telecom	pers_telcom	Correo electrónico del profesional	
cargo	staf_cargo	Cargo que desempeña el profesional	
cargoEspecialidad	staf_nombre	Especialidad médica del profesional	

3.2.2.4. Objeto Organización

Objeto que contiene la información de base de datos de una organización implicada en la asistencia sanitaria del paciente.

Tabla 3-3. Objeto Organización

Atributo	Relación con BD	Descripción	Objeto modelado
idOrganizacion	orga_id	Identificador del objeto	<div style="border: 1px solid black; padding: 10px; background-color: #fff9c4;"> <p style="text-align: center;">Organizacion</p> <p style="text-align: center;"><i>attributes</i></p> <p>-idOrganizacion : int -nombre : String -direccion : String -telefono : String -telecom : String</p> <hr/> <p style="text-align: center;"><i>operations</i></p> <p>+Getters & Setters()</p> </div>
nombre	orga_name	Nombre de la organización	
direccion	orga_address	Dirección de la organización	
telefono	orga_telefono	Teléfono de la organización	
telecom	orga_telecom	Correo electrónico de la organización	

3.2.2.5. Objeto Cita

Objeto que contiene la información de base de datos relativa a la cita de un paciente con un profesional sanitario, lo que genera dicho encuentro.

Tabla 3-4. Objeto Cita

Atributo	Relación con BD	Descripción	Objeto modelado
idCita	appo_id	Identificador del objeto	<div style="border: 1px solid black; padding: 10px; background-color: #fff9c4;"> <p style="text-align: center;">Cita</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idCita : int -idPlan : int -idPaciente : int -idProfesional : int -localizacion : String -fechaInicio : date -causa : String -descripcion : String -obserMed : String -obserEnf : String -comenPac : String <p style="text-align: center;"><i>operations</i></p> <p style="text-align: center;">+Getters & Setters()</p> </div>
idPlan	appo_id_plan	Identificador del plan al que pertenece la actividad	
idPaciente	plan_pacienteusuario	Identificador del paciente implicado en la cita	
idProfesional	appo_id_citadocon	Identificador del profesional implicado en la cita	
localizacion	acti_location	Lugar de la cita	
fechaInicio	acti_startdate	Fecha de inicio de la cita	
causa	acti_problemaa	Razón por la que la cita tiene lugar	
descripcion	acti_descripcion	Descripción de la cita	
obserMed	appo_observacionpriv	Observaciones médicas realizadas durante la cita	
obserEnf	appo_observacionenfe	Observaciones de enfermería realizadas durante la cita	
comenPac	cric_comentarios	Comentarios para el paciente	

3.2.2.6. Objeto Actividad Toma de medicamento

Objeto que contiene la información de una actividad de tipo medicación de un paciente.

Tabla 3-5. Objeto Actividad Toma de medicación

Atributo	Relación con BD	Descripción	Objeto modelado
idMedAct	mact_id	Identificador el objeto	<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center;">Acividad Toma de medicamento</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idMedAct : int -idPlan : int -idPaciente : int -idPrescriptor : int -medicamento : String -dosis : String -unidadMedida : String -frecuencia : String -dias : String -cadaDias : String -cadaHoras : String -relComidas : String -fechaInicio : date -fechaFin : date -causa : String -descripcion : String <p style="text-align: center;"><i>operations</i></p> <p style="text-align: center;">+Getters & Setters()</p> </div>
idPlan	mact_id_plan	Identificador del plan al que pertenece la actividad	
idPaciente	plan_pacienteusuario	Id del paciente implicado en la toma de medicación	
idPrescriptor	acti_creador	Id del profesional que prescribe el medicamento	
medicamento	reso_codigo	Medicamento a tomar	
dosis	mact_amount	Dosis a tomar	
unidadMedida	mact_unitmeasure	Unidad de medida de la dosis	
frecuencia	acti_frequency	Frecuencia de la toma	
dias	acti_dias	Días que tiene que tomar el medicamento	
cadaDias	acti_cadaxdias	Cada cuántos días tiene que tomar el medicamento	
cadaHoras	acti_cadaxhoras	Cada cuántas horas tiene que tomar el medicamento	
relComida	acti_relcomidas	Relación de la toma con las comidas	
fechaInicio	acti_startdate	Fecha de inicio del tratamiento	
fechaFin	acti_enddate	Fecha de fin del tratamiento	
causa	acti_problemaa	Causa de la toma del medicamento	
descripcion	acti_description	Descripción de la actividad a realizar	

3.2.2.7. Objeto Registro Toma de medicamento

Objeto que contiene la información de un registro de tipo medicación de un paciente.

Tabla 3-6. Objeto Registro Toma de medicación

Atributo	Relación con BD	Descripción	Objeto modelado
idMedReg	macr_id	Identificador del objeto	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Registro Toma de medicamento</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idMedReg : int -idMedAct : int -fecha : date -comentario : String <p style="text-align: center;"><i>operations</i></p> <p style="text-align: center;">+Getters & Setters()</p> </div>
idMedAct	mact_id	Identificador de la actividad de la que se genera el registro	
fechaToma	crir_date	Fecha de creación del registro	
comentario	crir_comentarios	Comentarios sobre el registro	

3.2.2.8. Objeto Actividad Medida de variable

Objeto que contiene la información de una actividad de tipo medida de variable fisiológica de un paciente.

Tabla 3-7. Objeto Actividad Medida de variable

Atributo	Relación con BD	Descripción	Objeto modelado
idVarAct	phya_id	Identificador el objeto	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Actividad Medida de variable</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idVarAct : int -idPlan : int -idPaciente : int -idPrescriptor : int -variable : String -frecuencia : String -dias : String -cadaDias : String -cadaHoras : String -relComida : String -fechalnicio : date -fechaFin : date -causa : String -descripcion : String <p style="text-align: center;"><i>operations</i></p> <p style="text-align: center;">+Getters & Setters()</p> </div>
idPlan	phya_id_plan	Identificador del plan al que pertenece la actividad	
idPaciente	plan_pacienteusuario	Id del paciente implicado en la actividad	
idPrescriptor	acti_creador	Id del profesional que prescribe la actividad	
variableMedida	phys_codigo	Variable fisiológica medida	
frecuencia	acti_frequency	Frecuencia de la actividad	
dias	acti_dias	Días que tiene que realizar la actividad	
cadaDias	acti_cadaxdias	Cada cuántos días tiene que realizar la actividad	
cadaHoras	acti_cadaxhoras	Cada cuántas horas tiene que realizar la actividad	

Atributo	Relación con BD	Descripción	Objeto modelado
relComida	acti_relcomidas	Relación de la actividad con las comidas	
fechaInicio	acti_startdate	Fecha de inicio de realización de la actividad	
fechaFin	acti_enddate	Fecha de fin de realización de la actividad	
causa	acti_problemaa	Causa de tener que realizar la actividad	
descripcion	acti_description	Descripción de la actividad	

3.2.2.9. Objeto Registro Medida de variable

Objeto que contiene la información de un registro de tipo medida de variable fisiológica realizada por un paciente.

Tabla 3-8. Objeto Registro Medida de variable

Atributo	Relación con BD	Descripción	Objeto modelado
idVarReg	phyr_id	Identificador del objeto	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Registro Medida de variable</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -IdVarReg : int -IdVarAct : int -fecha : date -medida : String -comentario : String <p style="text-align: center;"><i>operations</i></p> <p style="text-align: center;">+Getters & Setters()</p> </div>
idVarAct	phya_id	Identificador de la actividad de la que se genera el registro	
medida	phyr_medida	Valor de la medida	
fecha	crir_date	Fecha de creación del registro	
comentario	crir_comentario	Comentarios sobre el registro	

3.2.2.10. Objeto Actividad Toma de imagen

Objeto que contiene la información de una actividad de tipo toma de imagen de un paciente.

Tabla 3-9. Objeto Actividad Toma de imagen

Atributo	Relación con BD	Descripción	Objeto modelado
idImgAct	take_id	Identificador el objeto	<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Actividad Toma de imagen</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idImgAct : int -idPlan : int -idPaciente : int -idPrescriptor : int -frecuencia : String -dias : String -cadaDias : String -cadaHoras : String -fechaInicio : date -fechaFin : date -causa : String -descripcion : String <p style="text-align: center;"><i>operations</i></p> <p>+Getters & Setters()</p> </div>
idPlan	take_id_plan	Identificador del plan al que pertenece la actividad	
idPaciente	plan_pacienteusuario	Identificador del paciente implicado en la actividad	
idPrescriptor	acti_creador	Id del profesional que prescribe la actividad	
causa	acti_problemaa	Causa de tener que realizar la actividad	
dias	acti_dias	Días que tiene que realizar la actividad	
cadaDias	acti_cadaxdias	Cada cuántos días tiene que realizar la actividad	
cadaHoras	acti_cadaxhoras	Cada cuántas horas tiene que realizar la actividad	
fechaInicio	acti_startdate	Fecha de inicio de realización de la actividad	
fechaFin	acti_enddate	Fecha de fin de realización de la actividad	
frecuencia	acti_frequency	Frecuencia de la actividad	
descripcion	acti_description	Descripción de la actividad	

3.2.2.11. Objeto Registro Toma de imagen

Objeto que contiene la información de un registro de tipo toma de imagen realizada por un paciente.

Tabla 3-10. Objeto Registro Toma de imagen

Atributo	Relación con BD	Descripción	Objeto modelado
idImgReg	takr_id	Identificador del objeto	<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center;">Registro Toma de imagen</p> <p style="text-align: center;"><i>attributes</i></p> <p>-IdImgReg : int -idImgAct : int -fecha : date -ruta : String -descripcion : String -comentario : String</p> <p style="text-align: center;"><i>operations</i></p> <p>+Getters & Setters()</p> </div>
idImgAct	take_id	Identificador de la actividad de la que se genera el registro	
fecha	docu_fechainsercion	Fecha de creación del registro	
ruta	docu_ruta	Ruta de la imagen tomada	
descripcion	takr_picture	Descripción de la imagen tomada	
comentario	crir_comentarios	Comentarios sobre el registro	

3.2.2.12. Objeto Incidencia

Objeto que contiene la información de una incidencia registrada por un paciente.

Tabla 3-11. Objeto Incidencia

Atributo	Relación con BD	Descripción	Objeto modelado
idIncidencia	heal_id	Identificador del objeto	<div style="border: 1px solid black; padding: 10px;"> <p style="text-align: center;">Incidencia</p> <p style="text-align: center;"><i>attributes</i></p> <p>-idIncidencia : int -idPaciente : int -fecha : date -descripcion : String -recomendacion : String -comentarios : String</p> <p style="text-align: center;"><i>operations</i></p> <p>+Getters & Setters()</p> </div>
idPaciente	clin_pacienteusuario	Identificador del paciente que registra la incidencia	
fecha	crir_date	Fecha de registro de la incidencia	
descripcion	tinc_descripcion	Descripción de la incidencia	
recomendacion	tinc_recomendacio	Recomendación a la incidencia	
comentarios	crir_comentario	Comentarios sobre la incidencia	

3.2.2.13. Objeto Plan de cuidados

Objeto que contiene la información del plan de cuidados seguido por un paciente.

Tabla 3-12. Objeto Plan de cuidados

Atributo	Relación con BD	Descripción	Objeto modelado
idPlan	plan_id	Identificador del objeto	<div style="border: 1px solid black; padding: 10px; background-color: #fff9c4;"> <p style="text-align: center;">Plan de cuidados</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idPlan : int -idPaciente : int -descripcion : String -razon : String -tiempoDefinido : date -estado : boolean <p style="text-align: center;"><i>operations</i></p> <p style="text-align: center;">+Getters & Setters()</p> </div>
idPaciente	plan_pacienteusuario	Identificador del paciente que sigue el plan de cuidados	
descripcion	plan_description	Descripción de plan de cuidados	
razon	plan_reason	Razón de existencia del plan de cuidados	
tiempoDefinido	plan_definitiontime	Tiempo definido para el plan de cuidados	
estado	plan_active	Estado del plan de cuidados	

3.2.2.14. Objeto Periodo de seguimiento

Objeto que contiene la información del periodo de seguimiento de un paciente.

Tabla 3-13. Objeto Periodo de seguimiento

Atributo	Relación con BD	Descripción	Objeto modelado
idPeriodo	peri_id	Identificador del objeto	<div style="border: 1px solid black; padding: 10px; background-color: #fff9c4;"> <p style="text-align: center;">Periodo de seguimiento</p> <p style="text-align: center;"><i>attributes</i></p> <ul style="list-style-type: none"> -idPeriodo : int -idPaciente : int -fechaInicio : date -fechaFin : date -problema : String -resumen : String <p style="text-align: center;"><i>operations</i></p> <p style="text-align: center;">+Getters & Setters()</p> </div>
idPaciente	peri_id_usuario	Identificador del paciente en seguimiento	
fechaInicio	peri_startdate	Fecha de inicio del periodo de seguimiento	
fechaFin	peri_enddate	Fecha de fin del periodo de seguimiento	
problema	peri_problema	Problema que origina el periodo de seguimiento	
resumen	peri_resumen	Descripción del periodo de seguimiento	

3.2.2.15. Relaciones entre objetos

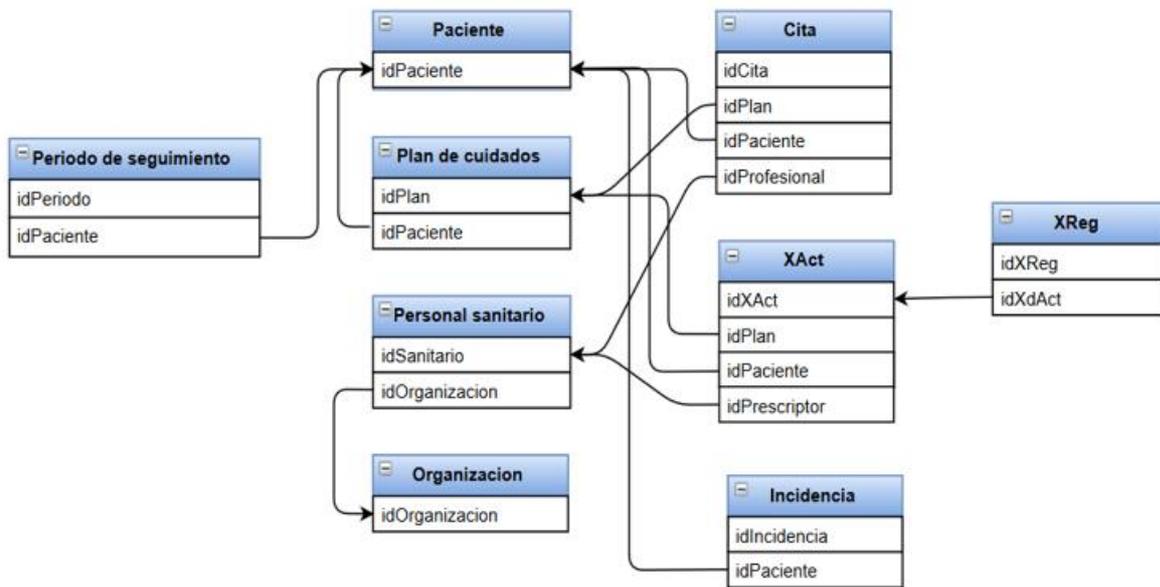


Figura 3-14. Relaciones entre objetos de información

Los objetos de información se relacionan entre sí mediante los atributos que siguen la nomenclatura idX (Figura 3-14). Con ellos, es posible obtener el resto de la información necesaria para crear un determinado recurso, realizando las subconsultas pertinentes.

Por ejemplo, si quisiéramos obtener la información de un profesional sanitario, es necesario, además de su información personal, incluir a qué organización pertenece, y los datos de ésta. Para ello, construiríamos un objeto Personal sanitario, y a partir de su atributo idOrganizacion, haríamos una nueva consulta a base de datos y construiríamos un objeto del tipo Organización. Con ambos objetos, ya tendríamos toda la información referente al profesional en cuestión, que se modelaría a un recurso FHIR, lo cual se verá en el siguiente capítulo.

En el siguiente diagrama, vemos las relaciones existentes entre los objetos. Por simplicidad, sólo se han incluido los atributos ya mencionados utilizados para las relaciones, así como una única clase que engloba a los tres tipos de actividades y registros.

3.3. Identificación y modelado de los Recursos FHIR

Una vez definidos y modelados los objetos de información, podemos pasar a modelar los recursos FHIR que utilizará nuestro servicio para enviar la información. Habrá que dar forma a la información contenida en estos objetos ya definidos, de acuerdo a lo que el estándar propone.

Por ello, el primer paso será hacer un estudio de qué recursos ofrece el estándar [6], viendo cuáles son interesantes para un sistema de este tipo, así como de sus campos de información, para ver con qué atributos de nuestros objetos se corresponden, y si es necesario extender dicho recurso.

Tras esto, se pasará a modelar los recursos FHIR que se intercambiarán con el resto de sistemas, indicando los atributos que contendrá, y el tipo de datos y cardinalidad de éstos.

3.3.1. Identificación de los Recursos FHIR

Para la identificación de los Recursos FHIR que va a implementar nuestro servicio web, se ha realizado un profundo estudio de la lista de recursos que ofrece la especificación [9], viendo cuáles se ajustan más a los objetos de información identificados, e intentando evitar las extensiones.

En total, se van a emplear catorce Recursos FHIR, habiendo tenido que extender sólo cuatro de ellos. En la Tabla 3-14 se pueden ver estos recursos, donde se especifica a qué objeto propio corresponde la información que contienen, y si han sido extendidos.

Tabla 3-14. Lista de recursos FHIR

Recurso FHIR	Objeto de Información	Extendido
Patient	Paciente	No
Practitioner	Personal sanitario Cuidador	No
Organization	Organización	No
Appointment	Cita	No
Encounter	Cita	Sí
Location	Cita	No
MedicationOrder	Actividad Toma de medicamento	No
MedicationAdministration	Registro Toma de medicamento	No
ProcedureRequest	Actividad Medida de variable Actividad Toma de imagen	Sí
Observation	Registro Medida de variable Registro Toma de imagen	No
DetectedIssue	Incidencia	Sí
CarePlan	Plan de cuidados	No
EpisodeOfCare	Periodo de seguimiento	Sí
Condition	Plan de cuidados Periodo de seguimiento	No

En el siguiente apartado, se verá cómo se han modelado estos recursos, explicando cuál es su funcionalidad en FHIR, y qué información van a contener en nuestro caso práctico.

3.3.2. Modelado de los Recursos FHIR

3.3.2.1. Recurso Patient

Este recurso contiene la información relativa a los pacientes que participan en alguna actividad relacionada con la salud, incluyendo, actividades sanitarias, atención psiquiátrica, servicios sociales, cuidados durante el embarazo, enfermería y de vivienda asistida, servicios dietéticos, seguimiento de la salud personal y datos de ejercicios. Sus atributos se centran en la información necesaria para apoyar los procedimientos clínicos, administrativos, financieros y logísticos que tienen lugar en torno al paciente.

En nuestro caso, contiene la información relativa al **objeto Paciente**, no siendo necesario extender el recurso para ello.

Tabla 3-15. Recurso Patient

Atributo	Tipo de Dato	Cardinalidad	Descripción
active	boolean	0..1	Indica si el registro del paciente sigue activo.
identifier	Identifier	0..*	Identificador del paciente.
name	HumanName	0..*	Nombre y apellidos del paciente.
gender	code	0..1	Sexo del paciente.
birthDate	date	0..1	Fecha de nacimiento del paciente.
telecom	ContactPoint	0..*	Teléfono/Correo electrónico del paciente.
address	Address	0..*	Dirección del paciente.

3.3.2.2. Recurso Practitioner

Este recurso engloba a todas las personas que están involucradas en el proceso de cuidado de la salud de un paciente, como parte de sus actividades y responsabilidades formales. Entre otros roles, contempla ser utilizado para el personal médico, personal de enfermería y cuidadores.

En nuestro caso, contendrá la información de los **objetos Personal sanitario** y **Cuidador**, no siendo necesario extender el recurso para ello.

Tabla 3-16. Recurso Practitioner

Atributo	Tipo de Dato	Cardinalidad	Descripción
active	boolean	0..1	Indica si el profesional sigue activo.
identifier	Identifier	0..*	Identificador del profesional.
name	HumanName	0..*	Nombre y apellidos del profesional.
gender	code	0..1	Sexo del profesional.
birthDate	date	0..1	Fecha de nacimiento del profesional.
telecom	ContactPoint	0..*	Teléfono/Correo electrónico del profesional.
practitionerRole	BackboneElement	0..*	Cargo, especialidad y organización del profesional.

Contiene un recurso del tipo Organization, organización sanitaria a la que pertenece el profesional, referenciado en el campo *managingOrganization*



Figura 3-15. Referencias en el recurso Practitioner

3.3.2.3. Recurso Organization

Este recurso contiene la información relativa a las organizaciones que pueden formar parte de la asistencia sanitaria a un paciente. Por lo general, suele utilizarse contenido en otros recursos.

En nuestro caso, contendrá la información del **objeto Organización**, sin ser necesario extenderlo, y, como se ha visto anteriormente, irá contenido dentro del recurso Practitioner.

Tabla 3-17. Recurso Organization

Atributo	Tipo de Dato	Cardinalidad	Descripción
type	CodeableConcept	0..1	Tipo de organización.
name	string	0..1	Nombre de la organización.
address	Address	0..*	Dirección de la organización.
telecom	ContactPoint	0..*	Teléfono/Correo electrónico de la organización.

3.3.2.4. Recurso Appointment

Este recurso contiene la información de la cita de un paciente con un profesional sanitario, conteniendo información como los participantes implicados (paciente, profesional y localización), cuándo tendrá lugar o causa de la cita.

En nuestro caso, contiene parte de la información del **objeto Cita**, pues se excluye la referente a las observaciones y comentarios realizados durante esta, que irá contenida en un recurso más apropiado, por lo que no es necesario extenderlo.

Tabla 3-18. Recurso Appointment

Atributo	Tipo de Dato	Cardinalidad	Descripción
participant	BackboneElement	0..*	Participantes de la cita: Paciente, Profesionales y Localización.
start	instant	0..1	Instante de comienzo de la cita.
reason	CodeableConcept	0..1	Causa de la cita.
description	string	0..1	Descripción de la cita.

Así mismo, en este recurso van contenidos un recurso Patient, paciente implicado en la cita, un recurso Practitioner, profesional con el que está citado el paciente, y un recurso Location, lugar donde ocurre la cita, ambos referenciados en el campo *actor* (Figura 3-16).

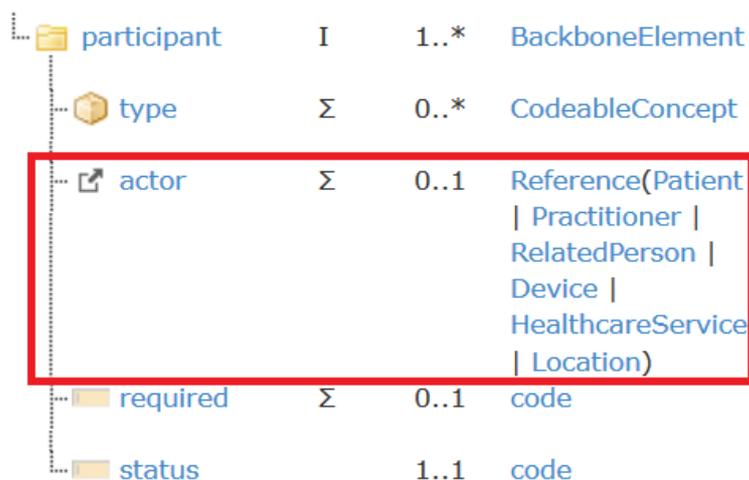


Figura 3-16. Referencias en el recurso Appointment

3.3.2.5. Recurso Encounter

Este recurso contiene la información del encuentro entre un paciente y un profesional sanitario, conteniendo información sobre sus participantes, cuándo y dónde tuvo lugar, la razón del encuentro, y las observaciones y comentarios realizados durante este.

En nuestro caso, contiene la información del **objeto Cita**, siendo necesario extender tres campos de información: *medicalComment*, *nursingComment* y *generalComment*; que contendrán las observaciones y comentarios realizadas por los profesionales durante la cita.

Tabla 3-19. Recurso Encounter

Atributo	Tipo de Dato	Cardinalidad	Descripcion
patient	Reference(Patient)	0..1	Paciente que participa en el encuentro.
participant	BackboneElement	0..*	Profesional que participa en el encuentro.
location	Reference(Patient)	0..*	Lugar del encuentro.
appointment	Reference(Appointment)	0..1	Cita que origina el encuentro.
medicalComment	string	0..1	Observaciones del profesional médico.
nursingComment	string	0..1	Observaciones de enfermería.
generalComment	string	0..1	Comentarios para el paciente.

Así mismo, en este recurso van contenidos un recurso Patient, paciente implicado en el encuentro, un recurso Practitioner, profesional implicado en el encuentro, un recurso Location, lugar donde ocurre el encuentro, y un recurso Appointment, cita que origina el encuentro; referenciados en los campos *patient*, *individual*, *location* y *appointment*, respectivamente (Figura 3-17).

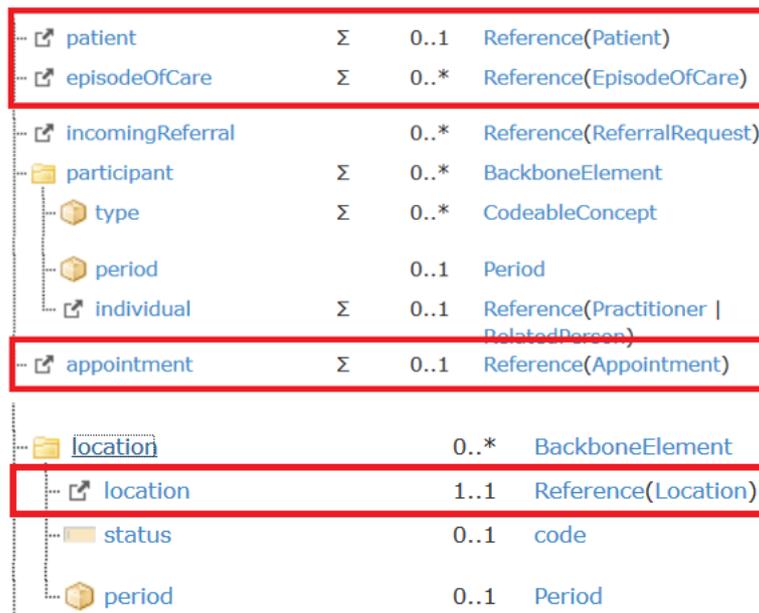


Figura 3-17. Referencias en el recurso Encounter

3.3.2.6. Recurso Location

Este recurso contiene la información relativa a lugares que pueden formar parte de la asistencia sanitaria a un paciente. Por lo general, suele utilizarse contenido en otros recursos.

En nuestro caso, contiene parte de la información del **objeto Cita**, concretamente la localización donde tiene lugar

Tabla 3-20. Recurso Location

Atributo	Tipo de Dato	Cardinalidad	Descripción
name	string	0..1	Nombre de la localización.

3.3.2.7. Recurso MedicationOrder

Este recurso se utiliza para prescribir el suministro de un medicamento al paciente, así como las instrucciones para su administración (frecuencia, dosis...).

En nuestro caso, contendrá la información del **objeto Actividad Toma de medicamento**, no siendo necesario extender el recurso.

Tabla 3-21. Recurso MedicationOrder

Atributo	Tipo de Dato	Cardinalidad	Descripción
patient	Reference(Patient)	0..1	Paciente al que se le prescribe el medicamento.
prescriber	Reference(Patient)	0..1	Prescriptor del medicamento.
medication	CodeableConcept	1..1	Medicamento prescrito.
reason	CodeableConcept	0..1	Causa de la prescripción.
dosageInstruction	BackboneElement	0..*	Instrucciones de la toma.

Atributo	Tipo de Dato	Cardinalidad	Descripcion
dateWritten	dateTime	0..1	Fecha de comienzo del tratamiento.
dateEnded	dateTime	0..1	Fecha de fin del tratamiento.
note	string	0..1	Información sobre la prescripción.

Así mismo, en este recurso van contenidos un recurso Patient, paciente que ha de tomar el medicamento, y un recurso Practitioner, profesional que lo prescribe, en los campos *patient* y *prescriber* respectivamente (Figura 3-18).

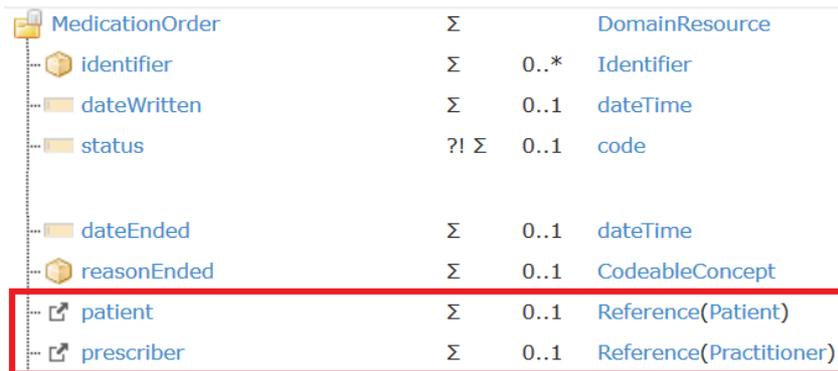


Figura 3-18. Referencias en el recurso MedicationOrder

3.3.2.8. Recurso MedicationStatement

Este recurso es utilizado para los registros generados por la toma de medicamentos de un paciente. Contiene información como el medicamento tomado, la fecha de la toma o comentarios sobre ésta.

En nuestro caso, contiene la información relativa al **objeto Registro Toma de medicamento**, no siendo necesario extender el recurso.

Tabla 3-22. Recurso MedicationStatement

Atributo	Tipo de Dato	Cardinalidad	Descripcion
status	Reference(Patient)	1..1	Estado en el que se encuentra el tratamiento.
patient	Reference(Patient)	1..1	Paciente al que se le prescribe el medicamento.
medication	CodeableConcept	1..1	Medicamento a tomar.
effectiveTime	dateTime	1..1	Fecha de la toma del medicamento.
note	string	0..1	Información sobre la toma.

Así mismo, contiene un recurso Patient, siendo éste el paciente implicado en la toma de medicamento, referenciado a través del campo *patient* (Figura 3-19).



Figura 3-19. Referencias en el recurso MedicationStatement

3.3.2.9. Recurso ProcedureRequest

Este recurso se utiliza para la solicitud de realización de alguna actividad que se lleva a cabo con o en un paciente como parte de su plan de cuidados.

En nuestro caso, contiene la información de los **objetos Actividad Medida de variable** y **Actividad Toma de imagen**, siendo necesario extender un campo *frequency*, que contiene la frecuencia con la que esta actividad debe ser realizada.

Tabla 3-23. Recurso ProcedureRequest

Atributo	Tipo de Dato	Cardinalidad	Descripcion
subject	Reference(Patient)	0..1	Paciente sobre el que se realiza la actividad.
orderer	Reference(Practitioner)	0..1	Profesional que manda la actividad.
code	CodeableConcept	1..1	Tipo de observación
reason	CodeableConcept	0..1	Causa de tener que realizar la actividad.
scheduledPeriod	Period	0..1	Durante cuánto tiempo se tiene que realizar la actividad.
notes	string	0..1	Descripción de la actividad.
frequency	string	1..1	Frecuencia con la que hay que realizar la actividad.

Así mismo, contiene un recurso Patient, paciente sobre el que se realiza la actividad, y un recurso Practitioner, el profesional que la prescribió, en los campos *subject* y *orderer* (Figura 3-20).



Figura 3-20. Referencias en el recurso ProcedureRequest

3.3.2.10. Recurso Observation

Dentro de este recurso se engloban todos los resultados obtenidos de las actividades realizadas con o sobre el paciente. Entre ellos, encontramos medidas de signos vitales, datos de laboratorio o imágenes médicas.

En nuestro caso, contiene la información de los **objetos Registro Medida de variable** y **Registro Toma de imagen**, no siendo necesario extenderlo.

Tabla 3-24. Recurso Observation

Atributo	Tipo de Dato	Cardinalidad	Descripcion
status	code	1..1	Estado de la actividad
subject	Reference(Patient)	0..1	Paciente que realiza la actividad.
code	CodeableConcept	1..1	Tipo de observación
effective	dateTime	0..1	Momento en el que se realiza la actividad.
value	string Attachment	1..*	Valor de la medida. Imagen adjunta.
comments	string	0..1	Comentarios sobre la realización de actividad.

Aunque se utiliza el mismo recurso para ambos objetos de información, en el caso de la medida de variable, el campo *value* será de tipo string, conteniendo el valor de la medida, mientras que, en la toma de imagen, será del tipo Attachment, conteniendo la ruta de la imagen tomada.

Así mismo, contiene un recurso Patient, siendo el paciente sobre el que se realizó la actividad y del que se ha generado el registro, referenciado a través del campo *subject* (Figura 3-21).



Figura 3-21. Referencias en el recurso Observation

3.3.2.11. Recurso DetectedIssue

Se emplea para indicar un problema o evento anómalo al margen de las actividades que realiza el paciente (por ejemplo, dolores abdominales, náuseas o fiebre).

En nuestro caso, contiene la información de un **objeto Incidencia**, siendo necesario extender un campo *comments*, para los comentarios realizados en el registro de la incidencia.

Tabla 3-25. Recurso DetectedIssue

Atributos	Tipo de Dato	Cardinalidad	Descripcion
patient	Reference(Patient)	0..1	Paciente que registra la incidencia.
date	dateTime	0..1	Fecha de registro de la incidencia.
detail	string	0..1	Descripción de la incidencia.
mitigation	BackboneElement	0..*	Momento en el que se realiza la actividad.
comments	string	0..1	Comentarios sobre la incidencia.

Contiene un recurso Patient, el paciente que registra la incidencia, referenciado a través del campo *patient* (Figura 3-22).



Figura 3-22. Referencias en el recurso DetectedIssue

3.3.2.12. Recurso CarePlan

Este recurso contiene la información del plan de cuidados que sigue un paciente, donde se enmarcan las actividades que debe realizar para el cuidado de su salud.

En nuestro caso, contiene la información del **objeto Plan de cuidados**, no siendo necesario extender el recurso.

Tabla 3-26. Recurso CarePlan

Atributos	Tipo de Dato	Cardinalidad	Descripción
subject	Reference(Patient)	0..1	Paciente que registra la incidencia.
description	string	0..1	Descripción del plan de cuidados.
status	string	1..1	Estado del plan de cuidados.
period	Period	0..1	Duración del plan de cuidados.
addresses	Reference(Condition)	0..*	Problemas que trata el plan de cuidados.

Contiene un recurso Patient, el paciente que registra la incidencia, y un recurso Condition, problema que trata de cubrir el plan de cuidados, referenciados en los campos *subject* y *addresses*, respectivamente (Figura 3-23).

The screenshot shows the FHIR CarePlan resource definition with the following fields and their details:

Field	Type	Cardinality	Reference
identifier	Identifier	Σ 0..*	Identifier
subject	Reference(Patient Group)	Σ 0..1	Reference(Patient Group)
status	code	?! Σ 1..1	code
context	Reference(Encounter EpisodeOfCare)	Σ 0..1	Reference(Encounter EpisodeOfCare)
period	Period	Σ 0..1	Period
author	Reference(Patient Practitioner RelatedPerson Organization)	Σ 0..*	Reference(Patient Practitioner RelatedPerson Organization)
modified	dateTime	Σ 0..1	dateTime
category	CodeableConcept	Σ 0..*	CodeableConcept
description	string	Σ 0..1	string
addresses	Reference(Condition)	Σ 0..*	Reference(Condition)

Figura 3-23. Referencias en el recurso CarePlan

3.3.2.13. Recurso EpisodeOfCare

Contiene información acerca de la asociación de un paciente con un proveedor de atención médica durante un período de tiempo en el que pueden producirse actividades y encuentros médicos.

En nuestro caso, contiene la información del **objeto Periodo de seguimiento**, siendo necesario extender un campo *reasonEnded*, que indica por qué finalizó dicho periodo de seguimiento.

Tabla 3-27. Recurso EpisodeOfCare

EpisodeOfCare	Tipo de Dato	Cardinalidad	Descripcion
status	code	1..1	Estado del periodo de seguimiento.
patient	Reference(Patient)	0..1	Paciente que registra la incidencia.
period	Period	0..1	Descripción del plan de cuidados.
condition	Reference(Condition)	0..*	Problemas de salud por el que existe este periodo de seguimiento.
reasonEnded	string	0..1	Causa del fin del periodo de seguimiento.

Contiene un recurso Patient, el paciente que registra la incidencia, y un recurso Condition, problema que trata de cubrir el plan de cuidados, referenciados en los campos *subject* y *condition*, respectivamente (Figura 3-24).

type	Σ	0..*	CodeableConcept
condition		0..*	Reference(Condition)
patient	Σ	1..1	Reference(Patient)
managingOrganization	Σ	0..1	Reference(Organization)
period	Σ	0..1	Period
referralRequest		0..*	Reference(ReferralRequest)

Figura 3-24. Referencias en el recurso EpisodeOfCare

3.3.2.14. Recurso Condition

Se utiliza para enfermedades o problemas médicos que padece un paciente y que originan que tenga que seguir un determinado tratamiento o actividades para el cuidado de su salud.

En nuestro caso, contiene parte de la información de los **objetos Plan de cuidados y Periodo de seguimiento**, concretamente, la enfermedad que causa que el paciente forme parte de ellos, no siendo necesario extender el recurso.

Tabla 3-28. Recurso Condition

Condition	Tipo de Dato	Cardinalidad	Descripcion
patient	Reference(Patient)	1..1	Paciente que sufre el problema
code	CodeableConcept	1..1	Problema de salud.

3.4. Definición de los métodos de acceso a la información

Una vez identificados y modelados los recursos que se van a ofrecer al resto de sistemas, hay que definir los métodos de los que podrán hacer uso los clientes para acceder a ellos.

3.4.1. Forma de acceso a la información

Como se ha comentado, FHIR se basa en el intercambio de recursos que contienen la información solicitada. Debido a este enfoque de construir la especificación en torno a estos elementos básicos, es necesario definir los métodos de acceso a la información para cada recurso de manera independiente. De este modo, el cliente, formalmente, estará solicitando únicamente un tipo de recurso concreto utilizando los métodos definidos, recursos contenidos aparte.

Los métodos definidos son operaciones *Search*, interacción definida por la propia especificación para que los clientes puedan acceder a recursos siguiendo algún criterio de búsqueda. Esta interacción puede llevarse a cabo con métodos GET o POST indistintamente [53] (Figura 3-25).

```
GET [base]/[resourcetype]?name=value&...
```

```
POST [base]/[type]/_search{?[parameters]{&_format=[mime-type]}}
```

Figura 3-25. Sintaxis GET/POST de la operación Search

Cuando un cliente solicita un determinado recurso, lo que recibe es un recurso de tipo *Bundle* [54], que contendrá los recursos que el cliente había solicitado, y con el que podrá trabajar para extraer dicha información.

3.4.2. Lista de métodos de acceso a los recursos FHIR

En este apartado, se mostrará la lista completa de los métodos con los que contarán los clientes para acceder a la información, clasificados según el tipo de recurso suministrado, a través de una serie de tablas que contendrán el nombre del método, una breve descripción, los parámetros de entrada necesarios para la consulta, la respuesta esperada y los posibles errores.

3.4.2.1. Métodos para el recurso Patient

Contamos con dos métodos para acceder a la información modelada en el recurso Patient, el primero para solicitar un paciente concreto, y el segundo para solicitar todos los pacientes a cargo de un profesional.

Tabla 3-29. Método para el recurso Patient (I)

Nombre del método	findPatientByNUH
Descripción	Devuelve el paciente identificado por el número de historia pasado como parámetro.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-30. Método para el recurso Patient (II)

Nombre del método	findPatientByPractitioner
Descripción	Devuelve los pacientes a cargo del profesional identificado por el DNI pasado como parámetro.
Parámetros de entrada	DNI (String, obligatorio): Documento Nacional de Identidad del profesional.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.2. Métodos para el recurso Practitioner

Contamos con un método para acceder a la información modelada en el recurso Practitioner, con el que se obtiene la información del profesional solicitado.

Tabla 3-31. Método para el recurso Practitioner

Nombre del método	findPractitionerByID
Descripción	Devuelve el profesional identificado con el Documento Nacional de Identidad pasado como parámetro.
Parámetros de entrada	DNI (String, obligatorio): Documento Nacional de Identidad del profesional.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.3. Métodos para el recurso Appointment

Contamos con cuatro métodos para acceder a la información modelada en el recurso Appointment. Con el primero, se obtienen todas las citas futuras del paciente, mientras que, con el segundo, obtenemos aquellas comprendidas entre las dos fechas pasadas como parámetros. De manera análoga, tendremos otros dos métodos para obtener las citas que tiene concertadas un profesional.

Tabla 3-32. Método para el recurso Appointment (I)

Nombre del método	findAppointmentByPatient
Descripción	Devuelve todas las citas futuras del paciente.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-33. Método para el recurso Appointment (II)

Nombre del método	findAppoinmentByPatDate
Descripción	Devuelve las citas del paciente comprendidas entre las dos fechas pasadas como parámetros.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
	Desde (Date, obligatorio): Fecha desde la que iniciar la búsqueda.
	Hasta (Date, obligatorio): Fecha hasta la que parar la búsqueda.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-34. Método para el recurso Appointment (III)

Nombre del método	findAppointmentByPractitioner
Descripción	Devuelve todas las citas futuras del profesional.
Parámetros de entrada	DNI (String, obligatorio): Documento Nacional de Identidad del profesional.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-35. Método para el recurso Appointment (IV)

Nombre del método	findAppointmentByPracDate
Descripción	Devuelve las citas del profesional, comprendidas entre las dos fechas pasadas como parámetros.
Parámetros de entrada	DNI (String, obligatorio): Documento Nacional de Identidad del profesional.
	Desde (Date, obligatorio): Fecha desde la que iniciar la búsqueda.
	Hasta (Date, obligatorio): Fecha hasta la que parar la búsqueda.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.4. Métodos para el recurso Encounter

Contamos con cuatro métodos para acceder a la información modelada en el recurso Encounter. Con el primero, se obtienen todos los encuentros que ha realizado el paciente, mientras que, con el segundo, obtenemos aquellos comprendidos entre las dos fechas pasadas como parámetros. De manera análoga, tendremos otros dos métodos para obtener los encuentros en los que ha participado un profesional.

Tabla 3-36. Método para el recurso Encounter (I)

Nombre del método	findEncounterByPatient
Descripción	Devuelve todas las citas del paciente.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-37. Método para el recurso Encounter (II)

Nombre del método	findEncounterByPatDate
Descripción	Devuelve las citas del paciente comprendidas entre las dos fechas pasadas como parámetros.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.

	Desde (Date, obligatorio): Fecha desde la que iniciar la búsqueda.
	Hasta (Date, obligatorio): Fecha hasta la que parar la búsqueda.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-38. Método para el recurso Encounter (III)

Nombre del método	findEncounterByPractitioner
Descripción	Devuelve todas las citas del profesional.
Parámetros de entrada	DNI (String, obligatorio): Documento Nacional de Identidad del profesional.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-39. Método para el recurso Encounter (IV)

Nombre del método	findEncounterByPracDate
Descripción	Devuelve las citas del profesional, comprendidas entre las dos fechas pasadas como parámetros.
Parámetros de entrada	DNI (String, obligatorio): Documento Nacional de Identidad del profesional.
	Desde (Date, obligatorio): Fecha desde la que iniciar la búsqueda.
	Hasta (Date, obligatorio): Fecha hasta la que parar la búsqueda.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.5. Métodos para el recurso MedicationOrder

Contamos con dos métodos para acceder a la información modelada en el recurso MedicationOrder. Con el primero, se obtienen todas las actividades de este tipo del plan actual que sigue un paciente, mientras que, con el segundo, obtenemos las del plan pasado que pasemos como parámetro.

Tabla 3-40. Método para el recurso MedicationOrder (I)

Nombre del método	findMedOrderByNUH
Descripción	Devuelve las actividades de toma de medicamento del plan actual del paciente.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-41. Método para el recurso MedicationOrder (II)

Nombre del método	findMedOrderByPlan
Descripción	Devuelve las actividades de toma de medicamento del plan pasado como parámetro.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
	PLAN (String, obligatorio): Identificador del plan del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.6. Métodos para el recurso MedicationStatement

Contamos con dos métodos para acceder a la información modelada en el recurso MedicationStatement. Con el primero, se obtienen todos los registros de este tipo registrados por el paciente, mientras que, con el segundo, obtenemos aquellos comprendidos entre las dos fechas pasadas como parámetros.

Tabla 3-42. Método para el recurso MedicationStatement (I)

Nombre del método	findMedStatByNUH
Descripción	Devuelve todos los registros de toma de medicamento del paciente.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-43. Método para el recurso MedicationStatement (II)

Nombre del método	findMedStatByDate
Descripción	Devuelve los registros de toma de medicamento del paciente, comprendidos entre las dos fechas pasadas como parámetros.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
	Desde (Date, obligatorio): Fecha desde la que iniciar la búsqueda.
	Hasta (Date, obligatorio): Fecha hasta la que parar la búsqueda.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.7. Métodos para el recurso ProcedureRequest

Contamos con dos métodos para acceder a la información modelada en el recurso ProcedureRequest. Con el primero, se obtienen todas las actividades de medida de variable y toma de imagen del plan actual que sigue un paciente, mientras que, con el segundo, obtenemos las del plan que pasemos como parámetro.

Tabla 3-44. Método para el recurso ProcedureRequest (I)

Nombre del método	findProcReqByNUH
Descripción	Devuelve las actividades de medida de variable y toma de imagen del plan actual del paciente.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos

	solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-45. Método para el recurso ProcedureRequest (II)

Nombre del método	findProcReqByPlan
Descripción	Devuelve las actividades de medida de variable y toma de imagen del plan pasado como parámetro.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
	PLAN (String, obligatorio): Identificador del plan del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.7. Métodos para el recurso Observation

Contamos con dos métodos para acceder a la información modelada en el recurso Observation. Con el primero, se obtienen todos los registros de medida de variable y toma de imagen del paciente, mientras que, con el segundo, obtenemos aquellos comprendidos entre las dos fechas pasadas como parámetros.

Tabla 3-46. Método para el recurso Observation (I)

Nombre del método	findObservationByNUH
Descripción	Devuelve todos los registros de medida de variable y toma de imagen del paciente.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-47. Método para el recurso Observation (II)

Nombre del método	findObservationByDate
Descripción	Devuelve los registros de medida de variable y toma de imagen del paciente, comprendidos entre dos fechas.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
	Desde (Date, obligatorio): Fecha desde la que iniciar la búsqueda.
	Hasta (Date, obligatorio): Fecha hasta la que parar la búsqueda.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.8. Métodos para el recurso DetectedIssue

Contamos con dos métodos para acceder a la información modelada en el recurso DetectedIssue. Con el primero, se obtienen todos los registros de incidencia del paciente, mientras que, con el segundo, obtenemos aquellos comprendidos entre las dos fechas pasadas como parámetros.

Tabla 3-48. Método para el recurso DetectedIssue (I)

Nombre del método	findDetectedIssueByNUH
Descripción	Devuelve todos los registros de incidencia del paciente.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

Tabla 3-49. Método para el recurso DetectedIssue (II)

Nombre del método	findDetectedIssueByDate
Descripción	Devuelve los registros de incidencia del paciente, comprendidos entre las dos fechas pasadas como parámetros.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
	Desde (Date, obligatorio): Fecha desde la que iniciar la búsqueda.

	Hasta (Date, obligatorio): Fecha hasta la que parar la búsqueda.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.9. Métodos para el recurso CarePlan

Contamos con un método para acceder a la información modelada en el recurso CarePlan, con el que se obtiene la información de los planes de cuidados que ha seguido un paciente, incluido el actual.

Tabla 3-50. Método para el recurso CarePlan

Nombre del método	findCarePlanByNUH
Descripción	Devuelve los planes de cuidado que ha seguido un paciente, incluido el actual.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.4.2.10. Métodos para el recurso EpisodeOfCare

Contamos con un método para acceder a la información modelada en el recurso EpisodeOfCare, con el que se obtiene la información de los periodos de seguimiento que han sido establecidos para un paciente, incluido el actual.

Tabla 3-51 Método para el recurso EpisodeOfCare

Nombre del método	findEpisCareByNUH
Descripción	Devuelve los periodos de seguimiento que han establecido para el paciente, incluido el actual.
Parámetros de entrada	NUH (String, obligatorio): Número de historia del paciente.
Respuesta	data (XML Object, obligatorio): Contenedor con los recursos solicitados.
Posibles errores	Límite de tiempo excedido.
	Error en el servidor.

3.5. Implementación del servicio web

Para cerrar el capítulo, se describirá la implementación del servicio web, partiendo por su diseño, a nivel de funcionalidad y estructura, e integrando los elementos que se han diseñado en los apartados anteriores. Tras esto, se pasará a ver los aspectos más importantes de su desarrollo. Por último, se detallarán y expondrán las pruebas realizadas sobre la implementación.

3.5.1. Diseño del servicio web

Aunque la Figura 1-1 representaba la funcionalidad del servicio web, es interesante completarla con los elementos que se han ido diseñando en los apartados anteriores (objetos propios, recursos FHIR y métodos de acceso), para ver qué papel juegan en ésta. Para ello, se ha creado un diagrama de secuencia, que podemos ver en la Figura 3-26.

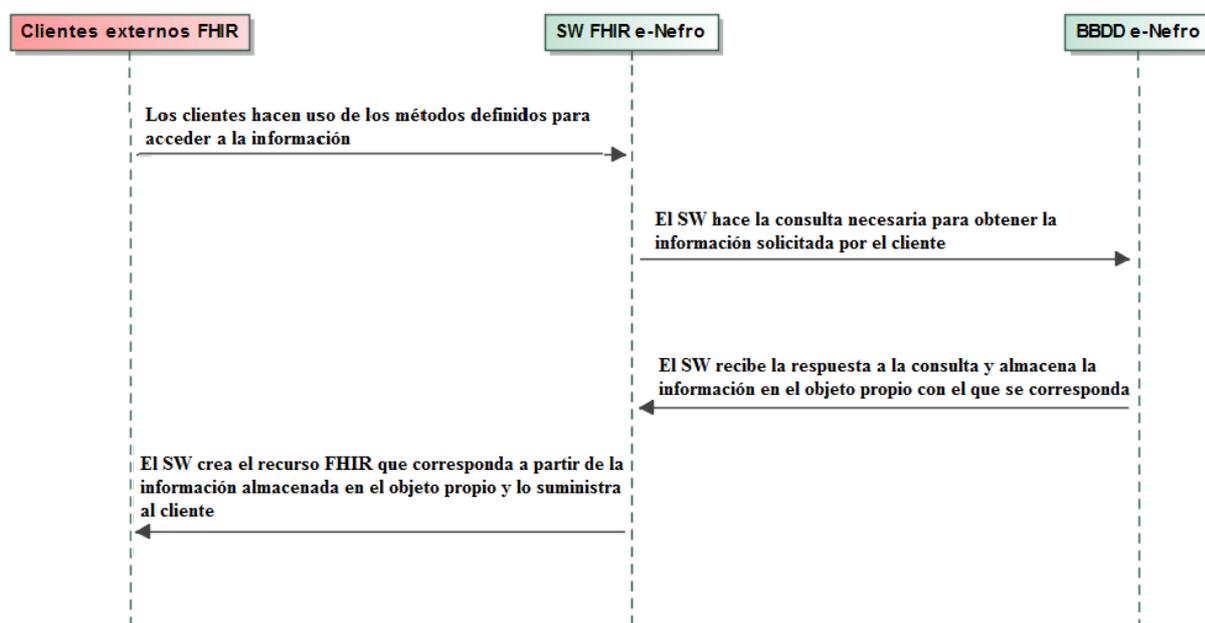


Figura 3-26. Diagrama de secuencia de la funcionalidad del servicio web

Los clientes solicitan la información almacenada en la base de datos mediante los métodos de acceso a recursos, utilizando uno u otro en función de qué información requieran. Cuando el servicio web recibe la petición, realiza la pertinente consulta a la base de datos de e-Nefro, almacenando la información que recibe en el objeto propio que corresponda. Por último, se crea el recurso FHIR que se suministrará al cliente, utilizando para ello la información almacenada en el objeto propio.

El servicio web se ha diseñado buscando una arquitectura modular y adaptable, a fin de que sea cómodo trabajar con él, así como ser reutilizado para otros sistemas, añadiendo nuevos módulos si fuera necesario. Por ello, se han agrupado los distintos archivos que forman el proyecto en paquetes funcionales, y organizando estos en función del tipo de recurso. En la Figura 3-27 se muestra la estructura del servicio web mediante un diagrama de paquetes.

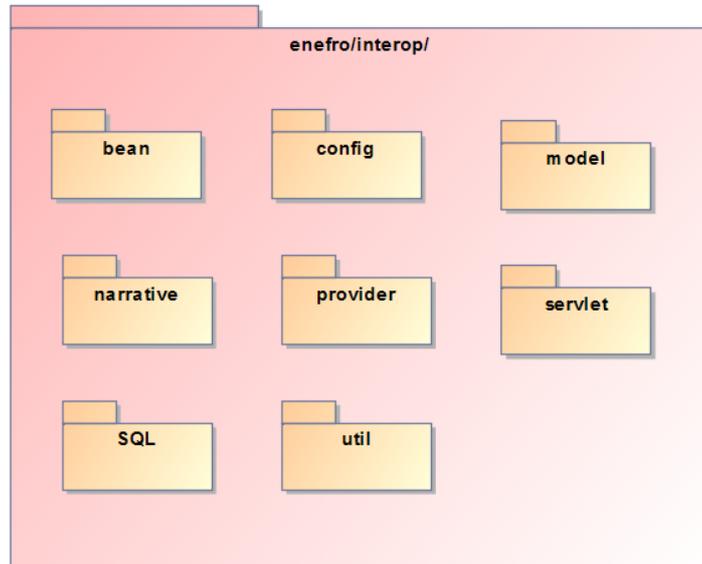


Figura 3-27. Diagrama de paquetes de la estructura del servicio web

- **Paquete bean:** contiene las clases de los objetos de información de e-Nefro.
- **Paquete config:** contiene la clase de configuración de la web de prueba que suministra la propia API, que se verá en un apartado posterior.
- **Paquete model:** contiene las clases donde se definen los recursos extendidos de nuestro sistema.
- **Paquete narrative:** contiene los archivos necesarios para generar los elementos narrativos de los recursos.
- **Paquete provider:** contiene las clases proveedoras de recursos, donde se crean y suministran a los clientes.
- **Paquete servlet:** contiene la clase en la que se define el servicio web, y se establecen sus proveedores de recursos y parámetros.
- **Paquete SQL:** contiene las clases para la conexión y consulta a la base de datos de e-Nefro.
- **Paquete util:** contiene clases para la creación de recursos que no se suministran al cliente directamente, sino que van contenidos en otros que sí.

Todos estos elementos trabajan conjuntamente en la recepción de la petición de información, la obtención de la información solicitada, y la formación y suministro del recurso correspondiente, como se verá en el siguiente apartado.

3.5.2. Desarrollo del servicio web

Una vez visto cómo funciona y cómo se estructura el servicio web, podemos pasar a ver cómo se han desarrollado los principales elementos que lo componen, y de qué forma trabajan conjuntamente para ofrecer la información al cliente.

En esta primera implementación, se ha incluido todo lo relativo a los recursos Patient, Practitioner, Appointment, Encounter y Organization, en los que se hace uso de todo lo previamente explicado sobre los recursos FHIR: extensiones, recursos contenidos y elementos narrativos.

Para poder trabajar con la especificación HL7 FHIR, es necesario incluir en el pom.xml las dependencias de las librerías HAPI FHIR:

```

<!-- This dependency includes the core HAPI-FHIR classes -->
<dependency>
  <groupId>ca.uhn.hapi.fhir</groupId>
  <artifactId>hapi-fhir-base</artifactId>
  <version>1.5-SNAPSHOT</version>
</dependency>
<!-- At least one "structures" JAR must also be included -->
<dependency>
  <groupId>ca.uhn.hapi.fhir</groupId>
  <artifactId>hapi-fhir-structures-dstu2</artifactId>
  <version>1.5-SNAPSHOT</version>
</dependency>

<!-- This dependency is used for the "FHIR Tester" web app overlay -->
<dependency>
  <groupId>ca.uhn.hapi.fhir</groupId>
  <artifactId>hapi-fhir-testpage-overlay</artifactId>
  <version>1.5-SNAPSHOT</version>
  <type>war</type>
  <scope>provided</scope>
</dependency>
<dependency>
  <groupId>ca.uhn.hapi.fhir</groupId>
  <artifactId>hapi-fhir-testpage-overlay</artifactId>
  <version>1.5-SNAPSHOT</version>
  <classifier>classes</classifier>
  <scope>provided</scope>
</dependency>

```

El resto de elementos se explicarán agrupados por el paquete funcional en el que están contenidos.

3.5.2.1. Paquete servlet

Contiene la clase **ExampleRestfulServlet** de donde se define el servicio web (Figura 3-28).



Figura 3-28- Paquete servlet

En primer lugar, en el constructor, se establece la versión de FHIR que se implementa, siendo la actual versión oficial, DSTU 2:

```

/**
 * Constructor
 */
public ExampleRestfulServlet() {
    super(FhirContext.forDstu2()); // Support DSTU2
}

```

A continuación se definen qué proveedores de recursos se van a implementar en el servicio. En esta primera implementación tendremos cuatro, para los recursos Patient, Practitioner, Appointment y Encounter:

```

/**
 * Three resource providers are defined. Each one handles a specific
 * type of resource.
 */
List<IResourceProvider> providers = new ArrayList<IResourceProvider>();
providers.add(new PatientResourceProvider());
providers.add(new PractitionerResourceProvider());
providers.add(new AppointmentResourceProvider());
providers.add(new EncounterResourceProvider());
setResourceProviders(providers);

```

También se declaran los generadores de elementos narrativos que implementará nuestro servicio. En nuestro caso, se implementan el generador narrativo propio que provee la API, en la que únicamente se genera para el recurso Patient, y un generador propio, en el que hemos definido el elemento narrativo del recurso Practitioner:

```
/*
 * Use a narrative generator. This is a completely optional step,
 * but can be useful as it causes HAPI to generate narratives for
 * resources which don't otherwise have one.
 */
//Generador narrativo por defecto
INarrativeGenerator narrativeGen = new DefaultThymeleafNarrativeGenerator();
getFhirContext().setNarrativeGenerator(narrativeGen);
//Generador narrativo propio
String propFile = "file:C:/Users/migpedjim/Desktop/hapi-fhir-master/restful-server-
example/src/main/java/enefro/interop/narrative/narratives.properties";
CustomThymeleafNarrativeGenerator myNarrativeGen = new CustomThymeleafNarrativeGenerator(propFile);
getFhirContext().setNarrativeGenerator(myNarrativeGen);
```

Por último, se establecen una serie de parámetros para que sean más cómodas de leer las respuestas del servidor:

```
/*
 * This server interceptor causes the server to return nicely
 * formatted and coloured responses instead of plain JSON/XML if
 * the request is coming from a browser window. It is optional,
 * but can be nice for testing.
 */
registerInterceptor(new ResponseHighlighterInterceptor());

/*
 * Tells the server to return pretty-printed responses by default
 */
setDefaultPrettyPrint(true);
```

3.5.2.2. Paquete narrative

Este paquete contiene los archivos necesarios para la implementación del generador narrativo propio (Figura 3-29).

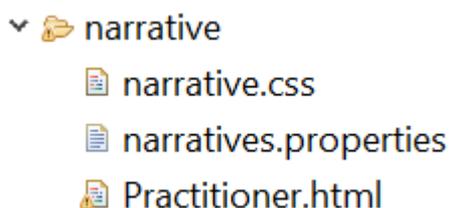


Figura 3-29. Paquete narrative

En **narrative.properties** se establecen las propiedades del generador. Concretamente, para qué recursos va a generar su elemento narrativo, y dónde se encuentran estos:

```
#Recurso Practitioner
practitioner.class=ca.uhn.fhir.model.dstu2.resource.Practitioner
practitioner.narrative=file:C:/Users/migpedjim/Desktop/hapi-fhir-master/restful-server-
example/src/main/java/enefro/interop/narrative/Practitioner.html
```

En **Practitioner.html** se define el elemento narrativo que se va a generar. En primer lugar, se implementa la hoja de estilos **narrative.css**, donde se define el estilo:

```
<head>
  <link rel="stylesheet" type="text/css" href="narrative.css"/>
</head>
```

Por último, se definen los elementos que lo componen, indicando qué atributos del recurso contiene y su estructura HTML:

```
<div>
  <th:block th:if="{not resource.name.empty}">
    <div class="hapiHeaderText" th:narrative="{resource.name}" />
  </th:block>
  <table class="hapiPropertyTable">
    <tbody>
      <tr th:if="{not resource.identifier.empty}">
        <td>Identificier</td>
        <td th:narrative="{resource.identifier[0]}"></td>
      </tr>
      <tr th:if="{not resource.telecom.empty}">
        <td>Teléfono</td>
        <td th:narrative="{resource.telecom[0].valueElement}"></td>
      </tr>
      <tr th:if="{not resource.telecom.empty}">
        <td>Email</td>
        <td th:narrative="{resource.telecom[1].valueElement}"></td>
      </tr>
      <tr th:if="{not resource.practitionerRole.empty}">
        <td>Cargo</td>
        <td th:narrative="{resource.practitionerRole[0].role}"></td>
      </tr>
      <tr th:if="{not resource.practitionerRole.empty}">
        <td>Especialidad</td>
        <td th:narrative="{resource.practitionerRole[0].specialty[0]}"></td>
      </tr>
      <tr th:if="{not resource.practitionerRole.empty}">
        <td>Organización</td>
        <td th:narrative="{resource.practitionerRole[0].managingOrganization.resource.nameElement}"></td>
      </tr>
    </tbody>
  </table>
</div>
```

3.5.2.3. Paquete bean

Este paquete contiene las clases de los objetos propios definidos en el sistema e-Nefro, a los que se mapea la información extraída de la base de datos (Figura 3-30).

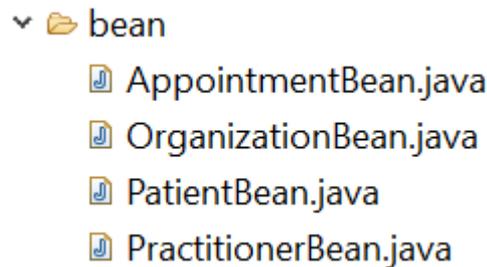


Figura 3-30. Paquete bean

Esta clase contiene los atributos definidos para el objeto al que representa. Por ejemplo, en el caso de la clase **AppointmentBean**:

```
private int idCita;
private int idPlan;
private int idPaciente;
private int idProfesional;
private String localizacion;
private Date fechaInicio;
private String causa;
private String descripcion;
private String obserMed;
private String obserEnf;
private String comenPac;
```

Y una serie de métodos para obtener dicha información:

```
//Metodo para obtener el id de la cita
public int getIdCita() {
    return idCita;
}
//Metodo para obtener el id interno del paciente implicado
public int getIdPaciente() {
    return idPaciente;
}
```

3.5.2.4. Paquete SQL

Este paquete contiene las clases con los métodos necesarios para realizar las consultas a la base de datos de eNefro (Figura 3-31).



Figura 3-31. Paquete SQL

En primer lugar, se establece el manejador de la base de datos, en este caso implementada con tecnología postgresSQL:

```
//Manejador de la base de datos
Class.forName("org.postgresql.Driver");
```

Antes de realizar la consulta, es necesario conectarse a la base de datos, pasando para ello un usuario con los permisos necesarios, su contraseña y la URL del servidor de base de datos:

```
//Parametros y conexión a la bbdd
String usuario = "enefro_owner";
String password = "enefro_owner";
String url = "jdbc:postgresql://gibserv.us.es:4502/enefro";
Connection conexion = DriverManager.getConnection(url, usuario, password);
```

Una vez conectados, definimos la consulta y la ejecutamos. Esta sería una consulta para obtener la información de las citas y encuentros de un profesional, dentro de la clase **AppointmentSQL.java**:

```
//Consulta a base de datos
String sql = "SELECT * FROM "
+ "(((e_appointmentactivity left join e_appointmentregistry on appo_id=appr_appointmentacti) "
+ "INNER JOIN e_activity ON appo_base_acti = acti_id) INNER JOIN e_plan ON appo_id_plan=plan_id) "
+ "LEFT JOIN e_clinicalregistry ON appr_base_crir=crir_id) WHERE appo_citadocon = (SELECT usua_id "
+ "FROM e_usuario INNER JOIN e_person ON usua_persona=pers_id WHERE pers_document = '"+idPrac+"')";
Statement st = conexion.createStatement();
ResultSet rs = st.executeQuery(sql);
```

Y pasamos a recorrer el resultado mapeando la información de cada elemento obtenido en el objeto propio correspondiente. Estos objetos se irán agrupando en una lista que será devuelta por el método:

```

//Recorremos el resultado de la consulta
while (rs.next())
{
    AppointmentBean appointment = new AppointmentBean();
    appointment = new AppointmentBean();
    appointment.setIdPlan(rs.getInt("appo_id_plan"));
    appointment.setIdPaciente(rs.getInt("plan_pacienteusuario"));
    appointment.setIdProfesional(rs.getInt("appo_citadocon"));
    appointment.setLocalizacion(rs.getString("acti_location"));
    appointment.setFechaInicio(rs.getTimestamp("acti_startdate"));
    appointment.setCausa(rs.getString("acti_problemaa"));
    appointment.setDescripcion(rs.getString("acti_description"));
    appointment.setObserMed(rs.getString("appo_observacionpriv"));
    appointment.setObserEnf(rs.getString("appo_observacionenfe"));
    appointment.setComenPac(rs.getString("crir_comentarios"));
    listAppointments.add(appointment);
}
rs.close();
st.close();
conexion.close();
}
catch (Exception e){
    e.printStackTrace();
}
return listAppointments;
}

```

3.5.2.5. Paquete model

Este paquete contiene las clases donde se definen los recursos extendidos (Figura 3-32).

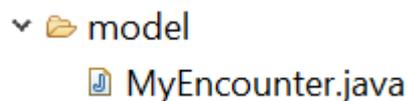


Figura 3-32. Paquete model

En nuestro caso, hemos extendido el recurso Encounter, mediante la clase **MyEncounter**, para que contenga la información relativa a las observaciones médicas, observaciones de enfermería y comentarios para el paciente realizados durante el mismo.

Esta clase es necesario extenderla con el recurso al que está extendiendo, para que cuando se utilice posteriormente, además de los recursos extendidos, contenga los propios:

```

@ResourceDef(name = "Encounter")
public class MyEncounter extends Encounter { ... }

```

A continuación, se van definiendo y creando las extensiones, indicando el nombre, su descripción y algunos parámetros: **url** (dirección que nos devuelve el recurso StructureDefinition con la información de la estructura de la extensión), **isModifier** (si el atributo modifica el significado del recurso que lo contiene) y **definedLocally** (si la extensión está definida localmente):

```

/**
 * This is a basic extension, with a DataType value (in this case, StringDt)
 */

//Atributo extendido Observaciones medicas
@Child(name = "obsMedico")
@Description(shortDefinition = "Contiene las observaciones hechas por el profesional médico")
@Extension(url = "localhost:8080/extensión/encounter/obsmed", isModifier = false, definedLocally = true)
private StringDt myObsMedico;

```

Y para poder trabajar con estos atributos extendidos del recurso, es necesario crear los métodos *Get* y *Set* correspondientes:

```

//Obtenemos las observaciones medicas del encuentro
public StringDt getObsMedico() {
    return myObsMedico;
}
//Establecemos las observaciones medicas del encuentro
public void setObsMedico(StringDt obsMedico) {
    this.myObsMedico = obsMedico;
}

```

3.5.2.6. Paquete provider

Este paquete contiene los proveedores de recursos, es decir, las clases en las que se forman y suministran los recursos al cliente (Figura 3.33).



Figura 3-33. Paquete provider

Estas clases tienen una gran importancia en el servicio web, y son de especial complejidad, pues hacen uso de todos los elementos explicados en la estructura del servicio. Para explicar su funcionamiento, se va a ver el proveedor **EncounterResourceProvider**, que hace uso de recursos contenidos y extensiones.

Esta clase implementa la interfaz **IResourceProvider**, necesaria en todos los proveedores de recursos. Además, es necesario definir qué tipo de recurso suministra, en nuestro caso **MyEncounter**, que extiende el recurso **Encounter** definido por FHIR:

```

public class EncounterResourceProvider implements IResourceProvider{

    @Override
    public Class<? extends IBaseResource> getResourceType() {
        return MyEncounter.class;
    }
}

```

Tras esto, se definen los métodos de acceso al recurso, indicando qué tipo de operación es, a través de un sistema de etiquetas que implementa la API, en nuestro caso *Search*. Este método en particular devuelve todos los encuentros en los que ha participado un profesional en una lista de recursos **MyEncounter**, pasándole como parámetro el DNI del profesional:

```

@Search()
public List<MyEncounter> findEncounterByPractitioner (@RequiredParam (name=Encounter.SP_PRACTITIONER)
StringParam idPractitioner){
    ...
    return listaEncounter;
}

```

Se va a describir cómo funcionan nuestros proveedores de recursos con este método como ejemplo.

En primer lugar, se recupera la información de la base de datos, mediante el método correspondiente de la clase **AppointmentSQL**, y se mapea a una lista de **AppointmentBean**, que recorreremos con un iterador:

```

//Recuperamos los encuentros de base de datos
LinkedList<AppointmentBean> listaAppointmentsBean =
AppointmentSQL.getAppointmentsByPractitioner(idPractitioner.getValue());
Iterator<AppointmentBean> it = listaAppointmentsBean.iterator();

```

Es necesario crear la lista de recursos MyEncounter que se suministrará al cliente, en la que iremos incluyendo los recursos formados:

```
//Creamos la lista de recursos MyEncounter
List<MyEncounter> listaEncounter = new ArrayList<MyEncounter>();
```

Dentro de un bucle *While*, vamos recorriendo la lista de objetos AppointmentBean y formando los recursos con su información:

```
//Recorremos la lista de PacientesBean para ir formando los recursos FHIR
while (it.hasNext()) {
AppointmentBean AppointmentActual = (AppointmentBean) it.next();
MyEncounter encounter = new MyEncounter();
... }
}
```

Dentro del bucle, se van añadiendo los atributos del recurso con la información correspondiente. Por ejemplo, para añadir el *Id* del recurso, campo obligatorio:

```
//Identificador del recurso
encounter.setId(Long.toString(AppointmentActual.getIdCita()));
```

Para añadir un recurso contenido, como, por ejemplo, el paciente que participa en el encuentro, utilizamos la clase **PatientUtil**, con la que podemos obtener un recurso Patient pasándole el identificador interno almacenado en el campo idPaciente del objeto Encuentro.

Aunque todo el mecanismo de formación del recurso Patient (consulta a base de datos y mapeo al recurso FHIR) se podría hacer sin utilizar esta clase externa, es más eficiente la creación de estas clases *útiles* y utilizarlas directamente en todos los recursos donde se necesite. Una vez obtenido el recurso, se referencia a través del atributo *patient* del recurso Encounter:

```
//Creamos un recurso Patient y lo incluimos
Patient patient = PatientUtil.findPatientById(AppointmentActual.getIdPaciente());
if(patient!=null){
    encounter.getPatient().setResource(patient);
}
}
```

Los atributos extendidos se añaden del mismo modo que los propios, ya que hemos definido el recurso extendido **MyAppointment** que los implementa:

```
//Observaciones medicas
StringDt ObsMedico = new StringDt();
ObsMedico.setValueAsString(AppointmentActual.getObserMed());
encounter.setObsMedico(ObsMedico);
//Observaciones de enfermeria
StringDt ObsEnf = new StringDt();
ObsEnf.setValueAsString(AppointmentActual.getObserEnf());
encounter.setObsEnferm(ObsEnf);
//Comentarios para el paciente
StringDt ComPac = new StringDt();
ComPac.setValueAsString(AppointmentActual.getComenPac());
encounter.setComPaciente(ComPac);
```

Una vez formada la lista de recursos Encounter, tendríamos toda la información solicitada por el cliente.

3.5.3. Validación del servicio web

Una vez visto cómo ha sido desarrollado el servicio web, se van a mostrar las pruebas de funcionamiento más relevantes realizadas sobre él.

Se han realizado dos bloques de pruebas, el primero desde el módulo web que suministra la propia API para el testeo del servicio, y el segundo desde un cliente FHIR Android que se ha implementado específicamente para ello.

Aunque se van a hacer uso de estos clientes para que sean más entendibles los resultados obtenidos, también es posible realizar peticiones directamente al servidor desde la dirección base **http://localhost:8080/restful-server-example/fhir**, establecida en el archivo web.xml:

```
<servlet-mapping>
  <servlet-name>fhirServlet</servlet-name>
  <url-pattern>/fhir/*</url-pattern>
</servlet-mapping>
```

3.5.3.1. Módulo web de testeo

La API implementa un módulo web de testeo, con el que es posible hacer uso, de manera sencilla, de los proveedores de recursos que soporta nuestro servicio web. Se configura desde la clase **FhirTesterConfig** dentro del paquete config (Figura 3-34).

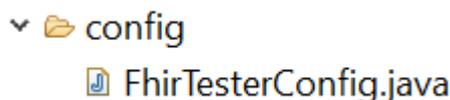


Figura 3-34. Paquete config

Indicando qué servidor se va a testear con el módulo:

```
TesterConfig retVal = new TesterConfig();
    retVal
        .addServer()
        .withId("home")
        .withFhirVersion(FhirVersionEnum.DSTU2)
        .withBaseUrl("${serverBase}/fhir")
        .withName("Local Tester")
```

Se accede a ella a través de la dirección **http://localhost:8080/restful-server-example/tester/**, tal como se establece en el archivo web.xml:

```
<servlet-mapping>
  <servlet-name>spring</servlet-name>
  <url-pattern>/tester/*</url-pattern>
</servlet-mapping>
```

El módulo provee de una interfaz agradable para realizar las pruebas sobre nuestro servicio. En la página principal encontramos (Figura 3-35):

1. Información sobre nuestro servicio web: nombre, versión que implementa y dirección base.
2. Las opciones del servicio, donde se puede establecer si queremos la respuesta como XML o JSON (*Encoding*) o si queremos que salgan de manera agradable visualmente (*pretty*).
3. La lista de recursos que provee nuestro servicio, desde la que seleccionar el proveedor que queremos probar.

The screenshot shows the main interface of the HAPI-FHIR RESTful server tester. On the left, there is a navigation sidebar with three main sections: 'Options' (containing Encoding, Pretty, and Summary settings), 'Resources' (a list of FHIR resource types like Appointment, Encounter, Patient, etc.), and 'Server' (Server Home/Actions). The main content area has a header with the HAPI-FHIR logo and tagline 'fhir made simple.'. Below the header, there is a description: 'This is a RESTful server tester, which can be used to send requests to, and receive responses from the server.' A table (1) displays server information: Server (HAPI FHIR Server), Software (HAPI FHIR Server - 1.5-SNAPSHOT), and FHIR Base (http://localhost:8080/restful-server-example/fhir). Below the table, there are 'Server Actions' including 'Retrieve the server's conformance statement' (with a 'Conformance' button), 'Retrieve the update history across all resource types on the server' (with a 'History' button, 'Since' field, and 'Limit # (opt)' field), and 'Post a bundle containing multiple resources to the server and store all resources within a single atomic transaction' (with a 'Transaction' button and a 'Bundle *' field).

Figura 3-35. Módulo de validación: Página principal

Si seleccionamos uno de los recursos desde la lista, por ejemplo, Patient, nos dirige a la página de testeo de dicho proveedor, en la que podemos ver (Figura 3-36):

1. Información sobre qué recurso se provee.
2. Las operaciones implementadas en el proveedor. En nuestro caso, sólo la operación *Search*.
3. Los parámetros de búsqueda necesario para que el método del proveedor pueda realizar la búsqueda de información. En función de qué parámetros se establezcan, estaremos haciendo uso de un método u otro.

The screenshot shows the 'Patient' resource page in the HAPI-FHIR RESTful server tester. The left sidebar has 'Patient' selected under the 'Resources' section. The main content area has a header with the HAPI-FHIR logo and tagline 'fhir made simple.'. Below the header, there is a description: 'This is a RESTful server tester, which can be used to send requests to, and receive responses from the server at the following URL:'. A table (1) displays resource information: Resource: Patient. Below the table, there is a 'Search' button (2) and a 'Search Parameters' section (3) with a dropdown menu set to 'identifier - A patient identifier' and a 'Matches' field. Below the search parameters, there are 'Sort Results' options (Sort By: Default, Direction: Default) and 'Other Options'.

Figura 3-36. Módulo de validación: Página de recurso

Tras establecer los parámetros y pulsar en el botón Search, se realiza la consulta y obtenemos los recursos solicitados en la siguiente ventana, donde se incluye (Figura 3-37):

1. El tiempo que ha demorado la consulta.
2. El código que habría que implementar en el cliente para realizar la consulta.
3. La petición realizada al servidor.
4. La respuesta obtenida del servidor.
5. El resultado obtenido, en nuestro caso, en XML.

The screenshot shows the Swagger UI interface for a FHIR RESTful server. The main content area displays the results of a GET request. The interface is annotated with red boxes and numbers 1 through 5, corresponding to the list items above. Box 1 highlights the execution time: 'Executed request against FHIR RESTful server in 128ms'. Box 2 highlights the 'Client Code' section, which contains a code snippet for executing the request in a client. Box 3 highlights the 'Request' section, showing the request method (GET) and the URL ('http://localhost:8080/restful-server-example/fhir/Patient?identifier=625218621'). Box 4 highlights the 'Response' section, showing the response status (HTTP/1.1 200 OK) and the response headers. Box 5 highlights the 'Result Body XML' section, which displays the raw XML message of the response, showing a bundle containing one Patient resource.

Figura 3-37. Módulo de validación: Página de resultado

3.5.3.2. Pruebas desde el módulo web de testeo

Debido a que algunas respuestas son muy extensas al incluir muchos recursos, se mostrará la respuesta completa sólo en algunos casos, y se obviará si es redundante.

- **Acceso a la información del paciente con NUH 625218621**

Se provee un recurso Patient cuyo NUH coincide con el que se le pasa como parámetro, en ese caso 625218621.

La petición se realiza mediante un GET, en el que viaja el parámetro de búsqueda que identifica al paciente. Responde con un HTTP 200 OK, indicando que no ha habido errores durante el proceso (Figura 3-38).

Request	GET http://localhost:8080/restful-server-example/fhir/Patient?identifier=625218621
Request Headers	Accept-Charset: utf-8 Accept: application/xml+fhir;q=1.0, application/json+fhir;q=1.0 User-Agent: HAPI-FHIR/1.5-SNAPSHOT (FHIR Client; FHIR DSTU2; apache) Accept-Encoding: gzip
Response	✓ HTTP/1.1 200 OK
Response Headers	Server: Apache-Coyote/1.1 X-Powered-By: HAPI FHIR 1.5-SNAPSHOT RESTful Server Last-Modified: Sun, 26 Jun 2016 16:31:00 GMT Content-Type: application/xml+fhir; charset=UTF-8 Transfer-Encoding: chunked Date: Sun, 26 Jun 2016 16:31:00 GMT
Result Body	☐ Bundle contains 1 / 1 entries

Figura 3-38. Petición/respuesta consulta de un paciente

El *bundle* contiene sólo una entrada, la correspondiente al recurso del paciente solicitado:

```
<Bundle xmlns="http://hl7.org/fhir">
  <id value="0e2ddd06-47ed-4318-a629-66ff7774b769"/>
  <meta>
    <lastUpdated value="2016-06-26T12:20:41.704+02:00"/>
  </meta>
  <type value="searchset"/>
  <total value="1"/>
  <link>
    <relation value="self"/>
    <url value="http://localhost:8080/restful-server-example/fhir/Patient?identifier=625218621"/>
  </link>
  <entry>
    <fullUrl value="http://localhost:8080/restful-server-example/fhir/Patient/5"/>
    <resource>
      <Patient xmlns="http://hl7.org/fhir">
        <id value="5"/>
        <text>
          <status value="generated"/>
          <div xmlns="http://www.w3.org/1999/xhtml">
            <div class="hapiHeaderText"> Lucía
              <b>PAZ MARTÍNEZ </b>
            </div>
            <table class="hapiPropertyTable">
              <tbody>
                <tr>
                  <td>Identifier</td>
                  <td>625218621</td>
                </tr>
                <tr>
                  <td>Address</td>
                  <td>
                    <span>Calle Álava 34 </span>
                    <br/>
                  </td>
                </tr>
                <tr>
                  <td>Date of birth</td>
                  <td>
                    <span>25 abril 1973</span>
                  </td>
                </tr>
              </tbody>
            </table>
          </div>
        </text>
        <identifier>
          <type>
            <coding>
              <system value="http://hl7.org/fhir/v2/0203"/>
              <code value="MR"/>
            </coding>
          </type>
          <value value="625218621"/>
        </identifier>
      </Patient>
    </resource>
  </entry>
</Bundle>
```

```

<identifier>
  <type>
    <text value="3"/>
  </type>
  <value value="87743365K"/>
</identifier>
<active value="true"/>
<name>
  <family value="Paz Martínez"/>
  <given value="Lucía"/>
</name>
<telecom>
  <system value="phone"/>
  <value value="698336655"/>
</telecom>
<telecom>
  <system value="email"/>
  <value value="lucia@us.es"/>
</telecom>
<gender value="female"/>
<birthDate value="1973-04-25"/>
<address>
  <line value="Calle Álava 34"/>
</address>
</Patient>
</resource>
</entry>
</Bundle>

```

En la Figura 3-39 vemos el elemento narrativo que se genera junto a este recurso:

Lucía PAZ MARTÍNEZ	
Identifier	625218621
Address	Calle Álava 34
Date of birth	25 abril 1973

Figura 3-39. Elemento narrativo de recurso Patient

- **Acceso a la información de un paciente inexistente**

Se provee un recurso Patient cuyo NUH coincide con el que se le pasa como parámetro, en este caso inexistente en la base de datos.

La petición se realiza mediante un GET, en el que viaja el parámetro de búsqueda que identifica al paciente, que para la prueba se ha establecido como 'PacienteInexistente'. Responde con un HTTP 200 OK, indicando que no ha habido errores durante el proceso (Figura 3-40).

➤ Request	GET http://localhost:8080/restful-server-example/fhir/Patient?identifier=PacienteInexistente
Request Headers	Accept-Charset: utf-8 Accept: application/xml+fhir;q=1.0, application/json+fhir;q=1.0 User-Agent: HAPI-FHIR/1.5-SNAPSHOT (FHIR Client; FHIR DSTU2; apache) Accept-Encoding: gzip
◀ Response	✔ HTTP/1.1 200 OK
Response Headers	Server: Apache-Coyote/1.1 X-Powered-By: HAPI FHIR 1.5-SNAPSHOT RESTful Server Last-Modified: Sun, 26 Jun 2016 16:32:11 GMT Content-Type: application/xml+fhir; charset=UTF-8 Transfer-Encoding: chunked Date: Sun, 26 Jun 2016 16:32:11 GMT
Result Body	Bundle contains no entries

Figura 3-40. Petición/Respuesta consulta de un paciente inexistente.

El *bundle* no contiene ninguna entrada, pues no existe ningún paciente con ese NUH:

```
<Bundle xmlns="http://hl7.org/fhir">
  <id value="c63a3718-f572-4d23-8711-42af26dd5dfd"/>
  <meta>
    <lastUpdated value="2016-06-26T12:41:42.564+02:00"/>
  </meta>
  <type value="searchset"/>
  <total value="0"/>
  <link>
    <relation value="self"/>
    <url value="http://localhost:8080/restful-server-
example/fhir/Patient?identifier=PacienteInexistente"/>
  </link>
</Bundle>
```

- **Acceso a la información de los pacientes a cargo del profesional con DNI 57722647F**

Se provee los recursos Patient de los pacientes que están a cargo del profesional cuyo DNI coincide con el que se le pasa como parámetro, en ese caso inexistente en este caso 57722647F.

La petición se realiza mediante un GET, en el que viaja el parámetro de búsqueda que identifica al profesional. Responde con un HTTP 200 OK, indicando que no ha habido errores durante el proceso (Figura 3-41).

➤ Request	GET http://localhost:8080/restful-server-example/fhir/Patient?careprovider=57722647F
Request Headers	Accept-Charset: utf-8 Accept: application/xml+fhir;q=1.0, application/json+fhir;q=1.0 User-Agent: HAPI-FHIR/1.5-SNAPSHOT (FHIR Client; FHIR DSTU2; apache) Accept-Encoding: gzip
◀ Response	✔ HTTP/1.1 200 OK
Response Headers	Server: Apache-Coyote/1.1 X-Powered-By: HAPI FHIR 1.5-SNAPSHOT RESTful Server Last-Modified: Sun, 26 Jun 2016 16:29:27 GMT Content-Type: application/xml+fhir; charset=UTF-8 Transfer-Encoding: chunked Date: Sun, 26 Jun 2016 16:29:27 GMT
Result Body	☐ Bundle contains 3 / 3 entries

Figura 3-41. Petición/Respuesta consulta de pacientes a cargo de un profesional

El *bundle* contiene tres entradas, correspondientes a tres pacientes que tiene este profesional bajo su cargo. Por extensión y redundancia no se incluye.

- **Acceso a la información del profesional con DNI 57722647F**

Se provee un recurso Practitioner cuyo DNI coincide con el que se le pasa como parámetro, en ese caso 57722647F.

La petición se realiza mediante un GET, en el que viaja el parámetro de búsqueda que identifica al paciente. Responde con un HTTP 200 OK, indicando que no ha habido errores durante el proceso (Figura 3-42).

➤ Request	GET http://localhost:8080/restful-server-example/fhir/Practitioner?identifier=57722647F
Request Headers	Accept-Charset: utf-8 Accept: application/xml+fhir;q=1.0, application/json+fhir;q=1.0 User-Agent: HAPI-FHIR/1.5-SNAPSHOT (FHIR Client; FHIR DSTU2; apache) Accept-Encoding: gzip
◀ Response	✓ HTTP/1.1 200 OK
Response Headers	Server: Apache-Coyote/1.1 X-Powered-By: HAPI FHIR 1.5-SNAPSHOT RESTful Server Last-Modified: Sun, 26 Jun 2016 16:33:46 GMT Content-Type: application/xml+fhir; charset=UTF-8 Transfer-Encoding: chunked Date: Sun, 26 Jun 2016 16:33:46 GMT
Result Body	📦 Bundle contains 1 / 1 entries

Figura 3-42. Petición/Respuesta consulta de un profesional

El *bundle* contiene sólo una entrada, la correspondiente al recurso del profesional solicitado:

```
<Bundle xmlns="http://hl7.org/fhir">
  <id value="d670f660-0683-478e-8887-f6629aa2d2d3"/>
  <meta>
    <lastUpdated value="2016-06-26T12:51:09.573+02:00"/>
  </meta>
  <type value="searchset"/>
  <total value="1"/>
  <link>
    <relation value="self"/>
    <url value="http://localhost:8080/restful-server-example/fhir/Practitioner?identifier=57722647F"/>
  </link>
  <entry>
    <fullUrl value="http://localhost:8080/restful-server-example/fhir/Practitioner/4"/>
    <resource>
      <Practitioner xmlns="http://hl7.org/fhir">
        <id value="4"/>
        <text>
          <status value="generated"/>
          <div xmlns="http://www.w3.org/1999/xhtml">
            <div class="hapiHeaderText"> Alberto
              <b>NÚÑEZ GARCÍA </b>
            </div>
            <table class="hapiPropertyTable">
              <tbody>
                <tr>
                  <td>Identificador</td>
                  <td>57722647F</td>
                </tr>
                <tr>
                  <td>Teléfono</td>
                  <td>698454545</td>
                </tr>
                <tr>
                  <td>Email</td>
                  <td>alberto@us.es</td>
                </tr>
                <tr>
                  <td>Cargo</td>

```

```

        <td>MEDICO</td>
    </tr>
    <tr>
        <td>Especialidad</td>
        <td>Especialista Nefrología</td>
    </tr>
    <tr>
        <td>Organización</td>
        <td>Hospital de prueba</td>
    </tr>
</tbody>
</table>
</div>
</text>
<contained>
    <Organization xmlns="http://hl7.org/fhir">
        <id value="1"/>
        <name value="Hospital de prueba"/>
        <telecom>
            <system value="phone"/>
            <value value="656895262"/>
        </telecom>
        <telecom>
            <system value="email"/>
            <value value="hospital@us.es"/>
        </telecom>
        <address>
            <line value="41092"/>
        </address>
    </Organization>
</contained>
<identifier>
    <type>
        <text value="3"/>
    </type>
    <value value="57722647F"/>
</identifier>
<active value="true"/>
<name>
    <family value="Núñez García"/>
    <given value="Alberto"/>
</name>
<telecom>
    <system value="phone"/>
    <value value="698454545"/>
</telecom>
<telecom>
    <system value="email"/>
    <value value="alberto@us.es"/>
</telecom>
<gender value="male"/>
<birthDate value="1971-04-09"/>
<practitionerRole>
    <managingOrganization>
        <reference value="#1"/>
    </managingOrganization>
    <role>
        <text value="MEDICO"/>
    </role>
    <specialty>
        <text value="Especialista Nefrología"/>
    </specialty>
</practitionerRole>
</Practitioner>
</resource>
</entry>
</Bundle>

```

En la Figura 3-43 vemos el elemento narrativo que se genera junto a este recurso.

Alberto NÚÑEZ GARCÍA	
Identifier	57722647F
Teléfono	698454545
Email	alberto@us.es
Cargo	MEDICO
Especialidad	Especialista Nefrología
Organización	Hospital de prueba

Figura 3-43. Elemento narrativo de recurso Practitioner

- **Acceso a la información de un profesional inexistente**

Se provee un recurso Practitioner cuyo DNI coincide con el que se le pasa como parámetro, en ese caso inexistente en la base de datos.

La petición se realiza mediante un GET, en el que viaja el parámetro de búsqueda que identifica al paciente, que para la prueba se ha establecido como 'ProfesionalInexistente'. Responde con un HTTP 200 OK, indicando que no ha habido errores durante el proceso (Figura 3-44).

Request	GET http://localhost:8080/restful-server-example/fhir/Practitioner?identifier=ProfesionalInexistente
Request Headers	Accept-Charset: utf-8 Accept: application/xml+fhir;q=1.0, application/json+fhir;q=1.0 User-Agent: HAPI-FHIR/1.5-SNAPSHOT (FHIR Client; FHIR DSTU2; apache) Accept-Encoding: gzip
Response	✓ HTTP/1.1 200 OK
Response Headers	Server: Apache-Coyote/1.1 X-Powered-By: HAPI FHIR 1.5-SNAPSHOT RESTful Server Last-Modified: Tue, 28 Jun 2016 13:07:08 GMT Content-Type: application/xml+fhir;charset=UTF-8 Transfer-Encoding: chunked Date: Tue, 28 Jun 2016 13:07:07 GMT
Result Body	Bundle contains no entries

Figura 3-44. Petición/Respuesta consulta de un profesional inexistente

El *bundle* no contiene ninguna entrada, pues no existe ningún profesional con ese DNI:

```
<Bundle xmlns="http://hl7.org/fhir">
  <id value="bc0844b2-b286-46e9-9471-6948462f04b3"/>
  <meta>
    <lastUpdated value="2016-06-28T15:07:08.218+02:00"/>
  </meta>
  <type value="searchset"/>
  <total value="0"/>
  <link>
    <relation value="self"/>
    <url value="http://localhost:8080/restful-server-example/fhir/Practitioner?identifier=ProfesionalInexistente"/>
  </link>
</Bundle>
```

- **Acceso a la información de todas las citas del profesional con DNI 57722647F**

Se proveen los recursos Appointment de las citas del profesional cuyo DNI coincide con el que se le pasa como parámetro, en este caso 57722647F.

La petición se realiza mediante un GET, en el que viaja el parámetro de búsqueda que identifica al profesional. Responde con un HTTP 200 OK, indicando que no ha habido errores durante el proceso (Figura 3-45).

➤ Request	GET http://localhost:8080/restful-server-example/fhir/Appointment?practitioner=57722647F
Request Headers	Accept-Charset: utf-8 Accept: application/xml+fhir;q=1.0, application/json+fhir;q=1.0 User-Agent: HAPI-FHIR/1.5-SNAPSHOT (FHIR Client; FHIR DSTU2; apache) Accept-Encoding: gzip
◀ Response	✔ HTTP/1.1 200 OK
Response Headers	Server: Apache-Coyote/1.1 X-Powered-By: HAPI FHIR 1.5-SNAPSHOT RESTful Server Last-Modified: Sun, 26 Jun 2016 16:35:25 GMT Content-Type: application/xml+fhir; charset=UTF-8 Transfer-Encoding: chunked Date: Sun, 26 Jun 2016 16:35:25 GMT
Result Body	📦 Bundle contains 14 / 14 entries

Figura 3-45. Petición/Respuesta consulta de las citas de un profesional

El *bundle* contiene catorce entradas, correspondientes a las citas del profesional. Por extensión, sólo se incluye una de ellas.

```
<Bundle xmlns="http://hl7.org/fhir">
  <id value="5a07e44f-454a-4bf3-a102-56c8925aaa69"/>
  <meta>
    <lastUpdated value="2016-06-26T18:22:56.635+02:00"/>
  </meta>
  <type value="searchset"/>
  <total value="14"/>
  <link>
    <relation value="self"/>
    <url value="http://localhost:8080/restful-server-example/fhir/Appointment?practitioner=57722647F"/>
  </link>
  <entry>
    <fullUrl value="http://localhost:8080/restful-server-example/fhir/Appointment/1"/>
    <resource>
      <Appointment xmlns="http://hl7.org/fhir">
        <id value="1"/>
        <contained>
          <Patient xmlns="http://hl7.org/fhir">
            <id value="1"/>
            <identifier>
              <type>
                <coding>
                  <system value="http://hl7.org/fhir/v2/0203"/>
                  <code value="MR"/>
                </coding>
              </type>
              <value value="625218621"/>
            </identifier>
            <identifier>
              <type>
                <text value="3"/>
              </type>
              <value value="87743365K"/>
            </identifier>
            <active value="true"/>
            <name>
              <family value="Paz Martínez"/>
              <given value="Lucía"/>
            </name>
            <telecom>
              <system value="phone"/>
              <value value="698336655"/>
            </telecom>
          </Patient>
        </contained>
      </Appointment>
    </resource>
  </entry>
</Bundle>
```

```

        <telecom>
          <system value="email"/>
          <value value="lucia@us.es"/>
        </telecom>
        <gender value="female"/>
        <birthDate value="1973-04-25"/>
        <address>
          <line value="Calle Álava 34"/>
        </address>
      </Patient>
    </contained>
  </contained>
  <Practitioner xmlns="http://hl7.org/fhir">
    <id value="2"/>
    <identifier>
      <type>
        <text value="3"/>
      </type>
      <value value="57722647F"/>
    </identifier>
    <active value="true"/>
    <name>
      <family value="Núñez García"/>
      <given value="Alberto"/>
    </name>
    <telecom>
      <system value="phone"/>
      <value value="698454545"/>
    </telecom>
    <telecom>
      <system value="email"/>
      <value value="alberto@us.es"/>
    </telecom>
    <gender value="male"/>
    <birthDate value="1971-04-09"/>
    <practitionerRole>
      <role>
        <text value="MEDICO"/>
      </role>
      <specialty>
        <text value="Especialista Nefrología"/>
      </specialty>
    </practitionerRole>
  </Practitioner>
</contained>
</contained>
  <Location xmlns="http://hl7.org/fhir">
    <id value="3"/>
    <name value="Sala 4"/>
  </Location>
</contained>
<reason>
  <text value="DIALISIS"/>
</reason>
<description value="Cita semanal"/>
<start value="2016-04-27"/>
<participant>
  <type>
    <text value="Paciente"/>
  </type>
  <actor>
    <reference value="#1"/>
  </actor>
</participant>
<participant>
  <type>
    <text value="Profesional sanitario"/>
  </type>
  <actor>
    <reference value="#2"/>
  </actor>
</participant>
<participant>
  <type>
    <text value="Localización"/>
  </type>
  <actor>
    <reference value="#3"/>
  </actor>
</participant>
</Appointment>
</resource>
</entry>

```

- **Acceso a la información de todos los encuentros del profesional con DNI 57722647F**

Se provee los recursos Encounter de los encuentros del profesional cuyo DNI coincide con el que se le pasa como parámetro, en este caso 57722647F.

La petición se realiza mediante un GET, en el que viaja el parámetro de búsqueda que identifica al profesional. Responde con un HTTP 200 OK, indicando que no ha habido errores durante el proceso (Figura 3-46).

Request	GET http://localhost:8080/restful-server-example/fhir/Encounter?practitioner=57722647F
Request Headers	Accept-Charset: utf-8 Accept: application/xml+fhir;q=1.0, application/json+fhir;q=1.0 User-Agent: HAPI-FHIR/1.5-SNAPSHOT (FHIR Client; FHIR DSTU2; apache) Accept-Encoding: gzip
Response	✓ HTTP/1.1 200 OK
Response Headers	Server: Apache-Coyote/1.1 X-Powered-By: HAPI FHIR 1.5-SNAPSHOT RESTful Server Last-Modified: Sun, 26 Jun 2016 16:36:26 GMT Content-Type: application/xml+fhir; charset=UTF-8 Transfer-Encoding: chunked Date: Sun, 26 Jun 2016 16:36:26 GMT
Result Body	Bundle contains 14 / 14 entries

Figura 3-46. Petición/Respuesta consulta de los encuentros de un profesional

El *bundle* contiene catorce entradas, correspondientes a los encuentros del profesional. Por extensión, sólo se incluye uno de ellos.

```
<Bundle xmlns="http://hl7.org/fhir">
  <id value="10e978e6-7fd1-4c2f-b522-5f999320c9e8"/>
  <meta>
    <lastUpdated value="2016-06-26T18:26:35.650+02:00"/>
  </meta>
  <type value="searchset"/>
  <total value="14"/>
  <link>
    <relation value="self"/>
    <url value="http://localhost:8080/restful-server-example/fhir/Encounter?practitioner=57722647F"/>
  </link>
  <entry>
    <fullUrl value="http://localhost:8080/restful-server-example/fhir/Encounter/4"/>
    <resource>
      <Encounter xmlns="http://hl7.org/fhir">
        <id value="4"/>
        <contained>
          <Patient xmlns="http://hl7.org/fhir">
            <id value="1"/>
            <identifier>
              <type>
                <coding>
                  <system value="http://hl7.org/fhir/v2/0203"/>
                  <code value="MR"/>
                </coding>
              </type>
            </identifier>
            <identifier>
              <type>
                <text value="3"/>
              </type>
            <value value="42266532F"/>
            </identifier>
            <active value="true"/>
            <name>
              <family value="Alvarez Castro"/>
              <given value="Juan"/>
            </name>
            <telecom>
              <system value="phone"/>
              <value value="625698695"/>
            </telecom>
            <telecom>
```

```

        <system value="email"/>
        <value value="juan@usuario.es"/>
    </telecom>
    <gender value="male"/>
    <birthDate value="1985-05-08"/>
    <address>
        <line value="Avenida Colonizadores 23"/>
    </address>
</Patient>
</contained>
<contained>
    <Practitioner xmlns="http://hl7.org/fhir">
        <id value="2"/>
        <identifier>
            <type>
                <text value="3"/>
            </type>
            <value value="57722647F"/>
        </identifier>
        <active value="true"/>
        <name>
            <family value="Núñez García"/>
            <given value="Alberto"/>
        </name>
        <telecom>
            <system value="phone"/>
            <value value="698454545"/>
        </telecom>
        <telecom>
            <system value="email"/>
            <value value="alberto@us.es"/>
        </telecom>
        <gender value="male"/>
        <birthDate value="1971-04-09"/>
        <practitionerRole>
            <role>
                <text value="MEDICO"/>
            </role>
            <specialty>
                <text value="Especialista Nefrología"/>
            </specialty>
        </practitionerRole>
    </Practitioner>
</contained>
<contained>
    <Appointment xmlns="http://hl7.org/fhir">
        <id value="3"/>
        <reason>
            <text value="DIALISIS"/>
        </reason>
        <description value="Cita ordinaria de seguimiento"/>
        <start value="2016-05-18"/>
    </Appointment>
</contained>
<contained>
    <Location xmlns="http://hl7.org/fhir">
        <id value="4"/>
        <name value="Sala 4"/>
    </Location>
</contained>
    <extension url="ObsMed">
        <valueString value="El paciente hace un uso escaso del sistema. Recordarle que las actividades de su
plan de cuidados puede consultarlas en el mismo."/>
    </extension>
    <patient>
        <reference value="#1"/>
    </patient>
    <participant>
        <type>
            <text value="Profesional sanitario"/>
        </type>
        <individual>
            <reference value="#2"/>
        </individual>
    </participant>
    <appointment>
        <reference value="#3"/>
    </appointment>
    <location>
        <location>
            <reference value="#4"/>
        </location>
    </location>
</Encounter>
</resource>
</entry>

```

- **Acceso a método no implementado**

Se accede a un método no implementado en ningún proveedor de recursos. Concretamente, a un método para el recurso Patient al que le estamos pasando como parámetro el identificador del paciente y del profesional a la vez.

La petición se realiza mediante un GET, en el que viajan ambos parámetros. Responde con un HTTP 400 Petición Incorrecta, indicando esa petición no está implementada (Figura 3-47).

➤ Request	GET http://localhost:8080/restful-server-example/fhir/Patient?identifier=prueba&careprovider=prueba
Request Headers	Accept-Charset: utf-8 Accept: application/xml+fhir;q=1.0, application/json+fhir;q=1.0 User-Agent: HAPI-FHIR/1.5-SNAPSHOT (FHIR Client; FHIR DSTU2; apache) Accept-Encoding: gzip
◀ Response	⚠ HTTP/1.1 400 Petición incorrecta
Response Headers	Server: Apache-Coyote/1.1 X-Powered-By: HAPI FHIR 1.5-SNAPSHOT RESTful Server Content-Type: application/xml+fhir;charset=UTF-8 Transfer-Encoding: chunked Date: Sun, 26 Jun 2016 16:40:27 GMT Connection: close

Figura 3-47. Petición/Respuesta acceso a método no implementado

En el contenido del mensaje se representa la causa del error, indicando que el servidor no conoce cómo manejar una petición GET con esos parámetros:

```
<OperationOutcome xmlns="http://hl7.org/fhir">
  <text>
    <status value="generated"/>
    <div xmlns="http://www.w3.org/1999/xhtml">
      <h1>Operation Outcome</h1>
      <table border="0">
        <tr>
          <td style="font-weight: bold;">error</td>
          <td>[</td>
          <td>
            <pre>Invalid request: The FHIR endpoint on this server does not know how to handle
GET operation[Patient] with parameters [[careprovider, identifier]]</pre>
          </td>
        </tr>
      </table>
    </div>
  </text>
  <issue>
    <severity value="error"/>
    <code value="processing"/>
    <diagnostics value="Invalid request: The FHIR endpoint on this server does not know how to handle
GET operation[Patient] with parameters [[careprovider, identifier]]"/>
  </issue>
</OperationOutcome>
```

3.5.3.3. Cliente FHIR Android

Por último, se ha desarrollado un **cliente FHIR Android** para validar el servicio web desde un sistema externo real. Se escoge este tipo de sistema, por la creciente importancia de las aplicaciones móviles en salud, y en particular sobre este sistema operativo.

La funcionalidad de la aplicación es muy sencilla, pues el objetivo de esta es únicamente la validación del servicio. Concretamente, se introduce el DNI de un profesional y, pulsando el botón 'BUSCAR', devuelve la información de los pacientes a su cargo (Figura 3-48).



Figura 3-48. Cliente FHIR Android: Interfaz

Para trabajar con la especificación HL7 FHIR, es necesario añadir al Gradle la dependencia de la librería HAPI FHIR para Android.

```
dependencies {
    compile (group: 'ca.uhn.hapi.fhir', name: 'hapi-fhir-android', version: '1.5-SNAPSHOT', classifier:
'dstu2') {
        exclude module: 'javax.servlet-api'
        exclude module: 'hapi-fhir-base'
    }
}
```

En primer lugar, re realiza la conexión al servidor web, pasando para ello como parámetro su dirección base. Tras ello, realizamos la consulta pertinente contra este:

```
try {
    FhirContext fc = FhirContext.forDstu2();
    //Skip retrieval of conformance statement...
    fc.getRestfulClientFactory().setServerValidationMode(ServerValidationModeEnum.NEVER);
    IGenericClient gc = fc.newRestfulGenericClient("http://10.0.2.2:8080/restful-server-example/fhir");
    //NON-NLS-1$
    return gc.search().forResource(Patient.class)
        .where(new StringClientParam("careprovider").matches().value(profesional))
        .execute();
} catch (Throwable e) {
    Log.e("Err", "Err, handle this better", e);
}
```

En el caso de que el *bundle* suministrado no contenga entradas, se muestra un mensaje indicándolo:

```
if(o.isEmpty()){
    Toast.makeText(getApplicationContext(),
        "Profesional incorrecto o sin pacientes", Toast.LENGTH_SHORT).show();
}
else {
    List<Patient> list = o != null ? o.getResources(Patient.class) : Collections.<Patient>emptyList();
    mAdapter.setData(list);
}
```

Una vez recibido el *bundle*, se recorren sus recursos y se muestra su contenido en la aplicación, utilizando para ello su elemento narrativo:

```
Patient patient = data.get(position);
StringBuilder b = new StringBuilder("NUH/DNI: ");
for (IdentifierDt i : patient.getIdentifier()) {
    if (i.isEmpty()) {
        continue;
    }
    b.append(" ").append(i.getValue());
}
holder.header.setText(b.toString().trim());
holder.content.setText(Html.fromHtml(patient.getText().getDiv().getValueAsString()));
```

3.5.3.4. Pruebas desde el cliente FHIR Android

- **Acceso a la información de los pacientes a cargo del profesional con DNI 57722647F**

Se proveen los recursos Patient de los pacientes que están a cargo del profesional cuyo DNI coincide con el que se le pasa como parámetro, en ese caso inexistente en este caso 57722647F (Figura 3-49).



Figura 3-49. Cliente FHIR Android: Muestra de los pacientes obtenidos

- **Acceso a la información de los pacientes a cargo de un profesional inexistente**

Se proveen los recursos Patient de los pacientes que están a cargo del profesional cuyo DNI se le pasa como parámetro, en este caso inexistente en la base de datos (Figura 3-50).



Figura 3-50. Cliente FHIR Android: Notificación de no información

4 CONCLUSIONES Y LÍNEAS FUTURAS

“A veces creo que hay vida en otros planetas, y a veces creo que no. En cualquiera de los dos casos, la conclusión es asombrosa.”

Carl Sagan

En este proyecto se ha estudiado la especificación HL7 FHIR para el diseño y desarrollo de un servicio web que haga accesible la información de un sistema de e-Salud para pacientes crónicos. Concretamente, este sistema, e-Nefro, da servicio a pacientes renales crónicos, área médica de gran relevancia, en la que, hasta ahora, no se había planteado el uso de la especificación.

La principal característica de FHIR, es la sencillez y rapidez de implantación de los proyectos de integración, algo que en nuestro caso ha cumplido con nota. Una vez estudiado el modelo de datos de e-Nefro, y definidos sus objetos de información, tanto el modelado teórico de esta información a los recursos FHIR, como la implementación del servicio no ha supuesto gran complejidad.

La especificación HL7 FHIR entra en juego por primera vez a la hora de definir el trasvase de la información almacenada de manera propia a los recursos definidos por FHIR. En este aspecto, la especificación se adapta perfectamente a un sistema de las características de e-Nefro, pues de los catorce recursos FHIR que se han necesitado para modelar toda su información, sólo fueron necesarios extender cuatro de ellos. Esto ocurre gracias a la manera en la que FHIR define sus recursos, siendo estos conceptos médicos y administrativos generales, que forman parte de cualquier proceso de atención médica.

Para la implementación del servicio web, se ha empleado una API de código abierto para Java: HAPI FHIR. La API absorbe el principio con el que fue diseñada la especificación, y se aleja por completo de la complejidad a la hora de desarrollar con ella. Como idea base, en el servicio se definen una serie de proveedores de recursos, los cuales se encargan de formarlos y suministrarlos al cliente, de manera que el desarrollo está enfocado en los recursos, al igual que la propia especificación.

Como aspectos negativos, hay que mencionar la corta vida del estándar, lo que hace que aun esté en fase borrador y muchos de los recursos ahora definidos pueden verse modificados en posteriores versiones. Más allá de eso, hay aún muchos frentes abiertos en el desarrollo de HAPI FHIR, concretamente en las extensiones y elementos narrativos. Actualmente, es necesario que el cliente conozca la definición de la extensión y la tenga definida como tal, algo que dificulta la integración de sistemas y por lo que se está trabajando en ello para solventarlo. Así mismo, los elementos narrativos son muy útiles a la hora de visualizar la información de manera cómoda, pero en la API sólo lo implementa para algunos pocos recursos, por lo que es necesario definirlos localmente para los que no cuentan con él.

4.1. Líneas futuras

Aunque el objetivo del trabajo ha sido alcanzado, si se quisiera implementar el servicio web en un entorno real, habría que contemplar una serie de mejoras y ampliaciones:

- **Inclusión del resto de recursos FHIR.** Como se explicó en el apartado de desarrollo del servicio, solo se implementaron lo relativo a los recursos Patient, Practitioner, Organization, Appointment y Encounter, en los cuales aparecían los distintos elementos que caracterizan la especificación: extensiones, recursos contenido y elementos narrativos. Para una implementación real del servicio, sería necesario desarrollar lo referente a los nueve recursos restantes.
- **Mecanismos de seguridad.** En un servicio que se encarga de ofrecer información, y más aún información sanitaria, la seguridad es fundamental. En el servicio desarrollado no se han establecido mecanismos de seguridad, pues lo que se buscaba era estudiar el uso de la especificación en un sistema de las características de e-Nefro. No obstante, cuando se habló de FHIR en el estado del arte, se mencionaron los mecanismos de seguridad que propone, y que habría que implementar:
 - Comunicaciones cifradas mediante TLS/SSL, por ejemplo, utilizando el protocolo HTTPS.
 - Sistemas de identificación y autenticación de usuarios.
 - Sistemas de control de acceso a las operaciones del servidor, mediante etiquetas de seguridad.
 - Sistemas de generación de logs, con los que conseguir un registro de las acciones llevadas a cabo en el servidor y detectar intrusos.
 - Uso de firmas digitales.
- **Validación de recursos.** Una funcionalidad interesante que define la especificación es la de validar los recursos. Los recursos pueden ser validados de varias formas: mediante una plantilla XML, usando el validador que suministra HL7 o en el mismo servicio web mediante la operación *validate*. Sea de la forma que sea, sería necesario comprobar que los recursos están correctamente formados antes de enviarlos o trabajar con ellos.

5 REFERENCIAS

- [1] de Nefrología, Sociedad Española. "El libro blanco de la Nefrología española (III)." *Nefrología* 20.5 (2000): 396-402.
- [2] Arrieta, Javier. "Evaluación económica del tratamiento sustitutivo renal (hemodiálisis, diálisis peritoneal y trasplante) en España." *Nefrología* 1.Suppl 1 (2010): 37-47.
- [3] Montenegro, Jesús, Ricardo Correa-Rotter, and M. Riella. *Tratado de diálisis peritoneal*. Elsevier, 2009: 81-92.
- [4] Calvillo-Arbizu, Jorge, et al. "Aproximación metodológica al diseño de un sistema de teleasistencia para pacientes en prediálisis y diálisis peritoneal." *Nefrología (Madrid)* 34.2 (2014): 1-9.
- [5] Asuman Dogac, "Interoperability in eHealth Systems" en Proceedings of the VLDB Endowment Volume 5 Issue 12, August 2012 Pages 2026-2027
- [6] HL7.org. HL7 Version 2 Product Suite. [En línea]
http://www.hl7.org/implement/standards/product_brief.cfm?product_id=185. [Último acceso: 20 junio 2016]
- [7] HL7.org. HL7 Version 3 Product Suite.
[En línea] https://www.hl7.org/implement/standards/product_brief.cfm?product_id=186. [Último acceso: 20 junio 2016]
- [8] HL7.org. CDA Release 2.
[En línea]http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7. [Último acceso: 20 junio 2016]
- [9] HL7.org. FHIR Overview. [En línea] <https://www.hl7.org/fhir/overview.html>. [Último acceso: 20 junio 2016]
- [10] HL7.org. RESTful API. [En línea] <https://www.hl7.org/fhir/http.html>. [Último acceso: 20 junio 2016]
- [11] HL7.org. FHIR Security. [En línea] <https://www.hl7.org/fhir/security.html>. [Último acceso: 20 junio 2016]
- [12] HL7.org. Resource Index. [En línea] <https://www.hl7.org/fhir/resourcelist.html>. [Último acceso: 20 junio 2016]
- [13] HL7.org. Resource. [En línea] <https://www.hl7.org/fhir/resource.html>. [Último acceso: 20 junio 2016]
- [14] HL7.org. Data Types. [En línea] <https://www.hl7.org/fhir/datatypes.html>. [Último acceso: 20 junio 2016]
- [15] HL7.org. Resource References. [En línea] <https://www.hl7.org/fhir/references.html>. [Último acceso: 20 junio 2016]
- [16] HL7.org. Extensibility. [En línea] <https://www.hl7.org/fhir/extensibility.html>. [Último acceso: 20 junio 2016]
- [17] HL7.org. Narrative. [En línea] <https://www.hl7.org/fhir/narrative.html>. [Último acceso: 20 junio 2016]
- [18] No Magic, Inc. MagicDraw. [En línea] <http://www.nomagic.com/products/magicdraw.html>. [Último acceso: 20 junio 2016]

- [19] Eclipse. Eclipse Mars. [En línea] <https://eclipse.org/mars/> [Último acceso: 20 junio 2016]
- [20] Spring. Spring Framework. [En línea] <https://spring.io/> [Último acceso: 20 junio 2016]
- [21] Hibernate.org. Hibernate Framework. [En línea] <http://hibernate.org/> [Último acceso: 20 junio 2016]
- [22] Apache Maven. Maven Project. [En línea] <https://maven.apache.org/> [Último acceso: 20 junio 2016]
- [23] Apache Tomcat. Tomcat 8.0. [En línea] <https://tomcat.apache.org/download-80.cgi> [Último acceso: 20 junio 2016]
- [24] PostgreSQL.org. PostgreSQL. [En línea] <https://www.postgresql.org/> [Último acceso: 20 junio 2016]
- [25] Android Studio. Android Studio and SDK. [En línea] <https://developer.android.com/studio/index.html> [Último acceso: 20 junio 2016]
- [26] HAPI. HAPI FHIR. [En línea] <http://hapifhir.io/> [Último acceso: 20 junio 2016]
- [27] Haveman-Nies, Annemien, Lisette CPGM de Groot, and Wija A. van Staveren. "Dietary quality, lifestyle factors and healthy ageing in Europe: the SENECA study." *Age and ageing* 32.4 (2003): 427-434.
- [28] Thomas A. Buchanan and Anny H. Xiang, "Gestational diabetes mellitus" en *J Clin Invest.* 2005;115(3):485–491.
- [29] Duane Bender, "HL7 FHIR: An Agile and RESTful approach to healthcare information exchange" en *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, 2013 Page(s):326 – 331
- [30] Mark L. Braunstein, "Patient — Physician collaboration on FHIR (Fast Healthcare Interoperability Resources)" en *Collaboration Technologies and Systems (CTS)*, 2015 International Conference. Pages 501-503
- [31] McCallie Jr, David P. "Emerging Clinical Decision Support Technology for the Twenty First Century." *Healthcare Information Management Systems*. Springer International Publishing, 2016. 477-491.
- [32] Beredimas, Nikolaos, et al. "A reusable ontology for primitive and complex HL7 FHIR data types." *Engineering in Medicine and Biology Society (EMBC)*, 2015 37th Annual International Conference of the IEEE. IEEE, 2015.
- [33] Luz, Maria Penna, et al. "Providing Full Semantic Interoperability for the Fast Healthcare Interoperability Resources Schemas with Resource Description Framework." *Healthcare Informatics (ICHI)*, 2015 International Conference on. IEEE, 2015.
- [34] Dong, Xiao, Reza Samavi, and Thodoros Topaloglou. "COC: An Ontology for Capturing Semantics of Circle of Care." *Procedia Computer Science* 63 (2015): 589-594.
- [35] Cornet, R. "Combining Archetypes with Fast Health Interoperability Resources in Future-proof Health Information Systems." *Digital Healthcare Empowering Europeans: Proceedings of MIE2015* 210 (2015): 180.
- [36] Franz, Barbara, Andreas Schuler, and O. Kraus. "Applying FHIR in an Integrated Health Monitoring System." *EJBI* 11.2 (2015).
- [37] Schwartze, Jonas, et al. "An HL7-FHIR-based Object Model for a Home-Centered Data Warehouse for Ambient Assisted Living Environments." *Studies in health technology and informatics* 216 (2014): 1060-1060.
- [38] Schuler, Andreas, Barbara Franz, and Oliver Krauss. "A Comparison of Data Traffic in Standardized Personal Health Monitoring Solutions." *International Journal of Electronics and Telecommunications* 61.2 (2015): 143-149.
- [39] Dagale, Haresh, et al. "CyPhyS+: A Reliable and Managed Cyber-Physical System for Old-Age Home Healthcare over a 6LoWPAN Using Wearable Motes." *Services Computing (SCC)*, 2015 IEEE International Conference on. IEEE, 2015.

- [40] Alterovitz, Gil, et al. "SMART on FHIR Genomics: Facilitating standardized clinico-genomic apps." *Journal of the American Medical Informatics Association* 22.6 (2015): 1173-1178.
- [41] Kimura, E., and K. Ishihara. "Internal domain-specific language based on Arden Syntax and FHIR." *Studies in health technology and informatics* 216 (2014): 955-955.
- [42] Morrison, James J., et al. "Constructing a Computer-Aided Differential Diagnosis Engine from Open-Source APIs." *Journal of digital imaging* (2016): 1-4.
- [43] Luz, Maria Penna, et al. "Providing Full Semantic Interoperability for the Fast Healthcare Interoperability Resources Schemas with Resource Description Framework." *Healthcare Informatics (ICHI), 2015 International Conference on. IEEE, 2015.*
- [44] Lamprinakos, Georgios C., et al. "Using FHIR to develop a healthcare mobile application." *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on. IEEE, 2014.*
- [45] Mandel, Joshua C., et al. "SMART on FHIR: a standards-based, interoperable apps platform for electronic health records." *Journal of the American Medical Informatics Association* (2016): ocv189.
- [46] Pais, Sarita, et al. "Data integration for mobile wellness apps to support treatment of GDM." *Proceedings of the Australasian Computer Science Week Multiconference. ACM, 2016.*
- [47] Kay, Misha, Jonathan Santos, and Marina Takane. "mHealth: New horizons for health through mobile technologies." *World Health Organization* (2011): 66-71.
- [48] Kasthurirathne, S. N., et al. "Towards Standardized Patient Data Exchange: Integrating a FHIR Based API for the Open Medical Record System." *Studies in health technology and informatics* 216 (2014): 932-932.
- [49] Kasthurirathne, Suranga N., et al. "Enabling Better Interoperability for HealthCare: Lessons in Developing a Standards Based Application Programming Interface for Electronic Medical Record Systems." *Journal of medical systems* 39.11 (2015): 1-8.
- [50] Mamlin, Burke W., et al. "Cooking up an open source EMR for developing countries: OpenMRS-a recipe for successful collaboration." *AMIA. 2006.*
- [51] Jawaid, Hira, et al. "Healthcare Data Validation and Conformance Testing Approach Using Rule-Based Reasoning." *Health Information Science. Springer International Publishing, 2015. 241-246.*
- [52] RINNER, Christoph; DUFTSCHMID, Georg. Bridging the Gap between HL7 CDA and HL7 FHIR: A JSON Based Mapping. En *Health Informatics Meets EHealth: Predictive Modeling in Healthcare—From Prediction to Prevention. Proceedings of the 10th EHealth2016 Conference.* IOS Press, 2016. p. 100.
- [53] HL7.org. Search. [En línea] <https://www.hl7.org/fhir/search.html>. [Último acceso: 20 junio 2016]
- [54] HL7.org. Resource Bundle. [En línea] <https://www.hl7.org/fhir/bundle.html>. [Último acceso: 20 junio 2016]