

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“INTEROPERABILIDAD EN IMÁGENES MÉDICAS:
IMPLEMENTACIÓN DEL ESTÁNDAR HL7-FHIR EN XNAT
PARA LA TRANSFERENCIA SEGURA DE IMÁGENES
MÉDICAS”

NICOLÁS ESTEBAN VIVANCO LOBOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Julio Sotelo Parraguez
Profesor Correferente: Mauricio Araya López

Julio - 2025



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título; ☐ Tesis de Postgrado;

Título del trabajo: INTEROPERABILIDAD EN IMÁGENES MÉDICAS: IMPLEMENTACIÓN DEL ESTÁNDAR HL7-FHIR EN XNAT PARA LA TRANSFERENCIA SEGURA DE IMÁGENES MÉDICAS

Nombre del candidato(a): Nicolás Esteban Vivanco Lobos

Carrera / Grado: Ingeniería Civil Informática

Campus: Casa Central Valparaíso ; Departamento: Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Julio Sotelo Parraguez, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL

☒ El trabajo **NO contiene información que amerite confidencialidad** y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (embargo) por:

☐ 6 meses; ☐ 12 meses; ☐ 2 años; ☐ 3 años; ☐ 5 años; ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 11-07-2025 ; Firma: 

Estudiante o Candidato(a):

Fecha: 10-07-2025 ; Firma: 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Dedicado a mi familia y amigos que me han acompañado a lo largo de este proceso y durante toda mi vida, entregándome su cariño y apoyo incondicional. Para esos angelitos que cuidan de mí desde el cielo. A todos ustedes los quiero mucho. Muchas gracias.

AGRADECIMIENTOS

Quisiera agradecer en primer lugar a mis padres, Edith y Juan, por brindarme todo el apoyo, cariño y amor durante este proceso, agradecer todo el sacrificio y esfuerzo que han hecho para que yo llegue hasta acá y que no me faltara nada durante el trayecto a pesar de las dificultades.

A mi tía, Cathy, por ser mi segunda mamá en la vida, por enseñarme el cómo estudiar, por exigirme para siempre alcanzar las mejores calificaciones, por regalarme y por hacerme hinchita del hermoso club Universidad Católica.

A mi hermanita, Ámbar, por ser la razón por la que siempre me mantuve en pie y pude lograr superar los obstáculos a lo largo de este proceso, por compartir esa pasión por el fútbol que nos une y nos saca risas o enojos cada fin de semana.

A mi tata, Lobos, por preocuparse al igual que mis padres que nunca me faltara nada, por ser mi figura paterna e inspiración en esta vida, gracias por todas las enseñanzas y sabiduría que me has entregado.

A mis primos, que son otros hermanos más, que siempre me han sacado sonrisas y han estado presente en todo, celebrando cada logro como si fueran suyos, que me han dado palabras de aliento y consejos cuando los necesitaba, sobretodo a esa burra que le gusta la redacción y me ha ayudado para que suene bonito este documento.

También quisiera agradecer a mis amigos que me han acompañado durante todo este trayecto, por ser mi escape en los días difíciles, por estar siempre dispuestos a escuchar mis problemas, por las risas y burlas en cada carrete y horas perdidas jugando algún videojuego. Pero especialmente a Manuel, José y Ale, aquellos que fueron luz cuando solo veía oscuridad, por convertirse en mi gran soporte y apoyo emocional en esta etapa universitaria, gracias por todo ese desvelo escuchando lo que me aconplejaba, no sé que haría sin su amistad. Por otro lado, quisiera agradecer a Manuel por presentarme esta carrera y que estaba equivocado en el primer año, carrera que ahora es mi pasión.

Agradecer a mi profesor guía, Julio Sotelo, por su dedicación y tiempo para sacar este proyecto adelante, que a pesar de ser de distintas sede y no tenerlo como profesor de ningún ramo, estuvo siempre dispuesto a escuchar mi idea y apoyarme a desarrollarla, ambos logramos aprender mucho de este mundo de la interoperabilidad.

Finalmente a mis angelitos que me cuidan a diario, ojalá poder hacerlos sentir orgullosos de este logro, los extraño demasiado.

Gracias a todos los que han sido participe de una u otra forma y han dejado huella en mi vida y en este proceso.

RESUMEN

Resumen— La presente memoria aborda el proceso de instalación, configuración e integración del estándar HL7 FHIR en XNAT, una plataforma de informática en imagenología médica de código abierto, con el objetivo de facilitar el intercambio seguro de datos de imágenes médicas para fines de investigación. Se describen dos métodos de despliegue, uno basado en la virtualización tradicional mediante máquinas virtuales y otro mediante contenedores Docker. El desarrollo se dividió en dos fases: una implementación local y otra en un servidor ubicado en la Universidad Técnica Federico Santa María. Tras realizar pruebas piloto en ambos entornos, se evaluaron las ventajas y limitaciones de cada enfoque, optando por Docker para su despliegue final en el servidor por su eficiencia y menor consumo de recursos. Finalmente, se logró integrar exitosamente el complemento que permite la estandarización de los datos a HL7 FHIR, dejando la plataforma accesible públicamente para investigadores, bajo ciertas restricciones propias del servidor utilizado.

Palabras Clave— Interoperabilidad; XNAT; HL7 FHIR; Imágenes Médicas.

ABSTRACT

Abstract— The present thesis addresses the installation, configuration and integration process of the HL7 FHIR standard in XNAT, an open source medical imaging informatics platform, with the goal of easing the safe exchange of medical images for research purposes. Two deployment methods are described, one based on traditional virtualization through virtual machines and another through Docker containers. The development was divided in two stages: a local implementation and another in a server located in the Universidad Técnica Federico Santa María. After conducting pilot tests in both environments, the advantages and limitations of each approach were evaluated, choosing Docker for the final deployment in the server for its efficiency and lower resource consumption. Finally, the plugin that allows the standardization of data to HL7 FHIR was successfully integrated, leaving the platform publicly accessible for researchers, subject to certain restrictions specific to the server used.

Keywords— Interoperability; XNAT; HL7 FHIR; Medical Images.

GLOSARIO

ANSI: American National Standards Institute.

AWS: Amazon Web Services

DICOM: Digital Imaging and Communication in Medicine

DNS: Domain Name System.

EHR: Registro Electrónico de Salud.

HCP: Human Connectome Project.

HIE: Health Information Exchange.

HISP: Health Information Service Provider.

HL7: Health Level Seven.

HL7 CDA: HL7 Clinical Document Architecture.

HL7 FHIR: Fast Healthcare Interoperability Resources.

HL7 V2: HL7 Version 2.

HL7 V3: HL7 Version 3.

OASIS: Open Access Series of Imaging Studies.

OHIF: Open Health Imaging Foundation.

SIDRA: Sistema de Información de Redes Asistenciales.

SIRE: Sistema de Información de Receta Electrónica.

TICS: Tecnologías de la Información y la Comunicación.

UTFSM: Universidad Técnica Federico Santa María.

VM: Máquina Virtual

WSL: Windows Subsystem for Linux

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 Interoperabilidad en el mundo	3
1.2 Situación nacional actual	4
1.3 Antecedentes de interoperabilidad sanitaria en Chile	6
1.3.1 SIDRA	6
1.3.2 SIRE	7
1.3.3 Modificación Ley N° 20.584	7
1.4 Beneficios de la Interoperabilidad Sanitaria	8
1.5 Objetivos	9
1.5.1 Objetivo General	9
1.5.2 Objetivos Específicos	9
CAPÍTULO 2: MARCO CONCEPTUAL	10
2.1 Niveles de Interoperabilidad	10
2.1.1 Interoperabilidad Básica	10
2.1.2 Interoperabilidad Sintáctica	10
2.1.3 Interoperabilidad Semántica	10
2.1.4 Interoperabilidad Organizacional	11
2.2 Estándares Internacionales de Interoperabilidad	11
2.2.1 HL7 FHIR	12
2.3 XNAT	14
2.3.1 Integración de HL7 FHIR a XNAT	14
2.4 Virtualización	15
2.4.1 Virtualización de Hardware	16
2.4.2 Virtualización basada en contenedores	16
2.4.3 Comparación entre Contenedores y Máquinas Virtuales	17
2.5 Estado del Arte	18
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	23
3.1 Instalación local de XNAT en Máquina Virtual	23
3.1.1 Prerequisitos de Instalación	23

3.1.2	Instalación de VirtualBox	23
3.1.3	Instalación de Git	24
3.1.4	Instalación de Vagrant	24
3.1.5	Instalación de xnat-vagrant	25
3.2	Configuraciones adicionales para xnat-vagrant	27
3.2.1	Cambio de contraseña de usuarios	27
3.2.2	Tomcat	28
3.2.3	Plugins	29
3.2.4	Servidor SMTP	29
3.3	Instalación local de XNAT usando Docker	31
3.3.1	Prerrequisitos de Instalación	31
3.3.2	Instalación de Docker	31
3.3.3	Instalación de PostgreSQL	32
3.3.4	Instalación de xnat-docker-compose	32
3.3.5	Plugins	34
3.4	xnat-fhir-plugin	35
3.4.1	Prerrequisitos para generar el complemento	35
3.4.2	Instalar Gradle	35
3.4.3	Instalar xnat-fhir-plugin	36
3.5	Instalación de XNAT en un servidor	38
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		41
4.1	Pruebas pilotos de la instalación de XNAT de forma local	41
4.1.1	Creación de usuario	41
4.1.2	Acceso a Proyectos	43
4.1.3	Acceso a Plataforma desde otro dispositivo	46
4.1.4	Subir una sesión de imágenes médicas	47
4.1.5	Estandarizar la información de un paciente con el complemento xnat-fhir-plugin	50
4.2	Pruebas piloto de la instalación de XNAT en el servidor	52
4.2.1	Creación de Usuario	52
4.3	Acceso a Proyectos	55
4.3.1	Subir y visualizar un archivo de imagen médica	58
4.3.2	Estandarizar la información de un paciente con el complemento xnat-fhir-plugin previamente instalado en el servidor	59
CAPÍTULO 5: CONCLUSIONES		62
5.1	Conclusiones Generales	62
5.2	Cumplimiento de Objetivos	65
5.3	Trabajo a Futuro	66
5.4	Palabras finales del Autor	67
ANEXOS		68
	Anexo A	68

ÍNDICE DE FIGURAS

1	Portal de Resultados de Exámenes.	2
2	Agenda horaria de un doctor en Centromed.	3
3	Árbol del problema.	6
4	Logo HL7 FHIR.	12
5	Ejemplo de Recurso: Paciente.	13
6	Inicio Plataforma XNAT.	14
7	Pequeño extracto del formulario del complemento <code>xnat-fhir-plugin</code>	15
8	Diagrama de funcionamiento de una VM.	16
9	Diagrama de funcionamiento de Contenedores.	17
10	Logo VirtualBox.	24
11	Logo Git.	24
12	Comandos de Instalación de Vagrant.	25
13	Logo Vagrant.	25
14	Comando para clonar repositorio utilizando WSL.	25
15	Comando para crear archivo <code>local.yaml</code> en WSL.	25
16	Comando para acceder a <code>local.yaml</code>	26
17	Comando para clonar repositorio utilizando Bash.	26
18	Comando para clonar repositorio utilizando Bash.	26
19	Comando para editar <code>local.yaml</code>	26
20	Plataforma XNAT.	27
21	Comando Vagrant SSH.	27
22	Cambio de contraseña usuario <code>xnat</code>	28
23	Cambio de contraseña usuario <code>vagrant</code>	28
24	Acceso a máquina virtual con comandos Vagrant.	28
25	Acceso a máquina virtual con comandos SSH.	28
26	Comando inicio automático Tomcat.	29
27	Instalación de plugin.	29
28	Configuración de Postfix.	30
29	Logo Docker.	31
30	Logo PostgreSQL.	32
31	Copia de archivo <code>default.env</code> en <code>.env</code>	32
32	Comando para iniciar la creación de los contenedores e instalación de XNAT .	32
33	Contenedor de XNAT en ejecución.	33
34	Inicio XNAT Docker.	34
35	Comando <code>docker ps -a</code>	34
36	Comando para transferir plugins al contenedor de Docker	35
37	Comando para instalar SDKMAN	36
38	Paquetes que se deben agregar en el archivo <code>build.gradle</code>	36
39	Código original de <code>xnat-fhir-plugin</code>	37
40	Cambios que se deben hacer al código de la Figura 39	37
41	Código que se debe agregar al final del código en el archivo <code>build.gradle</code> .	38
42	Ejemplo de conexión SSH al servidor	38

43	Sitio web https://albert-xnat.inf.santiago.usm.cl/	39
44	Opción de activación de cuenta de forma automática deshabilitada	39
45	Opciones de Registro.	41
46	Comandos para ingresar a ver la tabla de datos de los usuarios	42
47	Usuarios de la tabla <code>xnat_user</code>	42
48	Página principal del usuario <code>testUser</code>	43
49	Página principal del usuario <code>User2</code>	44
50	Página principal del usuario <code>userProject</code>	44
51	Tabla <code>xnat_projectdata</code> de la base de datos que registra los proyectos de la plataforma	45
52	Consulta para obtener las relación de los usuarios con los proyectos.	45
53	Resultados de la consulta en la base de datos que muestra las relaciones de los usuarios con los proyectos del sistema.	46
54	Página principal desde el portátil.	46
55	Página principal del usuario <code>userProject</code> desde el portátil.	47
56	Imágenes almacenadas en el contenedor <code>xnat-web</code> al ser cargadas desde la plataforma web.	48
57	Visualizador OHIF con las imágenes cargadas en la plataforma XNAT.	48
58	Imágenes médicas del Proyecto Prueba accedido desde el usuario <code>User2</code>	49
59	Visualizador OHIF accedido desde el usuario <code>User2</code>	50
60	Archivos generados luego de seguir las instrucciones de la sección 3.4.	50
61	Complemento <code>xnat-fhir-plugin</code> ya instalado en la plataforma XNAT.	51
62	XML del experimento del paciente SJ2.	52
63	Error al iniciar sesión con el nuevo usuario registrado	53
64	Datos del nuevo usuario registrado	54
65	Acceso exitoso luego de habilitar la cuenta del nuevo usuario	54
66	Página principal del usuario <code>mainUser</code>	56
67	Página principal del usuario <code>secondUser</code>	56
68	Página principal del usuario <code>thirdUser</code>	57
69	Subir imagenes médicas a la plataforma utilizando la opción de Compressed Uploader	58
70	Visor <code>ohif-viewer-3.7.0-XNAT-1.8.10</code> instalado en el servidor	59
71	Complemento <code>xnat-fhir-plugin</code> correctamente instalado en el servidor	60
72	Complemento <code>xnat-fhir-plugin</code> correctamente instalado en el servidor	61
73	Configurar la información de la página web XNAT.	68
74	Configuración de almacenamiento de XNAT.	68
75	Servidor SMTP de XNAT.	69
76	Configuraciones generales de XNAT.	69

ÍNDICE DE TABLAS

1	Parámetros de configuración del servidor SMTP Gmail.	30
2	Formulario para crear usuario.	42

3	Integrantes de cada proyecto	43
4	Datos a ingresar en el formulario del estándar HL7 FHIR	51
5	Datos para registrar un nuevo usuario en XNAT	53
6	Integrantes de cada proyecto en el servidor	55
7	Datos a ingresar en el formulario del estándar HL7 FHIR del servidor	60

INTRODUCCIÓN

La interoperabilidad es la capacidad de las aplicaciones y los sistemas para intercambiar datos de forma segura y automática, independientemente de los límites geográficos, políticos u organizativos(AWS, s.f.). Este concepto ha cobrado una creciente relevancia en el ámbito de las tecnologías de la información con el fin de facilitar el intercambio de datos permitiendo que distintos sistemas puedan trabajar de manera eficiente. En áreas como la salud, donde se manejan datos sensibles, este concepto es fundamental para asegurar que la información se comparta de manera correcta y oportuna entre hospitales, centros médicos o investigadores, mejorando así la atención de los pacientes, el funcionamiento de los servicios o realizando nuevos descubrimiento para el desarrollo de la sociedad.

Con el objetivo de contribuir al avance de la investigación médica, la presente memoria se centra en exponer e implementar XNAT, una plataforma de informática en imagenología diseñada para facilitar tareas comunes relacionadas con la gestión, el control de calidad y la productividad en el manejo de imágenes médicas. XNAT destaca por su alta capacidad de personalización y su compatibilidad con estándares internacionales en salud, como HL7-FHIR(Health Level Seven, 2011), así como su integración con visores especializados, lo que la convierte en una herramienta versátil. Gracias a estas características, XNAT permite respaldar una amplia variedad de proyectos basados en imágenes, contribuyendo a la gestión clínica(Neuroinformatics Research Group, 2017).

Este trabajo se estructura en 5 capítulos. En el *Capítulo 1: Definición del Problema*, se presenta el contexto en el que surge la problemática abordada, se analiza la situación actual del país en relación con el concepto de interoperabilidad, y se describen los beneficios que esta puede aportar. Además, se exponen los objetivos generales y específicos que guían el desarrollo de esta memoria.

En el *Capítulo 2: Marco Conceptual*, se abordan conceptos claves en los que se basa esta memoria. Se definen conceptos técnicos relevantes, se explican algunos de los estándares utilizados y se presenta un estudio del estado del arte en torno a la interoperabilidad a nivel internacional, junto con antecedentes del uso de la plataforma XNAT.

En el *Capítulo 3: Propuesta Solución*, se presentan los dos métodos de instalación más utilizados de la plataforma XNAT. Además, se detallan los pasos necesarios para su configuración, personalización y uso, con el objetivo de adaptar la plataforma a diferentes contextos y necesidades específicas.

En el *Capítulo 4: Validación de la Solución*, se presentan las pruebas piloto realizadas sobre el correcto funcionamiento de la plataforma XNAT con sus respectivos resultados.

Finalmente, en el *Capítulo 5: Conclusiones*, se exponen las conclusiones del uso de la plataforma XNAT, cumplimiento de objetivos y trabajo a futuro a realizar.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

La evolución de las tecnologías ha ido en aumento con el pasar de los años y el concepto de interoperabilidad en los sistemas va tomando fuerza con el fin de poder mejorar los sistemas de información ya existentes en diferentes contextos de la vida cotidiana, pudiendo facilitar el traspaso de datos de manera confiable. La interoperabilidad es la capacidad de las aplicaciones y los sistemas para intercambiar datos de forma segura y automática, independientemente de los límites geográficos, políticos u organizativos(AWS, s.f.).

Quizás no nos damos cuenta que estamos en constante presencia de interoperabilidad en nuestro día a día, un ejemplo claro es en los sistemas web de locales comerciales, donde si queremos ver la disponibilidad de los productos, los distintos centros de distribución a lo largo del país comparten la información de la cantidad disponible del objeto deseado en tiempo real y le presentan a los usuarios en qué lugar se puede retirar su compra.

En el ámbito de la salud, el sitio web de Centromed la interoperabilidad está presente en diversas áreas, como la reserva de horas médicas, la solicitud de exámenes y el acceso a resultados de laboratorio e imagenología. Los pacientes pueden consultar estos resultados desde cualquier dispositivo con conexión a internet, ingresando a su cuenta creada por el centro médico. Para garantizar la seguridad de los datos, el acceso requiere un código especial enviado por mensaje de texto al celular del paciente. (Ver figura 1)

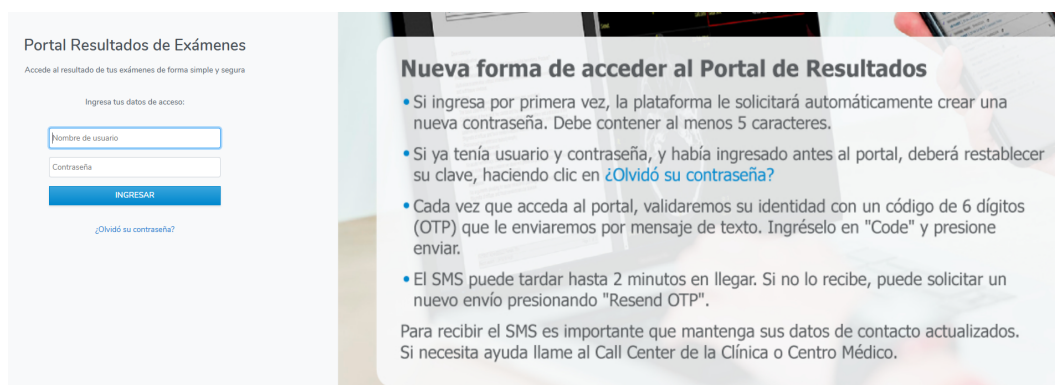


Figura 1: Portal de Resultados de Exámenes.
Fuente: Página web Centromed

Si deseamos reservar una hora con un doctor, podemos buscarlo por especialidad, nombre o previsión. Al seleccionar al profesional, se muestra su agenda con los horarios disponibles y aquellos que ya han sido reservados, permitiéndonos elegir el que mejor se adapte a nuestras necesidades.

Paso 2 de 5:

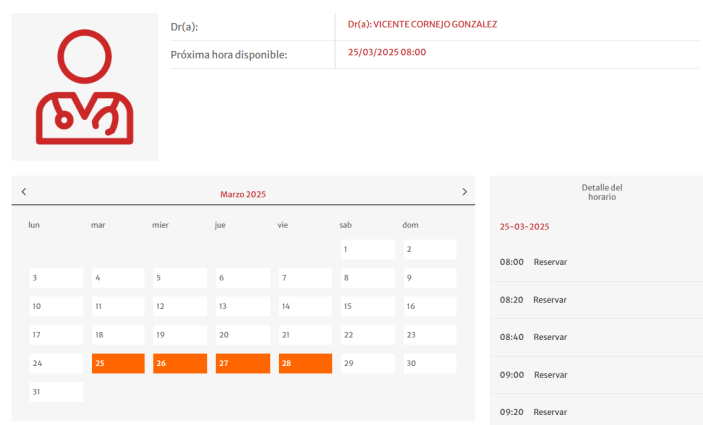


Figura 2: Agenda horaria de un doctor en Centromed.
Fuente: Página web Centromed

En el ámbito de la salud, sin duda, es donde este concepto puede aprovecharse al máximo, ya que la disponibilidad de la información es crucial para que especialistas clínicos actúen oportunamente y protejan la vida de los pacientes. Por ello, esta memoria se enfocará en este sector.

1.1. Interoperabilidad en el mundo

Health Level Seven International (HL7) es una organización que se encarga de proporcionar normas que faciliten la interoperabilidad de los datos sanitarios a escala mundial. Ha creado varios estándares que facilitan la transformación digital de los sistemas alrededor del mundo, entre ellos se destaca HL7 FHIR¹(Health Level Seven, 2007).

Los países desarrollados están prefiriendo utilizar el estándar HL7 FHIR debido a: a) su compatibilidad tecnológica, b) su alineación con los objetivos de innovación y eficiencia en los sistemas de salud, c) y por su capacidad de facilitar la interoperabilidad(HL7 International & Firely, 2025).

Esta última característica es la que está consolidando al estándar HL7 FHIR como el preferido para la interoperabilidad en salud debido a su capacidad para facilitar una conectividad sencilla, flexible y moderna, facilitando la transformación digital del sector sanitario.

La encuesta realizada en 2025 por HL7 con el fin de saber el estado del estándar, señala que el 71 % de los países encuestados ya está utilizando FHIR en algunos casos de uso, y un 54 % de ellos desea implementar y adoptar este, para futuros proyectos en los próximos años(HL7 International & Firely, 2025).

¹Explicado en profundidad en la sección 2.2.1

La versión más utilizada actualmente es FHIR R4, considerada el estándar principal por el 40 % de los países participantes. Por otro lado, el 79 % de los países encuestados disponen de una guía nacional de implementación(HL7 International & Firely, 2025).

Entre los proyectos más destacados a nivel internacional se encuentra el caso de Brasil, donde el Ministerio de Salud ha impulsado la creación de **BR-Core**, una guía de implementación del estándar HL7 FHIR promoviendo la interoperabilidad semántica y técnica entre los sistemas de salud, para una mayor eficiencia, seguridad y calidad en el intercambio de información sanitaria(Ministério da Saúde do Brasil / HL7 Brasil, 2023).

En India, las iniciativas nacionales de *Ayushman Bharat Digital Mission* y *National Health Claims Exchange* establecen que el uso obligatorio del estándar FHIR para el intercambio de datos clínicos a nivel nacional(BioSpectrum India, 2024).

La Unión Europea presenta la iniciativa **Espacio Europeo de Datos Sanitarios**(EHDS) que tiene como objetivo la adopción armonizada del estándar FHIR en todos los Estados Miembros creando un marco seguro y unificado para compartir datos sanitarios, permitiendo a los proveedores de atención médica, investigadores y formuladores de políticas acceder a información crítica con mayor facilidad, mejorando los resultados de los pacientes, impulsando la innovación y mejorando la salud pública(BC Platforms, 2024).

1.2. Situación nacional actual

El sistema de salud chileno enfrenta un desafío crucial para garantizar una atención médica eficiente y segura en un entorno digitalizado: la interoperabilidad de sus servicios y la protección de los datos de los pacientes. Estos conceptos están estrechamente vinculados, ya que el intercambio de información entre centros médicos debe realizarse de manera segura, dado que los datos médicos de los usuarios son extremadamente sensibles y su uso indebido puede tener graves consecuencias(Portal Red Salud, 2023).

La falta de interoperabilidad genera descoordinación entre los distintos centros asistenciales del país, lo que a menudo conlleva la duplicación y dispersión de los datos médicos en los diversos servicios de salud que atienden al paciente. Como consecuencia, se incurre en gastos innecesarios al repetir exámenes médicos debido a la falta de disponibilidad de los resultados para cualquier profesional que los requiera.

Además, la situación actual implica que el paciente asuma el papel de portador y transmisor de su historial clínico, lo que en ocasiones dificulta la calidad y continuidad de la atención médica. Esta imposibilidad de acceder al historial clínico desde cualquier centro médico y que el paciente sea el encargado de informar sus antecedentes representa una carga adicional de trabajo para los profesionales de la salud, y de esta forma limita la entrada de un diagnóstico correcto.

En este contexto, se pone en riesgo la seguridad del paciente, ya que la falta de acceso oportuno a información clínica relevante puede comprometer la toma de decisiones médicas adecuadas y afectar negativamente el curso del tratamiento.

Este problema en el país se atribuye en gran medida a la falta de voluntad política para mejorar el sistema de salud. Los planes gubernamentales cambian cada cuatro años, lo que resulta en asignaciones presupuestarias variables para el sector de la salud y en una falta de enfoque en la modernización hacia un sistema de salud digital. Además, la ausencia de leyes que regulen la interoperabilidad es otro aspecto relevante; a principios de 2023, se modificó la Ley N° 20.584, que regula los derechos y deberes en la atención sanitaria, con el objetivo de establecer la interoperabilidad de las fichas clínicas, abordando así solo una parte de la necesidad de regular la compartición de información entre centros médicos. Es evidente que existen otras áreas críticas que también van a requerir una regulación adecuada para facilitar el intercambio de datos de manera segura entre diferentes entidades de salud.

Chile cuenta con un sistema de salud fragmentado que combina servicios públicos y privados, así como diversos proveedores de atención médica. Esta fragmentación dificulta la estandarización de sistemas y datos, lo que a su vez afecta a la interoperabilidad entre las diferentes entidades de salud.

Por último, la falta de conocimiento sobre los estándares internacionales que facilitan la implementación de la interoperabilidad también contribuye a esta problemática. Es fundamental promover la capacitación y la difusión de estos estándares entre los profesionales de la salud y los responsables políticos, para que comprendan su importancia y adopten las mejores prácticas en el desarrollo de sistemas de salud digital.

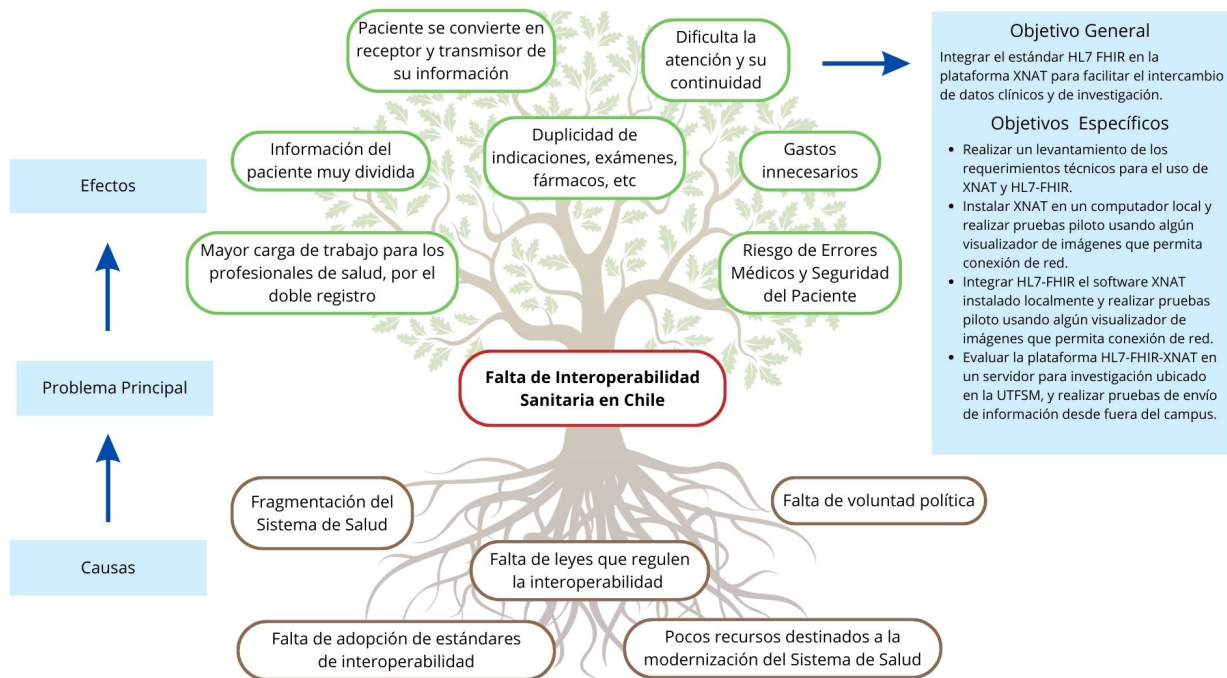


Figura 3: Árbol del problema.
Fuente: Elaboración Propia

1.3. Antecedentes de interoperabilidad sanitaria en Chile

Chile ha intentado avanzar en los desafíos e iniciativas propuestas tanto por la Organización Mundial de la Salud como la Organización Panamericana de la Salud sobre la transformación digital de los sistemas de la salud, destinando presupuesto para la creación de proyectos, servicios o leyes que aportan al desarrollo del país.

1.3.1. SIDRA

El Sistema de Información de Redes Asistenciales (SIDRA) puesto en práctica en el año 2015 es un programa diseñado para mejorar la calidad y eficiencia de los cuidados de salud pública dentro de los Servicios de Salud de Chile, al proveer software para automatizar la entrega local de cuidados e intercambiar información clínica (SSMC, 2019). Consta de 5 módulos:

- **Sistema de lista de espera:** automatiza el proceso de derivar pacientes desde una institución médica a otra. El registro médico del paciente, ya completo con todos los pedidos de pruebas y resultados, se comparte con seguridad entre las dos instituciones.
- **Agenda:** le permite gestionar recursos, personas, equipo y suministros.

- **Urgencias:** automatiza y estandariza el registro de pacientes en entornos de cuidados de urgencias.
- **Farmacia:** gestiona el inventario de medicamentos de cada hospital.
- **Programa registro de población bajo control:** aprueba y promueve programas de salud implementados por el gobierno.

El proyecto se basó en introducir tecnologías en tareas independientes, pero a pesar de que se digitalizó y automatizó procesos, solo se comparte la información dentro del mismo servicio de salud y no entre los distintos centros.

1.3.2. SIRE

El Sistema de Información de Receta Electrónica(SIRE) tuvo su marcha blanca desde el año 2021 y ya está implementado desde octubre del año 2023, que tiene como objetivo disminuir las barreras de acceso a medicamentos, disminuir efectos adversos prevenibles a medicamentos, seguimiento y trazabilidad en los tratamientos, y tener un control legal en el sistema de prescripción de medicamentos (Pro Salud Chile, 2023).

SIRE es una plataforma web que facilita la emisión, transmisión y gestión de las recetas médicas, consta de un módulo de prescripción, un módulo de dispensación y un módulo de paciente web y móvil, con el fin de que cualquier profesional prescriptor autorizado pueda emitir una receta sin mayores complicaciones, que el paciente tenga la receta disponible en cualquier momento y lugar, y que el farmacéutico pueda tener un mejor control y comprensión de lo solicitado por el paciente.

El sistema tiene varios trabajos futuros que se espera que lleven a cabo hasta el 2026, pero es lo más cercano a interoperabilidad que tiene nuestro país, ya que la información farmacológica se encuentra centralizada y disponible desde cualquier centro asistencial y por los profesionales prescriptores autorizados, evitando situaciones de pérdida, mejorando la trazabilidad farmacológica del paciente y teniendo datos de calidad y accesibles para la toma de decisiones clínicas.

1.3.3. Modificación Ley N° 20.584

En enero de 2023 ingresó el proyecto ley a la Cámara de Diputados y Senadores sobre la modificación de la ley N° 20.584, que regula los derechos y deberes que tienen las personas en relación con acciones vinculadas a su atención en salud, con el objeto de establecer la interoperabilidad de las fichas clínicas, actualmente se encuentra en trámite de aprobación presidencial(Cámara de Diputados y Diputadas, 2023).

El proyecto tiene como objetivo garantizar la continuidad del cuidado del paciente independiente del prestador, ya sean públicos o privados, fijando estándares para la ficha clínica, reforzando la seguridad de la información y la protección de los datos, logrando modernizar el sistema de salud actual requiriendo una buena conexión a internet en todos los centros asistenciales del país.

1.4. Beneficios de la Interoperabilidad Sanitaria

Resolver esta problemática conlleva múltiples beneficios. En primer lugar, facilita una mayor coordinación y mejora la continuidad en la atención del paciente, al centralizar y hacer accesible la información para todos los profesionales de la salud. Esto se traduce en una mayor transparencia de la información clínica para los pacientes, quienes pueden acceder a sus diagnósticos, resultados de exámenes y recetas independientemente del sistema de salud en el que hayan sido atendidos. Como resultado, se logra una mayor eficiencia por parte de los profesionales al disponer de información completa para realizar recomendaciones más precisas, además de una disminución de costos para los pacientes al evitar la repetición de procedimientos médicos en distintos centros asistenciales.

Además, la interoperabilidad mejora significativamente la calidad de la información disponible, beneficiando tanto la toma de decisiones de los médicos como el área de investigación médica. La capacidad de analizar grandes conjuntos de datos de pacientes puede conducir a avances importantes en la medicina y el tratamiento de enfermedades (Cortez, 2023).

La evolución hacia sistemas de salud más interconectados y eficientes representa un paso fundamental en la modernización del sistema de salud actual hacia un enfoque digital, lo que implica un significativo desarrollo para el país. Este avance promueve mejoras sustanciales en la atención sanitaria, la cual es objeto de críticas en la actualidad. La adopción de tecnologías de interoperabilidad y digitalización no solo optimiza la gestión de la información clínica y facilita una atención más coordinada, sino que también impulsa la eficiencia y calidad del sistema sanitario en general. Este progreso es clave para abordar los desafíos actuales y mejorar la experiencia de los pacientes, posicionando al país en la vanguardia de la innovación en salud.

El tema tratado es complejo para la ingeniería por varios factores. En primer lugar, implica la integración de tecnologías y estándares complejos en un entorno crítico como es el sector de la salud, donde la seguridad y la privacidad de los datos son fundamentales. Además, requiere un profundo conocimiento tanto de la ingeniería de software como de los procesos clínicos y de investigación médica, lo que demanda una colaboración interdisciplinaria entre ingenieros, profesionales de la salud y expertos en normativas sanitarias.

1.5. Objetivos

1.5.1. Objetivo General

Integrar el estándar HL7 FHIR en la plataforma XNAT para facilitar el intercambio seguro de datos de imágenes médicas para fines de investigación.

1.5.2. Objetivos Específicos

- Realizar un levantamiento de los requerimientos técnicos para el uso de XNAT y HL7-FHIR.
- Instalar XNAT en un computador local y realizar pruebas piloto usando algún visualizador de imágenes que permita conexión de red.
- Integrar HL7-FHIR el software XNAT instalado localmente y realizar pruebas piloto usando algún visualizador de imágenes que permita conexión de red.
- Evaluar la plataforma HL7-FHIR-XNAT en un servidor para investigación ubicado en la UTFSM, y realizar pruebas de envío de información desde fuera y dentro del campus.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1. Niveles de Interoperabilidad

Se pueden identificar 4 niveles de interoperabilidad que pueden utilizar las organizaciones según AWS.

2.1.1. Interoperabilidad Básica

La interoperabilidad básica se refiere al nivel más fundamental de intercambio de información entre sistemas informáticos. En este nivel, los sistemas pueden enviar y recibir datos a través de una red, pero no son capaces de interpretar automáticamente esos datos. Esto significa que, aunque los datos pueden transferirse entre sistemas, no hay una comprensión automática del significado de los datos por parte del sistema receptor. Por lo tanto, suele ser necesario que intervenga un ser humano para procesar manualmente los datos y cerrar la brecha de comprensión entre los sistemas.

2.1.2. Interoperabilidad Sintáctica

La interoperabilidad sintáctica, también conocida como estructural, se refiere a la capacidad de proporcionar una estructura o formato de datos que permite su intercambio entre diferentes sistemas. Este enfoque garantiza que los datos puedan ser interpretados por diversos sistemas, permitiéndoles recuperar, interpretar y trabajar con información proveniente de fuentes externas de manera efectiva.(AWS, s.f.)

Durante el desarrollo inicial de esta memoria, estará presente la **interoperabilidad sintáctica**, al utilizar formatos estandarizados para el intercambio de datos entre la plataforma XNAT(explicada en la sección 2.3) y otros sistemas. Dentro de las funcionalidades de esta plataforma, es posible cargar desde un ordenador o portátil, imágenes médicas estandarizadas en el formato DICOM o XML para ser almacenadas y gestionadas dentro del sistema.

2.1.3. Interoperabilidad Semántica

La interoperabilidad semántica permite que diferentes sistemas trabajen juntos con una interpretación mínima o nula de los datos. Esto significa que no solo los sistemas entienden la

estructura de los datos, sino que también comparten una comprensión común del significado de esos datos. En otras palabras, los sistemas semánticamente interoperables tienen una comprensión compartida y precisa de lo que representan los datos que intercambian. Este nivel de interoperabilidad es crucial para evitar errores y malentendidos en la interpretación de los datos (Organización Panamericana de la Salud (OPS), 2021).

La **interoperabilidad semántica** es el nivel que busca alcanzar esta memoria mediante la integración del estándar HL7 FHIR a la plataforma XNAT, con el objetivo de permitir una comprensión compartida del significado clínico de los datos intercambiados en las imágenes médicas.

2.1.4. Interoperabilidad Organizacional

La interoperabilidad organizativa facilita que diferentes sistemas puedan intercambiar datos de manera comprensible, sin importar sus diferencias técnicas. Esto implica alinear los sistemas de información en varios aspectos, como sus objetivos, procesos de trabajo y las expectativas específicas de la organización. La interoperabilidad organizativa garantiza una buena gestión de datos y facilita la colaboración e intercambio de información incluso entre diferentes áreas o ubicaciones administrativas.

2.2. Estándares Internacionales de Interoperabilidad

Fundada en 1987, Health Level Seven International es una organización de normalización sin fines de lucro, acreditada por el Instituto Nacional Estadounidense de Estándares (ANSI por sus siglas en inglés), organización que supervisa el desarrollo de estándares para productos, servicios, procesos y sistemas en los Estados Unidos (American National Standards Institute, 2025).

HL7 se dedica a proporcionar un marco integral y normas relacionadas para el intercambio, la integración, la puesta en común y la recuperación de información sanitaria electrónica que respalde la práctica clínica y la gestión, prestación y evaluación de servicios sanitarios. Tiene como misión proporcionar normas que faciliten la interoperabilidad de los datos sanitarios a escala mundial (Health Level Seven, 2007).

Se han creado muchos estándares que ayudan a la transformación digital de los sistemas de salud mundial, como lo son HL7 V2, HL7 V3, HL7 CDA o HL7 FHIR; este último es el más utilizado a nivel internacional en los últimos cinco años (Mar Cornelio & Wong Pérez, 2023) y será de utilidad para la creación de nuestro sistema.

2.2.1. HL7 FHIR

Recursos Rápidos de Interoperabilidad en Salud (FHIR, por sus siglas en inglés) es un estándar desarrollado por HL7 en 2011 que ha experimentado una rápida evolución. Este estándar combina las mejores características de HL7 v2, HL7 v3 y HL7 CDA, los cuales son estándares dedicados al intercambio de mensajes y documentos clínicos completos, facilitando así su implementación. El objetivo principal de FHIR es promover una interoperabilidad más efectiva y moderna al permitir un intercambio seguro y eficiente de datos clínicos entre sistemas y aplicaciones en el ámbito de la salud (Health Level Seven, 2011).



Figura 4: Logo HL7 FHIR.

Fuente: HL7.

FHIR describe formatos y elementos de datos conocidos como '*recursos*', así como interfaces de programación de aplicaciones (APIs), diseñadas específicamente para facilitar el intercambio de diversos tipos de datos de salud, que incluyen información clínica, diagnóstica, medicamentos, flujos de trabajo y aspectos financieros relacionados con la atención médica.

Los modelos de datos llamados recursos son la pieza clave de este estándar y representan cualquier concepto de la realidad asistencial, como pacientes, doctores, organizaciones o resultados de pruebas. Esta información se puede representar tanto en formato XML como en formato JSON teniendo varias características comunes (Ver figura5):

- Una URL que identifica el dato que desea intercambiarse.
- Metadatos propios de la estructura utilizada.
- Un resumen escrito en lenguaje natural para la fácil comprensión del usuario que lo va a utilizar (profesionales de la salud)
- Información básica y clínica del paciente



Figura 5: Ejemplo de Recurso: Paciente.

Fuente: HL7 FHIR

La principal ventaja de FHIR es su capacidad para incluir un campo denominado 'extensión', diseñado para incorporar datos que no están cubiertos por los elementos estándar de FHIR. Esto proporciona una mayor flexibilidad y adaptabilidad del estándar a diversas necesidades y contextos de implementación en el ámbito de la atención médica.

Otro beneficio del uso de FHIR radica en la facilidad para intercambiar datos entre instituciones de salud. Al emplear tecnologías como XML o JSON, el proceso de transferencia de datos es más rápido y sencillo debido a la estructura específica que estas tecnologías ofrecen. A diferencia de otros estándares, no es necesario enviar grandes cantidades de información en bloque; los datos pueden utilizarse de manera independiente y modular, facilitando su manejo y adaptación a diferentes contextos de implementación en el ámbito sanitario.

Además, FHIR se destaca por su facilidad de implementación gracias al uso de tecnologías comunes y familiares para los programadores, como HTTP, APIs RESTful, JSON y XML. Esta familiaridad con tecnologías establecidas facilita la adopción e integración de FHIR en sistemas existentes, permitiendo a los desarrolladores aprovechar herramientas y habilidades que ya dominan. Esta compatibilidad con tecnologías comunes reduce la curva de aprendizaje y acelera el desarrollo de soluciones interoperables en el sector de la salud, promoviendo así una implementación más rápida y eficiente de FHIR en diversos entornos médicos y tecnológicos.

2.3. XNAT

XNAT es una plataforma de informática en imagenología de código abierto desarrollada por el grupo de investigación en neuroinformática de la Universidad de Washington. Esta plataforma simplifica tareas comunes de gestión, productividad y control de calidad para imágenes y datos asociados. Gracias a su capacidad de extensión, XNAT puede ser utilizado para respaldar una amplia gama de proyectos basados en imágenes. Entre las principales funcionalidades que ofrece este sistema PACS se encuentran: a) carga de datos, organización y compartición de datos, b) visualización y descarga de datos, c) seguridad y gestión de datos, d) búsqueda y exploración de grandes conjuntos de datos, e) y por último, ejecución de procesamiento complejos de los datos utilizando computación de alto rendimiento (Neuroinformatics Research Group, 2017).

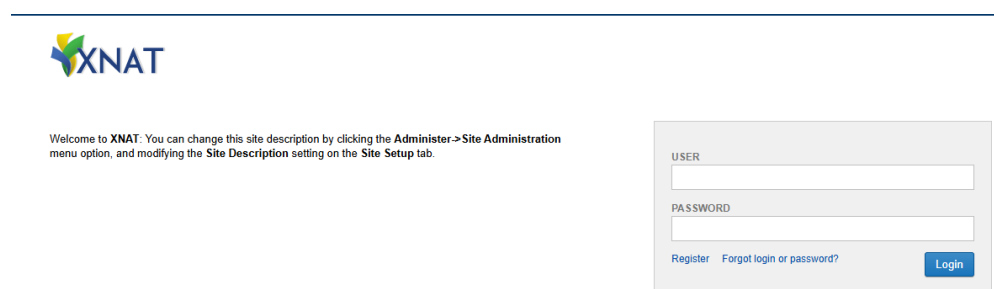


Figura 6: Inicio Plataforma XNAT.
Fuente: Elaboración Propia.

2.3.1. Integración de HL7 FHIR a XNAT

Como se mencionó en la sección 2.3, la capacidad de extensión que ofrece XNAT, ha motivado a la comunidad a realizar complementos que enriquecen el desarrollo de la plataforma. Uno de esos es la integración del estándar HL7 FHIR. Un equipo liderado por Maryna Khvastova logró desarrollar el complemento `xnat-fhir-plugin` que mejora la interoperabilidad y gestión de datos médicos en la plataforma, adaptándolos a los principios del estándar HL7 FHIR, ampliamente utilizado en el área de la salud para realizar un intercambio seguro y estructurado de la información.

El complemento agrega una nueva opción a la interfaz de los experimentos cargados dentro de la plataforma, permitiendo completar un formulario que estandariza la información del paciente según el modelo de recurso HL7 FHIR, mapeando y estructurando los datos ingresados en el formato XML que ya permite XNAT, facilitando así su interoperabilidad al tener la información estandarizada tanto con DICOM y HL7 FHIR para los usuarios que tengan acceso al proyecto (Khvastova et al., 2020).

[PROJECT.EP1](#) > [SJ2](#)

Patient Information

Project: PP1

Subject: XNAT_S00003

Patient identifier

Add identifier

Active

Is record active in use? ☒ yes ☐ no

Patient name

Add Name

The screenshot shows a web form for patient information. At the top, there's a red trash icon. Below it, the 'Use' field is a dropdown menu with 'SELECT' selected. The 'Full name' field is a text input with 'Full name' as a placeholder. Below that are three required fields: 'Given *' with 'Given name' placeholder, 'Prefix *' with 'Prefix' placeholder, and 'Suffix *' with 'Suffix' placeholder. The 'Period' section has two rows: 'Start' and 'End'. Each row contains three dropdown menus, all with '(SELECT)' selected.

Figura 7: Pequeño extracto del formulario del complemento `xnat-fhir-plugin`.
Fuente: Elaboración Propia.

2.4. Virtualización

La virtualización es una tecnología que permite crear entornos informáticos simulados en lugar de físicos, optimizando el uso del hardware y mejorando la eficiencia de los recursos. A través de una capa de abstracción sobre el hardware, el software de virtualización divide los componentes de una sola máquina física, como procesadores, memoria y almacenamiento, en múltiples máquinas virtuales (VM). Cada VM opera de manera independiente, ejecutando su propio sistema operativo y aplicaciones, como si se tratara de un equipo completamente separado. Esto permite a las organizaciones maximizar el aprovechamiento de sus servidores, reducir costos operativos y aumentar la flexibilidad en la gestión de infraestructura. Además, la virtualización es la base de la computación en la nube, ya que facilita la distribución dinámica de recursos y la escalabilidad de los servicios tecnológicos, contribuyendo a una administración más eficaz y adaptable de los sistemas informáticos.

2.4.1. Virtualización de Hardware

La virtualización de hardware, también conocida como virtualización de servidores, es una de las tecnologías más utilizadas en entornos empresariales. Permite la abstracción y distribución dinámica de los recursos físicos, como la memoria RAM, el almacenamiento y la CPU, entre múltiples máquinas virtuales. Cada una de estas VM puede ejecutar distintos sistemas operativos y aplicaciones de manera independiente, optimizando el uso del hardware y garantizando un aislamiento eficiente dentro de un mismo servidor físico con un bajo costo.

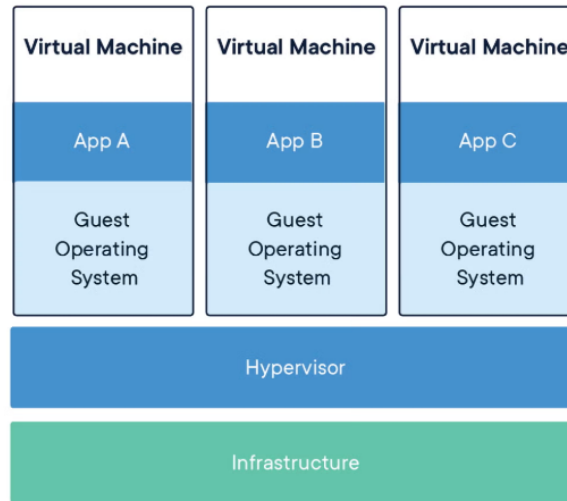


Figura 8: Diagrama de funcionamiento de una VM.

Fuente: Sitio web de Docker.

Como se muestra en el diagrama de funcionamiento de una máquina virtual (Ver Figura 8), la estructura se compone de diversas capas. En la base se encuentra la infraestructura que puede corresponder tanto al hardware físico (servidores locales) o infraestructura virtual como servicio en la nube. Sobre esta capa se sitúa el hipervisor, un software especializado encargado de gestionar la creación y ejecución de las máquinas virtuales. El hipervisor administra los recursos del sistema físico (CPU, memoria RAM, almacenamiento, entre otros) y los abstrae para que puedan ser asignados de manera independiente a cada entorno virtual. Finalmente, en la capa superior se encuentran las máquinas virtuales, cada una de las cuales ejecuta su propio sistema operativo y aloja sus respectivas aplicaciones, operando como si se tratara de un sistema físico autónomo, gracias al aislamiento proporcionado por el hipervisor.

2.4.2. Virtualización basada en contenedores

La virtualización basada en contenedores es un enfoque de virtualización que permite ejecutar aplicaciones en un entorno aislado, donde cada aplicación y sus dependencias se empaquetan en contenedores que comparten el mismo núcleo del sistema operativo subyacente.

A diferencia de las máquinas virtuales completas que requieren un hipervisor para ejecutar diferentes sistemas operativos, los contenedores utilizan el sistema operativo del host para llevar a cabo sus operaciones.

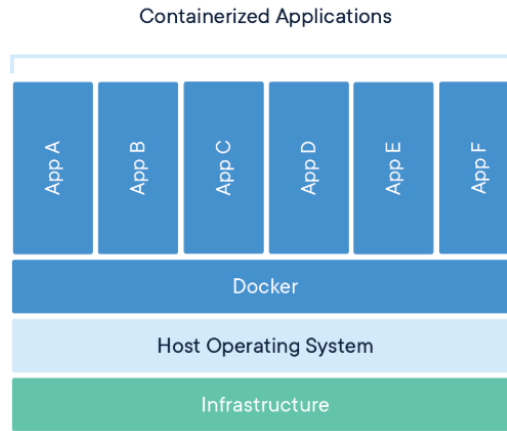


Figura 9: Diagrama de funcionamiento de Contenedores.
Fuente: Sitio web de Docker.

Como se muestra en el diagrama de funcionamiento de los contenedores (Ver figura 9), su estructura es conceptualmente similar a la de una máquina virtual, aunque presenta diferencias clave. En la base se encuentra la infraestructura, que puede ser tanto física como virtual, al igual que en el caso de las máquinas virtuales. Sobre esta capa se sitúa el sistema operativo del host, el cual es compartido por todos los contenedores que se ejecutan en el sistema. A continuación, se encuentra el motor de contenedores (Docker, en este caso), que se instala sobre el sistema operativo y se encarga de la creación, ejecución y gestión de los contenedores. Este motor permite que todos los contenedores compartan el mismo núcleo del sistema operativo, lo cual reduce significativamente el consumo de recursos en comparación con las máquinas virtuales tradicionales. Finalmente, en la capa superior se encuentran las aplicaciones contenerizadas, cada una ejecutándose dentro de su propio contenedor, junto con sus dependencias, librerías y código, en entornos aislados a nivel de proceso.

2.4.3. Comparación entre Contenedores y Máquinas Virtuales

Tanto los contenedores como las máquinas virtuales ofrecen ventajas en términos de aislamiento y asignación de recursos, sin embargo, su funcionamiento difiere significativamente.

Los contenedores operan como una abstracción de la capa de aplicación, empaquetando el código junto con sus dependencias. Varios contenedores pueden ejecutarse simultáneamente en la misma máquina, compartiendo el núcleo del sistema operativo mientras se ejecutan como procesos aislados en el espacio de usuario. Esta arquitectura (Ver figura 9) optimiza el

uso de recursos al reducir el consumo de espacio y mejorar la eficiencia operativa, permitiendo gestionar más aplicaciones con menos máquinas y sistemas operativos.

Por otro lado, las máquinas virtuales representan una abstracción del hardware físico, permitiendo que un único servidor funcione como múltiples servidores independientes. El hipervisor permite ejecutar varias máquinas virtuales en un mismo sistema, cada una de ellas incluye una copia completa de un sistema operativo, la aplicación, los binarios necesarios y las bibliotecas. Debido a esta estructura(Ver figura 8), las VMs requieren un mayor consumo de almacenamiento, alcanzando decenas de gigabytes y presentan tiempos de arranque más prolongados en comparación con los contenedores.

2.5. Estado del Arte

Los países en desarrollo cada vez van optando por integrar el uso de tecnologías móviles como tablets, smartphones o aplicaciones en el área de la salud con el objetivo de fortalecer estos sistemas, mejorar el seguimiento de enfermedades y facilitar el diagnóstico, independiente de la ubicación geográfica del paciente.

También, se está teniendo en cuenta utilizar registros electrónicos para almacenar la información sobre la salud de los pacientes, por ejemplo, los diagnósticos, tratamientos, historias médicas o resultados de pruebas, teniendo como propósito ser accesibles para los profesionales de la salud, facilitando la gestión de la atención, la coordinación de los servicios y la recopilación de los datos para la investigación y mejora del sistema.

Un estudio realizado en los sistemas de salud de Botsuana indica que la implementación de soluciones utilizando tecnologías móviles puede ser significativamente mejorada si se asegura la interoperabilidad entre estas y más aún con sistemas de registro electrónicos, identificando cuatro marcos para guiar el desarrollo de esta, aspectos de infraestructura, estándares de interoperabilidad, seguridad de los datos y la usabilidad. Enfatiza en la necesidad de tener un enfoque centrado en el usuario para diseñar sistemas que faciliten el uso de las tecnologías móviles para que sean accesibles y fácilmente utilizables por los profesionales de la salud, por lo que una integración efectiva de estas TICs puede contribuir en el fortalecimiento de los sistemas de salud y la mejora de resultados en Botsuana y en contextos similares en otros países en desarrollo(Ndlovu et al., 2021).

En enero de 2023, la Oficina del Coordinador Nacional de Tecnología de Información de Salud publicó el progreso en interoperabilidad en hospitales de Estados Unidos entre 2017 y 2021, encontrando hallazgos clave.

La cantidad de hospitales con información electrónica disponible proveniente de fuentes externas del punto de atención creció un 22 %, alcanzando el 62 % en 2021, destacando el aumento considerable en hospitales rurales y pequeños. Casi el 90 % de los hospitales habían actualizado sus tecnologías de Registro Electrónico de Salud(EHR) a la edición certificada del

2015 y el 74 % utilizó tecnología de exportación de datos para análisis y gestión de salud poblacional(Pylypchuk & Everson, 2012).

En el contexto de intercambio de información en salud, existen dos conceptos que se relacionan. Los **Proveedores de servicios de información sanitaria**(HISPs por sus siglas en inglés) son los operadores de redes que permiten el intercambio de datos clínicos a nivel nacional mediante servicios de mensajería seguros. Los HISPs actúan como intermediarios que permiten a diferentes sistemas de EHR enviar, recibir y encontrar información clínica de forma segura y estandarizada(Pylypchuk & Everson, 2012).

También existe el concepto de **Intercambio de Información Sanitaria**(HIE por sus siglas en inglés), que se refiere a las redes de intercambio de información de salud que permiten la transferencia y el acceso de datos clínicos entre distintos proveedores y sistemas(Pylypchuk & Everson, 2012).

En 2021, los HISPs y HIEs fueron los métodos más comunes para enviar y recibir registros de atención e incluso más del 60 % de los hospitales utilizaron HIEs para consultar información de salud externa. Concluyendo en que ha habido un progreso significativo en la interoperabilidad desde 2017 y que es fundamental seguir avanzando en la superación de barreras y la optimización de los sistemas de intercambio de información en salud para lograr mejoras en la atención al paciente(Pylypchuk & Everson, 2012).

Unos de los sectores que enfrenta mayores desafíos en cuanto a interoperabilidad son los laboratorios de neuroimagen. Estos deben gestionar grandes volúmenes de datos, que provienen de diversas fuentes, lo que implica procesos complejos y el uso de protocolos experimentales complicados.

Sin embargo, un estudio publicado en 2012 señala que gran parte de los datos de neuroimagen no se comparten debido a una falta de incentivo para los investigadores, quienes dudan de los beneficios de compartir sus hallazgos por temor a perder reconocimiento o ser superados por la competencia. También por problemas éticos como el respeto a la privacidad u obligaciones legales que pueden perjudicar la confianza con el paciente, más aún si los datos no son manejados con el debido cuidado. Finalmente, la falta de herramientas adecuadas y procedimientos estandarizados dificulta aún más el intercambio de información. Mejorar el acceso y la compartición de datos no solo facilitaría la investigación en diagnóstico y tratamiento de trastornos psiquiátricos y neurológicos, sino que también aceleraría el progreso científico. Lo anterior resalta la importancia de fomentar la colaboración y establecer estándares que optimicen el uso de estos datos en beneficio de toda la comunidad científica y médica, por lo que el artículo señala varias opciones emergentes como lo es XNAT(Poline et al., 2012).

XNAT es un esfuerzo colaborativo de investigadores y académicos, quienes diseñaron esta plataforma de software publicada en 2007, la cual facilita las tareas comunes de gestión y producción de datos, asegurando establecer procedimientos para garantizar la calidad de los datos, proporcionar acceso y almacenamiento seguros para la información, permitir la inte-

gración de datos de diferentes enfoques experimentales en un conjunto de datos unificado y facilitando la colaboración entre investigadores y el acceso público a los datos (Marcus et al., 2007).

XNAT ayuda a mantener la integridad de los datos y mejora la capacidad de los laboratorios para gestionar y compartir la información con la comunidad de neuroimagen a través de las funciones mencionadas. XNAT está diseñado bajo una arquitectura de tres niveles que comprende un backend de base de datos relacional, un middleware basado en java y una interfaz de usuario basada en la web.

El primer nivel se encarga de almacenar los datos no relacionados con imágenes generados a partir de esquemas definidos en XML permitiendo estructurar y organizar los datos de manera eficiente.

El segundo componente tiene un papel intermediario, gestionando interacción entre la base de datos y la interfaz de usuario, facilitando la lógica de negocio y desempeñando un papel crucial en la gestión de las solicitudes de datos.

El último nivel es el encargado de proporcionar herramientas para que los usuarios gestionen grandes volúmenes de datos, accedan a ellos o realicen diversas operaciones mientras se mantiene la calidad y seguridad de estos.

Al distribuirse como una herramienta de código abierto, la comunidad le ha agregado varias mejoras que se pueden implementar como complementos al software.

Una de ellas es la integración del visor **Open Health Imaging Foundation** (OHIF por sus siglas en inglés) en el año 2018 debido a que XNAT carecía de una herramienta de anotación y visualización modernas para imágenes médicas.

El visor OHIF satisface una necesidad que fue en crecimiento en la comunidad de investigación al ofrecer herramientas modernas y flexibles que facilitan un análisis cuantitativo más preciso, logrando convertirse en una herramienta de integración en los sitios de XNAT que en el año 2022 ya tenía más de 3700 descargas (Doran et al., 2022).

Actualmente, XNAT está implementado con estándares que facilitan la gestión de datos clínicos y de investigación. Entre ellos se incluye XSD, utilizado para definir las estructuras de datos dentro de la plataforma, y DICOM, el estándar ampliamente adoptado para el manejo, almacenamiento y transmisión de imágenes médicas.

Además, XNAT emplea una API RESTful que permite la comunicación entre la interfaz de usuario y el backend, facilitando la integración con otros sistemas. También incorpora protocolos de seguridad que garantizan el acceso controlado y la protección de la información sensible.

Es por eso que en el año 2018 se comienza a desarrollar un complemento para implementar una mejora en la interoperabilidad y gestión de datos bajo el estándar HL7 FHIR. Este plugin

permite estandarizar el manejo de datos de pacientes y facilita el intercambio de información entre diferentes sistemas a través de una API REST que permite comunicar y transferir datos de manera eficiente. También logra traducir y mapear los elementos de datos entre los tipos de datos de XNAT y los recursos FHIR. Esto con el fin de permitir un flujo de información más ágil y estructurado entre distintas plataformas y sistemas de salud, desarrollando un paso fundamental hacia la construcción de sistemas de investigación más integrados y eficientes para abordar los desafíos complejos en la salud y medicina actual(Khvastova et al., 2020).

La adaptabilidad de XNAT lo convierte en una herramienta altamente efectiva y versátil para la gestión de datos en el ámbito de la neuroimagen, es por eso que este software ha estado presente en diferentes proyectos de investigación en esta área de la salud. Un ejemplo es **Open Access Series of Imaging Studies**(OASIS) que tiene como objetivo proporcionar conjuntos de datos de neuroimagen de resonancia magnética de acceso abierto y gratuito para la comunidad científica(Marcus et al., 2010).

El proyecto pretende facilitar la investigación de neurociencia básica y clínica al compilar y distribuir de manera pública conjuntos de datos de neuroimágenes. XNAT es fundamental en OASIS debido a su gestión eficaz, acceso e integración de los datos del proyecto, facilitando un entorno colaborativo que fomenta la investigación en neurociencia y contribuyendo al avance del entendimiento de las enfermedades neurodegenerativas y del envejecimiento.

Human Connectome Project es una iniciativa de investigación científica que tiene como objetivo mapear los circuitos neuronales que determinan cómo se conectan diferentes áreas del cerebro y cómo esto se relaciona con el comportamiento y la función cognitiva. Al igual que OASIS, XNAT también ocupa un papel fundamental en este proyecto, ya que proporciona una infraestructura robusta y flexible que facilita la gestión eficaz de grandes volúmenes de datos neurocientíficos debido a que se espera generar aproximadamente 1 petabyte de datos(Marcus et al., 2011).

Por otro lado, permite un flujo de trabajo sistemático, permite el control de calidad y promueve el acceso abierto y la colaboración en la investigación para realizar nuevos descubrimientos en el campo, llevando un registro detallado de los procesos realizados.

Somononetz es otro proyecto que se basa en la infraestructura de XNAT, este es una plataforma de investigación virtual diseñada para facilitar el uso y el análisis de datos de polisomnografía en el ámbito de la investigación sobre trastornos del sueño con el objetivo de proporcionar a los investigadores un entorno colaborativo que soporte el intercambio y procesamiento de datos biosignales relacionados con el sueño(Beier et al., 2015).

Al igual que los proyectos antes mencionados, XNAT representa una solución viable en este proyecto debido a su capacidad para gestionar grandes volúmenes de datos de manera eficiente. Además, proporciona un control de acceso robusto y altos niveles de seguridad en la manipulación de información sensible, como los datos de los pacientes.

Por otro lado, XNAT facilita el análisis mediante la automatización de tareas comunes y per-

mite la estandarización de datos, como los provenientes de estudios de polisonografía, lo cual contribuye a mejorar la calidad de la información y facilita su comparación entre distintos estudios e instituciones.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

El plan de trabajo en que se basa esta solución consta de 2 fases, la primera de ellas es la instalación de XNAT de forma local con el objetivo de familiarizarse con el contenido y uso de la plataforma para luego realizar pruebas piloto. Una vez aprendido todos los detalles se procederá a la segunda fase, relacionada con la instalación de XNAT en un servidor con todas las configuraciones necesarias para tener acceso remoto y realizar pruebas piloto al igual que en la fase local.

3.1. Instalación local de XNAT en Máquina Virtual

Para esta fase de la solución se optará por la creación de una máquina virtual en la cual se le instalará la plataforma con los requisitos previos para su posterior uso y configuración. Se utilizarán sistemas operativos tanto Windows² como Linux³.

3.1.1. Requisitos de Instalación

- VirtualBox (versión 6,1 o superior)
- Git
- Vagrant (versión 2,2 o superior)
- Windows Subsystem for Linux⁴ (en caso de tener Windows como sistema operativo)

3.1.2. Instalación de VirtualBox

VirtualBox es una aplicación de virtualización multiplataforma de código abierto, es decir, extiende las capacidades del ordenador en uso para poder ejecutar varios sistemas operativos, dentro de varias máquinas virtuales, al mismo tiempo.

Para poder descargar esta herramienta, se debe acceder a la página oficial⁵ y seleccionar el paquete correspondiente al sistema operativo del ordenador que se está utilizando:

²Se utilizará Windows 10

³Se utilizará Ubuntu 24,04

⁴Se utilizará Ubuntu 24,04 como distribución de Linux

⁵www.virtualbox.org/wiki/Downloads

- **Para sistemas Windows:** Finalizada la descarga, ejecutar el archivo descargado y seguir las instrucciones del instalador.
- **Para sistemas Linux:** Finalizada la descarga, dirigirse a la ubicación del paquete y abrir una terminal para utilizar el siguiente comando `sudo dpkg -i nombredelarchivo.deb`



Figura 10: Logo VirtualBox.
Fuente: WorldVectorLogo.

3.1.3. Instalación de Git

Git es un sistema de control de versiones distribuido diseñado para rastrear cambios en archivos y gestionar proyectos de desarrollo de software.

Para poder descargar este sistema:

- **En caso de Windows:** Se debe acceder a la página oficial⁶, descargar el archivo y una vez finalizada la descarga, ejecutar el instalador siguiendo los pasos correspondientes del asistente.
- **En caso de Linux:** Solo se debe abrir una terminal y ejecutar el comando `sudo apt-get install git`.



Figura 11: Logo Git.
Fuente: WorldVectorLogo.

3.1.4. Instalación de Vagrant

En caso de estar utilizando Windows como sistema operativo, solo se debe acceder al siguiente link de descarga y seguir los pasos del instalador.

En caso de Linux, se debe abrir una terminal para copiar los siguientes comandos en ella:

⁶www.git-scm.com/downloads/linux

```
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --  
dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg  
  
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.  
gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main"  
| sudo tee /etc/apt/sources.list.d/hashicorp.list  
  
sudo apt update && sudo apt install vagrant
```

Figura 12: Comandos de Instalación de Vagrant.
Fuente: Elaboración Propia



Figura 13: Logo Vagrant.
Fuente: HashiCorp.

3.1.5. Instalación de xnat-vagrant

■ Windows:

Utilizando un sistema operativo Windows, se debe acceder a la distribución de Linux de WSL⁷ en la ubicación en que se desea clonar el repositorio para la instalación de la plataforma e ingresar el siguiente comando:

```
git clone --branch master https://bitbucket.org/xnatdev/xnat-  
vagrant.git && cd xnat-vagrant
```

Figura 14: Comando para clonar repositorio utilizando WSL.
Fuente: Elaboración Propia

Una vez clonado el repositorio, se debe acceder a la carpeta ubicada en `configs/xnat-release` y copiar el archivo `sample.local.yaml` que contiene las configuraciones de la máquina virtual:

```
cd configs/xnat-release  
cp sample.local.yaml local.yaml
```

Figura 15: Comando para crear archivo local.yaml en WSL.
Fuente: Elaboración Propia

Luego de haber creado el nuevo archivo con las configuraciones se le debe agregar la

⁷Es necesario realizar el comando en esta terminal ya que si se hace desde otra, arrojará errores más adelante

siguiente línea correspondiente a la ram recomendada, `ram: 8192` usando el editor de texto que más acomode:

```
nano local.yaml
```

Figura 16: Comando para acceder a local.yaml.

Fuente: Elaboración Propia

Finalmente, abrir una terminal en `git` y ejecutar el siguiente comando: `./setup.sh` para crear la máquina virtual con todo lo necesario para utilizar XNAT.

■ **Linux:**

No es mucha la diferencia de los pasos a realizar utilizando Linux como sistema operativo. Lo primero que se debe hacer es abrir una terminal en la ubicación dónde se desea clonar el repositorio para la instalación de la plataforma e ingresar el siguiente comando:

```
git clone --branch master https://bitbucket.org/xnatdev/xnat-vagrant.git && cd xnat-vagrant
```

Figura 17: Comando para clonar repositorio utilizando Bash.

Fuente: Elaboración Propia

Terminado el proceso de clonación del repositorio que contiene el instalador de la plataforma, se debe acceder a la carpeta ubicada en `configs/xnat-release` y copiar el archivo `sample.local.yaml` que contiene ejemplos de las configuraciones de la máquina virtual para una personalización:

```
cd configs/xnat-release  
cp sample.local.yaml local.yaml
```

Figura 18: Comando para clonar repositorio utilizando Bash.

Fuente: Elaboración Propia

Luego de haber creado el nuevo archivo (`local.yaml` en la Figura 18) con las configuraciones se le debe agregar la siguiente línea de comandos correspondiente a la ram recomendada, `ram: 8192` usando el editor de texto que más acomode:

```
nano local.yaml
```

Figura 19: Comando para editar local.yaml.

Fuente: Elaboración Propia

Luego de haber editado el archivo `local.yaml`, ingresar el comando `./setup.sh` para crear la máquina virtual con todo lo necesario para utilizar XNAT.

Una vez creada la máquina virtual con `xnat-vagrant` ya podemos acceder a la plataforma a través del enlace web `release.xnat.org` o utilizando la dirección IP especificada en el archivo de configuraciones (192.168.56.101 por defecto). Las credenciales de usuario para ingresar son:

- **Usuario:** admin
- **Contraseña:** admin

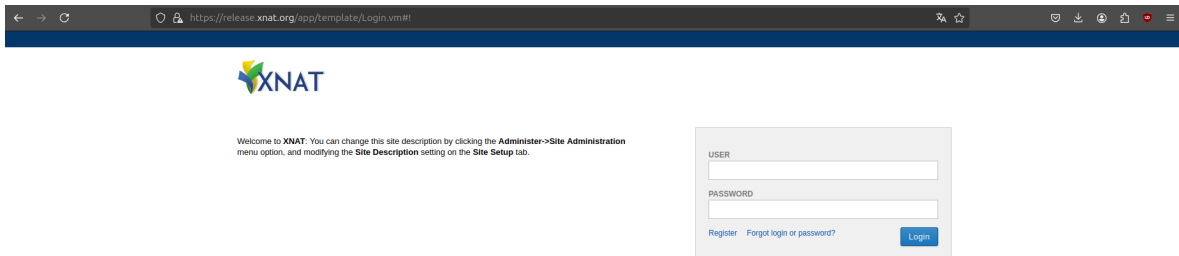


Figura 20: Plataforma XNAT.
Fuente: Elaboración Propia.

3.2. Configuraciones adicionales para `xnat-vagrant`

3.2.1. Cambio de contraseña de usuarios

Si se requiere cambiar las contraseñas de los usuarios existentes en la máquina virtual creada (como `vagrant` y `xnat`), es posible acceder a ella utilizando los comandos proporcionados por Vagrant. Una vez dentro de la máquina virtual, se pueden modificar las contraseñas siguiendo los pasos correspondientes. Para ello, se deben ejecutar los siguientes comandos:

```
vagrant ssh
```

Figura 21: Comando Vagrant SSH.
Fuente: Elaboración Propia

Ya dentro de la máquina virtual, se debe acceder a una sesión de superusuario interactivo con el comando `sudo -i`, para cambiar la contraseña del usuario que se desea:

```
passwd xnat
New password:
Retype new password
```

Figura 22: Cambio de contraseña usuario xnat

Fuente: Elaboración Propia

```
passwd vagrant
New password:
Retype new password:
```

Figura 23: Cambio de contraseña usuario vagrant

Fuente: Elaboración Propia

3.2.2. Tomcat

La máquina virtual se configura con todos los prerequisites necesarios para garantizar el correcto funcionamiento de la plataforma web de XNAT. Sin embargo, al apagarse la máquina virtual, el servicio de Tomcat, responsable de servir los archivos de la página web, también se detiene. Por este motivo, es fundamental configurar el inicio automático de este servicio al encender la máquina virtual.

Para realizar esta configuración, es necesario acceder a la máquina virtual mediante SSH. Esto se puede lograr utilizando los comandos proporcionados por Vagrant o los comandos estándar de SSH:

■ Comandos Vagrant:

```
vagrant ssh
```

Figura 24: Acceso a máquina virtual con comandos Vagrant.

Fuente: Elaboración Propia

■ Comandos básicos SSH:

```
ssh vagrant@192.168.56.101
vagrant@192.168.56.101's password:
```

Figura 25: Acceso a máquina virtual con comandos SSH.

Fuente: Elaboración Propia

Ya dentro de la máquina virtual se debe utilizar el siguiente comando para habilitar el Tomcat con inicio automático:

```
sudo systemctl enable tomcat9
```

Figura 26: Comando inicio automático Tomcat.
Fuente: Elaboración Propia

Para verificar el cambio realizado dentro de los servicios de arranque de la máquina virtual se puede utilizar el siguiente comando `systemctl is-enabled tomcat9` y esperar que la salida sea `enabled`.

3.2.3. Plugins

XNAT es una plataforma altamente personalizable que permite adaptarse a las necesidades específicas del usuario mediante la integración de una amplia variedad de plugins. Estos complementos, disponibles en el mercado oficial de XNAT, amplían significativamente las funcionalidades de la plataforma. Una vez descargado el plugin deseado en el sistema local, es necesario transferir el archivo correspondiente al directorio de plugins en la máquina virtual asociada al usuario `xnat`. Esto se realiza ejecutando los siguientes comandos desde el directorio donde se descargó el plugin:

```
scp *.jar xnat@192.168.56.101:/data/xnat/home/plugins  
vagrant reload
```

Figura 27: Instalación de plugin.
Fuente: Elaboración Propia

Cabe destacar, como se observa en la Figura 27, que es necesario utilizar el comando `vagrant reload` para reiniciar la máquina virtual si el proceso se llevó a cabo mientras esta permanecía encendida. Esto asegura que el plugin se instale correctamente y permite verificar los cambios reflejados en la plataforma.

Es importante señalar que el carácter comodín `*` utilizado en el comando puede reemplazarse por el nombre específico del archivo que se desea transferir. De no hacerlo, se debe tener precaución para evitar que otro archivo `.jar` presente en el mismo directorio sea transferido a la máquina virtual de forma errónea.

3.2.4. Servidor SMTP

XNAT ofrece varias funcionalidades que requieren el envío de correos electrónicos a los usuarios, como las notificaciones de invitaciones a proyectos o la finalización de procesos de análisis de datos. Sin embargo, el sistema no incluye un servidor SMTP integrado. En estos casos,

se puede utilizar un servidor de correo electrónico proporcionado por su organización o institución. Alternativamente, es posible configurar una cuenta de Gmail válida para cumplir con este propósito.

La cuenta de Gmail en general tienen una configuración de seguridad que impide que las aplicaciones menos seguras utilicen los servicios asociados a la cuenta, por lo que necesitamos activar esta opción en primer lugar, luego se debe crear una *Contraseña de Aplicación* que se utilizará para en los valores descritos en la tabla 1 y en un comando más adelante.

Una vez creada la *Contraseña de Aplicación*, hay que configurar Postfix⁸ de la máquina virtual, por lo que debemos acceder a ella vía SSH (Ver figura 24 o 25) e ingresar el siguiente código con las credenciales de la cuenta Gmail a utilizar:

```
configure - postfix --server smtp.gmail.com --port 587 --username "
username123@gmail.com" --host xnat.org
Please enter the password or token for the SMTP server: #
Contraseña de Aplicación
```

Figura 28: Configuración de Postfix.

Fuente: Elaboración Propia

Para configurar correctamente el servidor SMTP de Gmail en XNAT ya realizadas las configuraciones previas, se debe iniciar sesión en la plataforma web con credenciales de administrador. En la barra de navegación, dirigirse a **Administer>Site Administration**, seleccionar la opción **Email Server** y completar los campos con los siguientes valores:

Tabla 1: Parámetros de configuración del servidor SMTP Gmail.

Fuente: Elaboración Propia.

Configuración	Valor
Enable SMTP	Enabled
Host	smtp.gmail.com
Port	587
Protocol	smtp
SMTP Auth?	Enabled
Username	(Cuenta que tomará el rol de emisor de correos, incluyendo el "@gmail.com")
Password	(Contraseña de Aplicación configurada en la cuenta Gmail)
Start TLS	Enabled
SSL Trust	
Email Prefix	XNAT

Para verificar que todo está funcionando correctamente, se puede acceder desde la barra

⁸Servidor de correos utilizado en sistemas Unix y Linux

de navegación a Adminester>Send Email para enviar un correo entre los usuarios creados. Si llegase a existir un error, debemos acceder a la máquina virtual vía SSH (Ver figura 24 o Figura 25) a la siguiente dirección `sudo nano /data/xnat/home/logs/turbine.log` que detallará la información correspondiente.

3.3. Instalación local de XNAT usando Docker

Para esta opción de instalación de la primera fase, se utilizó solamente el sistema operativo Windows 10 para comprobar su funcionamiento.

3.3.1. Prerrequisitos de Instalación

- Docker
- Postgresql
- Git o WSL(Opcionales⁹)

3.3.2. Instalación de Docker

Docker es un proyecto de código abierto que utiliza virtualización basada en contenedores para ejecutar aplicaciones de manera aislada, portátil y eficiente, sin necesidad de máquinas virtuales completas. Se recomienda instalar Docker Desktop desde la página oficial de Docker¹⁰, una aplicación para Linux, Windows y Mac que permite crear, compartir y ejecutar aplicaciones en contenedores con una interfaz gráfica intuitiva. Facilita la gestión de contenedores, aplicaciones e imágenes directamente desde tu máquina, reduciendo la complejidad de la configuración para centrarse solamente en el desarrollo. Además, maneja automáticamente asignaciones de puertos, problemas del sistema de archivos y otras configuraciones predeterminadas.



Figura 29: Logo Docker.
Fuente: WorldVectorLogo.

⁹Solo serán necesarios para copiar el repositorio de forma más cómoda

¹⁰<https://www.docker.com>

3.3.3. Instalación de PostgreSQL

PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto con más de 35 años de desarrollo activo que le ha valido una sólida reputación por su fiabilidad, robustez de características y rendimiento. Para instalar este sistema, debemos acceder a la página oficial¹¹, seleccionar el sistema operativo de nuestro ordenador y seguir las instrucciones del instalador.



Figura 30: Logo PostgreSQL.
Fuente: Sitio web oficial de PostgreSQL.

3.3.4. Instalación de xnat-docker-compose

Para iniciar este proceso, es necesario clonar el repositorio¹² de XNAT. Para ello, abrimos Git o la distribución de Linux instalada y nos ubicamos en el directorio donde deseamos almacenar los archivos de la plataforma e ingresamos el siguiente comando `git clone https://github.com/NrgXnat/xnat-docker-compose`. Una vez completada la clonación, accedemos a la carpeta generada.

En el interior de ella, se encuentra un archivo que contiene las variables de entorno predeterminadas utilizadas para configurar los parámetros del archivo de Docker. Para personalizar estos valores, debemos copiar el archivo mediante la línea de comandos y modificar los campos según nuestras necesidades. En caso contrario, podemos conservar las configuraciones originales.

```
cp default.env .env
```

Figura 31: Copia de archivo default.env en .env.
Fuente: Elaboración Propia

Una vez completada la configuración de las variables de entorno, es necesario abrir una línea de comandos de PowerShell para ingresar el siguiente comando:

```
docker-compose up -d
```

Figura 32: Comando para iniciar la creación de los contenedores e instalación de XNAT
Fuente: Elaboración Propia

¹¹<https://www.postgresql.org/download/>

¹²<https://github.com/NrgXnat/xnat-docker-compose>

El comando mostrado en la Figura 32 se encarga de descargar las imágenes necesarias, crea los contenedores(xnat-web, xnat-db y xnat-nginx) y configura todos los componentes necesarios para iniciar la plataforma de XNAT. Una vez terminado todo el proceso de instalación, es posible acceder a la aplicación de Docker Desktop para verificar que el contenedor se encuentra en ejecución.

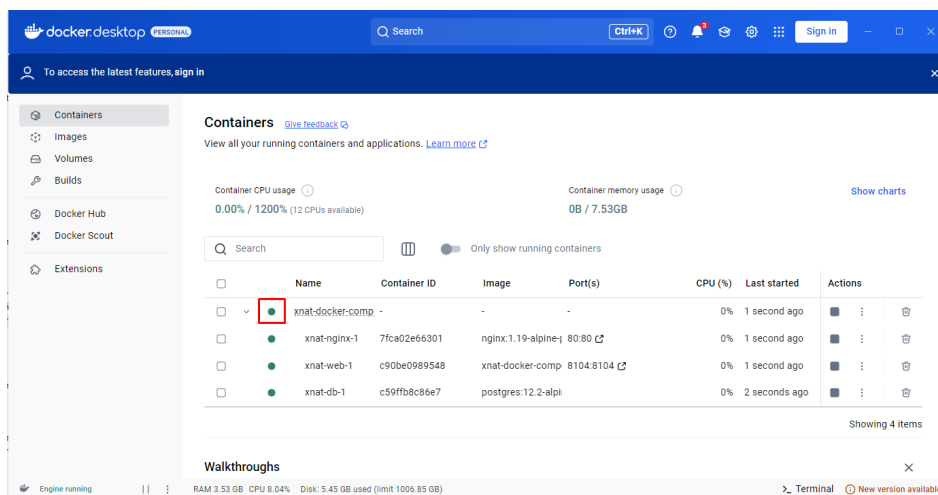


Figura 33: Contenedor de XNAT en ejecución.

Fuente: Elaboración propia.

El círculo verde mostrado en la Figura 33, ubicado a la izquierda de los nombres de los contenedores de la aplicación de Docker Desktop, indica que estos se están ejecutando correctamente y sin errores. A continuación, es posible acceder a la plataforma XNAT abriendo un navegador web e ingresando la URL `http://localhost`.

Es importante tener en cuenta que si la base de datos no ha sido inicializada no se mostrará ningún contenido. Para verificar esto, se deben revisar los registros del contenedor "xnat-web" y buscar la línea que indica `Database initialization complete`.

Como será la primera vez que se inicia XNAT, el sistema solicitará rellenar algunos formularios para terminar con las configuraciones del sitio antes de redirigir al usuario a la página principal¹³. Una vez finalizada esta configuración, se podrá acceder con las credenciales de administrador y comenzar a utilizar la plataforma, las cuales son:

- **Usuario:** admin
- **Contraseña:** admin

¹³Los formularios se detallan en el Anexo A

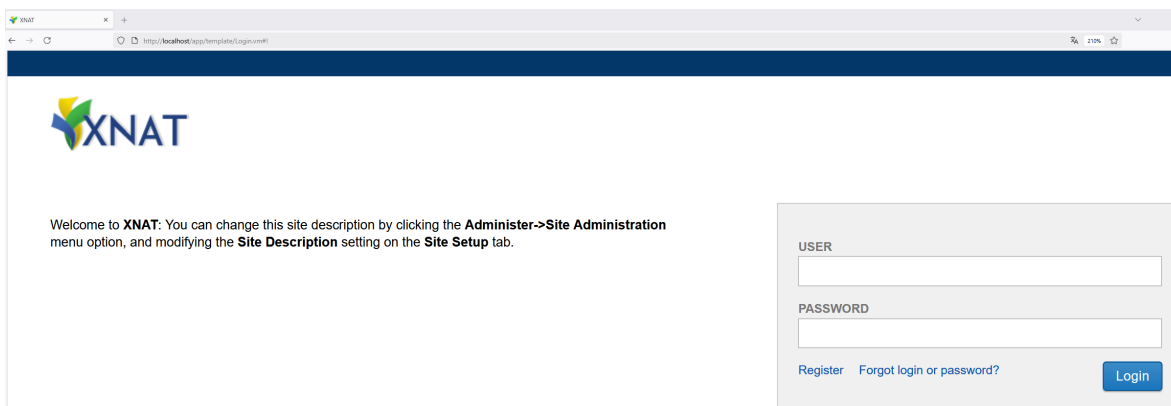


Figura 34: Inicio XNAT Docker.
Fuente: Elaboración propia.

3.3.5. Plugins

Como se menciona en la sección 3.2.3, XNAT Docker también permite la personalización de la plataforma para adaptarse a las necesidades del usuario mediante la instalación de una amplia variedad de complementos disponibles en el mercado oficial de XNAT, lo que amplía sus funcionalidades.

Para instalar un complemento, es necesario transferir los archivos correspondientes al contenedor de Docker `xnat-web`. En primer lugar, se debe descargar el plugin desde el mercado oficial de XNAT. Una vez finalizada la descarga, es fundamental verificar que Docker Desktop esté en ejecución y que los contenedores estén apagados.

A continuación, se debe acceder al directorio en que se ubican los archivos recién descargados, abrir una terminal de PowerShell e ingresar el siguiente comando `docker ps -a`

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7fca02e66301	nginx:1.19-alpine-perl	"/docker-entrypoint..."	46 hours ago	Exited (0) 22 hours ago		xnat-docker-compose-xnat-nginx-1
c90be0989548	xnat-docker-compose-xnat-web	"wait-for-postgres.s..."	46 hours ago	Exited (143) 22 hours ago		xnat-docker-compose-xnat-web-1
c59ffb8c86e7	postgres:12.2-alpine	"docker-entrypoint.s..."	46 hours ago	Exited (0) 22 hours ago		xnat-docker-compose-xnat-db-1

Figura 35: Comando `docker ps -a`.
Fuente: Elaboración propia.

Este comando mostrará la lista de contenedores creados en la aplicación Docker Desktop, lo que permitirá verificar el nombre completo del contenedor `xnat-web` (`xnat-docker-compose-xnat-web-1` como se observa en la Figura 35).

Obteniendo el nombre del contenedor, se puede ejecutar el siguiente comando para transferir el complemento que queremos instalar en la plataforma XNAT:

```
docker cp *.jar xnat-docker-compose-xnat-web-1:/data/xnat/home/  
plugins
```

Figura 36: Comando para transferir plugins al contenedor de Docker
Fuente: Elaboración Propia

Cabe destacar que el * en el comando de la Figura 36 se debe reemplazar por el nombre del archivo .jar que queremos instalar en la plataforma XNAT, ya que, a diferencia de la instalación de complementos utilizando Vagrant descritos en la sección 3.2.3, el carácter comodín * no copiará todos los archivos .jar que se encuentren en el directorio y arrojará un error.

Terminados los pasos anteriores, se pueden encender los contenedores para acceder a la plataforma y verificar si el plugin ha sido instalado y reconocido. Para ello, se debe ingresar a la barra de navegación y seleccionar Administer>Plugin Settings

3.4. xnat-fhir-plugin

Para poder integrar el estándar HL7-FHIR en la plataforma XNAT, es necesario descargar el repositorio de xnat-fhir-plugin desde GitHub, el cual contiene el código fuente y los archivos requeridos para construir el compilado del complemento; sin embargo, antes de realizar este proceso, se deben hacer algunas modificaciones por desactualizaciones de versiones. El archivo generado se deberá transferir al entorno donde se ejecuta XNAT, ya sea el contenedor de Docker o a la máquina virtual, dependiendo del método de despliegue utilizado.

3.4.1. Prerrequisitos para generar el complemento

- Gradle
- WSL (en caso de tener Windows como sistema operativo)

3.4.2. Instalar Gradle

Sistema operativo Windows: Se debe acceder a la página oficial de Gradle¹⁴, seleccionar la opción Binary-only. Una vez terminada la descarga, se debe descomprimir el archivo en la ubicación deseada. Luego se debe agregar la carpeta bin de los archivos descomprimidos a las variables de entorno del sistema, para ello se debe acceder a Variables de

¹⁴<https://gradle.org/install/>

Entorno>Path>Editar>Examinar y buscar la ubicación de la carpeta mencionada anteriormente y guardar los cambios realizados.

Sistema operativo Linux Ubuntu: Se debe abrir una línea de comando y utilizar los siguientes comandos:

```
curl -s "https://get.sdkman.io" | bash
source "$HOME/.sdkman/bin/sdkman-init.sh"
```

Figura 37: Comando para instalar SDKMAN
Fuente: Elaboración Propia

Una vez instalado SDKMAN, se debe ingresar el siguiente comando para instalar Gradle `install gradle`.

3.4.3. Instalar xnat-fhir-plugin

Como se mencionó al principio de la sección 3.4, se debe clonar el repositorio que contiene el código fuente desde GitHub. Para ello se debe abrir una terminal de WSL en caso de tener Windows como sistema operativo o una línea de comando si tenemos Linux como sistema operativo e ingresar el siguiente comando `git clone https://github.com/somnonetz/xnat-fhir-plugin.git`

Una vez descargados los archivos, se debe editar el archivo `build.gradle` ya que contiene algunas desactualizaciones que no funcionan con ciertas versiones de Java o Gradle. El primer fragmento que debemos editar es relacionado a las dependencias, ya que existe un paquete que se eliminó a partir de Java 11, por lo que para funcione independiente de la versión de Java, se debe agregar lo siguiente:

```
dependencies{
    ...
    ...
    implementation 'jakarta.xml.bind:jakarta.xml.bind-api:2.3.3'
    implementation 'org.glassfish.jaxb:jaxb-runtime:2.3.3'
}
```

Figura 38: Paquetes que se deben agregar en el archivo `build.gradle`
Fuente: Elaboración Propia

El siguiente cambio que debemos hacer es en el método `classifier` ya que no existe en versiones superiores a Gradle 8. Dentro de la misma sección donde se utiliza el método, a

partir de Gradle 8, se debe explicitar qué hacer en caso de archivos duplicados. Por lo descrito anteriormente se debe cambiar lo siguiente:

```
task sourcesJar(type: Jar,
    dependsOn: classes) {
    classifier "sources"
    manifest {
        from gitManifest
    }
    from sourceSets.main.
        allSource
}

// This builds a javadoc
// jar that pulls in the
// manifest with VCS info
// from above.
task javadocJar(type: Jar)
{
    classifier "javadoc"
    manifest {
        from gitManifest
    }
    from javadoc.
        destinationDir
}
```

Figura 39: Código original de xnat-fhir-plugin
Fuente: Elaboración Propia

```
task sourcesJar(type: Jar,
    dependsOn: classes) {
    archiveClassifier.set("sources")
    duplicatesStrategy =
        DuplicatesStrategy.
            EXCLUDE
    manifest {
        from gitManifest
    }
    from sourceSets.main.
        allSource
}

// This builds a javadoc
// jar that pulls in the
// manifest with VCS info
// from above.
task javadocJar(type: Jar)
{
    archiveClassifier.set("javadoc")
    manifest {
        from gitManifest
    }
    from javadoc.
        destinationDir
}
```

Figura 40: Cambios que se deben hacer al código de la Figura 39
Fuente: Elaboración Propia

Lo último que debemos hacer es agregar un fragmento de código que le proporcione acceso a Spring para realizar las pruebas del código fuente sin importar la versión de Java que se utilice. Este es el código que se debe agregar al archivo:

```
test {
    useJUnitPlatform()
    jvmArgs += [
        '--add-opens=java.base/java.lang=ALL-UNNAMED',
        '--add-opens=java.base/java.lang.reflect=ALL-UNNAMED',
        '--add-opens=java.base/java.util=ALL-UNNAMED'
    ]
}
```

Figura 41: Código que se debe agregar al final del código en el archivo `build.gradle`
Fuente: Elaboración Propia

Una vez terminado de modificar el código fuente, se debe abrir una terminal para ejecutar el comando `gradle clean xnatPluginJar` que es el encargado de limpiar compilaciones anteriores y generar el archivo empaquetado del complemento. Finalizado el proceso de compilación con éxito, los archivos creados se encontrarán en la carpeta `build/libs`. Estos deberán ser copiados o transferidos al entorno de despliegue utilizado según las instrucciones detalladas en las secciones 3.2.3 o 3.3.5.

3.5. Instalación de XNAT en un servidor

Para esta fase, se optó por implementar la plataforma XNAT mediante el método de despliegue descrito en la sección 3.3, utilizando contenedores de Docker. La instalación se llevó a cabo en un servidor ubicado en la sede San Joaquín de la Universidad Técnica Federico Santa María. La elección de este método fue bajo la necesidad de minimizar el consumo de recursos del sistema, permitiendo así que el servidor mantenga un funcionamiento estable y sin interferir con otras funciones que cumpla este.

Por temas de seguridad del servidor, fue necesario establecer una conexión mediante un túnel SSH doble, primero se accede al servidor intermedio de la UTFSM para detectar un usuario seguro y, desde allí, conectarse al servidor de destino donde se realizará la instalación y descarga de los archivos de la plataforma¹⁵.

```
ssh -t usuarioinf@servidor.universidad ssh usuarioservidor@servidor
    .host
```

Figura 42: Ejemplo de conexión SSH al servidor
Fuente: Elaboración Propia

Antes de establecer la conexión, por motivos de comodidad, se creó un repositorio GitHub

¹⁵En adelante, nos referiremos a este como `servidor host`

con los complementos que se van a instalar en la plataforma más adelante, con el objetivo de dejarlos alocados dentro del servidor host ante cualquier eventualidad.

Luego de establecer la conexión siguiendo el comando descrito en la Figura 42, al ser un servidor con sistema operativo Ubuntu, se solicitó una petición de instalación de Docker debido a la falta de permisos de usuario.

Luego de verificar que estuvieran todos los prerequisites descritos en la sección 3.3.1, se continuó al proceso de despliegue y personalización descritos en las secciones 3.3.4 y 3.3.5 respectivamente, logrando la correcta instalación de la plataforma.

Una vez levantada la plataforma, se configuró un servidor con IP pública y se apuntó un subdominio (`albert-xnat.inf.santiago.usm.cl`) hacia dicha IP mediante los registros DNS correspondientes. En este servidor se instaló Nginx, que actúa como proxy inverso, permitiendo redirigir las solicitudes entrantes al servicio XNAT disponibles en el puerto 80. Para habilitar el acceso mediante HTTPS, se utilizó una herramienta automatizada para la obtención y configuración de certificados digitales, permitiendo conexiones seguras con el dominio, redirigiendo automáticamente el tráfico HTTP hacia HTTPS. Finalmente, se ajustó el parámetro `client_max_body_size` para permitir la carga de archivos grandes.

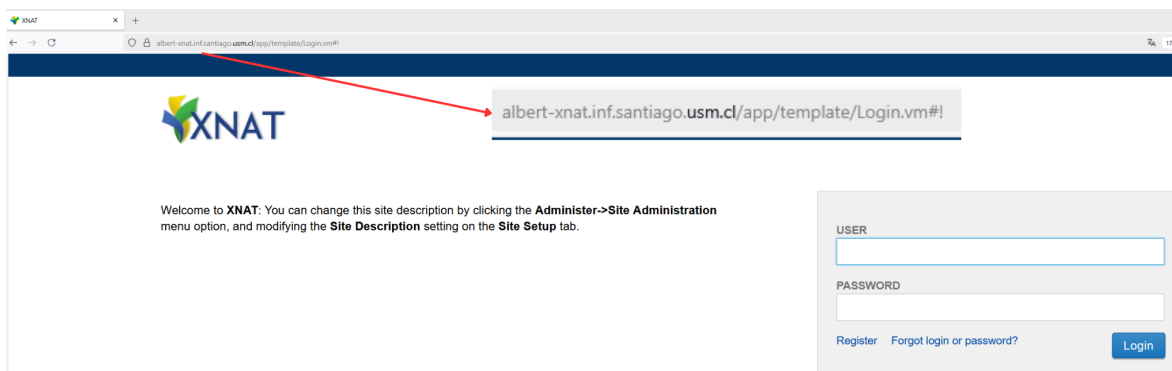


Figura 43: Sitio web `https://albert-xnat.inf.santiago.usm.cl/`
Fuente: Elaboración propia.

Terminado el proceso para exponer la plataforma XNAT a la web, se accedió a la dirección `https://albert-xnat.inf.santiago.usm.cl/` (Ver Figura 43) para la configuración inicial. Se cambió la contraseña de administrador y se configuró el registro de usuario.

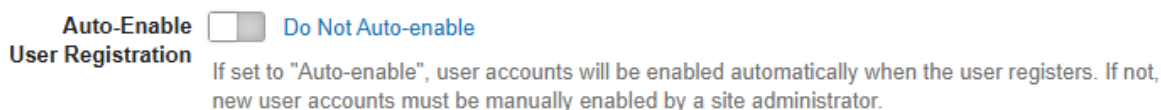


Figura 44: Opción de activación de cuenta de forma automática deshabilitada
Fuente: Elaboración propia.

Como se ve en la Figura 44, se deshabilitó la opción de activación de cuenta de forma automática para que el administrador tenga mayor control, seguridad y trazabilidad de las cuentas de usuario registradas en la plataforma.

Terminando todas las configuraciones descritas anteriormente, la plataforma XNAT quedó accesible desde la dirección web <https://albert-xnat.inf.santiago.usm.cl/> como se ve en la Figura 43, para cualquier usuario interesado, si es que los contenedores de Docker instalados en el servidor host están activos.

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

4.1. Pruebas pilotos de la instalación de XNAT de forma local

Para realizar las pruebas de la fase local de la solución, se optó por utilizar el método `xnat-docker-compose` descrito en la sección 3.3.4 en un ordenador de escritorio con sistema operativo Windows 10 conectado a internet a través de un cable ethernet.

4.1.1. Creación de usuario

Esta prueba consistió en la creación de un usuario desde la página principal de la plataforma, sin embargo, para facilitar el proceso, previamente se personalizaron algunas configuraciones de la sección *Registration Options*, desactivando la verificación por correo electrónico y habilitando la opción de activación automática de cuentas al momento de registro (Ver figura 45).

Registration Options

These settings control auto-enabling and auto-verification of user accounts on registration. Email notifications to site administrators on new user registrations are not affected by these settings.

Require Email Verification ☐ Not Required

If email verification is required (recommended), newly registered users and users who change their email address must verify their email address before their account is verified. Verification happens via a link sent to the submitted email.

Auto-Enable User Registration ☒ Auto-enable

If set to "Auto-enable", user accounts will be enabled automatically when the user registers. If not, new user accounts must be manually enabled by a site administrator.

Auto-Enable With Project Access Request ☒ Auto-enable

If an unregistered user receives a Project Access Request via email, the site administrator could assume that the project owner knows this user and can verify the validity of their account. This setting allows for auto-enabling users who register accounts after receiving a Project Access Request, overriding the Auto-enable setting above.

Allow User Comments On Registration ☒ Allow

User comments, if allowed, are displayed in the email notifications sent to the site administrator.

Discard Changes Save

Figura 45: Opciones de Registro.
Fuente: Elaboración Propia.

Accedimos a la opción *Register* de la página de inicio y completamos los campos requeridos con los siguientes datos para finalizar el registro de usuario (Ver tabla 2):

Tabla 2: Formulario para crear usuario.

Fuente: Elaboración Propia.

New User Registration	
Username	testUser
Password	userTest
Re-enter Password	userTest
First Name	Usuario
Last Name	Prueba
Email	testusr@testemail.com
Phone	12345678
Study Site or Lab	usm
Comments	Prueba piloto de creacion de usuario

Para confirmar que el nuevo usuario fue creado satisfactoriamente y almacenado en la base de datos, verificaremos los archivos del contenedor `xnat-db`. Para ello, abrimos una terminal de PowerShell para ingresar al contenedor y examinamos las tablas de datos utilizando los siguientes comandos(Ver figura 46):

```
docker exec -it xnat-docker-compose-xnat-db-1 bash
bash-5.0# psql -U xnat -d xnat
xnat=> SELECT * FROM xdat_user;
```

Figura 46: Comandos para ingresar a ver la tabla de datos de los usuarios .

Fuente: Elaboración Propia

El comando mostrado en la Figura 46 nos proporcionará una lista de todos los usuarios registrados en la plataforma junto con sus respectivos datos. Las contraseñas están encriptadas, y se utilizan binarios para indicar valores de activo o desactivado de ciertos datos(Ver rectángulo verde de la Figura 47).

Tal como se muestra en el rectángulo rojo de la Figura 47, el usuario creado con los valores de la Tabla 2 ha sido registrado correctamente.

xnat> SELECT * FROM xdat_user;											
login	firstname	lastname	email	primary_password	primary_password_encrypt	quarantine_path	enabled	verified	sa		
user_info	xdat_user_id	users_user_xdat_security_id									
admin	Admin	Admin	harmitage@miskatonic.edu	{bcrypt}\$2a\$10\$mpXf0iw5VDnanaNx3J3swus6ocupbotsdGAVqDhj1AKAv9yEpFVBK	1		1	1			
guest	XNAT	Guest	harmitage@miskatonic.edu	{bcrypt}\$2a\$10\$DcRLbi4Rs8dY6gI4DvAw0uL70qloU3wdPjqk0tYFW09ne84qv3gK	1		1	1			
testUser	Usuario	Prueba	testusr@testemail.com	{bcrypt}\$2a\$10\$8aPn0Z/8I.Si71D56dFwI.XAKTf46vi1shz76D4kF456ir5XC.j46			1	1			

Figura 47: Usuarios de la tabla `xdat_user`.

Fuente: Elaboración Propia.

4.1.2. Acceso a Proyectos

Para esta prueba piloto se crearon 2 proyectos privados, Proyecto Prueba(Se le asignó el identificador EP1) y Test Project(Se le asignó el identificador EP2), cada uno con integrantes distintos, distribuidos de la siguiente manera:

Tabla 3: Integrantes de cada proyecto
Fuente: Elaboración Propia

Usuarios	Proyecto Prueba	Test Project
testUser	✓	✓
User2	✓	
userProject		✓

Como se muestra en la Tabla 3, el usuario testUser pertenece a los 2 proyectos, Proyecto Prueba y Test Project, el usuario User2 pertenece a Proyecto Prueba solamente, al igual que userProject que solo pertenece a Test Project.

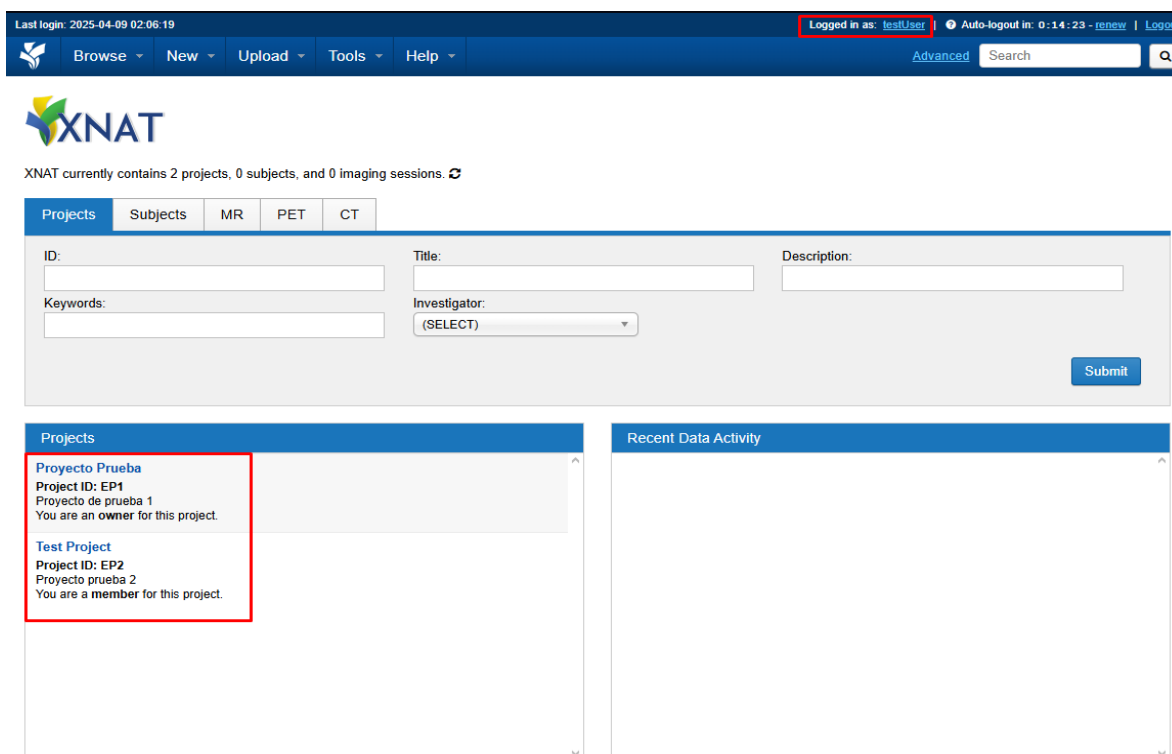


Figura 48: Página principal del usuario testUser.
Fuente: Elaboración Propia.

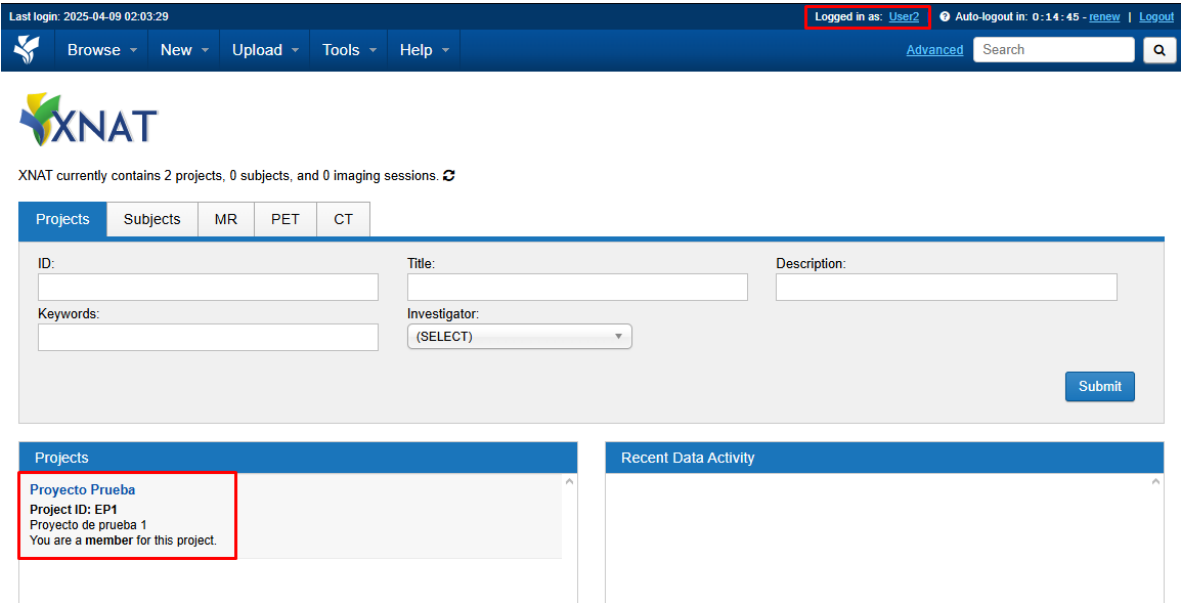


Figura 49: Página principal del usuario User2.
Fuente: Elaboración Propia.

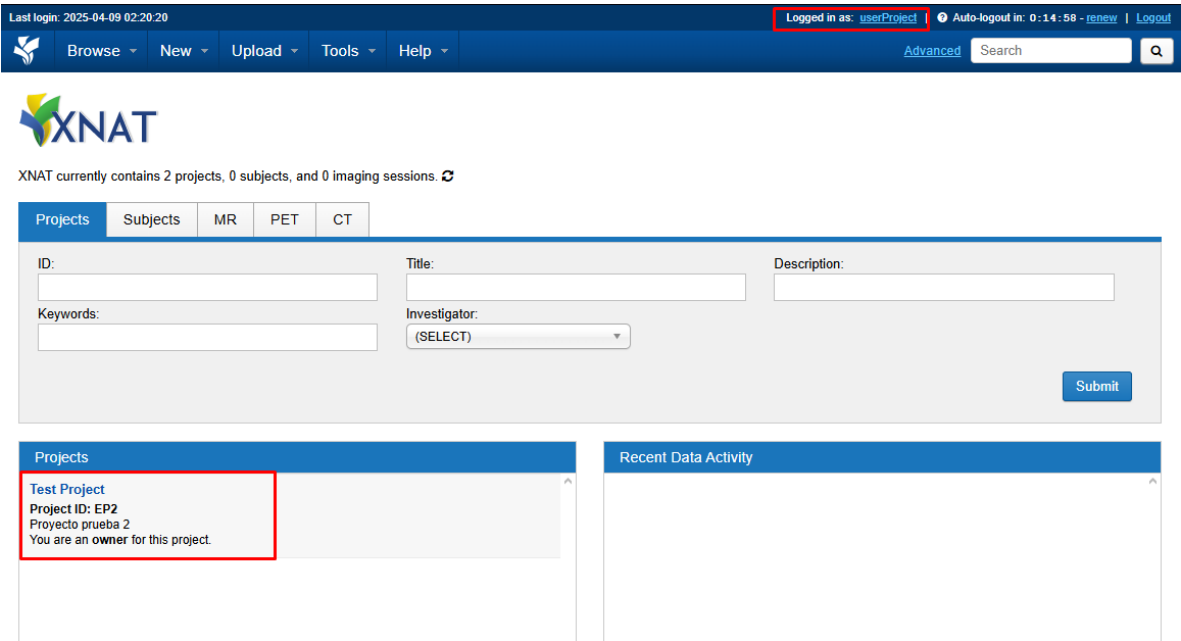


Figura 50: Página principal del usuario userProject.
Fuente: Elaboración Propia.

Por lo que se accedió a las sesiones de cada uno de los usuario listados en la Tabla 3 con el objetivo de verificar los proyectos asociados a cada uno. En la Figura 48, dentro del rectángulo rojo ubicado en la sección Project, se observan los proyectos correspondientes al

usuario testUser: el Proyecto Prueba, en el cual posee el rol de propietario, y el Test Project, donde participa como miembro.

De manera similar, en la Figura 49, el rectángulo rojo en la sección Project muestra que dicho usuario es únicamente miembro del proyecto Proyecto Prueba.

Finalmente, en la Figura 50, se aprecia que el usuario visualizado es propietario exclusivo del proyecto Test Project, como se indica también en el área resaltada en rojo de la sección Project.

Por lo que se verificó que tanto la creación de los proyectos como las relaciones establecidas entre estos y los usuarios del sistema quedaron correctamente registradas en la base de datos, lo cual se logró confirmar satisfactoriamente (ver Figura 51).

```
xnat=> SELECT * FROM xnat_projectdata;
```

name	type	description	keywords	pi_xnat_investigatordata_id	custom_fields	id	secondary_id
Proyecto Prueba		Proyecto de prueba 1			{}	EP1	PP1
Test Project		Proyecto prueba 2			{}	EP2	TP1

(2 rows)

Figura 51: Tabla xnat_projectdata de la base de datos que registra los proyectos de la plataforma

Fuente: Elaboración Propia.

A continuación, se procedió a verificar las relaciones establecidas entre los usuarios y los proyectos dentro de la plataforma. Para ello, se ejecutó la consulta indicada en la Figura 52, la cual permitió obtener exitosamente la información detallada en la Tabla 3, incluyendo los roles asignados a cada usuario dentro de los respectivos proyectos(Ver Figura 53).

```
SELECT
  ug.id AS grupo ,
  u.login AS usuario ,
  p.id AS proyecto
FROM
  xdat_user_groupid ugid
JOIN
  xdat_user u ON ugid.groups_groupid_xdat_user_xdat_user_id = u.
  xdat_user_id
JOIN
  xdat_usergroup ug ON ugid.groupid = ug.id
JOIN
  xnat_projectdata p ON ug.id LIKE p.id || '_%'
ORDER BY p.id;
```

Figura 52: Consulta para obtener las relación de los usuarios con los proyectos.

Fuente: Elaboración Propia

grupo	usuario	proyecto
EP1_owner	testUser	EP1
EP1_member	User2	EP1
EP2_owner	userProject	EP2
EP2_member	testUser	EP2
(4 rows)		

Figura 53: Resultados de la consulta en la base de datos que muestra las relaciones de los usuarios con los proyectos del sistema.

Fuente: Elaboración Propia.

Lo anterior demuestra que cada usuario solo puede visualizar y acceder a los proyectos a los cuales pertenece, respetando así los niveles de acceso definidos en la plataforma.

4.1.3. Acceso a Plataforma desde otro dispositivo

Para esta prueba, se utilizó un notebook con sistema operativo Linux Ubuntu 24.04 conectado vía Wi-fi a la misma red local que el ordenador de escritorio donde se encuentra instalado el contenedor de la plataforma XNAT.

Desde el navegador web del portátil, se accedió a la dirección IP del ordenador escritorio que aloja el contenedor con la plataforma XNAT. En este caso, la dirección corresponde a 192.16X.XXX.XXX¹⁶.

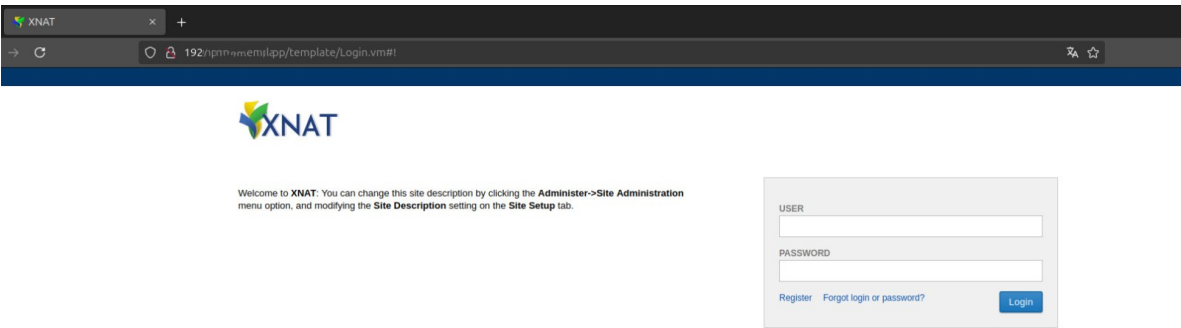


Figura 54: Página principal desde el portátil.

Fuente: Elaboración Propia.

En la Figura 54, se observa que se logró acceder correctamente a la plataforma XNAT del contenedor alojado el ordenador de escritorio. A continuación, se procedió a iniciar sesión

¹⁶Por temas de seguridad se ocultarán 7 dígitos de la dirección IP

como el usuario `testUser` con el objetivo de verificar que la información creada y organizada desde el ordenador de escritorio se mantiene al acceder desde otro dispositivo.

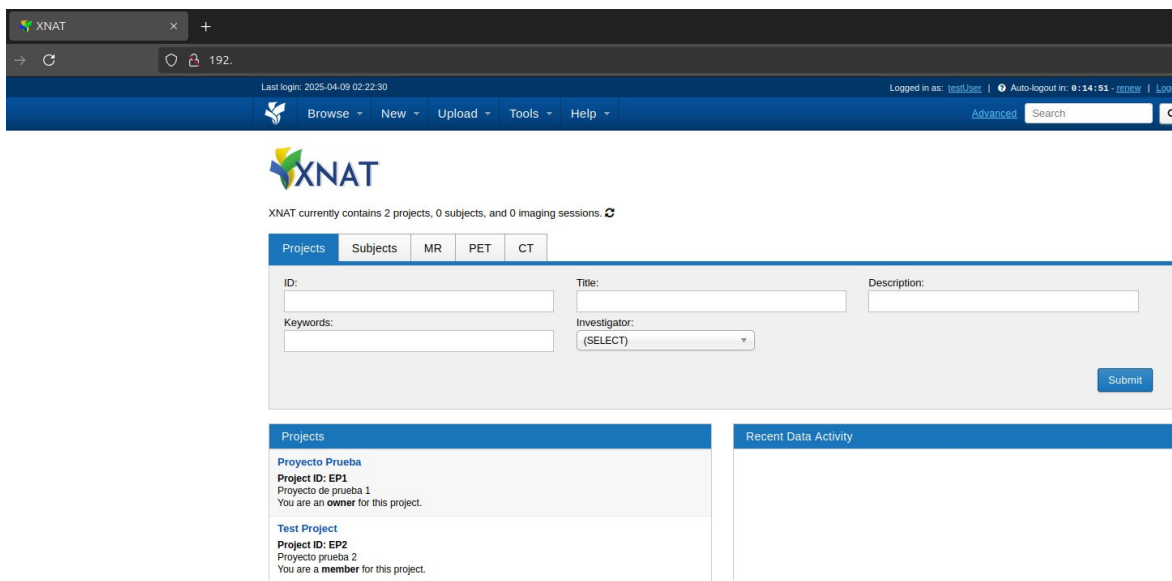


Figura 55: Página principal del usuario `userProject` desde el portátil.
Fuente: Elaboración Propia.

En la Figura 55, se observa que se logró acceder exitosamente también a la sesión del usuario `testUser1`, donde se visualizan todos los proyectos a los que tiene acceso, lo cual confirma la correcta persistencia y disponibilidad de los datos en la plataforma al ser consultados desde otro dispositivo(dentro de la misma red en este caso).

4.1.4. Subir una sesión de imágenes médicas

Para esta prueba piloto, se accedió a la sesión del usuario `userTest` desde el ordenador de escritorio en el que se encuentra instalado el contenedor con la plataforma XNAT. Durante la sesión, se procedió a cargar un conjunto completo de imágenes correspondientes a una sesión de resonancia magnética en el proyecto Proyecto Prueba. Para facilitar la posterior visualización de las imágenes, se instaló el complemento `ohif-viewer-3.7.0-XNAT-1.8.10` en el contenedor `xnat-web`.

La plataforma logró desempaquetar y almacenar correctamente todos los archivos contenidos en el archivo comprimido cargado en el directorio `/data/xnat/archive/EP1/arc001/anon100707BXC002_00001/SCANS` del contenedor `xnat-web`, organizándolos automáticamente en asuntos dentro de Proyecto Prueba lo que permite una gestión más ordenada y eficiente de los datos.

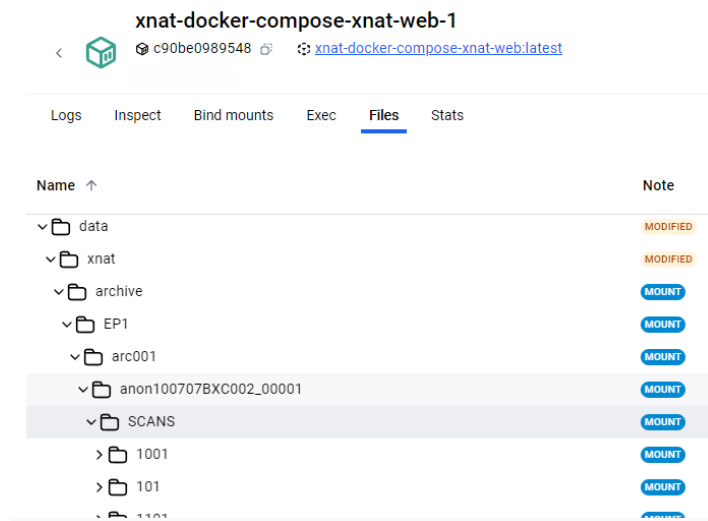


Figura 56: Imágenes almacenadas en el contenedor `xnat-web` al ser cargadas desde la plataforma web.

Fuente: Elaboración Propia.

Luego, se verificó la correcta visualización de las imágenes accediendo a Proyecto Prueba>Subjects>anon100707BXC002_00001>Actions>View Images, lo cual abrió una pestaña nueva en el navegador con el visor instalado, permitiendo explorar las imágenes médicas de manera óptima(Ver Figura 57).

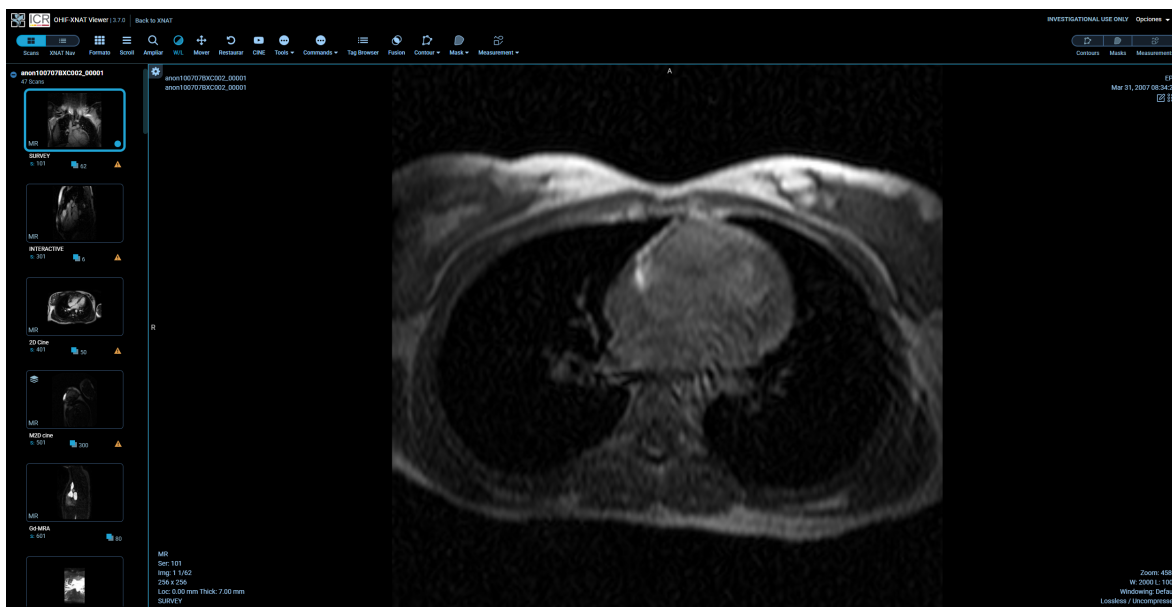


Figura 57: Visualizador OHIF con las imágenes cargadas en la plataforma XNAT.

Fuente: Elaboración Propia.

Una vez verificado el correcto funcionamiento de la carga y visualización de imágenes, se procedió a ingresar a la sesión del usuario User2 desde otro dispositivo, con el objetivo de confirmar que este usuario, teniendo el rol miembro del Proyecto Prueba, también tuviera acceso tanto a las imágenes almacenadas como al visualizador previamente instalado.

Para ello, se utilizó el mismo portátil descrito en la sección 4.1.3. Tras iniciar sesión con las credenciales del usuario User2, se accedió al proyecto mencionado, donde se pudo constatar que todas las imágenes previamente cargadas por el usuario userTest estaban disponibles(Ver Figura 58).Asimismo, se comprobó que al acceder al visualizador mediante la opción correspondiente, se podían visualizar y manipular correctamente todas las imágenes cargadas en la plataforma(Ver Figura 59).

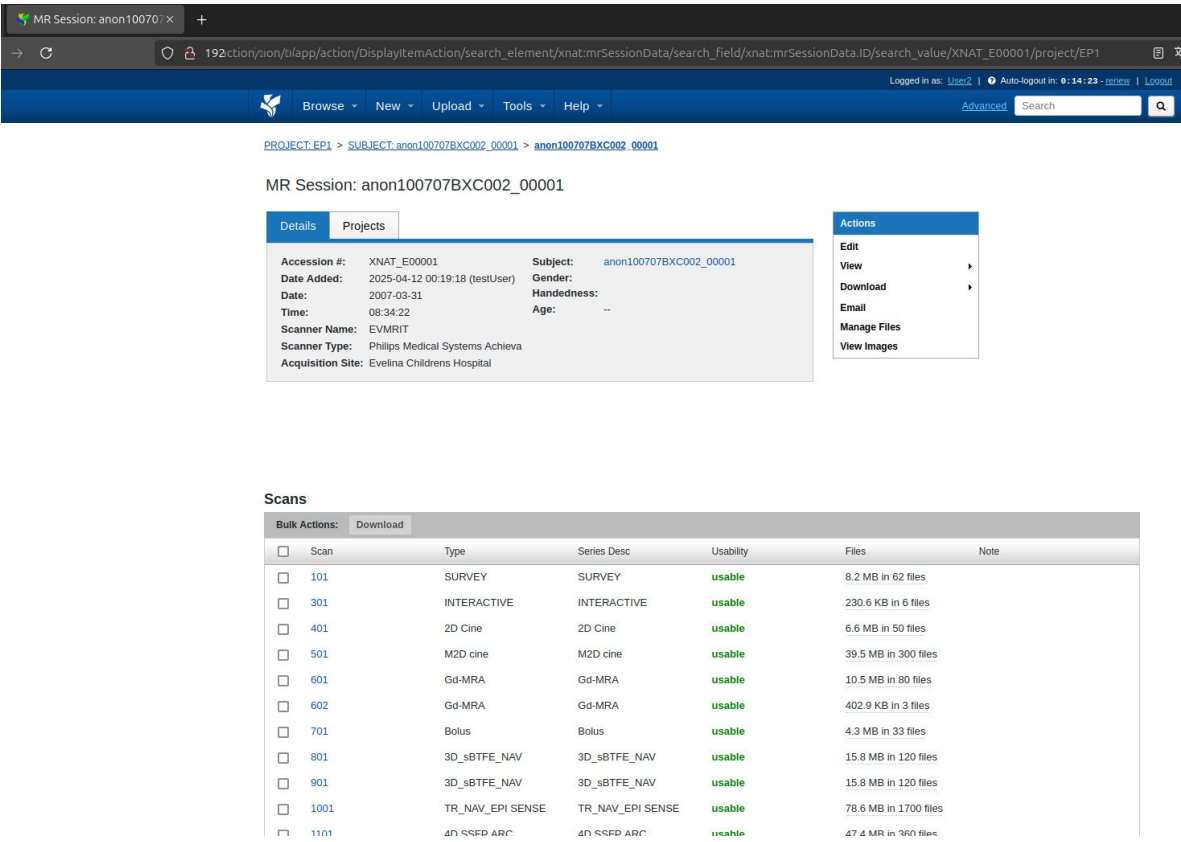


Figura 58: Imágenes médicas del Proyecto Prueba accedido desde el usuario User2.
Fuente: Elaboración Propia.

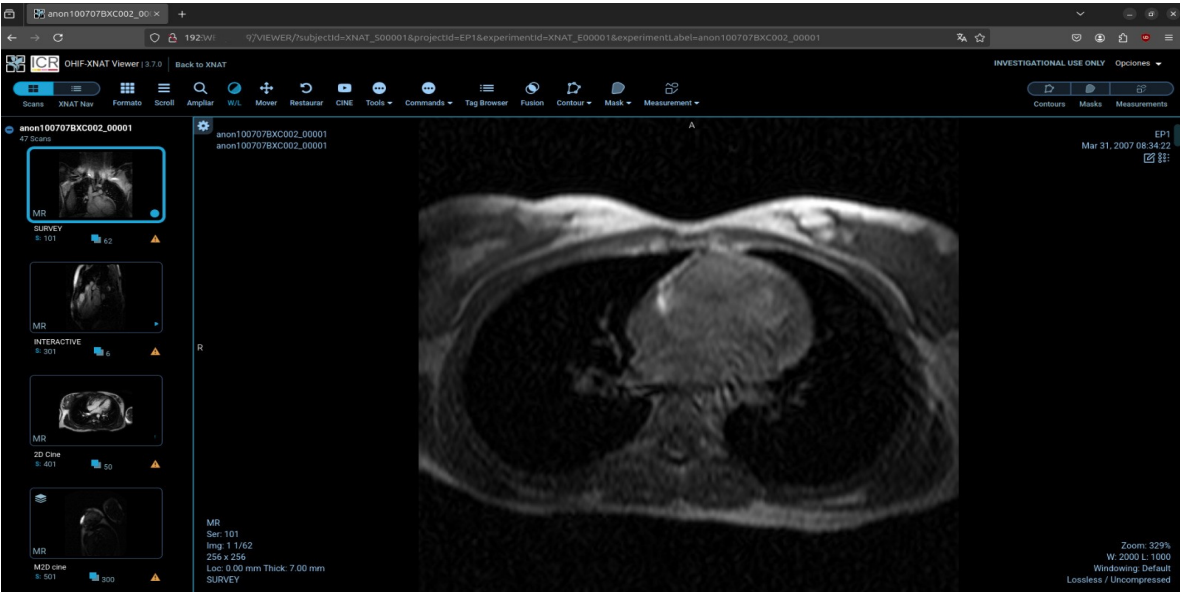


Figura 59: Visualizador OHIF accedido desde el usuario User2.
Fuente: Elaboración Propia.

Este resultado evidencia la correcta implementación de la colaboración multiusuario dentro de la plataforma XNAT, permitiendo que diferentes usuarios con permisos adecuados puedan acceder de forma remota a los mismos recursos.

4.1.5. Estandarizar la información de un paciente con el complemento xnat-fhir-plugin

Para esta prueba piloto se siguieron las instrucciones descritas en la sección 3.4 en el ordenador de escritorio que contiene los contenedores de Docker donde está instalada la plataforma XNAT, logrando obtener los tres archivos compilados para integrar el estándar HL7 FHIR a XNAT.

xnat-fhir-plugin > build > libs				
Nombre	Fecha de modificación	Tipo	Tamaño	
xnat-on-fhir-0.2-alpha2.jar	21-04-2025 18:02	Executable Jar File	338 KB	
xnat-on-fhir-0.2-alpha2-javadoc.jar	21-04-2025 18:01	Executable Jar File	1 KB	
xnat-on-fhir-0.2-alpha2-sources.jar	21-04-2025 18:01	Executable Jar File	126 KB	

Figura 60: Archivos generados luego de seguir las instrucciones de la sección 3.4.
Fuente: Elaboración Propia.

Como se ve en la Figura 60, los archivos generados están en la ubicación xnat-fhir-plugin/build/libs, tal como se describe en la sección 3.4.3. Luego se

siguieron las instrucciones de la sección 3.3.5 para copiar estos archivos en el contenedor `xnat-web-1` para integrar el complemento en la plataforma.

Una vez terminado el proceso de instalación, se procedió a acceder a la sesión del usuario `userTest` para verificar que el complemento estaba integrado en la plataforma, por lo que se ingresó en el proyecto Proyecto Prueba y luego al paciente SJ2.

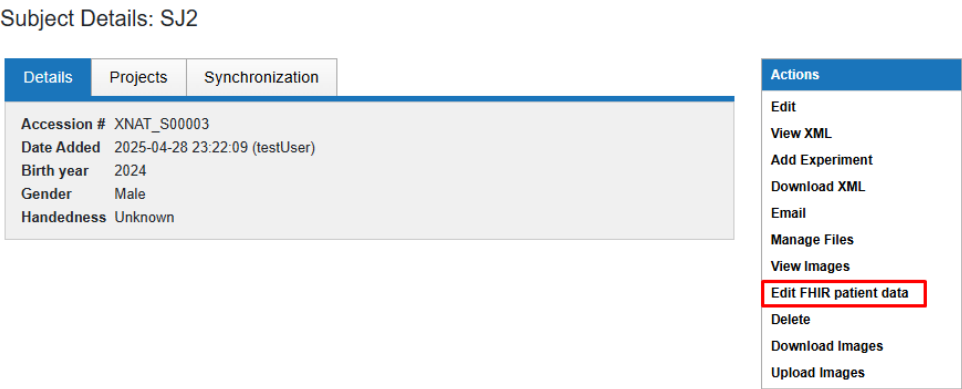


Figura 61: Complemento `xnat-fhir-plugin` ya instalado en la plataforma XNAT.
Fuente: Elaboración Propia.

Como se ve en la Figura 61, en el menú de acciones a la derecha existe la opción **Edit FHIR patient data**, por lo que si hacemos click en ella, nos redirigirá al formulario descrito en la sección 2.3.1 y que se puede ver en la Figura 7. Entonces procederemos a rellenar los campos que corresponden a `Patient Name` con los siguientes datos:

Tabla 4: Datos a ingresar en el formulario del estándar HL7 FHIR
Elaboración Propia.

Patient Name	
Use	nickname
Full name	Mr. Paciente Prueba Primero
Given	Paciente
Prefix	Mr.
Suffix	Primero
Period Start	January/18/2025
Period End	December/1/2025

Luego de haber completado los datos de la Tabla 4, accederemos a la sesión del usuario `User2`, para verificar que los datos ingresados están correctamente estandarizados y son accesibles por cualquier usuario que tenga acceso al proyecto. Entonces accedemos nuevamente al proyecto Proyecto Prueba y al paciente SJ2. Desde el menú de acciones accederemos a `View XML` y buscamos la etiqueta que contenga `fhir` para verificar si los datos ingresados están correctamente estandarizados.

```
<xnat:experiment ID="XNAT_E00006" project="EP1" label="fhir_patient_XNAT_S00003" xsi:type="fhir:patient">
  <xnat:subject_ID>XNAT_S00003</xnat:subject_ID>
  <fhir:name ID="3">
    <fhir:use>nickname</fhir:use>
    <fhir:text>Mr. Paciente Prueba Primero</fhir:text>
    <fhir:period>
      <fhir:start>Sat Jan 18 00:00:00 UTC 2025</fhir:start>
      <fhir:end>Mon Dec 01 00:00:00 UTC 2025</fhir:end>
    </fhir:period>
  </fhir:name>
```

Figura 62: XML del experimento del paciente SJ2.
Fuente: Elaboración propia.

Como se muestra en la Figura 62, se logra acceder al XML del paciente, el cual contiene las etiquetas correspondientes a los datos ingresados en el formulario. Esto evidencia la correcta implementación del complemento `xnat-fhir-plugin` en la plataforma XNAT, permitiendo estandarizar los experimentos del tipo DICOM a HL7 FHIR. Además, esta información queda disponible y accesible por cualquier usuario perteneciente al proyecto, promoviendo el intercambio estructurado y seguro de datos médicos.

4.2. Pruebas piloto de la instalación de XNAT en el servidor

Como se describe en la sección 3.5, se instaló XNAT en un servidor ubicado en la sede San Joaquín de la UTFSM, dejando la dirección web de manera pública accesible desde cualquier conexión de red¹⁷, por lo que se harán las pruebas desde el ordenador de escritorio conectado a internet a través de un cable ethernet.

4.2.1. Creación de Usuario

Esta prueba piloto consistió en la creación de un nuevo usuario desde la página principal de la plataforma con la opción de habilitación automática de usuario deshabilitada, por lo que inmediatamente después de completar el registro de usuario, se intentará acceder para verificar que el inicio de sesión no es posible hasta que la cuenta haya sido habilitada manualmente por el administrador.

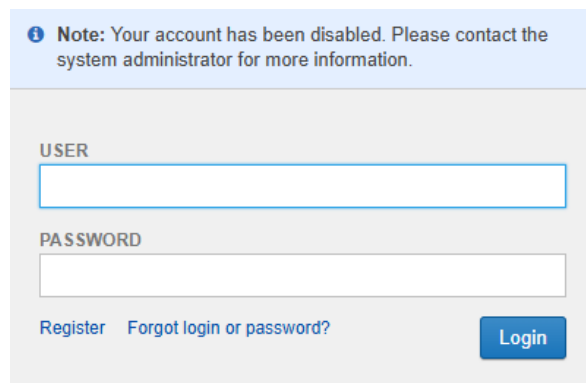
Entonces accedemos al sitio web de la plataforma instalada en el servidor host e ingresamos en la opción `Register`, completando los siguientes datos:

¹⁷Para efectos prácticos, cuando nos referimos acceder al sitio web durante esta sección, nos referiremos a ingresar a la dirección `https://albert-xnat.inf.santiago.usm.cl/`

Tabla 5: Datos para registrar un nuevo usuario en XNAT

New User Registration	
Username	pilotoServer
First Name	Piloto
Last Name	Server
Email	piloto@user.cl
Phone	11223344
Study Site or Lab	
Comments	

Luego de completar el formulario de registro, intentamos acceder a la plataforma y nos arroja este error:



The screenshot shows a web interface for user login. At the top, a light blue banner contains a message: "Note: Your account has been disabled. Please contact the system administrator for more information." Below this, there are two input fields labeled "USER" and "PASSWORD". At the bottom, there are three links: "Register", "Forgot login or password?", and a blue "Login" button.

Figura 63: Error al iniciar sesión con el nuevo usuario registrado
Fuente: Elaboración propia.

Como se observa en la Figura 63, la nota nos indica que la cuenta aún está deshabilitada, por lo que accederemos a la cuenta de administrador para activar manualmente el nuevo usuario registrado.

User Properties for **pilotoServer**

Account Information

Username

pilotoServer

Password

Change User Password

First Name

Piloto

Last Name

Server

Email

piloto@user.cl

Verified

☒

Enabled

☐

Authentication

[View Authentication Details](#)

Advanced

[Edit Advanced Settings](#)

Edit this user's project and security settings.

Figura 64: Datos del nuevo usuario registrado
Fuente: Elaboración propia.

Como se ve en la Figura 64, los datos del usuario coinciden con los señalados en la Tabla 5, por lo que confirma que el registro se llevó a cabo exitosamente. Solo nos queda habilitar la opción desactivada para que la cuenta del usuario registrado recientemente se encuentre activada y pueda acceder a la plataforma. Una vez realizado el cambio, intentaremos ingresar nuevamente a la cuenta.

Last login: 2025-06-25 04:45:34

Logged in as: **pilotoServer** | Auto-logout in: 0:14:48 | [is new](#) | [Logout](#)

New Upload Tools Help

Advanced Search



XNAT currently contains 1 projects, 1 subjects, and 1 imaging sessions.

Projects Subjects MR PET CT

Logged in as: **pilotoServer**

ID:

Title:

Description:

Keywords:

Investigator:

(SELECT)

Submit

Projects

Recent Data Activity

Figura 65: Acceso exitoso luego de habilitar la cuenta del nuevo usuario
Fuente: Elaboración propia.

Como se muestra en la Figura 65, se verificó que la creación de usuario está funcionando correctamente en el servidor. A diferencia de lo ocurrido en la sección 4.1.1, en este entorno las cuentas no se habilitan automáticamente tras completar el registro, reforzando las medidas de seguridad para controlar el acceso a la plataforma. Sin embargo, una vez activadas manualmente por el usuario administrador de la plataforma, los usuarios pueden ingresar sin mayores problemas.

4.3. Acceso a Proyectos

Para esta prueba piloto se siguió la misma metodología utilizada en la sección 4.1.2, creando 2 proyectos privados, Proyecto Prueba(Se le asignó el identificador EP1) y Test Project(Se le asignó el identificador EP2), cada uno con integrantes distintos, distribuidos de la siguiente manera:

Tabla 6: Integrantes de cada proyecto en el servidor
Fuente: Elaboración Propia

Usuarios	Proyecto Prueba	Test Project
mainUser	✓	✓
secondUser		✓
thirUser	✓	

Como se muestra en la Tabla 6, el usuario mainUser pertenece a los 2 proyectos, Proyecto Prueba y Test Project, el usuario secondUser pertenece solamente a un proyecto, que sería Test Project, al igual que thirdUser que solo pertenece a Proyecto Prueba.

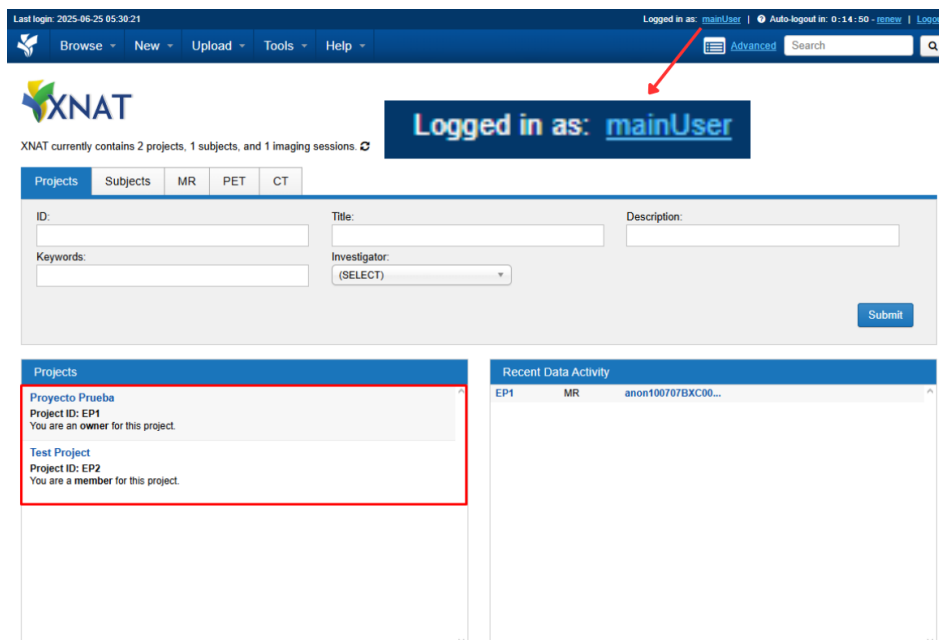


Figura 66: Página principal del usuario mainUser
Fuente: Elaboración propia.

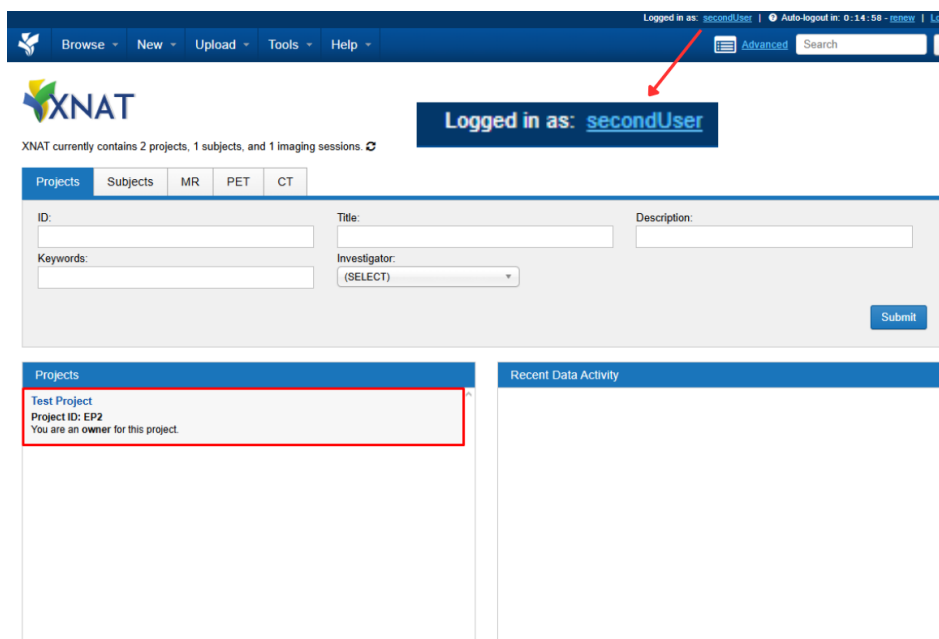


Figura 67: Página principal del usuario secondUser
Fuente: Elaboración propia.

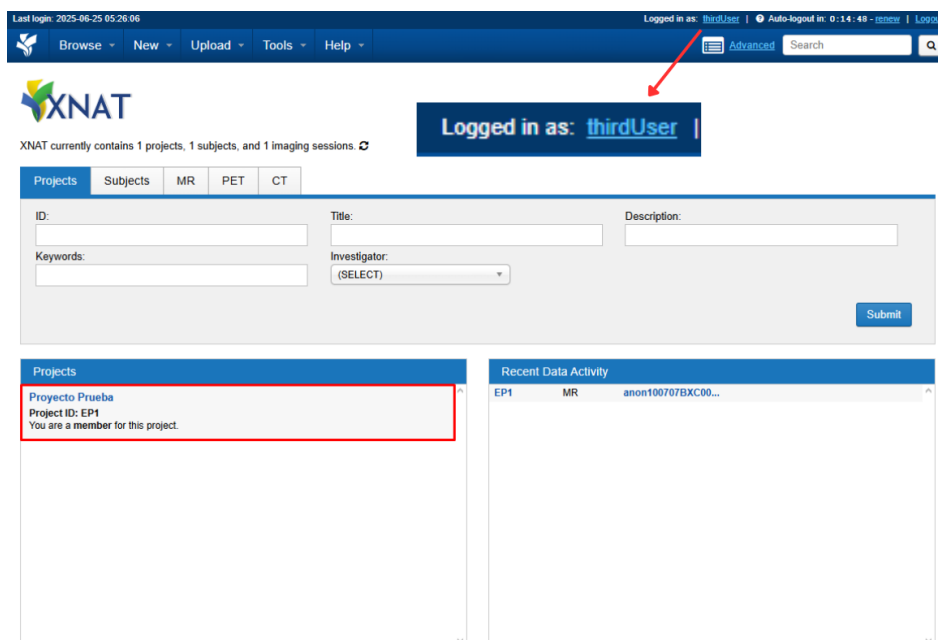


Figura 68: Página principal del usuario thirdUser

Fuente: Elaboración propia.

Como se puede observar en las Figuras 66, 67 y 68, se accedió a la sesión de cada usuario listado en la Tabla 6 con el objetivo de verificar los proyectos a los cuales tienen acceso. En el menú de acceso Projects se listan los proyectos que puede acceder el usuario, mostrando el nombre del proyecto, el identificador y una pequeña descripción del rol que ocupa dentro de este.

En la Figura 66, correspondiente a la sesión del usuario mainUser, este tiene acceso tanto a los proyectos Proyecto Prueba y Test Project, en los cuales posee los roles de **Propietario** y **Miembro** respectivamente.

De manera similar en la Figura 67, el usuario secondUser solamente tiene acceso al proyecto Test Project en la calidad de **Propietario**.

Finalmente, en la Figura 68, el usuario thirdUser tiene acceso en calidad de **Miembro** al proyecto Proyecto Prueba.

Por lo que se verificó que la creación de proyectos está funcionando correctamente en el servidor, más aún, la accesibilidad que tienen los usuarios respeta la distribución realizada en la Tabla 6.

4.3.1. Subir y visualizar un archivo de imagen médica

Para esta prueba piloto se accedió a la sesión del usuario `mainUser` para cargar un conjunto completo de imágenes correspondientes a una sesión de resonancia magnética en el proyecto Proyecto Prueba. Luego de terminar el proceso de cargar las imágenes, se accederá a la sesión del usuario `thirdUser` para acceder al proyecto y verificar que puede visualizar la sesión de resonancia magnética con el complemento `ohif-viewer-3.7.0-XNAT-1.8.10` previamente instalado en el servidor `host`.

Para subir la sesión a la plataforma, se utilizó el método de subir un empaquetado, por lo que una vez iniciada la sesión del usuario `mainUser` y desde el menú de navegación se accedió a `Upload/Images/Compressed Uploader`.

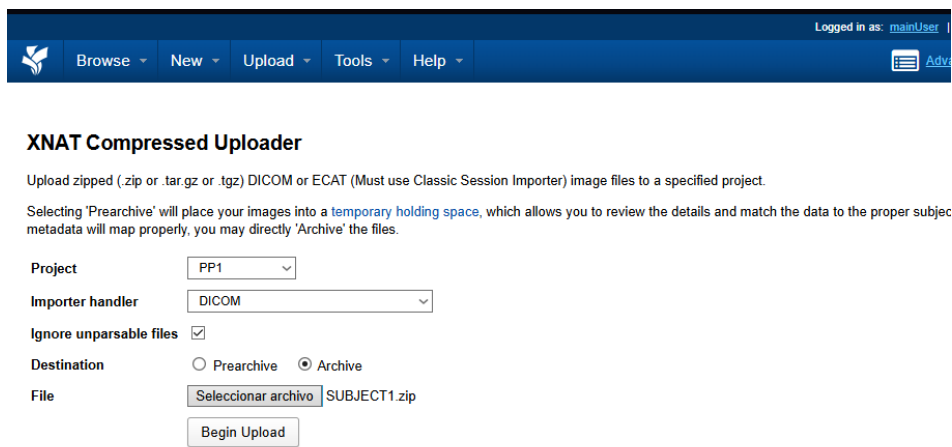


Figura 69: Subir imagenes médicas a la plataforma utilizando la opción de Compressed Uploader

Fuente: Elaboración propia.

Como se observa en la Figura 69, se selecciona la abreviación PP1 que corresponde al proyecto Proyecto Prueba, seleccionamos el archivo `SUBJECT1.zip` ubicado en nuestro ordenador, el cual contiene las imágenes de la sesión de resonancia magnética, y le damos al botón `Begin Upload` para iniciar el proceso de cargado de imágenes.

Una vez terminado el proceso de cargado de imágenes, cerramos la sesión de usuario `mainUser` para luego acceder a la sesión del usuario `thirdUser`. Desde la página principal de la sesión de este usuario, se ingresó al proyecto Proyecto Prueba y al paciente identificado como `anon100707BXC002_00001`. Finalmente, desde el menú lateral de navegación, se seleccionó la opción `View Images` que nos redirigió correctamente al visor web previamente instalado (Ver Figura 70), permitiendo verificar que las imágenes cargadas son accesibles por otros miembros del proyecto.

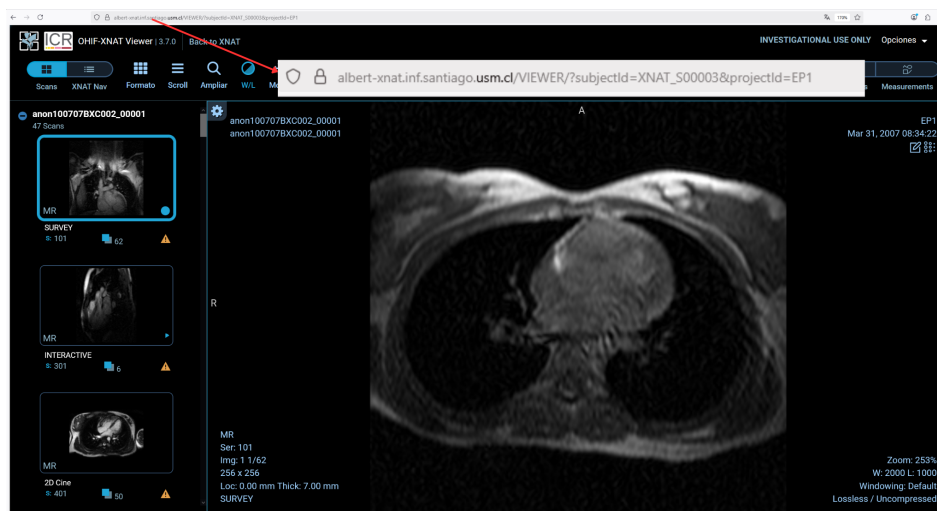


Figura 70: Visor ohif-viewer-3.7.0-XNAT-1.8.10 instalado en el servidor
Fuente: Elaboración propia.

4.3.2. Estandarizar la información de un paciente con el complemento xnat-fhir-plugin previamente instalado en el servidor

Para esta prueba piloto, no fue necesario compilar nuevamente el complemento xnat-fhir-plugin en el servidor. Se reutilizó el archivo compilado previamente durante la prueba descrita en la sección 4.1.5, por lo que únicamente fue necesario transferir dicho archivo al servidor. Tal como se menciona en la sección 3.5, el repositorio clonado en el servidor ya contenía entre sus archivos el complemento xnat-fhir-plugin en su versión compilada, lo cual facilitó su instalación.

Posteriormente, se procedió a verificar su funcionamiento accediendo a la sesión del usuario `thirdUser`, quien estandarizará la información de un paciente. Finalmente, se ingresará a la sesión del usuario `mainUser` para comprobar que los datos registrados fuesen correctos y accesibles desde otra cuenta con permisos dentro del proyecto correspondiente.

Se accedió al paciente `anon100707BXC002_00001` del proyecto Proyecto Prueba desde una sesión del usuario `thirdUser`, logrando ver en el menú de navegación lateral la opción `Edit FHIR patient data` (Ver Figura 71) y al seleccionar esta, redirige al formulario para estandarizar los datos del paciente al formato HL7 FHIR, por lo que la plataforma instalada en el servidor reconoce sin problemas el complemento xnat-fhir-plugin.

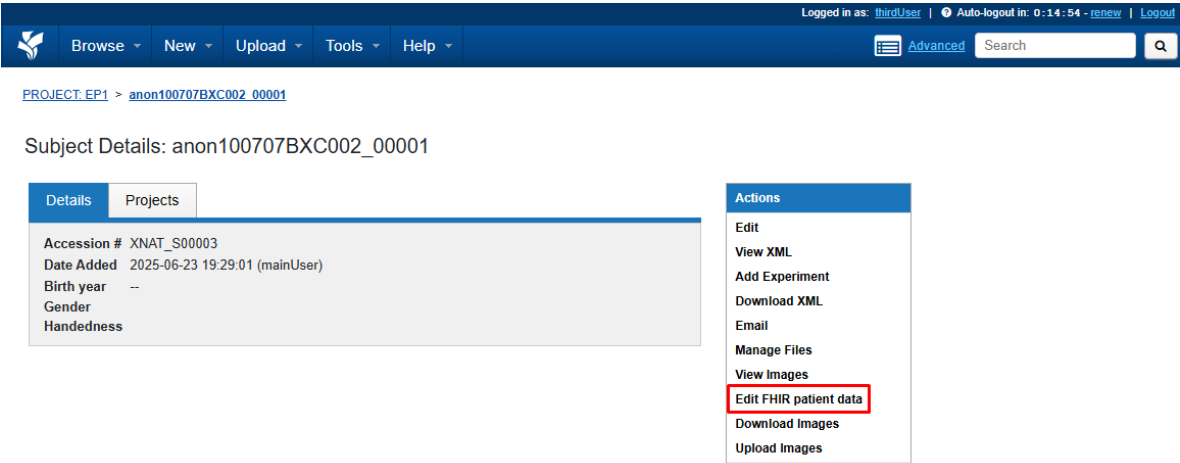


Figura 71: Complemento xnat-fhir-plugin correctamente instalado en el servidor
Fuente: Elaboración propia.

Por lo que se procedió a rellenar parte del formulario con los siguientes datos para estandarizar la información del paciente:

Tabla 7: Datos a ingresar en el formulario del estándar HL7 FHIR del servidor
Elaboración Propia.

Patient Name	
Use	official
Full name	Mrs. Test Paciente Server
Given	Test
Prefix	Mrs.
Suffix	Server
Period Start	February/01/1980
Period End	September/18/2007

Luego de completar el registro, se cerró la sesión actual para ingresar a una nueva sesión con el usuario `mainUser`. Desde esta cuenta, se accedió al paciente `anon100707BXC002_00001`, siguiendo el mismo procedimiento descrito anteriormente. Una vez dentro de la sección correspondiente al paciente, se seleccionó la opción `View XML`, donde fue posible visualizar la información estandarizada previamente registrada por el usuario `thirdUser`(Ver Figura 72).

```
▼<xnat:experiment ID="XNAT_E00003" project="EP1" label="fhir_patient_XNAT_S00003" xsi:type="fhir:patient">
  <xnat:subject_ID>XNAT_S00003</xnat:subject_ID>
  ▼<fhir:name ID="2">
    <fhir:use>official</fhir:use>
    <fhir:text>Mrs. Test Paciente Server</fhir:text>
    ▼<fhir:period>
      <fhir:start>Fri Feb 01 00:00:00 UTC 1980</fhir:start>
      <fhir:end>Tue Sep 18 00:00:00 UTC 2007</fhir:end>
    </fhir:period>
  </fhir:name>
  ▼<fhir:communication>
    <fhir:preferred>1</fhir:preferred>
  </fhir:communication>
```

Figura 72: Complemento xnat-fhir-plugin correctamente instalado en el servidor
Fuente: Elaboración propia.

Como se observa en la Figura 72, el archivo XML permitió confirmar que los datos ingresados mediante el complemento xnat-fhir-plugin fueron correctamente almacenados y son accesibles por otros usuarios del proyecto con los permisos adecuados.

CAPÍTULO 5

CONCLUSIONES

5.1. Conclusiones Generales

En esta memoria se aborda la presentación e implementación de la plataforma XNAT, con el objetivo principal de ofrecer a la comunidad científica una herramienta que permita almacenar y organizar imágenes médicas orientadas a la investigación. Además, se busca facilitar el intercambio de información desde cualquier ordenador o ubicación del mundo, de forma segura y eficiente. De este modo, se promueve el concepto de interoperabilidad en el ámbito de la salud, posicionando a XNAT como una alternativa útil para el desarrollo de proyectos de investigación médica.

En cuanto a la instalación de XNAT, se emplearon dos métodos, la creación de una máquina virtual y el uso de contenedores de Docker, lo que permitió evaluar 2 enfoques distintos de despliegue. Ambas opciones lograron cumplir con los requerimientos funcionales esperados, sin embargo, presentaron diferencias significativas en términos de complejidad, flexibilidad y mantenibilidad.

La instalación de XNAT en una máquina virtual permitió un mayor control sobre cada uno de los componentes del sistema, lo que facilitó una comprensión más profunda del funcionamiento interno de la plataforma a nivel de sistema. Sin embargo, este enfoque presentó algunas desventajas, ya que el proceso fue más lento, requirió un mayor tiempo de configuración y resultó más susceptible a errores relacionados con dependencias y versiones incompatibles.

El uso de contenedores Docker simplificó considerablemente la instalación de XNAT, ya que encapsula todas las dependencias necesarias dentro de los contenedores, lo que permitió un despliegue mucho más rápido, reproducible y con menor riesgo de errores relacionados con versiones o configuraciones del sistema. Esta ventaja se ve reforzada al utilizar la herramienta de `docker-compose` que automatiza y simplifica el proceso de creación de múltiples contenedores, facilitando aún más la puesta en marcha de la plataforma.

Respecto al rendimiento, la instalación mediante máquina virtual presentó un mayor consumo de recursos del sistema, mientras que Docker ofreció una solución más ligera y eficiente, lo que explica por qué es la opción más utilizada al momento de instalar XNAT. Esta diferencia se vuelve especialmente relevante en contextos con recursos limitados. En este caso particular, no fue posible trabajar adecuadamente en una máquina virtual utilizando un notebook con recursos insuficiente, ya que se produjeron latencias y cierres inesperados del complemento `ohif-viewer-3.7.0-XNAT-1.8.10`. Debido a esto, se optó por migrar a un ordenador de escritorio con mejores características y continuar con las pruebas utilizando el método basado en Docker.

Ambos métodos de instalación resultan válidos y funcionales. Sin embargo, para proyectos que priorizan la rapidez en el despliegue, la portabilidad entre entornos y un menor consumo de recursos del sistema, la opción basada en contenedores de Docker representa una alternativa moderna, eficiente y más adecuada frente a los desafíos actuales de implementación tecnológica.

XNAT ofrece un alto nivel de personalización, lo que la convierte en una plataforma versátil y adecuada para una amplia variedad de proyectos médicos y de investigación, adaptándose a las necesidades específicas de cada usuario. Esta flexibilidad se refleja tanto en el desarrollo de complementos como en las opciones de configuración interna.

A nivel de backend, es posible incorporar complementos para extender las funcionalidades de la plataforma. Por ejemplo, el visor **ohif-viewer-3.7.0-XNAT-1.8.10** permite una visualización más precisa de distintos tipos de imágenes médicas. Su interfaz es muy intuitiva y fácil de usar, lo que reduce la necesidad de recurrir constantemente a la documentación. Otro complemento relevante es el que integra el estándar **HL7-FHIR**, el cual facilita el intercambio de información médica de manera estandarizada y segura. Estos y muchos otros complementos se encuentran disponibles en el mercado oficial de XNAT.

Sin embargo, la instalación de complementos en XNAT requiere un nivel técnico avanzado, tal como se describe en las secciones 3.2.3 y 3.3.5. Para un usuario del área de la salud sin conocimientos de las tecnologías empleadas, puede resultar complejo comprender los comandos específicos para transferir los archivos, así como identificar a qué carpeta se tienen que transferir para una instalación exitosa.

Tenemos el complemento `xnat-fhir-plugin` como referencia. A diferencia de otros complementos que se distribuyen directamente en el formato `.jar` requerido por el sistema, este requiere que el usuario descargue el código fuente desde el repositorio oficial, trabajar previamente el código debido a problemas de incompatibilidad de versiones y luego compilar manualmente el archivo `.jar`.

Esta complejidad técnica representa una desventaja para adoptar XNAT en entornos médicos e investigativos, ya que limita su implementación a contextos donde se posee personal capacitado en informática para personalizar y configurar la plataforma a nivel de backend. Por lo tanto, es de gran importancia automatizar los procesos de instalación y configuración para reducir errores técnicos y permitir el acceso a las funcionalidades avanzadas, incluso para usuarios sin formación técnica especializada.

Asimismo, XNAT permite personalizar aspectos internos del sistema desde el frontend de la plataforma, como la modificación de esquemas para agregar nuevos campos a los tipos de imágenes (por ejemplo, incorporar atributos como el color de pelo), personalizar el proceso de anonimización de los datos según requerimientos específicos, o ajustar los mensajes de error, advertencias o correos que se envían al usuario. Todo esto permite adaptar la plataforma tanto desde el punto de vista funcional como de experiencia de usuario, fortaleciendo su utilidad en distintos contextos clínicos o investigativos.

XNAT ofrece un alto nivel de seguridad que se refleja tanto a nivel de usuario como de proyectos. En cuanto al acceso de usuarios, la plataforma requiere que cada cuenta sea verificada mediante correo electrónico y habilitada por el administrador del sistema. Este proceso asegura que solo usuarios autorizados puedan ingresar y trabajar con los datos médicos o de investigación. Por esta razón, es fundamental que el administrador configure correctamente el sistema de correo de la plataforma, de manera que los mensajes de verificación lleguen correctamente a los nuevos usuarios al momento de registrarse.

En cuanto a nivel proyecto, XNAT también proporciona mecanismos de seguridad que regulan la accesibilidad según el tipo definido por el propietario durante la creación del proyecto. Existen tres tipos: **público**, **privado** y **protegidos**. Los proyectos públicos pueden ser consultados por cualquier usuario que haya iniciado sesión, incluyendo usuarios invitados si esta opción está habilitada, permitiendo el acceso completo a la visualización y descarga de los datos. El tipo privado consiste en que el proyecto no será visible ni accesible para ningún usuario de XNAT, excepto aquellos designados por el propietario. Finalmente, los proyectos protegidos son visibles por todos los usuarios registrados en la plataforma(excepto usuarios invitados), sin embargo, requieren una solicitud de acceso al propietario para poder visualizar o descargar los datos. Esta clasificación permite un control detallado sobre la privacidad y confidencialidad de los datos almacenados.

Además del tipo de acceso del proyecto, XNAT permite asignar diferentes roles a los usuarios dentro de cada uno, lo que otorga un mayor control sobre las acciones que cada uno puede realizar. Entre los roles estándar: **propietario**, **colaborador** y **miembro**. El propietario tiene control total sobre el proyecto, incluyendo la gestión de usuarios, configuración de acceso o administración de los datos. Los colaboradores solo tienen acceso de lectura, pueden consultar y descargar la información, pero no pueden modificarla ni insertar nuevos datos. Los miembros tienen permitido gestionar los datos del proyecto, pudiendo visualizar, cargar y modificar los datos dentro de este, pero no cuentan con permisos administrativos. XNAT, además, permite definir roles personalizados con permisos específicos, adaptándose a las necesidades particulares de cada proyecto médico o investigativo. Esta estructura de roles favorece una distribución clara de responsabilidades, garantizando un uso seguro, organizado y controlado de la información médica almacenada en la plataforma.

XNAT también destaca por su alto nivel de interoperabilidad, al ser compatible con estándares médicos ampliamente utilizados como DICOM para el manejo de imágenes médicas. Además, cuenta con una API RESTful que permite su integración con lenguajes de programación como Python, facilitando la automatización de procesos y el desarrollo de herramientas personalizadas. A esto se suma la posibilidad de conectarse con servicios externos de autenticación institucional, como LDAP, lo que refuerza la seguridad y facilita su incorporación en entornos hospitalarios o de investigación. Estas características convierten a XNAT en una plataforma útil, confiable y extensible para el intercambio seguro de datos desde cualquier lugar del mundo. No obstante, para implementar estas funcionalidades, se requiere un conocimiento avanzado para realizar su configuración e integración a la plataforma correctamente.

Las características mencionadas anteriormente refuerzan que la plataforma XNAT representa una opción sólida para abordar diversas problemáticas en el ámbito de la salud. Su capacidad de almacenar, organizar y compartir imágenes médicas de forma segura facilita el trabajo colaborativo entre profesionales de la medicina e investigadores, permitiendo el intercambio de información y conocimientos desde cualquier lugar del país o del mundo. Esto no solo contribuye a mejorar la atención de los pacientes, sino que también impulsa el desarrollo de nuevos descubrimientos científicos orientados a comprender mejor el cuerpo humano y avanzar en la creación de tratamientos y medicamentos que benefician a la humanidad.

Sin embargo, existen ciertas limitaciones que XNAT no puede abordar, como los problemas éticos o morales de los médicos o investigadores que se escapan a sus capacidades técnicas. Si bien la plataforma facilita el intercambio de datos médicos y promueve la colaboración entre profesionales, la decisión de compartir información sigue dependiendo de factores humanos como la confianza, el reconocimiento o el temor a la competencia. El deseo de obtener prestigio por un gran descubrimiento científico puede dificultar la apertura de datos, en este sentido, la plataforma XNAT no puede resolver por sí sola barreras sociales o individuales que afectan la cooperación científica. Ahora bien, si el sector de la salud logra priorizar el bien común y el desarrollo de la humanidad por sobre los intereses personales, plataformas como XNAT pueden convertirse en pilares fundamentales para proyectos colaborativos a gran escala.

5.2. Cumplimiento de Objetivos

A continuación se presentará el grado de cumplimiento de los objetivos definidos en la sección 1.5, comenzando por los objetivos específicos, para luego validar el cumplimiento del objetivo general.

1. Realizar un levantamiento de los requerimientos técnicos para el uso de XNAT y HL7-FHIR.

Tanto en la sección 3.1.1 como en la sección 3.3.1, se describen los requerimientos técnicos necesarios para instalar la plataforma XNAT utilizando el método de una máquina virtual y utilizando contenedores de Docker respectivamente. Luego en las secciones 3.1.2, 3.1.3, 3.1.4, 3.3.2 y 3.3.3 se amplían en detalle en cada herramienta necesaria para la instalación de XNAT describiendo sus funcionalidades y los pasos a seguir para una correcta instalación.

2. Instalar XNAT en un computador local y realizar pruebas piloto usando algún visualizador de imágenes que permita conexión de red.

Según lo descrito en las secciones 3.1 y 3.3, se logró instalar el primer método en un Notebook pero al querer instalar el complemento del visualizador `ohif-viewer-3.7.0-XNAT-1.8.10` presentó cierres repentinos por falta de recursos, por lo que se optó migrar a un computador de escritorio con mejores características y por utilizar el segundo método correspondiente a la instalación mediante

contenedores de Docker para prevenir falta de recursos. En la sección 4.1 se describen las pruebas pilotos realizadas en el ordenador logrando el cumplimiento exitoso en cada una de ellas.

3. Integrar HL7-FHIR el software XNAT instalado localmente y realizar pruebas piloto usando algún visualizador de imágenes que permita conexión de red.

En la sección 3.4 se describen los requerimientos necesarios y las instrucciones para generar los archivos compilados del complemento `xnat-fhir-plugin`, el cual logra integrar el estándar HL7 FHIR a la plataforma XNAT, luego siguiendo las instrucciones descritas en la sección 3.3.5 se logró transferir los archivos al contenedor de Docker para instalar el complemento en XNAT (No se utiliza la sección 3.2.3 debido a que se optó por la opción de utilizar contenedores de Docker). Luego en la sección 4.1.5 se describen las pruebas pilotos realizadas relacionadas al complemento, logrando evidenciar la correcta implementación y funcionamiento de este, permitiendo el intercambio estructurado y seguro de datos clínicos.

4. Evaluar la plataforma HL7-FHIR-XNAT en un servidor para investigación ubicado en la UTFSM, y realizar pruebas de envío de información desde fuera y dentro del campus.

En la sección 3.5 se logró describir el proceso que se realizó para instalar la plataforma XNAT en un servidor de la UTFSM ubicado en la sede San Joaquín y dejarla de manera pública en la web. Luego durante la sección 4.2 se describen pruebas piloto para verificar correctamente el funcionamiento de la plataforma al acceder desde fuera del campus, logrando tener éxito en cada una de ellas, demostrando que XNAT es accesible por cualquier usuario interesado en el sistema, sin embargo tiene ciertas limitaciones por los recursos y seguridad del servidor.

Una vez cumplidos cada uno de los objetivos específicos planteados en esta memoria, se puede afirmar que se logró alcanzar el objetivo general de **integrar el estándar HL7 FHIR en la plataforma XNAT para facilitar el intercambio seguro de datos de imágenes médicas con fines de investigación**. Esto se evidenció en la correcta implementación del complemento correspondiente, así como en el despliegue exitoso de la plataforma en un entorno accesible públicamente, permitiendo que distintos usuarios puedan almacenar, organizar y compartir imágenes médicas de manera segura y eficiente, fortaleciendo así el enfoque de interoperabilidad en el ámbito de la investigación médica.

5.3. Trabajo a Futuro

Durante el proceso de instalación de la plataforma XNAT utilizando contenedores de Docker, se realizó un análisis de los archivos involucrados en su despliegue, lo que se identificó una oportunidad de mejora con el objetivo de aportar a la comunidad de usuarios de Docker, como también para ayudar a los futuros usuarios interesados en utilizar la plataforma que tengan un conocimiento técnico limitado. Se pensó en crear una **nueva imagen de Docker**, basada en la imagen oficial existente, que incorpore las personalizaciones descritas a lo largo de

esta memoria. Entre ellas, destaca la inclusión del visor `ohif-viewer-3.7.0-XNAT-1.8.10` y otros complementos ampliamente utilizados dentro de la comunidad de XNAT con el fin de facilitar aún más el proceso de instalación y configuración de la plataforma.

Otro trabajo que se desea hacer en un futuro consiste en realizar un estudio de mercado y análisis de necesidades tanto para la plataforma XNAT como en el ámbito de la salud. El objetivo es identificar oportunidades para desarrollar un complemento que aporte a la comunidad o bien una aplicación interoperable con las nuevas tecnologías emergentes. Una posible opción es la integración de la inteligencia artificial en el visor `ohif-viewer-3.7.0-XNAT-1.8.10`, lo que permitiría automatizar ciertos procesos de análisis de imágenes médicas. Sin embargo, como se mencionó anteriormente, se necesita un previo estudio riguroso que evalúe su factibilidad y su utilidad en contextos reales para llevar a cabo dicha iniciativa.

5.4. Palabras finales del Autor

La elaboración de esta memoria representó un desafío muy interesante y difícil, ya que todo partió como una idea propia por lo que carecía totalmente de conocimientos del contexto y, sobre todo, del término interoperabilidad. A medida que iba descubriendo información acerca de estándares o de avances que está realizando el país en cuanto a la tecnología en el área de la salud, me sentía cada vez más inspirado y motivado a aprender del enlace que tiene la informática y el mundo de la salud.

Adentrarme en el mundo de la interoperabilidad en salud y en el uso de la plataforma XNAT fue una experiencia de aprendizaje profundamente enriquecedora en mi formación académica. Haber participado en cursos formativos y obtenido certificaciones asociadas a esta herramienta fortaleció mis conocimientos técnicos y a comprender el impacto de la informática en el ámbito clínico y científico. Este proceso, me llevó a reflexionar sobre la importancia de que la informática y la medicina son disciplinas que tienen que colaborar en conjunto para mejorar la atención del paciente, agilizar procesos médicos y facilitar descubrimientos que contribuyen al avance de la sociedad. En este sentido, es evidente que el futuro de la salud dependerá de las soluciones tecnológicas que pueda diseñar e integrar eficazmente el área informática.

ANEXOS

Anexo A

Como se menciona en la sección 3.3.4, al ser la primera vez que se ingresa a la plataforma al utilizar contenedores de Docker, esta nos pedirá configurar ciertas características antes de redirigir a la pestaña de inicio. Estas son las configuraciones que se presentarán:

XNAT Site Setup

The settings below need to be configured before this XNAT system can be used. Please set the properties below and submit the form to continue.

Site Information

Site ID	XNAT	The id used to refer to this site (also used to generate ids). The Site ID must start with a letter and contain only letters, numbers and underscores. It should be a short, one-word name or acronym which describes your site.
Site URL	http://localhost	The address you want visible to users in emails, and other external links. This should be a functional address (i.e. if the user pasted this address in their web browser, they should come to the site). localhost only works if the web browser is located on the same machine. You are required to guarantee that this address is functional for reaching the site.
Administrator Email Address	harmitage@miskatonic.edu	Email address for site administrator.

Figura 73: Configurar la información de la página web XNAT.
Fuente: Elaboración Propia.

Data Storage

Archive Location	/data/xnat/archive
Pre-Archive Location	/data/xnat/prearchive
Cache Location	/data/xnat/cache
Build Location	/data/xnat/build
Pipeline Installation Location	/data/xnat/pipeline
Inbox Location	/data/xnat/inbox

Figura 74: Configuración de almacenamiento de XNAT.
Fuente: Elaboración Propia.

SMTP Server Settings

Enable SMTP? ☒ Enabled

SERVER

Host

Port

Protocol

AUTHENTICATION

SMTP Auth? ☐ Disabled

Username

Password

OTHER SETTINGS

Start TLS? ☐ Disabled

SSL Trust

Email Prefix

Figura 75: Servidor SMTP de XNAT.
Fuente: Elaboración Propia.

Miscellaneous Settings

Require User Login ☒ Required
If checked, then only registered users will be able to access your site. If false, anyone visiting your site will automatically be logged in as 'guest' with access to public data.

Auto-Enable User Registration? ☐ Do Not Auto-enable
If true, user accounts will be enabled automatically when the user registers. Users will be able to access the site and any 'public' projects immediately. If false, the site administrator will be required to manually enable user accounts. Either way the administrator receives an email notification when a user registers.

Require CSRF Token? ☒ Required
Should this site require the use of a token to prevent CSRF attacks on POST, PUT, and DELETEs?

Figura 76: Configuraciones generales de XNAT.
Fuente: Elaboración Propia.

Si dejamos las configuraciones tal como se muestran en las Figuras 73,74,75 y 76, se puede utilizar la plataforma sin ningún problema e incluso al ingresar como administrador, podremos configurar nuevamente estas características.

Como se ve en la Figura 75, si se configura siguiendo los datos de la Tabla 1, el sistema de correos funciona correctamente, de igual manera que cuando se utiliza XNAT en una máquina virtual.

Referencias

- American National Standards Institute. (2025). *About ANSI*. Consultado el 30 de junio de 2025, desde <https://www.ansi.org/about>
- AWS. (s.f.). ¿Qué es la interoperabilidad? <https://aws.amazon.com/es/what-is/interoperability/#:~:text=La%20interoperabilidad%20del%20sistema%20permite,y%20proteger%20mejor%20los%20datos.>
- BC Platforms. (2024). *Overview of EHDS*. Consultado el 1 de julio de 2025, desde https://ehds-bcplatforms-com.translate.goog/overview-of-ehds/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc
- Beier, M., Jansen, C., Mayer, G., Penzel, T., Rodenbeck, A., Siewert, R., Wu, J., & Krefting, D. (2015). Multicenter Data Sharing for Collaboration in Sleep Medicine. *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 880-889. <https://doi.org/10.1109/CCGrid.2015.148>
- BioSpectrum India. (2024, julio). *FHIR: Simplifying Healthcare Information Exchange*. Consultado el 1 de julio de 2025, desde <https://www.biospectrumindia.com/views/98/24972/fhir-simplifying-healthcare-information-exchange.html>
- Cámara de Diputados y Diputadas. (2023). Modifica la ley N° 20.584, que regula los derechos y deberes que tienen las personas en relación con acciones vinculadas a su atención en salud, con el objeto de establecer la interoperabilidad de las fichas clínicas. <https://www.camara.cl/legislacion/ProyectosDeLey/indicaciones.aspx?prmID=16154&prmBOLETIN=15616-11>
- Cortez, E. (2023, noviembre). *Interoperabilidad en salud: Transformando el futuro de la atención médica en Chile y más allá*. Consultado el 30 de mayo de 2024, desde <https://ehealthreporter.com/interoperabilidad-en-salud-transformando-el-futuro-de-la-atencion-medica-en-chile-y-mas-alla/>
- Doran, S. J., Al Sa'd, M., Petts, J. A., Darcy, J., Alpert, K., Cho, W., Sanchez, L. E., Alle, S., El Harouni, A., Genereaux, B., Ziegler, E., Harris, G. J., Aboagye, E. O., Sala, E., Koh, D.-M., & Marcus, D. (2022). Integrating the OHIF viewer into XNAT: Achievements, challenges and prospects for quantitative imaging studies. *Tomography*, 8(1), 497-512.
- Health Level Seven. (2007). About HL7. Consultado en 2024, desde <https://www.hl7.org/about/index.cfm?ref=nav>
- Health Level Seven. (2011). Getting Started with FHIR. <https://hl7.org/fhir/modules.html>
- HL7 International & Firely. (2025, mayo). *2025 State of FHIR Survey Results*. Consultado el 30 de junio de 2025, desde <https://www.hl7.org/fhir/survey2025>
- Khvastova, M., Witt, M., Essenwanger, A., Sass, J., Thun, S., & Krefting, D. (2020). Towards interoperability in clinical research - enabling FHIR on the open-Source research platform XNAT. *J. Med. Syst.*, 44(8), 137.
- Mar Cornelio, O., & Wong Pérez, D. (2023). HL7 un estándar de interoperabilidad en salud: Revisión sistemática de la literatura. *Revista Cubana de Informática Médica*, 15(2), e627.

- Marcus, D. S., Fotenos, A. F., Csernansky, J. G., Morris, J. C., & Buckner, R. L. (2010). Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults. *J. Cogn. Neurosci.*, 22(12), 2677-2684.
- Marcus, D. S., Harwell, J., Olsen, T., Hodge, M., Glasser, M. F., Prior, F., Jenkinson, M., Laumann, T., Curtiss, S. W., & Van Essen, D. C. (2011). Informatics and data mining tools and strategies for the human connectome project. *Front. Neuroinform.*, 5, 4.
- Marcus, D. S., Olsen, T. R., Ramaratnam, M., & Buckner, R. L. (2007). The extensible neuroimaging archive toolkit. *Neuroinformatics*, 5(1), 11-33.
- Ministério da Saúde do Brasil / HL7 Brasil. (2023). *Guia de Implementação do Core do Brasil (BR?Core)*. Consultado el 2 de julio de 2025, desde <https://hl7.org.br/fhir/core/>
- Ndlovu, K., Mars, M., & Scott, R. E. (2021). Interoperability frameworks linking mHealth applications to electronic record systems. *BMC Health Serv. Res.*, 21(1), 459.
- Neuroinformatics Research Group. (2017). *About XNAT*. Consultado el 30 de mayo de 2024, desde <https://www.xnat.org/about/>
- Organización Panamericana de la Salud (OPS). (2021). *Introducción hacia la Interoperabilidad Semántica [Versión 1.0]*. Washington, D.C., Estados Unidos. <https://www.who.int/iris/handle/10665/345957>
- Poline, J.-B., Breeze, J. L., Ghosh, S., Gorgolewski, K., Halchenko, Y. O., Hanke, M., Haselgrove, C., Helmer, K. G., Keator, D. B., Marcus, D. S., Poldrack, R. A., Schwartz, Y., Ashburner, J., & Kennedy, D. N. (2012). Data sharing in neuroimaging research. *Front. Neuroinform.*, 6.
- Portal Red Salud. (2023). Trazando el futuro de la salud en Chile: Interoperabilidad y protección de datos. <https://portalredsalud.cl/2023/10/23/trazando-el-futuro-de-la-salud-en-chile-interoperabilidad-y-proteccion-de-datos/>
- Pro Salud Chile. (2023). Sistema de Receta Médica Electrónica MINSAL. <https://www.youtube.com/watch?v=8H1JdG2Vcvo>
- Pylypchuk, Y., & Everson, J. (2012). Interoperability and Methods of Exchange among Hospitals in 2021. En *ASTP Health IT Data Brief* (pp. 1-17). Office of the Assistant Secretary for Technology Policy.
- SSMC. (2019). Proyecto SIDRA. https://www.ssmc.gob.cl/?page_id=2216