

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 1 de 9

21.1

FECHA	viernes, 10 de diciembre de 2021
--------------	----------------------------------

Señores
UNIVERSIDAD DE CUNDINAMARCA
 BIBLIOTECA
 Fusagasugá

UNIDAD REGIONAL	Sede Fusagasugá
TIPO DE DOCUMENTO	Trabajo De Grado
FACULTAD	Ingeniería
NIVEL ACADÉMICO DE FORMACIÓN O PROCESO	Pregrado
PROGRAMA ACADÉMICO	Ingeniería de Sistemas

El Autor(Es):

APELLIDOS COMPLETOS	NOMBRES COMPLETOS	No. DOCUMENTO DE IDENTIFICACIÓN
Galvis Diaz	Mateo	1007389353
Garcia Marquez	Yezny Yormary	1032508296

Director(Es) y/o Asesor(Es) del documento:

APELLIDOS COMPLETOS	NOMBRES COMPLETOS
Orjuela Endo	Luis Eduardo
Diaz Jurado	Lady Carolina

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
 Teléfono: (091) 8281483 Línea Gratuita: 018000180414
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
 NIT: 890.680.062-2

*Documento controlado por el Sistema de Gestión de la Calidad
 Asegúrese que corresponde a la última versión consultando el Portal Institucional*

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 2 de 9

TÍTULO DEL DOCUMENTO

Desarrollo De Prototipo Web Para La Gestión De Historias Clínicas Digitales Como Herramienta Práctica En El Programa De Enfermería De La Universidad De Cundinamarca.

SUBTÍTULO

(Aplica solo para Tesis, Artículos Científicos, Disertaciones, Objetos Virtuales de Aprendizaje)

EXCLUSIVO PARA PUBLICACIÓN DESDE LA DIRECCIÓN INVESTIGACIÓN

INDICADORES	NÚMERO
ISBN	
ISSN	
ISMN	

AÑO DE EDICIÓN DEL DOCUMENTO

10/12/2021

NÚMERO DE PÁGINAS

135

DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS (Usar 6 descriptores o palabras claves)

ESPAÑOL	INGLÉS
1. Historias Clínicas	1. Clinical Histories
2. Prácticas	2. Practices
3. Salud	3. Health
4. Pacientes	4. Patients
5. Enfermería	5. Nursing
6. Estudiantes	6. Students

FUENTES (Todas las fuentes de su trabajo, en orden alfabético)

Angular. (2021). Introducción a los conceptos de Angular.

<https://docs.angular.lat/guide/architecture>.


Azure Boards. Docs.microsoft.com. (2021). Artefactos de plantilla de proceso de scrum.

<https://docs.microsoft.com/es-es/azure/devops/boards/work-items/guidance/scrum-process?view=azure-devops>.

BBVA API_Market. (2016, 23 de marzo). API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. <https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/>.

Diagonal 18 No. 20-29 Fusagasugá – Cundinamarca
 Teléfono: (091) 8281483 Línea Gratuita: 018000180414
www.ucundinamarca.edu.co E-mail: info@ucundinamarca.edu.co
 NIT: 890.680.062-2

Documento controlado por el Sistema de Gestión de la Calidad
 Asegúrese que corresponde a la última versión consultando el Portal Institucional

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 3 de 9

Cases Sanchis, M.; Pons Torres, A. Registro de la actividad profesional enfermera en la historia clínica digital: experiencia en Menorca. Metas de Enfermería, 2014, vol. 17, no 8, p. 62-68.

Cases, Sanchi., Torres., A. Registro de la actividad profesional enfermera en la historia clínica digital: experiencia en Menorca. Metas de Enfermería, 2014, 17(8), p. 62-68. <https://www.picuida.es/registro-de-la-actividad-profesional-enfermera-en-la-historia-clinica-digital-experiencia-en-menorca/>.

Chá, Ghiglia., Mercedes, M. (2019). Historia clínica electrónica: herramienta para la continuidad de asistencia. Revista Médica del Uruguay, 35(3), 107-123. <https://dx.doi.org/10.29193/rmu.35.3.6>.

Clouding. (2021). VPS Cloud: Tu Servidor Virtual en España - ¡Pruébanos! <https://clouding.io/#>.

Cocina, E. G., & Torres, F. P. (2007). La historia clínica electrónica. Revisión y análisis de la actualidad. Diraya: la historia de salud electrónica de Andalucía. Revista Española de Cardiología Suplementos, 7(3), 37C-46C.

Consejo Profesional Nacional de Ingeniería. (2021). Código de ética | Copnia. Copnia.gov.co. <https://www.copnia.gov.co/tribunal-de-etica/codigo-de-etica>.

Crispín, L., Doria, M., Rivera, A., Garza, M. et al(2011). Aprendizaje autónomo: orientaciones para la docencia. Universidad Iberoamericana. http://biblioteca.clacso.edu.ar/Mexico/dcsyp-uia/20170517031227/pdf_671.pdf.

Genero Bocco, M., Piattini Velthuis, M. G., Cruz-Lemus, J. A.(2015). Métodos de investigación en Ingeniería del software. Ediciones de la U. <https://www-ebooks7-24-com.ucundinamarca.basesdedatosezproxy.com/?il=7734>.

Google Developers. (2021). Firebase. <https://firebase.google.com/>.

Jiménez Fernández, J. C., & Cerrillo Martín, D. (2010). Registros de Enfermería: un espejo del trabajo asistencial. Metas de Enfermería, 13(6), 8-11.


Microsoft (2021). Introduction to GitHub in Visual Studio Code - Learn. <https://docs.microsoft.com/en-us/learn/modules/introduction-to-github-visual-studio-code/>.

Microsoft. GitHub. [Windows]. <http://git-scm.com/download/win>.

Microsoft. VisualStudioCode. (Versión 1.61) [Windows]. <https://code.visualstudio.com/>.
Ministerio de Educación. (1980, 15 de diciembre). Normas en materia de Ética Médica. [articulos-103905_archivo_pdf.pdf](http://www.mineducacion.gov.co/articulos/articulos-103905_archivo_pdf.pdf) (mineducacion.gov.co).

MKLabs Co.,Ltd. (2021). STARUML [Windows]. <https://staruml.io/>.

Npm. (2021). Express-api-cli. <https://www.npmjs.com/package/express-api-cli>.

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 4 de 9

Postman Enterprise. (2021). Postman (Version 8.12) [Windows].
<https://www.postman.com/>.

Programador Clic. (2021,8 de noviembre). Conozca JWT (JSON WEB TOKEN).
<https://programmerclick.com/article/5839871491/>.

Ramoz, M., Alfaro, M., Sánchez, M. (2009). Archivo de historias clínicas Digitalizado, una solución previa a la Historia Clínica Electrónica. Papeles médicos, vol.(18). num 2.
<https://pdf4pro.com/cdn/archivo-de-historias-cl-237-nicas-digitalizado-una-soluci-243-n-46c7f2.pdf>.

Rodríguez, P. & Rodríguez, Luis. (1999). Principios técnicos para realizar la anamnesis en el paciente adulto. Revista Cubana de Medicina General Integral, 15(4), 409-414.
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21251999000400011&lng=es&tlng=es.


Rubinos, M., Nuevo, A. (2011) Seguridad en bases de datos. Revista Cubana de Ciencias Informática, 5(1). <http://www.redalyc.org/articulo.oa?id=378343671005>.

Solarte, R. (2015). Historia clínica electrónica desde un dispositivo móvil. Scientia et Technica Vol. 20(04).370-373. <https://doi.org/10.22517/23447214.9255>.

Tessa, R. (2019). Dificultades percibidas por estudiantes de la licenciatura de enfermería, en su inserción a la práctica clínica de paciente crítico. Uruguay. Salud Areandina, 2(2). <https://doi.org/10.33132/23229659.1284>.

TypeORM. (s.f). TypeORM. <https://typeorm.io/#/>.

Villa, A., Poblete, M., (2007). Aprendizaje basado en competencias. Una propuesta para la evaluación de las competencias genéricas. Bilbao: Editorial Mensajero, Colección Estudios e Investigación del ICE. Universidad de Deusto, 336 pp.

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 5 de 9

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS

(Máximo 250 palabras – 1530 caracteres, aplica para resumen en español):

Este documento es el informe final de trabajo de grado, como requisito para obtener el título de Ingeniería de Sistemas en la Universidad de Cundinamarca, en donde se ha dado origen la idea de crear un prototipo de ámbito práctico para estudiantes de Enfermería pertenecientes a la misma Universidad. En el presente se exhibe la recolección de datos, las características y el desarrollo del prototipo de historias clínicas digitales, para la práctica de los futuros enfermeros que actualmente cursan su formación profesional en la Universidad de Cundinamarca.

Por lo tanto, se crea el prototipo de un sistema de información, que permite mejorar los procedimientos del estudiante de Enfermería en sus prácticas, proporciona el registro de información para que él mismo pueda reforzar sus habilidades y sus competencias en su ámbito clínico.

Por último, se evidencian las etapas de desarrollo, la recolección de datos, indicios de investigación y la búsqueda de herramientas para el entorno de desarrollo que apoye a la construcción de este.

This document is the final report of the degree work, as a requirement to obtain the title of Systems Engineering at the University of Cundinamarca, where the idea of creating a prototype of a practical field for Nursing students belonging to the same has originated University. In the present, the data collection, characteristics, and development of the prototype of digital medical records are exhibited, for the practice of future nurses who are currently pursuing their professional training at the University of Cundinamarca.


Therefore, the prototype of an information system is created, which allows improving the procedures of the Nursing student in their practices, provides the information record so that they can reinforce their skills and competencies in their clinical field.

Finally, the development stages, data collection, research evidence and the search for tools for the development environment that support the construction of this are evidenced.

AUTORIZACIÓN DE PUBLICACIÓN

Por medio del presente escrito autorizo (Autorizamos) a la Universidad de Cundinamarca para que, en desarrollo de la presente licencia de uso parcial, pueda ejercer sobre mí (nuestra) obra las atribuciones que se indican a continuación, teniendo en cuenta que, en cualquier caso, la finalidad perseguida será facilitar, difundir y promover el aprendizaje, la enseñanza y la investigación.

En consecuencia, las atribuciones de usos temporales y parciales que por virtud de la presente licencia se autoriza a la Universidad de Cundinamarca, a los usuarios de la Biblioteca de la Universidad; así como a los usuarios de las redes, bases de datos y demás sitios web con los que la Universidad tenga perfeccionado una alianza, son:
 Marque con una "X":


	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 6 de 9

AUTORIZO (AUTORIZAMOS)	SI	NO
1. La reproducción por cualquier formato conocido o por conocer.	X	
2. La comunicación pública, masiva por cualquier procedimiento o medio físico, electrónico y digital.	X	
3. La inclusión en bases de datos y en sitios web sean éstos onerosos o gratuitos, existiendo con ellos previa alianza perfeccionada con la Universidad de Cundinamarca para efectos de satisfacer los fines previstos. En este evento, tales sitios y sus usuarios tendrán las mismas facultades que las aquí concedidas con las mismas limitaciones y condiciones.	X	
4. La inclusión en el Repositorio Institucional.	X	

De acuerdo con la naturaleza del uso concedido, la presente licencia parcial se otorga a título gratuito por el máximo tiempo legal colombiano, con el propósito de que en dicho lapso mi (nuestra) obra sea explotada en las condiciones aquí estipuladas y para los fines indicados, respetando siempre la titularidad de los derechos patrimoniales y morales correspondientes, de acuerdo con los usos honrados, de manera proporcional y justificada a la finalidad perseguida, sin ánimo de lucro ni de comercialización.

Para el caso de las Tesis, Trabajo de Grado o Pasantía, de manera complementaria, garantizo(garantizamos) en mi(nuestra) calidad de estudiante(s) y por ende autor(es) exclusivo(s), que la Tesis, Trabajo de Grado o Pasantía en cuestión, es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi(nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro (aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos de la Tesis o Trabajo de Grado es de mí (nuestra) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

Sin perjuicio de los usos y atribuciones otorgadas en virtud de este documento, continuaré (continuaremos) conservando los correspondientes derechos patrimoniales sin modificación o restricción alguna, puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún

	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 7 de 9

caso conlleva la enajenación de los derechos patrimoniales derivados del régimen del Derecho de Autor.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “*Los derechos morales sobre el trabajo son propiedad de los autores*”, los cuales son irrenunciables, imprescriptibles, inembargables e inalienables. En consecuencia, la Universidad de Cundinamarca está en la obligación de RESPETARLOS Y HACERLOS RESPETAR, para lo cual tomará las medidas correspondientes para garantizar su observancia.

NOTA: (Para Tesis, Trabajo de Grado o Pasantía):

Información Confidencial:

Esta Tesis, Trabajo de Grado o Pasantía, contiene información privilegiada, estratégica, secreta, confidencial y demás similar, o hace parte de la investigación que se adelanta y cuyos resultados finales no se han publicado.

SI ___ NO X.

En caso afirmativo expresamente indicaré (indicaremos) en carta adjunta, expedida por la entidad respectiva, la cual informa sobre tal situación, lo anterior con el fin de que se mantenga la restricción de acceso.

LICENCIA DE PUBLICACIÓN

Como titular(es) del derecho de autor, confiero(erimos) a la Universidad de Cundinamarca una licencia no exclusiva, limitada y gratuita sobre la obra que se integrará en el Repositorio Institucional, que se ajusta a las siguientes características:

- a) Estará vigente a partir de la fecha de inclusión en el repositorio, por un plazo de 5 años, que serán prorrogables indefinidamente por el tiempo que dure el derecho patrimonial del autor. El autor podrá dar por terminada la licencia solicitándolo a la Universidad por escrito. (Para el caso de los Recursos Educativos Digitales, la Licencia de Publicación será permanente).
- b) Autoriza a la Universidad de Cundinamarca a publicar la obra en formato y/o soporte digital, conociendo que, dado que se publica en Internet, por este hecho circula con un alcance mundial.
- c) Los titulares aceptan que la autorización se hace a título gratuito, por lo tanto, renuncian a recibir beneficio alguno por la publicación, distribución, comunicación pública y cualquier otro uso que se haga en los términos de la presente licencia y de la licencia de uso con que se publica.

 UDECA UNIVERSIDAD DE CUNDINAMARCA	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 8 de 9

d) El(Los) Autor(es), garantizo(amos) que el documento en cuestión es producto de mi(nuestra) plena autoría, de mi(nuestro) esfuerzo personal intelectual, como consecuencia de mi (nuestra) creación original particular y, por tanto, soy(somos) el(los) único(s) titular(es) de la misma. Además, aseguro(aseguramos) que no contiene citas, ni transcripciones de otras obras protegidas, por fuera de los límites autorizados por la ley, según los usos honrados, y en proporción a los fines previstos; ni tampoco contempla declaraciones difamatorias contra terceros; respetando el derecho a la imagen, intimidad, buen nombre y demás derechos constitucionales. Adicionalmente, manifiesto (manifestamos) que no se incluyeron expresiones contrarias al orden público ni a las buenas costumbres. En consecuencia, la responsabilidad directa en la elaboración, presentación, investigación y, en general, contenidos es de mí (nuestro) competencia exclusiva, eximiendo de toda responsabilidad a la Universidad de Cundinamarca por tales aspectos.

e) En todo caso la Universidad de Cundinamarca se compromete a indicar siempre la autoría incluyendo el nombre del autor y la fecha de publicación.

f) Los titulares autorizan a la Universidad para incluir la obra en los índices y buscadores que estimen necesarios para promover su difusión.

g) Los titulares aceptan que la Universidad de Cundinamarca pueda convertir el documento a cualquier medio o formato para propósitos de preservación digital.

h) Los titulares autorizan que la obra sea puesta a disposición del público en los términos autorizados en los literales anteriores bajo los límites definidos por la universidad en el “Manual del Repositorio Institucional AAAM003”

i) Para el caso de los Recursos Educativos Digitales producidos por la Oficina de Educación Virtual, sus contenidos de publicación se rigen bajo la Licencia Creative Commons: Atribución- No comercial- Compartir Igual.



j) Para el caso de los Artículos Científicos y Revistas, sus contenidos se rigen bajo la Licencia Creative Commons Atribución- No comercial- Sin derivar.



Nota:

Si el documento se basa en un trabajo que ha sido patrocinado o apoyado por una entidad, con excepción de Universidad de Cundinamarca, los autores garantizan que se ha cumplido con los derechos y obligaciones requeridos por el respectivo contrato o acuerdo.

 UDECA UNIVERSIDAD DE CUNDINAMARCA	MACROPROCESO DE APOYO	CÓDIGO: AAAR113
	PROCESO GESTIÓN APOYO ACADÉMICO	VERSIÓN: 6
	DESCRIPCIÓN, AUTORIZACIÓN Y LICENCIA DEL REPOSITORIO INSTITUCIONAL	VIGENCIA: 2021-09-14
		PAGINA: 9 de 9

La obra que se integrará en el Repositorio Institucional está en el(los) siguiente(s) archivo(s).

Nombre completo del Archivo Incluida su Extensión (Ej. Nombre completo del proyecto.pdf)	Tipo de documento (ej. Texto, imagen, video, etc.)
1. DESARROLLO DE PROTOTIPO WEB PARA LA GESTIÓN DE HISTORIAS CLÍNICAS DIGITALES COMO HERRAMIENTA PRÁCTICA EN EL PROGRAMA DE ENFERMERÍA DE LA UNIVERSIDAD DE CUNDINAMARCA.pdf	Texto
2.Historias_Clinicas_master.zip	Carpeta comprimida
3.	
4.	

En constancia de lo anterior, Firmo (amos) el presente documento:

APELLIDOS Y NOMBRES COMPLETOS	FIRMA (autógrafo)
Galvis Diaz Mateo	
Garcia Marquez Yezny Yormary	

21.1-51-20.

Desarrollo De Prototipo Web Para La Gestión De Historias Clínicas Digitales Como
Herramienta Práctica En El Programa De Enfermería De La Universidad De Cundinamarca.

Mateo Galvis Diaz

Yezny Yormary Garcia Marquez

Universidad De Cundinamarca

Programa De Ingeniería De Sistemas

Facultad De Ingeniería

Sede Fusagasugá

2021

Desarrollo De Prototipo Web Para La Gestión De Historias Clínicas Digitales Como
Herramienta Práctica En El Programa De Enfermería De La Universidad De Cundinamarca.

Mateo Galvis Diaz

Yezny Yormary Garcia Marquez

Documento De Proyecto De Grado Para Optar Por El Título De Ingeniero De Sistemas

Director (A) De Disciplinar: Luis Eduardo Orjuela Endo.

Asesor(A) Metodológico : Lady Carolina Diaz Jurado.

Universidad De Cundinamarca

Programa De Ingeniería De Sistemas

Facultad De Ingeniería

Sede Fusagasugá

2021

Tabla de contenido

Introducción	10
1. Planteamiento Del Problema	11
1.1. Descripción Del Problema	11
<i>1.1.1. Formulación Del Problema</i>	<i>12</i>
2. Justificación	12
2.1. Delimitación	13
3. Objetivo General.....	15
3.1. Objetivos Específicos.....	15
4. Alcance.....	16
5. Limitaciones.....	17
6. Marco Referencial.....	18
6.1. Marco Teórico	18
<i>6.1.1. Historia Clínica</i>	<i>18</i>
<i>6.1.2. Plataforma Y Servidores Web</i>	<i>19</i>
<i>6.1.3. Aplicación Web</i>	<i>19</i>
<i>6.1.4. Prototipo.....</i>	<i>20</i>
<i>6.1.5. Api</i>	<i>20</i>
<i>6.1.6. Dominio.....</i>	<i>20</i>
<i>6.1.7. Hosting.....</i>	<i>21</i>
<i>6.1.8. Servidor Web.....</i>	<i>21</i>
<i>6.1.9. Api Rest</i>	<i>21</i>
6.2. Marco Conceptual.....	22
<i>6.2.1. Estrategias Para El Aprendizaje</i>	<i>22</i>
<i>6.2.2. Aprendizaje Autónomo</i>	<i>23</i>
<i>6.2.3. Aprendizaje Basado En Competencias</i>	<i>23</i>
<i>6.2.4. Recursos Educativos Digitales</i>	<i>25</i>
6.3. Marco Ético Y Legal De Ingeniería.....	26
6.4. Marco Teórico Médico.....	26
6.5. Estado Del Arte.....	27
<i>6.5.1. Zeus Salud</i>	<i>27</i>
<i>6.5.2. Heon Health On Line</i>	<i>28</i>

6.5.3.	<i>Historia Clínica Electrónica Desde Un Dispositivo Móvil</i>	28
6.5.4.	<i>Software Médico</i>	29
6.6.	Marco Tecnológico	29
6.6.1.	<i>Visual Studio Code</i>	29
6.6.2.	<i>Node</i>	30
6.6.3.	<i>Postgres</i>	30
6.6.4.	<i>PgAdmin y PgShell</i>	30
6.6.5.	<i>Angular</i>	31
6.6.6.	<i>Express</i>	31
6.6.7.	<i>GitHub</i>	32
6.6.8.	<i>Orm</i>	32
6.6.9.	<i>TypeORM</i>	32
6.6.10.	<i>Firebase</i>	33
6.6.11.	<i>Clouding.io</i>	33
6.6.12.	<i>Postman</i>	33
6.7.	Marco De Trabajo	34
6.7.1.	<i>Marco De Trabajo Scrum</i>	34
6.8.	Marco Metodológico	41
6.8.1.	<i>Metodología Por Prototipos</i>	41
7.	Apropiación Metodología De Desarrollo Por Prototipos	42
7.1.	Etapas: Investigación Preliminar	42
7.1.1.	<i>Análisis Del Entorno</i>	42
7.1.2.	<i>Reglas De Negocio</i>	43
7.1.3.	<i>Cronograma</i>	44
7.1.4.	<i>Definición De Los Requerimientos Funcionales</i>	46
7.1.5.	<i>Diagrama De Casos De Uso</i>	50
7.2.	Etapas: Análisis	57
7.2.1.	<i>Diagramas UML</i>	57
7.2.2.	<i>Modelo Relacional</i>	76
7.2.3.	<i>Modelo Entidad Relación</i>	76
7.2.4.	<i>Modelo Vista Controlador</i>	78
7.2.5.	<i>Definición De Las Herramientas De Desarrollo</i>	78

7.2.6. <i>Librerías Y Funciones</i>	80
7.3. Etapa: Diseño	82
7.3.1. <i>Diagramas De Esquemización Del Prototipo</i>	82
7.3.2. <i>Maquetación y Diseño de Interfaces</i>	83
7.3.3. <i>Definición De Temática</i>	88
7.3.4. <i>Logo e Iconografía</i>	89
7.4. Etapa: Construcción De Prototipo	91
7.4.1. <i>Desarrollo del Prototipo</i>	91
7.4.2. <i>Repositorio del proyecto</i>	105
7.5. Etapa: Implementación	106
7.6. Etapa: Validación de Requisitos del Sistema	108
8. Conclusiones Y Recomendaciones	129
9. Referencias	132

TABLA DE FIGURAS

Figura 1. Captura reuniones Sprint Review	38
Figura 2. Proceso Scrum	39
Figura 3.. Épicas creadas para el prototipo en AzureDevOps.....	39
Figura 4.. Características creadas para el prototipo en AzureDevOps.....	40
Figura 5. Product BackLog creados para el prototipo en AzureDevOps.....	40
Figura 6. Tablero creado para el prototipo en AzureDevOps	41
Figura 7. Diagrama de caso de uso	58
Figura 8. Diagrama de secuencia de historia clínica.....	59
Figura 9.. Diagrama de secuencia de iniciar sesión	60
Figura 10. Diagrama de secuencia de cerrar sesión	61
Figura 11. Diagrama de secuencia de JWT.....	62
Figura 12. Diagrama de requerimientos “paquete usuario”	63
Figura 13. Diagrama de requerimientos “Paquete de historia clínica”	64
Figura 14. Diagrama de requerimientos "Paquete gestión del sistema"	65
Figura 15. Diagrama de requerimientos "Paquete portal web	65
Figura 16. .Diagrama de actividades historia clínica	66
Figura 17. Diagrama de actividades de inicio de sesión	67
Figura 18. Diagrama de actividades crear usuario	68
Figura 19. Diagrama de actividades consultar usuario.	69
Figura 20. Diagrama de actividades Borrar usuario.....	70
Figura 21 Diagrama de actividades Actualizar usuario	71
Figura 22. Diagrama de paquetes "ControllerExpressjs"	72
Figura 23. Diagrama de paquetes "ViewAngular"	73
Figura 24. Diagrama de despliegue.....	74
Figura 25. Diagrama de componentes.....	75
Figura 26. Diagrama Relacional Base De Datos.....	76
Figura 27. Modelo Entidad Relación	77
Figura 28. Diagrama De Modelo Vista Controlador Angular.....	78
Figura 29. Arquitectura de ficheros angular	79
Figura 30. Diagrama esquema prototipo.....	83
Figura 31- Contenido de usuario estudiante.....	84
Figura 32. Contenido de usuario Docente.....	84
Figura 33. Diseño página login	85
Figura 34. Diseño página registro de usuario.....	86
Figura 35. Diseño de registro de usuario	87
Figura 36. Diseño formulario historia clínica	88
Figura 37.. Paleta de colores institucionales	89
Figura 38. Logo del prototipo	90
Figura 39. Isologo del prototipo.....	91
Figura 40. Organización ficheros, proyecto actual	92
Figura 41. Carpeta Back-End.....	93
Figura 42. Declaración de dependencias.....	94

Figura 43. Scripts para compilación del proyecto.....	95
Figura 44. Creación del archivo tsconfig.json	95
Figura 45. Importación ECMAScript.....	96
Figura 46. Integración de base de datos	98
Figura 47. Archivos de rutas del sistema	100
Figura 48. Archivos de controladores del sistema	100
Figura 49. Repositorio en GitHub.....	105
Figura 50. Actividad en el repositorio.....	105
Figura 51. Captura de pantalla de página web de UdeClinick en funcionamiento con el enlace de acceso.....	106
Figura 52. Captura del proyecto en Clouding.io.....	107
Figura 53. Captura del proyecto en Firebase	107
Figura 54. Validación en Postman "Listar usuarios"	110
Figura 55. Validación en Postman "Crear usuarios".....	111
Figura 56. Validación en Postman "Modificar usuarios".....	112
Figura 57. Validación en Postman "Eliminar usuarios".....	113
Figura 58. Validación en Postman "Login Admin"	114
Figura 59. Validación en Postman "Login estudiante"	115
Figura 60. Validación en Postman "Permisos de usuarios"	116
Figura 61. Validación en Postman "Crear Historia Clínica".....	117
Figura 62. Validación en Postman "Borrar Historia Clínica"	118
Figura 63. Correo electrónico institucional.....	122
Figura 64. Evidencia de aplicación de encuesta y reunión con estudiantes de Enfermería.....	123
.....	
Figura 65. Resultados encuesta ítem usabilidad	124
Figura 66. Resultados encuesta.....	125
Figura 67. Resultados encuesta.....	125
Figura 68. Resultados encuesta ítem de portabilidad.....	126
Figura 69. Resultados encuesta.....	126
Figura 70. Resultados encuesta ítem funcionalidad.....	127

ÍNDICE DE TABLAS

Tabla 1. Metodología por prototipos.....	42
Tabla 2. Cronograma.....	44
Tabla 3. Descripción actor Docente	50
Tabla 4. Descripción actor Estudiante	50
Tabla 5. Caso de uso Iniciar sesión.....	50
Tabla 6. Caso de uso Cerrar Sesión	51
Tabla 7. Caso de uso Crear Usuario.....	52
Tabla 8. Caso de uso modificar usuario	53
Tabla 9. Caso de uso eliminar usuario	54
Tabla 10. Caso de uso registrar historia clínica	55
Tabla 11. Caso de uso observar historia clínica.....	56
Tabla 12. Tabla dimensión e indicador de variable	119

Resumen

La historia clínica es el documento elemental y el instrumento principal para el diagnóstico de enfermedades. Organiza una selección de datos valiosos por el personal médico y de Enfermería que, preparados de una forma ordenada y regulada, permiten establecer hipótesis y diagnósticos, en la presencia del paciente o cuando éste ha tenido ingresos anteriores. Además de incluir los datos de la consulta o de las preguntas hechas al paciente y del examen físico, se consignan en ella la discusión diagnóstica, la evolución y el tratamiento de los pacientes.

Según la RAE (Real Academia Española) la definición concreta de historias clínicas es:

Conjunto de los documentos relativos a los procesos asistenciales de cada paciente, que incluye los datos, valoraciones e informaciones sobre su situación y evolución clínica, así como la identificación de los médicos y de los demás profesionales que han intervenido.

Ley 41/2002, de 14 de noviembre, básica reguladora de la autonomía del paciente y de derechos y obligaciones en materia de información y documentación clínica, art. 3. (diccionario panhispánico del español jurídico, 2020)

Las prácticas profesionales son el medio a través del cual se facilita a los estudiantes de Enfermería potenciar y aplicar los diferentes saberes adquiridos (cognitivo, procedimentales, actitudinales, de investigación y valorativas) en el campo real de trabajo y en donde a futuro se desarrollarán como profesionistas (Universidad de sonora, 2017).

El proyecto surge de la necesidad evidenciada, donde los estudiantes del programa de Enfermería al llevar a la práctica sus conocimientos sobre el registro de historias clínicas, no cuentan con un sistema digital que les permita hacer el registro y utilizarlo como herramienta, por lo tanto nuestro proyecto es desarrollar el prototipo de un sistema de información que

permita a los estudiantes de Enfermería conocer y hacer uso de las tecnologías de la información para la trazabilidad y uso correcto de las historias clínicas, esto con el fin de que cuando salgan a un ambiente real logren tener un conocimiento previo para el buen manejo de la información en el área de salud.

Palabras clave: historias clínicas, prácticas, salud, pacientes, Enfermería, estudiantes.

Introducción

Este documento es el informe final de trabajo de grado, como requisito para obtener el título de Ingeniería de Sistemas en la Universidad de Cundinamarca, en donde se ha dado origen la idea de crear un prototipo de ámbito práctico para estudiantes de Enfermería pertenecientes a la misma Universidad. En el presente se exhibe la recolección de datos, las características y el desarrollo del prototipo de historias clínicas digitales, para la práctica de los futuros enfermeros que actualmente cursan su formación profesional en la Universidad de Cundinamarca.

Por lo tanto, se crea el prototipo de un sistema de información, que permite mejorar los procedimientos del estudiante de Enfermería en sus prácticas, proporciona el registro de información para que él mismo pueda reforzar sus habilidades y sus competencias en su ámbito clínico.

Por último, se evidencian las etapas de desarrollo, la recolección de datos, indicios de investigación y la búsqueda de herramientas para el entorno de desarrollo que apoye a la construcción de este.

1. Planteamiento Del Problema

1.1. Descripción Del Problema

Actualmente en el programa de Enfermería de la Universidad de Cundinamarca, no se cuenta con una herramienta tecnológica, para acercar a los estudiantes a la realidad del registro de historias clínicas hospitalarias con fines práctico-académico, para el análisis e interpretación de los registros de Enfermería.

En la proyección profesional del Ingeniero de Sistemas, es inherente el desarrollo de sus competencias comunicativas de manera interdisciplinaria, más aún con profesionales de la salud de la misma Universidad. Desde este punto de vista, las prácticas clínicas constituyen un factor fundamental en el proceso de formación y aprendizaje de los profesionales en Enfermería, permitiendo desarrollar las habilidades clínicas, en el proceso de registro y seguimiento en la evolución médica del paciente a través del registro en una historia clínica de manera digitalizada; esto en razón a un estudio realizado (Dra. Hernández y colaboradores, 2020) donde se midió la capacidad de los estudiantes frente al uso de las nuevas tecnologías, evidenciando falencias en el uso de herramientas tecnológicas.

Según Tessa R. en su investigación de ‘Dificultades percibidas por los estudiantes de Enfermería en su inserción a la práctica clínica de Enfermería Medico quirúrgica’ (2014) afirma que: La práctica clínica tiene un gran impacto educativo, porque significa aplicar los conocimientos adquiridos en las aulas al cuidado del paciente y la familia, debiendo ser el estudiante protagonista de su aprendizaje. Resulta entonces muy importante el papel que desempeña el docente en todo este proceso, brindando herramientas básicas y facilitando la

aplicación y aprendizaje del proceso de formación, método científico utilizado por el profesional de Enfermería para brindar sus cuidados. (p.1).

Los registros de Enfermería en las historias clínicas son la carta de navegación del profesional de la salud, la carencia de estos podría comprometer al enfermero en el incumplimiento legal de los mismos, ya que estos se deben realizar de manera oportuna registrando los procedimientos y actividades de Enfermería, evitando al máximo anotaciones inexactas e incompletas. La ausencia de una herramienta tecnológica dificulta en los estudiantes la práctica del registro de la información según los lineamientos de atención de pacientes.

En la información que se obtuvo de los estudiantes y docentes de la Facultad de Enfermería, se encontró que actualmente las historias clínicas que manejan como actividad académica se está llevando a cabo de forma física mediante formatos en papel; documentos cuya gestión generan incomodidades, retrasos en el acceso y análisis de la información, que en el caso de la salud son de gran importancia, además del impacto ambiental en el uso permanente de papel en las prácticas.

1.1.1. Formulación Del Problema

En un encuentro dialógico de saberes, por una parte, profesionales en Enfermería y por la otra, Ingenieros de Sistemas, en relación con el entendimiento del problema, se plantea lo siguiente:

¿De qué manera se puede mejorar la gestión manual de historias clínicas (carpetas, formatos) en las prácticas clínicas de los estudiantes del programa de Enfermería en la Universidad de Cundinamarca seccional Girardot?

2. Justificación

El proceso de registro y gestión de historias clínicas realizado como actividad práctica por los estudiantes de Enfermería se está llevando de forma manual, ocasionando una incomodidad para los docentes y estudiantes quienes tienen que transportar el papeleo a diferentes lugares, causando en varias situaciones posibles riesgos en la disponibilidad de la información que se está manejando, ya que son documentos de real importancia, se hace necesario la creación de un sistema de información, para la práctica de los estudiantes en el registro de historias clínicas en una plataforma web que les de acceso a los docentes, de plantear ejercicios prácticos, y a los estudiantes, poder aplicar de forma digital evitando la documentación manual, logrando que la conservación de los datos este en excelentes condiciones y cumpliendo con el buen manejo de la información.

Estas mismas herramientas se vienen integrando continuamente en el proceso laboral de los profesionales en Enfermería, es allí en donde la Ingeniería de Sistemas proporciona los conocimientos disciplinares para la creación de nuevos elementos que faciliten la administración de las historias clínicas, optimizando los procesos, facilitando el almacenamiento, registro y control de las historias clínicas.

Actualmente en el programa de Enfermería de la Facultad de Ciencias de la Salud, de la Universidad de Cundinamarca no se cuenta con una herramienta que apoye a los estudiantes a mejorar sus habilidades en el registro, control y manejo de historias clínicas digitales. Por lo tanto, el objetivo del proyecto es desarrollar un prototipo web como apoyo en el proceso de la formación a partir de IV semestre para registro de historias clínicas.

2.1. Delimitación

La plataforma Web se desarrolló para la Universidad de Cundinamarca, seccional Girardot, en la Facultad de Ciencias de la Salud, del programa de Enfermería; como herramienta practica para los estudiantes de IV semestre de Enfermería.

3. Objetivo General

Desarrollar un prototipo de una plataforma web para la práctica en el diligenciamiento de historias clínicas digitales, por parte de los estudiantes en la Universidad de Cundinamarca seccional Girardot, Facultad Ciencias de la Salud, programa de Enfermería.

3.1. Objetivos Específicos

- Organizar el levantamiento de la información clínica necesaria para el desarrollo del prototipo de la plataforma.
- Diseñar y desarrollar los módulos respectivos a partir de la información suministrada, que permitan la gestión de historias clínicas digitales.
- Verificar el comportamiento del prototipo web mediante pruebas.
- Implementar el prototipo de la plataforma Web como recurso práctico-académico.

4. Alcance

Desarrollar un prototipo de sistema de información, que se maneje en un entorno web, con el propósito de que futuros estudiantes de Ingeniería lo retomem para un desarrollo más amplio, que abarque nuevas necesidades y funcionalidades.

El prototipo de este sistema tiene como objetivo ser un espacio donde el estudiante pueda ingresar los datos digitalmente de sus pacientes asignados, así tener una mayor interacción con la tecnología al registro de datos de los pacientes, puesto que al enfermero se le reducirá el tiempo en la recolección de información siendo un apoyo para los diagnósticos médicos; esto también será evaluado por el docente para hacer retroalimentaciones al estudiante. Este proyecto de carácter interdisciplinario permite la integración de saberes de dos programas (Enfermería e Ingeniería de Sistemas) con el firme propósito de aportar soluciones a sus diferentes necesidades, por medio de propuestas factibles para beneficio de toda la comunidad académica.

El prototipo funciona bajo el modelamiento de una metodología de desarrollo basada en el desarrollo de prototipos, por ser un modelo de recolección de requerimientos que se obtienen en la consulta con el usuario final, como es el estudiante ante el manejo de historias clínicas, bajo la dirección y supervisión de sus docentes.

En el área tecnológica se necesita el uso de un computador personal con un sistema operativo basado en Windows 10, Linux o MacOs, el cual debe tener instalado VisualStudioCode como IDE de desarrollo, también instalado el Node para el funcionamiento de Angular, en una versión recomendada de 14.17.6 LTS, un sistema de gestión de base de datos, en este caso Postgres en su versión 13.3.

El proyecto se desarrolla con el uso del Framework Angular para aplicaciones web, desarrollado en TypeScript, de código abierto, que se emplea para crear, sostener y mantener aplicaciones web. Angular se utiliza para el desarrollo de FrontEnd, ya que permite el uso de un modelo vista controlador; el utilizar el lenguaje TypeScript permite una mayor seguridad, porque proporciona la depuración de errores en el código o realizar tareas de mantenimiento. Podemos decir que Angular para este proyecto, es un beneficio debido a que contiene librerías para que la parte visual del usuario sea más dinámica y moderna, permitiendo la escalabilidad del proyecto. Otra herramienta de desarrollo es la utilización de la infraestructura de aplicaciones web llamada Express NodeJS, que es conocida como un marco de aplicación web de BackEnd; se le denomina también como marco de servidor estándar para NodeJs. Este marco de desarrollo nos permite estructurar una aplicación de manera ágil, proporciona funcionalidades como el enrutamiento, opciones para gestionar la utilidad de HTTP y la creación de una API sólida y sencilla.

5. Limitaciones

El prototipo es la fase inicial de una historia clínica, debido a que corresponde a los núcleos temáticos de cuarto semestre de Enfermería. Es decir, hasta donde se desarrolló el sistema, el usuario puede trabajar en un primer momento con el prototipo, pero este es sólo un acercamiento de lo que puede ser el producto final. De esta manera, debido al limitado tiempo de desarrollo y el uso de herramientas no exploradas, como el uso de la herramienta Express, NodeJs, ORM y Tokens; el sistema cumple con una aproximación del formato de historia clínica que manejan en la Facultad de Enfermería en su cuarto semestre.

6. Marco Referencial

6.1. Marco Teórico

6.1.1. Historia Clínica

En el área de salud la historia clínica es un elemento fundamental e importante, pues en la relación entre paciente y médico, se genera un registro de información confidencial y relacionado con la salud del paciente. Esta correlación, es objetivo fundamental de la medicina, se encuentra consagrada en la Ley 23 de 1981, la cual enuncia en su artículo I, cuarto principio:

“[...] La relación médico-paciente es elemento primordial en la práctica médica. Para que dicha relación tenga pleno éxito, debe fundarse en un compromiso responsable, leal y auténtico [...]”. Debido a esto se entiende que todo dato que se conozca del paciente es parte integral de la historia clínica.

En definitiva, la historia clínica son todos los datos de salud que forman un documento privado donde contiene detalles íntimos sobre aspecto físicos, psíquicos y sociales en un paciente, lo cual crea en un registro cronológico las condiciones o el estado de su salud, los sucesos médicos y los procedimientos que ejecuta el personal de la salud, también se integra la relación personal y familiar. Fundamentalmente esta información no puede ser conocida por fuera del entorno asistencial sin la autorización del paciente, o perteneciente de la historia clínica, por la información sensible que puede contener.

La Resolución 1995 de 1999 en su artículo 182 señala, que los Prestadores de Servicios de Salud pueden utilizar medios físicos o técnicos para el registro y conservación de la historia clínica como computadoras y medios magneto ópticos, cuando así lo consideren conveniente, atendiendo lo establecido en la Circular 02 de 1997 expedida por el Archivo General de la Nación, o las normas que la modifiquen o adicionen.

6.1.2. Plataforma Y Servidores Web

Las plataformas Web hacen referencia a un fundamento que permite el ejercicio de determinados módulos simultáneos de hardware y software, estos procedimientos se definen mediante patrones que permiten establecer una arquitectura de hardware y plataforma de desarrollo de software.

Las plataformas digitales también se pueden entender como sistemas que pueden ser programados y personalizados por desarrolladores externos como los usuarios, y de esta forma pueden ser adecuados a incontables necesidades, debido a que su principal objetivo es facilitar la ejecución de sistemas.

También, Los servidores Web se usan para guardar contenidos llenos de información de Internet y proporcionar su recurso de forma persistente y fiable. Al momento de ingresar a una página Web desde un navegador, en realidad está ingresando a un servidor Web que remite o retransmite los datos y/o elementos de dicha página al ordenador. Esto quiere decir que para que una página Web sea asequible en cualquier instante, el servidor Web necesita estar permanentemente en funcionamiento.

6.1.3. Aplicación Web

Las aplicaciones Web son conocidas gracias a su sencillez a la hora de ejecutar en un navegador web, por la independencia de un sistema operativo, y la facilidad de actualizar y mantener las aplicaciones web sin anexar la instalación o distribución de un software, también la posibilidad de obtener miles de usuarios. Es importante entender que una página web o aplicación web puede tener y contener elementos que permiten una comunicación activa entre el

cliente y la información. Esto permite que el mismo acceda a la información de forma interactiva, gracias a que la página le responde a cada acción que se haga.

6.1.4. Prototipo

Se le nombra prototipo a una inicial versión de la solución o fragmento de la solución de un procedimiento construido en un periodo de tiempo y mejorado en varias o siguientes iteraciones para probar y evaluar la eficacia del diseño general que se utiliza para resolver una necesidad o problema.

6.1.5. Api

Application Programming Interface, en español, Interfaz de Programación de Aplicaciones, representa los procesos, funciones y métodos que explican cómo mantener una comunicación entre aplicaciones, permite que los servicios y productos se comuniquen con otros, sin tener que entrar en profundidad de cómo están implementados. Logrando así la simplificación en el desarrollo de aplicaciones, tiempo y dinero.

Las API son un medio simplificado para conectar su propia infraestructura a través del desarrollo de aplicaciones nativas de la nube, pero también le permiten compartir sus datos con clientes y otros usuarios externos. Las API públicas representan un valor comercial único porque simplifican y amplían la forma en que se conecta con sus partners y, además, pueden rentabilizar sus datos (un ejemplo conocido es la API de Google Maps). (RedHat, 2021)

6.1.6. Dominio

El dominio de Internet es un nombre único que identifica una subregión de Internet. El propósito principal del nombre de dominio y el sistema de nombres de dominio (DNS) en Internet es convertir la dirección IP de cada activo en la red en términos fáciles de encontrar y memorables. Esta abstracción hace posible que cualquier servicio (de red) se mueva de una

ubicación geográfica a otra en Internet, incluso si este cambio significa que tendrá una dirección IP diferente.

6.1.7. Hosting

El hosting o alojamiento Web, es un servicio en el que se puede desplegar todo el contenido necesario de un sitio o aplicativo (texto, imágenes, bases de datos, correo electrónico, entre otros), y cualquier usuario de Internet puede acceder desde un enlace en cualquier momento.

6.1.8. Servidor Web

El servidor web es el Hardware de almacenamiento que contempla un software de servicio para dar al usuario la información que solicite en una página Web. En otras palabras, es un recurso de servicio que aloja y distribuye a través de internet información a la que pueden acceder múltiples usuarios.

6.1.9. Api Rest

Representational State Transfer o Transferencia de Estado Representacional, según RedHat (2020) es:

Una API de REST, o API de RESTful, es una interfaz de programación de aplicaciones (API o API web) que se ajusta a los límites de la arquitectura REST y permite la interacción con los servicios web de RESTful. El informático Roy Fielding es el creador de la transferencia de estado representacional (REST).

En pocas palabras, una API REST es un conjunto de pautas que determinan como las aplicaciones o dispositivos pueden conectarse entre sí, permitiendo que la aplicación acceda a las bases de datos desde diferentes servidores.

En el sitio web de BBVA API_Market define REST como:

Buscando una definición sencilla, REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. ("API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos", 2021).

6.2. Marco Conceptual

6.2.1. Estrategias Para El Aprendizaje

De acuerdo con Weinstein y Mayer afirma que:

"las estrategias de aprendizaje pueden ser definidas como conductas y pensamientos que un aprendiz utiliza durante el aprendizaje con la intención de influir en su proceso de codificación" Wein stein y Mayer, 1986, p. 315).

En efecto, logramos expresar que las estrategias de aprendizaje organizan acciones conscientes y premeditadas que rigen las operaciones a alcanzar para conseguir explícitas metas de aprendizaje.

Terceros escritores (p.ej., Beltrán, García-Alcañiz, Moraleda, Calleja y Santiuste, 1987; Beltrán, 1993) las definen como actividades u operaciones mentales empleadas para facilitar la adquisición de conocimiento.

Según Carles Monereo (coord.), Montserrat Castelló, Mercè Clariana, Montserrat Palma, María L. Pérez. Desde nuestro punto de vista, la programación de los contenidos mediante el diseño de diversas unidades didácticas centradas en temas globalizadores que conecten con la realidad cotidiana del escolar puede contribuir a la integración curricular de las estrategias de

aprendizaje. (Estrategias de enseñanza y aprendizaje Formación del profesorado y aplicación en la escuela, 1994)

Podemos decir que las estrategias de aprendizaje orientan, aconsejan y ayudan a un estudiante a indagar la manera más práctica de educarse.

6.2.2. Aprendizaje Autónomo

El aprendizaje autónomo es un proceso donde el estudiante autorregula su aprendizaje y toma conciencia de sus propios procesos cognitivos y socioafectivos. Esta toma de conciencia es lo que se llama metacognición. El esfuerzo pedagógico en este caso está orientado hacia la formación de sujetos centrados en resolver aspectos concretos de su propio aprendizaje, y no sólo en resolver una tarea determinada, es decir, orientar al estudiante a que se cuestione, revise, planifique, controle y evalúe su propia acción de aprendizaje (Martínez,2005).

El proceso de enseñanza tiene como objetivo desarrollar conductas de tipo metacognitivo, es decir, potenciar niveles altos de comprensión y de control del aprendizaje por parte de los alumnos (Martí, 2000).

Podemos decir que el aprendizaje autónomo es una forma de educarse por cuenta propia, donde el estudiante va estableciendo sus objetivos, procedimientos, recursos, evaluación y momento de estudio, sabiendo sus necesidades actuales de formación, en la cual por sí mismo, puede y debe aportar recursos para revitalizar el aprendizaje y darle importancia.

6.2.3. Aprendizaje Basado En Competencias

En la actualidad, se ha retomado con enorme fuerza el enfoque de aprendizaje basado en competencias en la enseñanza universitaria y se está iniciando en la educación no universitaria (Sarramona, Domínguez, Noguera y Vázquez, 2005). Este modelo está en la base del proceso europeo de convergencia. El modelo de competencias intenta desarrollar un aprendizaje vinculado con el ejercicio de la profesión y, por tanto, con un cierto carácter profesionalizador, aunque siempre desde la consideración que los estudiantes están en un periodo formativo académico, y es aquí donde se evalúan las competencias y su aprendizaje.

El aprendizaje basado en competencias, en adelante ABC, consiste en desarrollar las competencias genéricas o transversales (instrumentales, interpersonales y sistémicas) necesarias y las competencias específicas (propias de cada profesión), con el propósito de formar a las personas sobre los conocimientos científicos y técnicos, su capacidad de aplicarlos en contextos diversos y complejos, integrándolos con sus propias actitudes y valores en un modo propio de actuar desde el punto de vista personal y profesional. Por ello, no se trata sólo de mejorar la preparación profesional de los estudiantes para ocupar un puesto laboral determinado, sino, principalmente, ofrecer una formación más sólida, más firme, más acorde con el enfoque de lo que debe ser una buena formación universitaria, que ayude a los estudiantes a saber, a saber hacer, a convivir y a ser. Ello requiere el desarrollo de competencias que van más allá del mero conocimiento, y pone el énfasis en una integración entre el contenido de lo que se aprende con su integración en la estructura mental de cada estudiante, logrando que ese aprendizaje sea más duradero y significativo (Villa y Poblete, 2007)

Además a esto, podemos decir que el aprendizaje basado en competencias es un enfoque que da la educación, donde se concentra en la evidencia de los resultados obtenidos, es decir, el dominio de los temas por parte del estudiante, que hace referencia explícitamente en el progreso

de este a través de planes de estudio. En estos modelos de aprendizaje, el estudiante se relaciona con diversos contenidos, ya sea habilidades o conceptos que se miden con el tiempo, donde se les permite continuar hasta que demuestren un total dominio en los resultados de aprendizaje.

6.2.4. Recursos Educativos Digitales

Los recursos educativos son un conjunto de materiales o herramientas organizadas que resultan útiles para cumplir un objetivo de aprendizaje, formación, capacitación o la instrucción y enseñanza. Así formar un proceso educativo, con esto el educador puede detallar o explicar mejor un determinado tema a sus estudiantes, esto quiere decir que los recursos educativos favorecen al docente a cumplir su función de formador y educativa utilizándolos como aporte de información, que sirven para poner en práctica lo aprendido y en muchos casos sirven de guía para los alumnos.

La UNESCO (2002) se refiere a los Recursos Educativos digitales Abiertos (REdA) como la provisión abierta de recursos educacionales mediados por las TIC para consultas, uso y adaptación por una comunidad de usuarios con propósitos no comerciales. Los REdA son elementos o materiales en formato digital que se distribuyen de forma gratuita para ser utilizados en el proceso enseñanza-aprendizaje, tanto por profesores y estudiantes como por investigadores. La distribución de estos recursos se realiza mediante licencias abiertas o Creative Commons para facilitar su uso, adaptación, derivación y redistribución pública. Estos recursos propician su uso, actualización y personalización con fines formativos, desde los cuales se pueden identificar rasgos e intenciones de la didáctica del profesor en favor de la comprensión y construcción de saberes por parte del estudiante. Bajo las licencias abiertas, el usuario (estudiante) puede generar sus propias versiones y propuestas de recursos y contenidos.

6.3. Marco Ético Y Legal De Ingeniería

COPNIA (Consejo Profesional Nacional de Ingeniería) creado mediante la ley 94 de 1937, es la entidad pública encargada de controlar, inspeccionar y vigilar el ejercicio de la Ingeniería y profesiones afines y auxiliares en general, en el territorio nacional.

El Código de Ética Profesional constituye el catálogo de conductas profesionales que se exigen, se prohíben o que inhabilitan a los ingenieros en general y a sus profesionales afines o auxiliares. Dicho catálogo es el marco legal del comportamiento profesional del ingeniero, por lo que el ejercicio profesional debe estar ajustado a sus disposiciones. (Consejo Profesional Nacional de Ingeniería, 2021)

6.4. Marco Teórico Médico

Consulta por enfermedad general, según varios autores pertenecientes a varias ordenaciones como lo es: la organización panamericana de la Salud / Organización Mundial de la Salud, Ministerio de Protección Social, Cooperativa de Hospitales de Antioquia – COHAN y CAPRECOM quienes realizaron un Manual Operativo Unidad Móvil de Salud denominado REGISTROS CLÍNICOS Y ADMINISTRATIVOS. Hacen referencia a los registros iniciales de un paciente, el orden que debe de llevar la historia clínica, que es la siguiente:

A. Anamnesis

“La anamnesis es el proceso de la exploración clínica que se ejecuta mediante el interrogatorio, para identificar personalmente al individuo, conocer sus dolencias actuales, obtener una retrospectiva de él y determinar los elementos familiares, ambientales y personales que sean relevantes.” (Rodríguez García & Rodríguez Pupo, 2021)

1. Identificación
 - 1.1. Nombre Completo.
 - 1.2. Edad y Sexo.
 - 1.3. Documento de Identidad.
 - 1.4. Afiliación al Sistema General de Seguridad Social en Salud - SGSSS
- B. Motivo de Consulta: Razón por la cual el paciente acude a la consulta.
- C. Enfermedad Actual: Síntomas principales, inicio, intensidad, frecuencia.

Tratamientos recibidos y quien se los ordenó.

D. Revisión por sistemas: En los sistemas no relacionados con la enfermedad actual, si se indagó por síntomas presentados se podrá escribir Normal, en los relacionados debe dejar escrito en forma clara los hallazgos.

E. Antecedentes Patológicos: Quirúrgicos, médicos, traumáticos, medicamentosos e inmunizaciones. tabaquismo, consumo de medicamentos, alergias.

F. AGO: En mujeres, con la información detallada de su historia ginecológica y obstétrica; FUM, número de gestaciones, partos, nacidos vivos, abortos/óbitos y cesáreas; FUP; última citología cervicovaginal según esquema actual 1-1-3.

6.5. Estado Del Arte

6.5.1. Zeus Salud

Es un software administrativo que incorpora todas las operaciones de una IPS, permite acceder a la información de: admisiones, facturación e historias clínicas, caja de copagos, farmacia, Enfermería entre otros módulos que conforman este sistema.

Este sistema cumple con el modelo integral y la normatividad de atención en la salud, su sistema permite manejo en línea o sin conexión a internet, permite acceder a él en puntos donde la conectividad no es estable con el servidor; en él se puede monitorear y hacer seguimiento a los pacientes hospitalizados o quienes ingresan por un motivo de salud. Y por último, tiene la facilidad de generar informes, indicadores y circulares para tener control de la información de los registros médicos.

6.5.2. Heon Health On Line

Una compañía colombiana enfocada en desarrollar e integrar soluciones tecnológicas competitivas y de alta calidad para el Sector Salud. En el texto: ("Heon: Heon Es")

Es un sistema de historias clínicas digitales para hospitales, clínicas e IPS que provee un modelo administrativo y operacional en el área de salud, es una solución basada en la metodología SOAP ((Subjetivo, Objetivo, Análisis, Plan de tratamiento) permite el registro y procedimiento de toda la información clínica de los pacientes, junto con herramientas especializadas en la inteligencia de negocios, puede realizar una correcta gestión en la información. La herramienta es parametrizable y personalizable de acuerdo con la necesidad médica.

6.5.3. Historia Clínica Electrónica Desde Un Dispositivo Móvil

En otras instituciones educativas en Colombia, crearon un sistema similar para dispositivos móviles.

En la Facultad de Ingeniería de la Universidad Tecnológica de Pereira, Colombia, se realizó un proyecto de historias clínicas electrónicas para dispositivos móviles construido en Java en la plataforma de desarrollo Eclipse, en el 2015 por Guillermo Roberto Solarte Martínez.

6.5.4. Software Médico

Empresa especializada en tecnología de la información y comunicación (TIC) dirigida al sector salud.

Con más de 8 años de experiencia esta empresa crea y diseña software para el área de salud, personalizado según el criterio médico, adaptable a la necesidad de cada especialista; permite gestionar tareas, control de citas, historias clínicas, RIPS, facturación reportes, ajustado a la normatividad vigente.

La ventaja de este sistema es que se crea a partir de la profesión del médico tratante, es decir permite enfocarse en la especialización de este para atender de manera más enfocada al paciente. Permite a las instituciones prestadoras de servicios médicos IPS incrementar y automatizar sus servicios por medio de este sistema, funciona de manera online, son creadas en WordPress, SEO y procesamiento web, con personal capacitado para recibir los requerimientos que solicita el área a necesitar el sistema.

Cuenta con un módulo de Diseño- Marketing digital, donde contiene piezas gráficas, imagen corporativa, membresías, firmas digitales y estrategias de publicidad digital que se pueden implementar en redes sociales también en Google ads, cuentan con el posicionamiento de la marca para obtención de resultados, todo esto se incluye en su portafolio de servicios.

6.6. Marco Tecnológico

6.6.1. Visual Studio Code

Editor de código fuente desarrollado por Microsoft para diferentes sistemas operativos como Windows Linux y MacOs. Es un software libre que incluye soporte para depuración, revisión de sintaxis, integración de Git, entre otros.

6.6.2. Node

Node es un entorno en tiempo de ejecución multiplataforma en JavaScript para la capa de servicio, que se orienta en eventos asincrónicos, con enfoque de Entrada-Salida de datos en una arquitectura orientada a eventos y se basa en el motor v8 (entorno de ejecución para JavaScript creado para Google Chrome).

Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web. En el texto: ("NodeJS: Wikipedia", 2021).

6.6.3. Postgres

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que utiliza y amplía el lenguaje SQL combinado con muchas características que almacenan y escalan de forma segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central. En el texto: ("PostgreSQL: About", 2021).

6.6.4. PgAdmin y PgShell

Existen dos formas de administrar la Base de Datos de PostgreSQL, mediante un Sistema gestor de base de datos, para una interfaz más amigable con el usuario; Y por medio de comandos desde una terminal.

PgAdmin es una plataforma de gestión y desarrollo de código abierto para gestionar y gestionar bases de datos PostgreSQL. Se emplea para responder a las necesidades de los usuarios para administrar la base de datos.

PgShell es la utilización de métodos y comandos para ejecutar y administrar la PostgreSQL, por medio de una interfaz que permite al usuario interactuar con el sistema,

procesando ordenes que se le indican, los Shells <https://www.npmjs.com/package/pg> los incorpora un lenguaje de programación más metódico para controlar los archivos y procesos, además de iniciar y controlar según la necesidad del usuario.

6.6.5. Angular

Angular es un framework (marco de trabajo) para desarrollar aplicaciones web desplegado en TypeScript de código abierto, sostenido por Google, se enfoca en la extensión de aplicaciones web con un estilo de arquitectura de software denominado modelo vista controlador (MVC) con el fin de que el desarrollo junto con las pruebas sea sencillo para su implementación, ya que se basa en clases de tipo “Componentes” para ser reutilizados en proyectos medianos y grandes. Para manejar Angular es necesario el manejo y la implementación de NodeJs ya que todo su funcionamiento Back-end depende del mismo.

6.6.6. Express

Express es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles. En el texto: ("Express: espressjs", 2021)

De este modo se logra decir que Express es un marco web (WF) o marco de aplicaciones web (WAF), esto quiere decir que, está diseñado para el desarrollo de aplicaciones, servicios y recursos que los usuarios puedan utilizar accediendo a un servidor conectado a una red informática.

Los marcos Web proporcionan una forma estándar de crear e implementar aplicaciones Web en la Word Wide Web. Los marcos web tienen como objetivo automatizar la sobrecarga asociada con las actividades comunes realizadas en el desarrollo web. En el texto: ("Web framework - Wikipedia", 2021).

6.6.7. *GitHub*

Es una plataforma gratuita que permite la gestión de proyectos y control de versiones de código. Permite la creación de herramientas y trabajos colaborativos, alojando todo en la nube.

Millones de desarrolladores y empresas construyen, envían y mantienen su software en GitHub, la plataforma de desarrollo más grande y avanzada del mundo. ("Build software better, together", GitHub, 2021)

6.6.8. *Orm*

ORM (Object Relational Mapping) en español el mapeo objeto-relacional, es una herramienta, modelo y técnica de programación que permite mapear estructuras de bases de datos relaciones, como PostgreSQL, Oracle, MySQL, entre otros. La estructura de la base de datos relacional se relaciona con entidades lógicas definidas en ORM, de tal forma que las acciones, operaciones CRUD (create, read, update, delete) a ejecutar en la base de datos se realicen por medio de ORM.

6.6.9. *TypeORM*

En el sitio oficial <https://typeorm.io/#/> hace referencia a esta librería como:

TypeORM es un ORM que se puede ejecutar en plataformas NodeJS, Browser, Cordova, PhoneGap, Ionic, React Native, NativeScript, Expo y Electron y se puede usar con TypeScript y JavaScript (ES5, ES6, ES7, ES8). Su objetivo es siempre admitir las últimas características de JavaScript y proporcionar características adicionales que lo ayuden a desarrollar cualquier tipo de aplicación que use bases de datos, desde pequeñas aplicaciones con algunas tablas hasta aplicaciones empresariales a gran escala con múltiples bases de datos.

TypeORM admite patrones de registro activo y mapeado de datos, a diferencia de todos los demás ORM de JavaScript que existen actualmente, lo que significa que puede escribir aplicaciones de alta calidad, poco acopladas, escalables y mantenibles de la manera más

productiva. ("TypeORM - Amazing ORM for TypeScript and JavaScript (ES7, ES6, ES5). Supports MySQL, PostgreSQL, MariaDB, SQLite, MS SQL Server, Oracle, WebSQL databases. Works in NodeJS, Browser, Ionic, Cordova and Electron platforms.", 2021)

6.6.10. Firebase

Firebase es una plataforma móvil creada por Google. Su función principal es desarrollar y promover rápidamente la creación de aplicaciones de alta calidad para aumentar la base de usuarios y obtener más ingresos. La plataforma se carga en la nube y es adecuada para diferentes plataformas como iOS, Android y Web. Contiene varias funciones, por lo que cualquier desarrollador puede combinar y ajustar la plataforma para satisfacer sus necesidades.

Firebase te da las herramientas para programar apps de alta calidad, aumentar tu base de usuarios y ganar más dinero. Nos encargamos de los aspectos básicos para que puedas monetizar tu negocio y concentrarte en tus usuarios. En el texto: ("Firebase", 2021)

6.6.11. Clouding.io

Hosting de Servidores Cloud VPS en la nube, Data Center en España, discos SSD, con Linux o Windows y facturados por hora. ("VPS Cloud: Tu Servidor Virtual en España - ¡Pruébanos!", 2021)

6.6.12. Postman

Es una herramienta que ayuda a los desarrolladores Web, ya que permite realizar peticiones de tipo HTTP a cualquier API. Cuando se utiliza Postman para la programación y las pruebas en tiempo real, nos brinda la posibilidad de verificar que el desarrollo está funcionando correctamente.

6.7. Marco De Trabajo

6.7.1. Marco De Trabajo Scrum

Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos. En el texto (Schwaber and Sutherland, 2020).

Con base en esto se entiende que Scrum es un marco de trabajo para el desarrollo y el mantenimiento de productos complejos, que permite compromiso colaborativo entre equipos. Es un proceso de gestión que reduce la complejidad en el desarrollo de producto para así satisfacer las necesidades del cliente.

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. (Schwaber and Sutherland, 2020)

Para esto se utiliza Microsoft Azure DevOps, que es un servicio de computación en la nube creado por Microsoft para construir, probar, desplegar y administrar Scrum.

Para el manejo e implementación de scrum se realizó los siguientes eventos:

- Organización del Backlog: En este evento se realiza una recolección de información con el propietario del producto, en la que está encargado el Product Owner; donde se especifica la visión del producto y los requerimientos.
- Planificación Sprint: En esta primera reunión se definieron los aspectos funcionales, objetivos, riesgos para cada futuro sprint, plazos de entrega, entre otros.

Posteriormente se define cada cuanto se hará una junta con el equipo y el jefe del proyecto para

explicar cómo se desarrollará cada punto, se evaluarán los cambios, la toma de decisiones, mejoras y más factores.

- **Sprint:** El sprint es una interacción en tiempo real que tiene el quipo Scrum, donde se trabaja de forma conjunta para finalizar una planificación previa, o incremento como se le denomina en la metodología.

En este caso, los Sprint son cada 8 días de 2h, debido a que no hay disponibilidad para una reunión diaria como se plantea en el marco de trabajo, supliendo esta actividad con varias horas de trabajo se logra concretar una primera fase del producto.

- En cada Daily Sprint (Sprint diario), se debe dar respuesta a las 3 preguntas: ¿qué hice ayer?, ¿qué voy a hacer hoy?, ¿he tenido algún tipo de problema?

- **Sprint Review:** Ocurre al final de cada sprint, así se puede inspeccionar el incremento y adaptar el Product Backlog en caso de que sea necesario.

- Dentro de la metodología Scrum se manejan 5 fases con 19 procesos para el desarrollo de un proyecto. En donde también se adapta las fases de desarrollo de software.

- **Sprint Retrospective:** Siendo el ultimo evento del Sprint, es el momento para que el equipo se inspeccione a sí mismo, y crear un plan de mejora para implementarlo en el siguiente Sprint.

6.7.1.1. Identificación De Scrum Master Y Interesados Del Proyecto Interesados Del Proyecto

Luis Eduardo Orjuela Endo

Docente universitario especialista en Informática para la Gestión de Proyectos, Administrador de Sistemas de Información y tecnólogo en sistemas de datos.

Lady Carolina Diaz Jurado

Enfermera profesional Magister en Salud Laboral y Seguridad Industrial, Especialista en Administración de la Informática Educativa, Conferencista Nacional Internacional sobre Nuevas Prácticas Educativas. Docente tiempo completo ocasional Universidad de Cundinamarca, primer docente embajadora del programa de Enfermería en la Universidad del Chubut Patagonia Argentina, líder del espacio radial esencial en salud, Enfermeros en sintonía, del programa de Enfermería Universidad de Cundinamarca.

Mateo Galvis Diaz Estudiante de Ingeniería de Sistemas de la Universidad de Cundinamarca. Habilidades en desarrollo de software Back-end y Front-end, bases de datos y Diseño de interfaz.

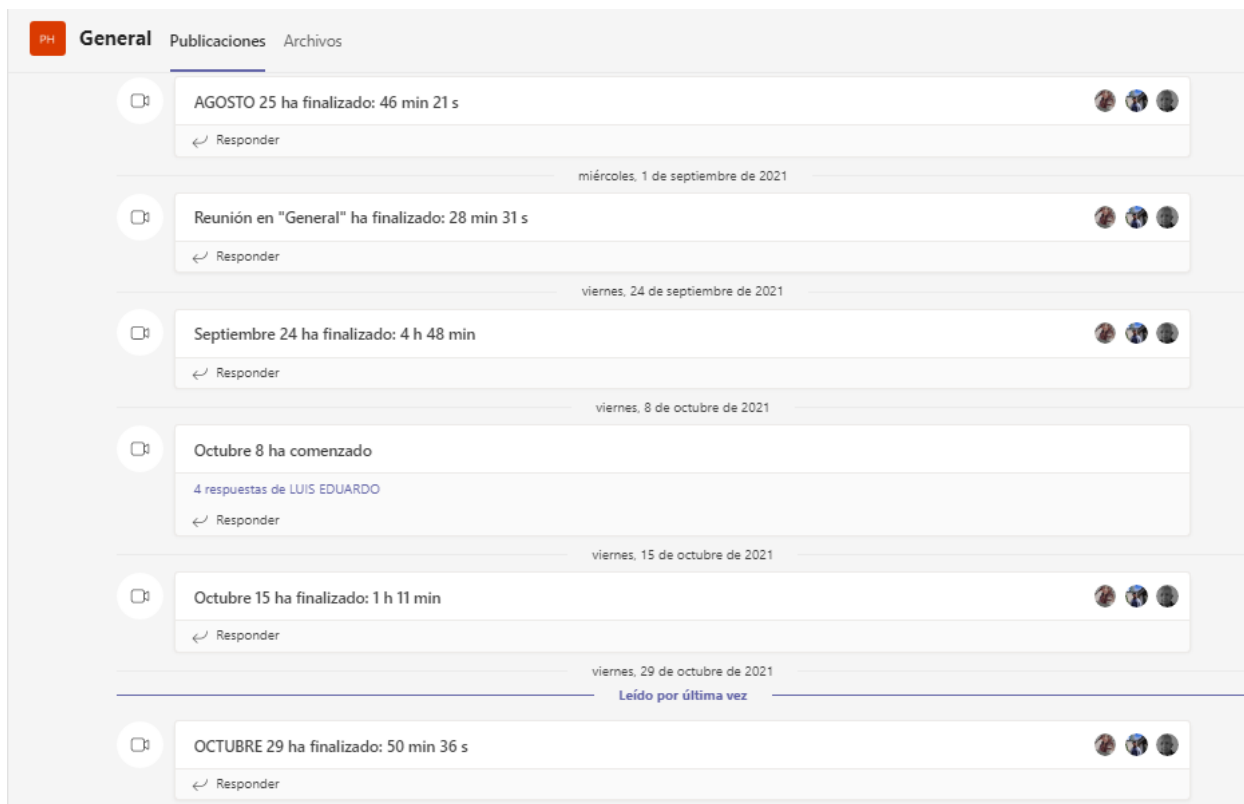
Yezny Yormary Garcia Marquez Estudiante de Ingeniería de Sistemas de la Universidad de Cundinamarca. Habilidades de diseño gráfico, desarrollo web, diseño de imágenes y contenido digital.

6.7.1.2. Formación Del Equipo Scrum

Scrum Master: Consta de dos funciones principales: gestionar el proceso Scrum y ayudar a eliminar impedimentos que afecten la entrega del producto. En este caso para esa función se cuenta con el Docente Luis Eduardo Orjuela Endo.

Product Owner: Docente Lady Carolina Diaz Jurado, encargada de traducir las necesidades y requerimientos del cliente (usuario final) en el producto que debe crear el equipo

del proyecto. Es decir, es quien tiene conocimientos de lo que el usuario necesita, es quien prioriza, aprueba y describe las necesidades en las historias de usuario o Product Backlog.



Usuarios: Estudiantes de Enfermería, usuario final. Quienes transmiten las necesidades al Product Owner para dar construcción al sistema.

Equipo: Personas con los conocimientos disciplinares y complementarios, quienes van a desarrollar el proyecto, ejecutando las actividades según las historias de usuario para dar cumplimiento con el Sprint. El equipo en este caso está conformado por: Mateo Galvis Diaz y Yezny Yormary García.

6.7.1.3. Desarrollo De Daily Sprints Y Sprint Review

DAILY SPRINTS

En este caso los Daily Sprints se hicieron por parte del equipo de desarrollo, donde se reunieron una vez por día 15min para realizar la planeación según el marco de trabajo, las reuniones se llevaron a cabo por medio de la plataforma TEAMS.

SPRINT REVIEW

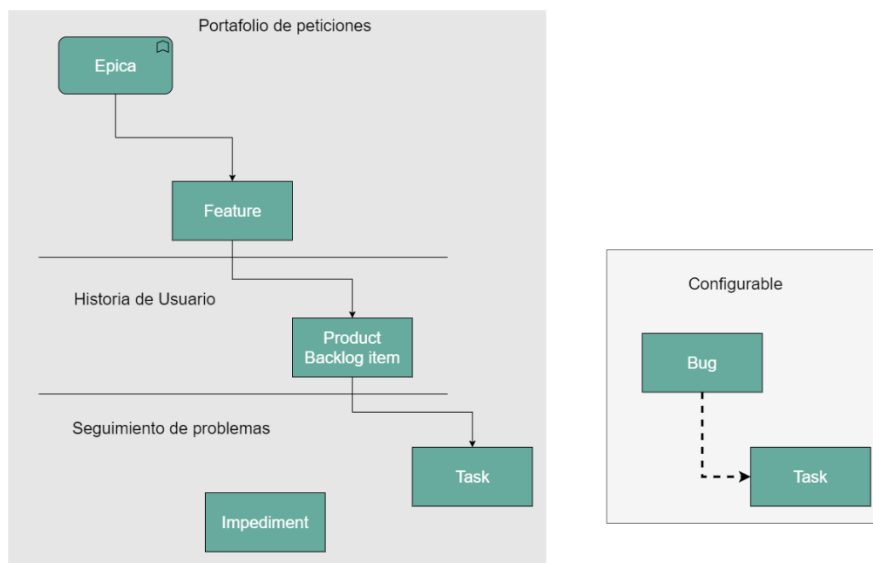
Para el Sprint Review se realizó mensualmente una reunión con el Scrum Master, comentando avances e impedimentos que se presentaban a medida que se desarrolló el prototipo se llevaron a cabo por medio de la plataforma TEAMS.

Fuente 1. Captura de Teams

Figura 1. Captura reuniones Sprint Review

6.7.1.4. Desarrollo De Épica(S)

Las épicas se presentan como el primer contacto con el cliente, donde se hace la recolección de las necesidades del usuario y, por lo tanto, se tienen que pasar un proceso de relación de actividades para desglosar la épica y así dar cumplimiento con una mejor organización y jerarquización de cada épica, para obtener entendimiento y seguimiento.



Fuente 2. Elaboración propia adaptada de ("Artefactos de plantilla de proceso de scrum - Azure Boards", 2021)

Figura 2. Proceso Scrum

Debido a que la herramienta AzureDevOps no permite asignar mas de una persona, se toma que el “no asignado”, como la relacion de que los dos desarrolladores estan al tiempo en la misma actividad o característica, en este caso las épicas.

Inicialmente en el primer contacto se desarrollaron las siguientes épicas:

ID	Title	Assigned To	State	Area Path
96	👑 Prototipo de Historias Clínicas Digitales	👤 Unassigned	● In Progress	Prototipo Historias Clínicas Di...
97	👑 Acceso a plataforma en un navegador	👤 Unassigned	● New	Prototipo Historias Clínicas Di...

Fuente 3. Elaboración propia adaptada de ("Artefactos de plantilla de proceso de scrum - Azure Boards", 2021)

Figura 3.. Épicas creadas para el prototipo en AzureDevOps

Estas épicas fueron divididas en Características:

ID	Title	Assigned To	State	Area Path
75	Problema Actual	Unassigned	New	Historias Clínicas Digitales
76	Perfil Del Paciente	Unassigned	New	Historias Clínicas Digitales
70	Paginas Web Del Sitio	Unassigned	New	Historias Clínicas Digitales
78	Observaciones Generales	Unassigned	New	Historias Clínicas Digitales
73	Motivo De Consulta	YEZNY YORMARY GARCIA ...	New	Historias Clínicas Digitales
71	Ingreso De Datos De Identificacion Del Paciente	YEZNY YORMARY GARCIA ...	New	Historias Clínicas Digitales
79	Describir: Las estructuras de datos de la base de datos	MATEO GALVIS DIAZ	New	Historias Clínicas Digitales
77	Antecedentes	Unassigned	New	Historias Clínicas Digitales
95	Acceso de Docentes y estudiantes	Unassigned	New	Historias Clínicas Digitales

Fuente 4. Captura de pantalla plataforma AzureDevOps para el manejo del proyecto

Figura 4.. Características creadas para el prototipo en AzureDevOps

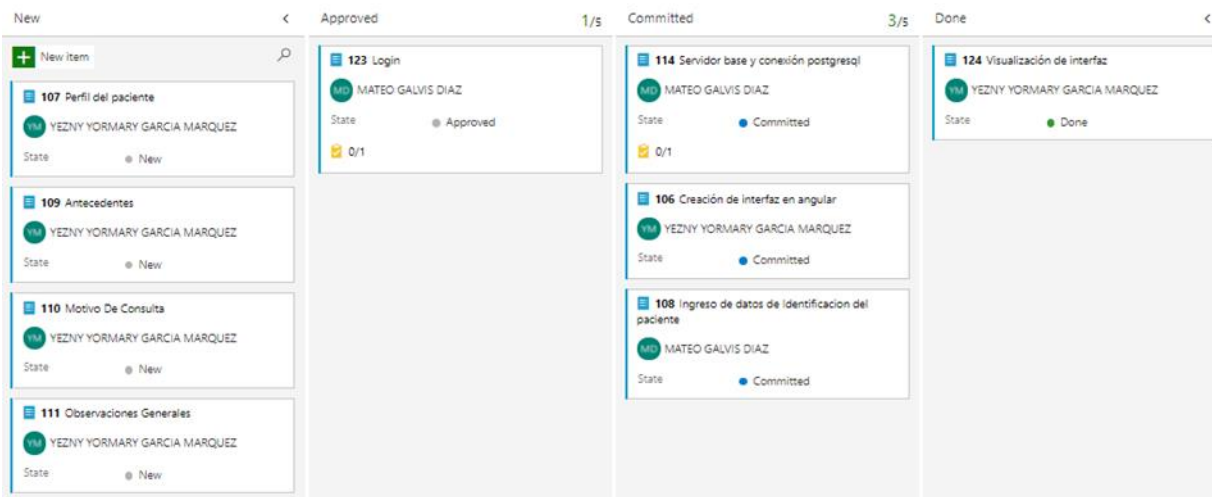
Según lo anterior, cada característica se desglosó en uno o varios Product Backlog para que el equipo lo desarrolle.

ID	Title	Assigned To	State	Area Path
107	Perfil del paciente	YEZNY YORMARY GARCIA ...	New	Prototipo Historias Clínicas Di...
108	Ingreso de datos de Identificacion del paciente	MATEO GALVIS DIAZ	New	Prototipo Historias Clínicas Di...
112	Paginas Web Del Sitio	YEZNY YORMARY GARCIA ...	New	Prototipo Historias Clínicas Di...
114	Servidor base y conexión postgresql	MATEO GALVIS DIAZ	New	Prototipo Historias Clínicas Di...
113	Identificación de usuario y contraseña	MATEO GALVIS DIAZ	New	Prototipo Historias Clínicas Di...
111	Observaciones Generales	YEZNY YORMARY GARCIA ...	New	Prototipo Historias Clínicas Di...
110	Motivo De Consulta	YEZNY YORMARY GARCIA ...	New	Prototipo Historias Clínicas Di...
109	Antecedentes	YEZNY YORMARY GARCIA ...	New	Prototipo Historias Clínicas Di...
106	Creación de interfaz en angular	YEZNY YORMARY GARCIA ...	New	Prototipo Historias Clínicas Di...
123	Login	MATEO GALVIS DIAZ	New	Prototipo Historias Clínicas Di...
124	Visualización de interfaz	YEZNY YORMARY GARCIA ...	New	Prototipo Historias Clínicas Di...
125	Acerca De	YEZNY YORMARY GARCIA ...	New	Prototipo Historias Clínicas Di...
126	Información de usuario	MATEO GALVIS DIAZ	New	Prototipo Historias Clínicas Di...

Fuente 5. Captura de pantalla plataforma AzureDevOps para el manejo del proyecto

Figura 5. Product BackLog creados para el prototipo en AzureDevOps

Creación de la lista priorizada de pendientes del producto, es decir un Board/tablero de trabajo que nos permite ver en tiempo real las actividades que está realizando cada usuario de la plataforma.



Fuente 6. Captura de pantalla plataforma AzureDevOps para el manejo del proyecto

Figura 6. Tablero creado para el prototipo en AzureDevOps

6.8. Marco Metodológico

6.8.1. Metodología Por Prototipos

Se entiende como prototipo la representación de un objeto diseñado para una demostración de cualquier tipo de servicio, producto, software, entre otros. Se puede decir, que es un mecanismo para identificar los requerimientos y aquellos aspectos que serán visibles para el cliente o usuario. Siendo el primer molde es la demostración de lo que se quiere realizar.

De acuerdo con lo anterior, se decide aplicar el modelo o metodología por prototipos, porque nos permite dar una idea más completa de cómo podría ser el producto final. Nos da un boceto de un producto adaptable al cambio y la identificación de elementos innecesarios.

La metodología por prototipos o prototipado se involucra en la mejora continua, interactivo, enfocado en diseñar, implementar, medir y ajustar un plan. Esta metodología se utiliza para dar al usuario final una vista precedente del software que se divide en las siguientes etapas:

Tabla 1. Metodología por prototipos

Etapa	Actividades
Investigación preliminar	<ul style="list-style-type: none"> ▪ Análisis del entorno ▪ Definición de los requerimientos funcionales ▪ Elaboración de casos de uso ▪ Definición de herramientas de desarrollo
Análisis	<ul style="list-style-type: none"> ▪ Diagrama de actividades ▪ Diagrama de Entidad-Relación ▪ Modelo Vista controlador ▪ Modelo de Evaluación ▪ Definición de las herramientas de desarrollo
Diseño	<ul style="list-style-type: none"> ▪ Diagramación en esquemas de prototipo ▪ Diseño del prototipo
Construcción de prototipo	<ul style="list-style-type: none"> ▪ Desarrollo del prototipo ▪ Programación de la primera versión del prototipo
Implementación	<ul style="list-style-type: none"> ▪ Producto Final
Validaciones	<ul style="list-style-type: none"> ▪ Validaciones hechas por el usuario verificando funcionalidad.

7. Apropiación Metodología De Desarrollo Por Prototipos

7.1. Etapa: Investigación Preliminar

7.1.1. Análisis Del Entorno

Crear un prototipo de plataforma web que le proporcione a los estudiantes su práctica académica para facilitar en el futuro su experiencia laboral. En cuanto a esto se dispone de un

tiempo estimado de desarrollo aproximado de siete meses, donde se le da una planeación metodológica para el cumplimiento de este.

7.1.2. Reglas De Negocio

En el momento de hacer la recolección de requerimientos, es de gran importancia identificar y concretar las reglas de negocio, puesto que permite con el cliente o interesado, definir los parámetros que el sistema debe de cumplir, esto permite que el prototipo cumpla con lo acordado, así entregarlo con lo estrictamente pactado y de acuerdo con las fechas establecidas, que se definieron en levantamiento de información. Por lo tanto, brindar una mayor claridad con los involucrados en el proyecto.

1. Para el desarrollo del prototipo de esta plataforma web se toma como apoyo el formato estándar que se emplea en la Universidad de Cundinamarca en la seccional Girardot, para atender a sus pacientes en la clínica donde la Universidad tiene convenio, es decir, la Clínica San Rafael Dumian. En este formato podemos encontrar la información básica del paciente, donde encontraremos fecha de nacimiento, sexo, religión, entre otros; en el siguiente segmento del formato vamos a encontrar el motivo de consulta, problema actual, información que solicitó la docente que debía estar en el sistema.

2. Los usuarios solo son estudiantes y docentes.

7.1.3. Cronograma

Tabla 2. Cronograma

Fase	Actividad	Tiempo en Ejecución									
		Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	
Documentación, recolección de información y refinamiento de requisitos	Investigación y documentación con Facultad de Enfermería	■	■								
	Análisis del entorno										
	Definición de los requerimientos funcionales										
	Elaboración de casos de uso										
	Definición de herramientas de desarrollo										
Modelado	Diagramación UML		■								
	Definición MVC										
	Definición Herramientas de desarrollo										
Diseño	Diagramación en esquemas del prototipo		■								
	Diseño del prototipo			■							
	Identificación de características										
	Desarrollo				■	■	■				

Construcción del prototipo	Identificación de funcionalidades	
	Programación versión 1.0 del prototipo	
Implementación	Despliegue del aplicativo en un hosting Web	
Validaciones	Pruebas automatizadas en Postman	
	Pruebas con un usuario final	
Documentación	Realización del documento	

7.1.4. Definición De Los Requerimientos Funcionales

Después de recoger las historias de usuario pasamos a convertirlas en procedimientos o requerimientos para su desarrollo.

Los requerimientos funcionales fueron establecidos de acuerdo con la principal actividad que se desea, como lo es el registro de historias clínicas digitalmente, permitiendo brindar un análisis más simple para los estudiantes y docentes, entre los cuales se establecieron los siguientes requerimientos funcionales:

7.1.4.1. Paquete Usuario

RF-1: El sistema debe permitir crear, consultar, modificar y eliminar un usuario que haga parte de la gestión de datos del aplicativo.

RF-2: El sistema debe contar con dos tipos de usuario (roles), el docente y el de estudiante, que permita gestionar la conformación de la práctica, cada uno de estos con sus respectivas restricciones.

RF-3: El sistema debe de darle únicamente al usuario docente permisos para realizar CRUD de los demás usuarios.

RF-4: El sistema debe almacenar información básica de un usuario (Nombres, apellidos, código institucional y contraseña de acceso).

7.1.4.2. Paquete Historia Clínica / Información De Pacientes

RF-5: El sistema debe tener un formulario simulando una historia clínica, el cual está acorde con los datos suministrados por el docente de Enfermería.

RF-6: El sistema permitirá el registro de la información del paciente, una vez guardado el estudiante no podrá modificar o eliminar.

RF-7: El sistema permitirá ingresar información del paciente como: Datos paciente, observación general. (Información establecida por el docente).

RF-8: El sistema solo mostrará la historia clínica almacenada al rol Admin/docente, también será el único que la puede borrar.

7.1.4.3. Paquete Aplicativo Web.

RF-9: El sistema debe contar con un apartado, en la que se observe una introducción del sistema. Esta deberá ser suministrada por la docente.

RF-10: El sistema debe permitir la visualización correcta de la información en cualquier dispositivo inteligente con el que se desee acceder al aplicativo, por medio de un navegador web.

7.1.4.4. Paquete Gestión Del Sistema.

RF-11: El sistema debe de tener un login de acceso para mayor seguridad y protección de datos. El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con su respectivo código institucional (Que será el usuario) y una contraseña.

RF-12: El sistema debe contar con la funcionalidad de la tecnología de JWT (JSON web Token) para determinar el acceso de cada usuario.

RF-13: El sistema debe tener un temporizador por sesión, si el usuario se encuentra inactivo debe cerrar sesión en determinado tiempo.

7.1.4.5. Requerimientos No Funcionales

RNF-1: El sistema debe contar con una eficiencia de respuesta al mostrar los datos solicitados en máximo 60 segundos.

RNF-2: El sistema debe contar con colores institucionales llamativos y agradables al usuario.

RNF-3: El sistema debe permitir que varios usuarios interactúen al mismo tiempo desde diferentes lugares de conexión.

RNF-4: Los datos modificados en la base de datos deben ser actualizados para todos los usuarios en tiempo real.

RNF-5: El sistema debe operar las 24 horas durante los 7 días de la semana.

RNF-6: El sistema debe contar con un manual de usuario que permita un rápido aprendizaje de su funcionamiento.

RNF-7: El sistema debe de contar con un manual técnico que permita entender los recursos utilizados para el desarrollo del aplicativo.

RNF-8: El sistema debe contener texturas y fuentes de texto claras y agradables a la vista.

RNF-9: La aplicación debe poder utilizarse sin necesidad de instalar ningún software adicional, únicamente un navegador web y conexión a internet

RNF-10: La aplicación debe poder utilizarse con los navegadores web Chrome, Microsoft Edge, Firefox y Opera mini, entre otros.

RNF-11: El sistema debe contar con una base de datos desarrollada en PostgreSQL.

RNF-11: El sistema debe ser desarrollado en Angular con tecnologías HTML5, CSS, JavaScript, TypeScript, Materials, NodeJs, Express, entre otros.

7.1.4.6. Funcionalidades Del Producto

Se establecieron funcionalidades principales de acuerdo con la solicitud del cliente, pretendiendo satisfacer cada una de sus necesidades, como se muestran en la siguiente lista:

1. Usuarios que permiten acceder al sistema, con su respectiva gestión de consulta, registro, modificación y eliminación de información, cada uno con su rol correspondiente, para un control de seguridad.
2. Registro historias clínicas, para que los estudiantes puedan establecer un diagnóstico en su ámbito profesional.

7.1.4.7. Definición De Usuarios

Cada uno de los usuarios establecidos fueron determinados de acuerdo con los requerimientos funcionales, entre los cuales se identificaron dos tipos de usuario con sus diferentes roles, permisos de acceso y características de funciones a realizar, entre los que se encuentran los siguientes:

Usuario Admin/Docente: Este usuario es quien tendrá acceso a toda la información registrada en el sistema, y así mismo el único usuario con permiso de crear, consultar, modificar y/o eliminar otros usuarios registrados, también podrá visualizar y verificar la información de las historias clínicas almacenadas en la base de datos.

Usuario Student/Estudiante: Este usuario cuenta con permisos básicos de registro, estrictamente de la historia clínica del paciente; Adicional a esto este usuario NO podrá modificar ni eliminar información de la historia clínica una vez guardada, así el docente podrá evaluar al estudiante.

7.1.5. Diagrama De Casos De Uso

El Diagrama debe seguir la notación para casos de uso establecida por UML, incluyendo

los elementos del modelo de casos de uso, es decir:

- Actores.
- Casos de Uso.
- Relaciones.

7.1.5.1. Descripción de actores/usuarios

Tabla 3. Descripción actor Docente

Actor	Docente	Identificador: 1
Descripción	Este actor contara con los accesos y funciones totales del sistema	
Características	Verificación de la cuenta, gestión de usuarios	
Relación	Ninguno	
Referencias	Casos de uso, diagrama de secuencias, diagrama de clases, diagrama de actividades	

Tabla 4. Descripción actor Estudiante

Actor	Estudiante	Identificador: 2
Descripción	Este actor contara con el acceso al registro de historia clínica	
Características	Permite realizar el registro de historia clínica y almacenarla	
Relación	Actor docente	
Referencias	Casos de uso, diagrama de secuencias, diagrama de clases, diagrama de actividades	

7.1.5.2. Especificación De Casos De Uso

a. Caso De Uso “Iniciar Sesión”

Tabla 5. Caso de uso Iniciar sesión

Nombre:	Iniciar Sesión / CU-1	
Actor:	Docente, Estudiante	
Descripción:	Describe el proceso de realizar el ingreso de cualquier usuario al sistema, siempre y cuando se encuentre registrado con anterioridad.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Digitar url del aplicativo en el navegador de internet	

	2. En la página principal seleccionar opción “Iniciar sesión”	2. Muestra en pantalla el formulario de login para acceder al sistema
	3. Digitar usuario o correo electrónico y contraseña, dar clic en el botón “Ingresar”	3. Verificación en la base de datos de la existencia del usuario y de su contraseña
	4. Ingreso al sistema	4. Usuario verificado correctamente, permitir el acceso al sistema Mostrar interfaz principal.
Excepcional:	1. Digitar url del aplicativo en el navegador de internet	
	2. En la página principal seleccionar opción “Iniciar sesión”	2. Muestra en pantalla el formulario de login para acceder al sistema
	3. Digitar usuario o correo electrónico y contraseña, dar clic en el botón “Ingresar”	3. Verificación en la base de datos de la existencia del usuario y de su contraseña
	4. Notificación de error “usuario o contraseña incorrectos, verifíquelos e intente de nuevo”	4. Usuario o contraseña no encontrados, acceso al sistema denegado
Precondición:	El usuario debe estar previamente registrado, para permitir el correcto ingreso.	
Poscondición:	Panel de administración del sistema, funcionando correctamente.	
Presunción:	Usuario existente en la base de datos.	

b. Caso De Uso “Cerrar Sesión”

Tabla 6. Caso de uso Cerrar Sesión

Nombre:	Cerrar Sesión / CU-2	
Actor:	Docente, Estudiante	
Descripción:	Describe el proceso de finalizar la sesión iniciada con anterioridad de cualquier usuario.	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Haber iniciado sesión	1. Verificar que la sesión aun este activa
	2. Sesión cerrada con éxito	2. Finalizar la sesión total del usuario

Excepcional:	1. Haber iniciado sesión	1. Verificar que la sesión aun este activa
	2. Sesión no cerrada	2. Sesión no iniciada, o error de seguridad
Precondición:	El usuario debe estar previamente registrado, y así mismo haber iniciado sesión con anterioridad.	
Poscondición:	Sesión finalizada en su totalidad, para iniciar una nueva sesión.	
Presunción:	Sesión disponible para ser iniciada.	

c. Caso De Uso “Crear Usuario”

Tabla 7. Caso de uso Crear Usuario

Nombre:	Crear Usuario / CU-3	
Actor:	Docente	
Descripción:	Describe el proceso de realizar el Registro de usuarios Docentes u Estudiantes	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Accede al aplicativo	
	2. Selecciona la opción “Usuarios/Users” y luego de esto la opción “Agregar usuario”	2. Muestra en pantalla el formulario de registro de datos (Nombre, rol, ID código institucional, contraseña, apellido).
	3. Realiza registro de datos del usuario	3. Muestra pantalla de Registro y confirmación si los datos fueron correctamente registrados y validados.
	4. Almacenar los datos registrados.	4. El Sistema Confirma el Almacenamiento Datos
Excepcional:	1. Acceder al aplicativo	
	2. Selecciona la opción “Usuarios/Users” y luego de esto la opción “Agregar usuario”	2. Muestra en pantalla el formulario de registro de datos
	3. Realiza registro de datos del usuario	3. Verificar si se registraron los datos completos en caso de no ser así volver a introducir los que faltan
	4. Almacenar los datos registrados	4. El Sistema Confirma el Almacenamiento de Datos

Precondición:	El usuario no debe estar previamente registrado, para permitir el correcto registro. El usuario Docente que va a llevar a cabo el registro debe estar previamente registrado en el sistema.
Poscondición:	El Docente se encuentra registrado, permitiendo su acceso al sistema.
Presunción:	La base de datos de registro de usuario está disponible.

d. Caso De Uso “Modificar Usuario”

Tabla 8. Caso de uso modificar usuario

Nombre:	Modificar Usuario / CU-4	
Actor:	Docente	
Descripción:	Describe el proceso de modificar datos de usuario	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Accede al aplicativo, y haber seleccionado el módulo de usuarios.	
	2. Identificar usuario a modificar, y presionar botón que le corresponde “modificar”.4	2. Muestra en pantalla el formulario de modificación de datos del usuario seleccionado únicamente de los datos básicos como: Rol, Nombre, Apellido, código institucional. El sistema debe de mostrar código institucional para identificar el usuario, pero no debe de permitir modificación de este dato.
	3. Realiza modificación de datos del usuario	3. Muestra en pantalla el Usuario modificado y confirmación si los datos fueron correctamente modificados.
	4. Notificación de que los datos fueron modificados correctamente.	4. El Sistema confirma la modificación de los datos
Excepcional:	1. Accede al aplicativo, y haber seleccionado el módulo de usuarios.	

	2. Identificar usuario a modificar, y presionar botón que le corresponde “modificar”	2. Muestra en pantalla el formulario de modificación de datos del usuario seleccionado
	3. Realiza modificación de datos del usuario	3. Muestra pantalla de Usuarios y Error al modificar los datos
	4. Notificación de error al modificar los datos.	
Precondición:	El usuario debe estar previamente registrado, para permitir su correcta modificación.	
Poscondición:	Usuario modificado con éxito, en la base de datos	
Presunción:	La base de datos de registro de usuario está disponible.	

e. Caso De Uso “Eliminar Usuario”

Tabla 9. Caso de uso eliminar usuario

Nombre:	Eliminar Usuario / CU-5		
Actor:	Docente		
Descripción:	Describe el proceso de realizar el Registro de usuarios Docentes o Estudiantes		
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA	
	1. Accede al aplicativo		
	2. Selecciona la opción “Usuarios/Users” y luego de esto identificar el usuario a eliminar y presionar la opción “Eliminar”	2. Muestra en pantalla ventana emergente de confirmación para eliminar usuario	
	3. Confirmar la eliminación	3. Eliminar usuario de la base de datos	
	Excepcional:	1. Acceder al aplicativo	
		2. Selecciona la opción “Usuarios” y luego de identificar el usuario, lo selecciona para eliminar y presionar la opción “Eliminar”	2. Muestra en pantalla ventana emergente de confirmación para eliminar usuario
3. Confirmar la eliminación		3. Error al eliminar usuario de la base de datos, el usuario es un Docente o el sistema no puede quedarse en un estado de 0 usuarios, pero si 0 estudiantes.	

Precondición:	El usuario no debe estar previamente registrado, para permitir eliminarlo.
Postcondición:	El usuario se elimina de la base de datos.
Presunción:	Usuario ya no existe

f. Caso De Uso “Registrar Historia Clínica”

Tabla 10. Caso de uso registrar historia clínica

Nombre:	Registro de historia clínica / CU-6	
Actor:	Estudiante	
Descripción:	Describe el proceso de realizar el registro de la historia clínica que realizan los estudiantes de Enfermería	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Accede al aplicativo	
	2. Selecciona la opción “Asignación de paciente” y luego de esto la opción “Registrar Paciente”	2. Muestra en pantalla el formulario de registro de pacientes
	3. Realiza registro de información del paciente	3. Muestra pantalla datos del paciente y un botón de siguiente para mostrar los siguientes datos que hacen parte de la historia clínica.
	4. Almacenar los datos registrados.	4. El Sistema Confirma el Almacenamiento Datos
Excepcional:	1. Acceder al aplicativo	
	2. Selecciona la opción “Asignación de paciente” y luego de esto la opción “Registrar Paciente”	2. Muestra en pantalla el formulario de registro de historia clínica
	3. Realiza registro de información del paciente	3. Muestra pantalla botón de guardar y confirmación si los datos fueron correctamente registrados
	4. Almacenar los datos registrados	4. El Sistema Confirma el Almacenamiento de Datos
Precondición:	El usuario Estudiante debe haber iniciado sesión	
Poscondición:	Valores de producción registrados y almacenados en la base de datos	

Presunción:	La base de datos de registro de valores de producción está disponible.
--------------------	------------------------------------------------------------------------

g. Caso De Uso “Observar Historia Clínica”

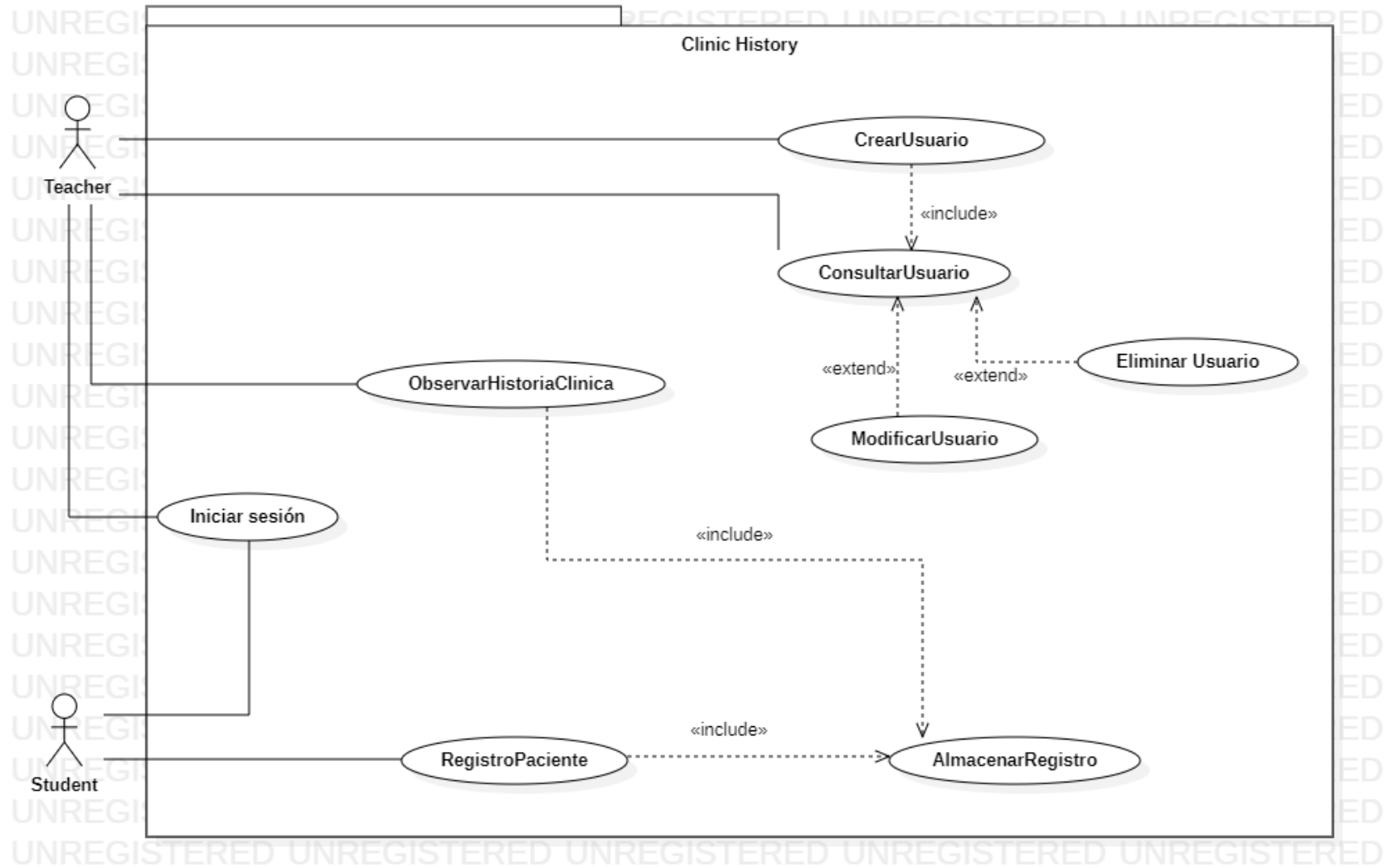
Tabla 11. Caso de uso observar historia clínica

Nombre:	Observar Historia Clínica / CU-7	
Actor:	Docente	
Descripción:	Describe el proceso de verificar la información de la Historia Clínica registrados con anterioridad por parte del usuario Estudiante, en caso de alguna corrección	
Flujo Principal:	Eventos ACTOR	Eventos SISTEMA
	1. Accede al aplicativo	
	2. Selecciona la opción “Historia Clínicas”	2. Muestra tabla contenedora de todas las Historia Clínicas registradas
	3. Identificar Historia Clínica a verificar, dar clic en el botón verificar	3. Mostrar modal de confirmación de acción
	4. Confirmar la acción presionando el botón “si”	4. Verificar Historia Clínica en base de datos cambiando estado a “verificado” observando que este según el criterio del docente
Excepcional:	1. Acceder al aplicativo	
	2. Selecciona la opción “Historia Clínicas”	2. Muestra tabla contenedora de todas las Historia Clínicas registradas
	3. Identificar Historia Clínica a verificar, dar clic en el botón verificar	3. Mostrar modal de confirmación de acción
	4. Confirmar la acción presionando el botón “no”	4. Verificar Historia Clínica en base de datos cambiando estado ha “verificado”
Precondición:	El usuario no debe estar previamente registrado, para permitir el correcto registro. El usuario Docente que va a llevar a cabo el registro debe estar previamente registrado en el sistema.	
Poscondición:	El Docente se encuentra registrado, permitiendo su acceso al sistema.	
Presunción:	La base de datos de registro de usuario está disponible.	

7.2. Etapa: Análisis

7.2.1. Diagramas UML

7.2.1.1. Diagrama De Caso De Uso

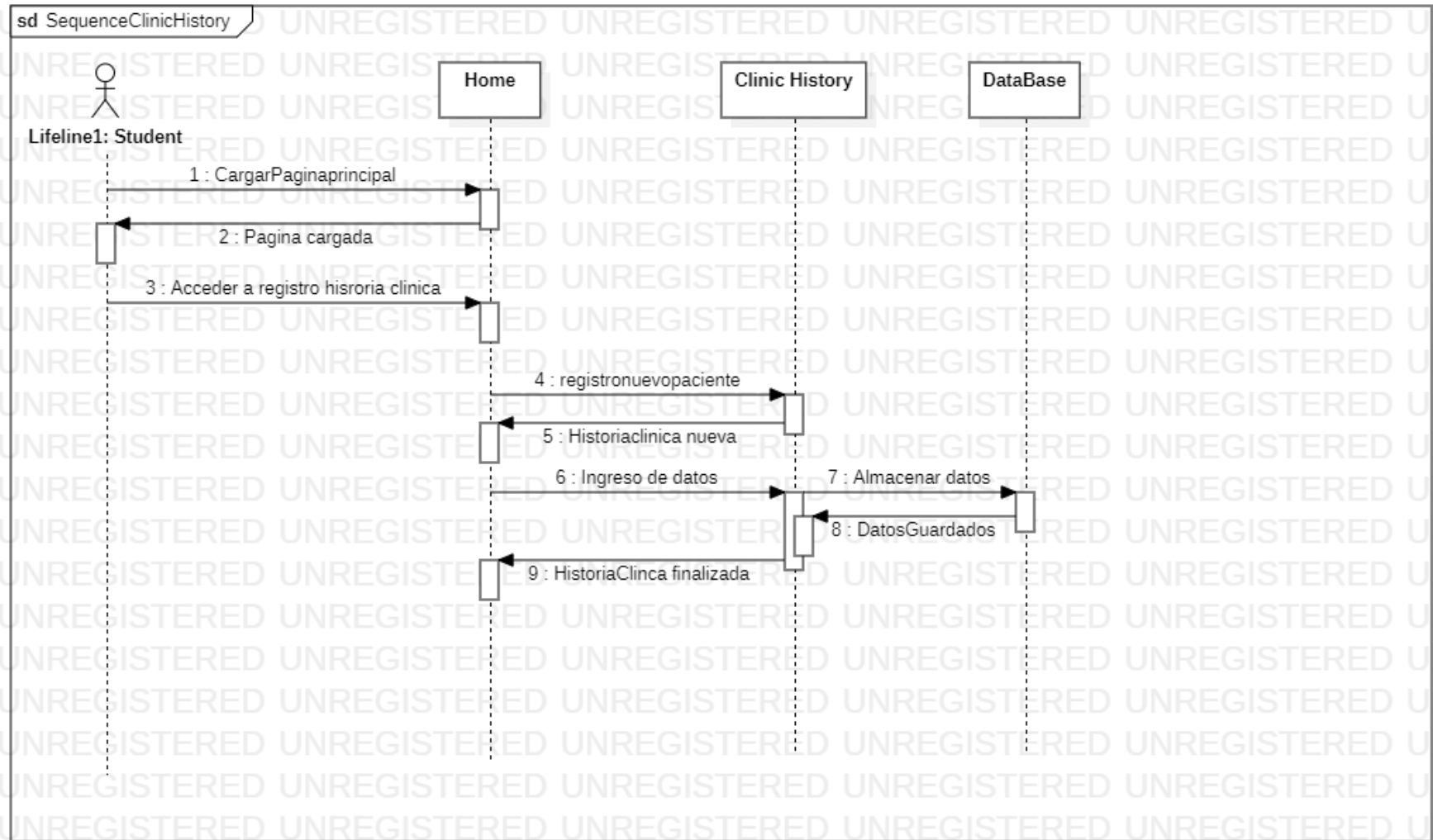


Fuente 7. Elaboración propia elaborado en StarUML

Figura 7. Diagrama de caso de uso

7.2.1.2. Diagramas De Secuencia

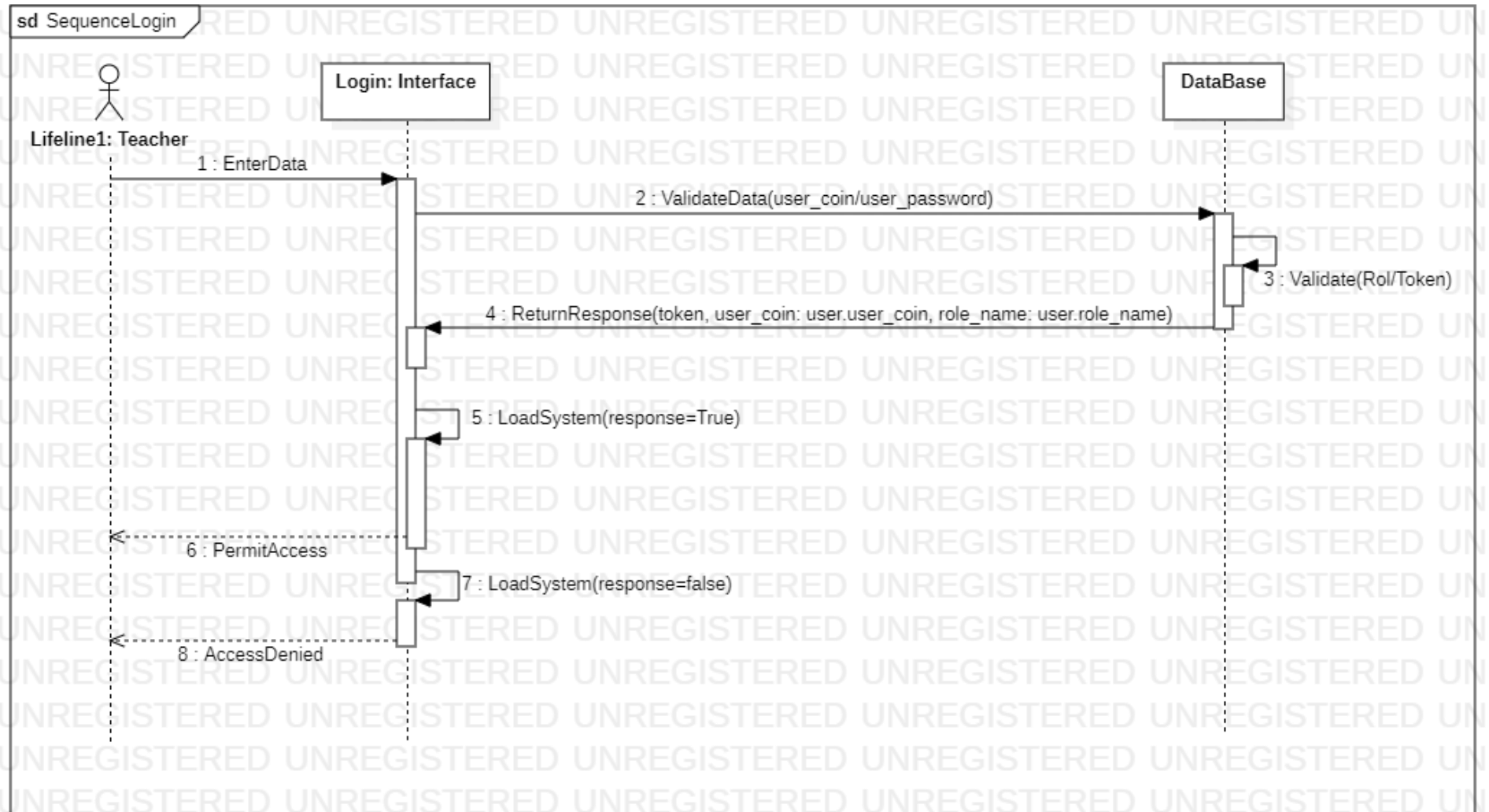
Diagrama de secuencia historia clínica



Fuente 8. Elaboración propia elaborado en StarUML

Figura 8. Diagrama de secuencia de historia clínica

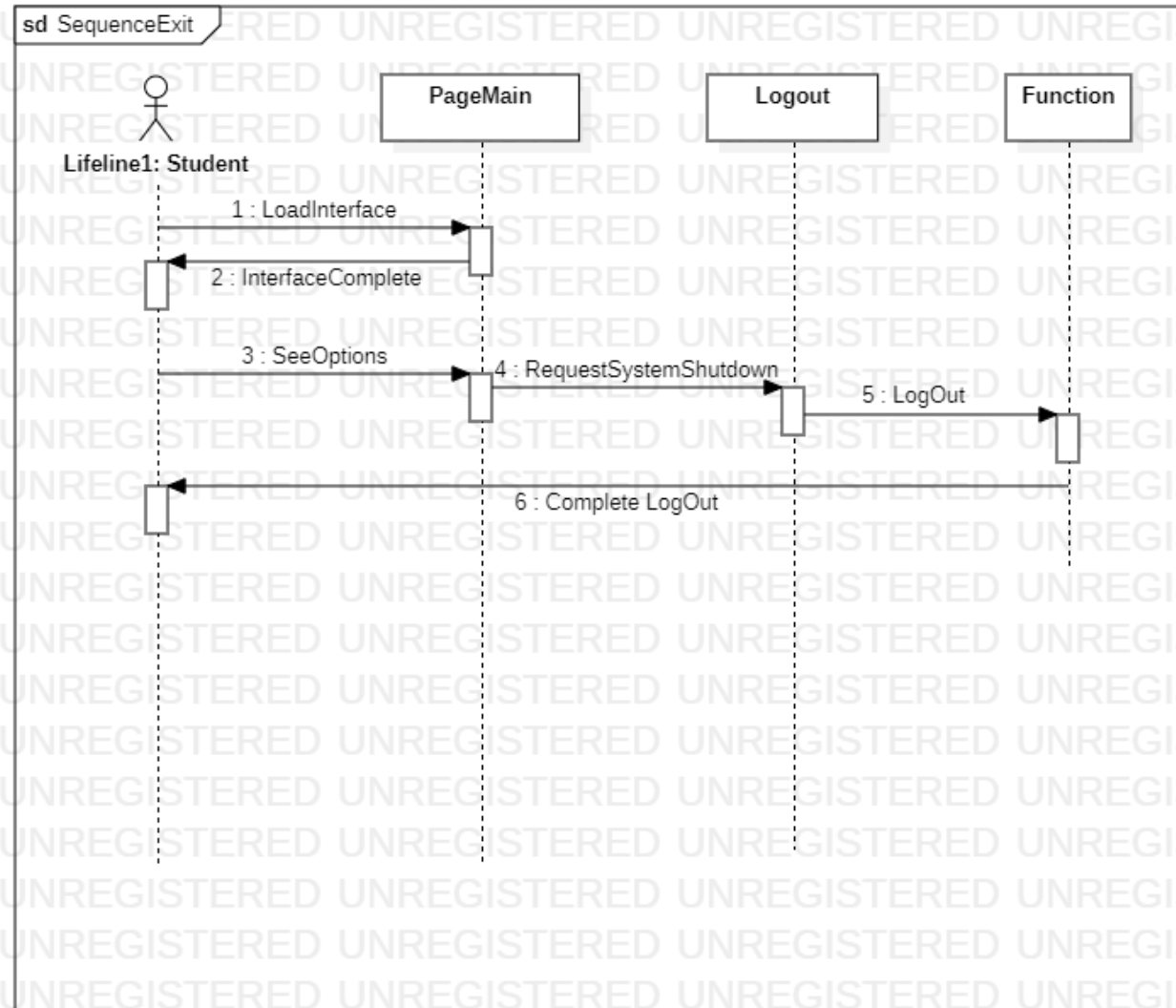
Diagrama de secuencia iniciar sesión



Fuente 9. Elaboración propia elaborado en StarUML

Figura 9.. Diagrama de secuencia de iniciar sesión

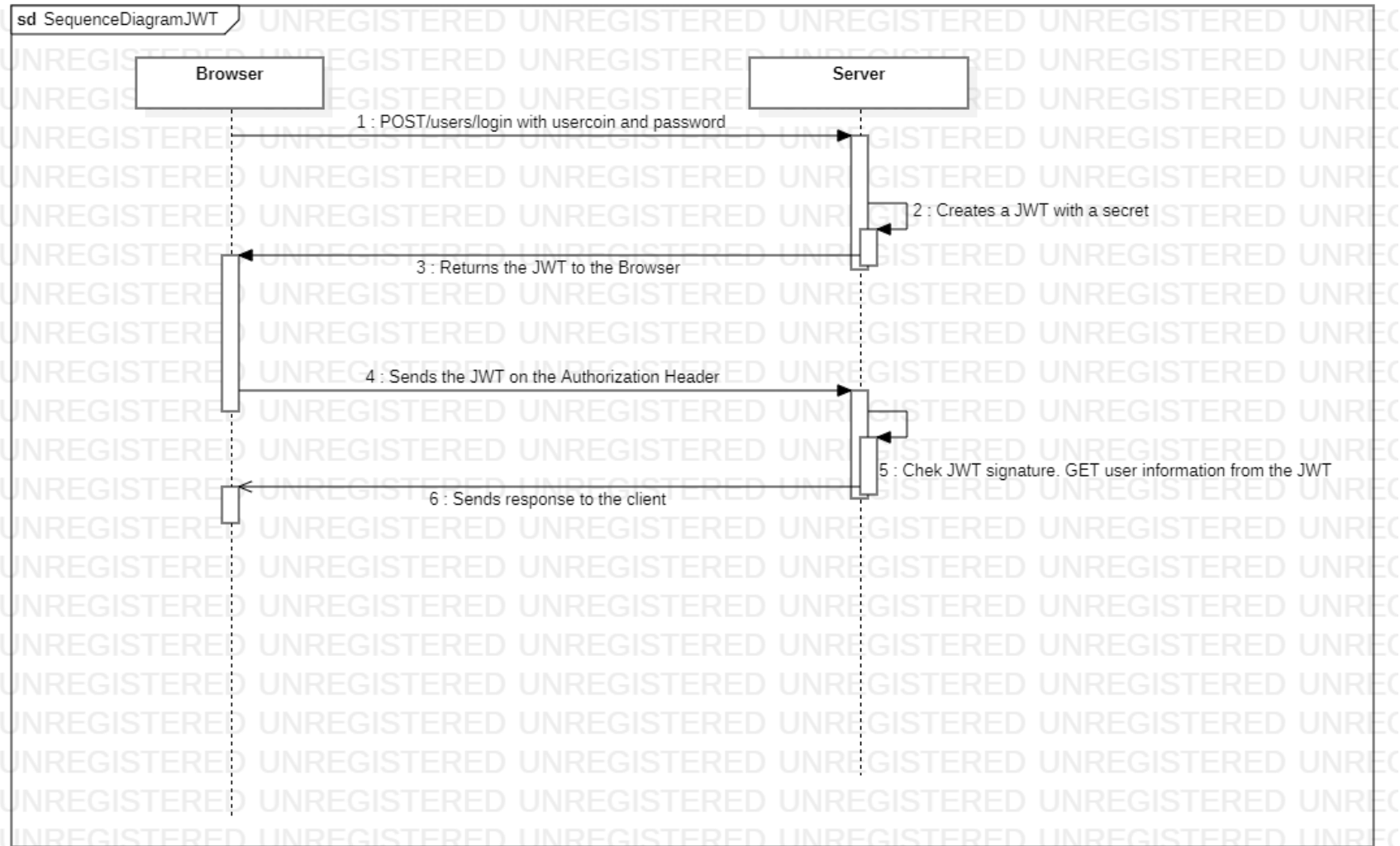
Diagrama de secuencia de Cerrar sesión



Fuente 10. Elaboración propia
elaborado en StarUML

Figura 10. Diagrama de secuencia
de cerrar sesión

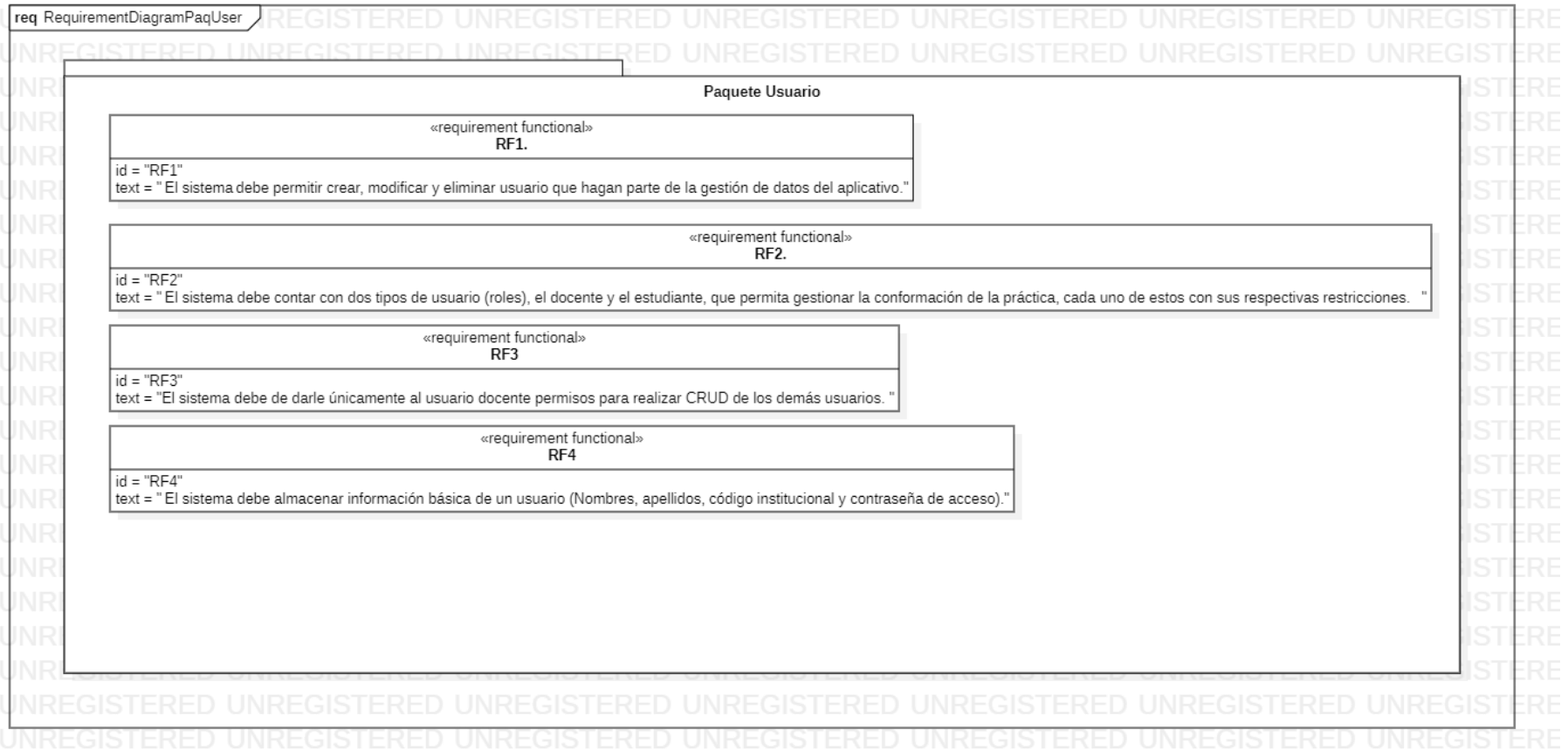
Diagrama de secuencia JWT



Fuente 11. Elaboración propia elaborado en StarUML

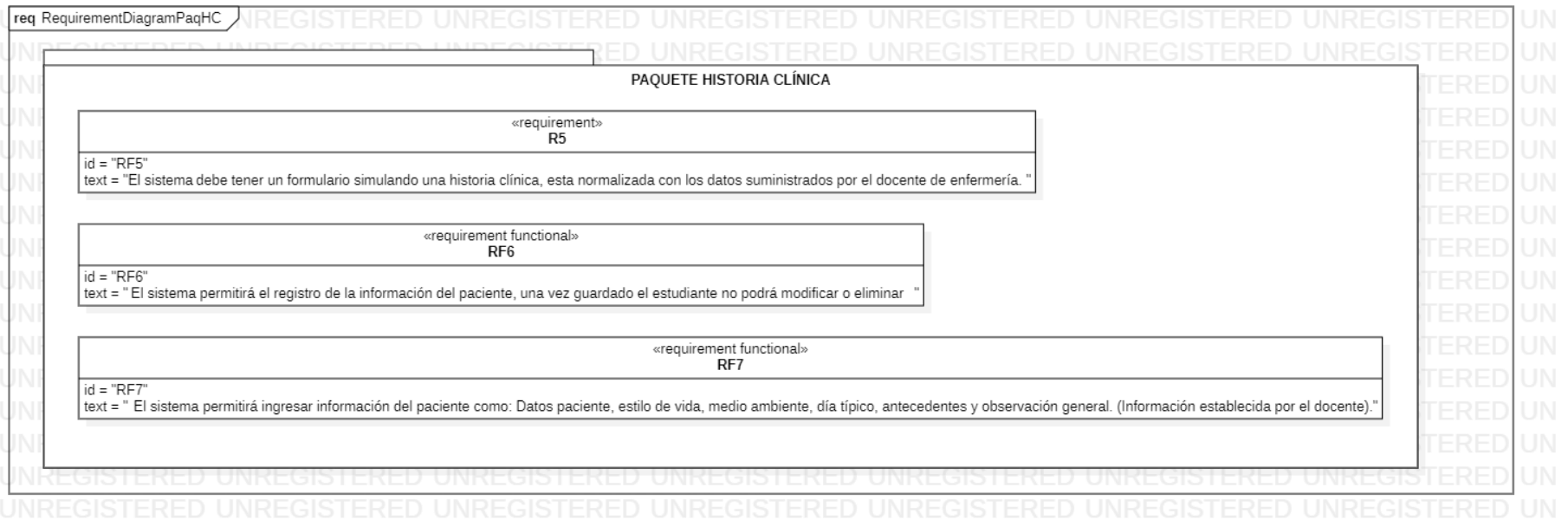
Figura 11. Diagrama de secuencia de JWT

7.2.1.3. Diagrama De Requerimientos



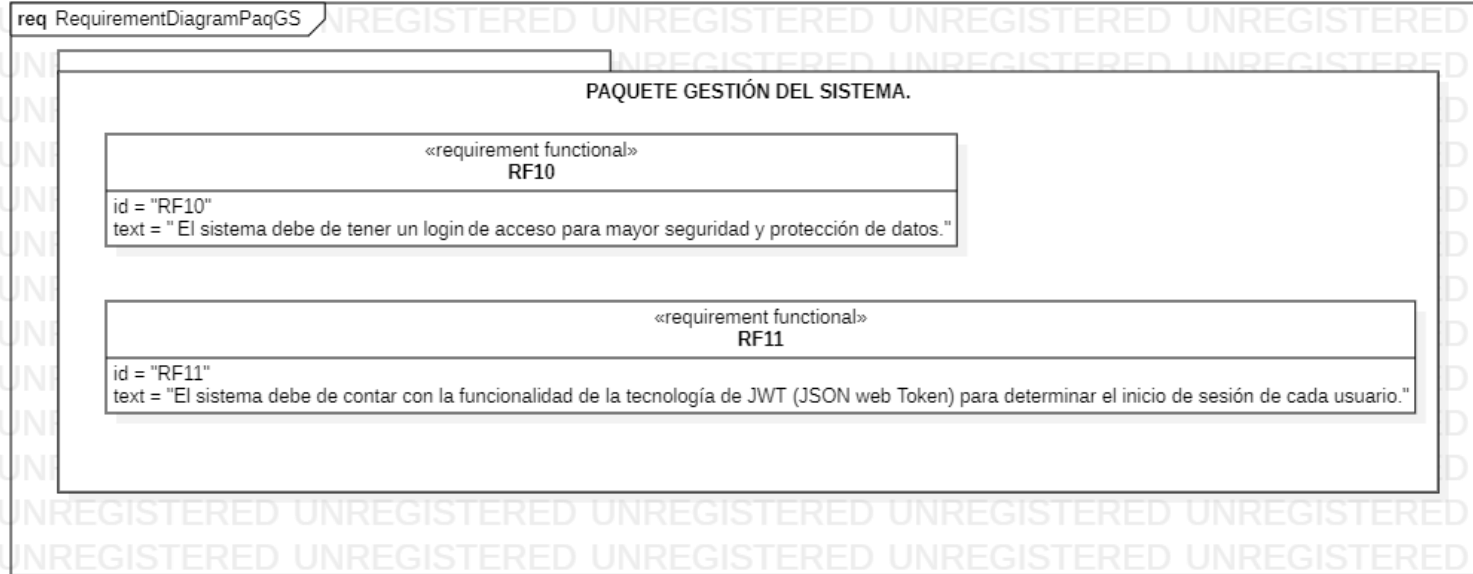
Fuente 12. Elaboración propia elaborado en StarUML

Figura 12. Diagrama de requerimientos "paquete usuario"



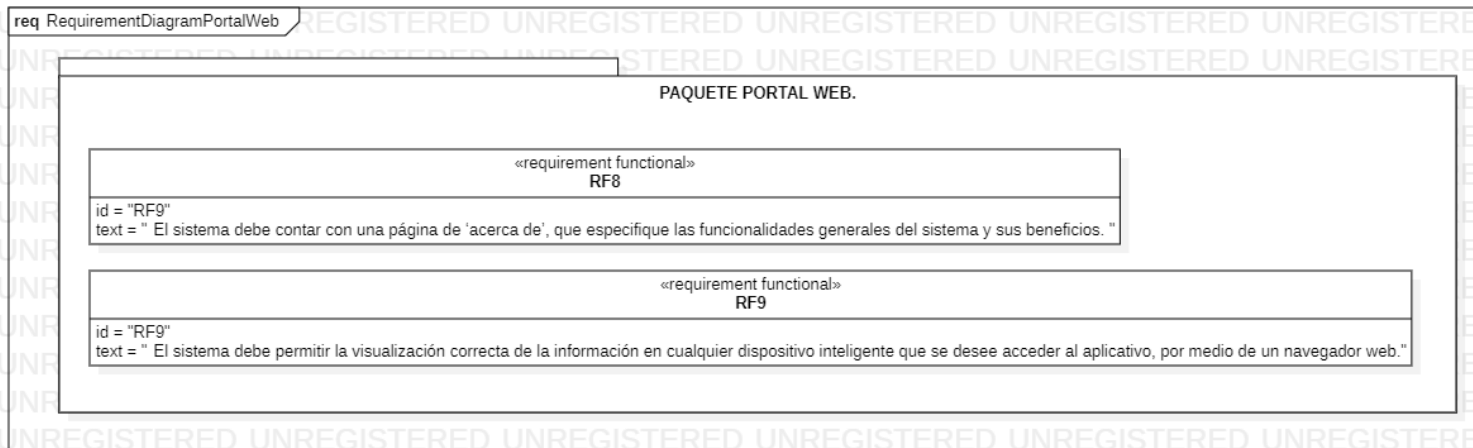
Fuente 13. Elaboración propia elaborado en StarUML

Figura 13. Diagrama de requerimientos "Paquete de historia clínica"



Fuente
14. Elaboración
propia elaborado
en StarUML

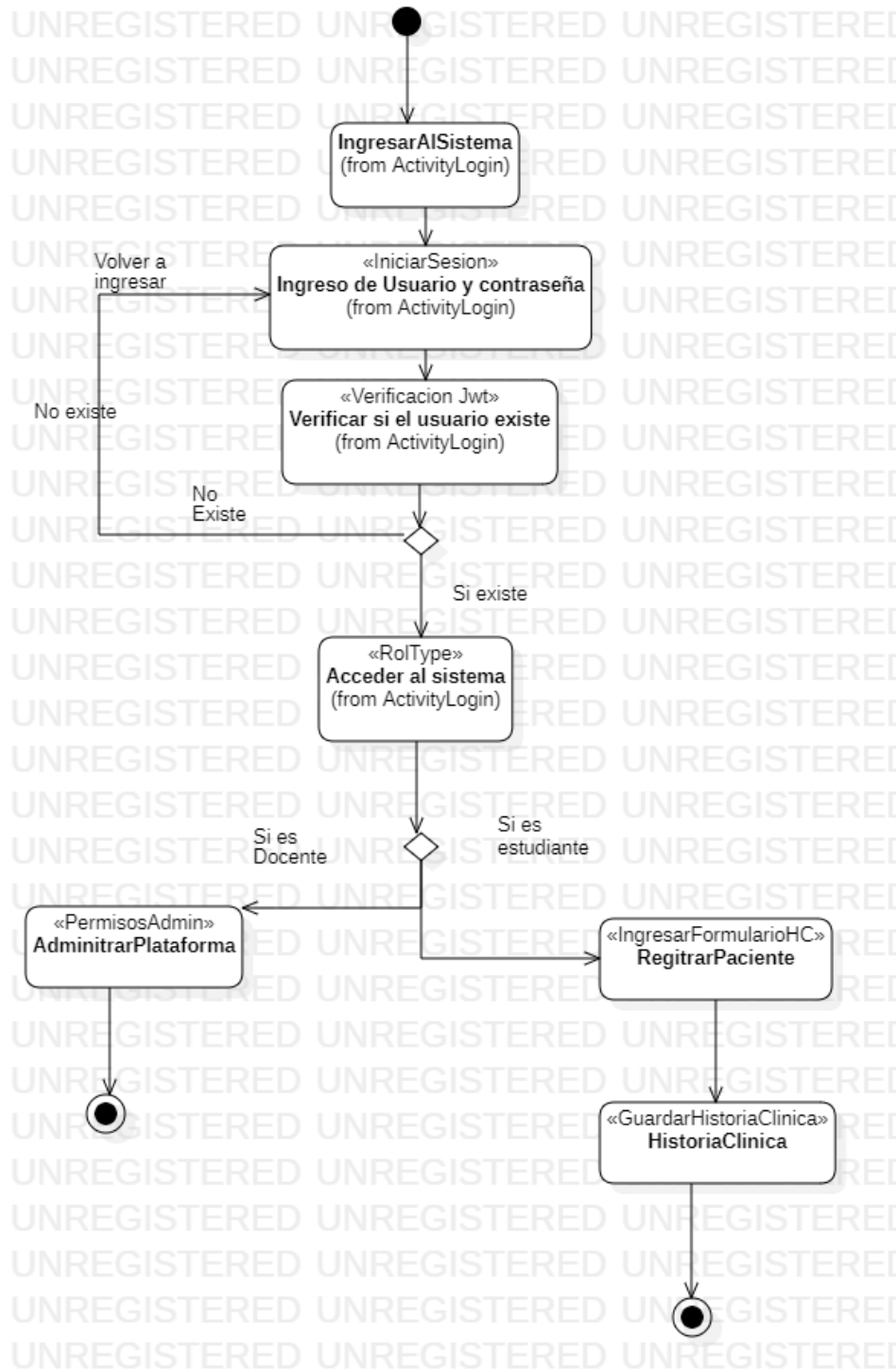
Figura
14. Diagrama de
requerimientos
"Paquete gestión
del sistema"



Fuente 15. Elaboración propia elaborado en StarUML

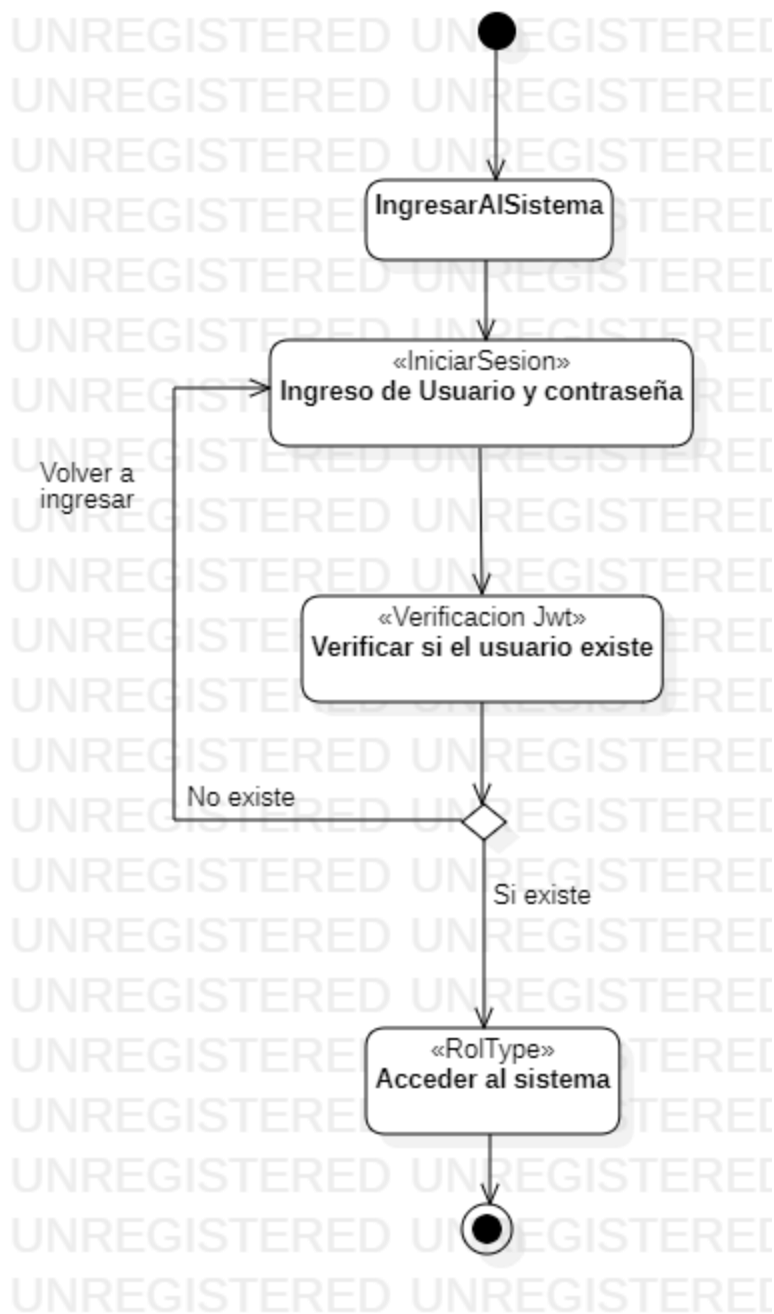
Figura 15. Diagrama de requerimientos "Paquete portal web"

7.2.1.4. Diagrama De Actividades



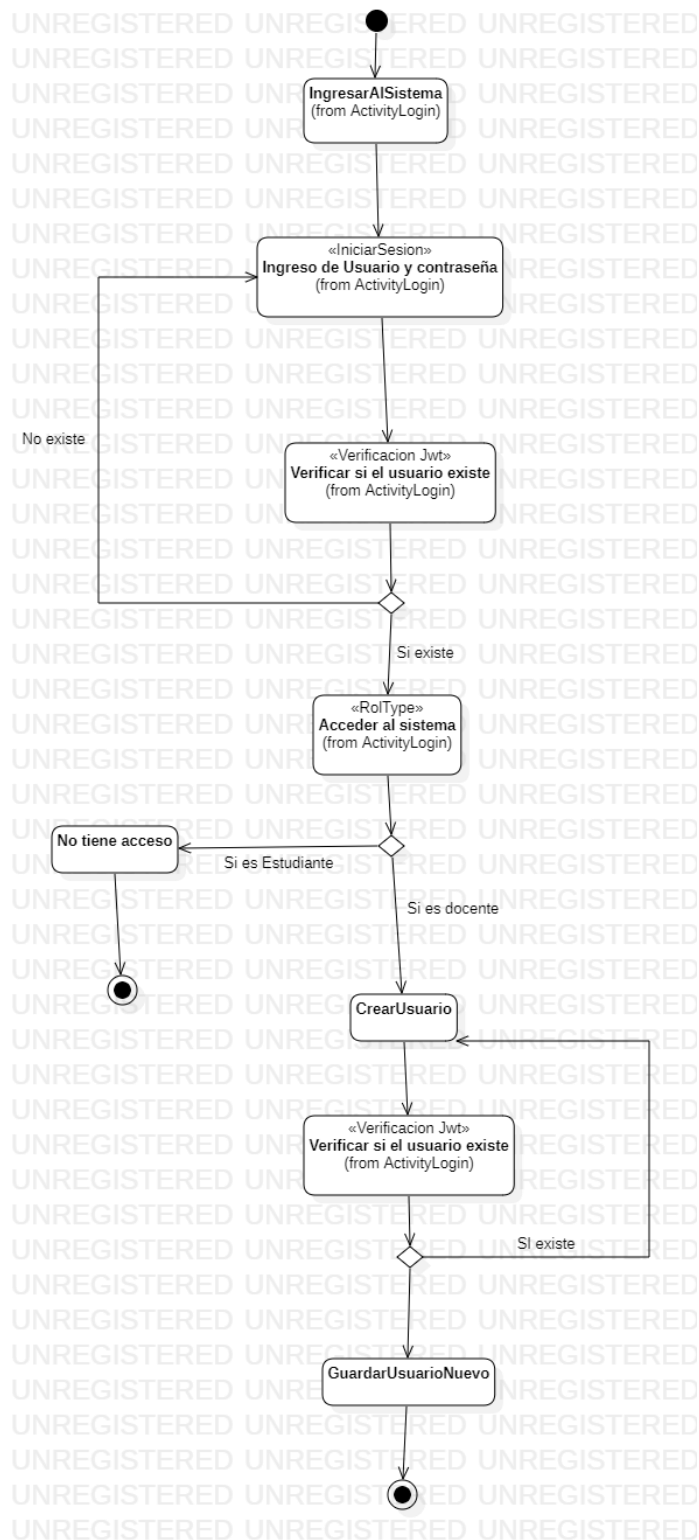
Fuente 16. Elaboración propia elaborado en StarUML

Figura 16. .Diagrama de actividades historia clínica



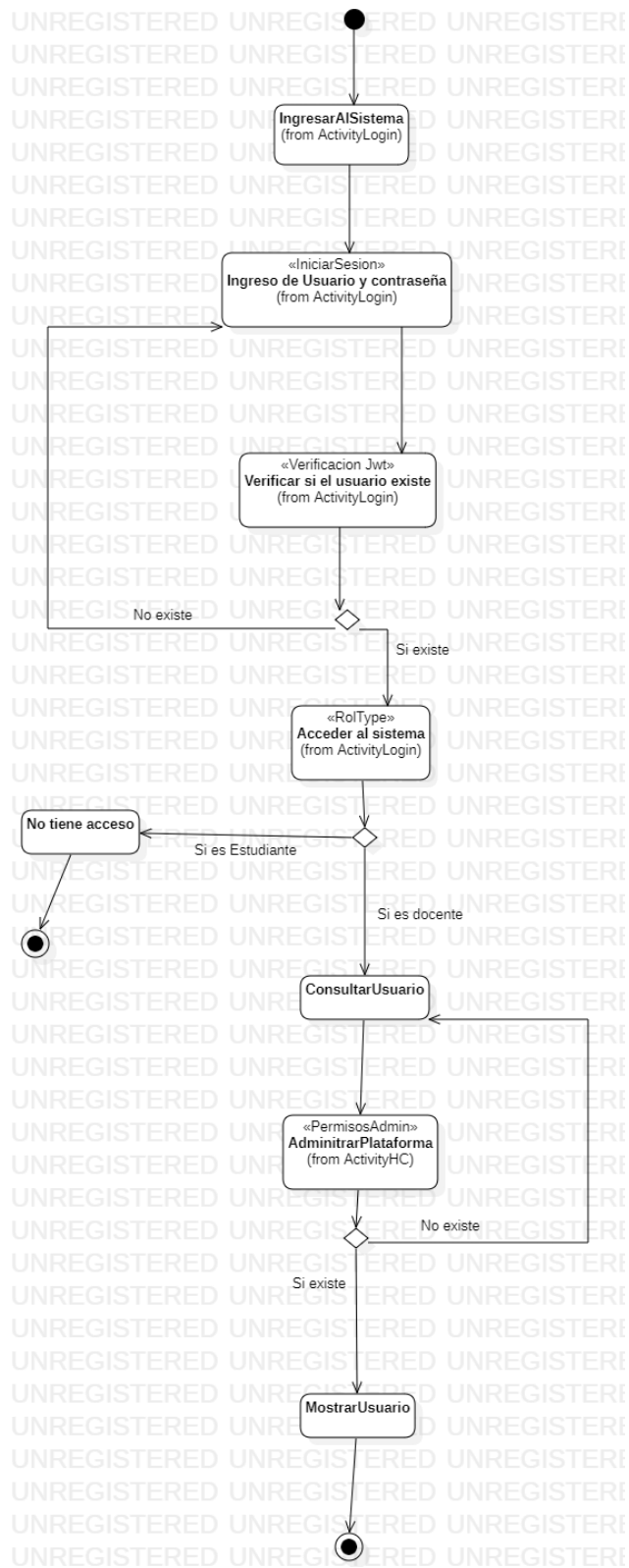
Fuente 17. Elaboración propia elaborado en StarUML

Figura 17. Diagrama de actividades de inicio de sesión



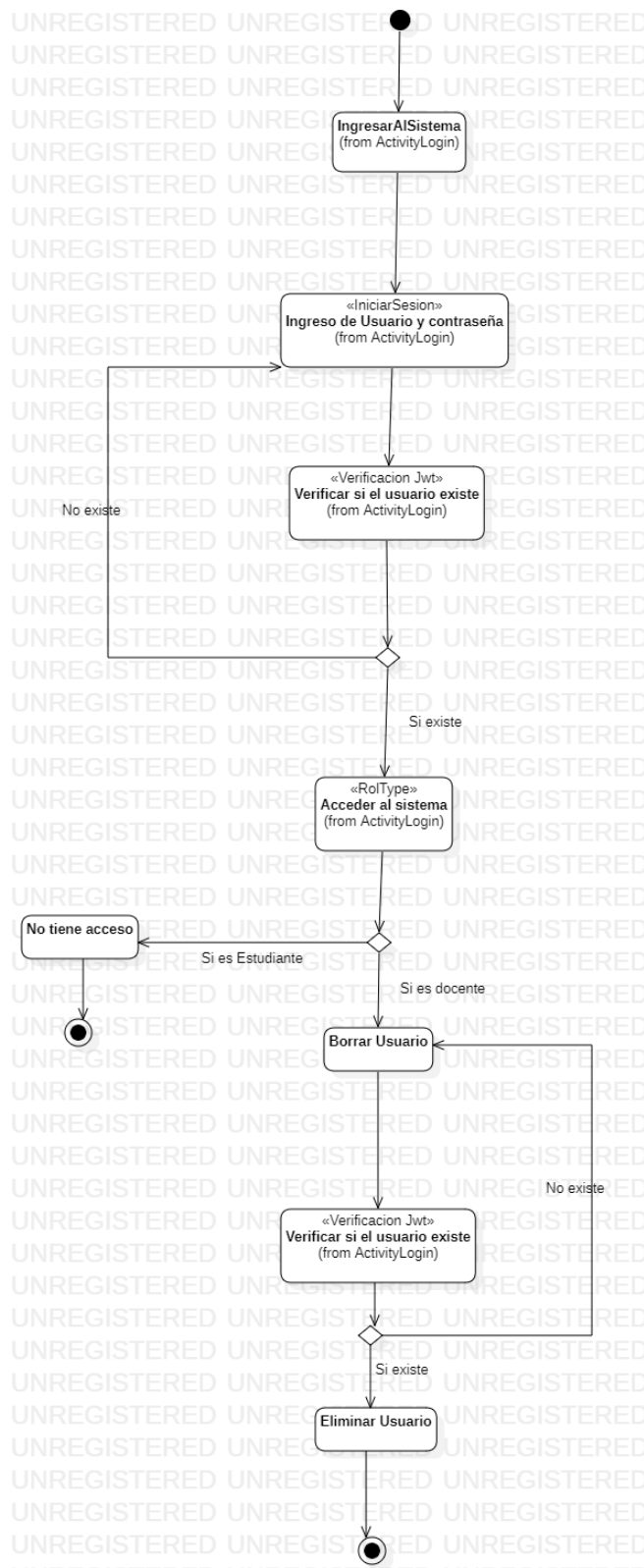
Fuente 18. Elaboración propia elaborado en StarUML

Figura 18. Diagrama de actividades crear usuario



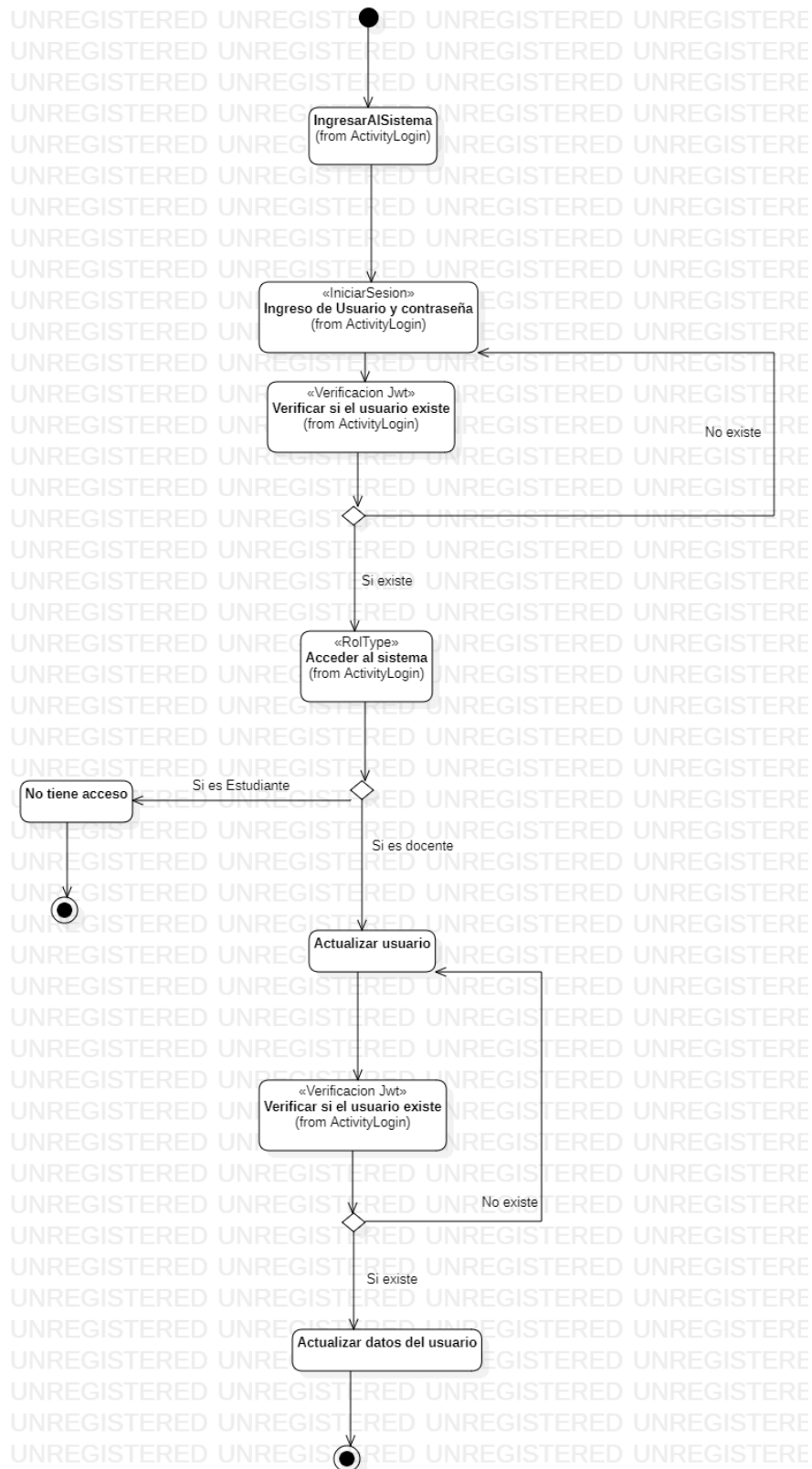
Fuente 19. Elaboración propia elaborado en StarUML

Figura 19. Diagrama de actividades consultar usuario.



Fuente 20. Elaboración propia elaborado en StarUML

Figura 20. Diagrama de actividades Borrar usuario

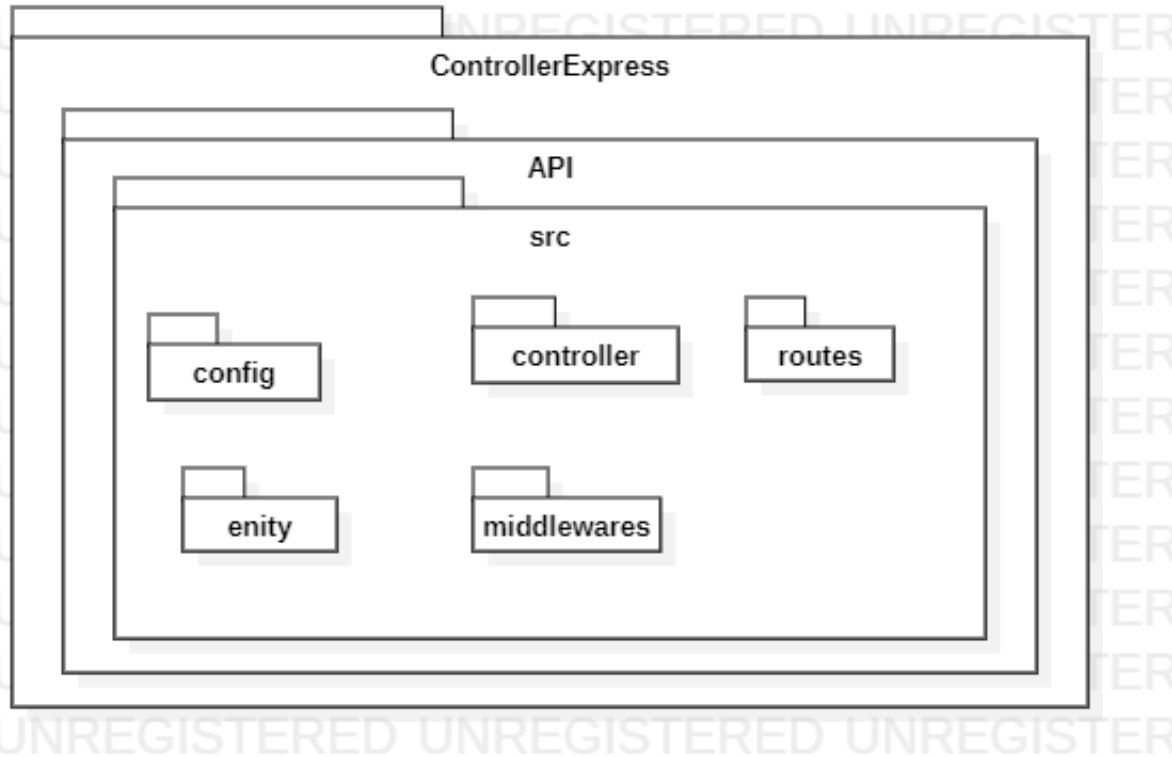


Fuente 21. Elaboración propia elaborado en StarUML

Figura 21 Diagrama de actividades Actualizar usuario

7.2.1.5. Diagrama De Paquetes

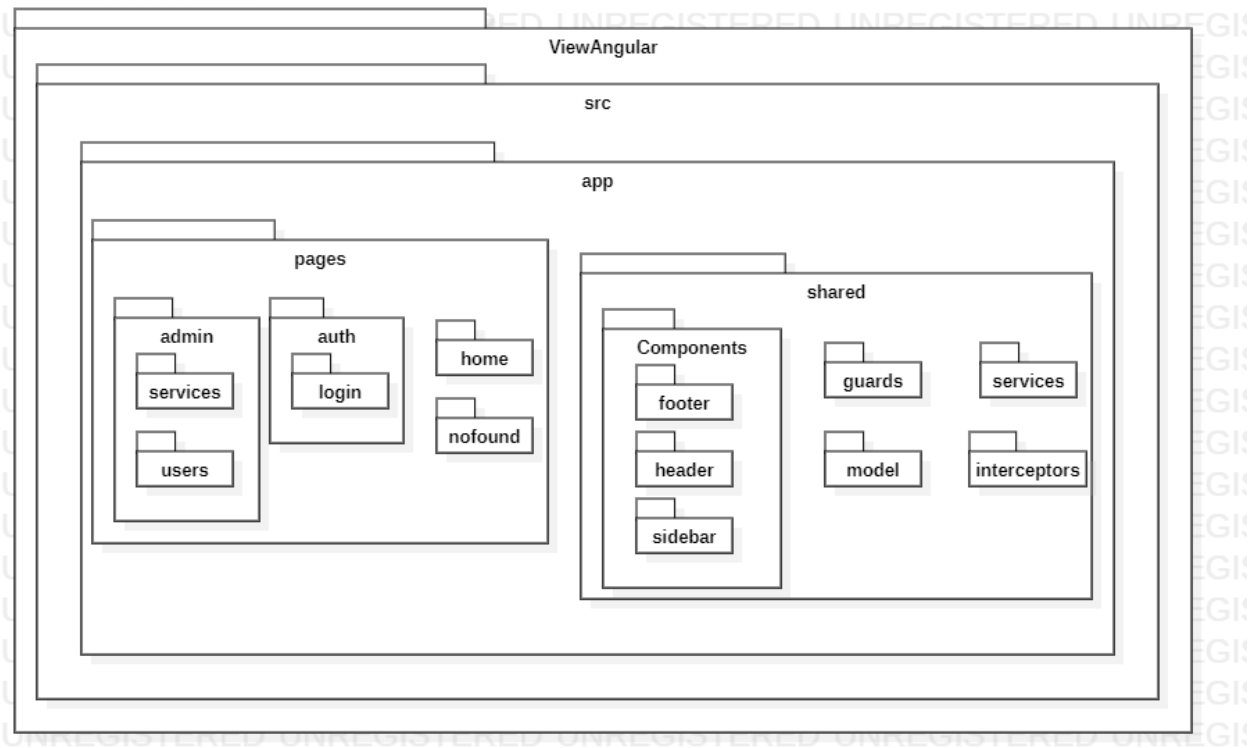
Paquete BackEnd



Fuente 22. Elaboración propia elaborado en StarUML

Figura 22. Diagrama de paquetes "ControllerExpressjs"

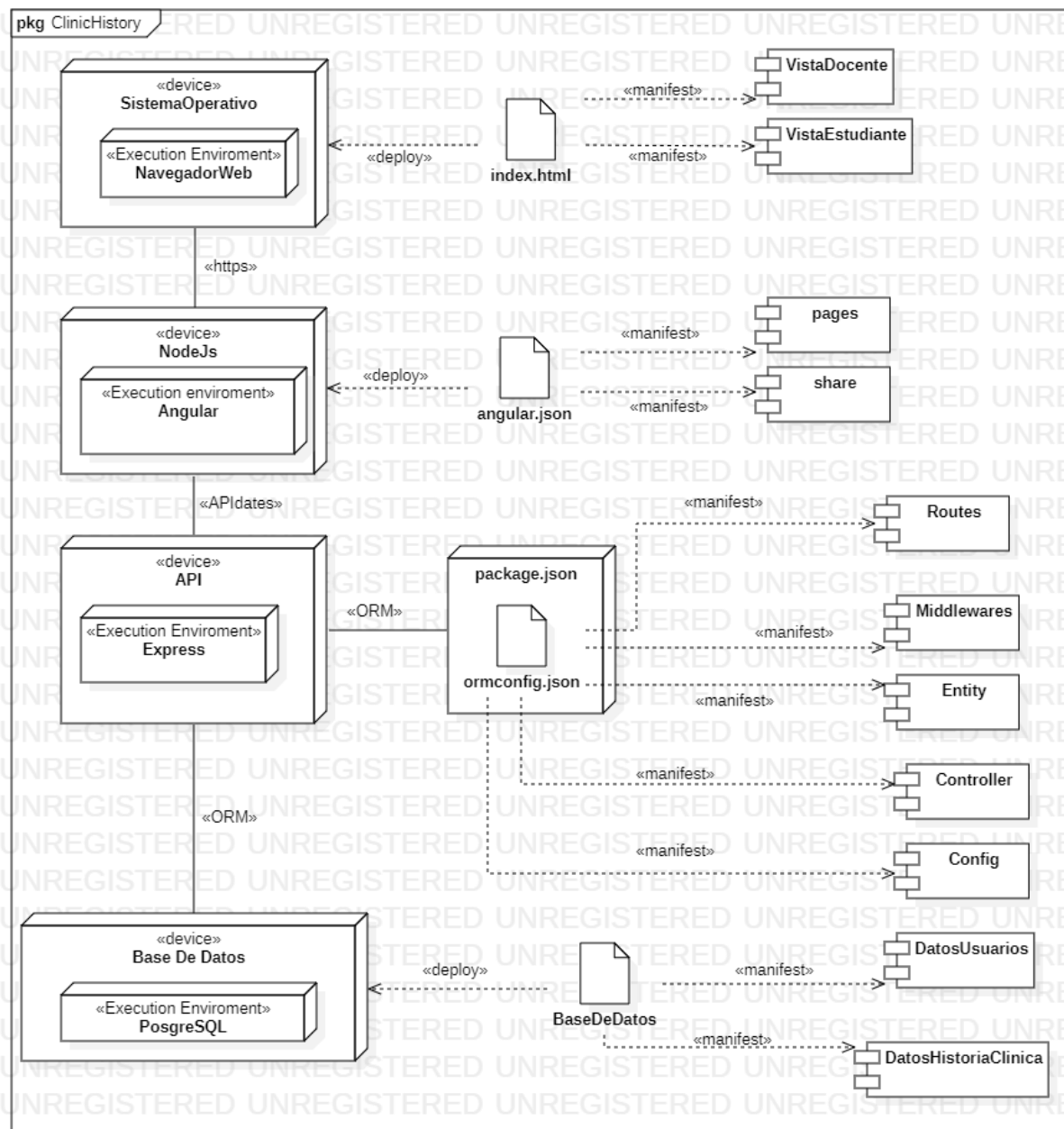
Paquete FrontEnd



Fuente 23. Elaboración propia elaborado en StarUML

Figura 23. Diagrama de paquetes "ViewAngular"

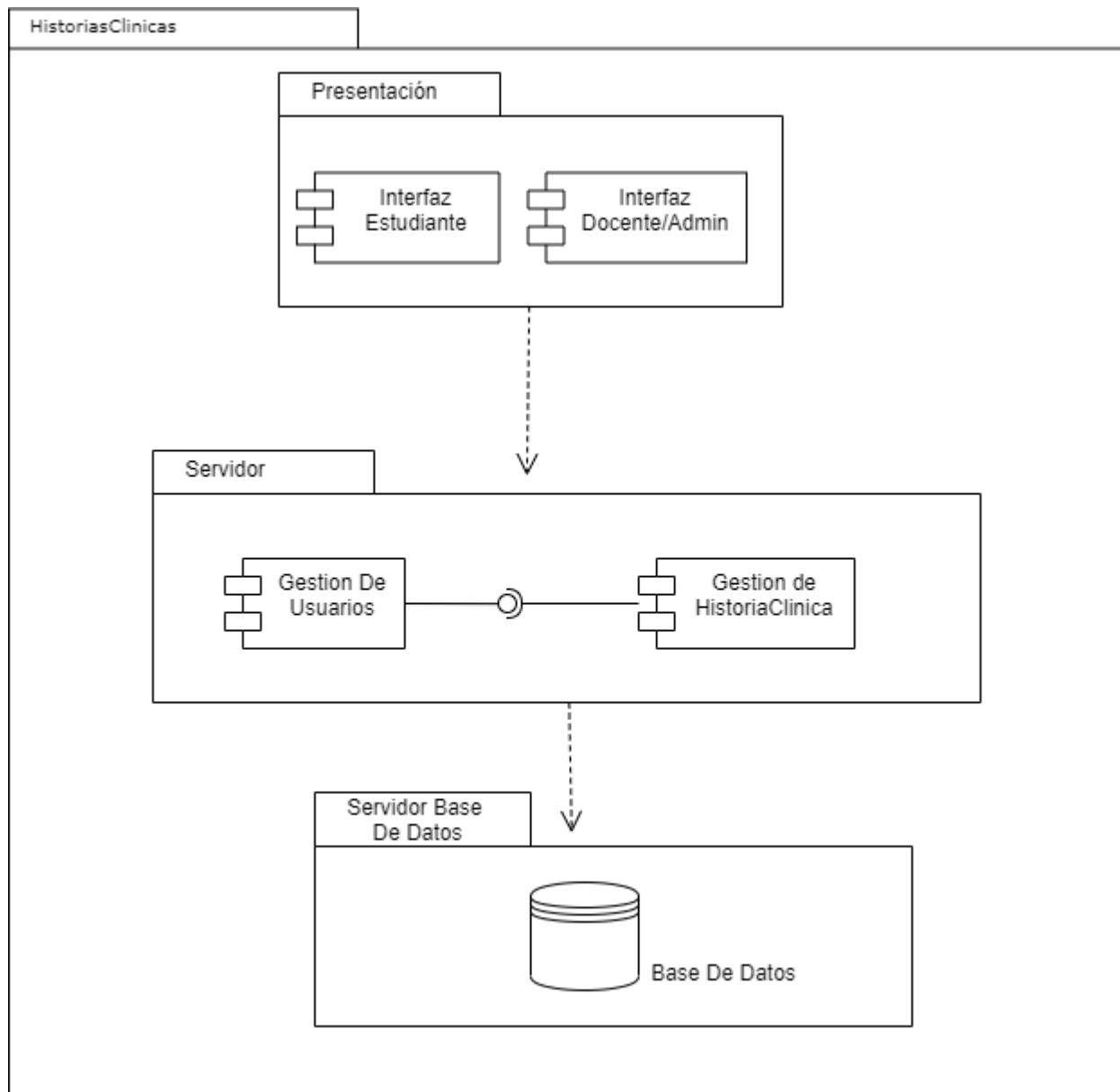
7.2.1.6. Diagrama De Despliegue



Fuente 24. Elaboración propia elaborado en StarUML

Figura 24. Diagrama de despliegue

7.2.1.7. Diagrama De Componentes



Fuente 25. Elaboración propia elaborado en draw.io

Figura 25. Diagrama de componentes

7.2.2. Modelo Relacional

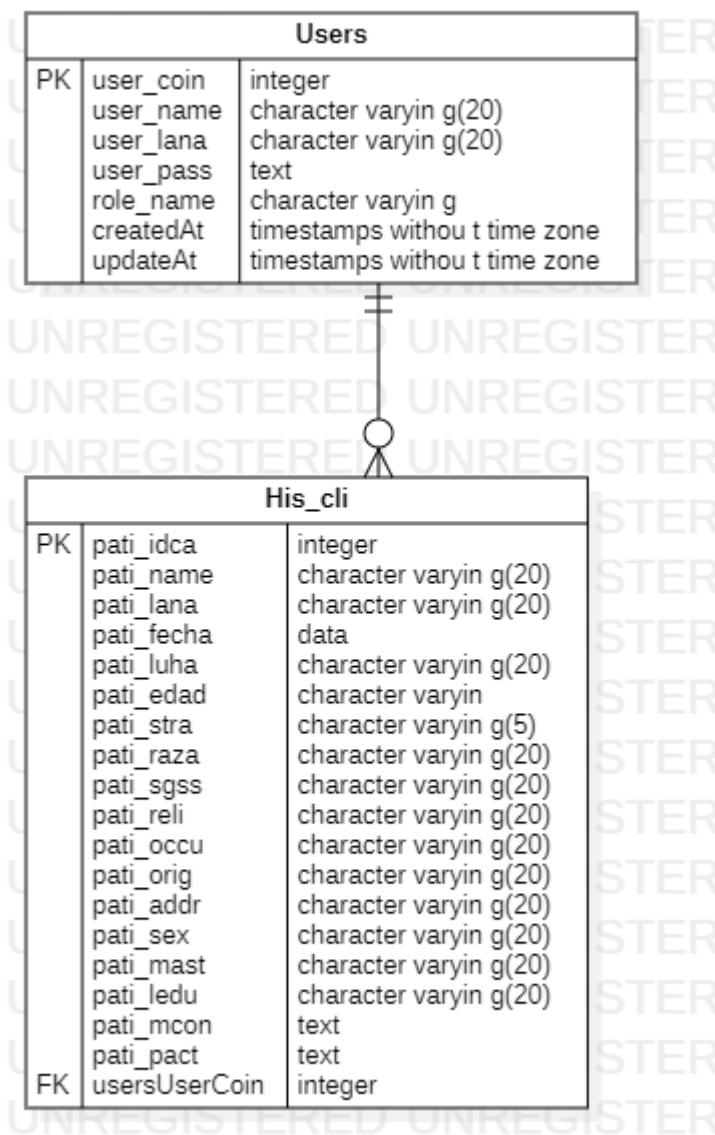
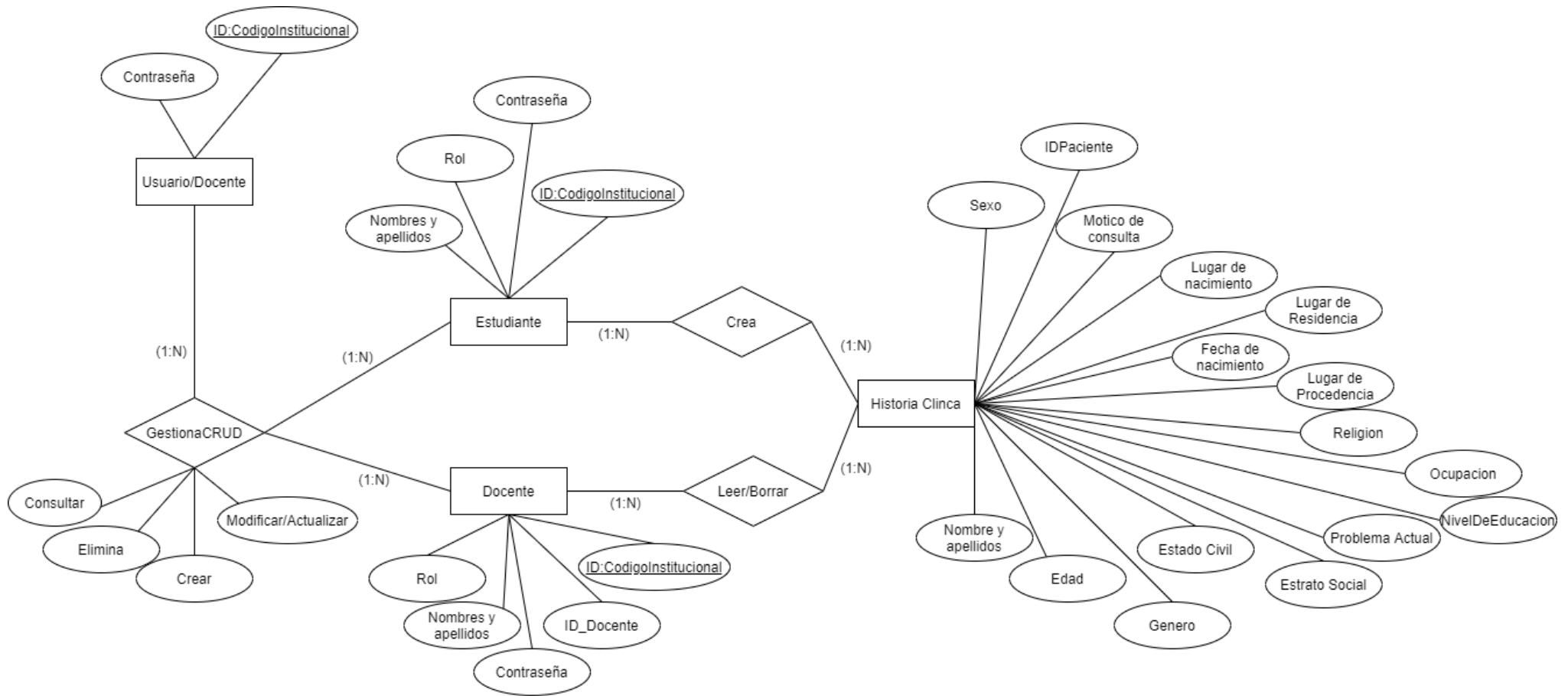


Figura 26. Diagrama Relacional Base De Datos

Fuente 26. Elaboración propia elaborado en StarUML

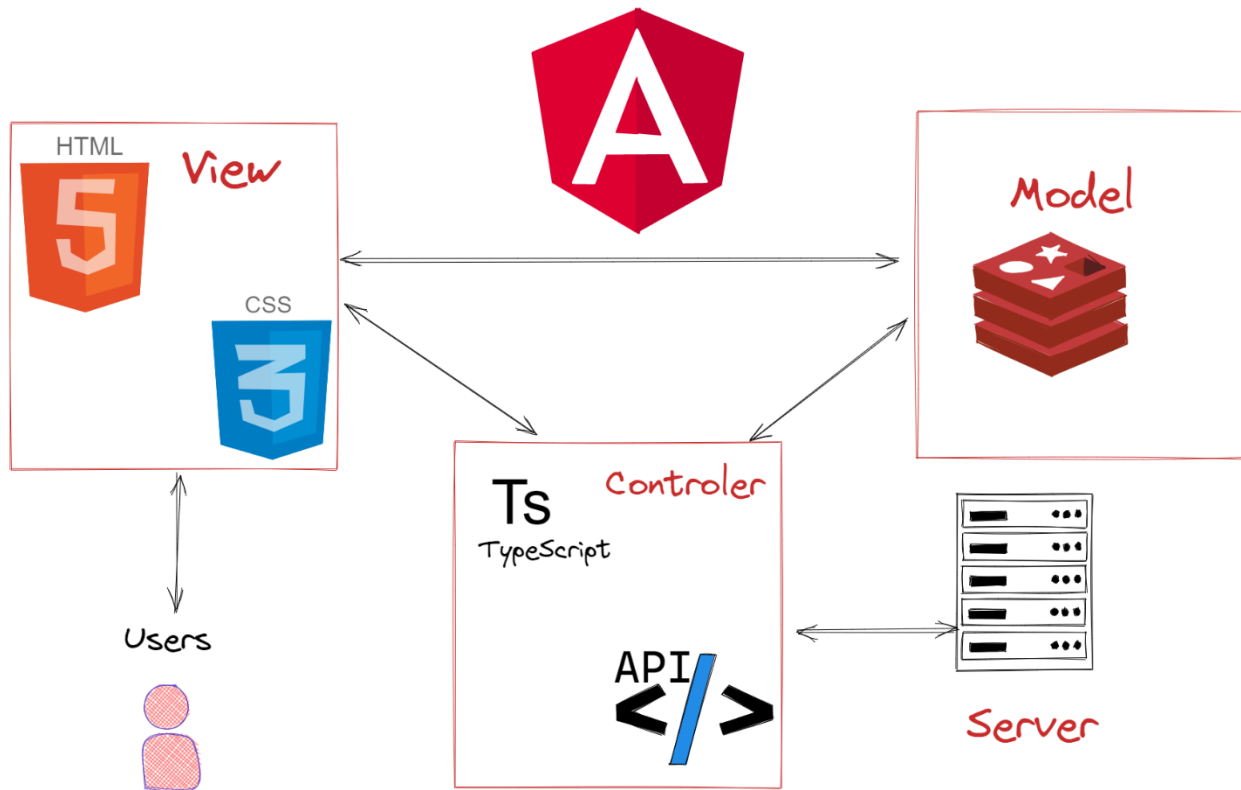
7.2.3. Modelo Entidad Relación



Fuente 27. Elaboración Propia

Figura 27. Modelo Entidad Relación

7.2.4. Modelo Vista Controlador



Fuente 28. Elaboración propia elaborado en Excalidraw

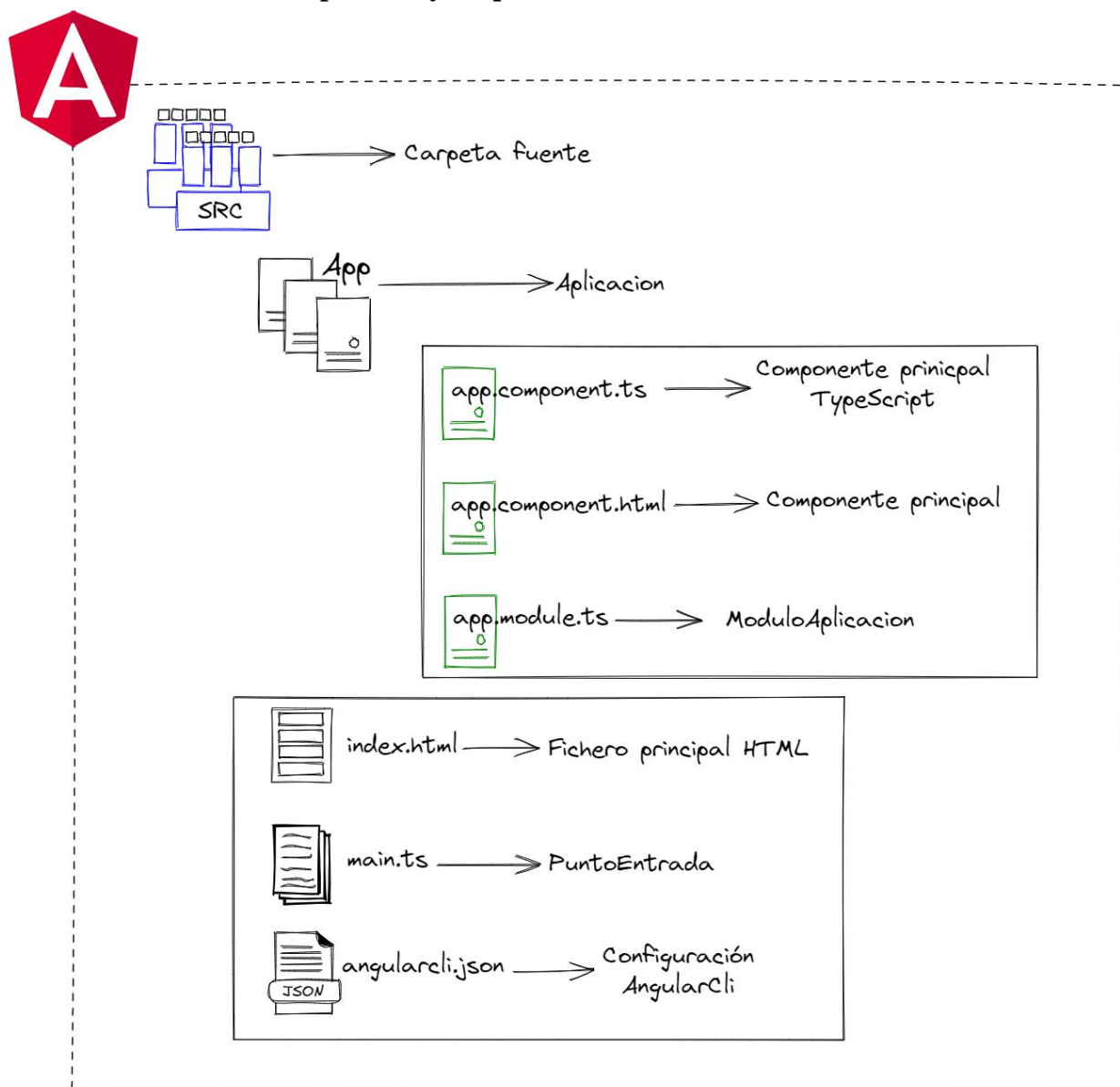
Figura 28. Diagrama De Modelo Vista Controlador Angular

7.2.5. Definición De Las Herramientas De Desarrollo

7.2.5.1. Entorno Operativo

El desarrollo del prototipo web de historias clínicas digitales en su versión 1.0, con diversas funciones y beneficios para su implementación en producción, dado que es un sistema Web fácil de usar, y se podrá acceder a él desde cualquier lugar y dispositivo inteligente que cuente con una conexión a internet, debido a que es desarrollado con tecnologías Web que son ANGULAR, HTML5, CSS, TYPESCRIPT, JAVASCRIPT, haciendo uso de una base de datos en POSTGRESQL, la cual almacena toda la información que será registrada en el sistema, ya sea de usuarios o las historias clínicas.

7.2.5.2. Composición y Arquitectura



Fuente 29. Elaboración propia elaborado en Excalidraw

Figura 29. Arquitectura de ficheros angular

Según la documentación oficial de angular, definen la arquitectura como:

La arquitectura de las aplicaciones Angular se basa en ciertos conceptos básicos. El mecanismo de construcción básico es NgModules, que provee el entorno de construcción para el componente. NgModules compila el código relacionado en un vinculado de funciones; las

aplicaciones angulares se precisan mediante un conjunto de NgModules. Una aplicación normalmente contiene al menos un módulo raíz de arranque y usualmente tiene más módulos funcionales.

Un componente define una vista, que es un conjunto de elementos de pantalla que Angular puede seleccionar y modificar de acuerdo con la lógica y los datos del programa.

Los componentes utilizan servicios que proporcionan funciones específicas que no están directamente relacionadas con la vista. Los proveedores de servicios pueden inyectarse a sí mismos como dependencias en los componentes para hacer que su código sea modular, reutilizable y eficiente. ("Angular", 2021).

7.2.6. Librerías Y Funciones

7.2.6.1. Bycript

Es una función de hashing de password, escrita en JavaScript lo que permite trabajar de forma simple y eficiente. Cuando se habla de hashing o hash, hace referencia a un método o algoritmo matemático para generar seguridad en los datos, ya que genera claves o llaves que representen un dato, también permite la identificación de un registro, es decir, transforma cualquier dato en una serie de caracteres con una longitud fija.

7.2.6.2. Class-Validator

Permite el uso de validación, internamente se utiliza para realizar la validación de clases, verifica que los campos definidos existan y contengan los valores correctos.

7.2.6.3. Cors

Sus siglas (cross-origin resource sharing) en español intercambio de recursos de origen cruzado,

es un mecanismo de seguridad del navegador que restringe las solicitudes HTTP de origen cruzado.

7.2.6.4. Helmet

Ayuda a proteger las aplicaciones desarrolladas en Express configurando los encabezados del HTTP.

7.2.6.5. JsonWebToken

JSON Web Token (JWT) es un estándar RFC 7519, que es un modelo basado en la identidad de notificaciones, define una forma compacta y autónoma de transmitir información de manera segura entre los fragmentos de un objeto JSON. Une el manejo de TOKEN, que es un identificador que regresa los datos en un proceso de sustitución de elementos sensibles por un equivalente no sensible. ("Conozca JWT (JSON WEB TOKEN) - programador clic", 2021)

7.2.6.6. Node-Postgres

PG Es una librería que se maneja para la conexión entre PostgreSQL y NodeJS, para realizar las conexiones básicas de las bases de datos, en pocas palabras crear una conexión para realizar CRUD (Crear, leer, actualizar y borrar).

En la documentación oficial de Pg en npm hace referencia a las siguientes funciones:

- El cliente JavaScript puro y los enlaces libpq nativos comparten la misma API
- Agrupación de conexiones
- JS extensible ↔ coerción de tipo de datos PostgreSQL
- Características de PostgreSQL compatibles:
 - Consultas parametrizadas
 - Instrucciones con nombre con almacenamiento en caché del plan de consultas
 - Notificaciones asincrónicas con LISTEN/NOTIFY

- Importación y exportación a granel con COPY TO/COPY FROM

7.2.6.7. *Reflect-Metadata*

Reflect se utiliza para describir el código que es capaz de inspeccionar otro código en el mismo sistema. Nos permite examinar un objeto desconocido en tiempo de ejecución y descubrir todo al respecto.

7.2.6.8. *TypeORM*

ORM es un mapeo objeto-relacional, es un método en programación para conversión de datos entre sistemas que no sean compatibles utilizando lenguajes de programación orientado a objetos.

Esto desarrolla un efecto de “base de datos de objetos virtuales” que se puede utilizar dentro del lenguaje de programación.

7.2.6.9. *Concurrently*

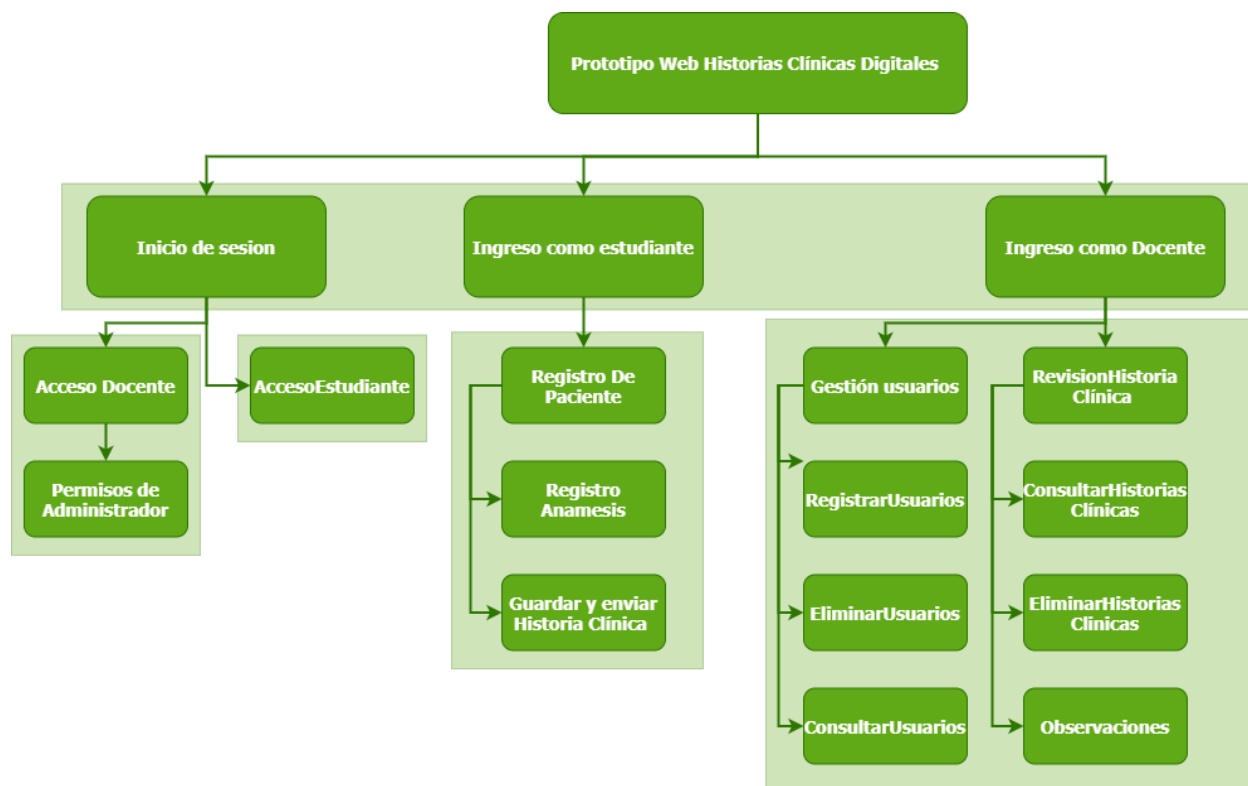
Permite ejecutar el servidor BackEnd Express y el servidor Node al mismo tiempo, se usa solo agregando scripts dentro del package.json.

7.2.6.10. *Rxjs*

Es una librería para la programación reactiva, facilita la composición de código asíncronico o basado en devolución de petición. Mejora el rendimiento, y las peticiones depurables.

7.3. Etapa: Diseño

7.3.1. *Diagramas De Esquemmatización Del Prototipo*



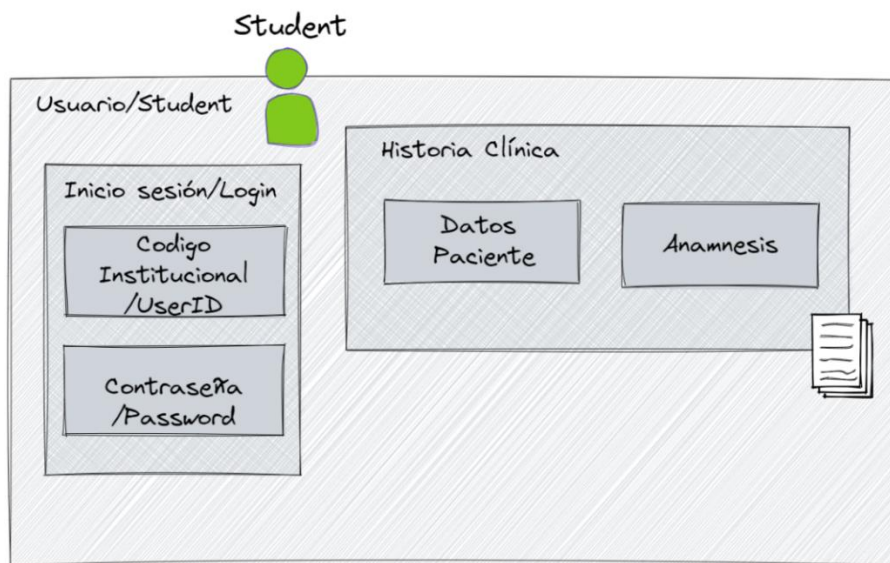
Fuente 30. Elaboración propia elaborado en Draw.io

Figura 30. Diagrama esquema prototipo

7.3.2. Maquetación y Diseño de Interfaces

Para la realización de prototipos se realiza la maquetación, que es un diseño digital de una aplicación y/o plataforma web. Se utiliza en la fase de diseño para visualizar las ideas y conceptos en el contexto del diseño web, la navegación del sitio, entre otras características de este. Este diseño de interfaces puede ser realizado en herramientas especiales de maquetado, en este caso se utilizó Excalidraw para la realización de los diseños de las historias clínicas digitales.

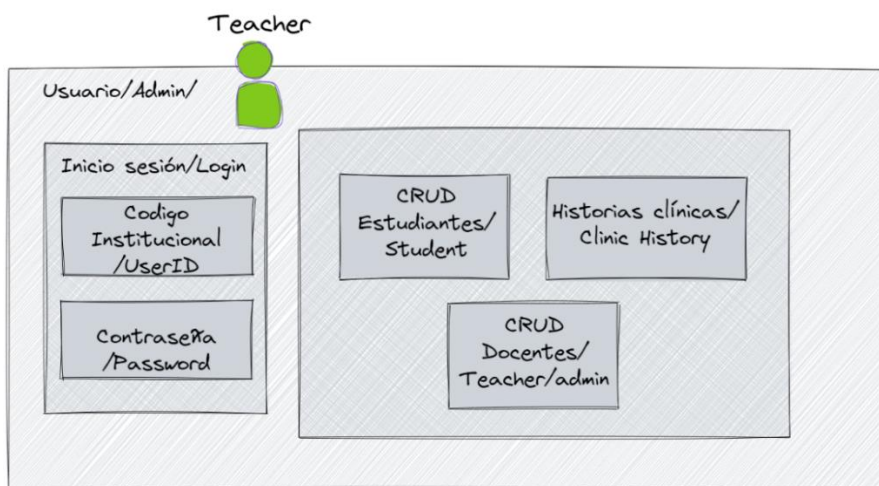
A. Determinación de contenido en el módulo de estudiante/Student.



Fuente 31. Elaboración propia elaborado en Excalidraw

Figura 31- Contenido de usuario estudiante

B. Determinación de contenido en el módulo de Docente/teacher /Admin



Fuente 32. Elaboración propia elaborado en Excalidraw

Figura 32. Contenido de usuario Docente

C. Diseño ejemplo de Login/inicio de sesión.

Username

test@mail.com

Password

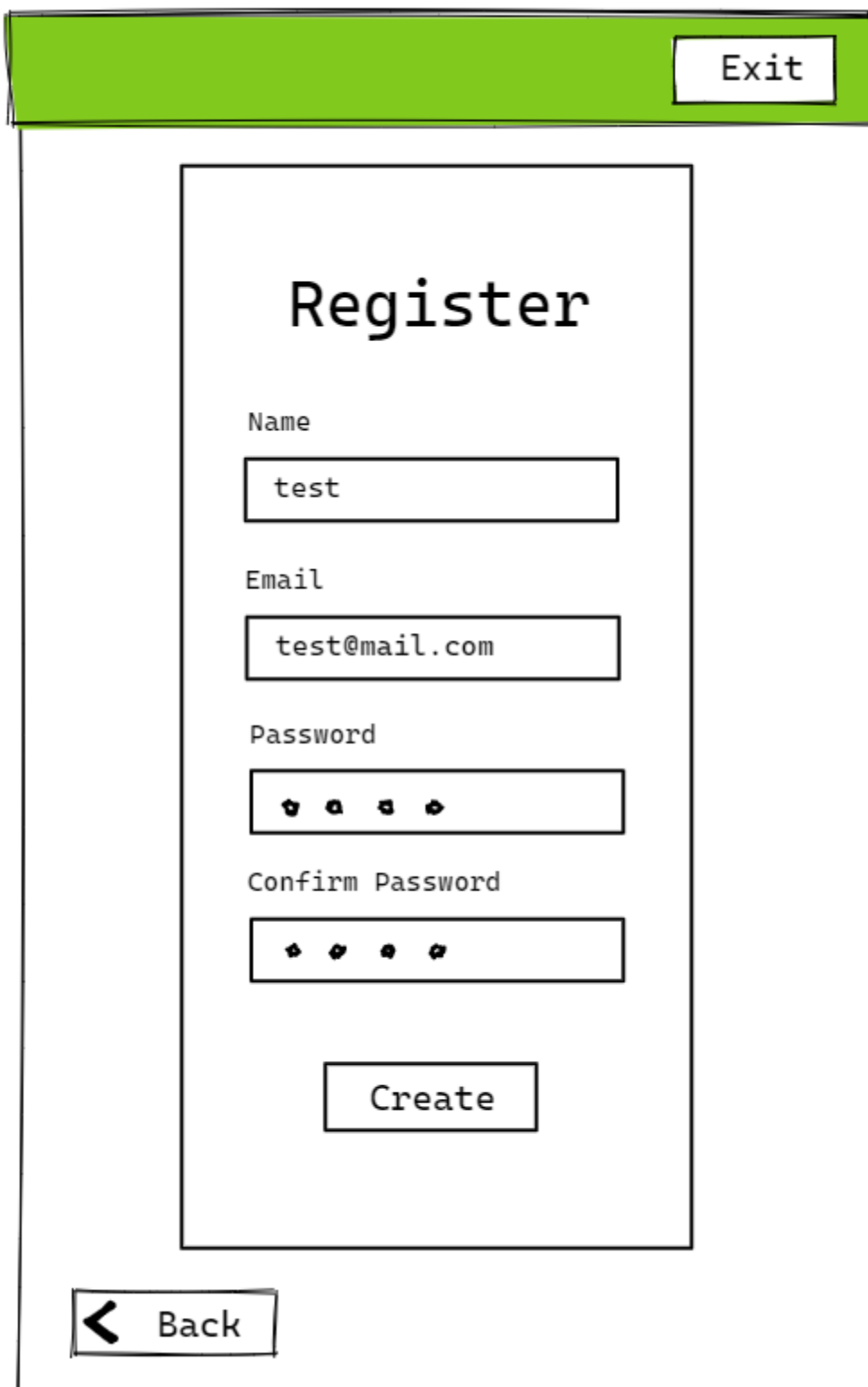
Forgot password?

Sign in

Fuente 33. Elaboración propia elaborado en Excalidraw

Figura 33. Diseño página login

D. Diseño ejemplo para el registro de un usuario.



The image shows a user registration form design. At the top, there is a green header bar with a white button labeled "Exit". Below the header, the main content area is a white rectangle with a black border. Inside this area, the word "Register" is centered in a large, bold, black font. Below the title, there are four input fields, each with a label above it: "Name" (containing "test"), "Email" (containing "test@mail.com"), "Password" (containing four asterisks), and "Confirm Password" (containing four asterisks). Below the input fields is a white button labeled "Create". At the bottom left of the main content area, there is a white button with a black left-pointing arrow and the text "Back".

Fuente 34. Elaboración propia elaborado en Excalidraw

Figura 34. Diseño página registro de usuario

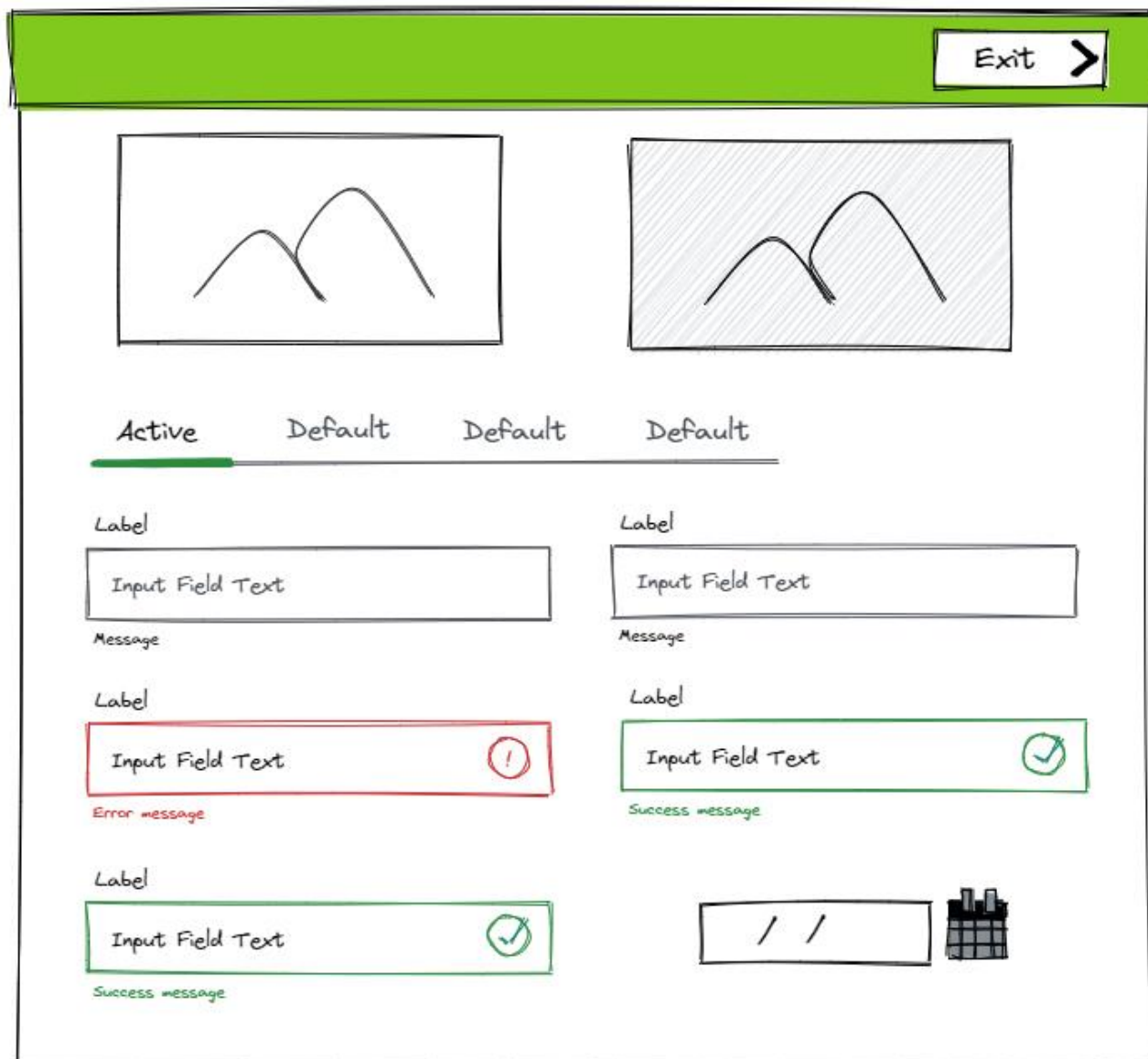
E. Diseño ejemplo para registro de paciente

The image shows a hand-drawn user registration form for a patient. The form is contained within a white rectangular area with a black border. At the top of this area, the word "Pacient" is written in a simple, hand-drawn font. Below the text is a purple, textured icon of a person's head and shoulders. Underneath the icon are four horizontal bars of varying lengths, representing input fields for text. At the bottom of the form area is a rounded rectangular button with the word "Registrar" and a right-pointing arrow. The entire form is set against a light green background that has a darker green header bar at the top right containing the text "Exit" and a right-pointing arrow.

Fuente 35. Elaboración propia elaborado en Excalidraw

Figura 35. Diseño de registro de usuario

F. Diseño ejemplo formulario de historia clínica

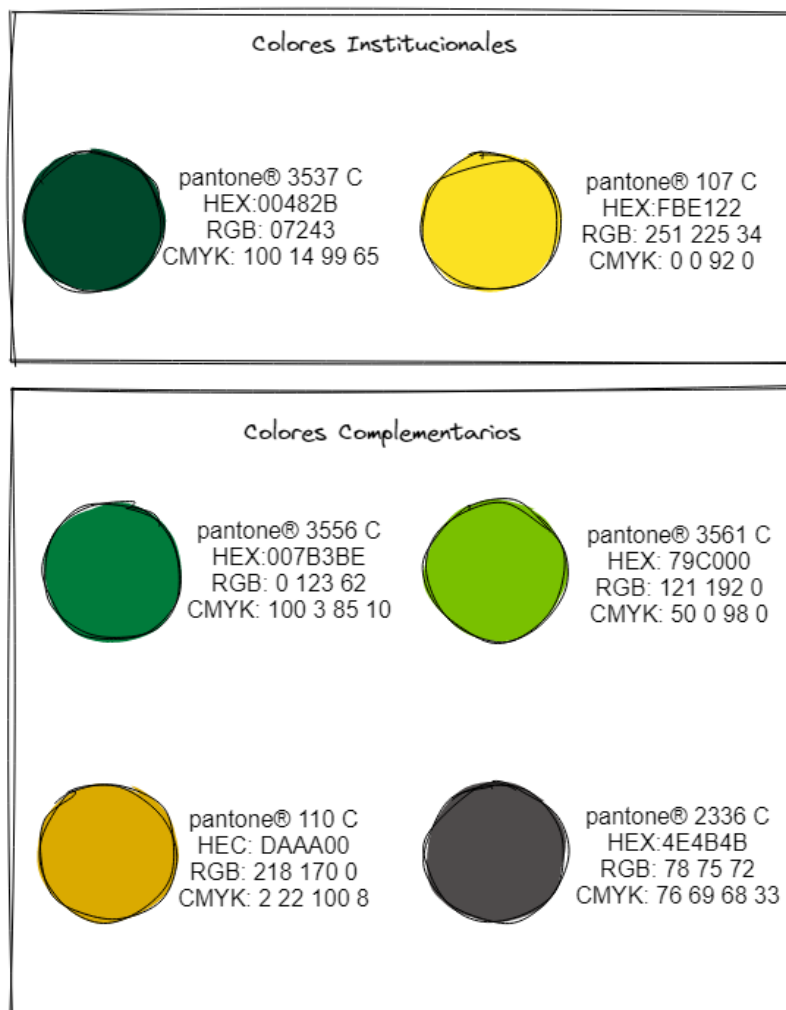


Fuente 36. Elaboración propia elaborado en Excalidraw

Figura 36. Diseño formulario historia clínica

7.3.3. Definición De Temática

Para la realización del prototipo, se solicita como requerimiento manejar los colores institucionales los cuales se ven reflejados en la siguiente paleta de colores.



Fuente 37. Elaboración propia elaborado en Excalidraw

Figura 37.. Paleta de colores institucionales

Simbología e Iconografía: Angular provee un material extenso para el uso de símbolos e iconos, son identificables y llamativos para comprender el uso del prototipo. Los botones y elementos gráficos que se utilizan a partir de librerías importadas del repositorio Angular Materials <https://material.angular.io/>, se utilizan como recurso para que el usuario interactúe en el sistema.

7.3.4. Logo e Iconografía

Al aplicativo web se le definió el nombre UDECLINICK.com haciendo uso del nombre de la Universidad, y enfocándolo al ámbito al cual pertenece este, así dando un sentido de pertenencia y que al momento de hacer alusión a él se le pueda llamar por su nombre.

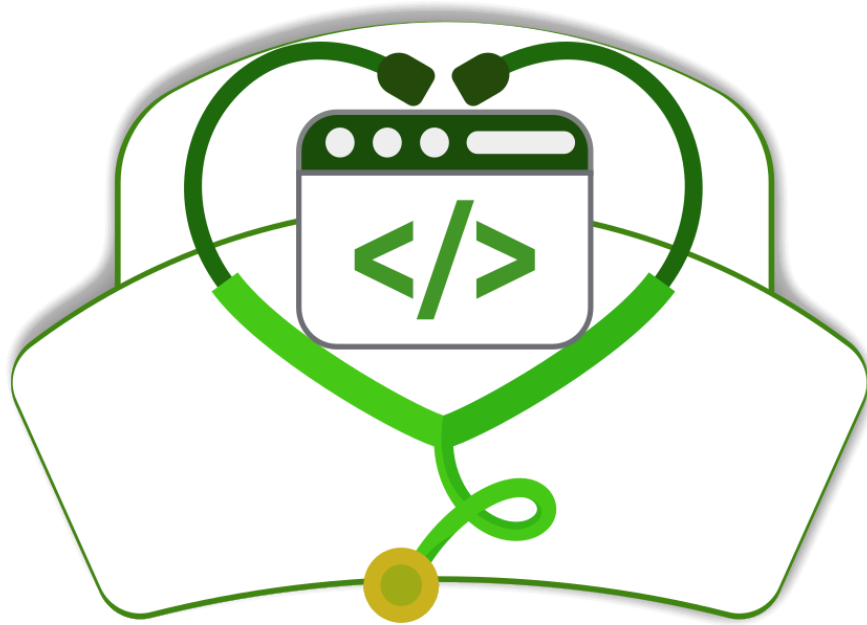
Se creó el logo en la herramienta CorelDraw, su estructura combina una página de navegación dando sentido a que es un aplicativo web, y una cofia, es decir, una especie de sombrero que usan los enfermeros.



Fuente 38. Elaboración propia elaborado en CorelDraw

Figura 38. Logo del prototipo

Se anexó un isologo, símbolo o representación gráfica que busca la representación del concepto a manejar pero de forma más simplificada, permitiendo usarlo en espacios reducidos y que sea visible y adaptable.



Fuente 39. Elaboración propia elaborado en CorelDraw

Figura 39. Isologo del prototipo

7.4. Etapa: Construcción De Prototipo

7.4.1. Desarrollo del Prototipo

El desarrollo del proyecto se lleva a cabo con la tecnología de NodeJs con la ayuda de la herramienta de framework Angular, en la cual queda la estructura del proyecto basado en objetos, y el modelo MVC (modelo vista controlador), en el proceso de construcción del software se realizó utilizando todo el contenido de Diseño y Análisis presentado previamente;

la gestión de almacenamiento y control de versiones se utilizó la herramienta GitHub, siento este de utilidad para el control de versiones y colaboración entre desarrolladoras.

7.4.1.1. Creación Del Programa En Node

Para iniciar con el desarrollo del proyecto, y después de instalar las herramientas necesarias, en Node se crea el proyecto denominado “Historias_Clinicas-master”, se utiliza NPM (Node Package Manager) que viene configurado con la instalación de NodeJs para gestionar las dependencias de un proyecto, así incrementando el desarrollo y reduciendo el tiempo.

Lo siguiente que se realizó, fue la creación de los ficheros y componentes que se necesitaron para crear el programa. Se dividió en dos carpetas principales, según el modelo vista controlador, en la primera carpeta denominada “ControllerExpressjs” se emplea MC (Modelo controlador) donde en el modelo se gestionó todos los accesos de la información o datos, las peticiones, consultas y actualizaciones, también, los privilegios de acceso, entre otras funcionalidades; Por otra parte, en el controlador se desarrolla los posibles eventos que el sistema debe responder, es decir, se desarrolla las peticiones que puede generar el usuario e invoca al modelo para determinar la solicitud de información por parte del usuario.

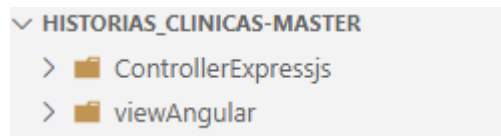


Figura 40. Organización ficheros, proyecto actual

7.4.1.2. Desarrollo BackEnd

El desarrollo Back-End está en la carpeta ControllerExpressjs que se estructura de diferentes archivos y componentes.

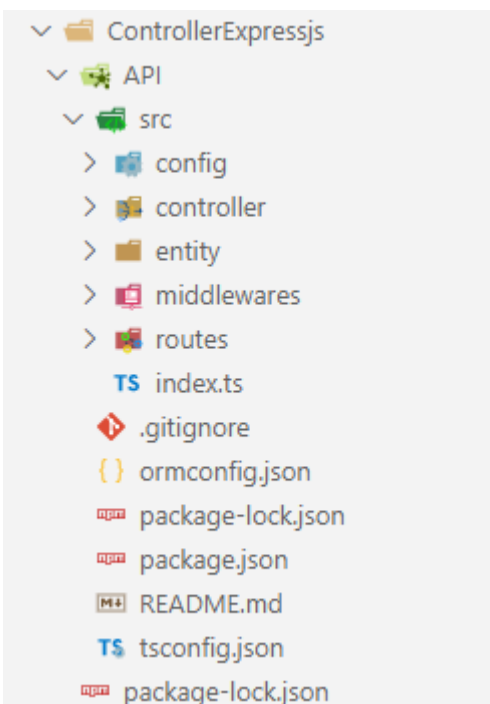


Figura 41. Carpeta Back-End

Para la realización del proyecto, inicialmente se reunió todo el análisis hecho con anterioridad y se empieza a hacer la interconexión del proyecto con las dependencias o bibliotecas que se necesitaron para establecer la conexión con la base de datos como la seguridad del proyecto. En donde nosotros identificamos que se exportan en el package-lock. json, que se genera automáticamente cuando se crea el proyecto. En este archivo agregamos la dependencia de Express para iniciar el marco de desarrollo Back-End, pero también fue necesario la utilización de comandos, para implementar esta dependencia y la utilización de puertos para

ejecución del sistema.

```
Historias_Clinicas-master > ControllerExpressjs > API > package.json > ...
1  {
2    "name": "API",
3    "version": "0.0.1",
4    "description": "Awesome project developed with TypeORM.",
5    "devDependencies": {
6      "@types/bcryptjs": "^2.4.2",
7      "@types/cors": "^2.8.12",
8      "@types/express": "^4.17.13",
9      "@types/jsonwebtoken": "^8.5.5",
10     "@types/node": "^8.10.66",
11     "ts-node": "3.3.0",
12     "ts-node-dev": "^1.1.8",
13     "typescript": "^4.4.3"
14   },
15   "dependencies": {
16     "bcryptjs": "^2.4.3",
17     "class-validator": "^0.13.1",
18     "cors": "^2.8.5",
19     "express": "^4.15.4",
20     "helmet": "^4.6.0",
21     "jsonwebtoken": "^8.5.1",
22     "pg": "^8.4.0",
23     "reflect-metadata": "^0.1.10",
24     "typeorm": "0.2.37"
25   },
  ...
}
```

Figura 42. Declaración de dependencias

Después de instalarlas se realiza un Script para que el navegador que se esté utilizando logre compilar. En el mismo se hace un llamado Dev, que hace un set de las variables que se utilizan, y se llama a Node de desarrollo, se hace la transferencia para ejecutar el index.ts, que más adelante se explica su funcionamiento.

```

Depuración de ▶
26   "scripts": {
27     "tsc": "tsc",
28     "dev": "set debug=* && ts-node-dev --respawn --transpile-only ./src/index.ts"
29   }
30 }

```

Figura 43. Scripts para compilación del proyecto.

Después de instalar el TypeScript, se crea automáticamente un archivo que se llama `tsconfig.json`

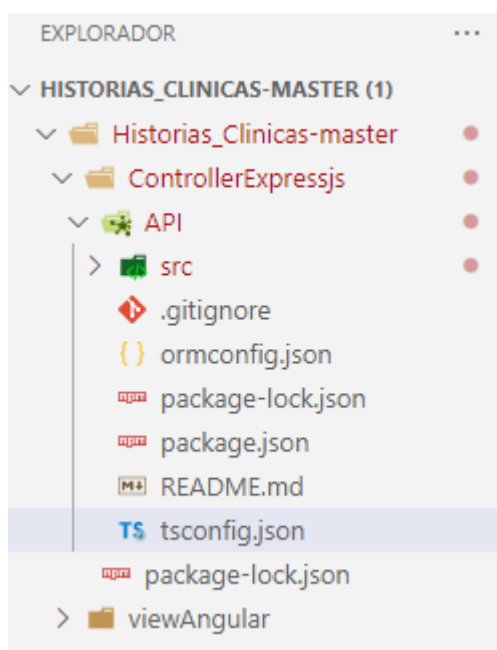


Figura 44. Creación del archivo `tsconfig.json`

En este archivo se configura las opciones de compilación, donde nosotros exportamos las librerías ECMAScript, que es un estándar que define la interoperabilidad de las páginas web en diferentes navegadores Web. Dentro del desarrollo del proyecto, manejamos dos versiones, ES5 y ES6 para que las funcionalidades que se utilizan tengan mejores compatibilidades con

diferentes navegadores que pueda usar el usuario final, y los procesos de desarrollo en la versión ES6 sean óptimos.

```
Historias_Clinicas-master > ControllerExpressjs > API > TS tsconfig.json > ...
 1  {
 2    "compilerOptions": {
 3      "lib": [
 4        "es5",
 5        "es6"
 6      ],
 7      "target": "es5",
 8      "module": "commonjs",
 9      "moduleResolution": "node",
10     "outDir": "./dist",
11     "rootDir": "./src",
12     "emitDecoratorMetadata": true,
13     "experimentalDecorators": true,
14     "sourceMap": true
15   }
16 }
```

Figura 45. Importación ECMAScript

Se utilizó CommonJs, como sistema de formato de módulos. Es un estándar para estructurar y constituir el código JavaScript. Colabora en el desarrollo de aplicaciones en la parte de servidor, este influye en la administración de módulo de NodeJs. Según lo anterior, un módulo se entiende por un conjunto de objetos y funciones realizadas en JavaScript, que se puede exportar para utilizarlas en otros archivos.

Regresando al archivo package.json donde se instalaron todas las librerías y por medio de investigaciones, se encontró una nueva tecnología llamada ORM, como anteriormente se mencionó, Object Relational Mapping o en español “mapeo objeto relacional”. Es un método o técnica utilizada en programación para transformar datos que existen en un lenguaje de programación orientado a objetos y la base de datos relacional. Se encarga de traducir la

instrucción del lenguaje de programación a sentencias SQL para que el gestor de la base de datos pueda entender. Es decir, permite la interacción de la base de datos con el sistema sin conocer las sentencias SQL según la base de datos, mediante la programación orientada a objetos. Esto nos da grandes beneficios para el manejo de la base de datos en el sistema, es un modelo de programación, el cual cumple con el objetivo de transformar las tablas de una base de datos a un modelo simplificado, donde al programador se le facilite la reducción de código. También, permite implementar diferentes gestores de bases de datos en un mismo proyecto, haciendo que este sea escalable para futuras funciones. En este caso, solo se utiliza la base de datos PostgreSQL, pero se tiene la posibilidad de la utilización de Oracle, como es el caso de la Universidad De Cundinamarca.

En base a lo anterior, ORM nos da una utilidad óptima para el proceso de desarrollo, logrando interactuar de forma sencilla con la base de datos aplicada, sin tener que conocer las sentencias SQL o los parámetros que maneja cada gestor de base de datos según el caso, además de manejar una capa de acceso de datos contra ataques.

Según lo anteriormente mencionado, la instalación de ORM se realiza como si fuera una dependencia, lo que generará un archivo `OrmConfig.json`, en este se realiza la conexión a la base de datos, indicando el tipo de base de datos, cual es el puerto de la base de datos, cual es el usuario y contraseña, cual es la base de datos a utilizar; Se configura la sincronización y el manejo de asignaciones y Express-api-cli que se maneja según sea el caso.

Según lo anterior se explica por medio de la paginan NpmJS (2020) afirma que api-cli es:

Express-api-cli es una herramienta de línea de comandos que genera una estructura de proyecto exprés preconfigurada con los paquetes más utilizados que pueden acelerar su flujo de

trabajo de desarrollo. En lugar de perder el tiempo configurando la estructura de su proyecto, express-api-cli hace el trabajo pesado para que pueda concentrarse en la creación de esa increíble aplicación.

```
Historias_Clinicas-master > ControllerExpressjs > API > {} ormconfig.json > ...
 1  {
 2    "type": "postgres",
 3    "host": "localhost",
 4    "port": 5432,
 5    "username": "postgres",
 6    "password": "asde71.4",
 7    "database": "clinicalhistories",
 8    "synchronize": true,
 9    "logging": false,
10    "entities": [
11      | "src/entity/**/*.ts"
12    ],
13    "migrations": [
14      | "src/migration/**/*.ts"
15    ],
16    "subscribers": [
17      | "src/subscriber/**/*.ts"
18    ],
19    "cli": {
20      | "entitiesDir": "src/entity",
21      | "migrationsDir": "src/migration",
22      | "subscribersDir": "src/subscriber"
23    }
24  }
```

Figura 46. Integración de base de datos

En este archivo, se puede integrar las bases de datos a utilizar, en este caso únicamente Postgres.

Una vez definida la base de datos en el archivo index.ts se importa la conexión con TypeORM, express, cors, helmet. En este caso, empezamos a definir los middlewares, que es un sistema de servicios y funciones para un aplicativo, nos permite gestionar los datos, los servicios,

autenticaciones y gestión de la API. Se indica las peticiones de tipo json que va a solicitar el FrontEnd, se incluyen las rutas de la API, las que se están utilizando para los usuarios e historia clínica.

Cabe aclarar, que en el middleware se crean los tokens, en este caso se hace uso del archivo `config.ts` para definir la clave o llave secreta que se usa para la creación del token, cada token se codificó para que su duración sea de una hora máximo. Es decir, al momento de un usuario ingresar al sistema y no registrar ninguna actividad, no realiza ningún movimiento en la página, automáticamente el token se vencerá y tendrá que ingresar de nuevo.

A su vez, se implementaron y se desarrollaron las entidades, las cuales son una clase que se representa mediante un objeto que simboliza cada tabla de la base de datos, se importa la librería Entity, junto con otras que nos define los datos que se van a manejar en la base de datos, esto según ORM. También en las entidades, se incluye `class-validator`, para indicar qué datos no pueden ser vacíos y el mínimo de datos puede quedar, también la dependencia `Bycrypt` para la encriptación de datos.

En este archivo se desarrollan dos funcionalidades, la primera es el `hashPassword`, que utiliza el `Bycrypt`, en donde se encripta la contraseña del usuario, por medio de saltos e indica donde se almacena la contraseña. Después, indicamos la función `checkPassword`, donde se realiza la verificación de la contraseña, encripta el dato que el usuario utilizó como contraseña y lo compara con la que esta almacenada en la base de datos.

Luego de crear entidades, como es el caso de usuarios, se desarrollaron las rutas o direccionamientos. La página oficial de Express ("[Direccionamiento de Express](#)", 2021) define las rutas como:

El direccionamiento hace referencia a la determinación de cómo responde una aplicación a una solicitud de cliente en un determinado punto final, que es un URI (o una vía de acceso) y un método de solicitud HTTP específico (GET, POST, etc.). Cada ruta puede tener una o varias funciones de manejador, que se excluyen cuando se correlaciona la ruta.

En el caso de nuestro aplicativo, tenemos definidas por el momento, cuatro rutas, como se muestra en la imagen.

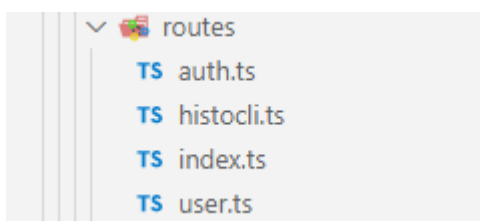


Figura 47. Archivos de rutas del sistema

En la primera ruta, que se define en el archivo Auth.ts se declara la URL <http://.../login>, se apunta a la dirección que el usuario utilizará para iniciar sesión, teniendo en cuenta los permisos que tiene cada usuario. En la segunda ruta, encontrada en el archivo user.ts, donde se utiliza Check JWT para la verificación de token y el CheckRole es la declaración del rol que puede hacer la petición en la ruta determinada. En el tercer archivo, denominado histocli.ts se inicializan las rutas necesarias para la creación de historia clínica. Y por último archivo index.ts, se recopilan ambas rutas anteriormente declaradas, y las rutas que se lleguen a declarar posteriormente.

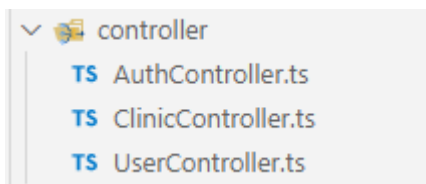


Figura 48. Archivos de controladores del sistema

En los controladores, hacemos uso de TypeORM importándolo para hacer el CRUD en la base de datos; utilizamos el request/response, que en pocas palabras es la petición que hace angular en la parte FrontEnd y la respuesta que se genera de parte de Express en BackEnd; después, se exportó las entidades, y el class validator. En los controladores, siendo los que interactúan entre el modelo y la vista, MV (Modelo vista), gestiona el flujo de información que hay entre ellos, también se encarga de realizar la adaptación de los datos para responder a las solicitudes de cada uno.

7.4.1.3. Desarrollo FrontEnd

En Angular se desarrolla por componentes, los cuales se estructuran por tres archivos, el primero es un archivo Css para el estilo, el segundo HTML para la parte visual, y el tercero TypeScript, es un lenguaje de programación de código abierto desarrollado por Microsoft, en este es en donde se desarrolló y en el compilador se traduce a código JavaScript; Así nos permite construir por aparte cada módulo del sistema que se identificó con anterioridad, y de manera organizada poder controlar lo que se desea en cada módulo.

Al momento de crear un proyecto en Angular, por defecto viene con archivos necesarios para empezar a realizar el desarrollo. En el primero que se empieza a desarrollar es en el tsconfig.json, donde se configuraron los path en la página oficial de java los define como :

PATH es la variable del sistema que utiliza el sistema operativo para buscar los ejecutables necesarios desde la línea de comandos o la ventana Terminal.

En este caso, hacemos uso de ellas para especificar las rutas de una forma organizada y más cómoda a la hora de importar librerías y carpetas. Es decir, en nuestro caso, al momento de utilizar una ruta solo hacemos uso de la variable declarada en el path

Por ejemplo, al llamar algún servicio, no queda la ruta “../src/Pages/auth/auth.service” si no, queda más simplificado con el “@app/Pages/auth/auth.service” obteniendo un resultado más simplificado y organizado estructuralmente.

Lo siguiente, fue en el package.json, en este se realiza un script que identifica la ejecución del proyecto con la librería Concurrently, que se definió anteriormente, se implementó con el objetivo de ejecutar tanto FrontEnd como BackEnd al mismo tiempo. Se definen cuales son los comandos que utiliza cada uno para ejecutar y se une usando la librería mencionada.

También en este mismo archivo, se añaden las dependencias que se utilizaran dentro del proyecto en la parte de FrontEnd en Angular, estas ya fueron definidas anteriormente.

Después de realizar el paso anterior, se procede a crear los componentes en este caso, lo estructuramos en componentes, está el componente share, Pages y enviroment. En la carpeta enviroment encontramos la configuración del Api- URL en donde indicamos donde está el BackEnd.

Otro importante archivo, fue la creación de material.module.ts, donde llamamos todas las librerías y funciones de Angular material, es un conjunto de componentes visuales que nos permiten desarrollar interfaces de usuario consistentes. Este conjunto de componentes los desarrolla el mismo equipo de Angular, el sitio material.angular.io/. Se dispone de un conjunto de componentes agrupados por categorías:

Form Controls: input, check box, radio button, select etc.

Navigation: menu, sidenav, toolbar

Data table: table, paginator, sort header

Popups y Modals: dialog, snackbar, tooltip, entre otros que se utilizaron para la creación de la historia clínica.

Luego de realizar ese procedimiento se hace la configuración de rutas en el paquete `routing.module.ts`. el concepto de enrutamiento lo define angular ("Angular", 2021) en sus:

El `NgModule Router` de Angular proporciona un servicio que te permite definir una ruta de navegación entre los diferentes estados de la aplicación y ver sus jerarquías.

Es decir, cuando se quiere que el usuario vea el módulo de login, la dirección o la ruta queda como `http://localhost4200/login`, y demás rutas que se necesitaron, por cada componente creado se especificó una ruta, usando la función `loadchildren` que se especificará más adelante.

En el componente `shared`, se encuentran los componentes que manejan el pie de página, donde se encuentra el logo institucional de la Universidad de Cundinamarca para todos los usuarios del sistema. En los dos componentes de header y slidevar, se encuentra el manejo por rol, es decir, según el tipo de rol se le mostrará la información que contiene cada componente.

En el componente `Pages`, tenemos otros subcomponentes donde se tiene el auth, donde está desarrollado el servicio `auth.service`, en este se encuentran todas las funciones desarrolladas para la verificación del inicio de sesión del usuario, si está activo o si no es correspondiente dentro del sistema. Se maneja el environment para hacer la conexión con el BackEnd y se hace la validación del JWT (Json Web Token), donde también se tiene un periodo de expiración, una vez terminado el tiempo de token, el sistema cierra la sesión del usuario. Esta información del

usuario es almacenada en el LocalStorage, este es la capacidad para almacenar información del usuario de manera local en el navegador que utiliza el usuario mientras navega.

Por otro lado, se realizó una descripción de la historia clínica en la carpeta home, ya que será lo primero que observara el estudiante al ingresar, y otra información suministrada por Enfermería, en esta se encuentran los Modals, que es en donde se tiene todo el formulario de la historia clínica, para que este se consuma con el BackEnd se creó una carpeta llamada “Services”, esta nos permitió el desarrollo de las funciones para la gestión de datos, es decir, se obtienen, se crean, se borran los datos que se solicitan. Esta carpeta también aplica el servicio para la gestión de usuarios, también se debe tener en cuenta que para que los datos sean correctamente enviados y recibidos, el enrutamiento que se le da debe ser el mismo en el BackEnd y FrontEnd.

En todo el desarrollo del aplicativo se tuvieron en cuenta características que se hace para una mejor experiencia de usuario y correcto funcionamiento y manejo del prototipo, con base en esto, en cada componente se desarrolló en caso de errores un aviso, no permitir al usuario dejar un campo vacío, la interfaz por usuario es diferente y contiene diferentes funciones, entre otras características que complementan el sistema.

7.4.2. Repositorio del proyecto

Se utilizó la herramienta para control de versiones de código GitHub, anteriormente mencionada, desde septiembre se utilizó como un repositorio virtual privado, que se conectó entre GitHub y Visual Studio Code permitiendo que los desarrolladores tengan un mejor control sobre las versiones del sistema.

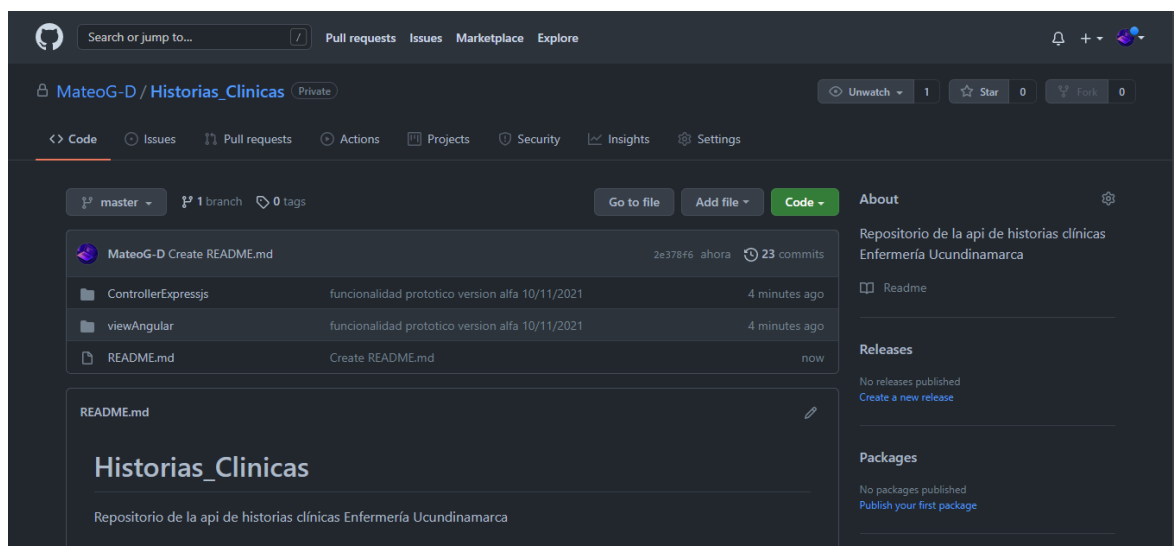
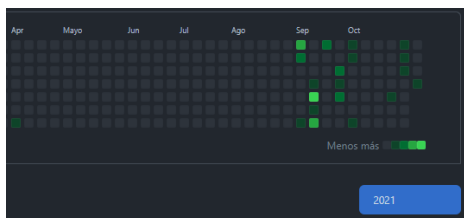


Figura 49. Repositorio en GitHub

Fuente 40. Captura de pantalla del repositorio creado en GitHub

Se subió al sistema con el nombre Historias_Clinicas, y se agregó un archivo readme.md para que se lean datos necesarios del aplicativo.



Fuente 41. Captura de pantalla del repositorio creado en GitHub

Figura 50. Actividad en el repositorio

Por ultimo, como se puede ver en la figura No 50, la mayor actividad se presento durante el mes de septiembre de 2021, debido a que el equipo de desarrollo empezó a subir el codigo a este medio. Sin embargo, se hicieron copias en un github local, y copias enviadas por correo institucional.

7.5. Etapa: Implementación

La implementación del proyecto se lleva a cabo en un servidor web apoyado por un hosting, con un dominio. Estas herramientas para desplegar el prototipo tuvieron costos que fueron asumidos por parte de los directores del proyecto como apoyo a los estudiantes desarrolladores.

El enlace final de acceso a la plataforma es: <https://udeclinickmy.web.app/>



Figura 51. Captura de pantalla de página web de UdeClinick en funcionamiento con el enlace de acceso

The screenshot shows the Clouding.io dashboard. At the top, there are navigation tabs for 'Servidores', 'DNS', and 'Cuenta'. The user is logged in as 'matus10hiut@gmail.com'. Below the navigation, there are tabs for 'Mis Servidores', 'Mis Snapshots', 'Mis Firewalls', and 'Mis Llaves SSH'. The 'Mis Servidores' tab is active, showing a table of servers. The table has columns for 'Servidor', 'Sistema Operativo', 'IP pública', and 'Estado'. One server is listed: 'UdeClinick' with 'Ubuntu 20.04 (64 Bit)' OS, IP '27.0.173.253', and status 'Activo'. Below the table, there are specifications: 'VCores 0.5', 'RAM 1 GB', and 'SSD 10 GB'. At the bottom, there is a pagination control showing 'Page: 1', 'Rows per page: 40', and '1 - 1 of 1'.

Figura 52. Captura del proyecto en Clouding.io

Se utilizó servidor de Clouding.io para subir el Back-End y la herramienta de Firebase para tener el hosting con la parte FrontEnd.

The screenshot shows the Firebase console for the project 'HistoriaClinicaUdeC'. The left sidebar contains navigation options for 'Compilación', 'Authentication', 'Firestore Database', 'Realtime Database', 'Storage', 'Hosting', 'Functions', 'Machine Learning', 'Lanzamiento y supervisión', and 'Extensions'. The 'Hosting' section is selected, showing a line graph of 'Downloads (7 días en total)' with a total of '15.7MB'. The graph shows a sharp increase in downloads starting on November 12th. To the right, there is a 'Historial de implementaciones' section showing a deployment titled 'Implementado' on '14 nov. 2021 00:42' by 'matus10hiut@gmail.com'. The 'Spark' plan is also visible at the bottom left.

Figura 53. Captura del proyecto en Firebase

7.6. Etapa: Validación de Requisitos del Sistema

Para la obtención de un producto funcional y que cumpla con lo solicitado por el cliente, es necesario realizar una etapa que valide si se cumplen con los requerimientos funcionales del sistema, por lo tanto realizamos una serie de pruebas evidenciadas con imágenes, que muestran el correcto funcionamiento de cada método desarrollado para satisfacer lo propuesto en el proyecto, esto se desarrolló en la herramienta POSTMAN.

También, se desarrolló una encuesta a los estudiantes de Enfermería, para comprobar el funcionamiento de la plataforma en tiempo real.

7.6.1. Postman

En el proyecto las pruebas y validaciones se realizaron a medida que se desarrollaba, se utilizó la herramienta Postman, es un cliente HTTP (Protocolo seguro de transferencia de hipertexto) que nos permite crear peticiones sobre la API, así generar pruebas de funcionamiento.

Al final de cada tarea se verificó con el requerimiento correspondiente, obtenidos por medio del marco de trabajo Scrum realizado con integrantes de la Facultad de Enfermería. Una vez desarrollado cada módulo, gestionando las direcciones, realizamos pruebas de tipo de petición o respuesta como lo son las funciones GET, POST, PUT, HEAD, DELETE, PATCH.

PAQUETE DE USUARIO

- ✓ RF-1: El sistema debe permitir crear, consultar, modificar y eliminar un usuario que hagan parte de la gestión de datos del aplicativo.
- ✓ RF-3: El sistema debe darle únicamente al usuario docente permisos para realizar CRUD de los demás usuarios.

- Peticiones por método GET, POST, PUT, DELETE, ingresando en Postman la dirección http que se configuró.
- Validación y pruebas de permisos de usuario

Como se ha mencionado anteriormente el aplicativo cuenta con dos tipos de roles, docente y estudiante. Cada uno tiene en el sistema permisos diferentes que son los que se desean validar y hacer pruebas para correcto funcionamiento según el requerimiento.

Para esto, en Postman se debe ingresar el enlace, ruta correspondiente dada en el sistema primeramente, se le debe de indicar que se ingresa con un usuario docente predeterminado con anterioridad, y configurando el token,

- Listar usuarios por medio de la ruta <http://localhost:3000/users>

Para hacer esta consulta se debe de haber hecho el paso anterior, del login, para que el uso del token nos permita dentro del sistema verificar el rol, si tiene los permisos para realizarlo.

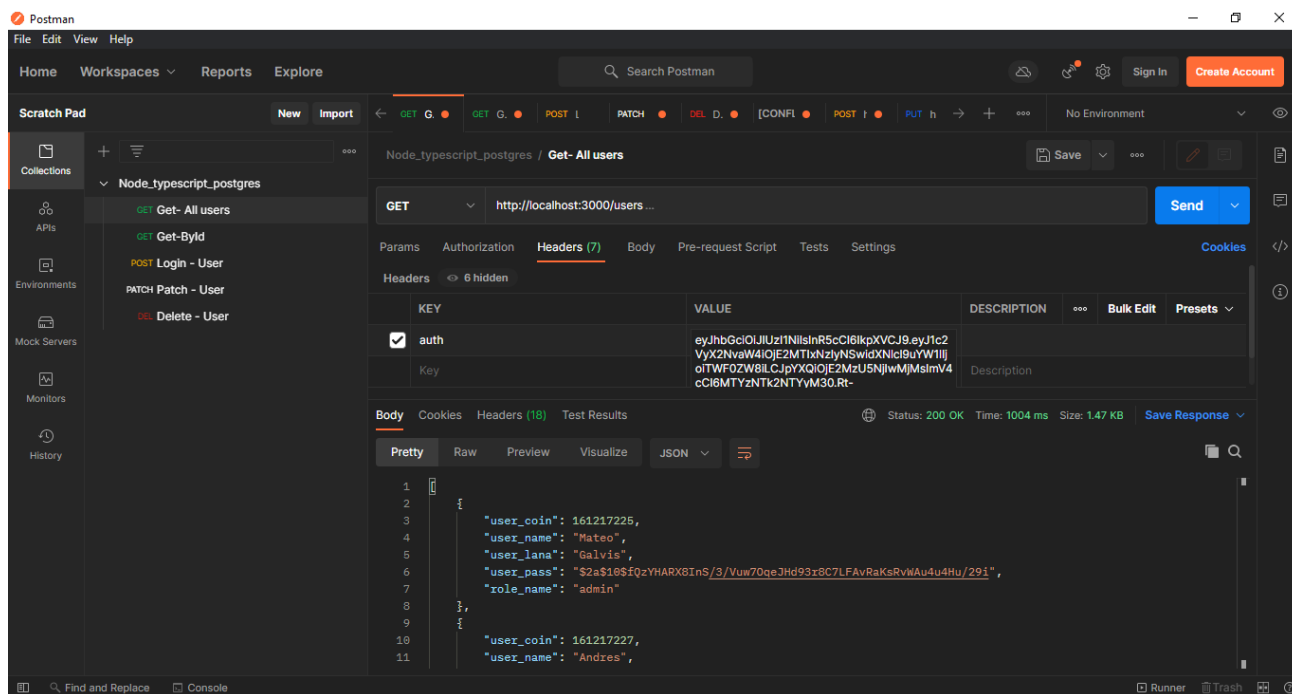


Figura 54. Validación en Postman "Listar usuarios"

Se hace una petición GET, indicando usuario que hace la petición, y la ruta declarada para el método.

Resultado esperado: listado de usuarios registrados, HTTP status 200.

Resultado alterno: Si el usuario que está haciendo la petición es un estudiante, o diferente al rol docente, el sistema no le debe de arrojar ningún dato.

- Crear Usuarios <http://localhost3000/users>

Se hace una petición POST, indicando el usuario que hace la petición, y la ruta declarada para la función. Al igual que el método anterior (listar Users) requiere del token para permitir crear usuarios en la plataforma.

Resultado esperado: Usuario creado, HTTP status 200. Mensaje "User created"

Resultado alternativo: Si el usuario que está haciendo la petición es un estudiante, o diferente al rol docente, el sistema no le debe de arrojar ningún dato. Si el usuario ya existe, arroja un mensaje de advertencia.

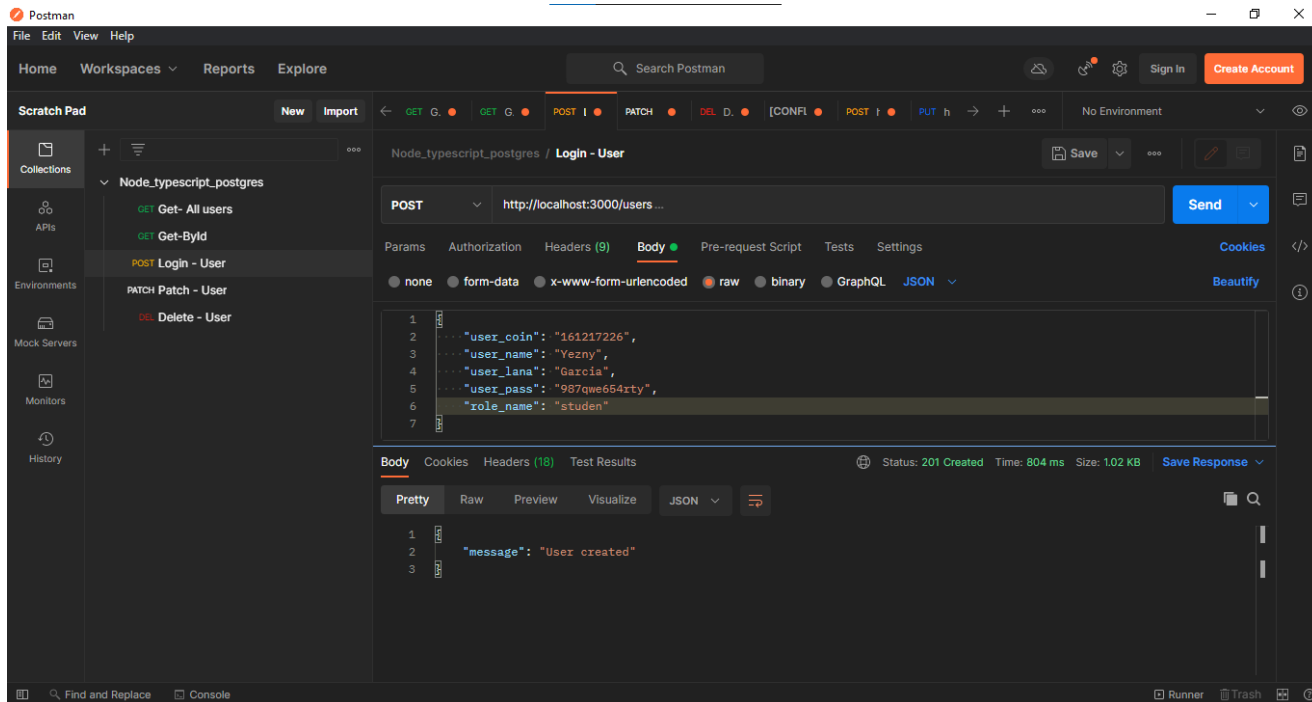


Figura 55. Validación en Postman "Crear usuarios"

- Modificar/Actualizar Usuario

Se hace una petición PATCH, indicando el usuario que hace la petición, y la ruta declarada para la función. Al igual que el método anterior (listar Users) requiere del token para permitir modificar usuarios en la plataforma. El usuario solo puede actualizar o modificar nombres, apellidos y rol.

Se le envía la ruta :

[http://localhost:3000/users/\(codigoinstitucionalquelepertenecealusuario\)](http://localhost:3000/users/(codigoinstitucionalquelepertenecealusuario))

Ejemplo: `http://localhost:3000/users/161217226`

Resultado esperado: Usuario modificado "Update User", HTTP status 201.

Resultado alterno: Si el usuario que está haciendo la petición es un estudiante, o diferente al rol docente, el sistema no le debe de arrojar ningún dato.

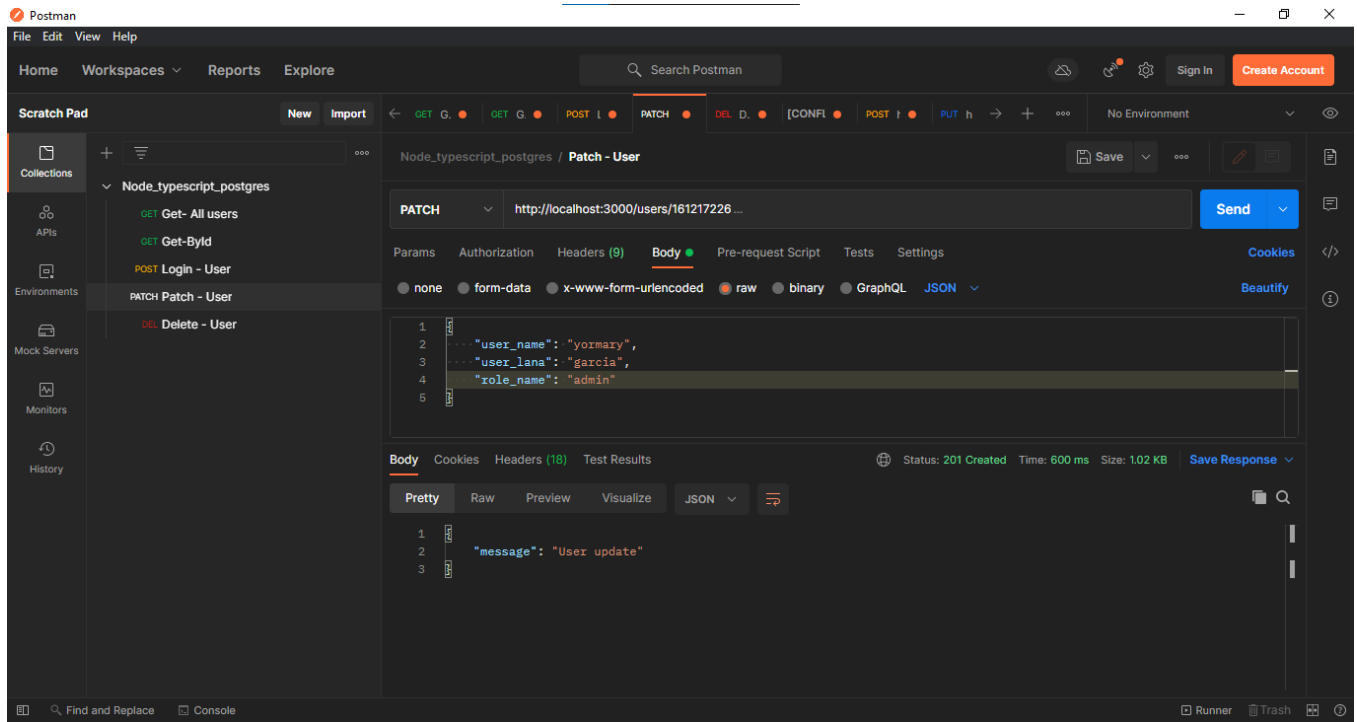


Figura 56. Validación en Postman "Modificar usuarios"

- Eliminar Usuarios <http://localhost:3000/users>

Se hace una petición DELETE, indicando usuario que hace la petición, y la ruta declarada para el método. Se le envía la ruta:

[http://localhost:3000/users/\(codigoinstitucionalquelepertenecealusuario\)](http://localhost:3000/users/(codigoinstitucionalquelepertenecealusuario)) Ejemplo:

`http://localhost:3000/users/161217226`

Resultado esperado: Usuario eliminado "User delete", HTTP status 201.

Resultado alternativo: Si el usuario que está haciendo la petición es un estudiante, o diferente al rol docente, el sistema no le debe de arrojar ningún dato. Si el usuario ya existe, arroja un mensaje de advertencia. No permitirá eliminar ningún usuario.

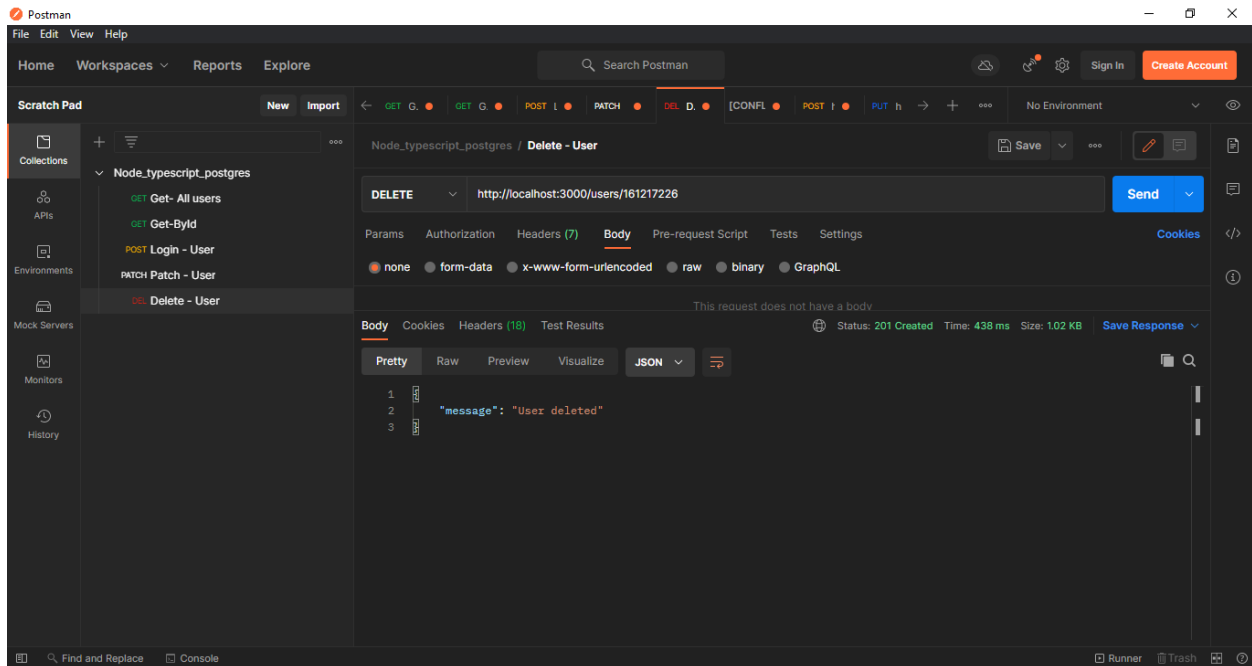


Figura 57. Validación en Postman "Eliminar usuarios"

PAQUETE GESTIÓN DEL SISTEMA.

- ✓ RF-11: El sistema debe tener un login de acceso para mayor seguridad y protección de datos.

El sistema controlará el acceso y permitirá acceder solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con su respectivo código institucional (que será el usuario) y contraseña.

- ✓ RF-12: El sistema debe contar con la funcionalidad de la tecnología de JWT (JSON web Token) para determinar el acceso de cada usuario

Para entender cuál es el usuario quien va a hacer las peticiones, se debe de ingresar al sistema a la ruta `http://localhost:3000/auth/login`, por el método POST, los datos de Código institucional, que es la llave primaria de la tabla usuario, y la contraseña asignada a este. El sistema arrojará una respuesta automática que será un TOKEN para la autenticación de usuario, un mensaje de confirmación “OK”, y el rol asignado.

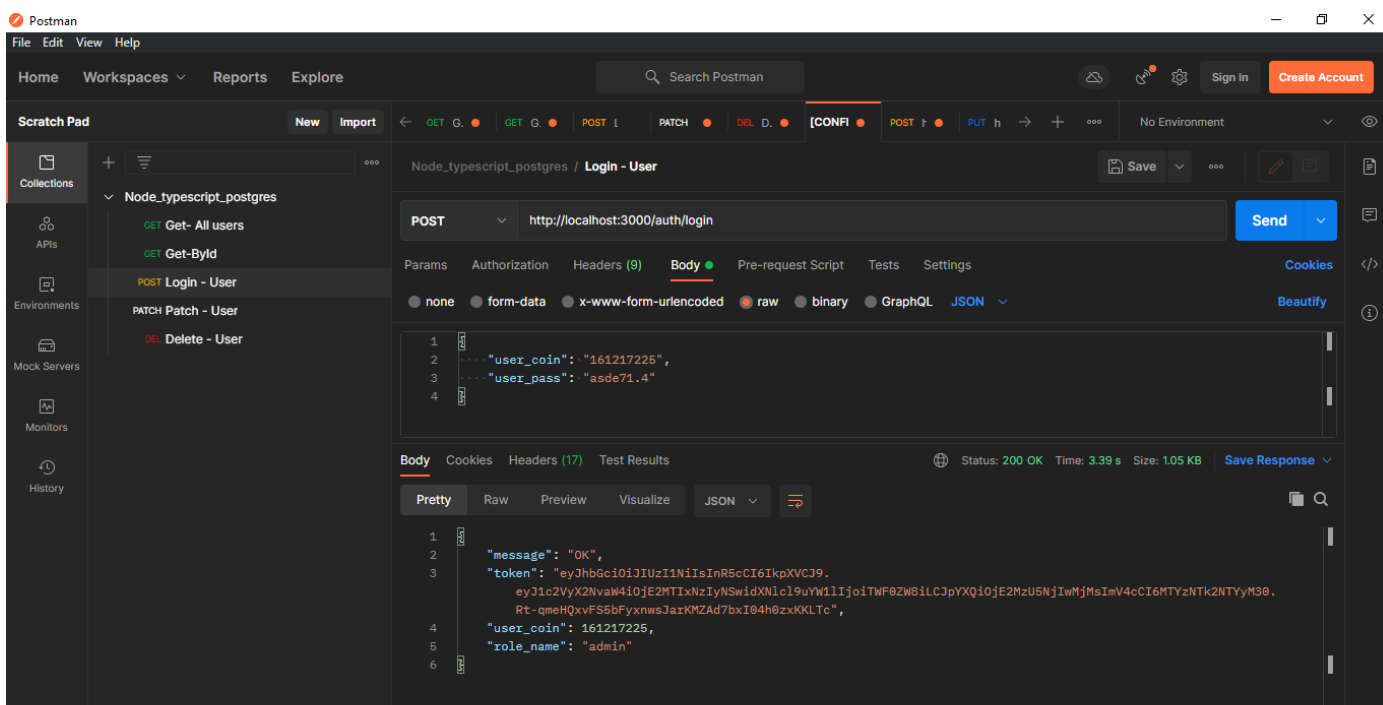


Figura 58. Validación en Postman "Login Admin"

- Login Estudiantes <http://localhost:3000/users>

Se hace una petición POST, indicando usuario que hace la petición, y la ruta declarada para el método. Se le envía la ruta: <http://localhost:3000/auth/login>.

Se hace una petición GET, indicando usuario que hace la petición, y la ruta declarada para el método

Resultado esperado: listado de usuarios registrados, HTTP status 200.

Resultado alternativo: Si el usuario que está haciendo la petición es un estudiante, o diferente al rol docente, el sistema no le debe de arrojar ningún dato.

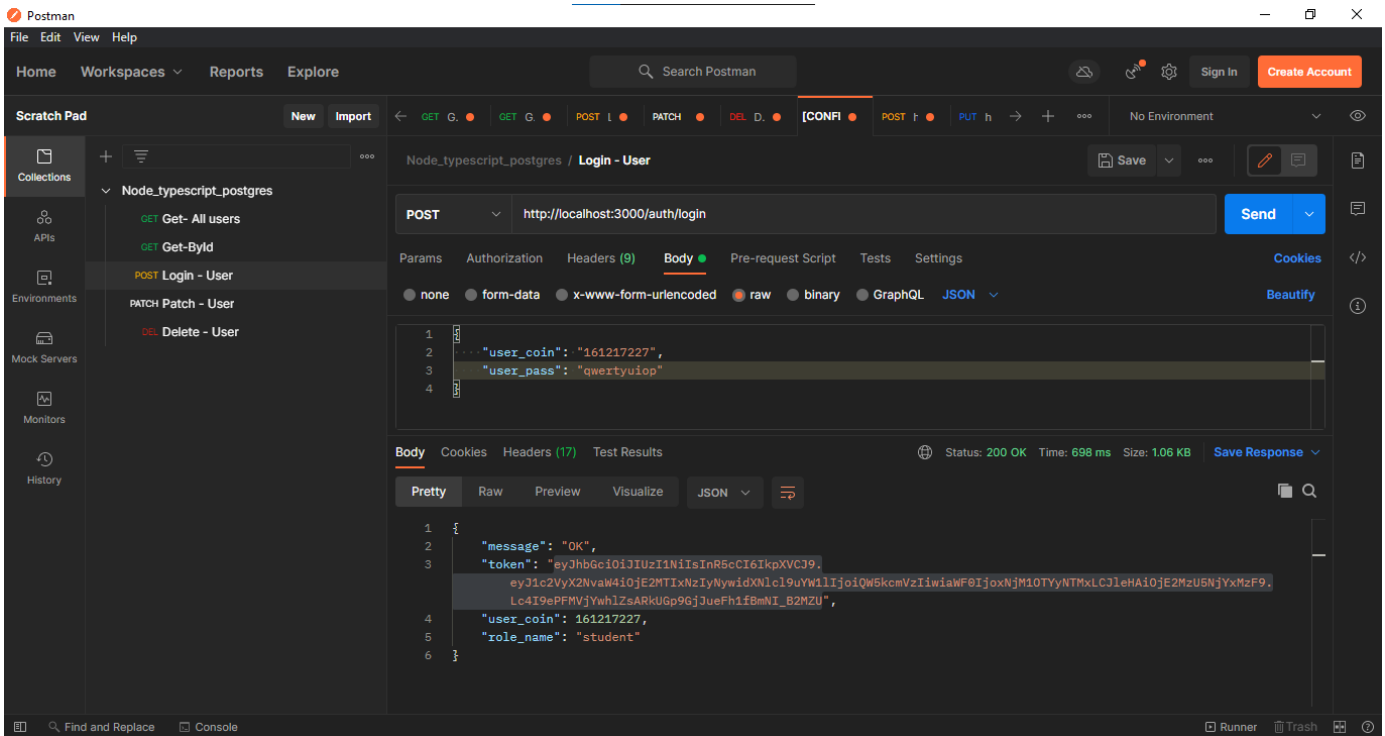


Figura 59. Validación en Postman "Login estudiante"

Para verificar si el usuario no tiene permisos, se copia el token que se generó al inicio de sesión y se actualiza en el valor auth en el headers o encabezado.

Resultado esperado: mensaje “Not authorized”, HTTP status 401.

Resultado alterno: inesperado acceso de parte del usuario no autorizado a los datos.

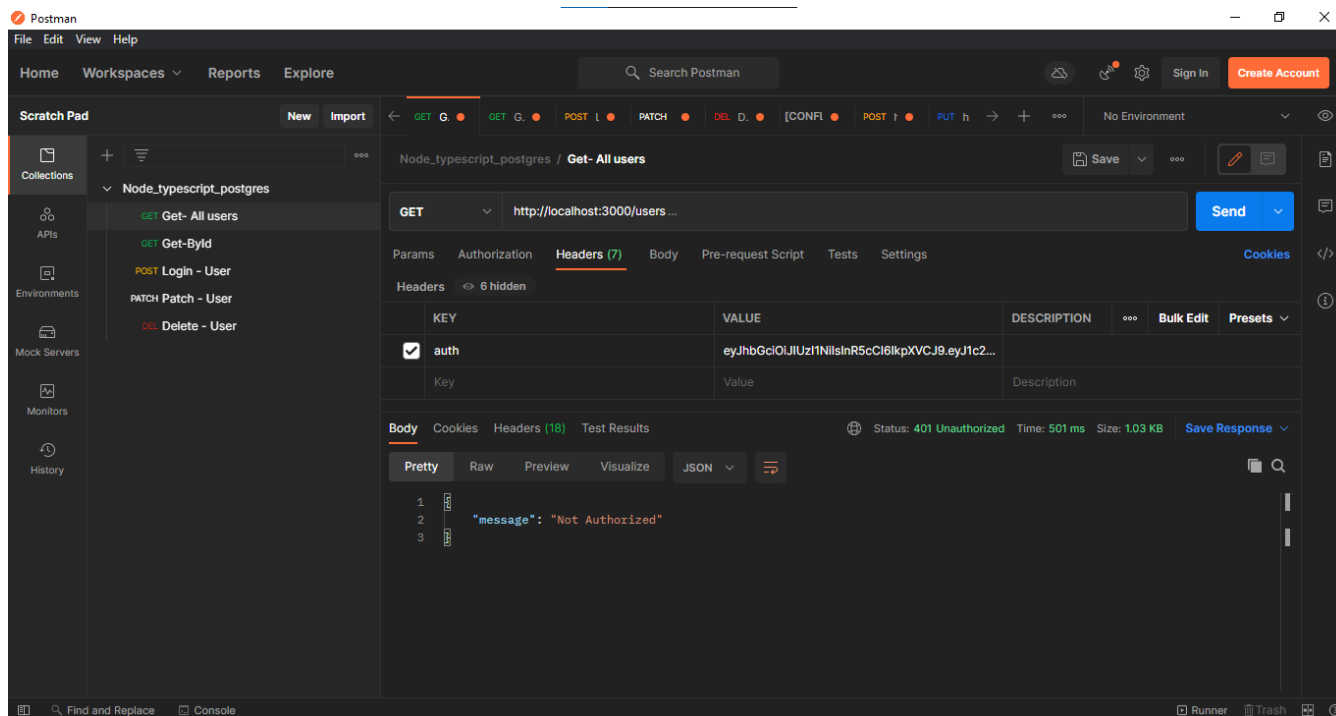


Figura 60. Validación en Postman "Permisos de usuarios"

PAQUETE HISTORIA CLÍNICA / INFORMACIÓN DE PACIENTES

- ✓ RF-6: El sistema permitirá el registro de la información del paciente, una vez guardado el estudiante no podrá modificar o eliminar.
- ✓ RF-7: El sistema permitirá ingresar información del paciente como: Datos paciente, estilo de vida, medio ambiente, día típico, antecedentes y observación general. (Información establecida por el docente).
 - Crear Historia clínica <http://localhost3000/histocli>

Se hace una petición POST, indicando el usuario que hace la petición, y la ruta declarada para la función. Este método requiere del token para permitir crear historias clínicas en la plataforma, este rol es el de estudiantes.

Resultado esperado: historia clínica creada, HTTP status 200. Mensaje “history clinic created”

Resultado alterno: No hay conexión con la base de datos, historia clínica no creada, falta un dato, entre otros.

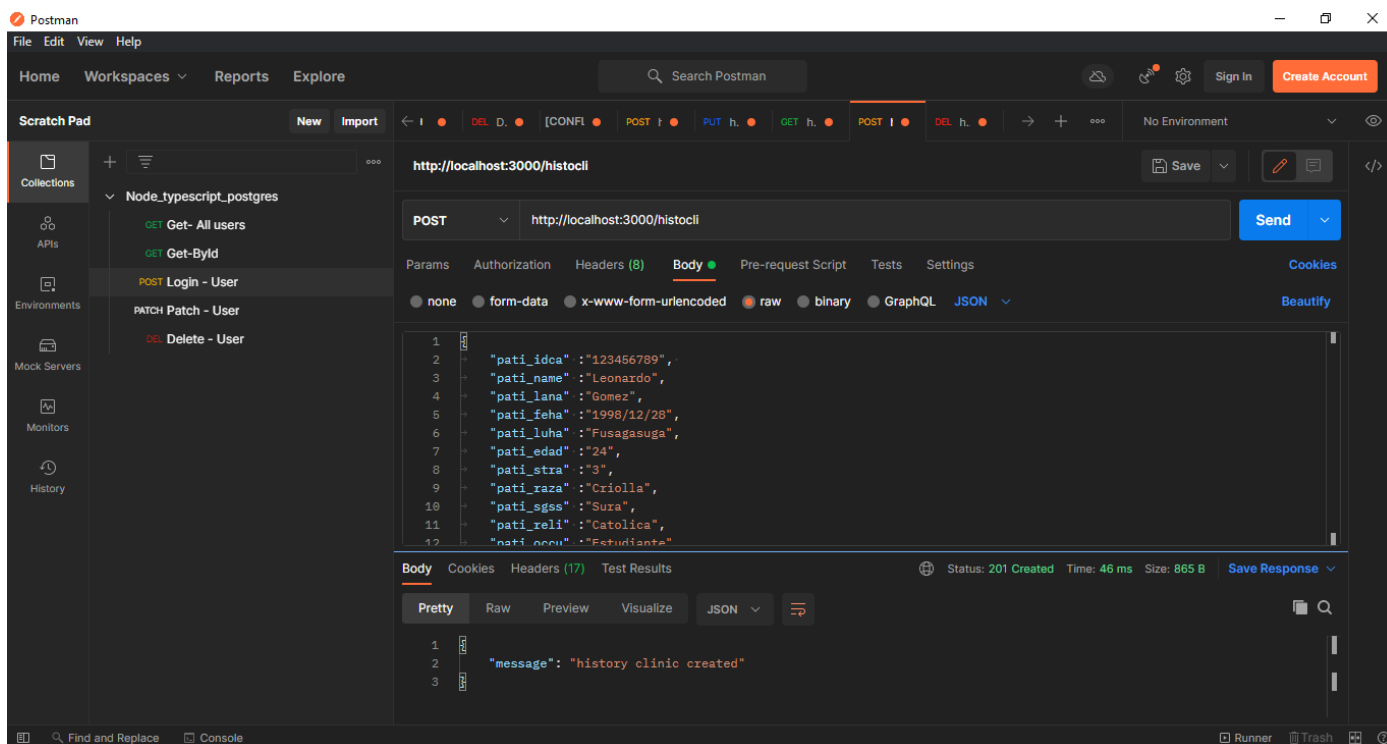


Figura 61. Validación en Postman "Crear Historia Clínica"

- ✓ RF-8: El sistema solo mostrará la historia clínica almacenada al rol Admin/docente, también será el único que la puede borrar

- Borrar Historia clínica <http://localhost3000/histocli/IDdelpaciente> Ejemplo
<http://localhost3000/histocli/123456789>

Se hace una petición DELETE, indicando El usuario que hace la petición, y la ruta declarada para la función. Este método requiere del token para permitir eliminar historias clínicas en la plataforma, este rol es el de docente.

Resultado esperado: historia clínica borrada, HTTP status 200. Mensaje “history clinic deleted”

Resultado alterno: No hay conexión con la base de datos, historia clínica no creada, falta un dato, entre otros.

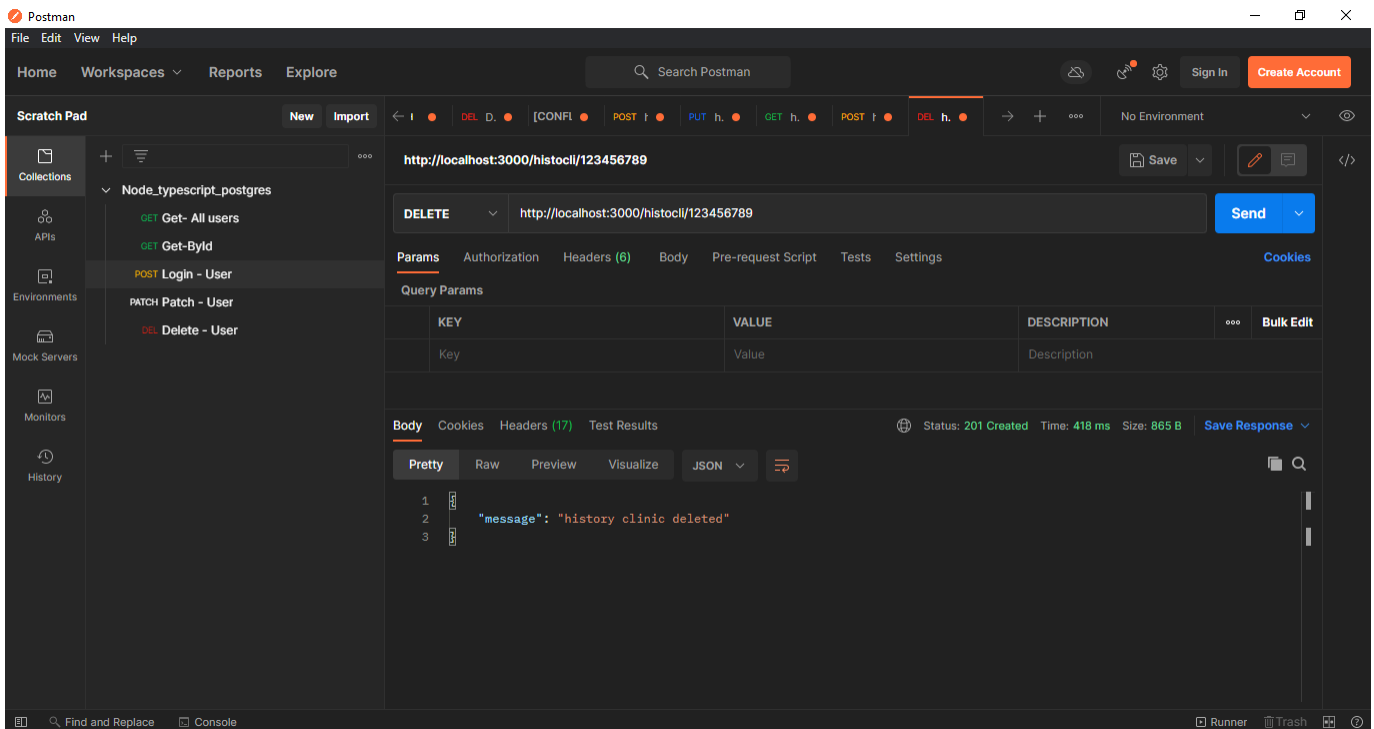


Figura 62. Validación en Postman "Borrar Historia Clínica"

7.6.2. Pruebas: Método De Encuesta

Una encuesta es un método empírico que se utiliza para recopilar información de o sobre personas para describir, comparar o explicar su conocimiento, sus actitudes o su comportamiento. También se pueden utilizar encuestas para la descripción de las características de métodos o herramientas. (Mesa López-Colmenar et al., 2007).

7.6.3. Hipótesis

A través de la integración de conocimientos entre Enfermería e Ingeniería de Sistemas, es posible crear un prototipo web para el registro de historias clínicas, que le permita a los estudiantes de Enfermería obtener experiencia usando la tecnología.

Hipótesis nula:

A través de la integración de conocimientos entre Enfermería e Ingeniería de Sistemas, no es posible crear un prototipo web para el registro de historias clínicas, que le permita a los estudiantes de Enfermería obtener experiencia usando la tecnología.

Variable dependiente: Prototipo web

Variable Independiente: experiencia, conocimientos.

7.6.4. Dimensión e indicador

Tabla 12. Tabla dimensión e indicador de variable

Variable	Dimensión	Indicador	Ítems
Prototipo	Usabilidad	Claridad de uso	¿La interfaz del prototipo es agradable a la vista? - Si - No
		Tratamiento de color	
		Tratamiento de texto	
		Ambiente gráfico	
		Accesibilidad	
		Documentación de usabilidad (manual de usuario)	¿La interfaz es intuitiva? - Si - No
			En caso de no entender el sistema, o cómo

		<p>utilizarlo, ¿Cuenta con un manual o instructivo para entender el funcionamiento?</p> <p>- Si - No</p> <p>¿La interfaz del aplicativo hace referencia a la Enfermería?</p> <p>- Si - No</p> <p>¿La interfaz del aplicativo cuenta con texto claro y entendible?</p> <p>- Si - No</p> <p>Para usar el aplicativo, ¿Tuvo una accesibilidad sencilla?</p> <p>- Si - No</p> <p>¿El sistema en el tiempo de uso presentó algún fallo?</p> <p>- Si - No</p> <p>Como usuario, se encuentra de acuerdo con las funcionalidades y propiedades del aplicativo</p> <p>- Si - No, ¿Por qué?</p>
	<p>Funcional</p> <p>Fluidez de interfaz</p> <p>Adaptabilidad de dispositivos</p> <p>Accesibilidad remota</p> <p>Seguridad</p>	<p>¿La aplicación tiene las funcionalidades y propiedades correspondientes para el registro de historias clínicas?</p>

		<p>Interoperabilidad Eficiencia</p>	<p>- Si - No</p> <p>¿Logró acceder al sistema y realizar funcionalidades sin tener que ingresar usuario o contraseña? - Si - No</p> <p>¿Puede mantener el nivel de rendimiento, bajo ciertas condiciones y por cierto tiempo? - Si - No</p> <p>¿Después de un periodo de tiempo, por seguridad su sesión fue cerrada? - Si - No</p> <p>¿En su rol como estudiante puede registrar, modificar o eliminar algún otro usuario? - Si - No</p> <p>¿Tuvo un rendimiento efectivo? - Si - No</p>
	Portabilidad	Adaptabilidad Fácil instalación	<p>¿El sistema tiene portabilidad? Es decir, tiene factibilidad de transferir de un ambiente a otro? - Si - No</p> <p>¿Logró usar el sistema al tiempo que otros usuarios? - Si</p>

			<p>- No</p> <p>Al ejecutar la aplicación desde otro dispositivo, sea computador, móvil, tableta, entre otros. ¿Se vio afectada de forma negativa la interfaz?</p> <p>- Si</p> <p>- No</p>
--	--	--	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Para implementar la encuesta se realizó en la herramienta de Forms de Microsoft Office con cuentas institucionales, se llevó a cabo una reunión citada por los estudiantes desarrolladores del prototipo.

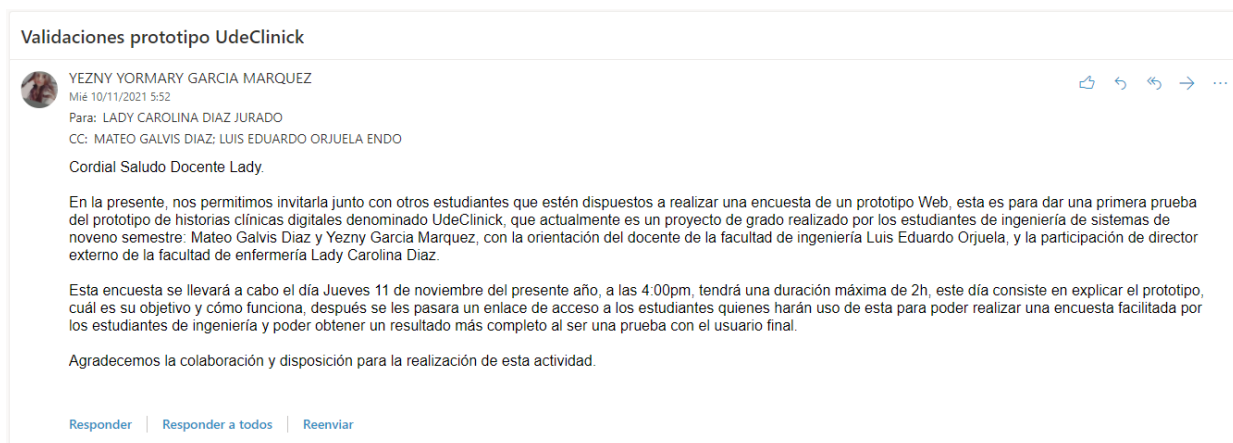


Figura 63. Correo electrónico institucional

Se utiliza el correo institucional para enviar la invitación formal para asistir a la reunión.

La reunión fue realizada por medio de la plataforma Teams, evidenciada por medio de una grabación.

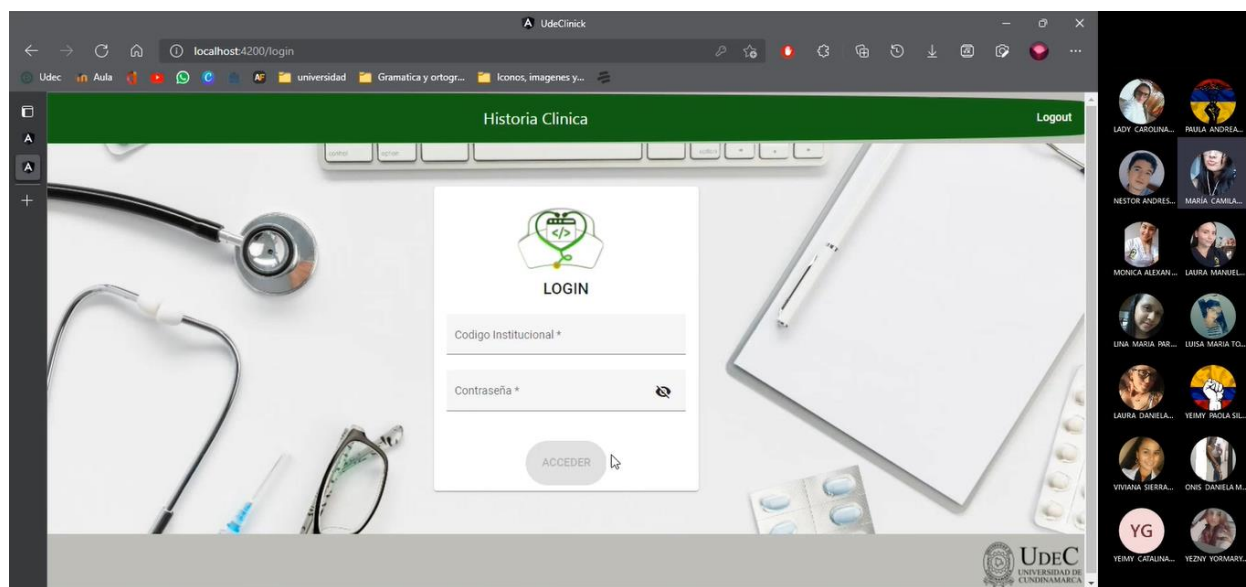


Figura 64. Evidencia de aplicación de encuesta y reunión con estudiantes de Enfermería

7.6.5. Encuesta Aplicada

Se tomó una muestra de 14 estudiantes de Enfermería, quienes observaron y evidenciaron el funcionamiento del proyecto. La encuesta arrojó los siguientes resultados:

2. Ítem Usabilidad del prototipo

[Más detalles](#)

■ Si ■ No

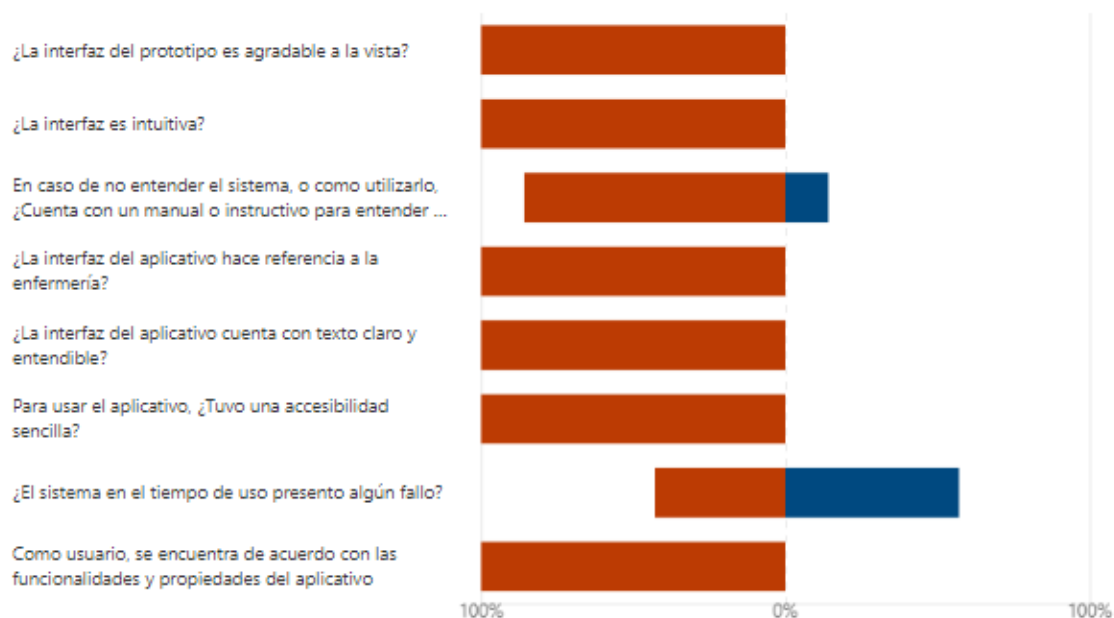


Figura 65. Resultados encuesta ítem usabilidad

Fuente 42. Captura de pantalla resultados de encuesta realizada en microsoft forms

Se hicieron 8 preguntas que hacían referencia a la usabilidad del prototipo, en este caso a la función de que tiene el sistema referente a la interfaz, dando como conclusión que la mayoría de las respuestas son asertivas, a excepción de la pregunta referente a un fallo del sistema, al momento de realizar la encuesta se evidenció un fallo del servidor que se empleó para la utilización de prototipo.

3. ¿El logo del prototipo le parece un logo aceptable?

[Más detalles](#)



Figura 67. Resultados encuesta

Fuente 44. Captura de pantalla resultados de encuesta realizada en Microsoft forms

El logo creado por los estudiantes de ingeniera fue aceptado por un 100% por parte de los estudiantes. Dando a entender, que si representa el sentido de la página.

4. ¿Qué probabilidades hay de que recomiende el prototipo para usos prácticos?

[Más detalles](#)

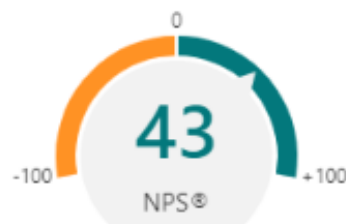


Figura 66. Resultados encuesta

Fuente 43 Captura de pantalla resultados de encuesta realizada en Microsoft forms

Se les preguntó a los estudiantes si recomendarían el prototipo, lo cual en una escala de 1 a 10, 8 muy probablemente lo promocionarían, 4 probablemente no harían nada, y 2 personas probablemente no.

5. Ítem: Portabilidad del prototipo

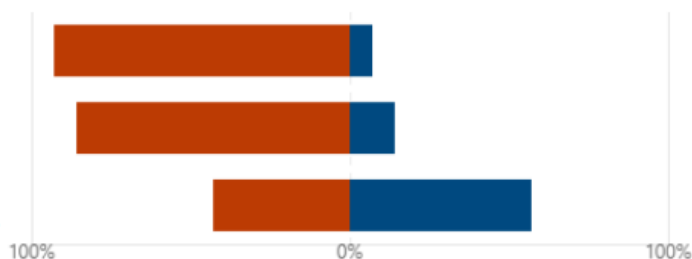
[Más detalles](#)

■ Si ■ No

¿El sistema tiene portabilidad? Es decir, tiene factibilidad de transferir de un ambiente a otro?

¿Logro usar el sistema al tiempo que otros usuarios?

Al ejecutar la aplicación desde otro dispositivo, diferente a un computador. ¿Se vio afectada de form...



Fuente 45 Captura de pantalla resultados de encuesta realizada en Microsoft forms

Figura 68. Resultados encuesta ítem de portabilidad

Para el ítem de portabilidad, el 100% calificó de que el sistema tiene una factibilidad de transferir el sistema de un ambiente a otro, lo cual hace referencia a utilizarlo en diferentes entornos. Otra pregunta que hacía referencia si se lograba usar al tiempo que otros usuarios el 85.7% respondió que sí podían utilizarlo al tiempo y el otro 14.3% respondió que no. Y la última pregunta para este ítem fue si la interfaz del prototipo se ve afecta al usarse en un dispositivo diferente a un ordenador, el 42.9% respondió que sí, y el 51.7% respondió que no se ve afectada.

6. Califique si su experiencia con el prototipo fue agradable y cumple con sus objetivos.

[Más detalles](#)

14

Respuestas



Clasificación media 4.57

Figura 69. Resultados encuesta

Fuente 46. Captura de pantalla resultados de encuesta realizada en Microsoft forms

La experiencia general con el uso del prototipo fue de 4.57 siendo una calificación media de 1 a 5. Lo cual nos da el resultado de una experiencia favorable.

7. Ítem: Funcionalidad del prototipo

[Más detalles](#)



Figura 70. Resultados encuesta ítem funcionalidad

Fuente 47. Captura de pantalla resultados de encuesta realizada en Microsoft forms

Para el ítem de funcionalidad, la primera pregunta hace referencia si la aplicación tiene propiedades y funcionalidades que corresponden al registro de historias clínicas, lo cual el 100% de estudiantes respondieron que sí. La siguiente pregunta hace énfasis al rendimiento del prototipo, para esto los usuarios hicieron uso en diferentes dispositivos, dando una respuesta favorable del 100%, entendiéndose que el aplicativo es funcional en diferentes condiciones. Otra pregunta hace referencia al periodo de tiempo que maneja el sistema para cerrar o terminar una sesión por inactividad a lo que el 84.6% respondió que sí, mientras el 15.4% respondió que no. Por último, la pregunta de rendimiento estuvo a favor del 100% que respondieron que si tenía un rendimiento efectivo.

7.6.6. Conclusiones Encuesta

Al crear diseño de investigación y el aporte de conocimiento de las diferentes personas expertas en el tema, y desde el ámbito de investigación con la formulación de hipótesis y la dimensión de la variable, considerando diferentes ítems de importancia para la gestión de los

diferentes procesos que se requieren, y que posterior a esto se pudo generar una recolección de datos, en los que pudimos evidenciar la falta de un sistema que ayude a los estudiantes con la gestión práctica de registro de historias clínicas. Con esta encuesta podemos evidenciar que el proyecto es viable y posteriormente brindaría un eficiente beneficio para los estudiantes. Se puede decir que el sistema cumple con las funcionalidades correspondientes a un prototipo, es decir, presenta un primer acercamiento a un producto real o un producto final. También, evidenciamos el interés de algunos estudiantes por obtener una herramienta que les ayude en sus prácticas. Algunos comentarios al realizar la encuesta fueron los siguientes:

“ Si la verdad, es muy buena idea el prototipo, ojalá se siga mejorando la idea para que beneficie a todos. En muchos sentidos es una herramienta que nos facilita registrar los datos de los pacientes de forma más fácil”

“Gracias a ustedes, no saben lo que nos ayudan para las prácticas clínicas.”

“Este es un producto que nos acerca a la realidad hospitalaria.”

Estudiantes de Enfermería de Universidad De Cundinamarca.

8. Conclusiones Y Recomendaciones

Este proyecto es un aporte de parte de estudiantes de Ingeniería de Sistemas, que buscan mediante su preparación dar una solución a una problemática, ya que su campo de estudio es interdisciplinario y permite conocer, analizar cualquier área, entender su realidad y como optimizar o solucionar el proceso que se maneje en esta para así implementar sistemas complejos.

UdeClinick es una oportunidad para la Universidad de implementar nuevas tecnologías, que se pueden emplear como herramientas para los estudiantes de Enfermería, quienes tienen poco acercamiento con esta área. Este trabajo interdisciplinario permitió crear una solución que beneficia a diferentes actores que hacen parte de la Universidad, como estudiantes y docentes; nuestro objetivo personal y profesional es dar solución y facilitar actividades por medio de tecnologías e innovar en la integración de herramientas de desarrollo, como lo fue la utilización de la herramienta ORM que nos permite no solo utilizar un gestor de base de datos, sino que brinda la oportunidad de integrar más tecnologías para el manejo de estas; También la utilización de librerías y dependencias de seguridad informática, que en el área de Enfermería es la protección de datos de un elemento tan importante como es la historia clínica real de un paciente, dejándoles una simulación de una plataforma Web segura, también la gestión de un registro clínico, y de cómo posiblemente sea su futuro laboral. A su vez, para nosotros como estudiantes y desarrolladores entender la debida gestión de los datos que se le debe dar a un sistema.

Con base a lo anterior, se cumple con los objetivos planteados. El producto debía ser un prototipo funcional, capaz de cumplir con los requerimientos propuestos en la recolección de información, donde los estudiantes y la docente Lady Carolina Diaz Jurado, nos mencionaron que desearían que el software, más que recibir una información clínica, simule la seriedad de un

sistema de historias clínicas, en donde se realizó un estado del arte y se comprobó que estos sistemas enfocan la seguridad de los datos del paciente. Gracias a la utilización de un marco de trabajo que se ajusta a los posibles cambios, logramos enfocar nuestro sistema, como un software seguro, con control de roles, encriptación de contraseñas, caducidad de sesión por usuario, entre otros beneficios, así, cumpliendo con el segundo objetivo de diseño y desarrollo de módulos respectivos a partir de la información suministrada, que permita la gestión de historias clínicas digitales. Al momento de desarrollar, como se explicó anteriormente en el documento se validó el comportamiento del sistema mediante pruebas realizadas en Postman, cumpliendo el tercer objetivo. Y para finalizar, se desplegó la plataforma Web en un hosting que ofrece servicio las veinticuatro horas para que los estudiantes implementen el prototipo, objetivo que también se validó y se evidenció haciendo una encuesta a los estudiantes de Enfermería.

Se debe entender que nuestro aplicativo es una primera aproximación prototipada de un sistema más complejo, es decir, el sistema se creó en una primera versión para dar espacio a generar un sistema que abarque completamente las necesidades del usuario final, se desea que en un futuro este proyecto se utilice y se avance en su desarrollo y poder agilizar procesos de Enfermería para que los estudiantes tengan una herramienta de apoyo en sus prácticas académicas.

Por esta razón, el prototipo cumple con los requerimientos solicitados y con unas limitaciones anteriormente mencionadas. Debido a esto se hizo una aproximación de un formulario de historia clínica, solo con los datos principales del paciente y dos registros anexos que corresponden a la misma, dados por la docente Lady Carolina Diaz Jurado. Cabe resaltar que mediante el proceso de despliegue hubo la necesidad de hacer la adquisición de un servidor de sistema operativo Linux, su distribución utilizada fue Ubuntu 20.4, para realizar el despliegue del BackEnd, y para realizar el despliegue FrontEnd se utilizó la plataforma de Google Firebase. La utilización de despliegue en dos servidores diferentes generó errores al momento de hacer peticiones, debido a que el servidor adquirido en Linux no contaba con una certificación SSL, en ese orden de ideas recurrimos en adquirir un certificado y un dominio para que no se presentara el error en la plataforma.

9. Referencias

Angular. (2021). Introducción a los conceptos de Angular.

<https://docs.angular.lat/guide/architecture>.

Azure Boards. Docs.microsoft.com. (2021). Artefactos de plantilla de proceso de scrum.

<https://docs.microsoft.com/es-es/azure/devops/boards/work-items/guidance/scrum-process?view=azure-devops>.

BBVA API_Market. (2016, 23 de marzo). API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos. <https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/>

CASES SANCHIS, M.; PONS TORRES, A. Registro de la actividad profesional enfermera en la historia clínica digital: experiencia en Menorca. *Metas de Enfermería*, 2014, vol. 17, no 8, p. 62-68.

Cases, Sanchi., Torres., A. Registro de la actividad profesional enfermera en la historia clínica digital: experiencia en Menorca. *Metas de Enfermería*, 2014, 17(8), p. 62-68.

<https://www.picuida.es/registro-de-la-actividad-profesional-enfermera-en-la-historia-clinica-digital-experiencia-en-menorca/>

Chá, Ghiglia., Mercedes, M. (2019). Historia clínica electrónica: herramienta para la continuidad de asistencia. *Revista Médica del Uruguay*, 35(3), 107-123.

<https://dx.doi.org/10.29193/rmu.35.3.6>

Clouding. (2021). VPS Cloud: Tu Servidor Virtual en España - ¡Pruébanos! <https://clouding.io/#>.

- Cocina, E. G., & Torres, F. P. (2007). La historia clínica electrónica. Revisión y análisis de la actualidad. *Diraya: la historia de salud electrónica de Andalucía. Revista Española de Cardiología Suplementos*, 7(3), 37C-46C.
- Consejo Profesional Nacional de Ingeniería. (2021). Código de ética | Copnia. Copnia.gov.co. <https://www.copnia.gov.co/tribunal-de-etica/codigo-de-etica>.
- Crispín, L., Doria, M., Rivera, A., Garza, M. et al(2011). Aprendizaje autónomo: orientaciones para la docencia. Universidad Iberoamericana. http://biblioteca.clacso.edu.ar/Mexico/dcsyp-uia/20170517031227/pdf_671.pdf.
- Genero Bocco, M., Piattini Velthuis, M. G., Cruz-Lemus, J. A.(2015). Métodos de investigación en Ingeniería del software. Ediciones de la U. <https://www-ebooks7-24-com.ucundinamarca.basesdedatosezproxy.com/?il=7734>
- Google Developers. (2021). Firebase. <https://firebase.google.com/>
- Jiménez Fernández, J. C., & Cerrillo Martín, D. (2010). Registros de Enfermería: un espejo del trabajo asistencial. *Metas de Enfermería*, 13(6), 8-11.
- Microsoft (2021). Introduction to GitHub in Visual Studio Code - Learn. <https://docs.microsoft.com/en-us/learn/modules/introduction-to-github-visual-studio-code/>
- Microsoft. GitHub. [Windows]. <http://git-scm.com/download/win>:
- Microsoft. VisualStudioCode. (Versión 1.61) [Windows]. <https://code.visualstudio.com/>
- Ministerio de Educación. (1980, 15 de diciembre). Normas en materia de Ética Médica. articles-103905_archivo_pdf.pdf (mineduacion.gov.co)

MKLLabs Co.,Ltd. (2021). STARUML [Windows]. <https://staruml.io/>

Npm. (2021). Express-api-cli. <https://www.npmjs.com/package/express-api-cli>.

Postman Enterprise. (2021). Postman (Version 8.12) [Windows]. <https://www.postman.com/>

Programador Clic. (2021,8 de noviembre). Conozca JWT (JSON WEB TOKEN).

<https://programmerclick.com/article/5839871491/>.

Ramoz, M., Alfaro, M., Sánchez, M. (2009). Archivo de historias clínicas Digitalizado, una solución previa a la Historia Clínica Electrónica. Papeles médicos, vol.(18). núm. 2.

<https://pdf4pro.com/cdn/archivo-de-historias-cl-237-nicas-digitalizado-una-soluci-243-n-46c7f2.pdf> .

Rodríguez, P. & Rodríguez, Luis. (1999). Principios técnicos para realizar la anamnesis en el paciente adulto. Revista Cubana de Medicina General Integral, 15(4), 409-414.

http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-21251999000400011&lng=es&tlng=es .

Rubinos., Nuevo, A. (2011) Seguridad en bases de datos. Revista Cubana de Ciencias Informática, 5(1). <http://www.redalyc.org/articulo.oa?id=378343671005>

Solarte, R. (2015). Historia clínica electrónica desde un dispositivo móvil. Scientia et Technica Vol. 20(04).370-373. <https://doi.org/10.22517/23447214.9255>

Tessa, R. (2019). Dificultades percibidas por estudiantes de la licenciatura de enfermería, en su inserción a la práctica clínica de paciente crítico. Uruguay. Salud Areandina, 2(2).

<https://doi.org/10.33132/23229659.1284>

TypeORM .(s.f). TypeORM. <https://typeorm.io/#/>

Villa, A., Poblete, M., (2007). Aprendizaje basado en competencias. Una propuesta para la evaluación de las competencias genéricas. Bilbao: Editorial Mensajero, Colección Estudios e Investigación del ICE. Universidad de Deusto, 336 pp.