

ARQUITECTURA DE SOFTWARE EN N-CAPAS. UN ENFOQUE PRÁCTICO

Hesmeralda Rojas Enríquez

Ronald Alberto Renteria Ayquipa

Virgilio Martínez Duran

Walter Jhon Huayllani Aguirre

Flor Cagniy Cárdenas Mariño

Hugo David Calderón Vilca

ISBN 978-628-96479-2-1

EDITORIAL
CICI

CENTRO DE INVESTIGACIONES
Y CAPACITACIONES
INTERDISCIPLINARES

Autores

Hesmeralda Rojas Enriquez

*Doctor en Ciencias de la
Computación*

Universidad Nacional Micaela

Bastidas de Apurímac

hrojas@unamba.edu.pe

 <https://orcid.org/0000-0002-9388-2991>

Ronald Alberto Renteria Ayquipa

*Doctor en Ciencias de la
Computación*

Universidad Nacional Micaela

Bastidas de Apurímac

rrenteria@unamba.edu.pe

 <https://orcid.org/0000-0001-7736-162X>

Virgilio Martínez Duran

Magister en Gestión Pública

Poder Judicial - Corte

Superior de Justicia de

Apurímac

vmartinez@unamba.edu.pe

 <https://orcid.org/0000-0001-5471-2423>

Walter Jhon Huayllani Aguirre

*Bach. Ing. Informática y de
Sistemas*

walterjhon7@gmail.com

 <https://orcid.org/0009-0005-5069-5917>

Flor Cagniy Cárdenas

Mariño

Doctor en Ciencias de la

Computación

Universidad Nacional Mayor

de San Marcos

fcardenasm@unmsm.edu.pe

 <https://orcid.org/0000-0002-1154-6266>

Hugo David Calderón Vilca

Magíster en Ingeniería de Software

Universidad Nacional Mayor de San

Marcos.

Trabajo actual: Profesor

hcalderonv@unmsm.edu.pe

 <https://orcid.org/0000-0002-1177-4947>

CICI | **CENTRO DE INVESTIGACIONES Y CAPACITACIONES INTERDISCIPLINARES**

Alain Fitzgerald Castro Alfaro: Editor
2024 –Arquitectura de software en n-capas. Un enfoque práctico
Primer Volumen

Autor: Hesmeralda Rojas Enríquez, Ronald Alberto Renteria Ayquipa, Virgilio Martínez Duran, Walter Jhon Huayllani Aguirre, Flor Cagniy Cárdenas Mariño, Hugo David Calderón Vilca.

ISBN Versión Digital: 978-628-96479-2-1

Editorial Centro de Investigaciones y Capacitaciones Interdisciplinarias SAS – CICI

Coordinadora: Nora González Pérez –Cartagena –Colombia
Portada y diagramación: Dean Castro Alfaro
Prohibida la reproducción total o parcial por cualquier medio sin la autorización escrita del titular de los derechos patrimoniales

Cartagena –Colombia, Agosto de 2024

ARQUITECTURA DE SOFTWARE EN N- CAPAS. UN ENFOQUE PRÁCTICO

Hesmeralda Rojas Enríquez
Ronald Alberto Renteria Ayquipa
Virgilio Martínez Duran
Walter Jhon Huayllani Aguirre
Flor Cagniy Cárdenas Mariño
Hugo David Calderón Vilca

Colombia
Latinoamérica

Tabla de contenido

1. ARQUITECTURA DE SOFTWARE	11
1.1. Introducción	11
1.2. Definición	11
1.3. Arquitectura de n capas	13
2. CONTEXTO DE DESARROLLO	17
2.1. Analisis del modelo de negocio	17
2.2. Gestión de farmacia.....	18
2.3. Control y gestión de disponibilidad de medicamentos	18
3. FASE DE ANALISIS	21
3.1. Revisión de sprint	21
3.2. Modelo conceptual granular para definición de requisitos.	22
3.3. Captura de requisitos bajo el modelo conceptual granular para definición de requisitos ...	24
3.4. Resultados de la aplicación del modelo conceptual granular para definición de requisitos	27
3.5. Resultados de la aplicación del modelo conceptual granular para definición de requisitos	28
3.6. Historias de usuarios	29
3.7. Tarjeta crc (clase – responsabilidad - colaboracion).....	39
3.8. Diagrama de flujo historias de usuario	47
4. DESARROLLO DE LA ARQUITECTURA N-TIER.....	54
4.1. Arquitectura de Sistema desarrollado.....	54
4.2. Abstracción de las capas del modelo propuesto y definición de tareas	55
4.3. Herramientas tecnológicas	59
4.4. Personas y roles del proyecto	59
4.5. Creación de capas.....	60
4.6. Utilización de las capas.....	61
4.7. Implementación de las capas en la solución de Visual Studio	62
5. EVALUACIÓN DE USABILIDAD	72
5.1. Usabilidad	72
5.2. Evaluación de la usabilidad del sistema.....	73
6. CONCLUSIONES Y REFERENCIAS BIBLIOGRAFICAS	78
6.1. Conclusiones	78
6.2. Referencias bibliográficas	78

RESUMEN

El desarrollo de software es una disciplina compleja, pues el software al tener que ser construido, requiere la interacción de múltiples factores técnicos, humanos y contextuales. El software inicia con la definición de su arquitectura, entendida como los planos sobre los cuales se sustentará todo el proyecto, esta estructura deberá soportar los objetivos del proyecto y asegurar que el sistema sea escalable, mantenible, seguro y eficiente. Esta decisión es crucial, ya que puede maximizar las posibilidades de éxito y minimizar los riesgos y costos asociados con problemas técnicos y operativos a largo plazo.

Dentro del equipo desarrollador, es el Arquitecto de Software, sobre quien recae la responsabilidad de definir la arquitectura sobre la cual se soportará todo el proyecto de software. Además de establecer otros aspectos técnicos como las tecnologías a utilizar, revisar los requisitos y especificaciones del modelo de negocio, establecer las buenas prácticas en el desarrollo; el Arquitecto de software deberá contextualizar las necesidades de la organización y proponer una arquitectura con los patrones de diseño que se adapten a las necesidades de la entidad.

En esta línea, este libro presenta el desarrollo de la Arquitectura conocida como "n-tier", en una variante a la más utilizada, la arquitectura de 3 capas; presentando cinco capas para el desarrollo que son: capa de presentación, capa de negocio, capa de repositorio, capa de acceso a datos y capa de aplicación.

Para definir la arquitectura, se presenta el contexto del modelo de negocio para el cual se planteó la solución y los procesos a ejecutar; seguidamente, se presenta la fase de análisis donde se recogieron, examinaron y documentaron los requisitos del sistema desarrollado para el área de Farmacia del Hospital Regional Guillermo Díaz de la Vega de Apurímac.

Seguidamente, se presenta el desarrollo de la arquitectura n-tier en cinco capas, que comprende la abstracción de las capas del modelo propuesto, la creación, utilización e implementación de las capas.

Finalmente, se presenta la evaluación aplicada al sistema de información desarrollado bajo la perspectiva de los usuarios, midiendo el atributo de usabilidad.

Palabras claves: n-tier, arquitectura de software, desarrollo de software, granularidad de requisitos, software hospital.

ABSTRACT

Software development is a complex discipline because software, having to be built, requires the interaction of multiple technical, human, and contextual factors. Software development begins with the definition of its architecture, understood as the blueprint upon which the entire project will be based. This structure must support the project's objectives and ensure that the system is scalable, maintainable, secure, and efficient. This decision is crucial as it can maximize the chances of success and minimize the risks and costs associated with long-term technical and operational problems.

Within the development team, the responsibility of defining the architecture upon which the entire software project will be supported falls on the Software Architect. In addition to establishing other technical aspects such as the technologies to be used, reviewing the requirements and specifications of the business model, and setting best practices in development, the Software Architect must contextualize the organization's needs and propose an architecture with design patterns that adapt to the entity's needs.

In this context, this book presents the development of the architecture known as "n-tier" in a variant of the more commonly used three-tier architecture, introducing five layers for development: presentation layer, business layer, repository layer, data access layer, and application layer.

To define the architecture, the context of the business model for which the solution was proposed and the processes to be executed are presented. Subsequently, the analysis phase is presented where the requirements of the system developed for the Pharmacy department of the Guillermo Díaz de la Vega Regional Hospital in Apurímac were collected, examined, and documented.

Following this, the development of the n-tier architecture in five layers is presented, which includes the abstraction of the layers of the proposed model, the creation, utilization, and implementation of the layers.

Finally, the evaluation applied to the information system developed from the users' perspective is presented, measuring the usability attribute.

Keywords: n-tier, software architecture, software development, requirement granularity, hospital software.

Capítulo I

ARQUITECTURA DE SOFTWARE

1. ARQUITECTURA DE SOFTWARE

1.1. Introducción

El desarrollo del software es una actividad compleja y relativamente reciente en comparación con otras disciplinas profesionales. Su conocimiento se ha generado en un período muy corto (Palacios 2011), y en este ritmo se ve enfrentado a los avances acelerados de la industria de la tecnología, el ritmo exponencial de la complejidad de los sistemas en los que se integra, la alta demanda de software en tiempos cada vez, más reducidos y con un proceso de ciclo de vida que garantice la calidad del producto.

Frente a ello, se han ido creando y construyendo una serie de enfoques y prácticas que buscan estandarizar los conocimientos en ingeniería de software. Tal es el caso del SWEBOK: Software Engineering Body of Knowledge, que comprende el cuerpo de conocimientos fundamentales y buenas prácticas generalmente aceptados (IEEE 2024).

Además, de esta guía, existen una combinación de procesos, prácticas y estándares diseñados para asegurar que el producto final cumpla con los requisitos y expectativas del cliente, sea confiable, funcione correctamente y sea fácil de mantener y actualizar. Todos ellos pueden expresarse como la arquitectura de software.

La arquitectura de software es el origen conceptual de cualquier sistema, en el que recae la organización de su estructura lógica, erigiéndose como una de las áreas de investigación y área práctica más importantes en la ingeniería de software. El logro correcto de una arquitectura para aplicaciones complejas a gran escala, determinará la continuidad y sostenibilidad de cualquier sistema (Ali Babar 2014). La arquitectura de software puede llegar a ser más determinante que los algoritmos y estructuras de datos. No obstante, su implementación se presenta como un enorme desafío.

1.2. Definición

La arquitectura de software ha sido definida a través del tiempo de forma evolutiva, respondiendo a los requerimientos de cada periodo. (Stafford y Wolf 2001) consideraban que la arquitectura de software es el diseño y especificación de la estructura general del

sistema, incluida la organización, los protocolos, la asignación de funciones y el rendimiento.

El estándar IEEE 1471 (IEEE 2007), describe la arquitectura de un "sistema intensivo en software" o también conocido como arquitectura de software, que es "la organización fundamental de un sistema incorporada en sus componentes, sus relaciones entre sí y con el medio ambiente, y los principios que guían su diseño y evolución".

(Cárdenas 2013) indica que es la descripción de los subsistemas y componentes de un sistema de software y las relaciones entre ellos. Subsistemas y componentes se especifican normalmente en diferentes vistas para mostrar las propiedades funcionales y no funcionales relevantes de un sistema de software. Se observa que esta definición incluye una separación de funciones, en elementos nombrados como: vistas.

Una de las definiciones más aceptadas, es la brindada por (Bass, Clemments y Katzman 2013), que se refieren genéricamente a la división de un sistema de software en un conjunto de componentes, sus relaciones visibles externamente y los patrones que definen la colaboración entre ellos para lograr una funcionalidad determinada. Sobre este concepto se puede indicar que permite ampliar el alcance de los sistemas a más de una estructura, ya que los componentes pueden ser bloques de construcción abstractos como una unidad de cálculo, un almacén de datos, protocolos de comunicación, sincronización y acceso a datos; como también, asignación de funcionalidad a elementos de diseño y distribución.

(Martínez et al. 2014) proponen que la arquitectura de software, es una forma de diseño, que ocurre en las primeras etapas de la creación de un sistema en un nivel de abstracción muy alto y su cuestión clave es la organización. Esta definición coincide en entenderla como un diseño sobre el cual se asentará el software, ya que es una representación a alto nivel.

(Jaiswal 2021) afirma que la arquitectura es un modelo o plano para un sistema que permite la comunicación y coordinación entre sus elementos. Esta resolución estructurada debe satisfacer todas las necesidades técnicas y operativas, optimizando al mismo tiempo los atributos de calidad comunes como el rendimiento y la seguridad.

Al respecto de las definiciones de arquitectura de software presentadas, se puede convenir que todas ellas se refieren a la estructura fundamental y el diseño de un sistema de software, que incluye componentes, módulos, interfaces, relaciones y principios que guían su construcción y funcionamiento. Todos estos atributos permitirán gestionar la complejidad inherente a cada modelo de negocio, facilitar la reutilización, mejorar la escalabilidad y flexibilidad, permitir el mantenimiento y la evolución continua, optimizar la eficiencia y el rendimiento de los sistemas informáticos y cómo se garantizará la seguridad y la integridad de la información.

1.3. Arquitectura de n capas

Una de las arquitecturas de software ampliamente desarrolladas es la arquitectura N-capas (N-Tier), considerada como una solución integral para diseñar, crear, desarrollar y mantener aplicaciones a gran escala (Pahlavan Tafti et al. 2011). Cada capa se compone de diversos elementos físicos y lógicos, permitiendo crear aplicaciones flexibles y reutilizables debido a la división en niveles, donde los cambios pueden realizarse en una capa específica, en lugar de tener que reescribir la aplicación completa.

El patrón de arquitectura en capas no especifica el número y los tipos de capas que deben existir, lo que permite una separación lógica en función de cada tipo de proyecto, los requisitos y la complejidad de la aplicación. A la actualidad, el diseño que más se utiliza es el diseño en tres capas tradicionales, funcionalmente separadas (Abdelrahman, Zhan y Chong 2020):

- **Capa de presentación.** Esta se ejecuta del lado del cliente y constituye la interfaz de usuario de la aplicación. Todas las interacciones persona-computador (HCI) ocurren en esta capa. Esta capa se comunica directamente con la capa de aplicación.
- **Capa de aplicación.** También conocida como capa de lógica de aplicación, controladores y gestión de procesos, traduciendo las solicitudes del cliente en consultas de bases de datos para luego devolvérselas. Este nivel intermedio contiene la mayor parte de la lógica de la aplicación. También sirve como interfaz para el nivel siguiente, la capa de datos. Por lo general, la lógica de negocios se

puede proporcionar a través de un middleware utilizando servicios web o mediante consultas de datos estándar como SQL, ODBC o JDBC (Frith 2015).

- **Capa de datos.** También conocida como almacenamiento de datos, Tiene como responsabilidad principal llevar a cabo operaciones de lectura y escritura en la base de datos, lo cual abarca la recuperación de información, el almacenamiento de datos y la gestión de transacciones. Es una capa que no tiene acceso hacia la capa de presentación, ni a ningún tráfico no mediado por la lógica empresarial.

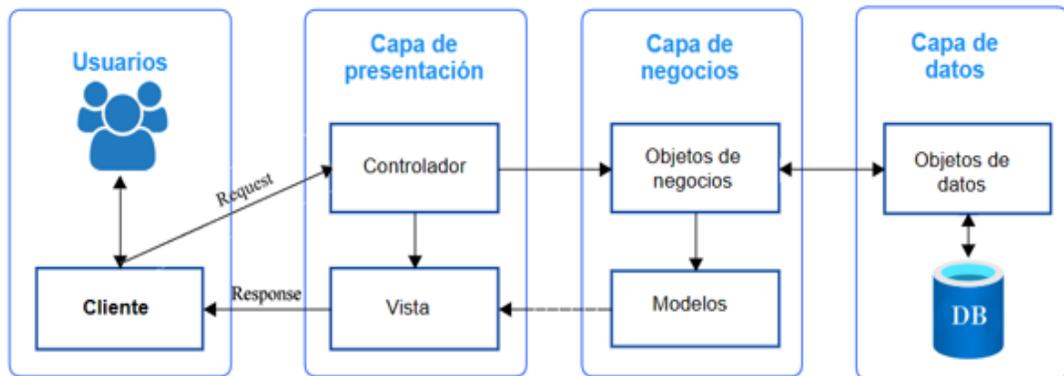


Figura 1. Arquitectura de 3 capas

En la Figura 1, se observa el diseño de la arquitectura en tres capas más utilizado. La capa de datos, es la que se encuentra en el nivel más profundo y solo puede ser accesada por la capa de negocios. La capa de negocios contiene la lógica del negocio, que son las reglas, procesos y operaciones que rigen el funcionamiento de una organización o empresa en particular para cumplir con sus objetivos y estrategias. Esta capa consume de la capa de datos, y alimenta a la capa superior que es la capa de presentación. La capa de presentación está diseñada para que los usuarios puedan interactuar de manera efectiva y comprensible con el software a través de diversos dispositivos, como computadoras, teléfonos móviles, tablets, entre otros. Esta capa es responsable de desarrollar una interfaz gráfica de usuario (GUI) que permita a los usuarios interactuar con el software de manera intuitiva y eficiente, brindar una experiencia positiva y satisfactoria al usar el software, realizar validaciones de

datos antes de enviar los datos al servidor, entre otros. El usuario únicamente se comunica con el sistema por medio de la capa de presentación.

La separación y organización se ve reflejada en cada capa lógica para facilitar el desarrollo, la escalabilidad y el mantenimiento. Cada capa se ocupa de una función específica y se comunica con su capa adyacente. También asegura que se pueda trabajar de forma paralela cada una de ellas, debido a su desacoplamiento, es posible avanzar una capa mientras se va implementando otra. Entre sus principales beneficios, se encuentra la mantenibilidad derivada de la separación de funciones, la reutilización de módulos y componentes independientes pueden volverse a usar en otros proyectos o en otras partes de la misma aplicación, la escalabilidad de componentes y un mejor manejo y comprensión del sistema en general.

Capítulo II

CONTEXTO DE DESARROLLO

2. CONTEXTO DE DESARROLLO

2.1. Analisis del modelo de negocio

El trabajo de investigación se realizó en el Hospital Regional Guillermo Díaz de la Vega (HRGDV), inaugurado en el año 1964 y es el primer hospital de la región Apurímac – Perú.

El sistema de información solicitado para el área de Farmacia del HRGDV surgió por la necesidad de automatizar y centralizar las tareas, puesto que se contaba con un sistema antiguo que no respondía a las necesidades actuales y además afrontar problemas significativos en el área, como son:

- Algunos medicamentos se agotan y no siempre se previene a tiempo su reabastecimiento, lo cual perjudica al paciente al tener que esperar hasta que sea repuesto en farmacia.
- Se debe garantizar que el stock físico corresponda al stock registrado en el sistema.
- El sistema permite controlar lotes y fechas de vencimiento.
- Se debe garantizar la existencia de stock de medicamentos estratégicos, estupefacientes psicotrópicos y precursores para el tratamiento del dolor moderado a severo para pacientes que lo necesitan; estos son distribuidos de forma gratuita por el Ministerio de Salud con el objetivo de limitar su uso a fines médicos y científicos y asegurar su disponibilidad para estos propósitos.
- Durante el proceso de atención en los servicios del HRGDV, un paciente puede haber sido atendido por consultorio externo, por emergencia y/o por hospitalización el mismo día. Se requiere que exista una diferenciación del consumo por cada servicio, para responder a auditorías y tener una visión clara del consumo diferenciado por cada atención.
- Sobre el proceso de generación de Kardex, es necesario asegurar que el sistema soporte cambios en las recetas que se afectan directamente a los Kardex de consumo y Kardex de enfermería. Estos cambios surgen por las modificaciones de recetas de una terapia debido a la evolución del paciente, o por otros motivos; el personal de farmacia

realiza los descargos de las recetas después de la visita médica y/o cuando el paciente se acerca a la ventanilla, para que el personal de farmacia haga correcciones a una receta ya procesada. Frente a esto, se debe anular la transacción de salida que conlleva a un “backtracking” en los registros del Kardex y el recálculo de stock para el producto que se desea modificar.

- Respecto a la generación de reportes, es necesario garantizar flexibilidad en los mismos para tener varias vistas con diferentes perspectivas y lograr una visión integral del proceso de farmacia.

2.2. Gestión de farmacia

La gestión de farmacia busca brindar una atención farmacéutica adecuada, garantizar una farmacoterapia óptima en términos de seguridad y eficacia, para mejorar la calidad del proceso asistencial del paciente. Esta gestión o funcionamiento integral de la farmacia está basado en dos enfoques: en el de la satisfacción al usuario y en el basado en procesos que son el conjunto de actuaciones, decisiones, actividades y tareas que se enmarcan de forma secuencial y ordenada para conseguir un resultado que satisfaga plenamente las necesidades del paciente (Páez et al. 2016) .

El objetivo general de una gestión de farmacia consiste en garantizar el suministro continuo y oportuno de los materiales y medios de producción requeridos para asegurar los servicios de forma ininterrumpida y rítmica.

2.3. Control y gestión de disponibilidad de medicamentos

El control y gestión de los medicamentos en una farmacia, se traduce en la disponibilidad de medicamentos, que es la condición de un medicamento de encontrarse listo para utilizarse en la cantidad necesaria (≥ 2 MED – Meses de Existencia Disponibles) y en buen estado de uso, en razón de atender las necesidades terapéuticas de las personas en un momento determinado (DIGEMID 2014).

La disponibilidad de medicamentos debe medirse a través de indicadores que representen su estado actual, tal como sigue:

— **Disponibilidad de Medicamentos en Normostock**

Condición de un medicamento donde su stock disponible puede cubrir una demanda de consumo entre 2 hasta 6 Meses de Existencia Disponible - MED (disponibilidad ≥ 2 y ≤ 6) (DIGEMID 2014).

— **Disponibilidad de Medicamentos en Sobrestock**

Condición de un medicamento donde su stock disponible puede cubrir una demanda de consumo mayor de 6 Meses de Existencia Disponible - MED, existiendo sobre stock y riesgo de vencimiento (disponibilidad > 6) (DIGEMID 2014).

— **Disponibilidad de Medicamentos en Substock**

Condición de un medicamento donde su stock disponible no puede cubrir una demanda de consumo de 2 Meses de Existencia Disponible - MED, pero es mayor a 0 Meses de Existencia Disponible - MED. (Disponibilidad > 0 y < 2)

— **Disponibilidad de Medicamentos en Desabastecido**

Condición de un medicamento donde stock para atender su demanda de consumo (disponibilidad=0) (DIGEMID 2014).

— **Disponibilidad de Medicamentos Sin rotación**

Condición de un medicamento donde su stock disponible puede cubrir una demanda de consumo mayor de 6 Meses de Existencia Disponible - MED, este se encuentra en sobre stock, sin rotación mayor o igual a 4 meses existiendo el riesgo de vencimiento (disponibilidad > 6 y sin rotación.) (DIGEMID 2014).

Capítulo III

FASE DE ANALISIS

—

3. FASE DE ANALISIS

3.1. Revisión de sprint

Se muestra en esta sección, la lista de sprint también conocida como backlog de sprint, donde se listan las tareas y actividades que se realizaron durante el sprint para lograr los objetivos acordados.

Tabla 1

Sprint 1 del Sistema de Información n-tier

Sistema de información de N-TIER				
Sprint	Inicio	Fin	Duración días	Avance total
1			180	100%
ID	Tarea	Tipo	Estado	Avance
T1	Registro de productos farmacéuticos	Alta	Terminado	100%
T2	Registro de Proveedores	Media	Terminado	100%
T3	Registro de farmacia y almacén	Alta	Terminado	100%
T4	Registro de Nota de ingreso	Alta	Terminado	100%
T5	Registro de Nota de Salida	Alta	Terminado	100%
T6	Registro de requerimiento de productos farmacéuticos	Media	Terminado	100%
T7	Visualizador de Kardex	Media	Terminado	100%
T8	Visualizar el stock de los almacenes	Media	Terminado	100%
T9	Control de fecha de vencimiento	Alta	Terminado	100%
T10	Nivel de disponibilidad de productos farmacéuticos	Alta	Terminado	100%

Tabla 2

Sprint 2 del Sistema de Información n-tier

Sistema de información de N-TIER				
Sprint	Inicio	Fin	Duración días	Avance total
2			150	100%
ID	Tarea	Tipo	Estado	Avance

T1	Registro de atención de pacientes	Alta	Terminado	100%
T2	Registro de atención de paciente	Alta	Terminado	100%
T3	Reporte de consumo de medicamentos de pacientes	Alta	Terminado	100%
T4	Reporte de Ventas	Alta	Terminado	100%

Tabla 3

Sprint 3 del Sistema de Información n-tier

Sistema de información de N-TIER				
Sprint	Inicio	Fin	Duración días	Avance total
3			90	100%
ID	Tarea	Tipo	Estado	Avance
T1	Migración de catálogo DIGEMID	Alta	Terminado	100%
T2	Migración de los productos farmacéuticos	Alta	Terminado	100%
T3	Migración de los documentos	Alta	Terminado	100%
T4	Migración de atención de recetas	Alta	Terminado	100%

3.2. Modelo conceptual granular para definición de requisitos.

En esta fase de desarrollo de software, se buscó estudiar y comprender los procesos del sistema que se va a construir bajo el enfoque de necesidades y expectativas de los usuarios, clientes y otras partes interesadas en relación con el sistema o la aplicación que se va a desarrollar.

Se trabajó con el Modelo conceptual granular para definición de requisitos funcionales propuesto por (Huayllani y Rojas 2020), bajo un marco de trabajo que permitió dilucidar la secuencia de eventos que forman parte de un requerimiento. El modelo brinda una guía en la definición de requisitos que normalmente no están explicitados, y dependen en gran medida del conocimiento y experiencia del analista de sistemas, por el contrario, con las cinco fases propuestas en el modelo, se hace énfasis en obtener una descripción detallada por parte de los clientes sobre cada una de las tareas y de este modo obtener mayor granularidad en los requisitos.

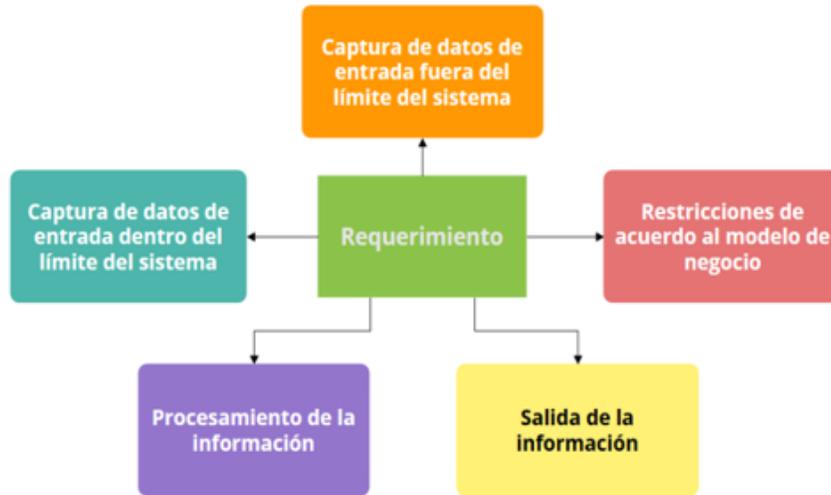


Figura 2 — Modelo conceptual para la definición de requisitos propuesto por (Huayllani y Rojas 2020).

- **Captura de datos de entrada dentro del límite del sistema:** el requisito refleja un proceso o procedimiento dentro de la organización que persiguen un objetivo, está compuesto por una serie de tareas o pasos, a pesar que estas tareas no son el objetivo del requerimiento, son pasos indispensables e intermedios para su ejecución. En esta fase, se debe entender el proceso y su principal objetivo.
- **Captura de datos de entrada fuera del límite del sistema:** en esta fase se debe evaluar un proceso, advirtiendo si las funcionalidades que realiza tienen relación con otros sistemas, si existe alguna información que deba ser consumida de otra aplicación. Esta propuesta ayuda a dilucidar el comportamiento del requerimiento, puesto que no se limita únicamente a la funcionalidad que abarca el proceso, sino que debe prever si existen otros datos a ser tratados fuera de su límite natural.
- **Restricciones de acuerdo al modelo de negocio:** se refiere a definir las reglas propias del negocio. En este caso, se puede hacer un paralelo a la sección Abstracción de las capas del modelo propuesto, Capa de negocios; en esta fase, se deberán definir tanto para los proyectos de Farmacia y Almacén, las restricciones que responden a la forma particular que la organización implanta sus reglas para sus procesos.
- **Procesamiento de la información:** en esta fase el requerimiento ya ha sido dilucidado previamente en su forma básica, ya se ha comprendido cuál es su propósito, si consume

alguna información fuera de los límites del sistema y cuáles son las reglas de negocio a las que está sujeta. En esta fase se procede a realizar los cálculos u operaciones de transformación de información, es decir, se deben realizar las operaciones matemáticas, operaciones sobre strings, listas u otro tipo de objetos, que reflejen el proceso por medio de fórmulas, procedimientos y cálculos.

- **Salida de la información:** finalmente, el requerimiento deberá emitir un resultado. Es interesante observar que, en esta fase del modelo, se solicita esclarecer cómo se mostrará la información, lo cual permite comprender y anticiparse a la forma en que un proceso culmine. Así, un requerimiento tiene una visión más global de su papel.

3.3. Captura de requisitos bajo el modelo conceptual granular para definición de requisitos

En la fase de análisis, se aplicó el modelo conceptual granular para la definición de requisitos. Guiados por el modelo, se presentan dos ejemplos de los procesos de la Farmacia:

Ejemplo 1: Realizar la atención de una receta médica

Tabla 4

Caso de uso de alto nivel: Realizar la atención de una receta médica

<p>Caso de Uso: Realizar la atención de una receta médica</p> <p>Actores: Paciente, operador de Farmacia</p> <p>Tipo: Primario</p> <p>Descripción: Un paciente presenta ante el operador de Farmacia, la receta médica física que le fue entregada en uno de los consultorios de salud, se verifica el stock de la lista de insumos médicos prescritos y se procede a registrarlos. El paciente obtiene los insumos médicos.</p>
--

Tabla 5

Requerimiento funcional: Realizar la atención de una receta médica

Tarea	Funcionalidades a realizar
-------	----------------------------

Captura de datos de entrada dentro del límite del sistema	Buscar al paciente ingresando su número de historia clínica o apellidos.
	Buscar al paciente ingresando sus apellidos o las primeras letras de su apellido y el sistema va filtrando y mostrando las coincidencias.
	La caja de texto para buscar al paciente por apellidos informa si el usuario ha introducido otros caracteres diferentes a las letras.
	La caja de texto para buscar al paciente por número de historia clínica informa si el usuario ha introducido otros caracteres diferentes a los números.
	Listar el tipo de atención y seleccionar la recibida por el paciente.
	Buscar el insumo médico ingresando su descripción o las primeras letras de su descripción y el sistema va filtrando y mostrando las coincidencias.
Restricciones de acuerdo al modelo de negocio	Se verificará la disponibilidad de stock del insumo médico, si para un insumo médico no existe stock se informará al usuario y no podrá ser seleccionado
	Se verificará que la cantidad solicitada en la receta no supere el stock disponible.
Captura de datos de entrada dentro del límite del sistema	Se mostrará el precio unitario de cada insumo médico.
Procesamiento de la información	Se calculará el monto subtotal por insumo médico
	Se calculará el monto total de la receta médica.
	Se aplicarán los impuestos al total de la receta médica
Captura de datos de entrada dentro del límite del sistema	Se listarán las formas de pago disponibles: proveídas por el estado (enlazadas a servicios estatales) o de forma particular costeadas por el paciente.

Captura de datos de entrada fuera del límite del sistema	Si la forma de pago es por un servicio proveído por el estado se consumirán los datos de la aplicación por medio de un web service.
Procesamiento de la información	Si la forma de pago es particular, costado por el paciente, se registrará el pago.
Salida de la información	El stock de los insumos médicos entregados está actualizado.
	Se informa al operador de Farmacia si el proceso fue completado exitosamente.

Ejemplo 2: Generar facturación electrónica por receta médica

Tabla 6

Caso de uso de alto nivel: Generar facturación electrónica por receta médica

<p>Caso de Uso: Generar facturación electrónica por receta médica</p> <p>Actores: Paciente, operador de Farmacia, Sistema tributario estatal</p> <p>Tipo: Primario</p> <p>Descripción: El operador de Farmacia, inicia la operación de facturación electrónica enviando al servidor HRGDV la solicitud, el servidor usa el certificado digital para firmar el comprobante de pago electrónico, realizando las validaciones solicitadas para emitir el documento electrónico y enviándolo a través de internet al servidor financiero estatal que confirma la operación. El paciente puede consultar en cualquier momento el estado de su documento electrónico.</p>

En la Tabla 6, se detalla el requerimiento funcional de generación de factura electrónica, bajo el modelo propuesto. La definición más ampliada del proceso se encuentra en el artículo de (Ayquipa et al. 2019).

Tabla 7

Requerimiento funcional: Generar facturación electrónica por receta médica.

Tarea	Funcionalidades a realizar
Captura de datos de entrada	Iniciar el proceso de facturación electrónica vinculado a una receta médica al pulsar sobre un control de la aplicación.

dentro del límite del sistema	
Procesamiento de la información	Enviar de detalle del pedido de la receta médica para generación de comprobante de pago electrónico al servidor local
Restricciones de acuerdo al modelo de negocio	Realizar la validación UBL (Universal Business Language). Standard
	Realizar la validación de contenido
	Realizar la validación de consistencia
	Realizar la validación de firma digital
Procesamiento de la información	Generar factura electrónica
	Enviar detalle de pedido de la receta médica con validaciones al servidor de la entidad tributaria estatal.
Captura de datos de entrada fuera del límite del sistema	El servidor estatal tributario emite un documento XML de confirmación.
Salida de la información	Se registra una copia del documento XML de confirmación en el servidor de la institución y en el servidor de la entidad tributaria estatal.

3.4. Resultados de la aplicación del modelo conceptual granular para definición de requisitos

Para evaluar la percepción de completitud funcional de los usuarios con respecto al sistema desarrollado, se formaron dos grupos de usuarios, cada uno compuesto por 2 personas. Dos analistas fueron responsables de capturar los requerimientos; el primer analista empleó el modelo propuesto para plantear sus preguntas y obtener la lista de funciones requeridas en el aspecto funcional, mientras que el segundo analista optó por la técnica de Brainstorming. Se llevaron a cabo dos revisiones para cada requerimiento funcional.

Los datos presentados en la Tabla 8 representan los promedios derivados de un cuestionario administrado a dos grupos de usuarios. Este cuestionario evaluó la percepción de completitud de las tareas según lo descrito en el documento de especificación de requisitos. Cada requerimiento funcional fue revisado dos veces, lo que permitió comparar los subcriterios proporcionados por cada analista. Los usuarios asignaron una calificación del 0 al 10 para determinar si la lista de funciones descrita en el requisito reflejaba correctamente el flujo normal del proceso.

3.5. Resultados de la aplicación del modelo conceptual granular para definición de requisitos

Tabla 8

Resultados de uso del modelo frente a la técnica de Brainstorming para captura de requisitos

Técnica utilizada	Revisión	Media	Desviación standard
RequerimientoFuncional_1_Modelo_Propuesto	1	8.3333	0.8876
RequerimientoFuncional_1_Modelo_Propuesto	2	9.0833	0.7929
RequerimientoFuncional_1_Brainstorming	1	7.0833	0.7929
RequerimientoFuncional_1_Brainstorming	2	8.6666	0.8876
RequerimientoFuncional_2_Modelo_Propuesto	1	8.3333	0.8876
RequerimientoFuncional_2_Modelo_Propuesto	2	9.1666	0.7177
RequerimientoFuncional_2_Brainstorming	1	8.0333	1.1645
RequerimientoFuncional_2_Brainstorming	2	8.5	1

Los resultados presentados en la Los datos presentados en la Tabla 8 representan los promedios derivados de un cuestionario administrado a dos grupos de usuarios. Este cuestionario evaluó la percepción de completitud de las tareas según lo descrito en el documento de especificación de requisitos. Cada requerimiento funcional fue revisado dos veces, lo que permitió comparar los subcriterios proporcionados por cada analista. Los

usuarios asignaron una calificación del 0 al 10 para determinar si la lista de funciones descrita en el requisito reflejaba correctamente el flujo normal del proceso.

Tabla 8, indican que los requerimientos funcionales descritos bajo el modelo propuesto, obtiene una media más alta tanto en la primera, como en la segunda revisión, frente a la descripción de requerimientos hecha bajo la técnica de Brainstorming.

3.6. Historias de usuarios

A continuación, se presentan las historias de usuario

Historia de usuario del proyecto Almacén

Tabla 9

Historia de usuario de registro de productos desde el catálogo DIGEMID

HISTORIA DE USUARIO	
Nombre historia:	Registrar medicamentos e insumos médicos quirúrgicos en el catálogo de productos del departamento de farmacia.
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Para el registro del catálogo de producto de medicamentos e insumos médicos quirúrgicos se debe consultar desde el catálogo de medicamentos e insumos médicos de la DIGEMID para saber si está vigente o no.
Observación	<ul style="list-style-type: none"> • Permita registrar varios productos con la ayuda de búsqueda desde un catálogo general de medicamentos e insumos médicos quirúrgicos de la DIGEMID. • Permitir la modificación del catálogo de producto los medicamento o insumos médicos quirúrgicos del departamento de farmacia.

Tabla 10

Historia de usuario registro de proveedores

HISTORIA DE USUARIO	
Nombre historia:	Registrar de proveedores.
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se registra los proveedores de medicamentos e insumos médicos quirúrgicos al sistema.
Observación	<ul style="list-style-type: none"> • Se permita Registrar todos los datos del proveedor • Se permita actualizar todo el dato del proveedor • Se permita Anular a un proveedor

Tabla 11

Historia de usuario registro de farmacia y almacenes

HISTORIA DE USUARIO	
Nombre historia:	Registrar farmacias y sus respectivos almacenes.
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	A medida de la expansión del departamento de farmacia en el Hospital Regional Guillermo Díaz de la Vega, se permita la creación de nuevas farmacias con sus respectivos almacenes.
Observación	<ul style="list-style-type: none"> • Se permita registrar nuevas farmacias con sus respectivos almacenes. • Se permita actualizar los datos de la farmacia y sus almacenes.

Tabla 12

Registrar Notas de Ingreso

HISTORIA DE USUARIO	
Nombre historia:	Registrar las Nota de Ingreso.
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Para el ingreso de medicamentos e insumos médicos quirúrgicos se realiza mediante los siguientes documentos Factura, Boleta, Guía, y entre otros documentos aceptados por el jefe del almacén especializado.
Observación	<ul style="list-style-type: none"> • Se permita registrar el documento y sus medicamentos e insumos médicos quirúrgicos detallados, así actualizar el stock del almacén. • Se permita actualizar el documento y sus medicamentos e insumos médicos quirúrgicos detallados, así actualizar el stock del almacén. • Se permita la anulación del documento ingresado y sus medicamentos e insumos médicos quirúrgicos que se detalla en el documento, así actualizar el stock del almacén.

Tabla 13

Historia de usuario Nota de salida

HISTORIA DE USUARIO	
Nombre historia:	Registrar las Nota de Salida.
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Para la salida de medicamentos e insumos médicos quirúrgicos del almacén especializado a otros almacenes se realiza mediante los siguientes documentos guías de distribución.
Observación	

	<ul style="list-style-type: none"> • Se permita registrar los detalles del documento y sus medicamentos e insumos médicos quirúrgicos, así actualizar el stock del almacén. • Se permita actualizar los detalles del documento y sus medicamentos e insumos médicos quirúrgicos, así actualizar el stock del almacén. • Se permita la anulación del documento ingresado y sus medicamentos e insumos médicos quirúrgicos que se detalla en el documento, así actualizar el stock del almacén.
--	--

Tabla 14

Historia de usuario registro de requerimiento

HISTORIA DE USUARIO	
Nombre historia:	Registrar los requerimientos de medicamentos e insumos médicos quirúrgicos de las farmacias.
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se registra requerimientos de medicamentos e insumos médicos de las farmacias hacia el almacén especializado
Observación	<ul style="list-style-type: none"> • Se permita registrar los medicamentos e insumos médicos quirúrgicos de las farmacias hacia el almacén especializado. • Se permita modificar el requerimiento de medicamentos e insumos médicos quirúrgicos hacia el almacén especializado. • Se permita la impresión del requerimiento de medicamentos e insumos médicos quirúrgicos de las farmacias solicitante.

Tabla 15

Historia de usuario Kardex

HISTORIA DE USUARIO

Nombre historia:	Control de salidas y entradas de medicamentos e insumos médicos quirúrgicos (KARDEX).
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se realiza el seguimiento de las entradas y salidas de medicamentos e insumos médicos de cada farmacia.
Observación	<ul style="list-style-type: none"> • Se permita visualizar e imprimir las salidas y entradas de medicamentos e insumos médicos quirúrgicos de las farmacias. • Se permita filtrar por fechas las salidas y entradas de medicamentos e insumos médicos quirúrgicos

Tabla 16

Historia de usuario Visualizar stock de almacén

HISTORIA DE USUARIO	
Nombre historia:	Visualizar el stock de los almacenes
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se realizará las consultas, se podrá visualizar el stock general como también detallado por cada almacén (almacén especializado, almacén de farmacia central, de farmacia de emergencia, de unidad de cuidados intensivos, Sala de operaciones, traumatología y farmacotecnia) y también detalles como lote y fecha de vencimiento.
Observación	Se permita visualizar e imprimir el stock general del de medicamentos e insumos del departamento farmacia y el stock por farmacia.

Tabla 17

Historia de usuario Control de fecha de vencimiento

HISTORIA DE USUARIO	
Nombre historia:	Control de fecha de vencimiento
Usuarios:	Técnico en Farmacia y Químico Farmacéutico

Descripción:	Se permita visualizar los medicamentos e insumos médicos próximos a vencer por farmacia y almacén
Observación	Se permita visualizar e imprimir los medicamentos e insumos médicos quirúrgicos por farmacia.

Tabla 18 — Historia de usuario Nivel de disponibilidad

HISTORIA DE USUARIO	
Nombre historia:	Nivel de disponibilidad de medicamentos e insumos médicos quirúrgicos
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se visualiza el nivel de disponibilidad de medicamentos e insumos médicos según manual de indicadores de disponibilidad. Lima. 2014.
Observación	<ul style="list-style-type: none"> Se permita visualizar e imprimir el nivel de disponibilidad de los medicamentos e insumos médicos quirúrgicos por farmacia y el almacén especializado.

a. Historia de usuario de la Farmacia

Tabla 19

Historia de usuario registro de atención de paciente

HISTORIA DE USUARIO	
Nombre historia:	Registro de atención de pacientes PARTICULAR, SIS, SOAT, FOSPOLI, ESSALUD y EXONERACIONES
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se registra a la atención de los pacientes: PARTICULAR, SIS, SOAT, FOSPOLI, ESSALUD y EXONERACIONES al sistema.
Observación	

	<ul style="list-style-type: none"> • Para Registrar las atenciones de los pacientes: PARTICULAR, SIS, SOAT, FOSPOLI, ESSALUD y EXONERACIONES. Se debe consumir desde la base de datos de estadística como atención, historia clínica del paciente. • Permita la búsqueda de los pacientes por DNI, numero de historia clínica, apellidos y nombres del paciente, desde la base de datos de estadística.
--	---

Tabla 20

Historia de usuario Registro de atención de recetas medicas

HISTORIA DE USUARIO	
Nombre historia:	Atención de recetas médicas de pacientes PARTICULAR, SIS, SOAT, FOSPOLI, ESSALUD y EXONERACIONES
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se registra la atención de la receta médica que contiene las cantidades de los medicamentos e insumos médicos quirúrgicos dispensados por el personal a pacientes, categorizando por su forma de pago: PARTICULAR, SIS, SOAT, FOSPOLI, ESSALUD y EXONERACIONES al sistema.
Observación	<ul style="list-style-type: none"> • Permite Registrar las atenciones de las recetas médicas de los pacientes: PARTICULAR, SIS, SOAT, FOSPOLI, ESSALUD y EXONERACIONES. Se debe consumir desde la base de datos de estadística como atención, historia clínica del paciente. • Permita la búsqueda del servicio que está siendo atendido. • Permite la búsqueda de medicamentos e insumos médicos quirúrgicos con su respectivo stock que cuenta la farmacia. • Permita la búsqueda de la receta médica y si se tiene un error, que el sistema permita la modificación de la receta • Permita la búsqueda de la receta y se pueda eliminar la receta médica.

Tabla 21

Historia de usuario Reporte de consumo de paciente

HISTORIA DE USUARIO	
Nombre historia:	Reporte de Consumo de paciente
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se visualizará el reporte de consumo de medicamentos e insumos médicos quirúrgicos de forma diaria o personalizada a pacientes, categorizando por su forma de pago: PARTICULAR, SIS, SOAT, FOSPOLI, ESSALUD y EXONERACIONES al sistema.
Observación	<ul style="list-style-type: none"> • Permite Registrar las atenciones de las recetas médicas de los pacientes: PARTICULAR, SIS, SOAT, FOSPOLI, ESSALUD y EXONERACIONES. Se debe consumir desde la base de datos de estadística como atención, historia clínica del paciente. • Permita la búsqueda del servicio que está siendo atendido. • Permite la búsqueda de medicamentos e insumos médicos quirúrgicos con su respectivo stock que cuenta la farmacia. • Permita generar un reporte eh imprimir.

Tabla 22

Historia de usuario Reporte de ventas

HISTORIA DE USUARIO	
Nombre historia:	Reporte de Ventas
Usuarios:	Técnico en Farmacia y Químico Farmacéutico
Descripción:	Se visualizará el reporte de Ventas de medicamentos e insumos médicos quirúrgicos de forma diaria o personalizada.
Observación	Permite generar el reporte mediante filtros de búsqueda como en intervalos de fechas, meses o personalizadas

b. Historia de usuario de Migración

Tabla 23

Historia de usuario Migración de Catálogo de DIGEMID

HISTORIA DE USUARIO	
Nombre historia:	Migración del catálogo de medicamentos e insumos médicos quirúrgicos de la DIGEMID
Usuarios:	Informático
Descripción:	Se cargará la información del catálogo de medicamentos e insumos médicos quirúrgicos y se migrará al sistema de información en N-TIER.
Observación	

Tabla 24

Historia de usuario Migración del catálogo de productos de farmacia

HISTORIA DE USUARIO	
Nombre historia:	Migración del catálogo de productos de medicamentos e insumos médicos quirúrgicos del departamento de farmacia.
Usuarios:	Informático
Descripción:	Se cargará la información del catálogo de productos de medicamentos e insumos médicos quirúrgicos y se migrará al sistema de información en N-TIER.
Observación	

Tabla 25

Historia de usuario Migración de documentos

HISTORIA DE USUARIO	
Nombre historia:	Migración de los documentos de las notas de ingreso y las notas de salidas del departamento de farmacia.
Usuarios:	Informático
Descripción:	Se cargará la información de las notas de ingreso y las notas de salida y se migrará al sistema de información en N-TIER.
Observación	

Tabla 26

Historia de usuario Migración de documentos de atención de recetas

HISTORIA DE USUARIO	
Nombre historia:	Migración de los documentos ventas del departamento de farmacia.
Usuarios:	Informático
Descripción:	Se cargará la información de las ventas por las distintas y se migrará al sistema de información en N-TIER.
Observación	

3.7. Tarjeta crc (clase – responsabilidad - colaboracion)

La tarjeta CRC es una técnica utilizada en el diseño de software para capturar y organizar información sobre las clases y sus responsabilidades en un sistema. Este enfoque ayuda a visualizar y planificar la estructura y el comportamiento del sistema de manera colaborativa entre los miembros del equipo (Dalton 2018).

Los componentes de la tarjeta CRC:

- Clase (Class): En esta sección se especifica el nombre de la clase que se está diseñando. Una clase en programación orientada a objetos representa un conjunto de objetos con características similares y comportamientos comunes.
- Responsabilidad (Responsibility): Aquí se detallan las responsabilidades o tareas que la clase debe cumplir dentro del sistema. Estas responsabilidades pueden incluir métodos, funciones o comportamientos específicos que la clase debe implementar para cumplir su propósito.
- Colaboración (Collaboration): En esta sección se identifican otras clases o componentes del sistema con los cuales la clase actual debe interactuar o colaborar para cumplir sus responsabilidades. Esto incluye relaciones de asociación, agregación, composición u otras formas de interacción entre clases.

Tarjeta de Clase – Responsabilidad – Colaboración de Almacén

Tabla 27

Tarjeta de CRC Productos

CLASE	Productos
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Registrar un nuevo producto • Listar los medicamentos e insumos médicos quirúrgicos. • Modificar datos del producto. 	<ul style="list-style-type: none"> • Buscador del catálogo de medicamentos e insumos médicos quirúrgicos de la DIGEMID.

<ul style="list-style-type: none"> • Dar de baja a los productos. 	
--	--

Tabla 28

Tarjeta de CRC Proveedores

CLASE	Proveedor
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Registrar un nuevo proveedor • Listar los proveedores • Modificar datos del proveedor. • Dar de baja a los proveedores. • Buscador de proveedor. • Mostrar detalles del proveedor 	<ul style="list-style-type: none"> • Buscador de proveedores por RUC

Tabla 29

Tarjeta de CRC Farmacia

CLASE	Farmacia
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • registrar una nueva farmacia • Listar las farmacias • Modificar datos de las farmacias. • Dar de baja a las farmacias. • Buscador de farmacia. • Mostrar detalles de las farmacias 	<ul style="list-style-type: none"> • Generador de almacén de farmacias • Buscador de farmacia

Tabla 30

Tarjeta de CRC Nota de Ingreso

CLASE	Nota de Ingreso
RESPONSABILIDAD	COLABORACIÓN

<ul style="list-style-type: none"> • Asignar de la lista el almacén origen y destino • Registrar los datos de los documentos de entrada • Buscar el producto • Obtener producto • Validar producto • Calcular stock • Calcular el monto • Generar impresión del documento ingresado 	<ul style="list-style-type: none"> • Catálogo de productos • Lista de almacenes • Actualizar stock
---	---

Tabla 31

Tarjeta de CRC nota de salida

CLASE	NOTA DE SALIDA
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Asignar de la lista el almacén origen y destino • Registrar los datos de los documentos de entrada • Buscar el producto • Obtener producto • Validar producto • Validar el stock disponible • Calcular stock • Calcular el monto • Generar impresión del documento de salida 	<ul style="list-style-type: none"> • Stock productos del almacén de origen • Lista de almacenes • Actualizar stock del almacén de origen y destino.

Tabla 32

Tarjeta de CRC Requerimiento

CLASE	Requerimiento
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Registrar un requerimiento • Consultar producto del almacén especializado • Obtener producto del almacén especializado. • Agregar cantidades del producto. • Listar los productos agregados • Generar e imprimir el requerimiento 	<ul style="list-style-type: none"> • Buscador del producto del almacén especializado • Stock del almacén

Tabla 33

Tarjeta de CRC Kardex

CLASE	Kardex
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Lista de almacenes • Obtener el almacén • Buscar el producto por almacén obtenido • Obtener producto del almacén especializado. • Lista las notas de entrada y de salida por fechas. • Generar e imprimir el requerimiento 	<ul style="list-style-type: none"> • Buscador del producto del almacén • Notas de ingreso • Notas de salida • Filtro de fechas de origen y fin

Tabla 34

Tarjeta de CRC Nivel de Disponibilidad

CLASE	Nivel de Disponibilidad
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Lista de almacenes • Obtener el almacén • Asignar mes • Lista el consumo los medicamentos e insumos médicos del almacén. • Lista las notas de entrada y de salida por fechas. • Generar e imprimir el requerimiento 	<ul style="list-style-type: none"> • Buscador del producto del almacén • Notas de ingreso • Notas de salida • Filtro de fechas de origen y fin

Tarjeta de Clase – Responsabilidad – Colaboración de Farmacia

Tabla 35

Tarjeta de CRC Atención de paciente

CLASE	Atención de paciente
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Se busca el paciente atendido • Obtener la atención del paciente • Se registra la atención del paciente • Se le asigna de la lista el tipo de atención • Asignación de servicio de atención. 	<ul style="list-style-type: none"> • Buscador la atención del paciente desde el la base de datos de estadística

Tabla 36

Tarjeta de CRC Atención de Recetas

CLASE	Atención de receta
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Se busca el paciente atendido 	<ul style="list-style-type: none"> • Consulta stock del almacén

<ul style="list-style-type: none"> • Obtener los datos de paciente y el tipo de atención. • Búsqueda de los productos con stock • Validación del producto • Calcular el monto • Registro de la atención de las recetas médicas. • Modificar la atención de la receta médicas. • Anular la receta medica 	<ul style="list-style-type: none"> • Busca pacientes atendidos • Lista la forma de pago • Busca los servicios de atención
--	--

Tabla 37

Tarjeta de CRC Atención de Recetas

CLASE	Atención de receta
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Se busca el paciente atendido • Obtener los datos de paciente y el tipo de atención. • Búsqueda de los productos con stock • Validación del producto • Calcular el monto • Registro de la atención de las recetas médicas. • Modificar la atención de la receta médicas. • Anular la receta médica 	<ul style="list-style-type: none"> • Consulta stock del almacén • Busca pacientes atendidos • Lista la forma de pago • Busca los servicios de atención

Tabla 38

Tarjeta de CRC Reporte de consumo de paciente

Clase	Reporte de consumo de paciente
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Se busca al paciente atendido • Obtener los datos de paciente y el tipo de atención y la fecha de atención. • Lista las atenciones de recetas médicas atendidas • Calcula el monto total de las atenciones de recetas atendidas del paciente 	<ul style="list-style-type: none"> • Buscar la atención al paciente • Buscar la atención de rectas medicas

Tabla 39

Tarjeta de CRC Reporte de Ventas

CLASE	Reporte de ventas
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Lista las ventas realizadas por día • Listar las todas las ventas por intervalos de fechas. • Listar todas las ventas por mes • Generar un impreso. 	<ul style="list-style-type: none"> • Buscar atención de recetas médicas por la forma de pago

Tarjeta de Clase – Responsabilidad -Colaboración de Migración de datos

Tabla 40

Tarjeta de CRC Catalogo de medicamentos e insumos médicos quirúrgicos DIGEMID

Clase	Migración catálogo de medicamentos e insumos médicos quirúrgicos
Responsabilidad	Colaboración
<ul style="list-style-type: none"> • Obtener la lista de medicamentos e insumos médicos quirúrgicos de la DIGEMID • Migrar los datos de la lista de catálogo de medicamentos e insumos médicos quirúrgicos DIGEMID. 	<ul style="list-style-type: none"> • Buscar el catálogo de medicamentos e insumos médicos quirúrgicos

Tabla 41

Tarjeta de CRC Catálogo de productos del departamento de farmacia

CLASE	Migración de catálogo de productos del departamento de farmacia
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Obtener la lista de los productos • Visualizar la lista de los productos • Migrar los datos de la lista de productos. 	<ul style="list-style-type: none"> • Buscar el catálogo de producto de medicamentos e insumos médicos quirúrgicos del SISMED

Tabla 42

Tarjeta de CRC Migración de las notas de salida y notas de ingreso

CLASE	Migración de notas de salida y notas de ingreso
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Obtener la lista de las notas de salida y notas de ingreso 	<ul style="list-style-type: none"> • Buscar los datos de las notas de ingreso

<ul style="list-style-type: none"> • Visualizar la lista de las notas de salida y notas de ingreso • Migrar los datos de las notas de salida y notas de ingreso. 	<ul style="list-style-type: none"> • Buscar los datos de las notas de salida
--	---

Tabla 43

Tarjeta de CRC Migración de las recetas médicas atendidas

CLASE	Migración de notas de salida y notas de ingreso
RESPONSABILIDAD	COLABORACIÓN
<ul style="list-style-type: none"> • Obtener la lista de las recetas médicas atendidas • Visualizar la lista de las recetas médicas atendidas • Migrar los datos de las recetas médicas atendidas 	<ul style="list-style-type: none"> • Buscar los datos de las recetas atendidas en el SISMED.

3.8. Diagrama de flujo historias de usuario

Diagrama de flujo de Historias de Usuarios de Almacén

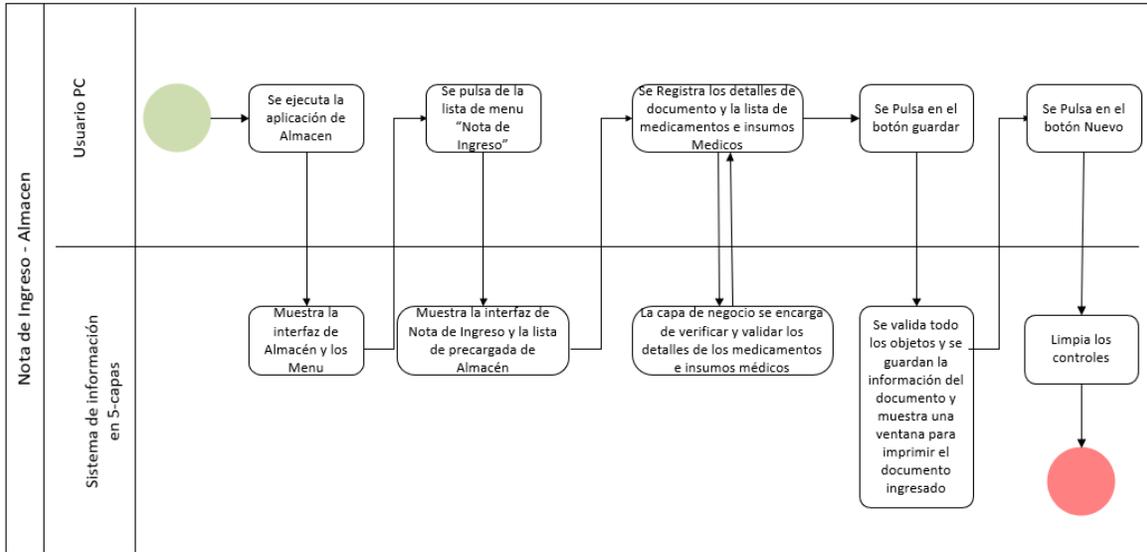


Figura 1

Diagrama de Flujo Nota de ingreso

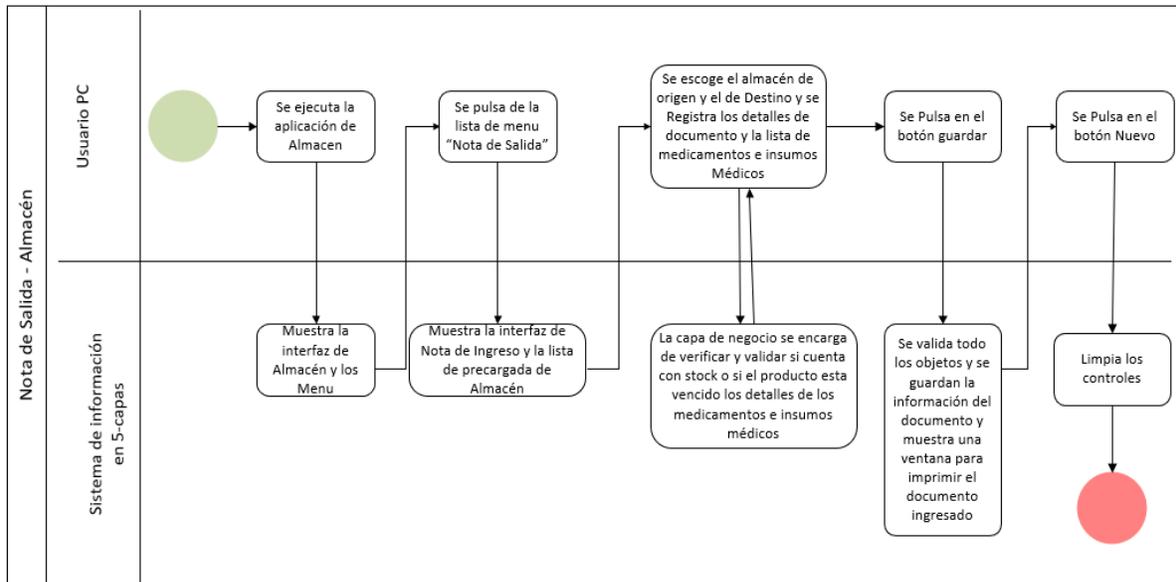


Figura 2

Diagrama de flujo de Nota de salida

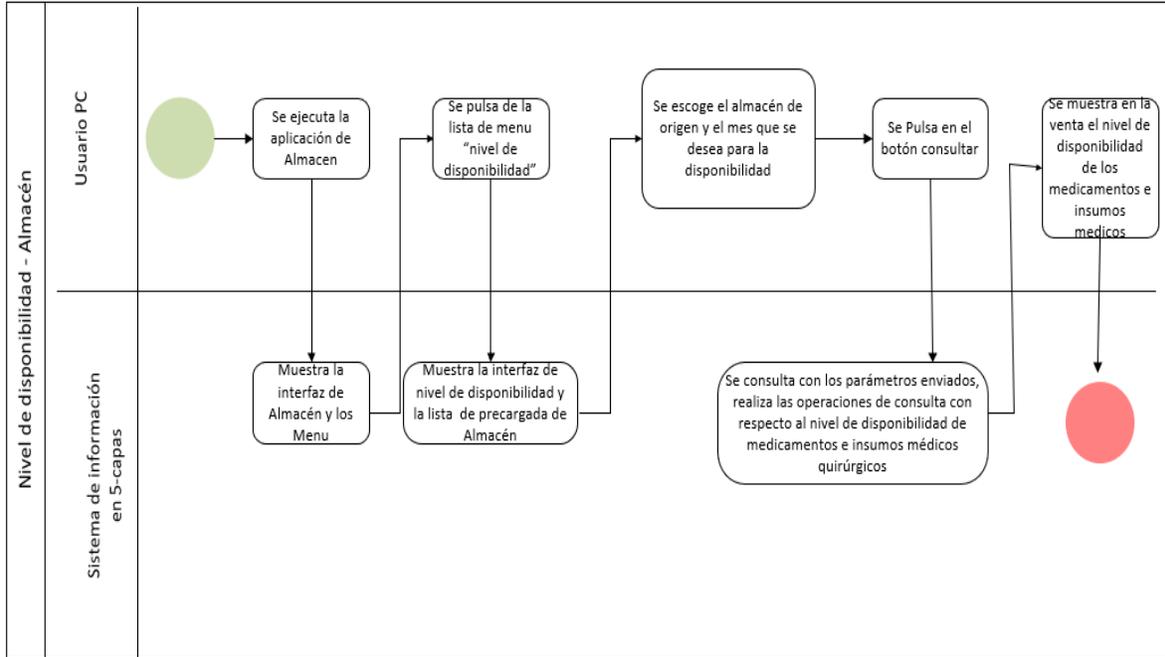


Figura 3

Diagrama de flujo Nivel de disponibilidad

Diagrama de flujo de Historias de Usuarios de Farmacia

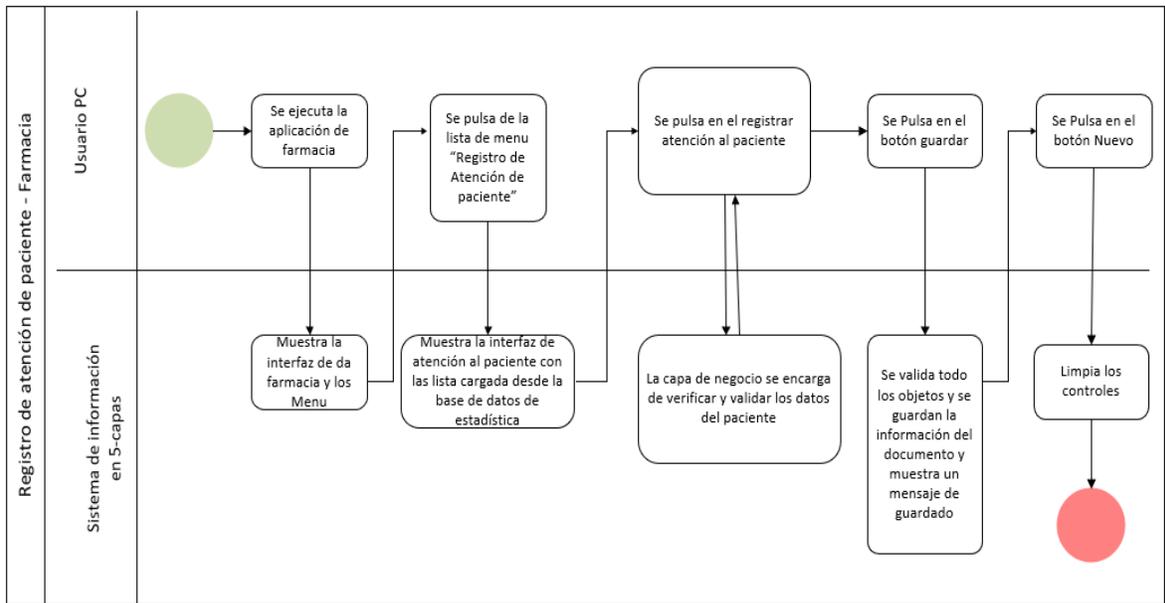


Figura 4

Diagrama de flujo Atención del paciente

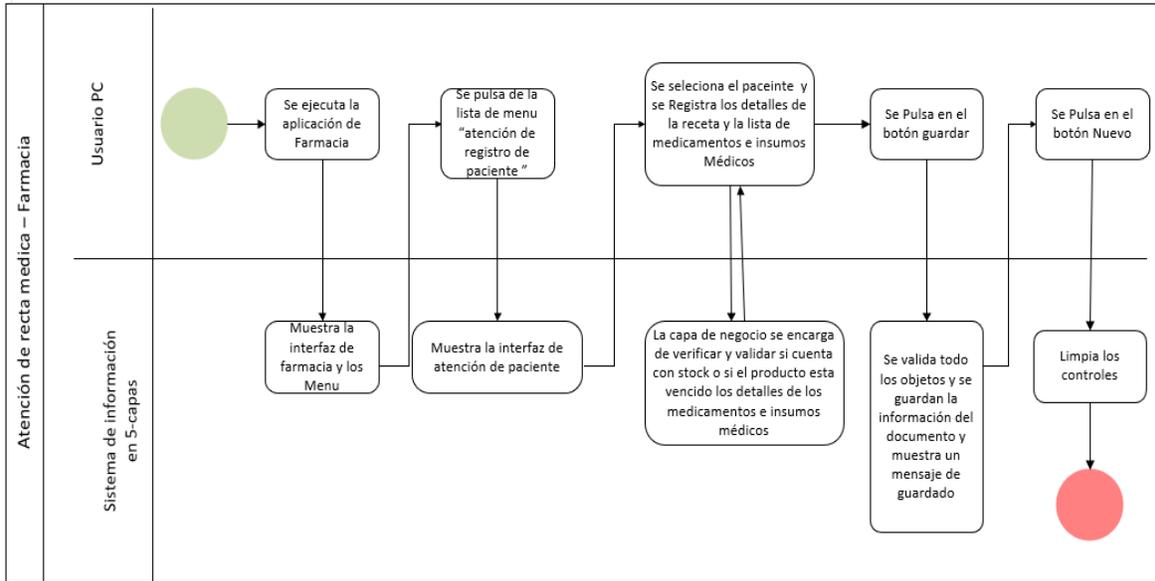


Figura 5

Diagrama de flujo Atención de receta médica

Diagrama de flujo de Historias de Usuarios de Migración

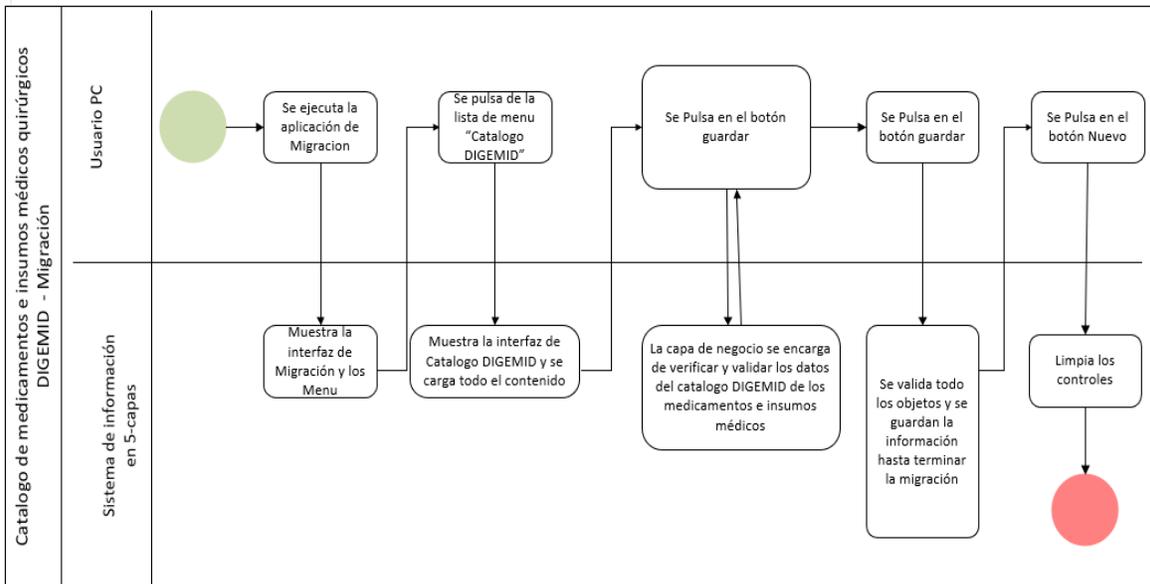


Figura 6

Diagrama de flujo Migración de Catálogo DIGEMID

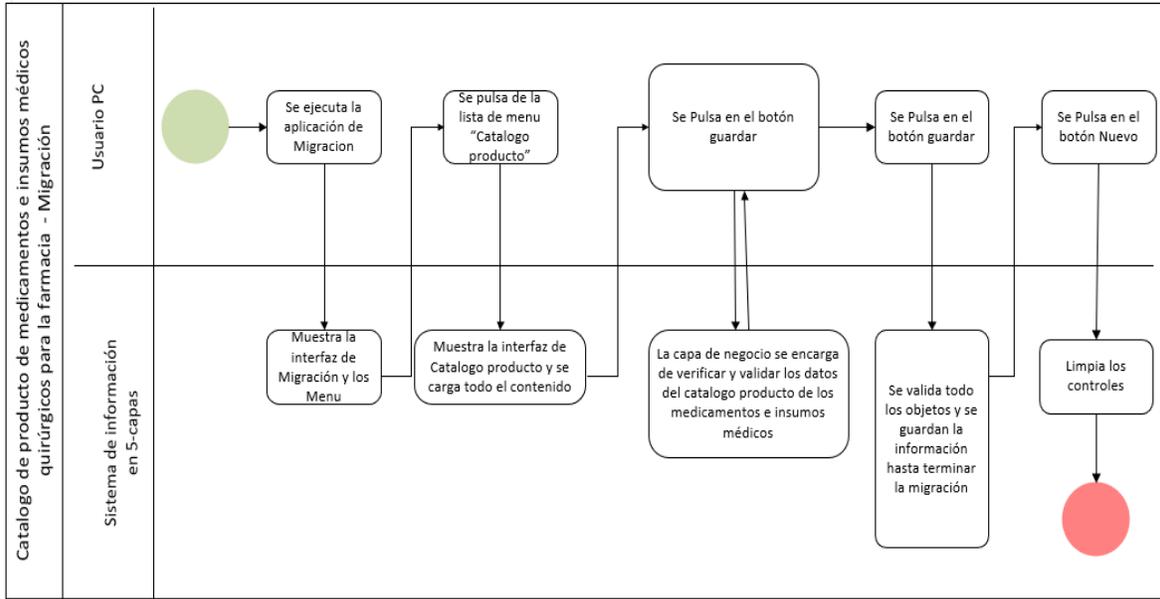


Figura 7

Diagrama de flujo Migración de productos farmacia

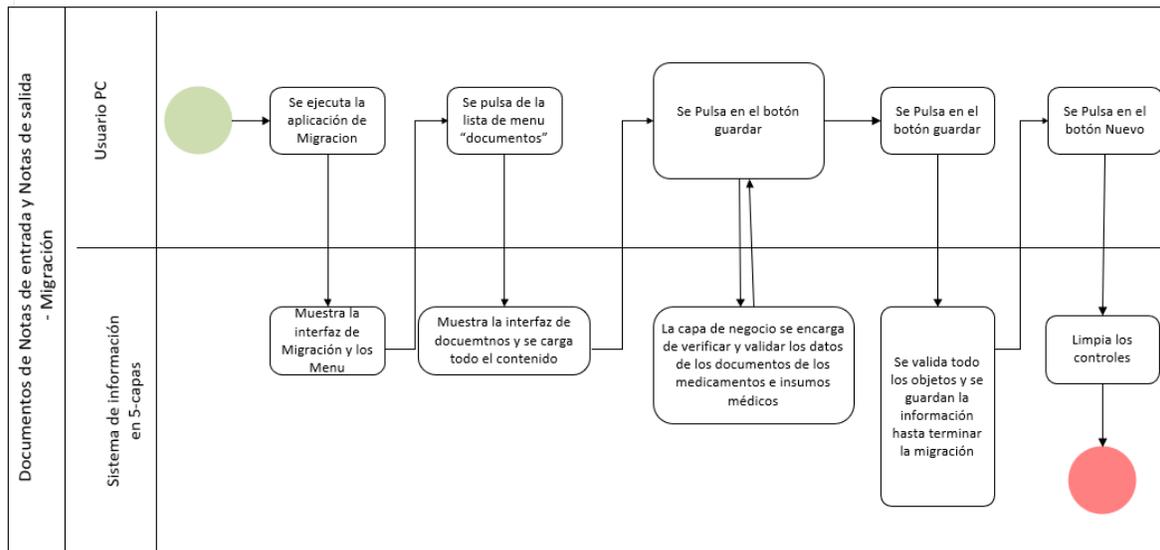


Figura 8

Diagrama de flujo Migración de Documentos

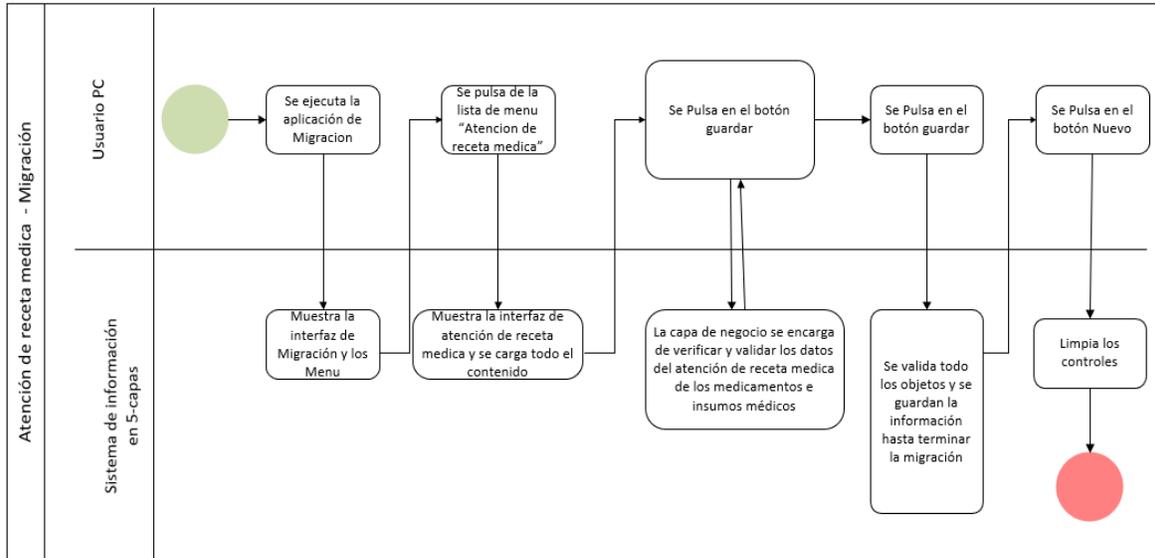


Figura 9

Diagrama de flujo Migración de Recetas médicas

Capítulo IV

DESARROLLO DE LA ARQUITECTURA N-TIER

—

4. DESARROLLO DE LA ARQUITECTURA N-TIER

4.1. Arquitectura de Sistema desarrollado

El sistema de información ha sido desarrollado bajo la arquitectura n-tier, proponiéndose cinco capas. Cada capa tiene un rol y responsabilidad única con la aplicación, así se modulariza y organizan las responsabilidades. Cada capa tiene crea una abstracción de la funcionalidad que está teniendo. Los componentes de cada capa solo deben hacer tareas relacionadas con la capa.

La arquitectura propuesta es una arquitectura vertical, como se puede observar en la Figura 12, donde la capa superior tiene conocimiento de la capa inferior; las capas inferiores no deben tener ningún conocimiento de las capas superiores lo cual garantiza que la dependencia esté bien organizada.

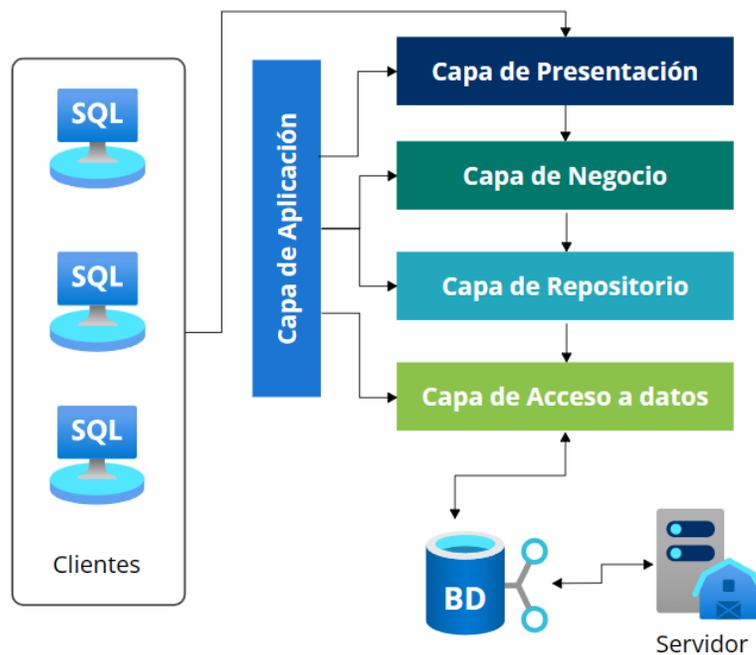


Figura 10 — Funcionamiento de la arquitectura N-TIER propuesta

En la Figura 10, se presenta la composición de arquitectura propuesta, con las siguientes capas:

1. **Capa de presentación:** Es la responsable de la presentación visual de la aplicación. Podemos decir que es la que se presenta al usuario, llamada también formulario o interfaz de presentación, esta captura los datos del usuario en el formulario e invoca a la

capa de negocio, trasmitiéndole los requerimientos del usuario, ya sea de almacenaje, edición, o de recuperación de la información para la consulta respectiva.

2. **Capa de negocio:** Es la responsable del procesamiento que tiene lugar en la aplicación. Esta capa interactúa con la capa de presentación para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al manejador de base de datos que realice una operación de almacenamiento, edición, eliminación, consulta de datos u otra.
3. **Capa Repositorio:** La capa de repositorio en la arquitectura N-tier es responsable de interactuar con la base de datos o cualquier otro sistema de almacenamiento de datos. Esta capa actúa como un intermediario entre la capa de negocios (o lógica de negocio) y la capa de datos, proporcionando una abstracción de la manipulación de datos para el resto de la aplicación.
4. **Capa Aplicación:** Es la que contiene los objetos que transportan datos entre procesos. La encargada de comunicar cliente y servidor, mediante llamadas usando un objeto (el DTO Data Transfer Object u Objeto de Transferencia de Datos) que agrega los datos que habrían sido transferidos por cada llamada, pero que son entregados en una sola llamada. Esta capa, se comunica directamente con las demás para transportar la información, por medio del mapeo de la estructura del objeto, que pueden ser listas, bool, string u otros.
5. **Capa de acceso a datos:** Esta capa se encarga de acceder a los datos, se debe usar la capa de datos para almacenar y recuperar toda la información de sincronización del Sistema. Es aquí donde se implementa las conexiones al servidor y la base de datos propiamente dicha, se invoca a los procedimientos almacenados los cuales reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio

Esta capa es la que accede directamente al sistema gestor de base de datos, asegurando que ninguna otra capa pueda consumir la data si no es pasando por ella, al convertirse en el único puente para almacenar y recuperar la información (Alseelawi et al. 2020).

4.2. Abstracción de las capas del modelo propuesto y definición de tareas

Módulo de Farmacia

El módulo de farmacia es el responsable de los siguientes procesos: i) Registro y seguimiento de inventario, que permite llevar un registro detallado de todos los

medicamentos y suministros médicos disponibles en la farmacia, incluyendo su cantidad, fecha de caducidad, ubicación en el almacén, entre otros datos relevantes, ii) Gestión de recetas, que es la dispensación de medicamentos a pacientes, incluye la tarea de validación de prescripciones, gestión electrónica de recetas médicas, iii) Facturación y cobros de los medicamentos y/o insumos médico quirúrgicos proporcionados por la farmacia, así como la gestión de pagos y cobros.

Tareas definidas para el módulo de farmacia, diferenciado por capas:

- **Capa de Presentación:** Se encuentran las ventanas para la atención de recetas. Esta capa garantiza tres funciones principales: Interfaz de programación visual/textual, paneles de control y formularios. La capa se construye para asegurar una experiencia de usuario positiva y satisfactoria al utilizar el software. Esto implica considerar aspectos como la usabilidad, accesibilidad, diseño visual, flujo de trabajo y respuesta rápida a las acciones del usuario. Es responsable además de la validación de datos en tiempo real o antes de enviar los datos al servidor para asegurar que la información ingresada por el usuario sea correcta y cumpla con ciertas reglas o restricciones. Esta capa debe mostrar de manera clara y organizada los resultados, informes, gráficos u otra información generada por el software en respuesta a las acciones del usuario.
- **Capa de aplicación:** Se encarga de transportar los datos de venta y sus detalles, por medio del Mapeo objeto-relacional (ORM), para mapear objetos de la aplicación a tablas de la base de datos y viceversa. Esto facilita la integración entre el modelo de datos de la aplicación y el modelo de datos de la base de datos.
- **Capa de negocio:** en esta capa se trabajan las reglas, procesos y operaciones que rigen el funcionamiento del Departamento de Farmacia del HRGDV. Esta lógica define el cumplimiento de validaciones de datos dentro de las reglas del negocio, como por ejemplo fechas de nacimiento válidas, fechas de emisión de una receta, verificación de paciente que solicita los medicamentos, cálculo de precios y verificación de disponibilidad de productos, validación de medios de pago, que en este caso se trabaja por dos modalidades: pago en efectivo y registro de usuarios asegurados por el Sistema Integral de Salud que no realizan el pago, pero el consumo se almacena en esta

categoría; en ambos casos se genera una facturación electrónica vinculada al sistema nacional de emisión de comprobantes de pago electrónico del Perú, se realizan además, los cálculos complejos para generar informes, métricas o resultados específicos de procesamiento de datos. En esta capa también se implementan las reglas de acceso y permisos para garantizar que solo usuarios autorizados puedan acceder a ciertas funcionalidades o datos sensibles.

- **Capa de Repositorio:** esta capa intermediaria entre la capa de negocios y la capa de datos se crea para que sea la que implemente métodos que acepten condiciones y filtros para realizar un gran número de consultas diferentes con respecto a la atención de recetas. Esta capa implementa la lógica para acceder a un registro, para editarlo o eliminarlo, ocultando la complejidad de la manipulación de datos a la capa de negocios. Se han implementado también estrategias para optimización de consultas para mejorar el rendimiento de las operaciones de acceso a datos, como el uso de cachés, consultas parametrizadas y técnicas de optimización de consultas SQL.
- **Capa de acceso a Datos:** La capa de acceso a datos establece y gestiona la conexión con la base de datos, permitiendo la comunicación bidireccional para enviar consultas y recibir resultados. La base de datos utilizada en el proyecto es SQL Server. Es la encargada de ejecutar consultas SQL u otras consultas de acceso a datos para realizar operaciones CRUD (la recuperación de datos, inserción de nuevos registros, actualización de registros existentes y eliminación de registros en la base de datos referidos a la atención de recetas).

Módulo Almacén

El área de almacén es la responsable de controlar el stock de los medicamentos y suministros médicos entregados por proveedores externos, asegurándose de que coincidan con los pedidos realizados y que estén en buenas condiciones y generar inventarios periódicos para verificar la disponibilidad de productos.

Tareas definidas para el módulo de almacén, diferenciado por capas:

- **Capa de Presentación:** Se encarga de gestionar la interfaz de usuario y la interacción entre el usuario y el sistema de gestión de almacén. Las funciones centrales son la

visualización a través de nivel de usuario sobre los stocks de productos y sus lotes de vencimiento. La interfaz debe garantizar una estructura de navegación lógica y coherente, realizar validaciones de datos, asegurando que la información proporcionada sea correcta y cumpla con los requisitos establecidos por el sistema. Las peticiones del usuario se envían hacia la capa de negocio para enviar y recibir datos, procesar solicitudes y realizar operaciones referidos al procesamiento del stock de medicamentos y/o insumos quirúrgicos.

- **Capa de negocios:** se encarga de la lógica empresarial y funcionalidades específicas relacionadas con la gestión de almacén referido al stock de los medicamentos y/o insumos quirúrgicos. Las reglas de negocio implementadas están orientadas a la disponibilidad de medicamentos, una condición en la que el insumo debe encontrarse listo para utilizarse en la cantidad necesaria. Los posibles estados de disponibilidad responden a reglas específicas de cálculo de indicadores, que pueden encontrarse en normostock, sobrestock, substock, en desabastecido o sin rotación. Una vez obtenidos estos datos, se generan métodos que realizan una visión global del estado de abastecimiento general del Departamento de Farmacia, así se pueden realizar los cálculos de Indicadores de Disponibilidad de Insumos.
- **Capa de aplicación:** esta capa se comunica directamente con las demás capas transportando los datos de stock y sus detalles.
- **Capa de repositorio:** esta capa intermedia con la capa de datos, facilitando su acceso y manipulación por medio de una abstracción que oculta la complejidad de las operaciones de base de datos a las capas superiores de la aplicación. En esta capa se incluyeron estrategias de optimización de consultas para mejorar el rendimiento de las operaciones de acceso a datos, como el uso de índices, cachés de datos y consultas parametrizadas.
- **Capa de datos:** esta capa gestiona la comunicación bidireccional con la capa de repositorio, está encargada de las operaciones de inserción de registros, selección de datos existentes, edición y eliminación de datos de acuerdo a las necesidades de la

aplicación. Se trabajan en ella búsquedas avanzadas, filtros, ordenamientos y cálculos.

4.3. Herramientas tecnológicas

Tabla 44

Herramientas utilizadas para el desarrollo del proyecto de sistema de información en N-TIER.

Nº	Herramienta
1.	Microsoft Visual Studio 2017
2.	Lenguaje de Programación C#
3.	Sql Server 2014 Enterprise
4.	Embarcadero 10.1 Diagramador de Base de Datos
5.	Entity Framework 6.0
6.	Windows Server 2016 Edición Estándar

4.4. Personas y roles del proyecto

Tabla 45

Descripción de los roles

Nro.	Rol	Descripción
1	Jefe del Departamento	Se trata de un profesional de la salud, un químico farmacéutico se encarga de gestionar la venta y la adquisición de los medicamentos e insumos médicos
2	Jefe del Almacén Especializado de Dpto. de Farmacia	Se trata de un químico farmacéutico, que se encarga de gestionar la disponibilidad de medicamentos e insumos médicos, como la buena distribución de ellos, también se encarga de controlar que el stock del sistema coincida con el stock físico.

3	Kardista	Es el encargado de velar las consistencias de ingresos y salidas de medicamentos e insumos médicos quirúrgicos del sistema con lo existente físicamente.
4	Dispensador	Son los químicos farmacéuticos encargados de realizar el seguimiento de dosis de medicamentos a los pacientes. Son encargados de realizar el reporte de consumo del paciente.
4	Expendedor	Es un técnico en farmacia, que se encarga de entregar los medicamentos e insumos de médicos quirúrgicos al paciente, así como también el registro al sistema de la receta médica atendida.
6	Programador	Desarrollador de los sistemas de información.
7	Tester	Programador

4.5. Creación de capas

Cada capa ha sido creada como una biblioteca de clases tal como sigue:

Tabla 46

Descripción de las capas

N-Tiers	Descripción
Lo1.WinFormsPHMS	Representa la capa de presentación
Lo2.Application	Representa la capa de aplicación la cual se encarga de transportar los datos entre las demás capas
Lo4.BusinessObjects	Representa la capa de negocio la cual se encarga de validar las propiedades de los objetos de negocio
Lo4.Repository	Representa la capa de Repositorio
Lo5.DataAccess.SqlEf	Representa la capa de Acceso a Datos

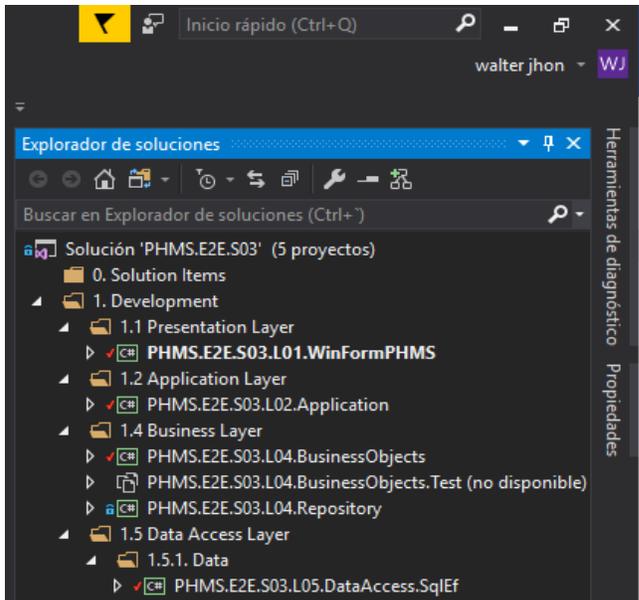


Figura 11 — Representación de las N-TIER física bibliotecas de clases

En la Figura 11, se muestra la solución construida, compuesta por varios proyectos, dentro de los cuales se realizó la separación lógica de funciones en capas de acuerdo a la arquitectura n-tier.

4.6. Utilización de las capas

Para poder utilizar las capas se tiene que referenciar como se muestra en la Figura 12, para la capa de Aplicación.

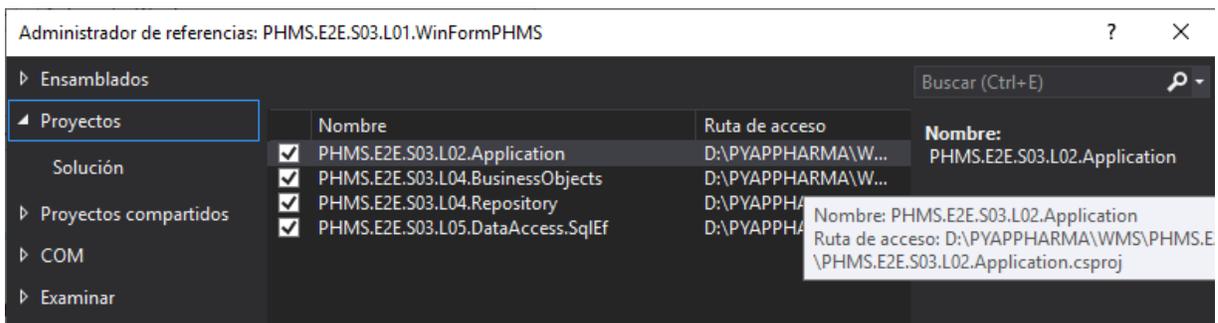


Figura 12 — Referenciación de proyectos dentro de la capa de presentación bajo la arquitectura n-tier

4.7. Implementación de las capas en la solución de Visual Studio

La capa de presentación esta realizada en formulario de escritorio que contiene un formulario de contenedor con sus respectivos menús.



Figura 15 — Diseño del formulario de la capa de presentación

En la Figura 13, se visualiza el formulario de la nota de ingreso que permite realizar un nuevo pedido de receta de farmacia.

Figura 13 — Formulario de la nota de ingreso

— Capa de negocios

Una vez registrados los datos de la receta, se procede a guardar la información la cual se envía a la capa de Negocio con los objetos de negocio DocumentEdit.

```
try
{
    documentEdit = DocumentEdit.NewDocumentEdit();
    documentEdit.DocumentTypeId = "f3de4184-ob10-4c4b-8e3d-c37f501e39c0";
    documentEdit.StoreOriginId=
    lueAlmacenOrigenVirtual.GetColumnValue("StoreId").ToString();
    documentEdit.StoreDestinationId=
    lueAlmacenDestinoVirtual.GetColumnValue("StoreId").ToString();
    documentEdit.Number = minumero;
    txtNumeroOperacion.Text = minumero.ToString();
    minumero = minumero + 1;
    documentEdit.Movement = "E";
    if (txtRuc.Text=="")
    {
        documentEdit.SupplierId = null;
    }
}
```

```
else
{
    documentEdit.SupplierId = supplierEdit.SuppliersId;
}
documentEdit.Total = decimal.Parse(txtTotal.Text);
documentEdit.Reference = txtReferencia.Text;
documentEdit.NumberPurchaseProcess = txtProceso.Text;
documentEdit.Status = 1;
if (gridView1.RowCount>o)
{
    for (int i = 0; i < gridView1.RowCount; i++)
    {
        documentDetailEdit = documentEdit.DocumentDetails.AddNew();
        documentDetailEdit.DocumentId = documentEdit.DocumentId;
        documentDetailEdit.ProductId=this.gridView1.GetRowCellValue(i,
ProductId").ToString();

        documentDetailEdit.Lot=this.gridView1.GetRowCellValue(i,
"Lot").ToString();
        documentDetailEdit.Quantity = int.Parse(
this.gridView1.GetRowCellValue(i, "Quantity").ToString());
        documentDetailEdit.Code= this.gridView1.GetRowCellValue(i,
"Code").ToString();
        documentDetailEdit.PurchasePrice =
decimal.Parse(this.gridView1.GetRowCellValue(i, "PurchasePrice").ToString());
        documentDetailEdit.ExpirationDate = DateTime.Parse(
this.gridView1.GetRowCellValue(i, "ExpirationDate").ToString());
        documentDetailEdit.Total = decimal.Parse(this.gridView1.GetRowCellValue(i,
"Total").ToString());
    }
}

if (gridView2.RowCount > o)
{
    for (int i = 0; i < gridView2.RowCount; i++)
    {
        documentFileEdit = documentEdit.DocumentFiles.AddNew();
        documentFileEdit.DocumentId = documentEdit.DocumentId;
        documentFileEdit.Name = this.gridView2.GetRowCellValue(i,
"Name").ToString();
        documentFileEdit.Route = this.gridView2.GetRowCellValue(i,
"Route").ToString();
        documentFileEdit.Observation = this.gridView2.GetRowCellValue(i,
"Observation").ToString();
    }
}
```

```

    }
  }
  SaveProductCommand.RegisterProduct(documentEdit);
  MessageBox.Show("Se Registro el documento");
  }
  catch (Exception ex)
  {

    MessageBox.Show(ex.ToString());
  }
}

```

Figura 14 — Implantación del botón guardar

— Capa de Aplicación

La capa encargada de transportar los datos aquí vemos el objeto de DocumentoDto.

```

public class DocumentoDto : DtoBase
{
    /// <summary>
    /// Gets or sets Document Id
    /// </summary>
    public string DocumentId { get; set; }

    /// <summary>
    /// Gets or sets DocumentTypeId
    /// </summary>
    public string DocumentTypeId { get; set; }

    /// <summary>
    /// Gets or sets SupplierId
    /// </summary>
    public string SupplierId { get; set; }

    /// <summary>
    /// Gets or sets StoreOriginID
    /// </summary>
    public string StoreOriginId { get; set; }

    /// <summary>
    /// Gets or sets StoreDestinationId
    /// </summary>

```

```

public string StoreDestinationId { get; set; }

/// <summary>
/// Gets or sets Movement
/// </summary>
public string Movement { get; set; }

/// <summary>
/// Gets or sets Number
/// </summary>
public int Number { get; set; }

```

Figura 15 — Implementación del objeto Documento en la capa Aplicación

— Capa de negocio

La capa encargada de manejar los datos de objeto de negocio DocumentEdit como sus propiedades y los detalles.

```

public partial class DocumentEdit : MefBusinessBase<DocumentEdit>
{
    #region "Properties"
    /// <summary>
    /// Metadata DocumentIdProperty
    /// </summary>
    public static readonly PropertyInfo<string> DocumentIdProperty =
        RegisterProperty<string>(c => c.DocumentId);

    /// <summary>
    /// Gets ID Document, auto Key generated
    /// </summary>
    public string DocumentId
    {
        .
    }
}

```

Figura 16 — Implementación del objeto DocumentEdit en la capa de negocio

Para realizar la inserción de datos se aplica una transacción la cual si ocurriese un error no ha de guardarse.

```
[Transactional(TransactionTypes.TransactionScope,
TransactionIsolationLevel.ReadCommitted)]

protected override void DataPortal_Insert()
{
    using (BypassPropertyChecks)
    {
        {
            var dto = new DocumentDto();
            InsertOnInData(dto);
        }
    }
}
```

Figura 17 — Implementación de la función `_Insert` para el envío objeto `DocumentDto` a la capa Repositorio

De ahí para la inserción envía a la capa de repositorio `DocumentRepository`

`DocumentMef.Insert`

— Capa de repositorio

La capa repositorio se implementa la lógica para acceder a la inserción actualización o eliminación, como recuperar lista de objetos que en este caso de realiza la inserción del registro de Nota de ingreso.

```

public interface IDocumentRepository : IRepositoryBase<DocumentDto>,
IQueryableBase<DocumentDto>
{
    void Delete(string id);
    T FetchById(string id);
    void Insert(T dto);
    void Update(T dto);

    // <summary>
    /// Fetch a list of document by status.
    /// </summary>
    /// <param name="active">true = active, false = Inactive</param>
    /// <returns>List of Document Data Transfer Object</returns>
    void Insert(T dto);

    List<DocumentDto> FetchAll(int status);
}

```

Figura 18 — Implementación de la interfaz *IDocumentRepository* en la capa repositorio

— Capa de Datos

Establece la conexión con la base de datos y recupera la representación de las tablas del base de datos con las entidades de la capa de accesos a datos.

```

public class EntityDbContext : DbContext
{
    #region Constructors
    public EntityDbContext()
    {
    }

    public EntityDbContext(string connectionString) : base(connectionString)
    {
    }

    #endregion
    public DbSet<SuppliersTable> Suppliers { get; set; }
    public DbSet<ProductTable> Products { get; set; }
}

```

```

public DbSet<StockProductTable> StockProducts { get; set; }
public DbSet<StoreTable> Stores { get; set; }
public DbSet<StockProductDetailsTable> StockProductDetails { get; set;}
public DbSet<WarehouseBusinessTable> WarehouseBusiness { get; set; }
public DbSet<PriceTable> Prices { get; set; }
public DbSet<DocumentTable> Documents { get; set; }

```

Figura 19 — Implementación de DbContext en la capa de Acceso a Datos

Representación de la tabla de la base de datos con Entity en donde se agregan los nombres del campo de la tabla de la base de datos y su clave principal.

```

[Table("C1S01T01_Documents")]
public class DocumentTable
{
    /// <summary>
    /// Gets or sets Adapter Type Id
    /// </summary>
    [Key, Column("varDocumentId")]
    public string DocumentId { get; set; }

    /// <summary>
    /// Gets or sets DocumentTypeId
    /// </summary>
    [Column("varDocumentTypeId")]

```

Figura 20 — Implementación de la entidad de DocumentTable

Se realiza la inserción del documento de Nota de Ingreso a la base de datos

```
public class DocumentSqlEf : IDocumentRepository
{
    public void Insert(DocumentDto dto)
    {
        using (var ctx =
DbContextManager<EntityDbContext>.GetManager(Implementation.DataBaseConnectionStringKey))
        {
            DocumentTable table = new DocumentTable();

            Mapper.Map(dto, table);
            if (dto.SupplierId=="")
                -
```

Figura 21 — Implementación del interfaz del Repositorio en la capa de acceso a datos

Capítulo V

EVALUACIÓN DE USABILIDAD

5. EVALUACIÓN DE USABILIDAD

5.1. Usabilidad

La usabilidad en software se refiere a la medida en que un sistema informático o una aplicación digital son fáciles de usar y comprender para los usuarios finales. Se enfoca en la experiencia del usuario al interactuar con el software y busca garantizar que las personas puedan realizar sus tareas de manera eficiente, efectiva y satisfactoria (Stewart 2009).

Establecimiento de la escala de medida de usabilidad.

- Xerox estableció la siguiente escala de cinco puntos para sus evaluaciones de usabilidad (González, Afra y Lorés 2006).
- Cosmético: valores de 8 a 10. No afectará a la usabilidad del sistema. Existen errores que no impactan en el diseño, pero sí en su apreciación.
- Menor: valores de 6 a 8. Los usuarios pueden fácilmente trabajar con algunos problemas.
- Medio: valores de 4 a 6. Los usuarios tropiezan con problemas, pero se adaptan rápidamente a ellos.
- Mayor: valores de 2 a 4. Los usuarios tienen dificultades, pero son capaces de encontrar la forma de trabajar a pesar de ellas. Puede poner en peligro el éxito de la tarea.
- Catastrófico: valores de 0 a 2. Los usuarios son incapaces de hacer su trabajo. Imposible terminar la tarea con éxito

Esta escala tiene un rango de 0 a 10, por lo cual, se puede hacer un paralelo a la escala porcentual de 0% a 100%. De tal forma, en la Figura 25 y Tabla 49, se propone la valorización de las medidas de usabilidad en tres categorías: aceptable, mejorable e inaceptable.

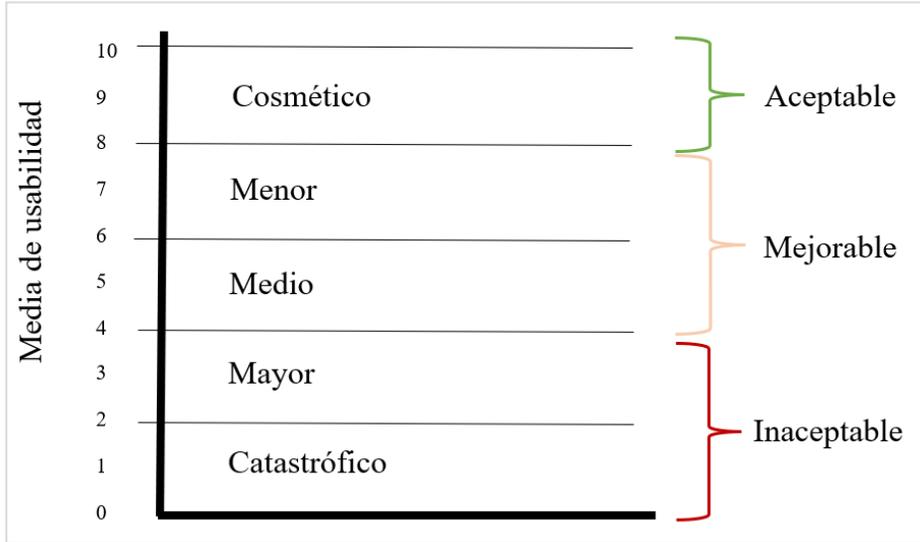


Figura 22
Valorización de media de usabilidad

Tabla 47
Escala y porcentaje de evaluación de Usabilidad

Escala de evaluación	Porcentaje
Aceptable	80%-100%
Mejorable	40%-80%
Inaceptable	0%-40%

5.2. Evaluación de la usabilidad del sistema

La usabilidad se evaluó a través del Cuestionario PSSUQ (Post Study System Usability Questionnaire), que mide la calidad del sistema y la satisfacción del usuario, por medio de 19 preguntas, que buscan determinar la utilidad del sistema, la calidad de la información y la calidad de la interfaz (Lewis 1992).

Los resultados se presentan en la Tabla 48 y lo respondido por los usuarios.

Tabla 48

Preguntas de Usabilidad bajo el PSSUQ respondidas por 10 usuarios del HRGV

Ítems	Preguntas	Usuario1	Usuario2	Usuario3	Usuario4	Usuario5	Usuario6	Usuario7	Usuario8	Usuario9	Usuario10
US1	En general, estoy satisfecho con lo fácil que es usar este sistema	X	X	X	X	X	X	X	X	X	X
US2	Era simple usar este sistema	X	X	X	X	X	X	X	X	X	X
US3	Puedo completar efectivamente mi trabajo usando este sistema	X	X	X		X	X	X	X	X	X
US4	Soy capaz de completar mi trabajo rápidamente usando este sistema	X		X	X			X	X	X	X
US5	Soy capaz de completar mi trabajo eficientemente usando este sistema	X	X	X	X	X	X	X	X	X	X
US6	Me siento cómodo usando este sistema	X	X	X	X	X	X	X	X	X	X
US7	Fue fácil aprender a usar este sistema.	X	X	X	X	X	X	X	X	X	X
US8	Creo que me hice productivo rápidamente usando este sistema.	X	X	X	X	X	X	X	X	X	X
US9	El sistema manda mensajes de error que claramente me dicen cómo solucionar problemas	X	X	X	X	X	X	X	X	X	X
US10	Siempre que me equivoco al usar el sistema, me permite recuperarme fácil y rápidamente	X	X	X	X	X	X	X	X	X	X

US11	La información (como ayuda en línea, mensajes en pantalla y otra documentación) proporcionada con este sistema es clara			X	X	X	X	X	X	X	X
US12	Es fácil encontrar la información que necesito	X	X	X	X	X	X	X	X	X	X
US13	La información proporcionada por el sistema es fácil de entender.	X	X	X	X	X	X	X	X	X	X
US14	La información es efectiva para ayudarme a completar las tareas en cada escenario.	X	X	X	X	X	X	X	X	X	X
US15	La organización de la información en las pantallas del sistema es clara.		X	X	X	X	X		X	X	X
US16	La interfaz de este sistema es agradable.	X	X	X	X	X	X	X	X	X	X
US17	Me gusta usar la interfaz de este sistema.	X	X	X	X	X	X	X	X	X	X
US18	Este sistema tiene todas las funciones y capacidades que espero que tenga	X	X	X	X	X	X	X	X	X	X
US19	En general, estoy satisfecho con este sistema	X	X	X		X	X	X	X	X	X

De acuerdo al cuestionario aplicado, se obtuvieron los siguientes resultados:

Tabla 49

Cuadro de resultado de las preguntas de usabilidad

Usuarios	Tareas completas (A)	Nº total de tareas (B)	A/B=X	Valor promedio obtenido
----------	----------------------	------------------------	-------	-------------------------

Usuario1	17	19	0.89	0.95
Usuario2	17	19	0.89	
Usuario3	19	19	1.00	
Usuario4	17	19	0.89	
Usuario5	18	19	0.95	
Usuario6	18	19	0.95	
Usuario7	18	19	0.95	
Usuario8	19	19	1.00	
Usuario9	19	19	1.00	
Usuario10	19	19	1.00	

El valor promedio obtenido es de 0.95 que equivaldría al 95% por lo que, de acuerdo a la escala y porcentaje de evaluación de usabilidad, el sistema se encuentra en el rango 80%-100% que pertenece en la escala como Aceptable.

Figura 23 — Implementación del interfaz del Repositorio en la capa de acceso a datos

Capítulo VI

CONCLUSIONES Y REFERENCIAS BIBLIOGRAFICAS

6. CONCLUSIONES Y REFERENCIAS BIBLIOGRAFICAS

6.1. Conclusiones

Se desarrolló e implementó un sistema de información con arquitectura N-TIER en el Departamento de Farmacia en el Hospital Regional Guillermo Díaz de la Vega Nivel II-2, compuesto por las capas de presentación, de negocio, de repositorio, de aplicación y la capa de acceso a datos. Esta separación lógica, encapsuló la abstracción de funcionalidades por cada capa, haciendo que cada una de ellas se encargara de una tarea específica dentro de la arquitectura, brindando desacoplamiento, separación de funciones, mayor facilidad de mantenibilidad del código, reutilización de componentes y facilitó la colaboración y trabajo en paralelo en el equipo desarrollador. La propuesta de esta arquitectura es una representación de las prácticas de la industria del software, adaptada a una necesidad específica.

La fase de análisis, ha sido guiada por el modelo conceptual granular para definición de requisitos, que resultó en una percepción más alta de completitud de las tareas, según lo descrito en el documento de especificación de requisitos por parte de los usuarios del software. Así mismo, se trabajó con la creación de la pila de Sprint, historias de usuario y tarjetas de Clase – Responsabilidad – Colaboración; que en su conjunto permitieron dilucidar y esclarecer en etapas tempranas el problema que el software debe resolver.

Finalmente, se realizó una evaluación de usabilidad bajo el Cuestionario PSSUQ, logrando una aceptación de 0.95, equivalente a Aceptable.

6.2. Referencias bibliográficas

ABDELRAHMAN, M.M., ZHAN, S. y CHONG, A., 2020. A Three-Tier Architecture Visual-Programming Platform for Building-Lifecycle Data Management. , pp. 445-452.

ALI BABAR, M., 2014. Making Software Architecture and Agile Approaches Work Together: Foundations and Approaches. *Agile Software Architecture*. S.l.: s.n., pp. 1-22.

- ALSEELAWI, N.S., ADNAN, E.K., HAZIM, H.T., ALRIKABI, H.T.S. y NASSER, K.W., 2020.** Design and implementation of an e-learning platform using N-tier architecture. *International Journal of Interactive Mobile Technologies*, vol. 14, no. 6, pp. 171-184. ISSN 18657923. DOI 10.3991/IJIM.V14I06.14005.
- AYQUIPA, R.A.R., ENRIQUEZ, H.R., HUAYLLANI, W.J.A., MEZARINA, Z.H.A. y CABRERA, M.J.I., 2019.** Challenges in the implementation of e-government for public institutions in Peru. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, pp. 347-351. ISSN 21531633. DOI 10.1145/3306500.3306572.
- BASS, L., CLEMENTS, P. y KATZMAN, R.,** *Software Architecture & Practice*. 3rd Edn. S.l.: s.n.
- CÁRDENAS, L., 2013.** El patrón de arquitectura n-capas con orientación al dominio como solución en el diseño de aplicaciones empresariales. *Tecnología y desarrollo*, vol. 11, no. 1, pp. 59-66. DOI <https://doi.org/10.18050/td.v11i1.679>.
- DALTON, J., 2018.** Class, Responsibilities, Collaborators (CRC) Cards. En: S. LINK (ed.), *Great Big Agile*. S.l.: s.n., pp. 153-154.
- DIGEMID, M.,** *Manual de indicadores de disponibilidad*. S.l.: s.n.
- FRITH, D., 2015.** Cloud Deployments: Is this the End of N-Tier Architectures? *Isse 2015*. S.l.: s.n., pp. 74-86. ISBN 9783658109349. DOI 10.1007/978-3-658-10934-9.
- GONZÁLEX, M.P., AFRA, P. y LORÉS, J.,** *Evaluación heurística*. S.l.: s.n.
- HUAYLLANI, W.J. y ROJAS, H., 2020.** A granular conceptual model to define requirements for evaluating the functional completeness of a pharmacy information system. *ACM International Conference Proceeding Series*, pp. 80-84. DOI 10.1145/3425329.3425344.
- IEEE, 2007.** IEEE 1471-2000™ Conceptual Framework for Architectural Description. . S.l.:
- IEEE, C.S., 2024.** Software Engineering Body of Knowledge (SWEBOK). *Bodies of Knowledge*.
- JAISWAL, M., 2021.** Software Architecture and Software Design. *SSRN Electronic Journal*, vol. 6, no. 11, pp. 2452-2454. DOI 10.2139/ssrn.3772387.

- LEWIS, J.R., 1992.** Psychometric Evaluation of the Post-Study System Usability Questionnaire: The PSSUQ. *Proceedings of the Human Factors Society Annual Meeting*. S.l.: s.n., pp. 1259-1260. DOI <https://doi.org/10.1177/154193129203601617>.
- MARTÍNEZ, E.E., LIPORACE, F.D.S., SOARES, R.D.P. y GALO A.C. LE ROUX, 2014.** Design and Implementation of a Real Time Optimization Prototype for a Propylene Distillation Unit. *Computer Aided Chemical Engineering*, vol. 33, pp. 1321-1326. DOI <https://doi.org/10.1016/B978-0-444-63455-9.50055-6>.
- PÁEZ, A., COIME, V., GÓMEZ, F., LAYANA, J., LAYANA, J. y AJILA, M., 2016.** Manual de procesos para la gestión farmacéutica. *Quito: S/N*,
- PAHLAVAN TAFTI, A., JANOSEPAH, S., MODIRI, N., MOHAMMADI NOUDEH, A. y ALIZADEH, H., 2011.** Development of a Framework for Applying ASYCUDA System with N-Tier Application Architecture. *Software Engineering and Computer Systems*. S.l.: s.n.,
- PALACIOS, J., Scrum Manager: Gestión de proyectos.** S.l.: s.n.
- STAFFORD, J. y WOLF, A., 2001.** Software architecture. *Handbook of Software Engineering and Knowledge Engineering*. S.l.: s.n., pp. 371-387.
- STEWART, T., 2009.** Usability evaluation. *Behaviour & Information Technology*, vol. 28, pp. 99-100. DOI <https://doi.org/10.1080/01449290902786510>.

ARQUITECTURA DE SOFTWARE EN N-CAPAS. UN ENFOQUE PRÁCTICO

Hesmeralda Rojas Enriquez

Doctor en Ciencias de la Computación

Docente Universitario, en la Universidad Nacional Micaela Bastidas de Apurímac

Correo Electrónico:

hrojas@unamba.edu.pe

ORCID:

0000-0002-9388-2991

Ronald Alberto Renteria Ayquipa

Docente Universitario, en la Universidad Nacional Micaela Bastidas de Apurímac

Correo Electrónico:

rrenteria@unamba.edu.pe

ORCID:

0000-0001-7736-162X

Virgilio Martínez Duran

Docente Universitario y Asistente de Informática, en Poder Judicial-corte

Superior de Justicia de Apurímac

Correo Electrónico:

vmartinez@unamba.edu.pe

ORCID:

0000-0001-5471-2423

Walter Jhon Huayllani Aguirre

Bach. Ing. Informática y de Sistemas

Desarrollo de Software, en Ministerio de Salud

Correo Electrónico:

walterjhon7@gmail.com

ORCID:

0009-0005-5069-5917

ISBN 978-628-96479-2-1