



UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

**FACULTAD DE CIENCIAS DE LA COMPUTACION
INGENIERÍA DE REQUISITOS**

NRC

23284

DOCENTE

Ing. Carlos Pillajo

NOMBRES

Acuña Marcelo
Arboleda Abner
Bonifaz Christian

UNIDAD 3

TÍTULO DEL PROYECTO

Healthy +

FECHA

08/03/2025

Índice de Contenidos

I.	Introducción	2
II.	Asignación de atributos a los requisitos.....	2
III.	Visualizaciones de los Requisitos	3
IV.	Priorización	5
	Matriz de Wieggers	5
V.	Versionado de Requisitos.....	7
	Control de versiones de requisitos	7
	Ejemplo de versiones	8
VI.	Gestión de Solicitudes de Cambio	8
	Flujo para proponer, evaluar y aprobar cambios a requisitos - específico.....	8
	1. Propuesta del Cambio	9
	2. Registro y Clasificación de la CR	9
	3. Análisis de Impacto	9
	4. Evaluación de la CR por la Junta de Control de Cambios (CCB).....	9
	5. Planificación e Implementación del Cambio (si fue aprobado)	10
	6. Verificación y Cierre	10
	Formato de Solicitud de Cambio:	11
VII.	Herramienta de Soporte.....	11
VIII.	Conclusiones	14
IX.	Recomendaciones.....	16

I. Introducción

Este documento detalla la gestión integral de los requisitos funcionales definidos para el sistema *Helthy+*, orientado a la asistencia de adultos mayores en la administración de sus medicamentos. Se abordan aspectos clave como la asignación de atributos, priorización, visualizaciones, control de versiones, gestión de cambios y uso de herramientas de soporte.

II. Asignación de atributos a los requisitos

- a. Definir qué atributos se manejarán (ID, estado, prioridad, versión, responsable, fuente, etc.).

Atributo	Justificación
ID	Se utiliza para identificar de forma única cada requisito y facilitar su trazabilidad a lo largo del proyecto.
Tipo	Facilita también la asignación de responsables adecuados.
Descripción	Brinda el detalle claro y completo del requisito.
Versión	Permite llevar un control de los cambios realizados a lo largo del tiempo.
Estado	Refleja el ciclo de vida de cada requisito.
Esfuerzo	Indica la complejidad relativa o la cantidad de recursos requeridos para implementar el requisito.
Responsable	Define claramente quién es la persona o equipo encargado del análisis, diseño o validación del requisito, favoreciendo la responsabilidad y comunicación.
Fuente	Indica el origen del requisito lo que ayuda a justificar su necesidad y a evaluar su prioridad desde el punto de vista del negocio.
Estabilidad	Estimar si un requisito es propenso a cambios permite anticipar riesgos y priorizar requisitos más estables en fases tempranas del desarrollo.
Riesgo	Refleja el impacto potencial del fallo o mal diseño del requisito.

- b. Presentar una tabla con todos los requisitos, donde se evidencien los atributos definidos.

ID	Tipo	Descripción	Versión	Estado	Esfuerzo	Responsable	Fuente	Estabilidad	Riesgo
RF01	Funcional	Gestión de medicamentos	v1.0	Aprobado	Alta	Marcelo Acuña	Análisis funcional Medisafe +	Estable	Medio
RF02	Funcional	Recordatorios de medicación	v1.0	Aprobado	Alta	Marcelo Acuña	Análisis funcional Medisafe + mejora propuesta	Estable	Medio
RF03	Funcional	Informes del historial de medicamentos	v1.0	Propuesto	Media	Abner Arboleda	Análisis funcional Medisafe + Aporte del grupo	Estable	Bajo
RF04	Funcional	Seguimiento de síntomas o efectos secundarios	v1.0	En revisión	Media	Christian Bonifaz	Aporte del grupo	Volátil	Medio
RF05	Funcional	Gestión de perfiles múltiples	v1.0	Aprobado	Media	Marcelo Acuña	Análisis funcional	Estable	Bajo

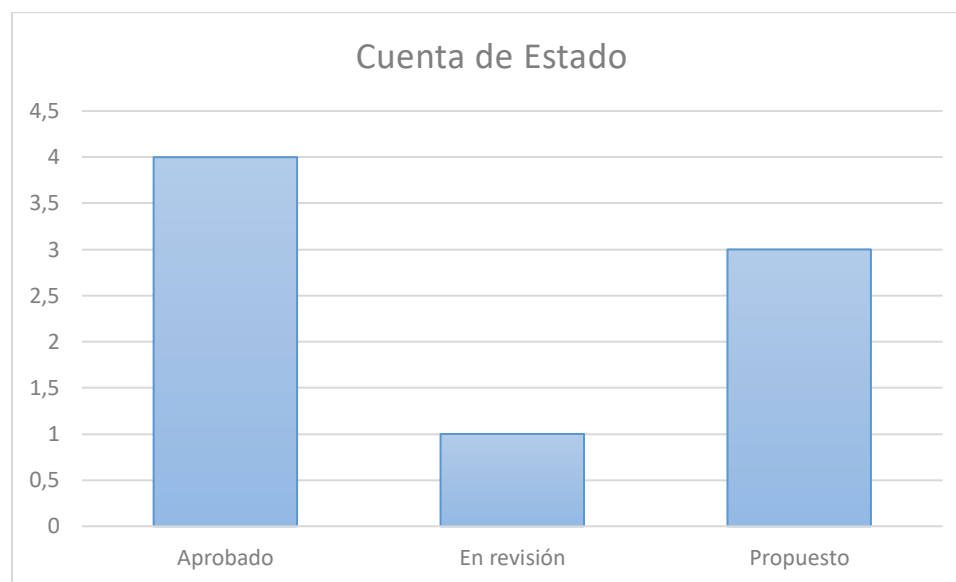
RF06	Funcional	Citas médicas	v1.0	Propuesto	Baja	Abner Arboleda	Análisis funcional Medisafe	Estable	Bajo
RF07	Funcional	Reconocimiento de recetas	v1.0	Propuesto	Media	Christian Bonifaz	Aporte del docente	Estable	Medio
RF08	Funcional	Configuración de idioma	v1.0	Aprobado	Baja	Abner Arboleda	Aporte del grupo	Estable	Bajo

III. Visualizaciones de los Requisitos

- a. Presentar diagramas o vistas que permitan entender mejor los requisitos.
Ejemplo: gráficos de dependencias o vistas filtradas por prioridad.

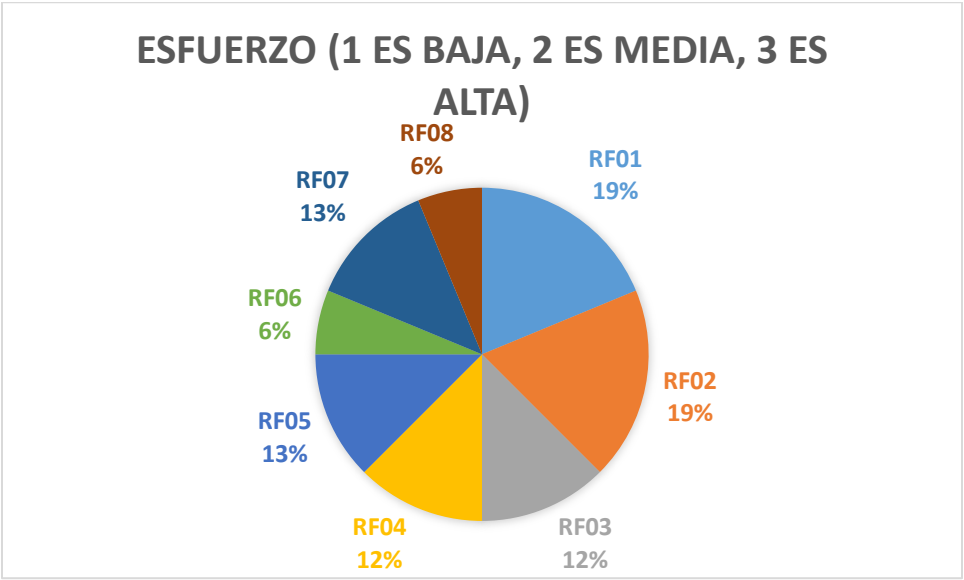
Cantidad de Requisitos en Estado

Estado	Cuenta de Estado
Aprobado	4
En revisión	1
Propuesto	3



Esfuerzo en Requisito

ID	Esfuerzo (1 es Baja, 2 es Media, 3 es Alta)
RF01	3
RF02	3
RF03	2
RF04	2
RF05	2
RF06	1
RF07	2
RF08	1



IV. Priorización

Matriz de Wiegers

Determinar los pesos relativos para beneficio, perjuicio, costo y riesgo.

Criterio	Peso General (0-10)	Justificación
Beneficio	9	El proyecto es una aplicación de salud donde el beneficio directo al usuario (cumplimiento de medicación, reducción de errores) es prioritario. Funciones que ayuden a mejorar adherencia y control tienen gran valor.
Perjuicio	8	Si no se implementa una función crítica (como recordatorios o historial), la aplicación pierde gran parte de su utilidad y puede generar riesgos en la salud del usuario.
Costo	6	Es importante, pero no es el factor más determinante.
Riesgo	7	Hay riesgos técnicos (OCR, recordatorios persistentes en móviles, multilinguaje) y de usabilidad, pero no son tan altos como el beneficio y perjuicio.

Determinar los pesos relativos de cada requisito para beneficio, perjuicio, costo y riesgo.

Req.	Beneficio (B)	Perjuicio (P)	Costo (C)	Riesgo (R)	Justificación breve
RF01 Gestión de Medicamentos	10	9	4	4	Función central del sistema; relativamente fácil de implementar.
RF02 Recordatorios	10	10	5	5	Vital para la adherencia; riesgo medio por notificaciones persistentes.
RF03 Informes	8	8	6	4	Útil para seguimiento médico, pero no tan crítico como recordatorios.
RF04 Seguimiento de Síntomas	8	7	6	5	Complementa el control, pero es extra frente a lo esencial.
RF05 Perfiles Múltiples	7	6	7	6	Útil para cuidadores, pero alto costo y riesgo en la gestión de datos.
RF06 Citas Médicas	8	7	5	4	Importante para salud, pero menor que medicación directa.
RF07 OCR de Recetas	7	6	8	8	Innovador, pero costoso y de alto riesgo técnico.
RF08 Idioma	6	5	3	3	Valor moderado, fácil de implementar.

Calcular los valores totales y los valores porcentuales para cada requisito

$$\text{Valor\% Beneficio } R_i = R_i \times \text{PesoBeneficio} + \text{Perjuicio } R_i \times \text{PesoPerjuicio}$$

Calcular las prioridades de los requisitos individuales:

$$\text{Prioridad}(R_i) = \frac{\text{Valor\%}(R_i)}{(\text{Costo\%}(R_i) \times \text{PesoCosto} + \text{Riesgo\%}(R_i) \times \text{PesoRiesgo})}$$

	Peso Beneficio	Peso Perjuicio			Peso Costo		Peso Riesgo			
Peso Relativo	9	8			6		7			
Requerimiento	Beneficio Relativo	Perjuicio Relativo	Total	Valor%	Costo Relativo	Costo%	Riesgo Relativo	Riesgo%	Prioridad	Rank
RF01	10	9	162	15.6	4	9.1	4	10.3	0.123	1
RF02	10	10	170	16.3	5	11.4	5	12.8	0.104	2
RF03	8	8	136	13.1	6	13.6	4	10.3	0.085	5
RF04	8	7	128	12.3	6	13.6	5	12.8	0.072	6
RF05	7	6	111	10.7	7	15.9	6	15.4	0.053	7
RF06	8	7	128	12.3	5	11.4	4	10.3	0.088	4
RF07	7	6	111	10.7	8	18.2	8	20.5	0.042	8
RF08	6	5	94	9.0	3	6.8	3	7.7	0.095	3
Total	64	58	1040	100	44	100	39	100		

Requerimientos ordenados:

Requerimiento	Beneficio Relativo	Perjuicio Relativo	Total	Valor%	Costo Relativo	Costo%	Riesgo Relativo	Riesgo%	Prioridad	Rank
RF01	10	9	162	15.6	4	9.1	4	10.3	0.123	1
RF02	10	10	170	16.3	5	11.4	5	12.8	0.104	2
RF08	6	5	94	9.0	3	6.8	3	7.7	0.095	3
RF06	8	7	128	12.3	5	11.4	4	10.3	0.088	4
RF03	8	8	136	13.1	6	13.6	4	10.3	0.085	5
RF04	8	7	128	12.3	6	13.6	5	12.8	0.072	6
RF05	7	6	111	10.7	7	15.9	6	15.4	0.053	7
RF07	7	6	111	10.7	8	18.2	8	20.5	0.042	8

Conclusión

Rank	Requisito	Motivo clave de su posición
1	RF01 (Gestión de Medicamentos)	Tiene altísimo beneficio y perjuicio , y un costo y riesgo bajo (4 en ambos). Esto hace que el numerador sea grande y el denominador pequeño, resultando en la mayor prioridad (0.123).
2	RF02 (Recordatorios)	Aunque tiene un riesgo y costo algo mayor (5 en ambos), su beneficio y perjuicio son los más altos del conjunto (10 cada uno), sosteniendo una prioridad alta (0.104).
3	RF08 (Idioma)	Tiene beneficio/perjuicio bajos, pero costos y riesgos muy bajos (3 y 3), lo cual reduce el denominador y le permite un buen puntaje (0.095).
4	RF06 (Citas médicas)	Beneficio y perjuicio aceptables (8 y 7), con riesgo/costo medianos. Balanceado, pero no destaca ni por alto valor ni bajo costo.
5	RF03 (Informes)	Igual que RF06, aunque con una leve diferencia en perjuicio, por eso cae un puesto.
6	RF04 (Síntomas / efectos secundarios)	Tiene un buen valor, pero su costo y riesgo son algo más altos (6 y 5), lo que reduce su prioridad (0.072).
7	RF05 (Perfiles múltiples)	Valor medio, pero también tiene costo alto (7) y riesgo moderado (6), bajando su prioridad a 0.053.

8	RF07 (OCR de recetas)	Aunque aporta valor, tiene el mayor costo y riesgo (8 y 8) de todos, haciendo que el denominador pese demasiado en la fórmula, resultando en la prioridad más baja (0.042) .
---	-----------------------	--

Observaciones generales:

- Los requisitos con alto beneficio y bajo costo/riesgo fueron los mejor posicionados.
- RF07, aunque innovador, queda relegado por su complejidad técnica.
- RF08 (Idioma), aunque no es funcionalmente crítico, sube en el ranking por su bajo esfuerzo.
- RF03, RF04 y RF05 tienen un impacto moderado y se ven penalizados por sus costos.

V. Versionado de Requisitos

Control de versiones de requisitos

El control de versiones de requisitos permite gestionar los cambios a lo largo del ciclo de vida del proyecto, asegurando trazabilidad, control de modificaciones y alineación entre equipos.

Se usará un esquema de versionado semántico, aplicado a cada requisito funcional (RF), donde se identificará cada cambio mediante un código de versión. Este sistema ayudará a:

- Controlar actualizaciones o correcciones en los requisitos.
- Mantener un historial claro de cambios.
- Evitar confusiones entre versiones antiguas y actuales.
- Facilitar auditorías y revisiones.

Reglas de versionado

Cada requisito tendrá un identificador único (por ejemplo: RF01).

Se le asociará una versión en el formato vX.Y, donde:

- X (mayor): Cambio significativo que modifica la intención, alcance o funcionalidad del requisito.
- Y (menor): Cambio menor como redacción, ajustes técnicos, aclaraciones o correcciones ortográficas.
- Toda modificación se registrará en un historial de versiones de requisitos como JIRA.

En este caso, se usa **vX.Y** porque:

- El sistema tiene una estructura clara y un número limitado de requisitos.

- El objetivo es gestionar cambios significativos o menores, sin entrar en complejidad técnica innecesaria.
- Es comprensible para todos los stakeholders del proyecto (clientes, desarrolladores, QA).

Ejemplo de versiones

Identificador	Versión	Responsable	Descripción del Cambio	Fecha
RF01-v1.0	1.0	Marcelo Acuña	Versión inicial: Gestión de medicamentos con edición, eliminación y personalización.	01/08/2025
RF01-v1.1	1.1	Abner Arboleda	Se aclara que los íconos pueden ser emojis o imágenes personalizadas.	10/08/2025
RF02-v1.0	1.0	Marcelo Acuña	Versión inicial: Recordatorios automáticos con vibración, sonido y persistencia.	01/08/2025
RF02-v2.0	2.0	Christian Bonifaz	Se agrega opción de repetir notificación si no se responde en 5 minutos.	12/08/2025
RF03-v1.0	1.0	Abner Arboleda	Versión inicial: Generación de historial y reporte PDF compartible.	01/08/2025
RF03-v1.1	1.1	Marcelo Acuña	Se incluye opción de exportar a Excel además del PDF.	15/08/2025
RF04-v1.0	1.0	Christian Bonifaz	Versión inicial: Registro de síntomas vinculados a medicamentos activos.	01/08/2025
RF04-v1.1	1.1	Abner Arboleda	Se añaden íconos ilustrativos para tipo de síntoma.	18/08/2025
RF05-v1.0	1.0	Marcelo Acuña	Versión inicial: Gestión de múltiples perfiles y datos separados.	01/08/2025
RF06-v1.0	1.0	Christian Bonifaz	Versión inicial: Gestión de citas médicas con recordatorios.	01/08/2025
RF06-v1.1	1.1	Abner Arboleda	Se incluye integración con el calendario del sistema operativo.	11/08/2025
RF07-v1.0	1.0	Christian Bonifaz	Versión inicial: OCR para escanear recetas médicas.	01/08/2025
RF07-v2.0	2.0	Christian Bonifaz	Se mejora la validación OCR usando IA para distinguir nombres similares.	20/08/2025
RF08-v1.0	1.0	Marcelo Acuña	Versión inicial: Interfaz en español e inglés con cambio dinámico.	01/08/2025
RF08-v1.1	1.1	Abner Arboleda	Se agrega soporte para francés en futuras versiones.	22/08/2025

VI. Gestión de Solicitudes de Cambio

Flujo para proponer, evaluar y aprobar cambios a requisitos - específico

1. Propuesta del Cambio

- **¿Quién lo realiza?**
Cualquier parte interesada: analista, desarrollador, tester, cliente, QA, etc.
- **¿Cómo se hace?**
Se llena un **formulario de solicitud de cambio (CR)** con toda la información requerida (ID, requisito afectado, descripción, justificación, impacto, prioridad, etc.).
- **¿Dónde se entrega?**
Se envía al **Ingeniero de Requisitos** o **Analista Funcional** responsable del módulo.
- **Herramienta o formato:**
 - Documento Word/Excel compartido.

2. Registro y Clasificación de la CR

- **¿Quién lo hace?**
Ingeniero de Requisitos o Administrador de cambios
- **¿Qué se hace?**
 - Registra la solicitud con su ID único.
 - Clasifica la CR como:
 - **Correctiva** (error en requisito),
 - **Adaptativa** (cambio por contexto o necesidad), o
 - **Excepcional** (urgente e inaplazable).
- **¿Dónde queda registrada?**
En el **repositorio central de control de cambios** del proyecto.

3. Análisis de Impacto

- **¿Quién lo hace?**
Un equipo técnico liderado por el **Ingeniero de Requisitos**, con apoyo del **Desarrollador Líder, QA y Gerente de Proyecto**.
- **¿Qué se evalúa?**
 - Impacto en el tiempo, costo, cronograma y riesgos.
 - Cambios en la arquitectura o en otros requisitos.
 - Esfuerzo estimado para aplicar el cambio.
- **¿Qué se produce?**
 - Informe breve de análisis de impacto.
 - Actualización del estado de la CR a: **"En análisis de impacto"**.

4. Evaluación de la CR por la Junta de Control de Cambios (CCB)

- **¿Quién participa?**
Junta de Control de Cambios (CCB), compuesta por:
 - Gerente de Proyecto
 - Ingeniero de Requisitos
 - Representante del cliente
 - QA
 - Desarrollador líder
 - Arquitecto de software (opcional)
 - Gerente de producto o funcional (si aplica)
- **¿Qué se hace?**
 - Se revisa el análisis de impacto.

- Se discute la necesidad real del cambio.
- Se decide si la CR es:
 - **Aprobada**
 - **Rechazada**
 - **Revisada para más información**
- **¿Cómo se comunica?**
El **Ingeniero de Requisitos** actualiza el estado de la CR y notifica la decisión al solicitante.

5. Planificación e Implementación del Cambio (si fue aprobado)

- **¿Quién lo hace?**
 - El **Gerente de Proyecto** lo incluye en el próximo sprint o fase.
 - Se asigna a un **Desarrollador responsable**.
 - QA planifica las pruebas necesarias.
- **¿Qué se hace?**
 - Se modifica el requisito original (por ejemplo, RF02-v1.0 → RF02-v2.0).
 - Se actualizan todos los documentos afectados.
 - Se registra la nueva versión del requisito.

6. Verificación y Cierre

- **¿Quién lo hace?**
 - **QA o Validador de cambios** designado por la CCB.
 - Validación puede ser funcional, documental o técnica.
- **¿Qué se hace?**
 - Se confirma que el cambio se implementó correctamente.
 - Se cierra oficialmente la CR y se archiva.

Formato de Solicitud de Cambio:

https://docs.google.com/document/d/18oGJd3w9jEZ24qQCAuMjFTdVT84O_uF-/edit?usp=sharing&oid=113799243958115587090&rtpof=true&sd=true

Sección	Campo	Responsable
1. Información General	ID del Cambio (CR-xxx)	Ingeniero de Requisitos
	Requisito Afectado (ej. RF02)	Solicitante
	Versión Actual del Requisito	Ingeniero de Requisitos
	Título del Cambio	Solicitante
	Fecha de Solicitud	Solicitante
	Solicitante (Nombre y Rol)	Solicitante
2. Descripción del Cambio	Detalle completo del cambio	Solicitante
3. Justificación del Cambio	Razón por la cual se solicita	Solicitante
4. Clasificación	Correctivo / Adaptativo / Excepcional	Ingeniero de Requisitos
5. Análisis de Impacto	Impacto en tiempo, costo, alcance, calidad	Ingeniero de Requisitos + Equipo Técnico
6. Evaluación del CCB	Estado del Cambio	Junta de Control de Cambios (CCB)
	Prioridad (Alta / Media / Baja)	Junta de Control de Cambios (CCB)
	Fecha de Evaluación	Junta de Control de Cambios (CCB)
	Responsable de Implementación	Gerente de Proyecto
7. Verificación y Cierre	Validación del cambio implementado	QA o Validador designado por la CCB
	Versión Nueva del Requisito (ej. RF02-v2.0)	Ingeniero de Requisitos
	Fecha de Cierre	Ingeniero de Requisitos

Ejemplo de llenado de la solicitud de cambio:

<https://docs.google.com/document/d/1aaLdCwBYkgK4xF47Dlyhd-jYHwltl4d5/edit?usp=sharing&oid=113799243958115587090&rtpof=true&sd=true>

VII. Herramienta de Soporte

- a. **Seleccionar una herramienta de gestión de requisitos (ejemplo: Jira, ReqView, Trello, ClickUp, IBM DOORS, etc.).**

Elegimos Jira ya que es una herramienta de gestión de proyectos desarrollada por Atlassian, ampliamente utilizada en entornos de desarrollo de software, aunque fue inicialmente diseñada para el seguimiento de incidencias, con el tiempo se ha consolidado como una solución completa para la

gestión de requisitos, especialmente bajo metodologías ágiles, su alta capacidad de personalización y control de trazabilidad la hacen ideal para proyectos que requieren seguimiento detallado, control de versiones, priorización y gestión colaborativa de los requisitos a lo largo del ciclo de vida del proyecto.

Característica	Jira	Trello	ClickUp
Orientación principal	Desarrollo de software / gestión ágil	Gestión de tareas general	Gestión de proyectos todo-en-uno
Gestión de estados personalizada	Sí, con flujos de trabajo configurables	Limitado (listas y etiquetas)	Sí, altamente configurable
Trazabilidad	Alta (enlaces entre tareas, dependencias)	Baja (no diseñada para trazabilidad)	Media (depende de la configuración)
Control de versiones	Sí, con versiones de producto	No nativo	Parcial (mediante documentos y tareas)
Priorización de requisitos	Avanzada (campos personalizados, filtros)	Básica (orden manual o etiquetas)	Media (listas, prioridades y etiquetas)
Visualización del flujo	Paneles Kanban, Scrum y reportes	Tableros Kanban simples	Tableros, diagramas y vistas múltiples
Escalabilidad en proyectos grandes	Muy alta	Baja	Media-alta
Colaboración y comentarios	Sí, con historial y seguimiento completo	Sí, básico	Sí, con más funciones de colaboración
Facilidad de uso para principiantes	Media (requiere curva de aprendizaje)	Alta (muy intuitiva)	Media (más funciones, requiere adaptación)
Ideal para	Equipos de desarrollo con procesos formales	Equipos pequeños, tareas simples	Equipos diversos con tareas variadas

Entre estas tres herramientas, Jira destaca como la más adecuada para la gestión de requisitos en proyectos de software, especialmente cuando se requiere trazabilidad, control de versiones, flujos de trabajo personalizados y una estructura organizada de priorización, aunque Trello y ClickUp pueden ser útiles en ciertos contextos, su enfoque generalista o limitado las hace menos apropiadas para un entorno de ingeniería de requisitos formal.

b. Justificar la elección, explicando sus principales funciones útiles para el PGR (gestión de estados, trazabilidad, versionado, priorización, visualización).

La elección de Jira como herramienta de gestión de requisitos se basa en su robustez, flexibilidad y alto nivel de personalización, lo que la convierte en una opción ideal para gestionar de manera integral los requisitos a lo largo del ciclo de vida de un proyecto de software, su uso permite mantener un control claro y estructurado de cada requisito, desde su creación hasta su implementación, facilitando también la colaboración del equipo.

Una de las funciones clave de Jira es la gestión de estados, esta característica permite definir flujos de trabajo personalizados, adaptados a los procesos específicos del equipo o proyecto, por ejemplo, se pueden establecer estados como "Propuesto", "En revisión", "Aprobado" o "Implementado", lo que permite saber exactamente en qué etapa del ciclo de vida se encuentra cada requisito.

Otra funcionalidad esencial es la trazabilidad. Jira permite vincular cada requisito con tareas técnicas, pruebas, bugs o incluso otros requisitos, creando así un mapa relacional que facilita el análisis de impacto y asegura que todas las necesidades del sistema estén cubiertas, esta trazabilidad es especialmente útil para validar que los requisitos han sido correctamente implementados y para identificar rápidamente el origen de un problema si surge alguna falla.

En cuanto al versionado, Jira permite asociar requisitos a versiones específicas del producto, lo cual resulta útil para organizar las entregas incrementales, además, al permitir registrar comentarios, fechas y cambios sobre cada ítem, se conserva un historial completo que documenta la evolución de cada requisito.

Respecto a la priorización, Jira permite agregar campos personalizados, etiquetas y filtros que ayudan a ordenar los requisitos según su nivel de importancia, urgencia o valor para el cliente, también es posible aplicar técnicas como MoSCoW o matrices de priorización mediante configuraciones internas o plugins externos.

Finalmente, Jira destaca por su capacidad de visualización, sus paneles interactivos (Kanban o Scrum), reportes automáticos y gráficos permiten ver el estado global del proyecto en tiempo real, lo que mejora la toma de decisiones y la comunicación dentro del equipo, además, la interfaz intuitiva facilita el uso incluso por parte de miembros que no son técnicos.

c. Incluir una captura de pantalla con un ejemplo de uso.

The image displays two screenshots of the Jira interface for a project named 'HEALTHY'.

Top Screenshot: Scrum Board

- Header:** 'HEALTHY' project name, search bar, and user 'MV'.
- Board:** 'Tablero Sprint 1' (6 ago - 3 sep) with 8 activities. The 'EN CURSO' column contains 8 items:
 - HEAL-1 RF01 - Gestión de Medicamentos
 - HEAL-2 RF02 - Recordatorios de Medicación
 - HEAL-3 RF03 - Informes del Historial de Medicamentos
 - HEAL-4 RF04 - Seguimiento de Síntomas o Efectos Secundarios
 - HEAL-5 RF05 - Gestión de Perfiles Múltiples
 - HEAL-6 RF06 - Citas Médicas
 - HEAL-7 RF07 - Reconocimiento de Recetas
 - HEAL-8 RF08 - Configuración de Idioma
- Right Panel:** Details for 'RF01 - Gestión de Medicamentos'. Description: 'El sistema permitirá registrar, editar y eliminar medicamentos sin restricciones de cantidad.'
 - El usuario podrá ingresar datos como nombre del medicamento, presentación (pastilla, jarabe, etc.), dosis, frecuencia, horario, duración y notas adicionales.
 - Los medicamentos podrán personalizarse con iconos y configuraciones de recordatorio.

Bottom Screenshot: Kanban Board

- Header:** 'HEALTHY' project name, search bar, user 'MV', and a 'Filtro' button.
- Board:** Three columns: 'POR HACER' (7 items), 'EN CURSO' (1 item), and 'LISTO' (1 item).
 - POR HACER:** RF02 - Recordatorios de Medicación (HEAL-2), RF03 - Informes del Historial de Medicamentos (HEAL-3), RF04 - Seguimiento de Síntomas o Efectos Secundarios (HEAL-4), RF05 - Gestión de Perfiles Múltiples (HEAL-5).
 - EN CURSO:** RF01 - Gestión de Medicamentos (HEAL-1).
 - LISTO:** Empty column with a '+ Crear' button.
- Right Panel:** Details for 'RF02 - Recordatorios de Medicación'. Description: 'El sistema generará alertas automáticas para cada toma de medicamento programada.'
 - Las alertas incluirán sonido, vibración y mensaje en pantalla, incluso en la pantalla de bloqueo.
 - El usuario podrá marcar la dosis como tomada, posponerla o indicar que fue omitida y registrar el motivo.
 - Las notificaciones persistirán hasta que se realice una acción.

VIII. Conclusiones

- El desarrollo del proyecto Healthy+, orientado a la asistencia de adultos mayores en la administración de medicamentos, demuestra una aplicación adecuada y estructurada de los principios de ingeniería de requisitos, desde la definición y clasificación de los atributos de cada requisito hasta el versionado y control de cambios, el equipo ha implementado prácticas sólidas que garantizan trazabilidad, claridad y control durante todo el ciclo de vida del sistema. La inclusión de atributos como estabilidad, riesgo, esfuerzo y fuente aporta una visión más completa y facilita la toma de decisiones tanto técnicas como de negocio.
- El proceso de priorización mediante la Matriz de Wiegers ha resultado especialmente valioso para el contexto de una aplicación de salud, donde los criterios de beneficio y perjuicio adquieren mayor relevancia que los factores puramente económicos. Los resultados obtenidos confirman que los requisitos fundamentales del sistema (RF01 - Gestión de medicamentos y RF02 - Recordatorios) ocupan las posiciones prioritarias, mientras que funcionalidades más complejas técnicamente como el reconocimiento OCR (RF07) quedan relegadas debido a su alto costo y riesgo de implementación. Esta priorización refleja una comprensión clara de las necesidades críticas del usuario final, especialmente considerando que el sistema está orientado a adultos mayores.
- La implementación del control de versiones semántico (vX.Y) proporciona un mecanismo eficiente para gestionar la evolución de los requisitos, permitiendo distinguir entre cambios significativos que afectan la funcionalidad (versión mayor) y ajustes menores de redacción o aclaración (versión menor), este sistema ha demostrado su utilidad práctica en el registro histórico de modificaciones, facilitando la comprensión de la evolución de cada requisito y manteniendo la consistencia documental del proyecto.
- El flujo de gestión de solicitudes de cambio establecido garantiza un proceso estructurado y controlado para la evaluación y aprobación de modificaciones, involucrando a todos los stakeholders relevantes a través de la Junta de Control de Cambios (CCB). Este mecanismo asegura que cualquier modificación sea analizada desde múltiples perspectivas (técnica,

funcional, económica y de riesgos) antes de su implementación, reduciendo significativamente la probabilidad de cambios inconsistentes o que generen impactos negativos no previstos.

- La selección de Jira como herramienta de soporte se justifica plenamente por su capacidad de integrar todas las funcionalidades requeridas para una gestión profesional de requisitos: gestión de estados personalizables, trazabilidad completa, control de versiones, priorización avanzada y visualización intuitiva, su flexibilidad permite adaptar la herramienta a las necesidades específicas del proyecto, mientras que su robustez garantiza el manejo eficiente de la información en proyectos de mayor escala.

IX. Recomendaciones

- Para fortalecer aún más el proceso de ingeniería de requisitos, se recomienda incorporar la validación de requisitos con usuarios finales reales (adultos mayores o sus cuidadores), con el fin de ajustar funcionalidades que realmente respondan a sus necesidades diarias, la inclusión de pruebas de usabilidad tempranas también permitiría detectar posibles fricciones en el diseño y mejorar la experiencia de usuario final.
- Asimismo, sería conveniente implementar métricas cuantitativas que midan el cumplimiento de requisitos y el impacto de los cambios, con el fin de evaluar de forma continua la eficiencia del proceso de gestión, estas métricas pueden incluir porcentaje de requisitos aprobados versus propuestos, número de cambios correctivos versus adaptativos, o tiempos promedio de implementación de cambios.