MODULO 1 Análisis y especificación de requisitos

**Técnicas de análisis de requisitos**

El proceso de análisis de requisitos se centra en estudiar las necesidades de los usuarios para definir los requisitos del sistema, plasmados en un documento de especificación de requisitos que describe lo que el sistema debe hacer, sin detallar cómo hacerlo. Este proceso implica tanto análisis como síntesis de información.

\*\*Técnicas para entender y organizar los requisitos:\*\*

1. \*\*Priorización de requisitos:\*\*

- \*\*Objetivo:\*\* Maximizar el valor entregado a los clientes identificando y priorizando requisitos según complejidad, dependencias y retorno de inversión.

- \*\*Responsabilidad:\*\* Generalmente recae en los gerentes de proyecto o dueños de producto, con participación activa de los analistas de requisitos. Los analistas asesoran al cliente para evitar priorizar requisitos con dependencias no resueltas.

- \*\*Continuidad:\*\* La prioridad de los requisitos puede cambiar a lo largo del proyecto, lo cual requiere decisiones constantes que pueden generar inconformidades pero son esenciales para fundamentar el desarrollo del sistema.

2. \*\*Técnica de clasificación de lista:\*\*

- \*\*Método:\*\* Asignar un valor numérico a cada requisito, iniciando por el número 1 y continuando de forma sucesiva.

- \*\*Ventajas:\*\* Evita problemas al tener un único número 1, aporta claridad y evita confusiones.

- \*\*Requisito:\*\* Conocimiento profundo de todos los requisitos definidos y esfuerzo del equipo para situar cada uno correctamente.

3. \*\*Técnica de puntos de historia y valor del negocio:\*\*

- \*\*Método:\*\* Priorizar según esfuerzo y opinión de gerentes de proyecto, clientes y equipo de desarrollo. Se asigna un valor numérico al negocio y otro a los puntos de historia (esfuerzo).

- \*\*Problemas:\*\* Usar solo el valor del negocio puede ser problemático; se debe considerar también el esfuerzo requerido.

- \*\*Cálculo:\*\* Dividir los puntos de valor del negocio entre los puntos de historia para priorizar los requisitos más sencillos de resolver y de mayor interés para el cliente.

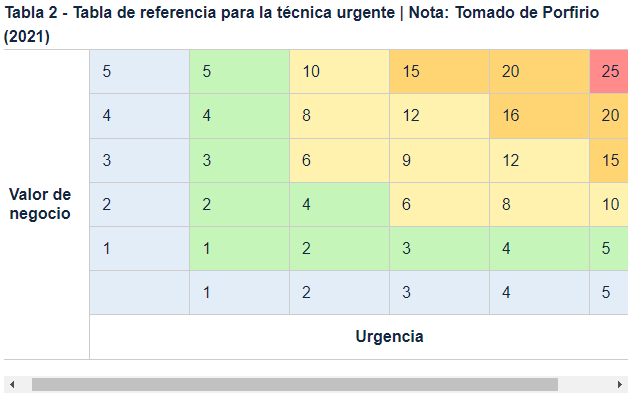
Estas técnicas permiten entender y ordenar los requisitos para su redacción formal, facilitando el desarrollo eficiente y enfocado en el valor del proyecto.



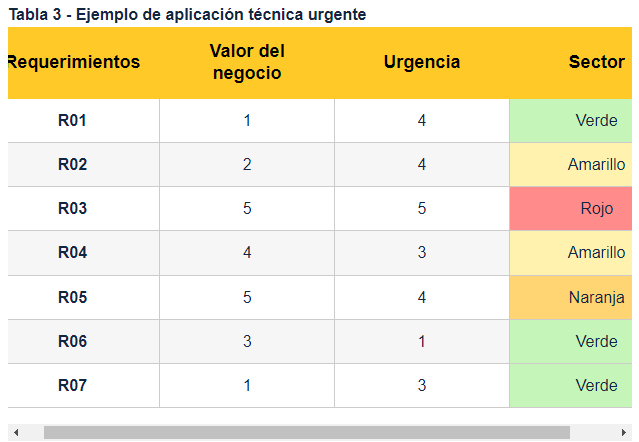
En un proyecto con siete requerimientos, se priorizan según el cociente entre valor de negocio y puntos de historia. El requerimiento R02 queda en primer lugar debido a su alto cociente, indicando que es fácil de construir y tiene un alto valor para el cliente. A continuación, siguen R01 y R05, ordenándose los demás en función de su cociente de mayor a menor.

**Técnica urgente:**

Se usa una tabla bidimensional donde la urgencia se valora de 1 a 5 (5 es la mayor urgencia) y el valor de negocio también de 1 a 5 (5 es el mayor valor). La prioridad final se determina mediante una escala de colores, resultante de multiplicar los valores de urgencia y valor de negocio.



Luego se consideran los requerimientos de mayor prioridad que están en el sector de color rojo, luego los de color naranja, luego los de color amarillo y, por último, los requerimientos del sector de color verde. Para entender mejor este estilo de priorización observar el siguiente ejemplo:



Al realizar la multiplicación de los valores de negocio y el valor de la urgencia se puede establecer en qué sector se encuentra cada requerimiento, tomando en cuenta los valores del ejemplo de la tabla anterior se puede concluir que el primer requerimiento a abordar sería el R03 que está en el sector de color rojo, luego el requerimiento R05 que está en el sector de color naranja y así sucesivamente.

1.1.4 Técnica MoSCoW

Esta técnica se basa en la asignación de etiquetas a cada requerimiento, y las disponibles se relacionan a continuación:



### Resumen

#### Proceso de priorización de requerimientos

Los requerimientos se priorizan en el siguiente orden:

1. \*\*Etiqueta M\*\*: Requerimientos obligatorios que, de no cumplirse, significarían el fracaso del proyecto.

2. \*\*Etiqueta S\*\*: Requerimientos importantes, pero no críticos.

3. \*\*Etiqueta C\*\*: Requerimientos deseables.

4. \*\*Etiqueta W\*\*: Requerimientos opcionales.

Este proceso requiere un consenso sobre el significado de cada etiqueta. Los requerimientos M deben ser útiles y coherentes; seleccionar todos como M diluiría la técnica, convirtiendo todos en obligatorios. En un desarrollo iterativo e incremental, un requerimiento W podría convertirse en M en futuras iteraciones.

#### Técnicas de análisis de requisitos

\*\*1. Juicio de expertos\*\*

- Basado en la opinión de un gerente de proyecto, dueño de producto, o stakeholder con conocimientos específicos.

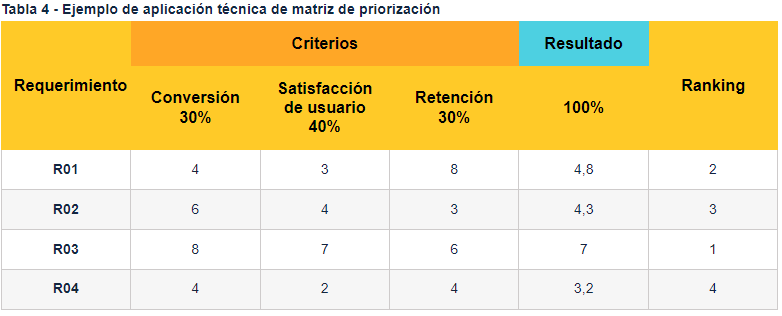
- Eficaz para proyectos pequeños con un dueño de negocio bien informado, pero puede ser sesgado.

\*\*2. Matriz de priorización\*\*

- Consiste en una tabla donde cada requerimiento se valora en una escala de 0 a 10 o 0 a 100, en función de diversas dimensiones alineadas con los objetivos del producto.

- Cada dimensión tiene un peso porcentual, y el valor final de cada requerimiento se obtiene sumando las multiplicaciones de cada puntuación por su peso.

Por ejemplo, se debe considerar la siguiente tabla:



\*\*Resumen\*\*

\*\*1. Evaluación de Requerimientos:\*\*

Los requerimientos (R01-R04) se evalúan en una escala de 0 a 10 en tres dimensiones: conversión (30%), satisfacción del cliente (40%) y retención de clientes (30%). Se calcula el valor porcentual para cada dimensión y se suman los valores parciales. Los requerimientos con mayores valores totales se priorizan.

\*\*2. Subjetividad de la Técnica:\*\*

Aunque la técnica tiene claros los criterios de evaluación, sigue siendo subjetiva, similar al juicio de expertos.

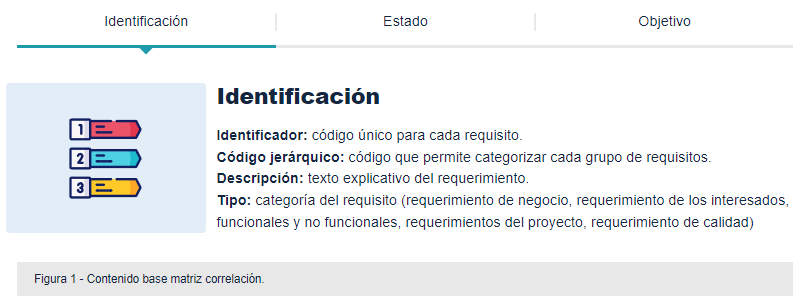
\*\*3. Matriz de Trazabilidad:\*\*

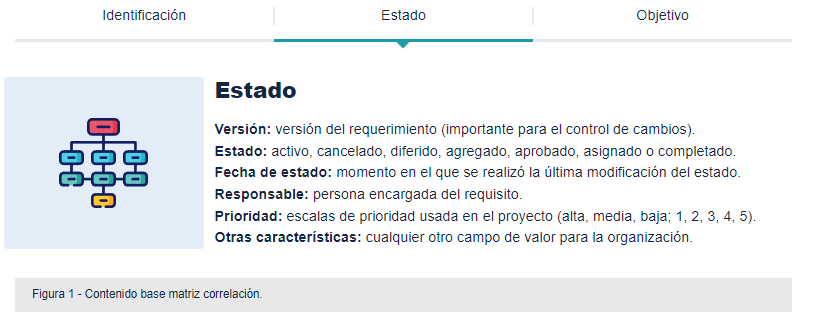
La matriz de trazabilidad alinea los requisitos del proyecto con los logros de los objetivos, relacionando cada requisito con el entregable correspondiente. Permite identificar los resultados alcanzados con cada requisito y visualizar qué requisitos son necesarios para cada entregable.

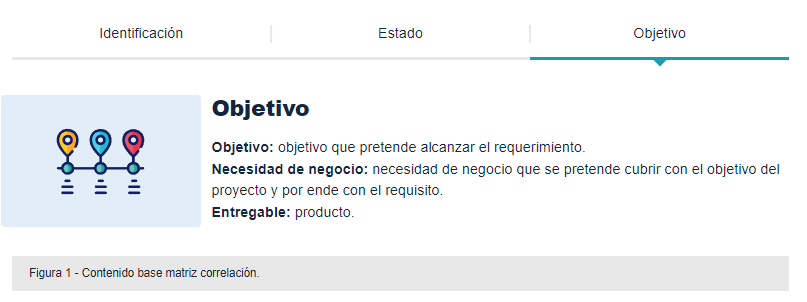
\*\*4. Proceso de Priorización:\*\*

Es crucial para el seguimiento y control de cambios, permitiendo modificar, eliminar o añadir requerimientos según sea necesario. Ayuda a identificar inconsistencias, visualizar rápidamente el estado de los requisitos y planificar las tareas pendientes del proyecto. La matriz puede construirse en una hoja de cálculo, relacionando requisitos y metas con valores complementarios que aportan coherencia.

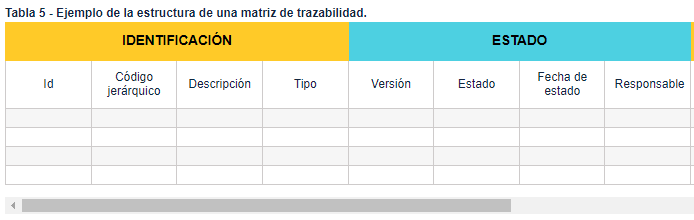
Cada organización es responsable de adaptar la matriz de correlación a sus necesidades particulares; en la siguiente figura se propone el contenido base de una matriz de correlación dividida por secciones.

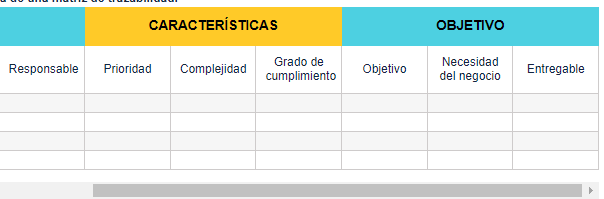






A continuación, se muestra el ejemplo de la estructura en Excel de una matriz de trazabilidad.



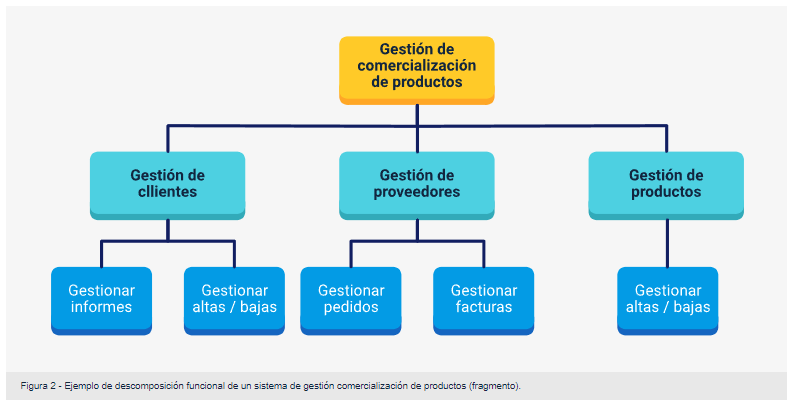


Para construir una matriz de trazabilidad, es esencial utilizar los campos que sean útiles para el proyecto específico, ya que no todos los proyectos son iguales y lo que funciona para uno puede no ser adecuado para otro. Esta matriz debe mantenerse actualizada durante todo el ciclo de vida del proyecto.

\*\*Descomposición funcional\*\*

Esta estrategia implica definir los requerimientos como una relación de entradas y salidas de un sistema. Inicialmente, estos requerimientos se establecen de manera general y poco detallada, y luego se descomponen en funcionalidades y subfuncionalidades más detalladas. Este enfoque, conocido como top-down, produce una estructura jerárquica (Pantaleo, 2018).

El objetivo de la descomposición funcional es identificar y comprender los componentes o partes que forman un todo, prestando especial atención a las interacciones entre estos componentes.



**2**

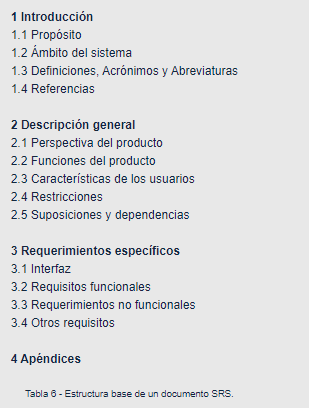
**Especificación de requisitos**

En esta sección se describen algunos estándares y/o técnicas que pueden ser usadas por las organizaciones para describir formalmente cada uno de los requisitos del sistema.

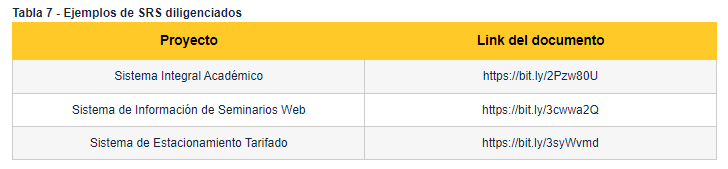
**2.1 Estándar IEEE 830**

Este estándar presenta un conjunto de prácticas recomendadas para la redacción de un documento de especificación de requerimientos mejor conocido como SRS. Este documento está dividido en secciones y cada una de ellas aborda aspectos particulares. A continuación, se describirá de forma general algunos de los elementos que conforman este documento (IEEE 830-1998).

La siguiente tabla muestra la estructura base de un documento SRS, indicando cuáles son los apartados principales.



Ahora se deben revisar algunos ejemplos que se presentan sobre el diligenciamiento del formato SRS:



### Resumen del Estándar IEEE 29148:2018

El estándar IEEE 29148:2018 reemplaza a los estándares IEEE 830, IEEE 1233 y IEEE 1362, y proporciona directrices para los procesos y productos relacionados con la ingeniería de requisitos para sistemas, productos y servicios de software a lo largo de su ciclo de vida. Este estándar define cómo construir buenos requisitos, sus atributos y características, y su aplicación iterativa y recursiva a lo largo del ciclo de vida del proyecto. Además, proporciona orientación sobre la gestión de los requisitos y define los elementos de información aplicables a la ingeniería de requisitos y su contenido.

#### Estructura del Estándar IEEE 29148:2018

1. \*\*Introducción, resumen y tabla de contenido\*\*: Proporciona una visión general del estándar.

2. \*\*Propósito y alcance\*\*: Define el objetivo y el alcance del estándar.

3. \*\*Explicación de los estándares relacionados\*\*: Detalla los otros estándares que conforman el IEEE 29148.

4. \*\*Referencias normativas\*\*: Enumera las normas que respaldan el estándar.

5. \*\*Clarificación de la terminología\*\*: Importante para establecer nuevos procesos de ingeniería de requisitos en una empresa.

6. \*\*Conceptos y procesos\*\*: Explica y define los conceptos y procesos involucrados.

7. \*\*Ítems de información\*\*: Describe los elementos de información necesarios para la especificación de requisitos.

8. \*\*Anexos adicionales\*\*: Proporciona detalles adicionales y específicos.

#### Tipos de Requerimientos del Sistema

1. \*\*Requerimientos funcionales\*\*: Necesidades de los interesados del software.

2. \*\*Requerimientos de usabilidad\*\*: Utilizados directamente por los involucrados en la solución.

3. \*\*Requerimientos de desempeño\*\*: Disponibilidad de servicios y procesos transaccionales.

4. \*\*Interfaces del sistema\*\*: Interacción entre personas y el software.

5. \*\*Operaciones del sistema\*\*: Describen cómo debe operar el sistema.

6. \*\*Modos y estados del sistema\*\*: Diferentes estados y modos de funcionamiento.

7. \*\*Características físicas\*\*: Requerimientos de hardware.

8. \*\*Condiciones del ambiente\*\*: Condiciones operativas y operacionales.

9. \*\*Seguridad del sistema\*\*: Requerimientos de seguridad.

10. \*\*Manejo de la información\*\*: Cómo se debe gestionar la información.

11. \*\*Políticas y regulación\*\*: Normas y estándares aplicables.

12. \*\*Ciclo de vida del sistema\*\*: Etapas y duración del desarrollo y uso en producción.

### Especificación de Requisitos a través de Marcos de Trabajo Ágiles

Los marcos de trabajo ágiles favorecen la comunicación oral sobre la documentación exhaustiva en la mayoría de los procesos del ciclo de vida, especialmente en la identificación de necesidades y diseño. Un artefacto clave en los marcos ágiles son las historias de usuario, que describen de manera general e informal una función del software desde la perspectiva del usuario final o cliente. Estas historias permiten describir brevemente un requerimiento, estimar prioridades, alcance y tiempo de realización.

En resumen, el estándar IEEE 29148:2018 proporciona una estructura integral para la gestión de requisitos a lo largo del ciclo de vida del software, mientras que los marcos de trabajo ágiles complementan esta estructura promoviendo una comunicación eficiente y centrada en el usuario.



Las historias de usuario tienen varios beneficios respecto a otros instrumentos de redacción de requerimientos, entre los cuales se pueden listar:

* Las historias de usuario se centran en solucionar problemas a usuarios reales.
* Las historias de usuario permiten la colaboración, ya que como su descripción es corta se necesita que el equipo colabore para decidir cómo dar solución a la historia para cumplir con la necesidad expresada por el usuario.
* Las historias impulsan la creatividad, ya que fomentan que el equipo piense de forma crítica y creativa sobre cómo solucionar de la mejor manera el objetivo.
* Las historias de usuario motivan, pensar en la mejor solución para una problemática particular representan retos y pequeñas victorias para el equipo.

Scrum es un marco de trabajo basado en un proceso de construcción iterativo e incremental que identifica tres roles principales:

1. **Equipo de trabajo (team):** Compuesto por desarrolladores, diseñadores, personal de calidad e infraestructura necesarios para construir el producto de software.
2. **Scrum Master:** Similar a un director de proyecto, pero más enfocado en asegurar que el equipo tenga todas las herramientas y recursos necesarios.
3. **Dueño del producto (product owner):** Representante del cliente y único responsable de la gestión de requisitos del proyecto.

El concepto de **sprint** se refiere a una iteración con tiempos fijos de 2 a 4 semanas. Cada sprint incluye:

* **Planeación:** Definición de los requisitos a desarrollar.
* **Construcción:** Desarrollo del producto.
* **Despliegue:** Demostración de lo construido en reuniones de revisión.

El **pila de producto (product backlog)** es una lista priorizada de todos los requisitos del sistema, descritos como historias de usuario. La priorización de estas historias, numeradas consecutivamente, es realizada exclusivamente por el dueño del producto basándose en el negocio y recomendaciones de expertos para maximizar el retorno de inversión. Esta priorización puede cambiar con el tiempo, pero solo el dueño del producto puede realizarla.

Scrum, siendo un marco de trabajo ágil, no requiere para su funcionamiento que todas las historias de usuario de la pila de producto estén detalladas, pero sí, por lo menos, las de mayor prioridad para poder iniciar el trabajo con el equipo en sus respectivos sprints (ScrumStudy, 2021).

Al inicio de cada sprint se realiza el proceso de planeación que involucra principalmente tres tareas:

**1**

Estimar el valor de esfuerzo requerido para un conjunto de historias de usuario de la pila de producto, trabajo que es realizado únicamente por el equipo de desarrollo.

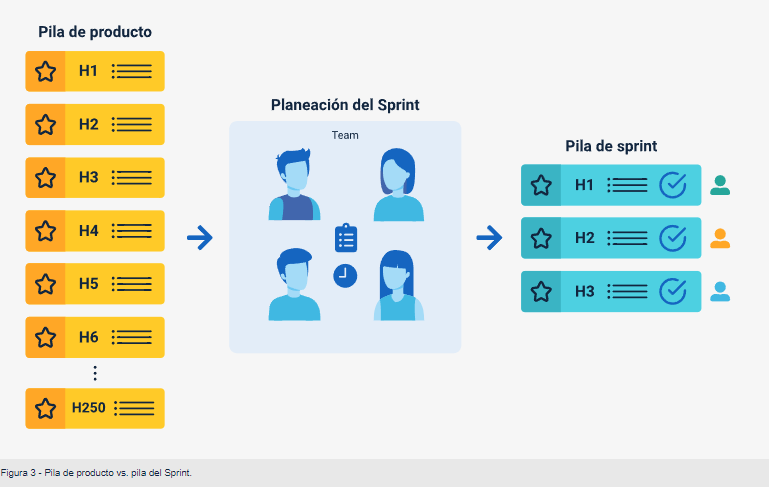
**2**

Selección de las historias d e usuario a desarrollar durante al sprint tomando como referencia la prioridad y el valor del esfuerzo asociado a cada historia. Esto genera un artefacto llamado pila del sprint (sprint backlog) que lista los requerimientos descritos como historias de usuario a ser realizadas y evaluadas en el sprint.

**3**

Descomposición de cada historia de usuario en tareas y, de ser necesario, asignar responsables a cada tarea.

A continuación, se expone una figura en la que se representan los artefactos generados dentro del marco de trabajo Scrum y que permiten la gestión de los requisitos y el evento desde el cual se construye inicialmente.



### Kanban y la Especificación de Requisitos

\*\*Orígenes de Kanban:\*\*

- Surge del Toyota Production System (TPS) a finales de los 1940s.

- Basado en la demanda del cliente, no en producción masiva.

- Fundamenta la producción ajustada, minimizando desperdicios y maximizando valor sin aumentar costos.

\*\*Adopción en la Industria del Software:\*\*

- A principios del siglo XXI, se adopta en software para mejorar la producción y entrega.

- Influenciado por metodologías ágiles, especialmente Scrum.

- Enfatiza la experimentación y mejora continua.

\*\*Principios Fundamentales de Kanban en Software:\*\*

1. \*\*Calidad garantizada:\*\* Prioriza la calidad sobre la rapidez.

2. \*\*Reducción del desperdicio:\*\* Hacer solo lo necesario, pero bien.

3. \*\*Mejora continua.\*\*

4. \*\*Flexibilidad:\*\* Permite priorizar tareas según necesidades actuales.

\*\*Prácticas de Kanban:\*\*

1. Visualizar el flujo de trabajo.

2. Eliminar interrupciones.

3. Gestionar el flujo.

4. Hacer políticas explícitas para visibilidad.

5. Circuitos de realimentación.

6. Mejorar colaborando.

\*\*Herramienta Principal: Tablero Kanban\*\*

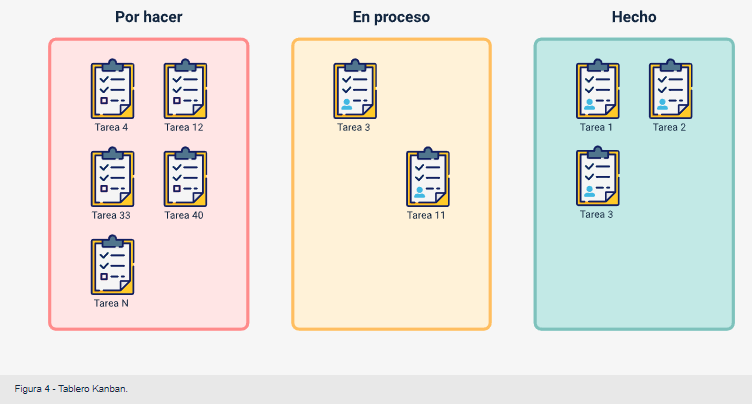
- Mapea y visualiza el flujo de trabajo.

- Dividido en columnas que representan fases del proceso.

- Filas representan diferentes tipos de actividades.

- Tres secciones típicas: por hacer, en proceso, hecho.

Cada equipo de trabajo puede realizar un mapeo más detallado de su proceso y agregar tanta sección como considere pertinente, como se muestra en la siguiente figura.



Dependiendo del marco de trabajo, varía la forma en la que se describen cada una de las tareas del tablero Kanban, por ejemplo, dentro de un marco de trabajo como Scrum cada una de las tareas se podría describir en el formato de historias de usuario.