**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Construcción de requisitos del *software* |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 3 |
| NOMBRE DEL COMPONENTE FORMATIVO | Análisis y especificación de requisitos |
| BREVE DESCRIPCIÓN | En este componente formativo se abordan el análisis de requisitos (priorización, descomposición funcional, matriz de trazabilidad) y estándares, y/o guías existentes para la especificación formal de los mismos dependiendo del tipo de marco de trabajo usado (tradición o ágil). |
| PALABRAS CLAVE | Ingeniería de requerimientos, priorización de requisitos, requisitos |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 220501092 - Construcción de Requisitos del *software*. | RESULTADOS DE APRENDIZAJE | 220501092-2  Realizar el informe de los requisitos del *software* según el análisis de la información recolectada. |

|  |  |
| --- | --- |
| ÁREA OCUPACIONAL | 2 - Ciencias naturales, aplicadas y relacionadas |
| IDIOMA | Español |

1. **TABLA DE CONTENIDOS:**

**Introducción**

1. **Técnicas de análisis de requisitos**
   1. Priorización de requisitos
      1. Técnica de clasificación de lista
      2. Técnica de puntos de historia y valor del negocio
      3. Técnica urgente
      4. Técnica MoSCoW
      5. Juicio de expertos
   2. Matrices de trazabilidad
   3. Descomposición funcional
2. **Especificación de requisitos**
   1. Estándar IEEE 830
   2. Estándar IEEE 29148:2018
   3. La especificación de requisitos a través de métodos y metodologías ágiles
   4. Scrum y la especificación de requisitos
   5. Kanbany la especificación de requisitos
3. **DESARROLLO DE CONTENIDOS:**

**Introducción**

En este componente formativo se estudiarán los principios básicos sobre los que se fundamenta el proceso de análisis de requisitos desde marcos de desarrollo de *software* tradicional y ágiles. También, introduce a diferentes formas (planillas y estándares) para realizar el proceso de documentación de requisitos.

1. **Técnicas de análisis de requisitos**

El proceso de análisis de requisitos permite, principalmente, el estudio de las necesidades de los usuarios para definir requisitos del sistema por medio de la producción del documento de especificación de requisitos donde se describe lo que el sistema debe hacer, pero no el cómo. Así es como este proceso, además de involucrar un proceso de análisis, también requiere de la síntesis de la información existente.

A continuación, se describen algunas técnicas que pueden ser utilizadas para entender y ordenar los requisitos identificados para su posterior redacción en un artefacto formal dependiendo del marco de desarrollo de *software* utilizado.

* 1. **Priorización de requisitos**

Esta es una actividad clave para el éxito en la construcción de producto de software y su objetivo es maximizar el valor entregado por el proyecto a sus clientes; es decir, identificar cuáles requisitos se deben priorizar tomando en cuenta factores como la complejidad, las dependencias y el retorno de inversión a los clientes, entre otros indicadores.

La priorización de requisitos, por lo general, es responsabilidad de los gerentes del proyecto o dueños de producto dependiendo del tipo de enfoque utilizado; sin embargo, se requiere de la participación activa de los analistas de requisitos, ya que si bien es el cliente o su representante quien determina cuáles requisitos son más importantes para su negocio, el analista debe asesorar, facilitar e informarle al cliente para no darle prioridad superior a requisitos que tienen otras dependencias no resueltas; por ejemplo, no se podría priorizar un requerimiento asociado a un carrito de compras sin antes tener resuelto los requerimientos asociados al registro y consulta de productos.

El proceso de priorización es continuo, es decir, la prioridad de un requisito puede variar a lo largo del proyecto, así como los requisitos también pueden variar a lo largo del mismo. La priorización no es un proceso sencillo, por lo que, además de conocer las diferentes técnicas, es importante la toma de decisiones y estas, en el marco de un proyecto, siempre van a generar inconformidades, pero si no se decide respecto a la priorización de requisitos es como si no se fundamentaran las bases sobre las que el equipo de desarrollo va a construir el sistema.

* + 1. **Técnica de clasificación de lista**

Esta técnica de clasificación no requiere ningún tipo de entrenamiento o preparación, ya que es una forma de priorización natural que se usa en la vida cotidiana. Este tipo de priorización simple es una de las más utilizadas y consiste en darle un valor numérico a cada requerimiento iniciando por el número 1 y continuando de forma sucesiva con 2, 3 y hasta el número total de requisitos definidos (Porfirio, 2021).

Esta técnica tiene la ventaja de que solo puede existir un número 1, lo cual evita muchos problemas por parte de los interesados o responsables del negocio que quieren que todos los requerimientos tengan prioridad 1; adicionalmente, aporta claridad y evita confusiones. Cada elemento se prioriza con relación al resto de elementos, lo que simplifica el proceso.

Para usar adecuadamente esta técnica se requiere de un profundo conocimiento de todos los requerimientos definidos y si bien el proceso parece simple, se requiere de un gran esfuerzo por parte del equipo para situar a cada requerimiento en la posición correcta.

* + 1. **Técnica de puntos de historia y valor del negocio**

Esta consiste en realizar el proceso de priorización de acuerdo con factores como el esfuerzo y la opinión de los gerentes de proyecto o dueños de producto, los clientes, el equipo de desarrollo o incluso una combinación de los anteriores. El valor del negocio corresponde a un valor numérico que cuanto más alto sea, más valor representa para el cliente (Porfirio, 2021).

Usar solo el valor del negocio como elemento de priorización puede generar problemas ya que el valor asociado por el cliente a un requerimiento puede ser muy superfluo y, además, no considera detalles clave como el esfuerzo que se requiere para su desarrollo, por esta razón también se usa el valor de puntos de historia, que no es más que otro valor numérico asignado por el equipo de desarrolladores a cada requerimiento donde expresa una estimación de esfuerzo, cuanto más grande sea el número implica más esfuerzo requerido con miras a realizar el requerimiento.

Para lograr la priorización de los requerimientos, se debe realizar el cálculo del cociente obtenido a partir de los puntos de valor del negocio dividido entre los puntos de historia, quedando entonces una priorización donde estarán en los primeros lugares los requerimientos más sencillos de resolver por los desarrolladores y que tengan mayor interés por parte del cliente. A continuación se da un ejemplo.

**Tabla 1**  
*Ejemplo de aplicación técnica de puntos de historia y valor del negocio*

|  |  |  |  |
| --- | --- | --- | --- |
| Requerimientos | Valor del negocio | Puntos de historia | Cociente |
| R01 | 6 | 3 | 2 |
| R02 | 7 | 1 | 7 |
| R03 | 9 | 5 | 1,8 |
| R04 | 3 | 3 | 1 |
| R05 | 4 | 2 | 2 |
| R06 | 5 | 4 | 1,25 |
| R07 | 2 | 8 | 0,25 |

Si se trata de un proyecto de siete (7) requerimientos con los valores de negocio, puntos de historia y cocientes como se describe en la tabla anterior, al realizar el proceso de priorización quedaría en el primer lugar el requerimiento R02 ya que tiene el cociente más alto; es decir, representa un requerimiento fácil de construir para el grupo de desarrolladores y, adicionalmente, tiene un alto valor de negocio para el cliente; luego, en el segundo lugar, el requerimiento R01 y R05 y así sucesivamente se registran las prioridades de acuerdo con el valor del cociente de mayor a menor.

* + 1. **Técnica urgente**

Aquí se utiliza una tabla de dos dimensiones, donde la horizontal estará determinada por el valor de la urgencia en el requerimiento, el cual corresponde a un valor numérico entre **1** y **5**, donde un valor de **5** implica la mayor urgencia y **1** que no hay tanto apuro en el desarrollo de requerimiento; y la dimensión vertical estará determinada por el valor del negocio, solo que esta vez, a diferencia de la técnica anterior, el valor del negocio también se rige por una escala de 1 a 5, siendo 5 el de mayor valor de negocio posible para un requerimiento (Porfirio, 2021).

Para determinar la prioridad final de un requerimiento, se utiliza una escala de colores que surge a partir de la multiplicación de los valores de las escalas de urgencia y de valor de negocio según la siguiente tabla:

**Tabla 2** *Tabla de referencia para la técnica urgente*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Valor de negocio | 5 | 5 | 10 | 15 | 20 | 25 |
| 4 | 4 | 8 | 12 | 16 | 20 |
| 3 | 3 | 6 | 9 | 12 | 15 |
| 2 | 2 | 4 | 6 | 8 | 10 |
| 1 | 1 | 2 | 3 | 4 | 5 |
|  | 1 | 2 | 3 | 4 | 5 |
| Urgencia | | | | | |

Nota: Tomado de Porfirio (2021).

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:

**Tabla 3** *Ejemplo de aplicación técnica urgente*

|  |  |  |  |
| --- | --- | --- | --- |
| Requerimientos | Valor del negocio | Urgencia | Sector |
| R01 | 1 | 4 | Verde |
| R02 | 2 | 4 | Amarillo |
| R03 | 5 | 5 | Rojo |
| R04 | 4 | 3 | Amarillo |
| R05 | 5 | 4 | Naranja |
| R06 | 3 | 1 | Verde |
| R07 | 1 | 3 | Verde |

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. **Técnica MoSCoW**

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

M: indica una funcionalidad que debe estar (Must).

S: indica una funcionalidad que debería estar (Should).

C: indica una funcionalidad que podría estar (Could).

W: indica una funcionalidad que no estará por ahora, de pronto más adelante (Wont).

Los requerimientos son priorizados utilizando el siguiente orden: primero los que tienen etiqueta **M**, luego los requerimientos con etiqueta **S**, después aquellos con **C** y, finalmente, los etiquetados como **W**.

Esta técnica requiere de un proceso de consenso sobre el significado de cada una de las etiquetas asignables a cada requerimiento. Los requisitos de tipo **M** son aquellos obligatorios y que, de no ser abordados, implicaría directamente el fracaso; es importante entonces acordar qué se puede entregar y que sea útil, adicionalmente deben formar parte de un conjunto coherente, ya que si, sencillamente se seleccionan todos los requerimientos de tipo **M**, automáticamente todos se transforman en requerimientos de tipo **M** y se pierde la dinámica de la técnica.

Normalmente el proceso de desarrollo de *software* es iterativo e incremental por lo que dependiendo del momento en que se encuentre un requerimiento, que ahora puede ser **W**, en la siguiente iteración puede asumir un valor de **M**.

* + 1. **Juicio de expertos**

Es una técnica basada principalmente en función de la complejidad y la exactitud en los resultados, básicamente consiste en realizar el proceso de priorización utilizando como base la opinión del gerente de proyecto o dueño de producto, o de algún *stakeholder* con conocimientos en la industria asociada al producto a desarrollar (Porfirio, 2021).

Este tipo de técnica funciona muy bien para proyectos pequeños con un dueño de negocio que tiene mucho conocimiento del problema y de la solución que necesita para resolverlo; sin embargo, se debe utilizar esta técnica con cuidado ya que se basa en una visión sesgada del negocio.

* + 1. **Matriz de priorización**

Esta técnica consiste en construir una tabla donde cada requerimiento es valorado en una escala de **0** a **10** o de **0** a **100** para cada dimensión alineada con los objetivos del producto. Normalmente cada dimensión tiene un peso porcentual, de modo que cada requerimiento tendrá un valor final a partir de la sumatoria de la multiplicación de cada puntuación por el peso de cada dimensión (Porfirio, 2021).

Por ejemplo, se debe considerar la siguiente tabla:

**Tabla 4** *Ejemplo de aplicación técnica de matriz de priorización*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Requerimiento | Criterios | | | Resultado | Ranking |
| Conversión  30% | Satisfacción de usuario 40% | Retención 30% | 100% |
| R01 | 4 | 3 | 8 | 4,8 | 2 |
| R02 | 6 | 4 | 3 | 4,3 | 3 |
| R03 | 8 | 7 | 6 | 7 | 1 |
| R04 | 4 | 2 | 4 | 3,2 | 4 |

Suponiendo las dimensiones de conversión con un peso porcentual del 30%, la satisfacción del cliente con un peso porcentual del 40% y la retención de clientes con peso porcentual del 30%, cada uno de los requerimientos (R01-R04) son evaluados en una escala de 0 a 10, a los cuales se les aplica el cálculo de acuerdo con el valor porcentual de cada dimensión evaluada y el resultado se obtendría de la sumatoria de los valores parciales de cada dimensión, con lo cual los mayores valores totales obtenidos serán priorizados sobre los de menor valor.

Esta técnica sigue siendo muy subjetiva al igual que la técnica de juicio de expertos, aunque quedan claros cuáles son los criterios de evaluación utilizados para determinar la prioridad de los requisitos.

* 1. **Matriz de trazabilidad**



Este es un instrumento clave para el responsable del proyecto, en especial en el proceso de seguimiento y control de cambios, ya que permite analizar qué requerimientos, eventualmente, convendría modificar, eliminar o añadir; así, se puede ajustar la planificación de tareas pendientes del proyecto. También permite identificar inconsistencias entre los requerimientos y los beneficios que se esperan alcanzar, tener una visualización rápida de los requisitos que han sido abordados y cuáles están pendientes por realizar y, por lo tanto, qué entregables están próximos a producirse y cuáles todavía requieren de mayor tiempo.

La estructura de la matriz de trazabilidad se puede construir en una hoja de cálculo, en ella se relacionan todos los requisitos y las metas a alcanzar junto con una serie de valores complementarios que aportan información y coherencia a los vínculos generados.



Fuente. <https://image.freepik.com/vector-gratis/ilustracion-concepto-hoja-calculo_114360-945.jpg>

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.

**Figura 1***Contenido base matriz correlación*



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

**Tabla 5.** *Ejemplo de la estructura de una matriz de trazabilidad*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IDENTIFICACIÓN | | | | ESTADO | | | | CARACTERÍSTICAS | | | OBJETIVO | | |
| Id | Código jerárquico | Descripción | Tipo | Versión | Estado | Fecha de estado | Responsable | Prioridad | Complejidad | Grado de cumplimiento | Objetivo | Necesidad del negocio | Entregable |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Nota: Tomado de Pantaleo (2018).

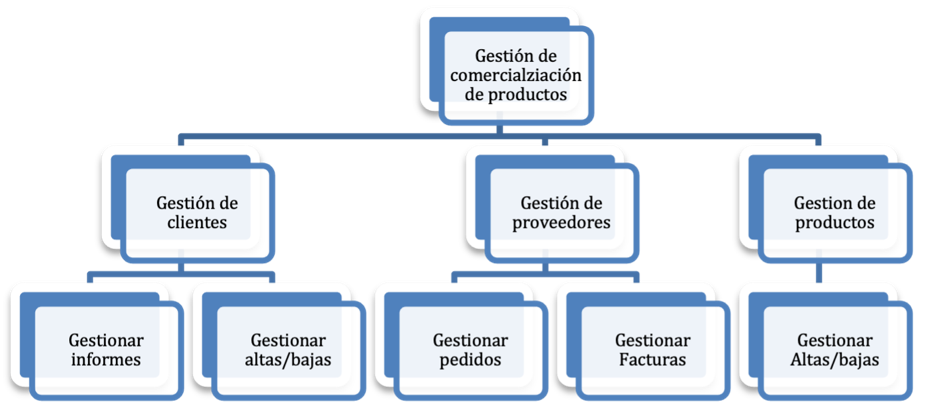
Al construir una matriz de trazabilidad se deben usar los campos que se consideren útiles para el proyecto, pues no todos los proyectos son iguales y la estructura definida para uno puede no resultar conveniente para otro proyecto. **Cuando se usa, esta matriz debe permanecer actualizada a lo largo del ciclo de vida de construcción del proyecto**.

* 1. **Descomposición funcional**

Esta estrategia consiste en definir los requerimientos como una relación de entradas y salidas de un sistema. Normalmente, estos requerimientos se definen de manera muy general y poco detallada y luego empieza a descomponerse en funcionalidades y subfuncionalidades un poco más detallados con el ánimo de analizarlas individualmente hasta lograr un nivel de detalle adecuado para el proyecto, a esta estructura se le conoce como *top-down*; el resultado es una estructura jerárquica (Pantaleo, 2018).

La descomposición funcional se realiza, por lo general, para identificar y entender los componentes o partes que constituyen un todo; en este proceso es vital identificar las interacciones entre componentes. A continuación, se expone un ejemplo gráfico sobre esta descomposición funcional.

**Figura 2***Ejemplo de descomposición funcional de un sistema de gestión comercialización de productos (fragmento)*



1. **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.

* 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).

En primer lugar, se encuentra la sección de introducción que tiene los siguientes elementos:

* Propósito: indica a qué parte del público va dirigido el documento y cuál es su propósito.
* Ámbito del sistema: se explica el nombre asignado al proyecto, su funcionamiento, beneficios percibidos y objetivos alcanzables con el proyecto; también indica qué está por fuera del alcance.
* Definiciones, acrónimos y abreviaturas: se especifica el significado de algunas palabras que pueden ser difíciles de entender y cuáles han sido abreviadas.
* Referencias: se indican todas las fuentes externas utilizadas para la construcción del documento.

En segundo lugar, está la sección de descripción general sobre los factores generales que afectan al producto y sus requerimientos. Esta sesión se subdivide en:

* Perspectiva del producto: pone el producto en perspectiva con otros productos o proyectos relacionados.
* Funciones del producto: resumen de las funciones que el *software* realizará.
* Características de los usuarios: descripción del usuario del producto incluyendo nivel académico, experiencia en el negocio y experiencia técnica.
* Restricciones generales: descripción general de cualquier ítem que limita las opciones de los desarrolladores en el diseño del sistema.
* Suposiciones y dependencias: se lista cada uno de los factores que afectan los requerimientos establecidos en el SRS.

La tercera sección es la especificación de requerimientos, la cual es la sección más larga e importante del documento, pues aquí se detallan cada uno de los requerimientos que serán la base para guiar los procesos de diseño, implementación y pruebas del *software*. Cada requerimiento descrito en esta sección debe cumplir con los siguientes criterios de calidad: corrección, rastreable, no ambiguo, verificable y priorizado.

Esta sección se subdivide para presentar primero los requerimientos funcionales y luego los no funcionales; los requerimientos pueden ser especificados usando el formato de casos de uso para lo cual existe una subsección específica.

En la cuarta sección se listan todos los modelos de análisis usados en el desarrollo específico de los requerimientos listados en la sección anterior, es de anotar que cada modelo va acompañado de una introducción y descripción narrativa. Los modelos propuestos por el estándar son los diagramas de secuencia, el diagrama de flujo de datos y el diagrama de transición de estados.

Finalmente, la quinta sección del documento se centra en identificar y describir el proceso que debe ser usado para la actualización del documento, define quién puede enviar cambios, por cuáles medios y cómo pueden ser aprobados estos cambios.

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

**Tabla 6** *Estructura base de un documento SRS*

|  |
| --- |
| 1. Introducción    1. Propósito    2. Ámbito del sistema    3. Definiciones, Acrónimos y Abreviaturas    4. Referencias 2. Descripción general    1. Perspectiva del producto    2. Funciones del producto    3. Características de los usuarios    4. Restricciones    5. Suposiciones y dependencias 3. Requerimientos específicos    1. Interfaz    2. Requisitos funcionales    3. Requerimientos no funcionales    4. Otros requisitos 4. Apéndices |

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

**Tabla 7** *Ejemplos de SRS diligenciados*

|  |  |
| --- | --- |
| Proyecto | Link del documento |
| Sistema Integral Académico | <https://bit.ly/2Pzw80U> |
| Sistema de Información de Seminarios Web | <https://bit.ly/3cwwa2Q> |
| Sistema de Estacionamiento Tarifado | <https://bit.ly/3syWvmd> |

* 1. **Estándar IEEE 29148:2018**

Este estándar reemplaza los estándares IEEE 830, IEE 1233, IEEE 1362, y contiene disposiciones para los procesos y productos relacionados con la ingeniería de requisitos para sistemas, productos y servicios de *software* a lo largo del ciclo de vida (Penzenstadler, 2021).

Además, define la construcción de un buen requisito, proporciona atributos y características de los requisitos, y analiza la aplicación iterativa y recursiva del proceso de requisitos a lo largo del ciclo de vida. También proporciona orientación adicional en la aplicación de procesos de ingeniería y gestión de requerimientos relacionados con la ingeniería de requisitos al tiempo que define los elementos de información aplicables a la ingeniería de requisitos y su contenido

El estándar IEEE 29148:2018 está estructurado de la siguiente forma:

* Introducción, resumen y tabla de contenido.
* Propósito, alcance del estándar, generalidades.
* Explicación de los otros estándares que lo conforman.
* Referencias a normas que lo conforman.
* Clarificación de la terminología, lo que es muy valioso para cuando se quiere establecer nuevos. procesos de ingeniería de requerimientos en una empresa.
* Clarificación de los conceptos y procesos.
* Explicación y contenido de los ítems de información que vienen a través de la especificación de requerimientos o que debemos considerar incluir en la especificación de requerimientos.
* Anexos adicionales para mayor detalle.

Esta norma propone un listado de requerimientos mínimos los cuales son la base de la especificación de requerimientos; en ese sentido se proponen los siguientes tipos de requerimientos del sistema:

* Requerimientos funcionales: representan necesidades de los interesados del *software.*
* Requerimientos de usabilidad: requerimientos que son utilizados directamente por los involucrados en la solución (requerimientos de uso).
* Requerimientos de desempeño: disponibilidad de servicios y procesos transaccionales.
* Interfaces del sistema: interacción entre personas con el *software*.
* Operaciones del sistema.
* Modos y estados del sistema.
* Características físicas (*hardware*).
* Condiciones del ambiente (operativas y operacionales).
* Seguridad del sistema.
* Manejo de la información.
* Políticas y regulación: normas y estándares que fundamenta el *software*.
* Ciclo de vida del sistema: establece las etapas y duración del desarrollo y uso en producción.
  1. **La especificación de requisitos a través de marcos de trabajo ágiles**

Los marcos de trabajo ágiles promueven la comunicación oral sobre la documentación exhaustiva en la mayoría de los procesos del ciclo de vida, particularmente en los procesos de identificación de necesidades y diseño. Sin embargo, uno de los artefactos presentes para el modelado de requerimientos son las historias de usuario.

Las historias de usuario son una explicación general e informal de una función del *software* escrita desde la perspectiva del usuario final o cliente. Permiten describir de una manera muy breve un requerimiento, estimar prioridades, alcance y tiempo de realización (Rivadeneira, 2014). En la siguiente tabla, se puede observar la estructura base de un documento de historia de usuario.

**Tabla 8***Estructura base de un documento de historia de usuario*

|  |  |
| --- | --- |
| Historia de usuario | |
| Número: # | Nombre de la historia de usuario: |
| Usuario: | |
| Prioridad: | Puntos estimados: |
| Descripción: | |
| Observaciones: | |
| Criterios de aceptación: | |

Nota: Tomado de Rivadeneira (2014).

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.
  1. **Scrumy la especificación de requisitos**

El marco de trabajo Scrumestá soportado en un proceso de construcción iterativo e incremental evolutivo, en el que se identifican tres roles principales: el equipo de trabajo (*team*) conformado por los desarrolladores, diseñadores, personal de calidad y de infraestructura requerido para la construcción del producto de *software*; el *scrum master* que realizan funciones parecidas a las de un director de proyecto, pero más enfocados en garantizar que el equipo de trabajo tenga todas las herramientas y recursos necesarios para el desarrollo de su trabajo; y, finalmente, el dueño del producto (*product owner*) que se convierte en un representante del cliente y quien es el único encargado de la gestión de requisitos del proyecto (ScrumStudy, 2021).



https://t4.ftcdn.net/jpg/04/25/38/31/240\_F\_425383128\_3ZXWAQYxxEbYZsw6akosUOjvjUJ89553.jpg

Scrumestablece el concepto de *sprint* para referirse a una iteración que contempla tiempos fijos entre 2 y 4 semanas dependiendo del equipo de trabajo, durante este tiempo se incluye la planeación del *sprint,* donde se definen los requerimientos a desarrollar en ese periodo de tiempo, una fase de construcción del producto y, finalmente, un proceso de despliegue para poder hacer la respectiva demostración de lo construido al final de iteración en reuniones de revisión; en este marco de trabajo se redefine el concepto de requerimiento hecho y normalmente va mucho más allá de construir el código, por lo general, se incluyen procesos de validación con pruebas unitarias y pruebas de integración.

El artefacto mediante el cual se condensan todos los requerimientos del sistema se denomina **pila de producto** (*product backlog*), la cual es una lista ordenada por prioridad de todos los requerimientos del sistema generalmente descritos en la forma de **historias de usuario**.

Todas las historias de usuario se priorizan utilizando números enteros consecutivos de **1 hasta N**, donde 1 representa la máxima prioridad. Esta priorización la realiza únicamente el dueño del producto y, para eso, utiliza la información que tiene sobre el negocio y recomendaciones de expertos buscando el mayor retorno de inversión a sus clientes. La priorización de las historias de usuario en la pila de producto puede variar en el transcurso del tiempo, pero solo la podrá realizar el dueño del producto.

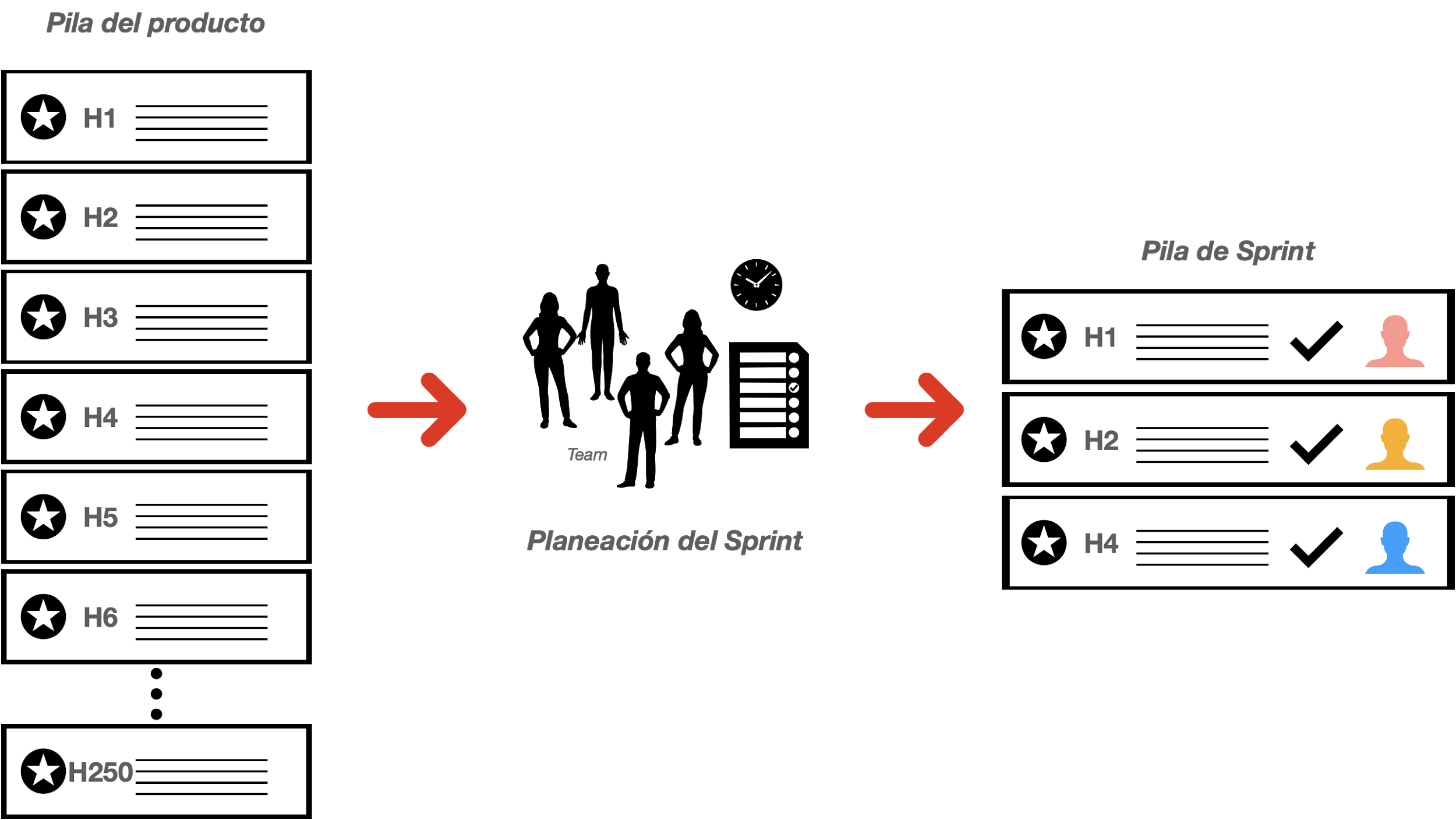
Scrum,siendo un marco de trabajo ágil, no requiere para su funcionamiento que todas las historias de usuario de la pila de producto estén detalladas, pero sí, por lo menos, las de mayor prioridad para poder iniciar el trabajo con el equipo en sus respectivos *sprints* (ScrumStudy, 2021).

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.

**Figura 3** *Pila de producto vs. pila del Sprint*



**2.5. Kanbany la especificación de requisitos**

Kanbanes una metodología para gestionar el trabajo que surge del sistema de producción *Toyota Production System (TPS)* a finales de la década de 1940, el cual representaba un sistema de producción basado en la demanda de los clientes y no en la producción masiva, lo anterior sentó los fundamentos para los sistemas de producción ajustada que consisten en minimizar los desperdicios sin afectar la producción y en crear más valor a los clientes sin generar más gastos.

A principios del siglo XXI, la industria del *software* adoptó el Kanbanpara cambiar la forma en la que se producía y entregaban productos y servicios; además, tiene en cuenta los principios de las metodologías ágiles en especial de Scrum, pero busca darle más protagonismo al proceso de experimentación y mejora continua (Rivadeneira, 2014).

Kanbanen la industria del *software* se basa en cuatro principios fundamentales:

1. Calidad garantizada, todo lo que se produce debe salir bien sin márgenes de errores, prima la calidad sobre la rapidez.
2. Reducción del desperdicio. Hacer solamente lo justo y necesario, pero hacerlo bien.
3. Mejora continua.
4. Flexibilidad: se pueden priorizar tareas entrantes según las necesidades del momento.

Además de los principios, Kanbanpropone seis (6) prácticas:

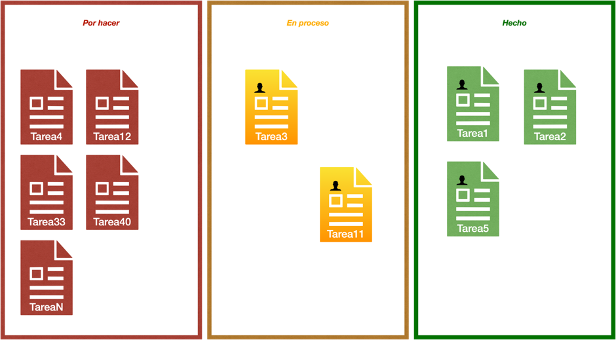
1. Visualizar el flujo de trabajo.
2. Eliminar interrupciones.
3. Gestionar el flujo.
4. Hacer políticas explícitas que fomenten la visibilidad.
5. Circuitos de realimentación.
6. Mejorar colaborando.

Tal vez la herramienta más conocida que permite implementar los principios de Kanbanes el **tablero Kanban**, el cual permite mapear y visualizar el flujo de trabajo.Este se divide en columnas a través de las cuales se pueden visualizar cada una de las fases del proceso; las filas del tablero representan los diferentes tipos de actividades específicas que se desarrollan en el marco del proyecto.

Normalmente el tablero tiene **tres secciones** que representan el estado de cada una de las tareas: **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.

**Figura 4** *Tablero Kanban*



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 Scrumcada una de las tareas se podría describir en el formato de historias de usuario.

1. **ACTIVIDADES DIDÁCTICAS (OPCIONALES SI SON SUGERIDAS)**

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la actividad | Emparejamiento entre término y definición. |
| Objetivo de la actividad | Afianzar algunos de los conceptos más importantes asociados al proceso de priorización de requisitos. |
| Tipo de actividad sugerida | Arrastrar y soltar el término con la definición que corresponde. |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo documento en Word llamado Actividad didáctica 1\_CF003. |

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Construcción de lista de priorizada de requisitos de acuerdo con técnica de priorización. |
| Objetivo de la actividad | Afianzar el proceso de priorización de requisitos por medio de algunas técnicas (puntos de historia y valor del negocio, técnica urgente, MoSCow y matriz de priorización). |
| Tipo de actividad sugerida | Realizar los cálculos y/o análisis correspondientes sugeridos en cada una de las técnicas para determinar cuál es la priorización correcta para cada requerimiento de acuerdo con la técnica utilizada. |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo documento en Word llamado Actividad didáctica 2\_CF003. |

1. **MATERIAL COMPLEMENTARIO:**

|  |  |  |
| --- | --- | --- |
| Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| UMNG. (2019). *Elementos de la norma IEEE 830*. [Video]. YouTube. https://www.youtube.com/watch?v=LjBOTZdd\_iE&feature=youtu.be | Video de YouTube | <https://youtu.be/LjBOTZdd_iE> |
| California State University Long Beach. (2021). *Requirements Engineering, CSU Long Beach, Penzenstadler.* <https://bit.ly/3rtBKXN> | Página Web - Curso | <https://bit.ly/3rtBKXN> |
| 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. (1998). *IEEE Standard | IEEE Xplore.* <https://ieeexplore.ieee.org/document/720574> | Página Web - Estándar | <https://ieeexplore.ieee.org/document/720574> |
| ISO/IEC/IEEE 29148:2011. (s. f.). *ISO.* <https://www.iso.org/standard/45171.html> | Página Web - Estándar | <https://www.iso.org/standard/45171.html> |
| Scrum Certification, Agile Certification | Scrum, Agile Training. (n.d.). *ScrumStudy.* https://www.scrumstudy.com | Página Web - Curso | <https://www.scrumstudy.com> |

1. **GLOSARIO:**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Estándar | Referencia, patrón o modelo que es utilizado a nivel general en un determinado ámbito. |
| Marcos de trabajo ágiles | Conjunto de estándares, metodologías, técnicas, frameworks o guías que rigen un proceso de desarrollo de software basadas en principios y/o valores ágiles como, por ejemplo: Scrum, Lean Software, XP, TDD, entre otros. |
| Marcos de trabajo tradicionales | Conjunto de estándares, metodologías, técnicas, frameworks o guías que rigen un proceso de desarrollo de software basadas en el ciclo de vida tradicional del software como, por ejemplo: RUP, CMMI, ISO 9001, Microsoft Solution Framework, entre otros. |
| Metodología | Síntesis de un conjunto de técnicas, métodos y procedimientos que se deben seguir durante el desarrollo de un proyecto. |
| Pruebas unitarias | Forma de comprobar el correcto funcionamiento de una unidad de código. |
| Pruebas de integración | Prueba que se ejecuta una vez se aprueban las pruebas unitarias y lo que busca es verificar que el conjunto de fragmentos de código funciona junto de forma correcta. Es una prueba de conjunto. |
| Técnica | Manera en la que un conjunto de procedimientos es aplicado en una tarea específica, con base en un conocimiento para obtener un resultado específico. |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

830-1998 - IEEE Recommended Practice for Software Requirements Specifications. (1998). *IEEE Standard | IEEE Xplore.* <https://ieeexplore.ieee.org/document/720574>

Pantaleo, L., y Rinaudo. (2018). *Ingeniería de* *software*. Alfaomega.

Penzenstadler, B. (s. f.). *Requirements engineering*. *CSU Long Beach.* <https://bit.ly/3rtBKXN>

Porfirio, C. (2021). *Técnicas de priorización: el desafío de conseguir un orden para las funcionalidades*. <https://bit.ly/3cvumqz>

Rivadeneira, M., S. G. (2014). Metodologías ágiles enfocadas al modelado de requerimientos. Informes Científicos Técnicos - *UNPA, 5*(1), 1-29. <https://doi.org/10.22305/ict-unpa.v5i1.66>

Scrum Certification, Agile Certification | Scrum, Agile Training. (n.d.). *ScrumStudy.* https://www.scrumstudy.com

Sommerville I. (2011). *Ingeniería del software.* Addison-Wesley.

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| Autor (es) | Jonathan Guerrero Astaiza | Experto temático | Regional Cauca, Centro de teleinformática y producción industrial | Marzo 2021 |
| Zulema Yidney León Escobar | Experto temático | Regional Cauca, Centro de teleinformática y producción industrial | Marzo 2021 |
| Alix Cecilia Chinchilla Rueda | Diseñador instruccional | Centro de Gestión Industrial. | Abril 2021 |
|  | José Gabriel Ortiz Abella | Corrector de estilo | Centro para la Industria y la Comunicación Gráfica. Regional Distrito Capital | Abril del 2021 |

1. **CONTROL DE CAMBIOS**

**(Diligenciar únicamente si realiza ajustes a la Unidad Temática)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia | Fecha | Razón del Cambio |
| Autor (es) |  |  |  |  |  |