**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Construcción de Requisitos de *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 - Establecer requisitos de la solución de *software* de acuerdo con estándares y procedimiento técnico. | RESULTADOS DE APRENDIZAJE | 220501092 - 2 Interpretar el informe de requisitos de la solución conforme a la identificación de las necesidades del negocio. |

|  |  |
| --- | --- |
| 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. Matriz de trazabilidad.
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. *Kanban* y 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 llegar a la definición de los 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, este proceso además de involucrar un proceso de análisis también requiere de un proceso de 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

La priorización de requisitos 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, dependencias y el retorno de inversión a los clientes, entre otros indicadores.

La priorización de requisitos generalmente 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 informar 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 también los requisitos 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. Tomar decisiones en el marco de un proyecto siempre va a generar inconformidades, pero si no se toman decisiones 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 nuestra vida cotidiana. Este tipo de priorización simple es una de las más utilizadas y consiste en dar 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 gran 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 un conocimiento profundo 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 técnica 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 entre más alto representa más valor para el cliente (Porfirio, 2021).

Usando solo el valor del negocio como elemento de priorización puede presentar 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 que es asignado por el equipo de desarrolladores a cada requerimiento donde expresa una estimación de esfuerzo, entre más grande el número implica más esfuerzo requerido para 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 estará en primeros lugares los requerimientos más sencillos de resolver por los desarrolladores y que además tengan mayor interés por parte del cliente. Observe a continuación 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 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 van registrando las prioridades de acuerdo con el valor del cociente de mayor a menor.

* + 1. **Técnica urgente.**

En esta técnica se utiliza una tabla de dos dimensiones, donde la dimensión 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 de acuerdo con 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 los 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 observe 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 podríamos concluir que el primer requerimiento a abordar sería el requerimiento **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, donde las etiquetas 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 requerimientos con etiqueta **M**, luego los requerimientos con etiqueta **S**, luego aquellos con **C** y finalmente los requerimientos 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 armar 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, considere 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, 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.**

La matriz de trazabilidad es una herramienta que permite alinear los requisitos del proyecto con los logros de los objetivos, es una tabla que relaciona cada uno de los requerimientos con el entregable solicitado. Es decir, permite identificar que se resultado se alcanza con cada requisito y a la vez permite visualizar qué requisitos son necesarios cumplir para un determinado entregable (Pantaleo, 2018).

Este es un instrumento clave para el responsable del proyecto especialmente 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. Así mismo 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 el siguiente gráfico, se propone el contenido base de una matriz de correlación dividida por secciones:

Identificador: código único para cada requisito.

Código jerárquico: código que permite categorizar cada grupo de requisitos.

Descripción: texto explicativo del requerimiento.

Tipo: categoría del requisito (requerimiento de negocio, requerimiento de los interesados, funcionales y no funcionales, requerimientos del proyecto, requerimiento de calidad)

Versión: versión del requerimiento (importante para el control de cambios)

Estado: activo, cancelado, diferido, agregado, aprobado, asignado o completado.

Fecha de estado: momento en el que se realizó la última modificación del estado.

Responsable: persona encargada del requisito.

Prioridad: escalas de prioridad usada en el proyecto (alta, media, baja; 1, 2, 3, 4, 5).

Otras características: cualquier otro campo de valor para la organización.

Objetivo: objetivo que pretende alcanzar el requerimiento.

Necesidad de negocio: necesidad de negocio que se pretende cubrir con el objetivo del proyecto y por ende con el requisito.

Entregable: producto.

*Tabla 5.*

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 considere útiles para el proyecto, 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. **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.**

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

En primer lugar, se encuentra la sesión de introducción, la cual cuenta con 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 que está por fuera del alcance.
* Definiciones, acrónimos y abreviaturas: se especifica el significado de algunas palabras que pueden ser difíciles de entender y cuales han sido abreviadas.
* Referencias: se indican todas las fuentes externas utilizadas para la construcción del documento.

En segundo lugar, se encuentra la sesión 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.

En la tercera sesión se encuentra la especificación de requerimientos, la cual es la sesión más larga e importante del documento ya que acá 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 sesión debe cumplir con los siguientes criterios de calidad: corrección, rastreable, no ambiguo, verificable y priorizado.

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

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

Finalmente, la quinta sesión 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 |

Revise ahora 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).

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. Proporciona orientación adicional en la aplicación de procesos de ingeniería y gestión de requerimientos relacionados con la ingeniería de requisitos. Se definen los elementos de información aplicables a la ingeniería de requisitos y su contenido

El estándar IEEE:29148 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 métodos y metodologías 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, 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 fomenta 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. *SCRUM* y la especificación de requisitos

El marco de trabajo *SCRUM* está 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 enfocado a 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

*SCRUM* establece 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 simplemente construir el código, generalmente se incluye 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 el dueño del producto únicamente 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 si por lo menos las de mayor prioridad para poder iniciar el trabajo con el equipo en sus respectivos *sprints* (SCRUMStudy, 2021).

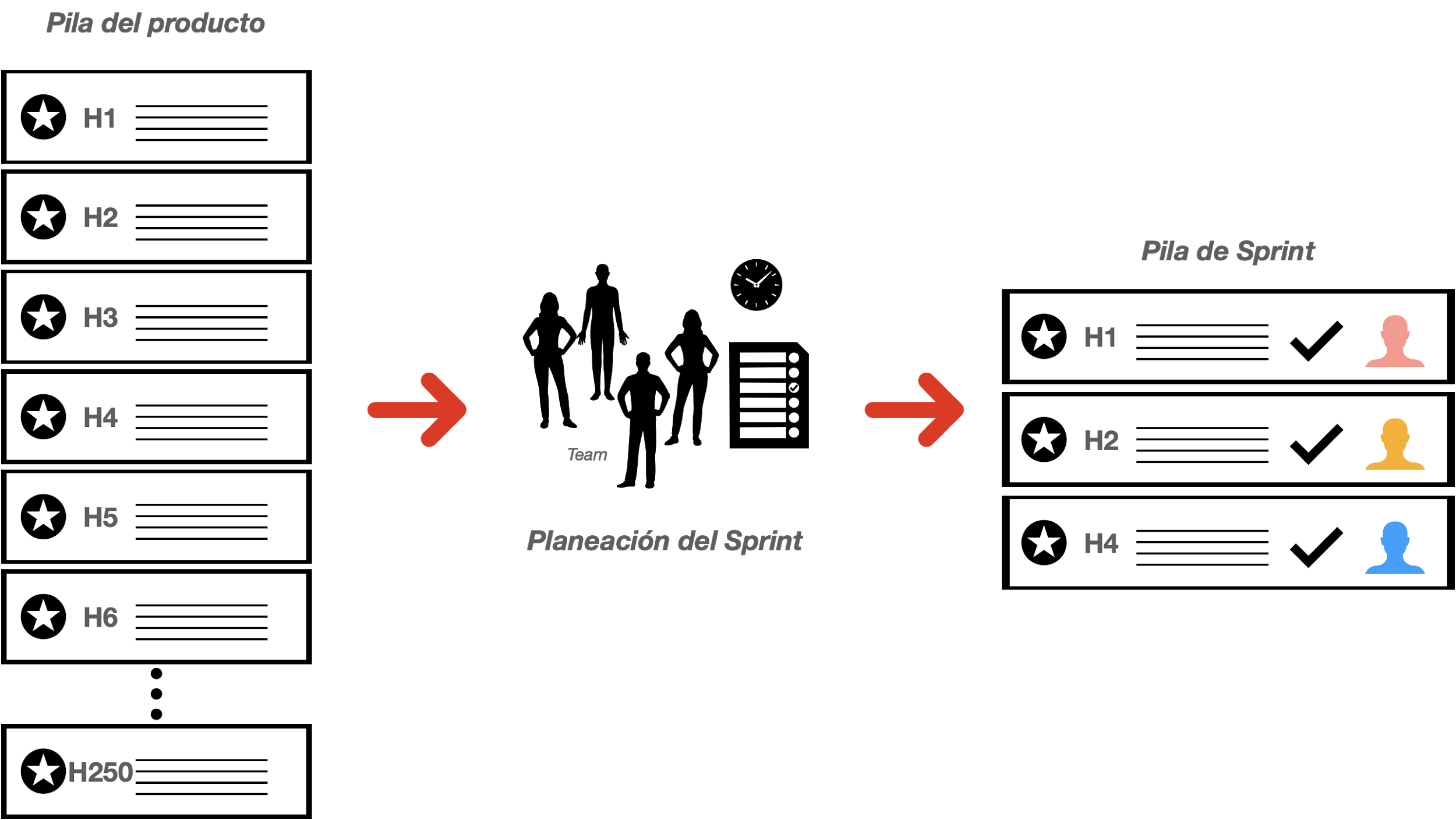
Al inicio de cada *sprint* se realiza el proceso de planeación que involucra principalmente tres tareas:

(i) 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.

(ii) selección de las historias de 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*.

(iii) descomposición de cada historia de usuario en tareas y de ser necesario asignar responsables a cada tarea.

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

*Ilustración 1. Pila de producto vs Pila del Sprint.*

**2.5. *Kanban* y la especificación de requisitos**

*Kanban* es una metodología para gestionar el trabajo que surge del sistema de producción *Toyota Production System (TPS)* a finales de los años 40, el cual representaba un sistema de producción que se basa en la demanda de los clientes y no en la producción masiva, lo anterior sentó las bases para los sistemas de producción ajustada que consisten en minimizar los desperdicios sin afectar la producción, crear más valor a los clientes sin generar más gastos. A principios del siglo XXI, la industria del *software* adoptó el *kanban* para cambiar la forma en la que se producía y entregaban productos y servicios. 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).

*Kanban* en 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, *Kanban* propone seis prácticas:

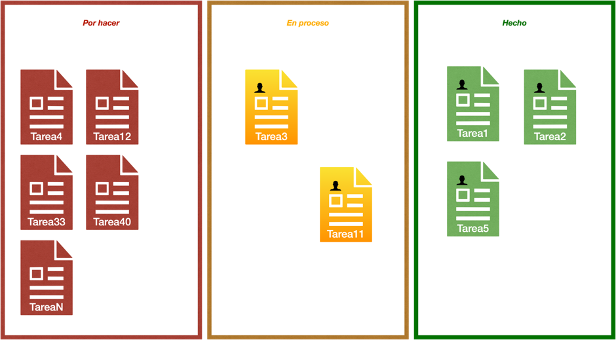
1. visualizar el flujo de trabajo.
2. eliminar interrupciones.
3. gestionar el flujo.
4. hacer políticas explícitas que fomenten la visibilidad.
5. circuitos de realimentación.
6. mejorar colaborando.

Tal vez la herramienta más conocida que permite implementar los principios de *kanban* es el **tablero *kanban***, el cual permite mapear y visualizar el flujo de trabajo. Este tablero se divide en columnas a través del cual se pueden visualizar gráficamente 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 gráfica.

*Ilustración 2. 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 *SCRUM* cada 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 correspondiente 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 |
| Elementos de la norma IEEE 830. (2019, 12 junio). [Vídeo]. YouTube. https://www.youtube.com/watch?v=LjBOTZdd\_iE&feature=youtu.be | Video de *Youtube* | <https://youtu.be/LjBOTZdd_iE> |
| Requirements Engineering, CSU Long Beach, Penzenstadler. R. Recuperado 30 de marzo de 2021, de <https://bit.ly/3rtBKXN> | Página Web - Curso | <https://bit.ly/3rtBKXN> |
| 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. (1998, 20 octubre). 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. Recuperado 30 de marzo de 2021, de <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. Retrieved April 6, 2021, from 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 | Se refiere al 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 | Se refiere al 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:**

* G. Pantaleo, L. Rinaudo. (2018). Ingeniería de *Software*. México: Alfaomega
* Sommerville I. (2011). Ingeniería del software. México: Addison-Wesley
* Porfirio, C. (s. f.). *Técnicas de priorización: el desafío de conseguir un orden para las funcionalidades*. atSistemas - CONSULTORÍA IT BLOG. Recuperado 28 de marzo de 2021, de <https://bit.ly/3cvumqz>
* Penzenstadler, B. (s. f.). *Requirements Engineering*. CSU Long Beach. Recuperado 28 de marzo de 2021, de <https://bit.ly/3rtBKXN>
* Rivadeneira Molina, 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>
* 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. (1998, 20 octubre). IEEE Standard | IEEE Xplore. <https://ieeexplore.ieee.org/document/720574>
* Scrum Certification, Agile Certification | Scrum, Agile Training. (n.d.). SCRUMStudy. Retrieved April 6, 2021, from https://www.scrumstudy.com

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Nombre | Cargo | Dependencia  *(Para el SENA indicar Regional y Centro de Formación)* | Fecha |
| Autor (es) | Santiago Muñoz dela Rosa | Experto temático | Regional Cauca, Centro de teleinformática y producción industrial | Noviembre 2021 |
| Zulema Yidney León Escobar | Experto temático | Regional Cauca, Centro de teleinformática y producción industrial | Noviembre 2021 |
|  |  |  |  |

1. **CONTROL DE CAMBIOS**

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

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