**PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL**

**FORMATO GUÍA DE APRENDIZAJE**

**PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL**

**FORMATO GUÍA DE APRENDIZAJE**

**IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE**

* Denominación del Programa de Formación: ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN.
* Código del Programa de Formación: 228106.
* Nombre del Proyecto ( si es formación Titulada): DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE DE ACUERDO LAS NECESIDADES DEL SECTOR TEXTIL EN ANTIOQUIA.
* Fase del Proyecto ( si es formación Titulada): PLANEAR.
* Actividad de Proyecto(si es formación Titulada): DEFINIR LOS REQUERIMIENTOS DEL SOFTWARE A DESARROLLAR APLICANDO EL ESTÁNDAR DE CALIDAD.
* Competencia: 220501006 - ESPECIFICAR LOS REQUISITOS NECESARIOS PARA DESARROLLAR EL SISTEMA DE INFORMACIÓN DE ACUERDO CON LAS NECESIDADES DEL CLIENTE.
* Resultados de Aprendizaje Alcanzar:
  + ELABORAR MAPAS DE PROCESOS QUE PERMITAN IDENTIFICAR LAS ÁREAS INVOLUCRADAS EN UN SISTEMA
  + PLANTEAR DIFERENTES ALTERNATIVAS, DE MODELOS TECNOLÓGICOS DE INFORMACIÓN EMPRESARIAL, TENIENDO EN CUENTA LA PLATAFORMA TECNOLÓGICA DE LA EMPRESA Y LAS TENDENCIAS DEL MERCADO, PARA DAR SOLUCIÓN A LAS SITUACIONES RELACIONADAS CON EL MANEJO DE LA INFORMACIÓN DE LA ORGANIZACIÓN
* Duración de la Guía: 90 horas

**2. PRESENTACIÓN**

Estimado aprendiz:

La presente guía de aprendizaje le entregará un conjunto de saberes y haceres relacionados a la elaboración de mapas de procesos para la identificación de áreas involucradas y especificación de requerimientos de un sistema de información, utilizando herramientas informáticas y las TIC. Esto, partiendo de sus saberes previos, sin duda alguna le permitirá desarrollar habilidades integrales asociadas a su formación profesional permitiéndole ser autónomo, sistemático y organizado en el desarrollo de cada una de las tareas asignadas, de igual modo se espera que tenga un papel activo, propositivo y colaborativo con el resto de participantes del proceso de formación.

Recuerde que tendrá a su disposición los recursos y materiales didácticos correspondientes según los requerimientos de la presente actividad de aprendizaje, entre ellos un equipo de instructores capacitados y dispuestos a atender cualquier inquietud que surja como consecuencia del desarrollo de la presente guía.

.

**3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE**

* **Descripción de la(s) actividad(es):** Las siguientes actividades están diseñadas para permitirle al aprendiz plantear la solución a situaciones relacionadas con el manejo de información de las organizaciones y la identificación de las áreas involucradas, así como la especificación de los requerimientos funcionales y no funcionales de un sistema de información.
* **Ambiente Requerido:** Aula convencional, ambiente de aprendizaje (sala de sistemas), laboratorio práctico de sistemas, unidad productiva del entorno e infraestructura institucional SENA.
* **Materiales:** Mesas, sillas, computador con conexión a internet, cable HDMI, cable VGA, marcadores borrables, borrador de tablero, hojas de papel, bolígrafos.

**3.1 Actividad de reflexión inicial:**

* **Descripción de las actividades:**  Con el objetivo de conocer su apreciación sobre una situación expuesta y tener un primer acercamiento a las necesidades de profesionales en la industria del software, planteamos las siguientes acciones:

**Acción 1:** Stop de palabras

Cada aprendiz debe investigar acerca de empresas dedicadas al desarrollo de software, también empresas comerciales de otro sector económico que cuenten con un departamento interno de sistemas a nivel local. Indicar para cada empresa encontrada:

* Nombre.
* Ubicación.
* Cargos actuales o cargos solicitados
* Servicios ofrecidos.
* Ofertas salariales.

**Actividad 1**

Para esta actividad se cuenta con una hora, deberán crear un documento donde anexaran un screen shot o pantallazo de las vacantes consultadas y sus respectivas direcciones web.

tras la cual el instructor dirigirá un juego de stop donde los aprendices deben socializar las empresas consultadas sin repetir las de sus compañeros.

**Acción 2: Contexto sobre los requisitos del software**

¿Por que es Importante el Levantamiento de la Información?

El desarrollo integral de soluciones requiere el levantamiento de la información clara y concreta, debido a que las empresas no tienen el personal con este conocimiento incurren en sobre costos por reproceso y definiciones incompletas, que son resueltas cuando se cuenta con el apoyo de consultores/asesores con formación en Ingeniería de Requisitos.

Buena parte de los problemas que surgen a lo largo del proceso de desarrollo se deben a la carencia de un proceso adecuado de definición y entendimiento del problema y a la definición poco clara de las necesidades del cliente.

**INGENIERÍA DE REQUERIMIENTOS**

Desde inicios del siglo ha empezado a estudiarse la importancia de las fases previas en el ciclo de vida del desarrollo del software y como, con ayuda de compañías que se encargan de medir los errores más comunes en estos proyectos de tecnología y desarrollo se puede apreciar que la obtención de requerimientos inexactos hace que los procesos y en general el proyecto se atrase y pierda en recursos, así como también desborda su alcance. Por supuesto se ha creado lo que conocemos como ingeniería de requerimientos que, partiendo de una definición, desglosa todo lo que se puede hacer en esta etapa de levantamiento de informacion o de requerimientos además de cómo podemos asegurar la calidad de ellos.

**DEFINICIÓN**

Según Pressman ingeniero de software, consultor y autor estadounidense, la ingeniería de requerimientos puede definirse como:

*“Aquella que proporciona el mecanismo apropiado para entender lo que desea el cliente, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación y administrar los requerimientos a medida que se transforman en un sistema funcional[[1]](#footnote-1)”*

**FASES DEL ANÁLISIS DE REQUERIMIENTOS**

Según Pressman, en su libro de ‘Ingeniería de software, un enfoque práctico’[[2]](#footnote-2), también define las cinco fases del proceso de análisis de requerimientos:

1. Reconocimiento del problema: se deben de estudiar inicialmente las especificaciones del sistema. El analista debe establecer un canal adecuado de comunicación con el equipo de trabajo involucrado en el proyecto. En esta etapa la función primordial del analista en todo momento es reconocer los elementos del problema tal y como los percibe el usuario.
2. Evaluación y síntesis En esta etapa el analista debe centrarse en el flujo y estructura de la información, definir las funciones del software, determinar los factores que afectan el desarrollo de nuestro sistema, establecer las características de la interfaz del sistema y descubrir las restricciones del diseño. Todas las tareas anteriores conducen fácilmente a la determinación del problema de forma sintetizada.
3. Modelización: durante la evaluación y síntesis de la solución, se crean modelos del sistema que servirán al analista para comprender mejor el proceso funcional, operativo y de contenido de la información.
4. Especificación: las tareas asociadas con la especificación intentan proporcionar una representación del software. Esto más adelante permitirá llegar a determinar si se ha llegado a comprender el software, en los casos que se lleguen a modelar se pueden dejar plasmados manuales.
5. Revisión: una vez que se ha descrito la información básica, se especifican los criterios de validación que han de servir para demostrar que se ha llegado a un buen entendimiento de la forma de implementar con éxito el software. La documentación del análisis de requerimientos y manuales, permitirán una revisión por parte del cliente, la cual posiblemente traerá consigo modificaciones en las funciones del sistema por lo que deberán revisarse el plan de desarrollo y las estimaciones previstas inicialmente

Lo anterior es una base y está implícito en cualquier proceso de software actualmente, pero a partir de este estudio y estandarización enfocado a los requerimientos es que se podrá estructurar de mejor manera como analizar el levantamiento de requerimientos, pero al mismo tiempo verificar la calidad de los mismos.

**SIETE Técnicas de levantamiento de requerimientos software**

Una etapa fundamental en proyectos de ingeniería de software, es la identificación y documentación de los requerimientos del futuro sistema al comienzo del proyecto, pues en numerosas ocasiones se ha demostrado que es cuando pueden prevenirse errores que puedan significar el fracaso del proyecto.  
  
En la Ingeniería de requisitos, el levantamiento de requerimientos se refiere a la identificación y documentación de los requerimientos de un sistema, a partir de los usuarios, clientes o interesados (Stakeholders). A la práctica también se le conoce como Recopilación de requerimientos.

**1. - Análisis de documentación**

* Consiste en obtener la información sobre los requerimientos funcionales y requerimientos no funcionales de software a partir de documentos que ya están elaborados.
* Es útil cuando los expertos en la materia no están disponibles para ser entrevistados o ya no forman parte de la organización.
* Utiliza la documentación que sea relevante al requerimiento que se está levantando.
* Ejemplos de documentación: Planes de negocio, actas de constitución de proyecto o acta de inicio, reglas de negocio, contratos, definiciones de alcance, memorándums, correos electrónicos, documentos de entrenamiento, entre otros.

**Sobre los requerimientos funcionales y no funcionales.**

La gestión de los requerimientos funcionales deficiente, es citada como una de las causas más frecuentes en el fracaso de los proyectos, es por ello que es importante entender que son los requerimientos funcionales, bajo qué metodologías deben identificarse y gestionarse para asegurar el logro de los objetivos.

**¿Qué son los requerimientos funcionales? Abreviado RF**

El Requerimiento funcional se acostumbra manifestar sobre un componente o parte especifica del sistema. El Requerimiento no funcional, por lo contrario, se acostumbra manifestar de manera general y se aplica a todo o casi todo el software.

Los **Requerimientos funcionales** definen los criterios que este debe cumplir para que este sea adecuado para su propósito (Fitness-for-purpose), mientras que los requerimientos no funcionales especifican los criterios que debe cumplir para que sea adecuado para su uso (Fitness-for-use).

La guía del [**Business Analysis Body of Knowledge (BABOK)**](https://www.amazon.com/-/es/IIBA/dp/1927584027/ref=sr_1_1?__mk_es_US=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=1SQU793A08C35&keywords=Business+Analysis+Body+of+Knowledge&qid=1658780826&sprefix=business+analysis+body+of+knowledge%2Caps%2C210&sr=8-1) en su version 3, proporciona la siguiente definición para los requerimientos funcionales de una solución:

*“Los requerimientos funcionales son las descripciones explicitas del comportamiento que debe tener una solución de software y que información debe manejar.”*

Profundizando en las causas de los problemas producto de loserrores en la gestión de requerimientos funcionales, entre las situaciones observadas con mayor frecuencia son:

* Requerimientos funcionales con descripciones muy ambiguas, produciendo interpretaciones inadecuadas por parte del equipo de desarrollo.
* El requerimiento funcional no fue entendido adecuadamente cuando fue levantado con el interesado, pasando información incorrecta al equipo de desarrollo.
* En su forma original, el requerimiento funcional no era [**factible técnicamente**](http://www.pmoinformatica.com/2016/04/modelo-estudio-de-factibilidad.html) y el equipo de desarrollo realizó modificaciones que no fueron aprobadas por los interesados.

Es fundamental la aplicación de metodologías probadas de gestión de requerimientos funcionales que eviten que estos problemas sucedan, algunas de las prácticas más recomendadas son:

* Ante la presencia de ambigüedades, solicitar información adicional, mesas de trabajo o reuniones con los interesados. Es menos costoso esperar a obtener una descripción que aclare las dudas antes que asumir y avanzar en el desarrollo.
* Validar las descripciones escritas de los requerimientos funcionales antes de comenzar su desarrollo.
* Cualquier duda que presente el equipo de desarrollo buscar comunicación con los interesados para su resolución.
* Si durante el desarrollo se presentará la necesidad de **modificar algún requerimiento funcional** debido a razones técnicas, solicitar mesas de trabajo o reunir, no proceder con un desarrollo e implementación que no ha sido validado por los interesados.

Siguiendo estas prácticas podemos asegurar una mejor aceptación de los requerimientos funcionales cuando llegue el momento de la implementación.

**Por lo tanto, los requerimientos funcionales:**

* Expresan las capacidades o cualidades que debe tener la solución para satisfacer los requerimientos de los interesados de proyecto (stakeholders)
* Se expresan en términos de cuál debe ser el comportamiento de la solución y que información debe manejar.
* Deben proporcionar una descripción lo suficientemente detallada para permitir el desarrollo e implementación de la solución.
* Son los que más influyen en si la solución será aceptada o no por los usuarios.

**Ejemplos de RF.**

* El sistema enviará un correo electrónico cuando se registre alguna de las siguientes transacciones: pedido de venta de cliente, despacho de mercancía al cliente, emisión de factura a cliente y registro de pago de cliente.
* Se permitirá el registro de pedidos de compra con datos obligatorios incompletos, los cuales podrán completarse posteriormente modificando el pedido. Antes de poder aprobarse los datos del pedido deben estar completos.
* Al aprobar un pedido, la solicitud pasará al siguiente paso del flujo de trabajo (workflow) de aprobación configurado en el sistema.
* El sistema permitirá a los usuarios autorizados el ingresar planes y cronogramas de [**proyecto**](http://www.pmoinformatica.com/2020/09/que-es-un-proyecto-pmi.html).
* El sistema permitirá aprobar, cambiar o actualizar planes y cronogramas de [**proyecto**](http://www.pmoinformatica.com/2019/02/que-es-un-proyecto.html).
* El sistema permitirá el envío automatizado de cartas de entrega de órdenes directamente al almacén.
* A cada orden se le asignará un identificador único, que será utilizado para identificarla en todos los procesos subsecuentes que se realicen sobre esta.
* Al ingresar ordenes de entrega, toda orden de entrega estará asociada a un pedido de venta.
* La facturación de pedidos de venta se realizara en lotes, por medio de una pantalla de pedidos pendientes de facturación, la cual mostrará los pedidos no facturados. Una vez facturados los pedidos no se mostrarán en esta lista.
* El sistema también permitirá el registro de facturas manuales no asociadas a pedidos, sin embargo, estas requerirán autorización por parte del grupo de Gerentes antes de ser contabilizadas.
* El proceso de compras en el sistema abarcará los siguientes pasos y transacciones: Ingreso de la requisición, emisión de la solicitud de cotización y emisión de la orden de compra.
* Los elementos de la solicitud de cotización serán los mismos de la requisición asociada, al igual que los de la orden de compra. El sistema permitirá la emisión de solicitudes de cotización y órdenes de compra parciales.
* La contabilización de transacciones de facturas de venta y facturas de compra podrá configurarse para realizarse de forma automatizada a su registro, o manualmente en lotes (Proceso Batch).
* El software debe poder emitir los siguientes estados financieros: Balance general, Estado de ganancias y pérdidas, Estado de flujos de efectivo. Además, debe poder emitir un listado de mayor general y mayor analítico.
* Los pedidos de compra que excedan los montos establecidos en el flujo de liberaciones de pedidos configurados, deberán pasar por las aprobaciones establecidas en dicho flujo de aprobación.

## Ejemplos de requerimientos funcionales de interfaz gráfica

* La solución validara automáticamente el cliente asociado a una orden con el sistema de gestión de contactos.
* El campo de monto acepta únicamente valores numéricos con dos decimales.
* El campo fecha de transacción acepta únicamente fechas anteriores al día de hoy (día actual).
* El campo nombre acepta caracteres alfabéticos únicamente.
* El campo dirección acepta caracteres alfabéticos, numéricos y especiales.
* El campo país consistirá en una lista de preselección. El país asociado a una dirección debe ser previamente registrado en el sistema.
* El campo estado, provincia o departamento consistirá en una lista de preselección. A los usuarios se les presentará únicamente los estados asociados al país seleccionado previamente. El departamento o provincia a seleccionar deberá ser registrado en la funcionalidad correspondiente.
* El campo material de elemento de la pantalla de requisiciones de compra será una lista de preselección, que mostrará únicamente los materiales registrados en el maestro de materiales.
* El campo fecha contable acepta únicamente fechas que correspondan con periodos contables que estén abiertos en el sistema.
* La pantalla de registro de pago puede imprimir los datos en pantalla a la impresora.
* Se mostrará el nombre, tamaño total, espacio disponible y formato de un pen drive o flash drive conectado al puerto USB del computador.

## Ejemplos de requerimientos funcionales legales o regulatorios

* El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados.
* La base de datos será implementada con trazas de auditoría.
* Las hojas de cálculo aseguraran los datos usando firmas electrónicas.
* El sistema permitirá elaborar y emitir el reporte regulatorio XX, según los requerimientos establecidos en el reglamento y ley aplicable.
* Los libros de venta y de compras serán emitidos en el formato establecido por las autoridades tributarias de dicha materia.

## Ejemplos de requerimientos de seguridad

* El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados. Los usuarios deben ingresar al sistema con un nombre de usuario y contraseña.
* El sistema enviará una alerta al administrador del sistema cuando ocurra alguno de los siguientes eventos: Registro de nueva cuenta, ingreso al sistema por parte del cliente, 2 o más intentos fallidos en el ingreso de la contraseña de usuario y cambio de contraseña de usuario.
* Los integrantes del grupo de usuarios de analistas pueden ingresar solicitudes pero no pueden aprobarlas o borrarlas.
* Los integrantes del grupo de usuarios de gerentes pueden ingresar y aprobar solicitudes, pero no pueden borrarlas.
* Los integrantes del grupo de usuario de administradores no pueden ingresar o aprobar solicitudes, pero si pueden borrarlas.
* Cualquier intercambio de datos vía internet que realice el software se realizará por medio del protocolo encriptado https.

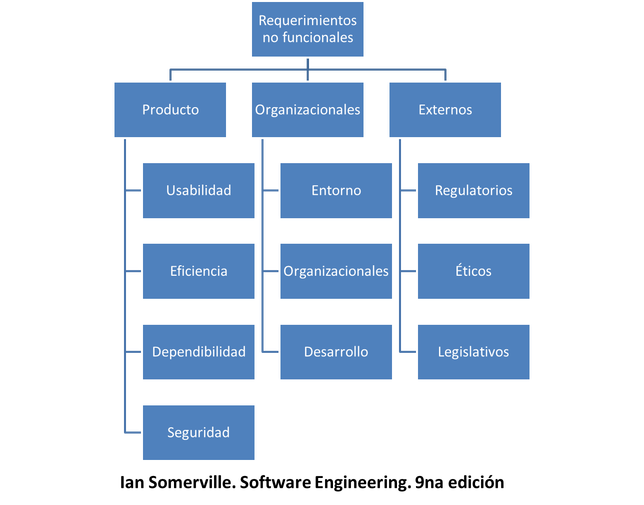
**¿Qué son los requerimientos NO funcionales? Abreviado RNF**

Todos los Servicios de Tecnología de Información (TI) en algún punto de su ciclo de vida, necesitan considerar los requerimientos no funcionales y las pruebas asociadas a los mismos.  
  
Para algunos proyectos, estos requerimientos implican una cantidad considerable de trabajo y esfuerzos, mientras que para otros no.

Los requerimientos no funcionales son los que especifican criterios para evaluar la operación de un servicio de tecnología de información, en contraste con los **requerimientos funcionales** que especifican los comportamientos específicos.  
  
Por lo general, el Plan para implementarlos requerimientos no funcionales se detalla en la Arquitectura del Sistema, mientras que el de los requerimientos funcionales se especifica en el Diseño.

Los **requerimientos no funcionales** representan características generales y restricciones de la aplicación o sistema que se esté desarrollando.  
  
Suelen presentar dificultades en su definición dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir.  
  
Entre los ejemplos de requerimientos no funcionales presentados, tenemos los referidos a atributos como: la eficiencia, seguridad, dependibilidad y usabilidad del sistema.

En un primer nivel, los requerimientos no funcionales pueden clasificarse en requerimientos de producto, organizacionales y externos,



En un segundo nivel, los requerimientos de producto pueden clasificarse en requerimientos de usabilidad, eficiencia, dependibilidad y seguridad. A su vez, los requerimientos organizacionales pueden clasificarse en requerimientos de entorno, organizacionales y de desarrollo. Asimismo, los requerimientos externos pueden clasificarse en requerimientos regulatorios, éticos y legislativos.

Otra forma de categorizarlos podría ser la siguiente

* Cualidades observables en tiempo de ejecución, como por ejemplo la usabilidad y la seguridad.
* Cualidades relacionadas con la evolución del sistema, como por ejemplo Mantenibilidad, Comprobabilidad, Extensibilidad y Escalabilidad, las cuales están inmersas en la estructura del sistema de software.

**Ejemplos de RNF**

**Ejemplos de requerimientos no funcionales de producto**  
  
**Eficiencia**

* El sistema debe ser capaz de procesar N transacciones por segundo. Esto se medirá por medio de la herramienta **SoapUI aplicada al Software Testing de servicios web**.
* Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.
* El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.
* Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 2 segundos.

**Seguridad lógica y de datos**

* Los permisos de acceso al sistema podrán ser cambiados solamente por el administrador de acceso a datos.
* El nuevo sistema debe desarrollarse aplicando **patrones y recomendaciones de programación que incrementen la seguridad de datos**.
* Todos los sistemas deben respaldarse cada 24 horas. Los respaldos deben ser almacenados en una localidad segura ubicada en un edificio distinto al que reside el sistema.
* Todas las comunicaciones externas entre servidores de datos, aplicación y cliente del sistema deben estar encriptadas utilizando el algoritmo RSA.
* Si se identifican ataques de seguridad o brecha del sistema, el mismo no continuará operando hasta ser desbloqueado por un administrador de seguridad.

**Seguridad industrial**

* El sistema no continuará operando si la temperatura externa es menor a 4 grados Celsius.
* El sistema no continuará operando en caso de fuego. (Ej. Un ascensor).

**Usabilidad**

* El tiempo de aprendizaje del sistema por un usuario deberá ser menor a 4 horas.
* La tasa de errores cometidos por el usuario deberá ser menor del 1% de las transacciones totales ejecutadas en el sistema.
* El sistema debe contar con manuales de usuario estructurados adecuadamente.
* El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final.
* El sistema debe contar con un módulo de ayuda en línea.
* La aplicación web debe poseer un **diseño “Responsive”** a fin de garantizar la adecuada visualización en múltiples computadores personales, dispositivos tableta y teléfonos inteligentes.
* El sistema debe poseer interfaces gráficas bien formadas.

**Actividad 2. En grupo**

Definir un conjunto de requerimientos funcionales para apps conocidas como Instragam o Telegram (por ejemplo, no deben de ser la misma app por equipo.)

A continuación se expone un listado de seis apps poco reconocidas. En grupo de trabajo deberán explorar en las aplicaciones, en sus sitios web y construir el respectivo listado de RF y RNF que consideren, posterioremente deberán compartirlo al grupo

|  |  |  |
| --- | --- | --- |
| IFTTT | Star Walk | My Script Calculator |
| Karma – Save Food with a tap | Hush City - La App que Ayuda a Encontrar los Espacios Silenciosos | Seek App que te Permite Identificar Plantas e Insectos |

**Actividad 3.**

Realizar la lectura Requerimientos Funcionales bajo Diferente Enfoques.

Ver Documento: PMO Documento de requerimientos de software plantilla. En Carpeta Documentos Anexos Requerimientos

En grupos de máximo cuatro integrantes.

Leer el documento y empezar la gestión del mismo para una aplicación que (seleccionar solo una):

* Promueva el comercio del café de origen en Europa.
* Elimine o reduzca los intermediarios en la la cadena de comecialización de productos alimenticios en Antioquia.
* Facilite la adopción de mascotas que busquen hogar
* Proponga una idea de proyecto.

Diligenciar el documento en su totalidad y socializar la propuesta de idea, asi como los items del documento

**2.- Observación**

Consiste en estudiar el entorno de trabajo de los usuarios, clientes e **interesados de proyecto** (Stakeholders).

Los stakeholders, un término muy difícil de traducir literalmente al español, son las personas y organizaciones como clientes, patrocinadores, organización ejecutante y el público, involucrados activamente en el proyecto, o cuyos intereses pueden verse afectados de manera positiva o negativa por la ejecución o conclusión del proyecto. También se les conoce como interesados, involucrados o actores del proyecto.

Es una técnica útil cuando se está documentando la situación actual de procesos de negocio.

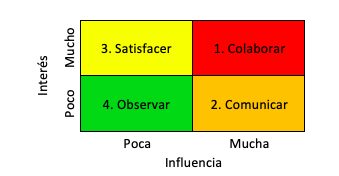
Puede ser de dos tipos, pasiva o activa.

En observación pasiva, el observador no hace preguntas, limitándose solo a tomar notas y a no interferir en el desempeño normal de las operaciones. En observación activa, el observador puede conversar con el usuario.

En la observaión activa se podría gestionar los interesados a partid e la Matriz de Interesados

**¿Qué es la Matriz de Stakeholders?**

La matriz de stakeholders es una herramienta que se utiliza para recopilar, clasificar, analizar y jerarquizar de manera sistemática información cualitativa y cuantitativa referente a todas aquellas personas, instituciones u organizaciones involucradas o interesadas en el proyecto, lo que permite determinar los intereses particulares que deben tenerse en cuenta a lo largo del proyecto. La utilización de esta herramienta de análisis permite clasificar a los involucrados en el proyecto según sus niveles de interés y poder sobre él, lo que facilita la priorización de los stakeholders más importantes para desarrollar así las estrategias de gestión correspondientes.



**Actividad 4**

Analice en el centro una situación donde se pueda resolver por medio de un software. Una vez hecho esto construya la matriz de intersados, defina el proyecto y la lista de RF y RNF.

**3.- Entrevistas**

* Se realizan con los usuarios o interesados clave.
* Direccionan al usuario hacia aspectos específicos del requerimiento a levantar.
* Son útiles para obtener y documentar información detallada sobre los **requerimientos y sus niveles de granularidad**.
* Pueden ser entrevistas formales o informales.
* Una clave es mantenerse enfocado en los objetivos de la entrevista.
* Las preguntas abiertas son útiles para identificar información faltante.
* Las preguntas cerradas son útiles para confirmar y validar información.
* El éxito de las entrevistas depende del grado de conocimiento del entrevistador y entrevistado, disposición del entrevistado de suministrar información, buena documentación de la discusión y en definitiva de una buena relación entre las partes.

En la **Ingeniería de requerimientos** y en el desarrollo de software, el nivel de granularidad es la mayor o menor amplitud de la descripción del comportamiento esperado para el software en una especificación funcional.

En esencia, la granularidad define que tan detallada o tan general es la descripción de una funcionalidad de software.  
  
Las distintas etapas de un proyecto de desarrollo de software requieren distintos niveles de granularidad en la descripción de los requisitos funcionales: En etapas tempranas del proyecto, se requiere una visión amplia del alcance, mientras que en etapas avanzadas se necesita tener un nivel de granularidad mayor, es decir, una visión profunda del software

**Los niveles de granularidad**  
  
Continuando con el ejemplo del sistema de cajero automático, las siguientes descripciones podrían estar presentes en su **especificación de requerimientos**:

1. Realizar transacciones con la cuenta corriente.
2. Transferir una cifra de una cuenta corriente para otra cuenta corriente.
3. Validar la tarjeta y la contraseña del cliente.
4. Garantizar que la suma de todas las transacciones del cliente en el día no puede ser superior a $5,000.

Aunque estos cuatro ejemplos sean casos validos de requerimientos funcionales, es posible percibir que el nivel de detalle (o granularidad) entre ellos es distinto.

**4.- Encuestas o cuestionarios**

* Es una técnica útil para recopilar eficientemente los requerimientos de muchas personas.
* La clave para el éxito es que tengan un propósito y audiencia claramente definida, establecer fechas topes para llenar la encuesta, con preguntas claras y concisas.
* Deben enfocarse en los objetivos de negocio que se necesitan identificar.
* Pueden apoyarse con entrevistas de seguimiento con usuarios individuales.
* Pueden contener tanto preguntas cerradas como preguntas abiertas.

**5.- Mesas de trabajo (Workshops)**

* Es una técnica efectiva para obtener información rápidamente de varias personas.
* Es recomendable tener una agenda predefinida y preseleccionar a los participantes, siguiendo **buenas prácticas para reuniones efectivas**.
* Se puede utilizar un facilitador neutral y un transcriptor (que no sea el mismo facilitador).
* Se puede utilizar un material común sobre el cual enfocar la atención y conversar, por ejemplo una presentación con un desglose del proceso que se está estudiando o un flujograma.
* Se pueden combinar con otras técnicas como pueden ser las entrevistas y cuestionarios.

**Las reuniones de trabajo: más productividad, menos reuniones**  
 **Limitar el número de reuniones**

* Evitar hacer reuniones para coordinar y dirigir el trabajo del día a día, estableciendo en su lugar roles, tareas y canales de comunicación en el equipo. Si todos saben que deben hacer y con quien deben comunicarse no tendrán que reunirse a cada momento.
* Fomentar un ambiente en el que los miembros del equipo tengan libre acceso unos a otros cuando se necesite acordar algo, evitando reunir a todo el equipo.
* Evite utilizar las reuniones para asignar trabajo adicional al equipo (además del planificado), pues dichas acciones con frecuencia no se han considerado y evaluado lo suficiente y pueden interferir con el trabajo que es realmente productivo.
* Pedir perdón y no permiso, la mayoría de las decisiones pueden ser tomadas por la misma persona. Si el equipo se está reuniendo para tomar cualquier pequeña decisión, es necesario fortalecer la confianza entre colaboradores y trabajo en equipo.
* Los buenos gerentes siempre aprecian a los colaboradores que toman la iniciativa, siempre y cuando tengan justificación de sus acciones. Fomentar esta cultura en los equipos de trabajo.
* Utilizar herramientas automatizadas de Gestión, la cual puede ayudar con la productividad del equipo, reemplazando a los correos electrónicos y las reuniones.
* Establecer un calendario compartido por todos, asegurando que todos lo usen y que este actualizado. Esto permitirá ver cuánto tiempo se está invirtiendo en reuniones para luego trabajar en reducir el tiempo de participación en reuniones de sus colaboradores.

**Reuniones más productivas**

* Definir objetivos claros y cuantificables para la reunión y hacerlos del conocimiento de los participantes. ¿Es para preparar un reporte?, ¿Es para recibir un reporte y revisarlo?, ¿es para evaluar el progreso de un proyecto? o ¿es para desarrollar el plan?.
* Determinar la duración, hora de inicio y hora de finalización. Una reunión no debería durar más de una hora y media dado que deja de ser productiva.
* Establecer una agenda y tiempos (duración) para cada punto de la misma.
* Evitar la convocatoria extraordinaria o “por sorpresa”,
* Evaluar cuidadosamente los participantes que deben asistir, en función de lo que pueden aportar, evitando la concurrencia multitudinaria (ocasiona mayor dispersión).
* **Asegurar que los participantes reciban y revisen los documentos a revisar con antelación.**
* Utilizar la tecnología, tales como herramientas de trabajo colaborativo, el dropbox y las redes sociales para adelantar la mayor cantidad de trabajo posible antes de la reunión.
* Evaluar hacer las reuniones a primera hora o a última hora del día, para evitar interferir con la jornada laboral y evitar interrupciones durante la reunión.
* Establecer reglas claras en cuanto a las intervenciones permitidas, momento en que se pueden hacer, hora de inicio y hora fin, entre otras.
* Cumplir la hora de inicio y duración de cada punto de la agenda.
* Moderar las reuniones en función de los objetivos y puntos de la agenda, evitar que se traten puntos fuera de secuencia o no relacionados con los mismos.
* Para reuniones de coordinación o estatus, evaluar hacerlas de pie (stand up meetings), de esta forma se pueden moderar la longitud de las intervenciones.
* Antes de pasar al siguiente punto de la agenda, debe hacerse un resumen, para asegurar que los participantes estén alineados en lo que se ha dicho y los acuerdos.
* Establecer claramente las conclusiones de la reunión, acciones a tomar, responsables de esas acciones y fechas compromiso.
* Evaluar siempre los resultados de la reunión y tomar en consideración para futuras, ¿se cumplieron los tiempos?, ¿Qué podemos mejorar en las reuniones para hacerlas más productivas?, etc.

Para estos fines se pueden utilizar diferente tipo de metodologías y técnicas de trabajo, que veremos de a poco.

**6.- Tormenta de ideas**

Es una sesión de trabajo estructurada orientada para obtener la mayor cantidad de ideas posibles.

Es recomendable limitarlas en el tiempo, utilizar ayudas visuales y designar un facilitador.

Las reglas son importantes, por ejemplo los criterios para evaluar ideas y asignarles un puntaje, no permitir las críticas a las ideas y limitar el tiempo de discusión.

En una primera fase, se deben identificar la mayor cantidad de ideas, para luego evaluarlas. Todas las ideas deben ser consideradas y deben limitarse que una idea se le ahogue o critique antes de tener tiempo de desarrollarla.

Para este fin, se suele utilizar herramientas como design thinking y desing sprint

**7.- Historia del usuario**

Las **historias de usuario**, son una aproximación simple al levantamiento de requerimientos de software, en la cual la conversación pasa a ser más importante que la formalización de requerimientos escritos.

Es recomendable que sean escritas por el mismo cliente o interesado (con apoyo del facilitador si es necesario), con énfasis en las funcionalidades que el sistema deberá realizar.

Al redactar una historia de usuario deben tenerse en cuenta describir el Rol, la funcionalidad y el resultado esperado de la aplicación en una frase corta.

Las historias de usuario son una de las técnicas más difundidas para levantar requerimientos de software en metodologías ágiles.

**3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje:**

* **Descripción de las actividades:** En esta etapa se acercará al aprendiz al contexto de las empresas y sus procesos internos, conocerá herramientas básicas para identificar/esquematizar necesidades, así como estructurar soluciones, se relacionará con herramientas como los mapas de procesos, mapas mentales, y mapas conceptuales. Para cumplir este objetivo, el aprendiz debe desarrollar las siguientes acciones:

**Acción 1:** Mapas conceptuales vs Mapas mentales

Durante la sesión de formación se deben abordar los conceptos de mapa mental y conceptual. Se propone al grupo de aprendices una introducción formal explicativa acerca del tema y luego se realiza la construcción diferentes productos con el apoyo de textos que impliquen el desarrollo de la lógica y la estructuración de ideas.

Para desarrollar los mapas conceptuales:

1. El organigrama escolar, ¿cómo se compone la estructura de cargos de una institución educativa?
2. Transformaciones que hacen la diferencia: Artículo publicado por la revista Enter.CO. Recuperado de:<https://www.enter.co/especiales/empresas/transformaciones-que-hacen-la-diferencia/>
3. Los fuegos, texto de literatura de Eduardo Galeano. Recuperado de:<https://image.slidesharecdn.com/mardefueguitosdiferentes-150403153350-conversion-gate01/95/mar-de-fueguitos-diferentes-5-638.jpg?cb=1428077586>

Para desarrollar los mapas mentales:

1. Qué es la inteligencia artificial: Artículo publicado por el blog de tecnología Xataka. Recuperado de:<https://www.xataka.com/robotica-e-ia/que-inteligencia-artificial>
2. Qué es la inteligencia artificial Parte 2: Artículo publicado por el blog de tecnología Xataka. Recuperado de:<https://www.xataka.com/robotica-e-ia/que-inteligencia-artificial>
3. Proyecto formativo: Componentes de hardware, ¿Qué necesito?

El instructor a cargo debe dirigir la sesión de tal manera que se pueda socializar cada mapa, al mismo tiempo que se pueda revisar y recomendar al aprendiz la manera correcta de enfrentarse a un texto de cualquier tipo, para realizar su síntesis o extraer componentes principales.

Para la socialización final se sugiere seguir la siguiente pregunta: ¿En dónde puedo utilizar los mapas mentales y conceptuales a la hora de programar? Se sugiere revisar el mapa mental que recomienda que lenguaje de programación aprender que se encuentra en la carpeta de Recursos de apoyo.

**Acción 2:** Mapas mentales

En esta acción, los aprendices se reunirán en los grupos de proyecto para preparar una presentación donde se presente a nivel global, y utilizando solo la argumentación, las ideas generales del proyecto que se ha idealizado.

Además, debe realizar un mapa mental en el cual indique los elementos principales (entradas, procesamiento, salidas, metas) del proyecto formativo que como grupo de proyecto pretende desarrollar a lo largo de su etapa lectiva.

Preguntas para socializar:

¿Sirven los mapas mentales para realizar presentaciones de ideas?

¿Qué ventaja tiene un mapa mental a uno conceptual?

¿Deberíamos hacer mapas mentales de nuestros proyectos para explicarlos mejor?

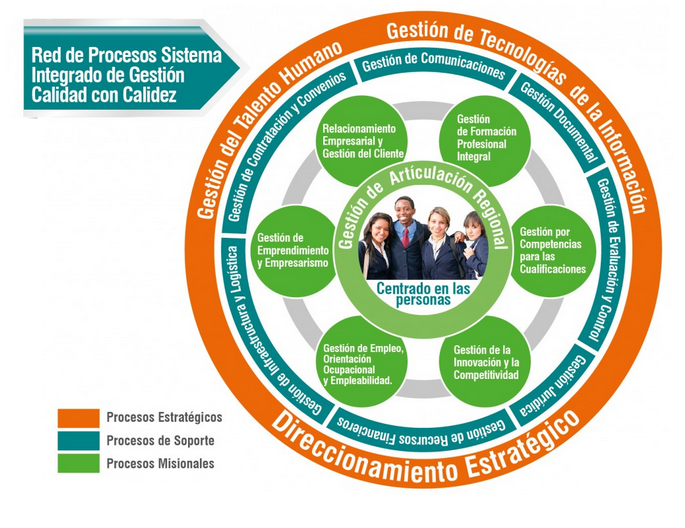
Por último, deberán adelantar la construcción de un mapa mental que implique los componentes de hardware que su proyecto usa o necesita.

Esta actividad corresponde a una evidencia de producto, los mapas mentales deberán subirse en el enlace de la plataforma Blackboard habilitado por el instructor.

**Acción 3:** Mapa de procesos

Las empresas manejan una serie de procesos que le permiten llevar de mejor manera su operación, para representarlos se utilizan mapas de procesos.

El instructor dará las nociones y características básicas para la elaboración de un mapa de procesos. Durante la sesión se analizará el mapa de procesos del SENA, momento en el que cada aprendiz debe pensar cómo se adaptaría ese mapa de procesos a un esquema de un proyecto, en este caso, su proyecto formativo. Con esta información, los aprendices, en los grupos del proyecto, deben buscar el mapa de procesos de una empresa con operaciones similares a su proyecto formativo y aplicarlo en cada uno de sus proyectos, como recurso de apoyo para la identificación de la problemática. Este mapa de procesos aplicado a su proyecto formativo es un insumo para el informe de entrega de fase.



Mapa de procesos SENA

**3.3 Actividades de apropiación:**

* **Descripción de las actividades:** Con el objetivo de fortalecer las orientaciones entregadas por el instructor y a la vez apropiarse de los temas, subtemas y conceptos acerca de la especificación de requerimientos de software o proyecto informático, el aprendiz debe desarrollar las siguientes acciones:

**Acción 1:** Formulación de proyectos

En esta acción, el instructor dará las pautas para la formulación de proyectos, incluyendo componentes básicos como la definición del nombre (comercial y académico), participantes, imagen corporativa, mapa de procesos desarrollada previamente, planteamiento del problema, la justificación del proyecto, la construcción de objetivos, alcance, beneficiarios, matriz DOFA, impactos (social, económico, ambiental, tecnológico), restricciones y limitaciones (recursos de equipos, herramientas, talento humano, materiales, instalaciones, tiempo, recursos económicos).

A partir de las conclusiones obtenidas en el paso anterior, los aprendices usarán la plantilla de presentaciones dada por el instructor, se debe crear la presentación del proyecto formativo.

En los grupos del proyecto, los aprendices deben consultar información acerca de los de modelos tecnológicos. Después de leer y analizar cada uno de los modelos consultados, se hará una búsqueda de empresas del área metropolitana en donde apliquen algunos de estos modelos para su funcionamiento. Cada grupo reconocerá el modelo más pertinente a su proyecto formativo y lo debe incluir en la presentación del proyecto.

**Acción 2:** Cuadro comparativo gestores bases de datos

El instructor organizará grupos de 3 aprendices y asignará a cada uno el nombre de un gestor de base de datos. Cada grupo completará la información que se pide en el siguiente cuadro, de acuerdo al gestor designado.

Cada equipo debe exponer su gestor de base de datos y los otros grupos completarán el cuadro con la información de sus compañeros.

**Acción 3.**

Debate sistemas operativos: MacOS (Capitan, Yosemite, Jaguar), Windows (XP, 7, 8, 10, Server), Linux (Ubuntu, Debian, Fedora, openSUSE, Kali Linux, Redhat)

El instructor realizará una breve introducción acerca de los sistemas operativos y los tipos de software.

El grupo debe dividirse en 3 subgrupos, los cuales seleccionarán uno de los sistemas operativos propuestos por el instructor.

Cada subgrupo debe elaborar una presentación que incorpore los elementos distintivos relacionados al sistema operativo seleccionado. Debe consultar de las diferentes versiones:

* Requerimientos de hardware.
* Vulnerabilidades.
* Licenciamiento.
* Usos principales.
* Clientes objetivos.
* Cuota de mercado.
* Reseña histórica.

Finalmente, cada subgrupo dispondrá de 20 minutos para exponer a sus compañeros la presentación preparada. EL instructor guiará las conclusiones de la actividad definiendo fortalezas y debilidades de cada uno de los sistemas operativos propuestos.

**3.4 Actividades de transferencia:**

* **Descripción de las actividades:** Una vez realizadas las actividades de reflexión, contextualización y apropiación sobre los conceptos básicos de los mapas de procesos, modelos tecnológicos, bases de datos, redes, y sistemas operativos, es el momento de construir un documento que asiente las bases de requerimientos de su proyecto formativo, y realizar la transferencia de lo aprendido, para ello, se plantea las siguientes acciones:

**Acción 1.** El documento de especificación de requerimientos de software

En esta acción, el instructor dará una introducción a la especificación de los requerimientos (funcionales y no funcionales) de un sistema de información. Luego de esto, el instructor pedirá la búsqueda de un documento ERS IEEE830 (ejemplo diligenciado).

Los aprendices deben reunirse por grupos de proyecto

* + - 1. Leer exhaustivamente el documento IEE830 revisar sus contenidos y la forma de redacción. Identificar los requisitos funcionales y no funcionales de su proyecto.

Desarrollar un mapa conceptual del documento.

* + - 1. Revisar documento con historias de usuario con metodologías agiles.
      2. De acuerdo con los requerimientos y necesidades identificadas en cada uno de sus proyectos redactar las historias de usuario de su proyecto teniendo en cuenta los requerimientos funcionales y no funcionales, basado en los lineamientos de la metodología SCRUM.

Esta actividad corresponde a una evidencia de desempeño, y producto, las historias de usuario se deben adjuntar a las presentaciones del proyecto previamente realizadas. Este documento se debe anexar en el enlace que el instructor disponga para tal fin en la plataforma TERRITORIUM.

**Acción 3.** Entrega de FASE

El grupo de instructores desarrollará una estrategia para que expongas ante otros grupos tu proyecto, se debe hacer la entrega de los documentos desarrollados para el proyecto y hacer una presentación siguiendo los lineamientos institucionales, esta entrega de fase se puede hacer tipo SIMPOSIO o tipo FERIA DE PROYECTOS. Los aprendices deben sustentar la presentación de sus proyectos.

Esta actividad corresponde a una evidencia de conocimiento.

**4. ACTIVIDADES DE EVALUACIÓN**

|  |  |  |
| --- | --- | --- |
| **Evidencias de Aprendizaje** | **Criterios de Evaluación** | **Técnicas e Instrumentos de Evaluación** |
| **Evidencias de Conocimiento:**   * ***AP1\_ESPECIFICACR\_GA2\_ENTREGA\_DE\_FASE***   **Evidencias de Desempeño**   * ***AP1\_ESPECIFICAR\_GA2\_FORMULACION\_PROYECTOS***   **Evidencias de Producto**   * ***AP1\_ESPECIFICAR\_GA2\_FORMULACION\_PROYECTOS*** | * (**DESEMPEÑO**)   PROPONE ALTERNATIVAS DE PLATAFORMAS TECNOLÓGICAS PARA EL  MEJORAMIENTO DE LOS PROCESOS, A PARTIR DE LA IDENTIFICACIÓN DE LOS RECURSOS  DE HARDWARE Y SOFTWARE DISPONIBLES EN LA ORGANIZACIÓN Y LAS TENDENCIAS DEL  MERCADO.   * (**PRODUCTO**)   PRESENTA INFORMES RELACIONADOS CON LAS NECESIDADES DEL USUARIO, FRENTE A LOS REQUERIMIENTOS DEL SISTEMA DE INFORMACIÓN, DE FORMA  ORGANIZADA, DE ACUERDO CON LOS PROTOCOLOS ESTABLECIDOS.   * (**PRODUCTO**)   REALIZA EL INFORME DE REQUERIMIENTOS, RELACIONADO CON LOS  DIFERENTES PROCESOS DE LA ORGANIZACIÓN, DE ACUERDO CON LAS NORMAS Y  PROTOCOLOS ESTABLECIDOS. | *Técnica didáctica activa*   * *Observación*   *Instrumentos*   * *Lista de verificación.* |

**5. GLOSARIO DE TÉRMINOS**

**Mapa mental:** Diagrama en el que se unen términos a una idea central, en el cuál debe explicarse un proyecto, idea, negocio o tema en particular. Su principal característica es que deben tener contenido multimedia.

**Mapa conceptual:** Diagrama jerárquico en el que se unen conceptos, definiciones o descriptores a través de líneas. Estos mapas son indicados para desarrollar procesos de síntesis o estructuras lógicas de funcionamiento.

**Mapa de procesos:** Diagrama relacional en el que se conectan las diferentes etapas de un proyecto u organización en una cadena de trabajo que implica varias fases o etapas.

**Red neuronal:** Tecnología asociada al desarrollo de sistemas de aprendizaje de máquina que trabajan bajo la lógica de un cerebro humano.

**Inteligencia Artificial:** Tecnología que hace parte de la cuarta revolución industrial en la que se tiene como gran protagonista la automatización inteligente de procesos a través del uso de tecnología digital.

**6. REFERENTES BIBLIOGRÁFICOS**

* Ramírez, Felipe.: Cognotécnicas: herramientas para pensar más y mejor / Felipe Ramírez.
* México, D.F.: Alfaomega Grupo Editor, c2014. Disponible Biblioteca complejo Norte.
* Tafur Portilla, Raúl: Cómo hacer un proyecto de investigación: uso de diagramas, matrices y mapas conceptuales / Raúl Tafur Portilla, Manuel Izaguirre Sotomayor. Disponible Biblioteca complejo Norte.
* Ejercicios para elaborar mapas conceptuales: guía del estudiante / Rocío Quesada Castillo.
  + México: Editorial Limusa, ©2009. Disponible Biblioteca complejo Norte.
* Métodos de investigación para el diseño del producto / Alex Milton, Paul Rodgers; traducción Cristóbal Barber Casanovas. Disponible Biblioteca complejo Norte.
* Roger S. Pressman, P. (2018). Ingeniería del software. Un enfoque práctico. Mansfield: McGRAW-HILL INTERAMERICANA.

**7. CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | LEE JARED ESCOBAR  AHARÓN AGUAS  ANDRES CUELLAR  EDILFREDO PINEDA  EDWYN PATIÑO  WILLIAM BERRIO  FREDY CAMACHO  JORGE MARIO ZAMORA | INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR | CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC |  |

**8. CONTROL DE CAMBIOS** (diligenciar únicamente si realiza ajustes a la guía)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del Cambio** |
| Autor (es) |  | INSTRUCTOR | TICS | 09/03/2020 | Modificación de actividades por cambio de plataforma |
| **Autor (es)** | LEE JARED ESCOBAR  ANDRES CUELAR C  AHARÓN AGUAS  EDILFREDO PINEDA | INSTRUCTORES | CTGI TIC | 11/03/2020 | Actualización formato, anexo de metodología SCRUM y modificación de entregables. |

1. Roger S. Pressman, P. (2018). Ingeniería del software. Un enfoque práctico. Mansfield: McGRAW-HILL INTERAMERICANA. [↑](#footnote-ref-1)
2. IDEM [↑](#footnote-ref-2)