**Taller Especificación de requisitos del software**

**Defina los siguientes términos:**

 **Análisis de requisitos**: Es el proceso de estudiar y entender qué necesita el usuario o el cliente de un software. Consiste en recopilar, examinar, y priorizar los requisitos o necesidades que el sistema debe cumplir, asegurando que el equipo de desarrollo sepa exactamente qué debe construir. (Analizar las características principales del programa).

 **Atributo**: Es una característica o propiedad que describe un aspecto específico de un objeto, sistema o componente. Por ejemplo, en un software, un atributo puede ser el "nombre de usuario" asociado a una cuenta.

 **Característica**: Es una funcionalidad o capacidad de un sistema que satisface una necesidad o cumple un objetivo. Las características son, en esencia, lo que el software puede hacer, como la opción de iniciar sesión o la posibilidad de enviar un mensaje. (bloque de acción).

 **Caso de Uso**: Es una descripción detallada de cómo un usuario interactúa con el sistema para lograr un objetivo específico. Explica los pasos que sigue un usuario y el sistema para completar una tarea, como "registrarse en una página web". (¿cómo lo usara el usuario?).

 **Ciclo de Vida**: Se refiere a las diferentes fases o etapas que atraviesa un software desde su concepción hasta su retiro. Incluye etapas como la planificación, el desarrollo, las pruebas, la implementación y el mantenimiento.



* **Planificación.** La fase de planificación es la que permite conocer sobre el alcance que tendrá el proyecto, que puntos abarcará, los posibles riesgos que puede llegar a presentar y el orden en el cual se ejecutarán todas las tareas en el proceso de su creación. (lluvia de ideas).
* **Análisis.** Esta fase es la que estudia las necesidades de información de los usuarios finales, consiste en documentar los requerimientos específicos para el software a desarrollar, tal como se fijaron en la etapa de planeamiento, constituyéndose la base del diseño de un sistema de información. (documentación).
* **Diseño.** En esta fase, se selecciona la arquitectura más adecuada para llevar adelante el desarrollo y cumplir con los requerimientos fijados en la documentación. (guía).
* **Implementación.** Consiste en el desarrollo propiamente dicho del producto, tomando como referencia los documentos de diseño. (desarrollo).
* **Pruebas.** En esta etapa se evalúa el desempeño del programa, para verificar la existencia de fallas o posibilidades de mejora hasta alcanzar los estándares de calidad definidos en la planificación. (verificación).
* **Instalación / Despliegue.** Luego de obtener la certificación de las pruebas donde se indica que éstas fueron exitosas, se procede con la instalación del software para su puesta en producción y utilización por parte de los usuarios finales. (despliegue).

 **Criterios de aceptación**: Son un conjunto de condiciones que un software debe cumplir para ser considerado completo y satisfactorio para el cliente o usuario. Estos criterios definen cuándo un requisito o función se considera correctamente implementada. (condiciones).

 **Documentación de requerimientos**: Es un conjunto de documentos que describe todas las necesidades y expectativas que el software debe cumplir. Incluye tanto requisitos funcionales (lo que el software debe hacer) como no funcionales (como debe comportarse). (documentación del funcionamiento del programa)

 **Épica**: Es una gran historia de usuario que aún no está suficientemente detallada y que puede dividirse en múltiples historias más pequeñas. Una épica representa un gran bloque de trabajo que debe descomponerse para ser más manejable. (categorización)

 **Especificación de requisitos**: Es un documento formal que describe de manera detallada todas las funcionalidades y características que debe tener el software. Sirve como guía para los desarrolladores y los testers, asegurando que el sistema final cumpla con las expectativas del cliente. (documentación guía de las características del programa)

 **Guía**: Es un conjunto de instrucciones o pautas que orientan a los usuarios o a los desarrolladores sobre cómo realizar ciertas tareas o tomar decisiones dentro del desarrollo de software. (plano del programa)

 **Historia de Usuario**: Es una descripción breve y simple de una funcionalidad desde el punto de vista del usuario. Se utiliza para capturar los requisitos de una manera ágil, por ejemplo: "Como usuario, quiero poder cambiar mi contraseña para proteger mi cuenta". (NO ENTENDI).

 **Método**: Es una secuencia de actividades organizadas y orientadas hacia un objetivo o meta común. En desarrollo de software, los métodos son las estrategias que se siguen para llevar a cabo tareas específicas, como diseñar, codificar o probar el sistema. ().

 **Metodología**: Es el conjunto de métodos, técnicas y herramientas que se utilizan para gestionar y desarrollar software. Una metodología proporciona una estructura clara para llevar a cabo el proyecto, como SCRUM o la metodología en cascada. ().

 **Necesidad**: Es un requerimiento o demanda que surge para resolver un problema o cumplir un objetivo. En software, las necesidades suelen traducirse en requisitos que el sistema debe satisfacer. ().

 **Producto**: Es el resultado final del desarrollo de software. Puede ser un sistema, una aplicación o un servicio que cumple con los requisitos establecidos por el cliente o usuario. ().

 **Product Backlog**: Es una lista priorizada de todas las funcionalidades, mejoras, correcciones de errores y otros cambios que se necesitan en el producto. El Product Backlog es dinámico y puede evolucionar a medida que se conocen nuevas necesidades o prioridades. ().

 **Requisito**: Es una condición o capacidad que un sistema debe cumplir. Los requisitos pueden ser funcionales, que describen qué hace el sistema, o no funcionales, que describen cómo debe comportarse el sistema. ().

 **Requisito Funcional**: Describe lo que el software debe hacer. Por ejemplo, "El sistema debe permitir a los usuarios iniciar sesión con un nombre de usuario y contraseña." ().

 **Requisito No Funcional**: Describe cómo el software debe comportarse, como el rendimiento, la seguridad o la usabilidad. Por ejemplo, "El sistema debe responder a cualquier solicitud en menos de 2 segundos." ().

 **Restricción**: Es una limitación o condición que restringe cómo se puede diseñar o desarrollar un sistema. Las restricciones pueden ser técnicas, presupuestarias, de tiempo, o relacionadas con el cumplimiento normativo. ().

 **Roles**: Son las responsabilidades o funciones asignadas a diferentes personas dentro del equipo de desarrollo o a los usuarios del sistema. Ejemplos de roles son el Product Owner, el desarrollador, o el tester. ().

 **SCRUM**: Es una metodología ágil de desarrollo de software que divide el trabajo en ciclos cortos llamados "sprints". SCRUM fomenta la colaboración constante entre el equipo y el cliente, adaptándose a los cambios rápidamente. ().

 **Sprint**: Es un ciclo de trabajo en SCRUM que normalmente dura entre 1 y 4 semanas. Durante un sprint, el equipo trabaja en un conjunto de tareas definidas y al final se revisa lo que se ha logrado para hacer mejoras en el siguiente sprint. ().

 **Stakeholders**: Son todas las personas o entidades que tienen interés en el proyecto o en el producto final. Esto incluye a los clientes, usuarios, desarrolladores, inversores y cualquier persona afectada por el proyecto. ().

**Ciclo de vida de desarrollo de software**

* **Planificación.** La fase de planificación es la que permite conocer sobre el alcance que tendrá el proyecto, que puntos abarcará, los posibles riesgos que puede llegar a presentar y el orden en el cual se ejecutarán todas las tareas en el proceso de su creación.
* **Análisis.** Esta fase es la que estudia las necesidades de información de los usuarios finales, consiste en documentar los requerimientos específicos para el software a desarrollar, tal como se fijaron en la etapa de planeamiento, constituyéndose la base del diseño de un sistema de información.
* **Diseño.** En esta fase, se selecciona la arquitectura más adecuada para llevar adelante el desarrollo y cumplir con los requerimientos fijados en la documentación.
* **Implementación.** Consiste en el desarrollo propiamente dicho del producto, tomando como referencia los documentos de diseño.
* **Pruebas.** En esta etapa se evalúa el desempeño del programa, para verificar la existencia de fallas o posibilidades de mejora hasta alcanzar los estándares de calidad definidos en la planificación.
* **Instalación / Despliegue.** Luego de obtener la certificación de las pruebas donde se indica que éstas fueron exitosas, se procede con la instalación del software para su puesta en producción y utilización por parte de los usuarios finales.

**Proceso de Análisis de Requisitos**

* **Captura y Análisis.** En esta fase el responsable del levantamiento de requerimientos establecerá contacto con los usuarios y stakeholders o interesados en el proyecto, para determinar el alcance de la solución que se desea construir; además, se debe identificar cuáles son los servicios que prestará el sistema, su rendimiento, sus necesidades y restricciones, la interoperabilidad que tendrá con otros servicios y cuáles son los objetivos esperados. Todos estos aspectos corresponden a los requerimientos que serán analizados y clasificados en requerimientos funcionales y no funcionales para determinar la mejor solución a la necesidad.
* **Especificación.** El objetivo principal de la especificación de requerimientos es obtener un documento; en el cual, se defina de una forma organizada, completa, precisa, detallada y verificable cada uno de los requerimientos que debe satisfacer el sistema a desarrollar, además de sus respectivas relaciones y restricciones. En la especificación se debe redactar y describir solamente aquella información veraz que haya sido obtenida en la sesión de trabajo con el usuario final, cliente o stakeholder y que haya sido tenida en cuenta dentro del proceso y debe ser comunicada de forma eficaz para reconocer cada componente del sistema.
* **Validación.** Consiste en mostrar o comprobar que cada uno de los requerimientos obtenidos define la funcionalidad que se va a construir y lo que desea el cliente, esta actividad debe realizarse antes de comenzar el análisis y desarrollo de software para evitar el riesgo de implementar una mala especificación, con el costo que eso conlleva y comprende dos partes: verificación y validación.
* **Gestión.** En esta etapa se realiza la comprensión y control de los cambios de cada uno de los requerimientos, incluyendo el análisis del problema, impacto, estimación del cambio y especificación del cambio.

**Clasificación de requisitos, RSM (Requirements Specification Model)**

**Responda las siguientes preguntas:**

* Defina que es el Modelo de Especificacion de Requisitos RSM
* Cuales son las dimensiones de requisitos segun el Modelo de Especificacion de Requisitos RSM
* Defina las Propiedades y Atributos de los requisitos
* Cual es la diferencia en Restriccion y Requisito
* Que problemas pueden presentarse si se lleva a cabo una mala o incompleta definición y analisis de requisitos
* En la siguiente situación, un usuario tiene una lista de deseos y necesidades, Defina detalladamente cuáles son los requisitos y cuáles son las restricciones
* **Caso de estudio:**

**Usuario:** Ana, una estudiante universitaria que busca organizar sus ideas y metas para el futuro.

**Descripción:** Ana tiene muchas ideas y metas que le gustaría lograr, pero a menudo se siente abrumada por la cantidad de información y no sabe por dónde empezar. Le gustaría tener una herramienta que le ayude a organizar sus deseos y necesidades de forma clara y concisa, para poder priorizarlas y tomar medidas para alcanzarlas.

**Enlace para el envio de la actividad**

<https://forms.gle/obpSCfQoUyd9Sfuj7>