|  |  |
| --- | --- |
| PROGRAMA DE FORMACIÓN | Gestión de analítica de datos |

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 220501092- Establecer requisitos de la solución de *software* de acuerdo con estándares y procedimiento técnico | RESULTADOS DE APRENDIZAJE | 220501092-01. Recopilar requisitos de información de acuerdo con criterios técnicos |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | CF5 |
| NOMBRE DEL COMPONENTE FORMATIVO | Recolección de los requisitos del *software* |
| BREVE DESCRIPCIÓN | Se expone la necesidad de realizar una correcta definición de los requisitos del *software* para que cubra las expectativas del cliente, cumpla con las definiciones de funcionalidad e interfaz enfocada al usuario. Se trabajan los fundamentos de recopilación de requisitos y la diferencia con la expectativa del cliente y reúne, además, la información para gestionar el proyecto de desarrollo, validando el alcance del producto final. |
| PALABRAS CLAVE | Requerimientos, mapa de proceso, plan de proyecto, usabilidad, validación |

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

**Tabla de contenidos Introducción**

[**1**](#_heading=h.1y810tw) **Entendiendo los requisitos del negocio**

[1.1](#_heading=h.qsh70q) Estudio de viabilidad del sistema y sus funcionalidades

[1.2](#_heading=h.147n2zr) Recogida de requisitos

[1.3](#_heading=h.3o7alnk) Herramientas y técnicas

[1.4](#_heading=h.23ckvvd) Salidas y entregables

[**2**](#_heading=h.ihv636) **Herramientas y técnicas para recolección de requisitos**

[2.1](#_heading=h.32hioqz) Entrevistas con patrocinador y usuarios

[2.2](#_heading=h.1hmsyys) Técnicas de creatividad grupal - Lluvia de ideas

[2.3](#_heading=h.41mghml) Cuestionarios y encuestas

[2.4](#_heading=h.2grqrue) Análisis de tareas y de dominio

[2.5](#_heading=h.vx1227) Fabricación del modelo prototipo

[2.6](#_heading=h.3fwokq0) Generación de documentación

**Introducción**

|  |
| --- |
| Cuadro de texto |
| Estimado aprendiz, reciba una cordial bienvenida a este recurso de aprendizaje. El presente componente formativo está orientado a la recolección de requisitos de *software*. Se abordan como ejes temáticos, los requisitos del negocio, así como las herramientas para dicha recolección. Asimismo, las principales actividades de la ingeniería de requerimientos, desde la toma de información al cliente, pasando por el análisis, la discusión y la validación, así como la relación que hay entre estas actividades y otros aspectos más. Al finalizar, obtendrá los conocimientos para desenvolverse en el campo de este tipo de recolección de *software*. Antes de empezar, se sugiere revisar el siguiente video para identificar el contexto de aprendizaje:  ¡Le deseamos una experiencia de aprendizaje significativa y memorable |

**Guion de video introductorio**

(VER ANEXO 1)

# Desarrollo de contenido Entendiendo los requisitos del negocio

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | **Slider presentación** | |
| **Introducción** | Al iniciar, es indispensable entender los requisitos del negocio, empresa u organización, con el fin de hacer un levantamiento adecuado de los requisitos. A continuación, se explican los pasos iniciales de este proceso: | |
| Desarrollar un proyecto de *software* parte de una idea o necesidad de negocio: resolver un problema, mejorar un proceso o explotar una nueva oportunidad.  Las personas u organizaciones que efectivamente tienen que ver con el proyecto de *software*, quienes tienen ese interés válido en cubrir esas necesidades constituyen el grupo de interés; dentro de este hay patrocinadores que impulsan el proyecto, clientes que tienen el problema a resolver y un equipo técnico de trabajo capaz de materializar esas ideas o necesidades en un producto de *software*. | | ilustración vectorial brainstorming, concepto de ayuda psicológica. personas con problemas y haciendo preguntas desbaratan la cuerda enredada, el símbolo de la solución encontrada, la búsqueda de una solución  Imagen: 228131\_i200 |
| La ingeniería de requerimientos se ocupa de:   1. Los objetivos reales de las funciones y restricciones del proyecto de *software*. 2. La relación entre los diversos factores y las especificaciones del comportamiento del *software* y su evolución en el tiempo. 3. La introducción de un *software* para resolver un problema real como resultado del proceso. 4. Un buen entendimiento inicial del problema y su contexto asociado, constituyen el núcleo de la especificación de requerimientos. 5. Las variaciones en las especificaciones, porque, aunque las partes estén de acuerdo, esta definición cambiará durante el proyecto. 6. La comprensión de las limitaciones, restricciones y el entorno de operación del *software*. | | Photo of business suit and tie with REQUIREMENTS concept paper cards  Imagen: 228131\_i201 |
| Independientemente de la metodología a seguir para el desarrollo del proyecto, la actividad inicial es determinar la viabilidad del producto final. Esta etapa de concepción del *software* tiene dos pasos: | | Imagen: 228131\_i202 |
| 1. Recopilar el catálogo de requerimientos – lo que el cliente quiere que el *software* haga. Se debe ***escuchar*** y ***conversar*** con los interesados, hacer lo que el cliente pide, evitando exagerar los requerimientos o ilusionar al cliente con algo que no se puede cumplir; lo importante es construir un sistema que funcione y cumpla las expectativas del cliente y los usuarios. | | Imagen: 228131\_i203 |
| 1. Especificar los requisitos del *software* – definir la arquitectura del sistema que cumpliría esos requerimientos definidos por el cliente, desde puntos de vista computacional y estructural, documentando y entregando la información validada por las partes interesadas a la siguiente etapa del proyecto. | | Imagen: 228131\_i204 |
| Las partes interesadas son todos los individuos u organizaciones que tienen un interés válido en el desarrollo del proyecto, quienes ganan o pierden con el éxito o el fracaso del sistema. | |  |
| Los interesados son importantes en la obtención y validación de requisitos, por eso es preciso escucharlos, ellos son:  1. Patrocinador (*sponsor*) que es quien provee los recursos.  2. Experto del dominio que valida la arquitectura.  3. Personas usuarios finales del sistema.  4. Entidades reguladoras y normalizadoras.  5. Equipo de trabajo que realiza el desarrollo. | | **Figura 1** *Interesados del proyecto*    *Nota:* Adaptada de *Análisis de los stakeholders.* Grandes Pymes (2017). <https://www.grandespymes.com.ar/2021/04/10/analisis-de-los-stakeholders/>  Imagen: 228131\_i205 |
| Todo proyecto comienza con el equipo de trabajo, intentando entender el problema a resolver y determinando cuáles resultados son importantes para las partes interesadas. Esto incluye comprender las necesidades del negocio que motivan el proyecto y las cuestiones técnicas que lo restringen. | | Imagen: 228131\_i206 |

## Estudio de viabilidad del sistema y sus funcionalidades

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | **Slider pasos** | |
| **Introducción** | El estudio de viabilidad examina el contexto del *software* a realizarse; las necesidades específicas que deben abordar el diseño y la construcción; las prioridades que guían el orden en el que se efectúa el trabajo; la información; las funciones y los comportamientos que afectarán el producto resultante.  Consiste en siete fases diferentes: concepción, indagación u obtención, elaboración, negociación, especificación, validación y administración. Algunas pueden ser concurrentes y todas se adaptan a las necesidades del proyecto de *software*:  **Figura 2** *Fases del estudio de viabilidad*    Imagen: 228131\_i207 | |
| **Slide 1** | **Concepción:**  El proyecto comienza cuando se identifica una necesidad del negocio o se descubre un mercado o servicio potencial. En esta fase se hace el entendimiento básico del problema, se identifican los usuarios finales y lo que quieren o necesitan, es decir, la naturaleza de la solución. | Imagen: 228131\_i208 |
| **Slide 2** | **Indagación u obtención:**  Consiste en preguntarle al cliente, a los usuarios y a los demás interesados cuáles son los objetivos para el sistema, qué se va a lograr, cómo se ajusta a las necesidades del negocio y, finalmente, cómo va a usarse en la operación diaria. La parte más importante de la indagación es entender los objetivos del negocio, estos pueden ser funcionales o no-funcionales.  Problemas que pueden surgir en esta etapa:  [OJO INSERTAR TARJETAS ANIMADAS. EL RECURSO SE ENCUENTRA DEBAJO DE LA TABLA 1](#TASLIDE2) | Imagen: 228131\_i209 |
| **Slide 3** | **Elaboración**:  Consiste en desarrollar, expandir y refinar la información obtenida del cliente durante la concepción e indagación. Debe identificar aspectos funcionales, del comportamiento e información del *software*. La elaboración toma los escenarios de usuario y los refina. Cada escenario o caso describe cómo el usuario va a interactuar con el sistema y se analiza para extraer las *clases* o entidades de negocio visibles al usuario. Se definen los atributos de cada clase analizada y los servicios que requiere cada una. | Grupo de personas que inician un proyecto empresarial.  Imagen: 228131\_i213 |
| **Slide 4** | **Negociación**:  Es usual que clientes y usuarios pidan más de lo que puede lograrse con la limitación de recursos del negocio. También es común que distintos clientes o usuarios propongan requerimientos conflictivos con el argumento de que su versión es “esencial para nuestras necesidades”.  Para conciliar durante la fase de negociación, se pide a clientes, usuarios y otros interesados que prioricen sus requerimientos y que analicen los conflictos. Mediante un enfoque iterativo se da prioridad a los requerimientos, evaluando su costo y riesgo, y se enfrentan los conflictos internos; algunos se eliminan, se combinan o se modifican de modo que ambas partes ganan pues se logra un acuerdo satisfactorio. | Tratos de negocios  Imagen: 228131\_i214 |
| **Slide 5** | **Especificación**:  Puede ser un documento, varios modelos gráficos, un modelo matemático, un conjunto de casos de uso, un prototipo o cualquier combinación de éstos. Puede usarse una plantilla estándar, lo que lleva a presentar los requerimientos en forma consistente y comprensible. Pero en ocasiones es necesario ser flexible cuando se desarrolla una especificación. Para sistemas grandes conviene realizar un documento escrito que combine descripciones en un lenguaje natural con modelos gráficos, pero para sistemas pequeños quizá todo lo que se requiera sean casos de uso. | Ilustración del concepto de diseño de interacción  Imagen: 228131\_i215 |
| **Slide 6** | **Validación**:  Consiste en evaluar la calidad del resultado generado desde la ingeniería de requerimientos analizando las especificaciones para garantizar que todos han sido enunciados sin ambigüedades; que se detectaron y corrigieron las inconsistencias, omisiones y errores, y que el resultado del trabajo se presenta conforme a los estándares establecidos para el proceso, el proyecto y el producto.  Ejemplos de los problemas que ocurren durante la validación de requisitos y que parecen inocuos:   * El *software* debe ser fácil de usar. * La probabilidad de una intrusión exitosa no autorizada en la base de datos debería ser menor que 0.0001.   El primer requisito es vago para los desarrolladores que lo prueben o evalúen. ¿Qué significa exactamente "fácil de usar"? Para validarlo, debe ser cuantificado o calificado de alguna manera.  El segundo requisito tiene un elemento cuantitativo (“menos de 0.0001”), pero estas pruebas de intrusión son difíciles y consumen mucho tiempo; además, ¿está justificado este nivel de seguridad para la aplicación?, ¿pueden otros requisitos complementarios de seguridad (por ejemplo, protección por contraseña, protocolo de enlace especializado) reemplazar el requisito cuantitativo anotado? | Plataforma o servicio en línea de etapa de madurez período del ciclo de vida del proyecto implementación y desarrollo de proyectos comerciales plan de consulta de tutoriales en línea ilustración plana vectorial  Imagen: 228131\_i216 |
| **Slide 7** | **Administración**:  Todos los requerimientos de *software* cambian, y el deseo o necesidad de modificarlos continúa durante el ciclo de vida del sistema. La administración es un conjunto de actividades que ayudan al equipo del proyecto a identificar, controlar hacer seguimiento y modificaciones en cualquier momento del desarrollo del proyecto. | Imagen: 228131\_i217 Socios con grandes piezas de rompecabezas |
| **Slide 8** | La importancia de la identificación de requerimientos radica en que define claramente lo que se va a hacer, entregando un documento de requisitos de *software* que servirá como insumo de entrada a las siguientes fases del proyecto, de manera que una falla inicial se propagará en las siguientes fases de elaboración y construcción, haciendo que su solución sea mucho más costosa que haberla detectado y corregido inicialmente. | Fondo equipo confirmando lista de comprobación giganteImagen: 228131\_i218 |

|  |
| --- |
| **Cuadro de texto** |
| A continuación, se presenta un ejemplo de lista de chequeo para validación de requerimientos. |

**Tabla 1** *Ejemplo de lista de chequeo*

|  |  |  |
| --- | --- | --- |
| 1 | ¿Se han establecido claramente los requerimientos?  ¿Pueden llegar a malinterpretarse? |  |
| 2 | ¿Se ha identificado la fuente del requerimiento?  ¿Se ha examinado la declaración final del requerimiento contra la fuente original? |  |
| 3 | ¿Es el requerimiento limitado en términos cuantitativos? |  |
| 4 | ¿Qué otros requerimientos se relacionan con este?  ¿Se ha verificado con una matriz de referencias cruzadas u otro mecanismo? |  |
| 5 | ¿El requerimiento viola alguna restricción del dominio del sistema? |  |
| 6 | ¿Se puede probar y testear el requerimiento?  ¿Se pueden especificar los criterios de validación para ejecutar la prueba del requerimiento? |  |
| 7 | ¿Se puede hacer la trazabilidad del requerimiento con alguno de los modelos creados? |  |
| 8 | ¿Está la especificación estructurada de tal forma que lleve al entendimiento, referenciación y fácil interpretación en productos de trabajo más técnicos? |  |
| 9 | ¿La especificación tiene un índice? |  |
| 10 | ¿Tienen claridad los requisitos asociados al desempeño, comportamiento y características operativas?  ¿Qué requisitos parecen estar implícitos? |  |

*Nota*. Adaptada de *Software Engineering: A Practitioner's Approach*. Pressman y Maxim (2019).

INSERTAR ESTE RECURSO DE TARJETAS ANIMADAS EN SLIDE 2 DE SER POSIBLE

|  |  |
| --- | --- |
| **Tipo de recurso** | Tarjetas animadas provenientes de Slide 2 |
| **Introducción** | Problemas que pueden surgir en esta etapa: |
| **Problemas de alcance**    Imagen: 228131\_i210 | Si se especifican detalles técnicos innecesarios, el límite de lo que debe hacer el sistema se confunde y se pierden los objetivos generales del sistema. |
| **Problemas de entendimiento**    Imagen: 228131\_i211 | Surgen porque hay diferencias entre la visión, el léxico e incluso la forma de ser entre clientes, usuarios, patrocinadores y el equipo de desarrollo. |
| **Problemas de volatilidad**    Imagen: 228131\_i212 | Los requerimientos no son estáticos pues cambian con el tiempo, el mercado, una nueva regulación o aparición de un producto de la competencia. |

|  |  |
| --- | --- |
| **Tipo de recurso** | Tarjetas animadas provenientes de Slide 6 |
| **Introducción** | La validación de requerimientos es un proceso continuo durante el desarrollo del *software*, para asegurar que el resultado represente las necesidades y expectativas de los clientes. Esta actividad contribuye a mejorar la calidad de los requerimientos, a reducir costos, tiempos y riesgos en el desarrollo de *software* (Santana et al., 2020). |
| Una ilustración de prueba en línea en el concepto de educación móvil vectorial de estilo plano  Imagen: 228131\_i219 | El primer requisito es vago para los desarrolladores que lo prueben o evalúen, ¿qué significa exactamente "fácil de usar"? Para validarlo, debe ser cuantificado o calificado de alguna manera. |
| Imagen: 228131\_i220 | El segundo requisito tiene un elemento cuantitativo (“menos de 0.0001”), pero en las pruebas de intrusión son difíciles y consumen mucho tiempo; además, ¿está justificado este nivel de seguridad?, ¿pueden otros requisitos asociados a seguridad (por ejemplo, protección de contraseña, protocolo de enlace especializado) reemplazar el requisito cuantitativo anotado? |
| Imagen: 228131\_i221 | Se hace mediante una revisión técnica en la que participan el equipo de trabajo y los demás interesados o sus representantes, que analizan la especificación en busca de errores de contenido o de interpretación, de aspectos en los que tal vez se requiera hacer aclaraciones, falta de información, inconsistencias y requerimientos conflictivos o inalcanzables |

## Recogida de requisitos

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Hasta ahora se ha estudiado la importancia de los requerimientos y el proceso conjunto por el cual las partes interesadas acuerdan los mismos; pero estos, además de ser identificados, deben ser descritos y documentados de forma estructurada y normalizada. | |
| Declaraciones como: “el programa debe escribirse en Java” no es un requerimiento porque es una decisión de implementación. Pero una declaración como: “un usuario puede retirar de la biblioteca hasta 10 libros diferentes al tiempo” si es un requerimiento porque define un atributo de la clase usuario. | |

|  |
| --- |
| **Cuadro de texto** |
| En el siguiente gráfico se pueden observar los elementos que intervienen en el proceso de análisis y recopilación de requisitos. |

**Figura 3** *Proceso de análisis y recopilación de requisitos*

**

*Nota*: Adaptada de *Análisis de los stakeholders*. Grandes Pymes (2021). <https://www.grandespymes.com.ar/2021/04/10/analisis-de-los-stakeholders/>

Imagen: 228131\_i222

|  |  |
| --- | --- |
| **Tipo de recurso** | **Acordeón tipo 1** |
| **Introducción** | Los requerimientos se deben precisar y eso implica entender el entorno en el que el *software* funcionará y definir en forma detallada las declaraciones que especifican lo que hará. Cinco criterios evalúan que los requerimientos sirvan: |
| Imagen: 228131\_i223 | |
| 1. Cada requerimiento debe ser validado por el patrocinador o patrocinadores. La mejor forma de lograr la obtención de requisitos es involucrar a los usuarios al interactuar con ellos, entre más se hable y escuche, mayor claridad habrá a medida que se logre acumular información sobre el *software* que se va a realizar. | |
| 1. Cada propiedad del dominio (entorno donde va a funcionar el *software*, no la máquina o sistema operativo) también debe ser validado por el patrocinador o patrocinadores. | |
| 1. La especificación de requerimientos no se restringe al entorno o se refiere al futuro. | |
| 1. Debe haber manera de comprobar que la implementación de la especificación satisface los requerimientos en el entorno donde trabajará. Para esto se diseñan y prueban prototipos. | |
| 1. Las especificaciones y propiedades del entorno (dominio) son consistentes. | |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Tarjetas animadas** |
| **Introducción** | Un requerimiento debe cumplir ciertos criterios y características, entre otros debe ser único, verificable, claro, realista y posible (Equipo Editorial Project Management, 2019). Las cualidades clave de los requerimientos son: |
| Benchmarking concept. Idea of business development and improvement.  Imagen: 228131\_i224 | Cuantificable en términos de mediciones. Por ejemplo, una declaración como “el tiempo de respuesta debe ser corto” no es cuantificable, pero si se cambia por “el tiempo de respuesta debe ser menor a 2 segundos” se puede medir y comprobar. |
| Ilustración del concepto completo  Imagen: 228131\_i225 | Debe ser completo. Los requerimientos deben definir todas las propiedades y restricciones del *software*. En la práctica se logran al cumplir las pautas para definición de requerimientos como asegurar que no falten referencias o funciones. |
| El hombre y la mujer trabajadores jóvenes eligen la marca de verificación y la ponen en el cuadro de encuesta de verificación, ilustración vectorial de dibujos animados  Imagen: 228131\_i226 | Corrección o validez. Se debe unificar el significado de los requerimientos. A menudo implica el cumplimiento de políticas, leyes y normas. |
| Ilustración de certificación iso con personas y bloc de notas  Imagen: 228131\_i227 | Consistencia. Al involucrar interesados con diferentes opiniones sobre el problema, se presentan contradicciones en las etapas tempranas de desarrollo. A través de la negociación y priorización se resuelven los conflictos y alcanza un acuerdo. |
| Imagen: 228131\_i228 | Unicidad en la definición. Las palabras y significados escritos en las declaraciones deben significar lo mismo para todos los interesados del proyecto. Un glosario elimina las ambigüedades semánticas. |
| Gente de negocios que estudia la lista de reglas, guía de lectura, lista de verificación.  Imagen: 228131\_i229 | Pertinencia. Definir claramente el problema a solucionar es fundamental, los requerimientos deben ser relevantes a las necesidades de los patrocinadores sin restringir al equipo de trabajo. |
| Diseño de ilustración de estilo plano del ciclo de vida del proyecto  Imagen: 228131\_i230 | Factibilidad. Los requisitos deben especificarse para que puedan implementarse utilizando los recursos disponibles, dentro del presupuesto y el cronograma. |
| Programming and coding, website, webpage optimization. Back end development, software development process, backend app developer concept. Vector isolated concept creative illustration  Imagen: 228131\_i231 | Trazabilidad. Ayuda a identificar el impacto de los cambios en otros partes del *software* y a evaluar cómo se debe propagar este cambio. |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Contar con una buena clasificación de requisitos permite establecer las necesidades de un proyecto de desarrollo cubriendo sus aspectos más importantes.  Los requerimientos del *software* a desarrollar pueden clasificarse según distintas reglas. En la siguiente tabla se detalla dicha clasificación. | |
| waterfall model flowchart hand drawing on blackboard  Imagen: 228131\_i232 | |

**Tabla 2***Clasificación de los requerimientos*

|  |
| --- |
| Clasificación de los Requerimientos |
| * Del negocio que no son negociables: normas y estándares aplicables. * Funcionales: qué es lo que el *software* hará. * No-Funcionales: características y restricciones como precisión, rendimiento, seguridad, usabilidad, mantenibilidad, calidad y mutabilidad. * De interfaz externa y restricciones que dependen de la tecnología. |
| Requisitos del nivel de:   * + Alcance – relativos a los objetivos del negocio.   + Dominio – relativos al área del problema.   + Producto – relativos al producto de *software*.   + Diseño – que se va a construir. |
| * Primarios: obtenidos de los patrocinadores e interesados. * Derivados: derivados de los primarios. |
| Otras clasificaciones:   * Requerimientos del negocio vs. técnicos. * Requerimientos del producto vs. del proceso; por ejemplo, necesidades del negocio contra el cómo los usuarios interactúan con el *software*. * Requisitos basados en roles, definidos por clientes, usuarios, equipo técnico o de seguridad. |

*Nota.* Adaptada de Mohapatra y Rath (2020).

## Herramientas y técnicas

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | **Slider presentación** | |
| **Introducción** | Los usuarios nunca saben lo que quieren del producto final, por esto se enfatiza en “escuchar y conversar” con ellos: Hay varias técnicas para este diálogo debido a que los usuarios no son un grupo monolítico, tienen diferentes necesidades, quieren diferentes funcionalidades y ni siquiera utilizarán los mismos equipos. | |
| Los usuarios se clasifican en grupos dependiendo del rol que van a tener cuando estén utilizando el *software* final. | | User personal profile characters set for social network. Employees, corporate male and female workers portraits. Team member, avatar metaphors. Vector isolated concept metaphor illustrations  Imagen: 228131\_i233 |
| La especificación se hace con historias y casos de usuario. Como el alcance del requerimiento ya está acordado y se han definido cuáles son los requerimientos, estos se analizan y se especifican usando estos dos métodos, entre otros. | | Social network reviews and comments from customers with workspace.  Imagen: 228131\_i234 |
| Análisis: como se obtienen muchos requerimientos, para organizarlos se mapean las historias de usuario (*Story Mapping Technique*) agrupando los usuarios por categorías y asignando prioridades. | | Sígueme en diseño de temática social y empresarial.  Imagen: 228131\_i235 |
| Finalmente, se averigua con los usuarios si el prototipo mostrado hace lo que se busca, validando la retroalimentación recibida para que estos entiendan el concepto. | | Business people studying list of rules, reading guidance, making checklist. Vector illustration for company order.  Imagen: 228131\_i236 |
| Se da la oportunidad de sugerir e introducir cambios o ajustes sobre la funcionalidad del *software*. | | Ilustración del concepto de prueba móvil  Imagen: 228131\_i237 |
| El prototipo es importante, es una maqueta del producto que se muestra al grupo de interés para que opinen, critiquen, sugieran y retroalimenten al equipo de trabajo: esta validación corrige desviaciones en el alcance del proyecto. | | Aplicación de planificación de creative web designer y desarrollo de marco de diseño de plantilla para teléfono móvil experiencia de usuario concepto ux  Imagen: 228131\_i238 |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Se usan plantillas normalizadas que sirven para articular el problema y ayudan en la delimitación de la visión del producto. Bautizar el proyecto con un nombre que contenga la esencia, le da vida al proyecto, facilita referirlo con el equipo de trabajo y la documentaciónIlustración de vector plano de proceso de trabajo de planificación de equipo de negocios. compañeros de dibujos animados hablando, compartiendo pensamientos y sonriendo en la oficina de la empresa. concepto de flujo de trabajo y trabajo en equipo.  Imagen: 228131\_i239 | |
| Para afianzar el conocimiento, se recomienda ver las plantillas normalizadas para uso en la recogida y análisis de requisitos. | |

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | **Pestañas o tabs horizontales** | |
| **Introducción** | Las técnicas de interacción con los patrocinadores y clientes son individuales o grupales, se reúnen los usuarios para preguntarles qué es lo que quieren o buscan de la aplicación, procesos del negocio o necesidades de la organización. | |
| **Entrevista** | Dado el carácter informal de la entrevista se hace inicialmente antes de pasar a técnicas más estructuradas, y es efectiva para desarrollar un reporte del punto de vista del cliente.  Antes de realizar la entrevista se presenta la agenda y las preguntas generales, pero a medida que se va desarrollando pueden surgir preguntas más abiertas dejando que libremente sean formuladas por los asistentes sin desviarse de la idea o intención central. | Pareja joven discutiendo en casa. hombre y mujer sentados en el sofá y hablando ilustración plana  Imagen: 228131\_i240 |
| **Talleres (Workshop)** | Como la entrevista, pero enfocada en la parte funcional, reúne los interesados clave: patrocinadores y expertos del dominio, que mediante trabajo colaborativo y discusiones resuelven asuntos sobre diferencias de opinión, necesidades, conflictos y requerimientos no clarificados. | Estudiantes viendo seminarios web en computadora, estudiando en línea  Imagen: 228131\_i241 |
| **Observación** | No siempre es sencillo para los usuarios expresarse o no saben cómo hacerlo, entonces es necesario ver la operación directamente y cómo se desempeñan al manejar las aplicaciones. Observar al usuario tiene la ventaja de entender qué partes usa de una aplicación o cuáles se les dificultan o le hacen falta. | Ilustración del concepto de comisión editorial  Imagen: 228131\_i242 |
| **Cuestionarios** | Si el grupo es grande o está en otra ciudad, se consignan las preguntas y se envían para diligenciarse. Las preguntas deben ser claras, correctas y enfocadas hacia los requerimientos específicos del encuestado. Recibir retroalimentación ayuda a mejorar la calidad y consistencia de las preguntas. | Ilustración del concepto de tomar notas  Imagen: 228131\_i243 |
| **Observación independiente** | La hace el equipo de trabajo antes de hablar con los usuarios, es útil cuando se hace una aplicación para reemplazar, actualizar o añadir funcionalidad a un *software* en uso, tener una idea general de cómo ha sido utilizado. Es decir, es consultar la documentación del *software* y hablar con quienes mejor lo trabajen. | Concepto de landing page de búsquedas  Imagen: 228131\_i244 |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| El tipo de aplicación determina cuál técnica de interacción usar: si es masiva se hacen entrevistas, grupos de enfoque y cuestionarios; si es individual o para un grupo pequeño se hace observación o, si se es nuevo en el dominio del *software*, hay que hacer talleres y observaciones para adquirir conocimiento y entender el dominio. | |
| **Figura 4** *Técnicas para recopilar requerimientos*    *Nota.* Reproducida de *Técnicas para la recolección de requisitos*. Team Asana (2021). <https://asana.com/es/resources/requirements-gathering>  Imagen: 228131\_i245 | |

**ANEXO 2 INFOGRAFÍA 1**

**ANEXO 3 INFOGRAFÍA 2**

**Mapeo de historias de usuarios**

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Los requerimientos desde la perspectiva del usuario se cuentan como una historia que resume el rol y los cómo, por qué y cuándo de una característica del *software*, mediante una descripción sencilla y corta, al menos dos oraciones escritas y diálogos sobre la funcionalidad o necesidad del cliente que se hagan con el equipo de desarrollo; es una forma conveniente, efectiva y popular para escribir requerimientos. | |
| Concepto de infografía de web plana de negocios de vista superior de mesa de oficina. el personal alrededor de la mesa informe de análisis de trabajo tablet portátil fondo vacío. piense en la planificación del informe. colección de personas creativas.  Imagen: 228131\_i246 | |

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Tarjetas avatar | |
| **Introducción** | El propósito del mapeo de historias es facilitar el descubrimiento de productos y la priorización de los esfuerzos de desarrollo. Se logra colocando las acciones y tareas del usuario en un mapa que se usa para mantenerlas en contexto (Digité, 2020). Para escribir historias de usuario usa una tarjeta con los tres componentes centrados en el valor que se busca obtener. | |
| **Rol**  Persona o grupo enfocado en su rol o papel dentro del proyecto. | | IlustraciÃ³n del concepto de equipo creativo  Imagen: 228131\_i247 |
| **Lo que se quiere**  (Objetivo, comportamiento, funcionalidad o característica) lo que se busca que haga o cumpla esta historia de usuario. | | Imagen: 228131\_i248 |
| **Para poder**  (Motivo, razón, valor, beneficio) escribir este propósito ayuda a los desarrolladores a entender mejor la funcionalidad, aunque algunas veces sea evidente.  . | | Imagen: 228131\_i249 |
| **El por qué**  Después del **haga-esto** en la historia de usuario define los criterios de aceptación o de satisfacción. | | Concepto de retroalimentaciÃ³n plana ilustrado  Imagen: 228131\_i250 |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| La historia de usuario es un compromiso con el cliente que evitará diferencias sobre lo que se quiere. En la siguiente figura se encuentra la tarjeta para definir y sistematizar las historias de usuarios. | |
| **Figura 5** *Modelo de tarjeta para capturar historias de usuario*    *Nota.* Adaptada de *Plantilla para las historias de usuario.* Scrumízate (2017).<http://scrumizate.com/post/58/plantilla-para-las-historias-de-usuario> | |

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | **Slider presentación** | |
| **Introducción** | Los criterios de aceptación consignados en la tarjeta permitirán la futura evaluación de la implementación, las pruebas a ejecutar y los puntos de vista para entender el trabajo a realizar, delimitar lo indispensable y definir cómo las funciones del sistema van a interactuar con el usuario. | |
| Cuando varias historias de usuario tienen funcionalidad relacionada se tienen las épicas, que son bloques funcionales muy amplios y a menudo de alto nivel dentro del negocio. | | IlustraciÃ³n del concepto de tablero scrum  Imagen: 228131\_i252 |
| Las épicas se descomponen en historias más granulares, pero relacionadas entre sí, a su vez, estas se agrupan en temas que ayudan a los diseñadores a conformar el mapa de historias, que es la representación tabular del flujo de trabajo (eje horizontal) y la prioridad (eje vertical). | | Concepto de infografÃ­a de proceso plano  Imagen: 228131\_i253 |
| Las épicas se ponen en la parte superior y se ubican verticalmente con prioridad descendiente, de esta forma el equipo de trabajo descompone los temas en tareas y estiman el esfuerzo y recursos necesarios para su implementación. | | Grupo creativo discutiendo ideas para proyecto en sala de juntas  Imagen: 228131\_i254 |
| En discusión con el cliente se refinan los requerimientos y se asigna la prioridad para implementación, escogiendo cuáles son más útiles al negocio y cuáles deben implementarse inicialmente. | | Imagen: 228131\_i255 |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| El mapa de historias captura las perspectivas de los usuarios, descompone los requerimientos de las historias en una secuencia como parte del flujo de trabajo, agrupándolas en temas relacionados y dándoles prioridad desde el punto de vista de usabilidad. | |
| **Figura 6** *Tablero con mapa de historias*    *Nota*. Adaptada de *User stories with examples and a template.* REHKOPF (2021). <https://www.atlassian.com/agile/project-management/user-stories> | |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Las historias de usuario son útiles para involucrar a los clientes en las actividades iniciales de obtención de requisitos, pero pueden carecer de integridad, haciendo difícil juzgar si son suficientes para cubrir todos los requisitos esenciales del sistema. | |
| Imagen: 228131\_i257 | |

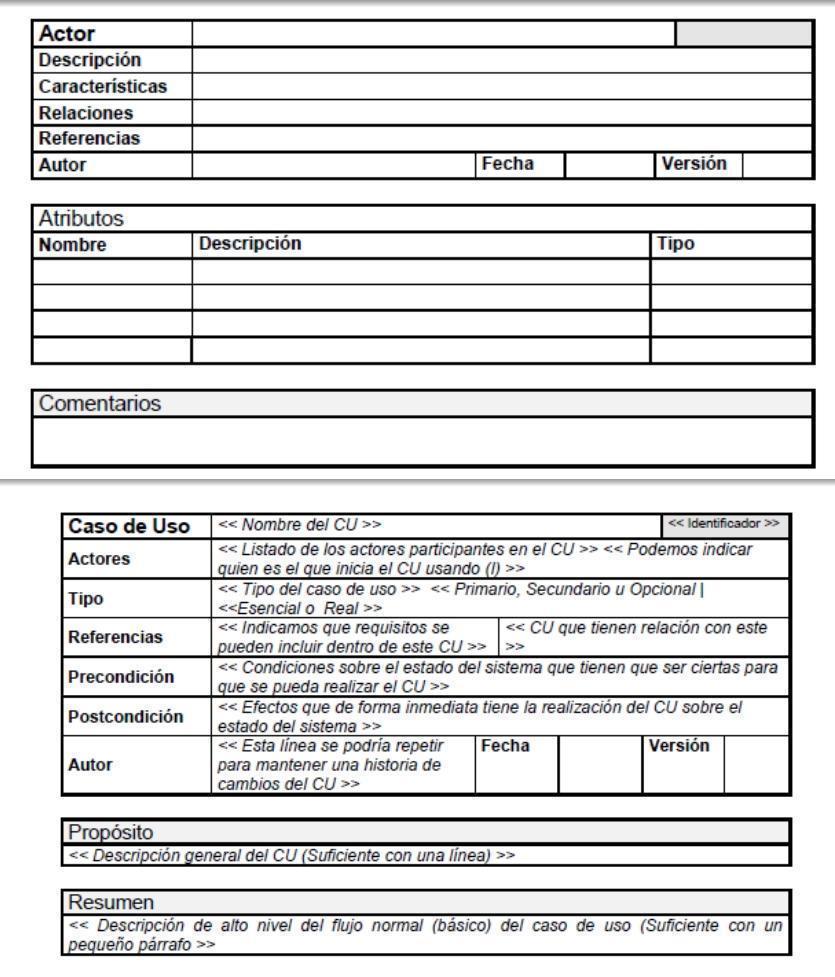
**Diagramas de casos de uso.**

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Se introduce el Lenguaje de Modelado Unificado (*UML*) un estándar usado para la representación visual tanto de los objetos, estados y procesos dentro de un sistema. Puede parecer lejano a la analítica de datos, pero al servir como lenguaje de modelado de *software*, ayuda a los desarrolladores a presentar la descripción del sistema de forma comprensible para los demás interesados. | |
| Forma  Descripción generada automáticamente con confianza media  Imagen: 228131\_i258 | |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Rutas / Pasos. Verticales 1** |
| **Introducción** | El caso de uso es una descripción formal de las acciones de un sistema desde el punto de vista del cliente/usuario y representa cada uno de los requisitos funcionales del sistema, también especifica el comportamiento de dicha funcionalidad de dos formas: básica sin errores y extendida con los errores que puedan ocurrir. |
| Imagen: 228131\_i259 | |
| **Botón 1** | **Funcionalidad**  La funcionalidad se escribe en una plantilla con lenguaje natural e incluye información:   * Identificador * Nombre del caso * Actores involucrados * Precondiciones, lo que tiene que cumplirse o hacerse antes de la ejecución del caso * Postcondiciones, lo que se debe cumplir después de ejecutar el caso de uso   La descripción del flujo de eventos determina qué hace el sistema y qué interacción tiene con los actores, para entregar la funcionalidad completa. |
| **Botón 2** | **Reglas para construir casos de uso:**   * Relacionar mínimo un actor. * Cada actor representa un rol, esto significa que si un actor interactúa con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores en los diferentes casos de uso. * Es un iniciador, es decir que responde a acciones del actor e inicia una secuencia de acciones. * Su resultado es relevante, depura los requisitos que tienen valor intrínseco. Si no aporta valor se llama caso de abuso, ya que especifica cosas que el actor no debería hacer. |
| **Botón 3** | **Plantilla de casos de uso**  Los casos de uso enfocan el proceso de desarrollo debido a que varias actividades posteriores, como el diseño y pruebas, parten de ellos. |

|  |
| --- |
| **Cuadro de texto** |
| A continuación, se encuentra una plantilla que contiene las especificaciones esenciales para escribir casos de uso. |

**Tabla 3***Plantilla para escribir el caso de uso*



*Nota*. Adaptada de *Casos de uso y diagramas de casos de uso*. Universitat Politècnica de València-UPV (2021). <https://youtu.be/iFcDoP6jEeE>

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Los diagramas de casos de uso ilustran la funcionalidad proporcionada por una unidad del sistema, representan gráficamente los diferentes casos de uso del sistema y muestran las interacciones entre estos y los actores. No incluye detalles sobre implementación, interacción interna o restricciones de orden entre estos. Capturan los requisitos funcionales del sistema a desarrollar y sus elementos:   * Entorno del sistema y nombre del diagrama. * Casos de uso, el conjunto de requisitos funcionales. * Actor o actores externos al sistema y que interactúan con este. * Líneas de comunicación entre los elementos y que expresan una relación de inclusión, extensión y herencia. * La relación entre los varios actores solo es de herencia.   **Figura 7** *Ejemplo de diagrama de caso de uso*    *Nota*. *Reproducida* de *Casos de uso y diagramas de casos de uso*. Universitat Politècnica de València-UPV (2021). <https://youtu.be/iFcDoP6jEeE> | |
| En la figura anterior se ven los elementos del diagrama: el rectángulo que representa el sistema y dentro de este los casos de uso que indican sus funciones; las figuras representan los actores externos al sistema. Las relaciones entre actor y caso de uso se dibujan con una línea simple. Las flechas etiquetadas representan las relaciones entre los casos de uso. La relación "incluir" indica que un caso de uso es necesitado por otro para poder cumplir una tarea y que el primero puede realizar todos los eventos del caso incluido. La relación "extender" indica opciones que pueden o no ejecutarse para cierto caso de uso.  Por ejemplo, solo el ACTOR 1 (Socio) puede ejecutar todos los casos de uso, el ACTOR 2 (Cliente) no puede ejecutar el caso de uso REPETIR ÚLTIMA COMPRA. El caso PAGO CON TARJETA DEBITO siempre incluye el caso IDENTIFICAR LA TARJETA, pero el caso OFRECER AYUDA es una alternativa, finalmente IDENTIFICAR LA TARJETA se puede hacer con IDENTIFICAR CON PIN o IDENTIFICAR POR CONTACTO que involucra el actor LECTOR NFC.  Los diagramas sirven para identificar actores externos y delimitar el sistema, indicando las relaciones y complementándose con las plantillas que describen cada uno de los casos de uso ilustrados. | |

|  |
| --- |
| **Cuadro de texto** |
| Para afianzar el aprendizaje del lenguaje UML se invita a ver el siguiente video |

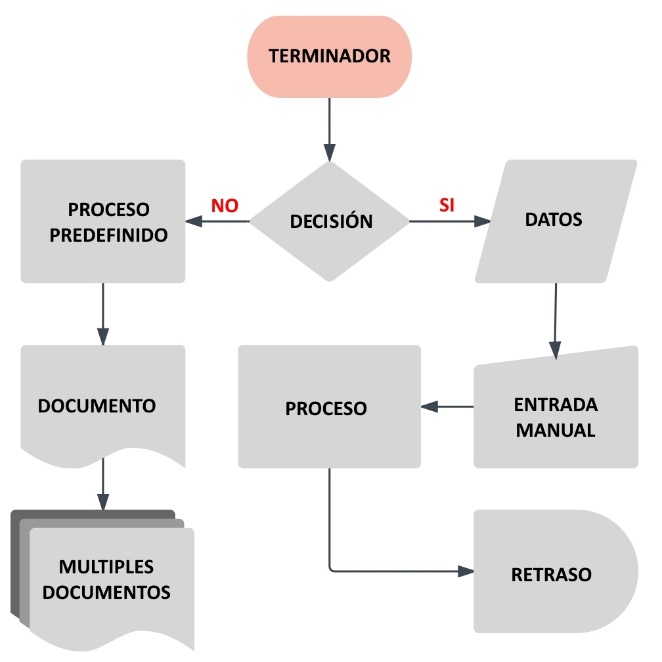
**VIDEO INSTRUCCIONAL 228131\_V2**

**Mapas de procesos y diagramas de flujo.**

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Los mapas de procesos se muestran visualmente mediante pasos y decisiones, también se conocen como diagramas de flujo que indican el sentido de las tareas, las decisiones que deben tomarse y las relaciones fundamentales entre los pasos. Ayuda a organizar los procesos y los hace entendibles. Diagrama de flujo metodologÃ­a de la informaciÃ³n icono de operaciÃ³n  El uso de mapas de procesos mejora la comunicación y documentación, ayudando al equipo de trabajo a identificar cuellos de botella, repeticiones y demoras.  Imagen: 228131\_i302 | |

|  |
| --- |
| **Cuadro de texto** |
| La figura muestra los elementos del diagrama de flujo que incluyen acciones, pasos de actividades, puntos de decisión, funciones, entradas o salidas, personas involucradas, cuantificación del proceso y tiempo necesario. Cada elemento se representa con un símbolo específico, como una flecha, un círculo, un diamante, un recuadro, un óvalo o un rectángulo. |

**Figura 8***Símbolos del diagrama de flujo (procesos)*



*Nota.* Adaptada de *¿Cuáles son tus necesidades de creación de mapas de procesos?* Lucidchart (2021). <https://www.lucidchart.com/pages/es/que-es-la-creacion-de-mapas-de-procesos>

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Carrusel de tarjetas | |
| **Introducción** | Los mapas de procesos facilitan la comprensión de sus características, permitiendo formular preguntas importantes para mejorar y generar datos útiles para la solución de problemas. Tipos de mapas de procesos: | |
| Diagrama abstracto del diagrama de flujo de negocio sobre fondo blanco  Imagen: 228131\_i304 | | |
| Mapa de proceso de actividades: representa las actividades que agregan valor y las que no. | | https://cdn-icons-png.flaticon.com/512/7018/7018534.png  Imagen: 228131\_i305 |
| Mapa de proceso detallado: precisa más cada paso del proceso. | | https://cdn-icons-png.flaticon.com/512/5672/5672909.png  Imagen: 228131\_i306 |
| Mapa de documentos: las entradas y salidas de un proceso. | | https://cdn-icons-png.flaticon.com/512/7527/7527163.png  Imagen: 228131\_i307 |
| Mapa de proceso de alto nivel: representación de un proceso que involucra las interacciones entre proveedor, entradas, procesos, salidas y clientes. | | https://cdn-icons-png.flaticon.com/512/1556/1556231.png  Imagen: 228131\_i308 |
| Mapa de proceso representado: muestra el estado actual o futuro de los procesos. | | https://cdn-icons-png.flaticon.com/512/1459/1459073.png  Imagen: 228131\_i309 |
| Mapa de carriles: separa las responsabilidades del subproceso. | | https://cdn-icons-png.flaticon.com/512/1459/1459091.png  Imagen: 228131\_i310 |
| Diagrama de flujo de trabajo: proceso de trabajo mostrado en un formato de "flujo". | | https://cdn-icons-png.flaticon.com/512/2519/2519375.png  Imagen: 228131\_i311 |

## Salidas y entregables

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Hasta el momento, se han definido los requerimientos para definir el *software*, desde la concepción del producto hasta las fases iniciales del desarrollo. El resultado de este trabajo son varios documentos, incluyendo el acuerdo comercial y las plantillas siguiendo las buenas prácticas. | |
| **Figura**  *Procesos de la ingeniería de requerimientos*    *Nota*. Adaptada de *Proceso de ingeniería de requisitos*. Sommerville, (2016).  Imagen: 228131\_i312 | |

|  |  |
| --- | --- |
| **Tipo de recurso** | Rutas / Pasos. Verticales 1 |
| **Introducción** | Al ser esta la fase inicial del proyecto, sus salidas alimentan las etapas sucesivas con varios productos: |
| IlustraciÃ³n del concepto de procesamiento  Imagen: 228131\_i313 | |
| **Desarrollar la visión general del sistema** | Requisitos generales del sistema, especificar las características principales del *software* a desarrollar.  Casos de uso del sistema, describir cómo utilizarán el *software* los clientes. |
| **Documentar los requisitos del sistema** | * Diagramas de historia, detallar cómo debe interactuar el *software* con los usuarios, para que puedan realizar sus tareas dentro del modelo de negocio. * Requisitos funcionales, Enunciarlos para que a partir de ellos pueda implementarse una solución. Tipos: * Requisitos de información, describen que deberá almacenar y gestionar el *software* para dar soporte a los procesos de negocio. * Reglas de negocio, incluyen las reglas o restricciones de cumplimiento obligatorio. * Requisitos de conducta, es decir, las capacidades o interacciones con el usuario. * Requisitos no funcionales, las características de calidad del sistema. * Requisitos de integración, establece cuáles servicios deben integrarse al *software*. * Restricciones técnicas, limitaciones de carácter tecnológico que debe cumplir el sistema. |
| **Analizar los requisitos del sistema** | * Arquitectura del sistema. Son los estándares y modelos que impactan el resto del desarrollo. * Modelo estático. Constituye el modelo del sistema a desarrollar. * Modelo dinámico/funcional. Detecta problemas con los casos de uso y el diseño de la capa de lógica. * Interfaz de usuario. Relevante para la validación de los requisitos con clientes y usuarios. * Interfaz de servicios. Importante si el sistema va a tener una interfaz que puede vulnerarse. * Problemas en los requisitos. Identificados mediante el análisis y priorización. |
| **Verificar la calidad de los requisitos del sistema** | * Problemas en los requisitos: surgen al verificar la calidad de estos. |
| **Validar los requisitos del sistema** | * Actas de reuniones, recogen los resultados obtenidos en las sesiones de validación. Si hay aceptación de los interesados, se registra con actas y contratos. * Problemas en los requisitos. identificados y validados entre los interesados. |
| **Registrar problemas en los requisitos** | * Problemas en los requisitos. Se anotan para aclarar su significado o suprimirlos si están duplicados. |
| **Registrar la trazabilidad de los requisitos** | * Especificación de Requisitos del Sistema (ERS). Elabora las matrices de trazabilidad que reflejan las dependencias entre los requisitos y elementos que la componen. Estas se registran dentro del ERS. |

# Herramientas y técnicas para recolección de requisitos

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| La definición de requisitos identifica los aspectos clave del *software* y descarta los irrelevantes, es decir, los que no agregan valor. A continuación, se revisan algunas técnicas para analizar requerimientos como FURPS+ y MoSCoW, el prototipado como un modelo inicial, las reuniones JAD (Desarrollo Conjunto de Aplicaciones) como práctica del grupo de desarrollo que busca en varios días concretar las necesidades del sistema a desarrollar. | |
| **Figura 10** *Comic sobre las reuniones para definir requerimientos*    *Nota.* Adaptada y traducida de Dilbert Comics (2006), de Adams, S. (29 de enero de 2006). https://dilbert.com/search\_results?year=2006 | |

|  |  |
| --- | --- |
| **Tipo de recurso** | Rutas / Pasos. Verticales 1 |
| **Introducción** | La adquisición y la comprensión de los requerimientos no es un proceso fácil, en ocasiones se presentan situaciones como las siguientes: |
| Imagen: 228131\_i315 | |
| **Botón 1** | * Los interesados no saben lo que quieren o esperan del *software*, tienen un conocimiento muy general. Entonces, se deben usar mecanismos para averiguar qué quieren que haga el sistema, definir lo qué es posible y delimitarlo. |
| **Botón 2** | * Los interesados expresan los requerimientos con sus términos y conocimientos de su trabajo, si los ingenieros no tienen experiencia en el dominio del cliente no los entenderán. |
| **Botón 3** | * Los diferentes roles de interesados pueden expresarse de forma diferente, de manera que el grupo de desarrolladores debe descubrir las fuentes de requerimientos e identificar similitudes y conflictos |
| **Botón 4** | * Las prioridades de los requerimientos cambian en el tiempo porque los ambientes económico y empresarial son dinámicos. También pueden surgir nuevos requerimientos o interesados en el proyecto. |
| **Botón 5** | * Los patrocinadores son la principal fuente de requerimientos, deben participar directamente en el proceso. |

**Figura 11** *Entrevista, la técnica más utilizada*



*Nota.* Adaptada de *Entrevistas eficaces para el trabajo de requerimientos.* Fatto consultorías y sistemas (s.f.) [Video] Youtube. https://www.youtube.com/watch?v=C\_r-hTknkf8

## Entrevistas con patrocinador y usuarios

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Slider Presentación | |
| **Introducción** | Las entrevistas son la técnica más utilizada en todo proceso de desarrollo de *software*, porque es una forma de comunicación natural entre personas. | |
| La técnica de entrevista se desarrolla en tres fases: preparación, realización y análisis. Una entrevista no se improvisa, previamente se deben hacen tareas como:   * Estudiar el dominio del problema * Seleccionar a las personas clave dentro de los interesados y ajustar sus agendas * Delimitar el objetivo y contenido * Planificar la logística | | Imagen: 228131\_i317 |
| Las entrevistas son un intercambio de información verbal entre personas y pueden ser individuales o colectivas. Las preguntas, previamente escritas, se formulan de manera formal o informal; si el entrevistador tiene experiencia, se pueden hacer de manera espontánea. Las respuestas se utilizan para derivar requisitos. | | Imagen: 228131\_i318 |
| El análisis de documentación consiste en obtener la información sobre los requerimientos a partir de documentos que ya están elaborados, por ejemplo, manuales del *software* que se está usando. Esta técnica es útil cuando no hay expertos en la materia o los entrevistados salieron de la organización. | | Imagen: 228131\_i319 |
| Las mesas de trabajo (*Workshops*) se utilizan para determinar, establecer prioridades y hacer cierre de los requisitos para el desarrollo del *software*. Se enfocan en usuarios claves y expertos seleccionados, se realizan durante un período corto, pero intensivo; se usan para generar ideas de nuevas características o productos, resolver conflictos y llegar a un consenso, o para revisar los requisitos. | | Imagen: 228131\_i320 |

## Técnicas de creatividad grupal - Lluvia de ideas

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| La lluvia de ideas es una actividad grupal de creatividad, usada para generar y recopilar ideas relacionadas con un mismo tema, en un ambiente libre de críticas o juicios. En la obtención de requisitos, se utiliza para profundizar el detalle sobre el tema. En la sesión se intercambian ideas sin orden ni filtro y luego se valoran. Sus fases son:   * Preparación * Generación * Consolidación (revisar, descartar y priorizar ideas) * Documentación | |
| **Figura 12** *Lluvia de ideas, creatividad en libertad*    *Nota*. Reproducida de *Reglas de la lluvia de ideas*. Hansen (2021). <https://www.wrike.com/blog/rules-of-brainstorming-managers-guide-producing-great-ideas/>  Imagen: 228131\_i321 | |

|  |  |
| --- | --- |
| **Tipo de recurso** | Rutas / pasos. Verticales 1 |
| **Introducción** | La capacidad de crear nuevas ideas es una potencialidad desde la infancia. Por lo general, el primer paso en la lluvia de ideas es que todos se reúnan en un lugar tranquilo y comiencen a pensar (Hansen, 2021). Por su naturaleza libre y espontánea, la lluvia de ideas tiene cuatro características: |
| Imagen: 228131\_i322 | |
| **Botón 1** | **Cantidad antes que calidad**  Como se busca recoger tantas ideas como sea posible, es importante que estas fluyan libremente, aunque al final muchas se desechen. |
| **Botón 2** | **No a las críticas, discusiones o comentarios durante la sesión**  Esta regla debe ser respetada en todo momento para que el flujo de ideas no se perturbe o detenga. |
| **Botón 3** | **Registrar todas las ideas**  Que la sesión pase a la siguiente fase de evaluación sin haber sido censurada o filtrada. Ignorar ideas desmotiva a los participantes para seguir contribuyendo. |
| **Botón 4** | **Pensar con originalidad e inspirarse mutuamente**  Como surgen muchas ideas independientes, otras nuevas surgen a partir de ellas. |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Otra técnica es el Diseño Conjunto de Aplicaciones (*JAD*) como alternativa a las entrevistas individuales. Ayuda a los clientes y usuarios a formular problemas y a explorar soluciones, involucrándolos dentro del desarrollo, mediante talleres colaborativos. | |
| Concepto creativo del diagrama de las ideas de la bombilla  Imagen: 228131\_i323 | |

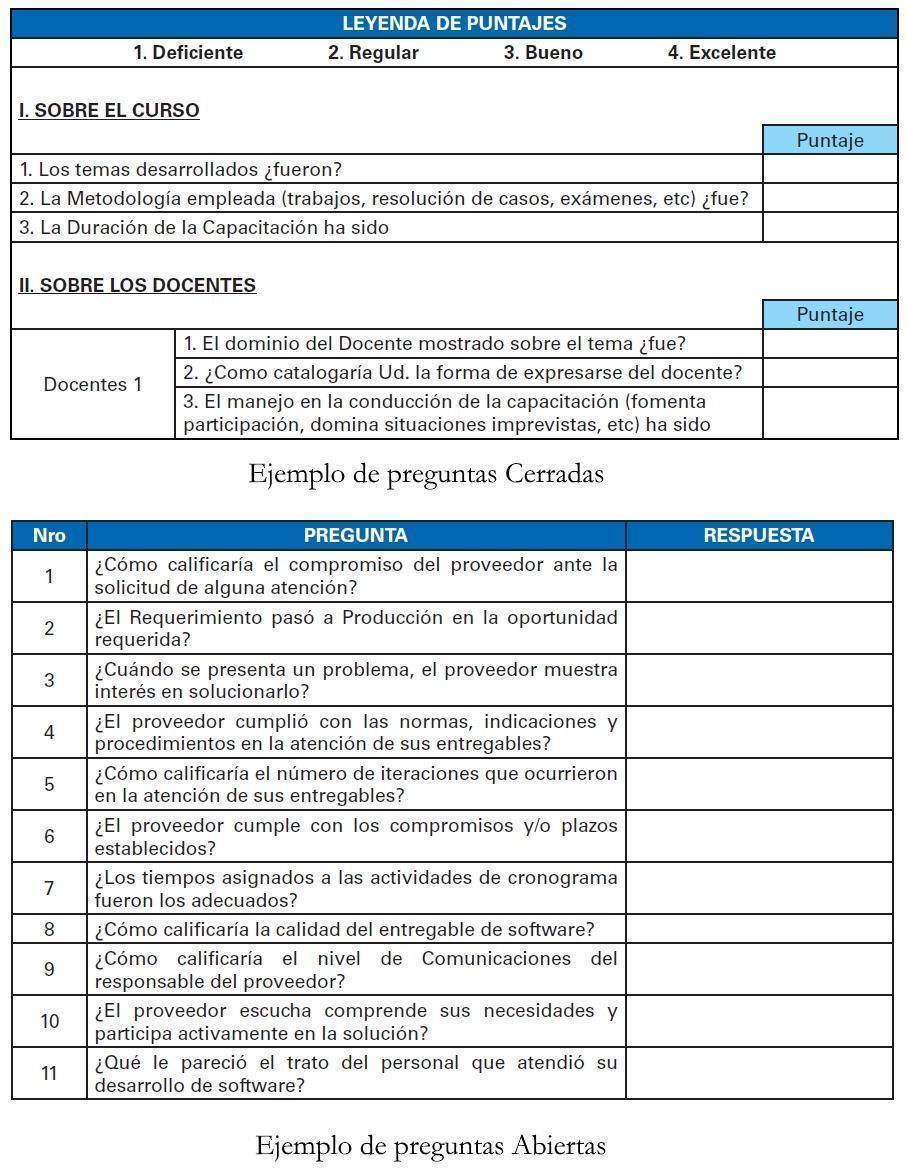
|  |  |
| --- | --- |
| **Tipo de recurso** | Rutas / pasos. Verticales 1 |
| **Introducción** | El *JAD* lleva a menores tiempos de desarrollo y mayor satisfacción del cliente involucrado durante todo el proceso. Tiene cuatro principios: dinámicas grupales, comunicación visual (diagramas, transparencias, multimedia, herramientas CASE), un proceso organizado y racional y la documentación tipo WYSIWYG (*what you see is what you get,* lo que se ve es lo que se obtiene). Durante las sesiones se trabajan los documentos a manera de diseño en fases. |
| Imagen: 228131\_i324 | |
| **Botón 1** | **Adaptación**  Establece un equipo de trabajo identificando los componentes y responsabilidades, buscando consenso entre las necesidades de los usuarios y los servicios del *software* en producción. |
| **Botón 2** | **La sesión JAD consiste en**  Presentación, definición de objetivos y requisitos, documentación de temas abiertos y conclusión. Las sesiones pueden ser largas; durante estas se elaboran modelos empleando diagramas fáciles de entender, elaborados mediante herramientas CASE (*Computer Aided Software Engineering*). |
| **Botón 3** | **Para concluir se completa, revisa y valida la documentación**  El producto final es un conjunto de modelos para ser aprobados por los participantes. |

## Cuestionarios y encuestas

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Los cuestionarios o encuestas son un conjunto de preguntas orientadas a obtener información específica de los usuarios. Se solicitan anónimamente y las respuestas se analizan y catalogan de forma cualitativa o cuantitativa. Los resultados sirven para tomar decisiones por parte de quienes solicitaron la información.  Las preguntas formuladas pueden ser cerradas o abiertas.   * Cerradas: de selección múltiple entre varias alternativas. Estas preguntas deben ser objetivas con respuestas predefinidas, su evaluación es rápida y cuantitativa. * Abiertas: el encuestado responde la pregunta con sus propias palabras, pues tiene libertad de expresión, lo cual permite obtener información más detallada. Las respuestas son más difíciles de clasificar porque son cualitativas. | |

|  |
| --- |
| **Cuadro de texto** |
| En Tabla 4 se muestran ambos tipos de preguntas. En las cerradas se asignan puntajes para que cada usuario escoja, lo que permite que sean analizadas cuantitativamente. En las respuestas abiertas, el usuario puede responder en cada caso lo que considere desde su perspectiva. |

**Tabla 4** *Tipos de pregunta en los cuestionarios*



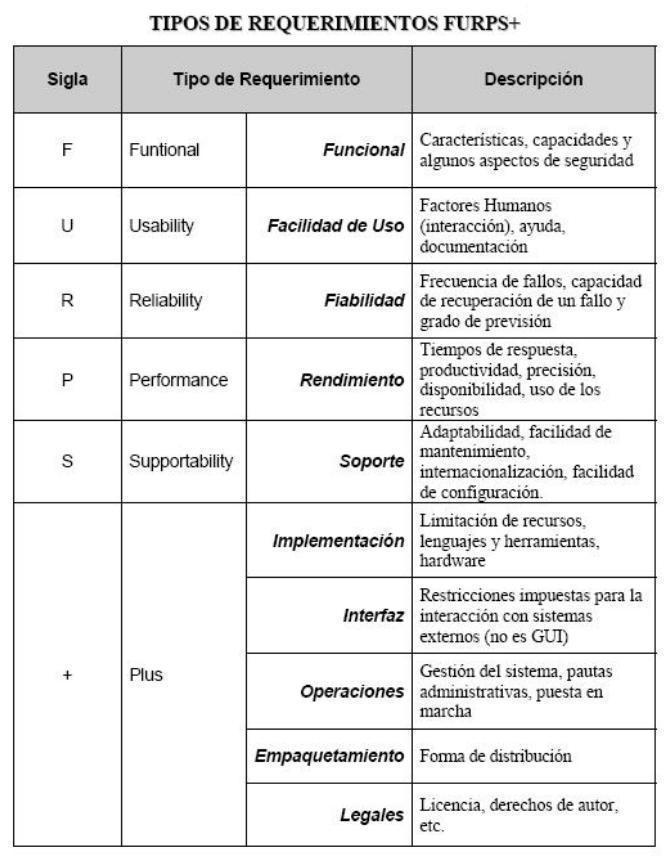
*Nota.* Adaptada de *Análisis y requerimientos de software: Manual autoformativo interactivo*. Wong, S. (2017). <https://hdl.handle.net/20.500.12394/4281>

## Análisis de tareas y de dominio

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| El análisis de requerimientos ejecuta las tareas que determinan las necesidades o condiciones que debe cumplir el *software*, ya sea un nuevo diseño o una versión de algo existente, teniendo en cuenta los conflictos entre requisitos de los diferentes roles. Analizar, documentar, validar y administrar los requisitos se hace en la fase de análisis. | |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| El análisis consiste en determinar si el requerimiento obtenido es claro, completo, conciso, no está duplicado, es válido, consistente y sin ambigüedades, resuelve conflictos y añade valor al proyecto. El proceso ha identificado las necesidades de los interesados, las implicaciones del *software*, representado los casos e historias de usuario, observado el sitio de trabajo y generado una cantidad de requisitos, | |
| **Figura 13** *Esquema del modelo FURPS+*    *Nota.* Adaptada de *FURPS+.QualidadeBR (10 de julio de 2008).* <https://qualidadebr.wordpress.com/2008/07/10/furps/>  Imagen: 228131\_i326 | |
| Para clasificar los requerimientos se usa la técnica *FURPS*+, (*Functionability, Usability, Reliability*, *Performance & Supportability)*: funcionalidad, facilidad de uso, confiabilidad, rendimiento y facilidad de soporte. El modelo clasifica dos categorías: requisitos funcionales (F) que especifican las funciones que el *software* debe hacer y requerimientos no-funcionales (*URPS*), que puntualizan atributos del sistema o del medio ambiente del sistema. El símbolo (+) representa restricciones no-funcionales que deben tenerse en cuenta, como diseño, implementación, legales e interfase y los que se derivan de la norma ISO-9126 como eficiencia, mantenibilidad y portabilidad. | |

**Tabla 5** *Restricciones del modelo FURPS+*



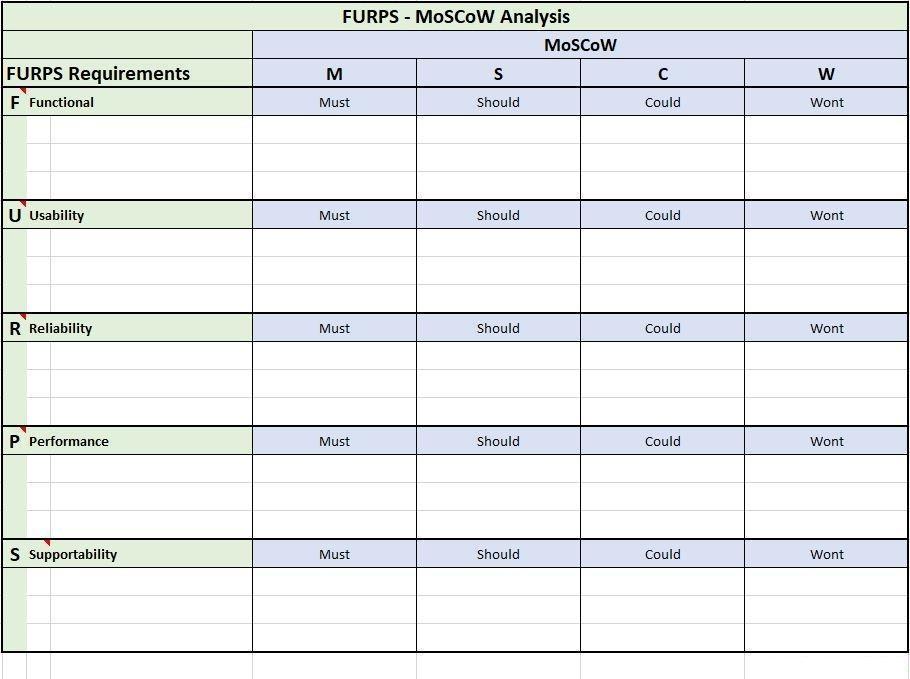
Nota. Muestra los tipos de requerimientos del modelo de calidad de FURPS y descripción de cada uno de ellos. CVUDES (2015)

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | | Pestañas o tabs verticales |
| **Introducción** | | Una vez clasificados, según el tipo, los requisitos se priorizan con el método *MoSCoW* que sirve para saber qué debe hacer primero, qué puede hacer después y qué no necesita hacer. El métodocorresponde a cuatro categorías: |
| Imagen: 228131\_i328 | | |
| ***M de Must Have*** | ***M de Must Have***  (Debe tener), requisitos indispensables, deben llevarse a cabo en primer lugar y no son negociables. | |
| ***S de Should Have*** | ***S de Should Have***  (Debería tener), agrupa los requisitos importantes a realizar una vez finalizada la categoría “debe tener”. Aportan valor agregado real y contribuyen a lograr los objetivos, pero pueden diferirse en el tiempo. | |
| ***C de Could have*** | ***C de Could have***  (Podría tener), requisitos que sería interesante tener si es posible, cuando haya tiempo al finalizar las dos primeras categorías. Su realización no debe afectar los demás requisitos. | |
| ***W de Won’t have*** | ***W de Won’t have but would like to in the future***  (No lo tendrá, pero de pronto en el futuro), son los requisitos secundarios que sería bueno realizar algún día, pero que por ahora se dejan de lado. | |

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| Las ventajas l método *MoSCoW* son:   * La forma de nombrar las categorías es pertinente y facilita la priorización. * La última categoría permite anotar las demandas del cliente y registrarlas para integrarlas en versiones o desarrollos posteriores. * Priorizar, facilita dividir los recursos adecuadamente.     Se deben equilibrar los requisitos en las categorías para evitar que todo sea prioritario, de lo contrario, algunos importantes como mejoras y mantenimiento, terminan en las últimas categorías. Ilustración del concepto de ventajas  Aplicando ambos modelos se tiene una herramienta que clasifica y prioriza los requerimientos, los hace entendibles para los interesados. El formato matricial permite visualizar rápidamente la relación Requisitos-Prioridad y participar eficientemente en el proceso de análisis y priorización de requisitos.  Imagen: 228131\_i329 | |

|  |
| --- |
| **Cuadro de texto** |
| El modelo *FURPS* se desarrolló por primera vez en Hewlett-Packard y se usa para clasificar los requisitos funcionales y no-funcionales de los sistemas (Dyson, 2019). A continuación, se encuentra un ejemplo de matriz que permite dicha tarea. |

**Tabla 6** *Matriz FURPS+ & MoSCoW*

**

*Nota*. Reproducida de *The FURPS y MoSCoW Model.* Dyson, J. (2019). <https://www.linkedin.com/pulse/conjoining-furps-moscow-analyse-prioritise-jonathan-dyson/>

## Fabricación del modelo prototipo

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Slider Presentación | |
| **Introducción** | Utilizar prototipos como técnica de validación de requisitos para mostrarle al usuario una versión aproximada del *software*, permite que explore y participe en la validación de la apariencia, funcionalidad y hacerse una idea de cómo quedará. Pueden ser interfases de usuario mediante maquetación, las pantallas gráficas y de aplicaciones web. | |
| El prototipo permite a los usuarios ganar experiencia y a los diseñadores recibir su retroalimentación para cumplir los requisitos y presentar una próxima versión. | | Ilustración de concepto de investigación de usuario  Imagen: 228131\_i331 |
| Los prototipos se usan en conjunción con otras técnicas como entrevistas y diseño conjunto de aplicaciones (JAD). Se desarrollan utilizando requisitos preliminares o ejemplos de sistemas similares. | | Juego de ilustración vectorial de concepto abstracto de diseño UX. Investigación de usuarios, wireframe, prototipado, proyecto de diseño, encuesta en línea, informes y análisis, diseño de páginas web, navegación de sitios web metáfora abstracta.  Imagen: 228131\_i332 |
| Hay varios métodos para crear prototipos de sistemas como guiones gráficos (*storyboards*) y ejecutables minimalistas. La ventaja de usar prototipos es la expectativa y ánimo que generan en los usuarios, al ser parte activa en el desarrollo de los requisitos. | | ux Diseñador gráfico de dibujo creativo de planificación de aplicación de desarrollo prototipo de modelo de alambre para el teléfono móvil web . Concepto de experiencia del usuario.  Imagen: 228131\_i333 |
| Una desventaja es que los usuarios se apegan a ellos y, por lo tanto, se resisten a soluciones alternativas. La validación de prototipos en etapas tempranas del ciclo de vida del desarrollo del *software* minimiza los riesgos de modificación luego de la etapa de construcción. | | ux designer creative Graphic planning application development for web mobile phone . User experience concept.  Imagen: 228131\_i334 |

|  |  |
| --- | --- |
| Tipo de recurso | Cajón de texto de color |
| Los prototipos son más que herramientas para ayudar a los desarrolladores a probar sus aplicaciones móviles o sitios web, son claves en el proceso de diseño centrado en el usuario. En la siguiente figura se muestra cómo hacer un prototipo. | |
| **Figura 14** *Prototipo de aplicación móvil utilizando software para maquetación*    *Nota.* Reproducida de *¿cómo hacer un prototipo*? Ronam, S. (2021). <https://saulromanjimenez.com/que-prototipo-sirve-ejemplos/>  Imagen: 228131\_i335 | |

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | Slider pasos | |
| **Introducción** | Los pasos para desarrollar un prototipo son: | |
| **Slide 1** | Entender el problema del cliente, para eso se hace todo el proceso de requerimientos. | Imagen: 228131\_i336 |
| **Slide 2** | Definir la solución en el documento de especificación de requerimientos (ERS) para que el equipo de trabajo defina detalles y restricciones técnicas antes de comenzar a codificar. | Imagen: 228131\_i337 |
| **Slide 3** | Desarrollar el prototipo con requisitos aprobados, para hacer elecciones técnicas y comenzar el desarrollo. | Imagen: 228131\_i338 |
| **Slide 4** | Poner el prototipo en funcionamiento | Imagen: 228131\_i339 |
| **Slide 5** | Analizar la retroalimentación proveniente de los usuarios. | Imagen: 228131\_i340 |

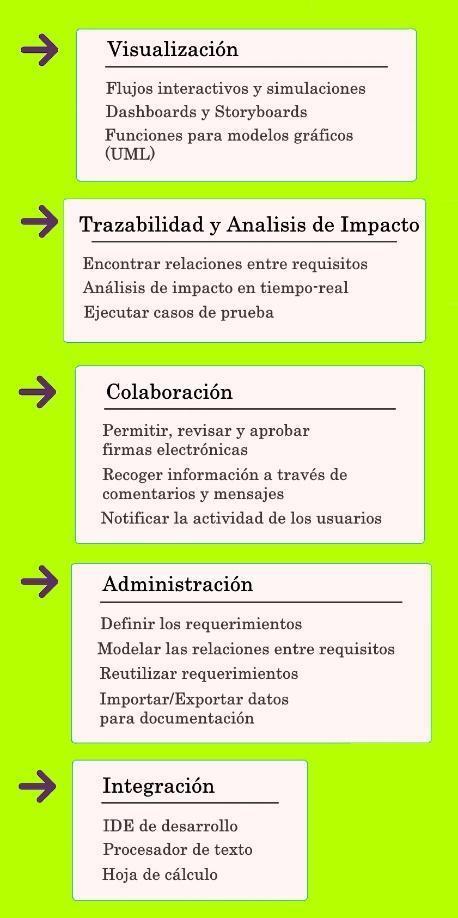
|  |  |
| --- | --- |
| Tipo de recurso | Cajón de texto de color |
| Ventajas del uso de prototipos:   * Minimiza riesgos al construir *software* que no satisfaga la necesidad del usuario. * Reduce el costo de iteración e incrementa la probabilidad de éxito del proyecto.Icono de pulgar hacia arriba y hacia abajo que muestra la sensación de me gusta o no me gusta en facebook * Prevé el uso de herramientas para el desarrollo de *software*. * Muestra al usuario final una vista rápida del desarrollo requerido. * Permite identificar funcionalidades no contempladas, para su inclusión en el análisis. * Provee un mecanismo de consenso de las interfaces de la aplicación.   Desventajas del uso de prototipos:   * El proceso de elaboración puede tomar mucho tiempo si el usuario final no tiene claro lo que desea o no decide rápidamente la validación. * Al recibir el prototipo puede pensar que la aplicación está terminada y no destina tiempo para mejorarla. * Durante la etapa de desarrollo el programador puede alterar el prototipo y generar una iteración que requiera nuevamente la validación del usuario y, en consecuencia, incrementar costo y tiempo. | |

## Generación de documentación

|  |  |
| --- | --- |
| **Tipo de recurso** | **Cajón de texto de color** |
| El documento de especificación de requisitos es el resultado del levantamiento, análisis y organización de la información recopilada con los clientes/usuarios y patrocinadores, escritas en lenguaje natural sin considerar términos técnicos. La finalidad de este documento es comunicar con precisión los requisitos a los encargados de desarrollar el *software*, áreas del negocio o clientes/usuarios que usarán el producto.Plataforma o servicio en línea de etapa de madurez período del ciclo de vida del proyecto implementación y desarrollo de proyectos comerciales ilustración plana vectorial  Imagen: 228131\_i342 | |

|  |  |  |
| --- | --- | --- |
| **Tipo de recurso** | | Pestañas o tabs Verticales |
| **Introducción** | | El documento de especificación de requisitos contiene la descripción de requerimientos funcionales y no-funcionales como: |
| Ilustración de concepto de archivos de texto  Imagen: 228131\_i343 | | |
| Requerimientos funcionales y no-funcionales | * Especificación de los datos a ser ingresados en el sistema. * Descripción de las operaciones a ser realizadas por cada pantalla. * Descripción de los flujos de trabajo realizados por el sistema. * Generación de reportes del sistema y otras salidas. * Control de acceso para ingresar datos en el sistema. * Conformidad con reglamentos y regulaciones aplicables. | |
| Documentación | La documentación técnica sirve para conocer todos los aspectos de la aplicación y del proceso de desarrollo. Automatizar la documentación de requerimientos facilita recolectar las ideas del equipo, conocimiento y distintos planes de acción a seguir, al tiempo que simplifica la presentación de informes y otras ventajas.   * Edición colaborativa. * Generar informes para patrocinadores e interesados. * Conformar el conocimiento base del negocio. * Presentar informes sobre avances, problemas o cambios. | |

**Figura 15** *Características de las herramientas para documentación de requisitos*



*Nota.* Adaptada de *Las mejores herramientas de gestión de requisitos en 2022*. Aston, B. (2021). <https://thedigitalprojectmanager.com/es/tools/herramientas-gestion-requisitos/>

**Síntesis**

|  |  |
| --- | --- |
| **Tipo de recurso** | Síntesis |
| Gestión de analítica de datos  Síntesis: recolección de los requisitos del *software* | |
| **Introducción** | En el siguiente mapa se muestran todos los conceptos relacionados con la recolección de los requisitos del *software*, de modo que se tenga una visión de conjunto de la temática desarrollada en este componente formativo. |
| **Figura 16** *Mapa mental de la ingeniería de requerimientos*  *Diagrama  Descripción generada automáticamente*  *Nota.* Reproducida de <https://drive.google.com/file/d/10ZTB4XJS0TFFTQCP-Z-yQyedgylMveze/preview> | |

**Actividad didáctica:**

|  |
| --- |
| Cuadro de texto |
| Para comprobar el aprendizaje en este componente formativo, lo invitamos a realizarla siguiente actividad didáctica.  **Anexo 4. Actividad didáctica** |

**Material complementario**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tipo de recurso | Material complementario | | | |
| Tema | Referencia APA del material | Tipo | Enlace | |
| Requerimientos de *software* | *Software* Engineering Coach. (5 de julio de 2022). *Serie 03: Ingeniería de Requerimientos de Software*. [videos]. YouTube. | Playlist de videos | | <https://www.youtube.com/playlist?list=PLCVGhLzsMEq8-Q7Mrvq5Q6jrHo4nDJpod> |
| Análisis y requerimientos de *software* | Wong, S. (2017). *Análisis y requerimientos de software: Manual autoformativo interactivo*. Universidad Continental de Huancayo. | E-Book | | <https://repositorio.continental.edu.pe/handle/20.500.12394/4281> |
| Requisitos de *software* | Mellado, R. (15 de agosto de 2020). *Requisitos de software.* [video]. YouTube. | Video | | <https://youtu.be/yYvqBDkWyiU> |
| Documento especificaciones de requerimientos | Varón Quimbayo, A. (30 junio 2021). *Cómo crear un documento de especificación de requerimientos de software*. [video]. YouTube. | Video | | <https://youtu.be/AotyBHVKp8I> |
| Análisis de requerimientos | Marlés, M. (16 de junio de 2020). *2. Análisis de requerimientos*. [videos]. YouTube. | Playlist de videos | | <https://www.youtube.com/playlist?list=PLRpLWFRnRWV0qnLWLwSVNPGUcSjlVfDNE> |
|  | Universidad Autónoma de México (2022). Análisis de requisitos. Studocu. | Curso web | | <https://www.studocu.com/es-mx/document/universidad-nacional-autonoma-de-mexico/analisis-de-requisitos/ers-este-documento-ayuda-a-la-especificacion-de-requerimientos-es-administrada/17977780> |
|  | Universidad Autónoma de México (2021-2022). *Especificación de Requisitos de Apoyo.* Studocu | Curso web | | <https://www.studocu.com/es-mx/document/universidad-nacional-autonoma-de-mexico/analisis-de-requisitos/ers-este-documento-ayuda-a-la-especificacion-de-requerimientos-es-administrada/17977780> |
|  | Universidad Nacional de Colombia (s/f). *Especificación de Casos de Uso.* La Oficina de Proyectos de Informática. Course Hero. | Curso web | | <https://www.coursehero.com/file/25099158/PMOInformatica-Plantilla-de-Casos-de-Usodoc/> |
|  | *(S/f). (S/f). Requerimientos de software. Plantilla. Guía para la Elaboración del Documento* | Link | | <https://independent.academia.edu/bocajr> |
|  | Escuela de informática y Telecomunicaciones, DUO UC. (2021). *Propuesta de proyecto y especificación de requisitos de software.* | Curso web | | <https://idoc.pub/documents/informe-ers-especificacion-de-requisitos-del-software-qn8517y222n1> |
|  |  |  | |  |

**Glosario**

|  |  |
| --- | --- |
| **Tipo de recurso** | Glosario |
| Atributo | característica de una entidad. Puede haber muchos atributos en una entidad. |
| Caso de uso | especificación de un tipo de interacción con el sistema. |
| Cliente | persona, organización o grupo de personas que encargan la construcción de un sistema, ya sea en su etapa inicial o refinando versiones sucesivas. |
| Dominio | área problemática o empresarial específica donde se usan los sistemas de *software*. Los ejemplos de dominio incluyen control en tiempo real, procesamiento de datos empresariales y conmutación de telecomunicaciones. |
| Fiabilidad | capacidad de un sistema para entregar los servicios como se especifican. La fiabilidad se puede especificar cuantitativamente como la probabilidad de que ocurra un fallo de funcionamiento o como la tasa de ocurrencia de estos. |
| Gestión de requerimientos | proceso de administrar los cambios a los requerimientos para asegurar que son analizados adecuadamente y rastreados a lo largo del sistema. |
| Interfaz | especificación de los atributos y operaciones asociados con un componente *software*. La interfaz es utilizada como el medio de tener acceso a la funcionalidad del componente. |
| Portabilidad | grado en que un sistema que opera en determinado entorno de ejecución pueda ser convertido para funcionar en otro entorno. |
| Proyecto | esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida. |
| Requerimiento | condición o capacidad requerida por una parte interesada para resolver un problema o alcanzar un objetivo. Debe cumplir con un componente de solución para satisfacer un contrato, norma, especificación u otros documentos. |
| Requerimiento funcional | declaración de alguna función o característica que se debe implementar en un sistema. |
| Requerimiento no funcional | declaración de una restricción o comportamiento que se aplica a un sistema. Esta restricción se puede referir a las propiedades emergentes del *software* que se está desarrollando o al proceso de desarrollo. |
| Sistema | colección de subsistemas interrelacionados e interdependientes, trabajando juntos para lograr metas y objetivos predeterminados. Todos los sistemas tienen entrada, procesos, salida y retroalimentación. |
| Usuario | individuo u organización que utiliza el sistema en operación para llevar a cabo una función específica. |
| Validación | confirmación mediante el suministro de evidencia objetiva de que se han cumplido los requerimientos para una utilización o aplicación específica prevista. |

**Referencias bibliográficas**

|  |  |
| --- | --- |
| **Tipo de recurso** | **Bibliografía** |
| Aston, B. (2021). *Las mejores herramientas de gestión de requisitos en 2022*. Recuperado el 24 de septiembre de 2022 <https://thedigitalprojectmanager.com/es/tools/herramientas-gestion-requisitos/> | |
| Bennaceur, G., Thein, T. Yijun, Y. y Bashar, N. (2019). *Requirements Engineering*. En Cha, S., Taylor, R. N., y Kang, K. (Eds.) *Handbook of Software Engineering*. (pp. 51-87). Springer Nature Switzerland | |
| CVUDES. (2015). *Modelo FURPS*. Obtenido de cvudes.edu.co: <https://sites.google.com/cvudes.edu.co/evaluacionred/grupo_eval_red_3/modelo-furps> | |
| Digité. (2020). *¿Qué es el mapeo de historias (de usuarios)? [Web log post]*. Blogspot. <https://www.digite.com/es/agile/mapeo-de-historias/> | |
| Dyson, J. (2019). *The FURPS Model*. Recuperado el 24 de septiembre de <https://www.linkedin.com/pulse/conjoining-furps-moscow-analyse-prioritise-jonathan-dyson/> | |
| Equipo Editorial Project Management. (2019). *Gestión de requerimientos II: características de los requerimientos [Web log post]*. Blogspot. <https://blogs.salleurl.edu/es/project-management/gestion-de-requerimientos-ii-caracteristicas-de-los-requerimientos> | |
| Fatto consultorías y sistemas (2016). *Entrevistas eficaces para el trabajo de requerimientos.*  [Video] Youtube. https://www.youtube.com/watch?v=C\_r-hTknkf8 | |
| Gómez, S. y Moradela, E. (2020). *Aproximación a la Ingeniería del Software*. Universitaria Ramón Areces | |
| Hansen, B. (19 de 04 de 2021). *Reglas de la lluvia de ideas: una guía del gerente para producir grandes ideas*. Recuperado el 24 de septiembre de <https://www.wrike.com/blog/rules-of-brainstorming-managers-guide-producing-great-ideas/> | |
| ISO-IEC-IEEE. (2018). *Systems and Software Engineering - Life Cycle Processes - Requirements Engineering*. (ISO/IEC/IEEE 29148-2018). | |
| Lucidchart. (2021). *¿Cuáles son tus necesidades de creación de mapas de procesos?* Recuperado el 24 de septiembre de <https://www.lucidchart.com/pages/es/que-es-la-creacion-de-mapas-de-procesos> | |
| Mohapatra, H. y Rath, A. (2020). *Fundamentals of Software Engineering*. BPB Publications | |
| Pressman, R., y Maxim, B. (2019). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education | |
| Pressman, R. y Maxim, B. (2021). *Ingeniería de software - Un enfoque práctico*. (9a Ed.). McGraw-Hill Interamericana. Obtenido de https://www-ebooks7-24-com.bdigital.sena.edu.co/?il=16414 | |
| REHKOPF, M. (2021). *User stories with examples and a template [Web log post]*. Blogspot. <https://www.atlassian.com/agile/project-management/user-stories> | |
| Ronam, S. (2021). *¿Qué es un prototipo y para qué sirve? – Ejemplos [Web log post]*. Blogspot. <https://saulromanjimenez.com/que-prototipo-sirve-ejemplos/> | |
| Santana, S., Perero, L. R., Delduca, A. G., & Dapozo, G. N. (2020). *Evaluación de técnicas para la validación de requerimientos en entornos de trabajo para el desarrollo de software.* Obtenido de <http://sedici.unlp.edu.ar/handle/10915/103994> | |
| Sommerville, L. (2016). *Software Engineering,* 10th edition. Escocia: Pearson. | |
| Team Asana. (2022). *Técnicas para la recopilación de requisitos*. Recuperado el 24 de septiembre de <https://asana.com/es/resources/requirements-gathering> | |
| Tsui, F., Karam, O. y Bernal, B. (2018). *Essentials of Software Engineering* (4a Ed.). Jones & Bartlett Learning | |
| Universitat Politècnica de València-UPV. (2021). *Casos de uso y diagramas de casos de uso* [video]. YouTube. <https://youtu.be/iFcDoP6jEeE> | |
| Valda, J. C. (2021). *Análisis de los stakeholders* [Web log post]. Blogspot. <https://www.grandespymes.com.ar/2021/04/10/analisis-de-los-stakeholders/> | |
| Wong, S. (2017). *Análisis y requerimientos de software: Manual autoformativo interactivo*. Universidad Continental de Huancayo. Obtenido de <https://hdl.handle.net/20.500.12394/4281> | |

# 