**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

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

|  |  |  |  |
| --- | --- | --- | --- |
| COMPETENCIA | 220501092 - Especificación de requisitos del *software*. | RESULTADOS DE APRENDIZAJE | 220501092-1.  Planear el proceso de recolección de información para identificar las necesidades del *software* a desarrollar. |

|  |  |
| --- | --- |
| NÚMERO DEL COMPONENTE FORMATIVO | 2 |
| NOMBRE DEL COMPONENTE FORMATIVO | La fase de elicitación de requisitos |
| BREVE DESCRIPCIÓN | La recolección de datos se refiere al uso de una gran diversidad de técnicas y herramientas que pueden ser utilizadas por el analista para desarrollar los sistemas de información, que pueden ser la entrevista, la encuesta, el cuestionario, la observación, las sesiones en grupo, la visita a instalaciones, entre otros. Cada técnica de recolección de información posee diferentes instrumentos o herramientas para ser llevadas a cabo con profesionalismo y confiabilidad. |
| PALABRAS CLAVE | Ciclo vida de *software*, ingeniería de requisitos, requisitos. |

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

1. **TABLA DE CONTENIDOS:**

Introducción

1. Elicitación de requisitos

1.1 Planeación

1. 2. Técnicas e Instrumentos para elicitar requisitos

1.2.1. Entrevista

1.2.2. Encuesta

. 1.2.3. Observación

1.2.4. Sesiones grupales

1.3. Herramientas para captura de requisitos

1.3.1. Diagrama de casos de uso

1.3.2. Historias de usuario

1.3.3. *Storyboard*

1.4. Herramientas de modelado

1. **DESARROLLO DE CONTENIDOS:**

**Introducción**

A través de este material, se tratan con detalle los pasos que se deben seguir en el proceso de recolección de datos, el uso de técnicas y los instrumentos para tal fin.

La recolección de datos se refiere al uso de una gran diversidad de técnicas y herramientas que pueden ser utilizadas por el analista para desarrollar los sistemas de información, las cuales pueden ser la entrevista, la encuesta, el cuestionario, la observación, las sesiones grupales, la recolección documental, entre otras.

En ese sentido, los analistas utilizan una variedad de métodos para recopilar los datos sobre una situación existente, como entrevistas, cuestionarios, inspección de registros (revisión en el sitio) y observación. Cada uno tiene ventajas y desventajas; es por ello que, por lo general, se utilizan dos o tres simultáneamente, para complementar el trabajo asegurar una investigación completa.

**1. Elicitación de requisitos**

El propósito de la elicitación de requerimientos es ganar conocimientos relevantes del problema, que se utilizarán para producir una especificación formal del software necesario para resolverlo. “Un problema puede ser definido como la diferencia entre las cosas como se perciben y las cosas como se desean” (Gause y Weinberg 1989). Aquí se ve la importancia que tiene una buena comunicación entre desarrolladores y clientes; de esta comunicación con el cliente depende que sus necesidades queden claras. Además, al final de la fase de análisis de requerimientos, el analista podría llegar a tener un conocimiento extenso en el dominio del problema.

La elicitación de requisitos es la actividad que se considera como el primer paso en un proceso de ingeniería de requisitos; su significado hace referencia a la puesta en marcha de técnicas que sirven para recopilar conocimiento o información y **los objetivos de esta fase de elicitación**, son:

* Conocer el dominio del problema para poder comunicarse con clientes y usuarios y entender sus necesidades.
* Conocer el sistema actual (manual o informatizado) y sus aspectos positivos y negativos.
* Identificar las necesidades, tanto explícitas como implícitas, de clientes y usuarios y sus expectativas sobre el sistema a desarrollar.
  1. **Planeación**

La planeación busca definir las tareas a realizar para elegir y planificar las técnicas a emplear durante la actividad de elicitación de la fase de ingeniería de requisitos del desarrollo de *software*. En la siguiente tabla se presenta una relación de estas tareas y sus correspondientes procesos.

**Tabla 1**

*Tareas para elicitación de requisitos*

|  |  |
| --- | --- |
| **Tareas** | **Proceso** |
| A. Identificar las fuentes. | Lista de fuentes de requerimientos. |
| B. Identificar interesados del producto. | Categorías de los interesados (*stakeholder*). |
| C. Matriz *stakeholders* (Describir necesidades y criterios de éxito). | Perfil de *stakeholder.* |
| D. Revisar técnicas. | Identificar combinaciones de técnicas entrevistas, grupos focales, encuestas, prototipos. |
| E. Captura de interesados. | Plan de captura de interesados. |

Nota: Tomado de Durán y Bernárdez (2001)

A continuación, se describen los procesos relacionados con las tareas para elicitación de requisitos:

1. **Identificar las fuentes de requerimientos**

Existe un conjunto de fuentes de requisitos en cada proyecto de desarrollo de *software*, así, usuarios y expertos abastecen de información detallada acerca del problema y necesidades del usuario. Los procesos y sistemas existentes representan, también, fuentes de requisitos; además, la documentación existente como manuales, formularios y reportes, incluso especificaciones de requisitos anteriores, puede proveer información útil acerca de la organización y su entorno.

En el proceso de esta actividad se identifican:

* Interesados relevantes.
* Documentación que se puede usar como fuente de información de los requerimientos.
* Fuentes de información externas.

Las fuentes de requerimientos incluyen los propietarios del problema, los *stakeholders*, documentos y otros sistemas (Pearson, 2002). En ese sentido, los requerimientos pueden obtenerse en diversas fuentes que pueden clasificarse en gente (*people*), productos o documentos, pero cualquiera sea la fuente de esos requerimientos deben ser chequeados con los *stakeholders*.

Estas fuentes de requerimientos, se pueden clasificar en:

* **Fuentes primarias:** aportan material de primera mano (es protagonista o testigo de los hechos), estas fuentes contienen información original, que ha sido publicada por primera vez y que no ha sido filtrada, interpretada o evaluada por nadie más.
* **Fuentes secundarias**: toman y reproducen la información que le aportó una fuente primaria. Son las que contienen información primaria, sintetizada y reorganizada y están especialmente diseñadas para facilitar y maximizar el acceso a las fuentes primarias o a sus contenidos. Parten de datos preelaborados, como pueden ser datos obtenidos de anuarios estadísticos, internet, medios de comunicación, bases de datos procesadas con otros fines, artículos y documentos relacionados con un tema, libros, tesis, informes oficiales, etc.
* **Fuentes terciarias**: son guías físicas o virtuales que contienen información sobre las fuentes secundarias. Forman parte de la colección de referencia de una biblioteca; facilitan el control y acceso a toda la gama de repertorios de referencia, como las guías de obras de referencia, o a un solo tipo, como las bibliografías.

Por otra parte, las fuentes de información, pueden ser orales, escritas o de otro tipo, dependiendo de cómo se transmitan los [datos](https://definicion.de/datos/). A continuación, se pueden revisar algunos ejemplos de fuentes de información.



1. **Identificar interesados del producto**

Uno de los primeros pasos en el proceso es el análisis e identificación de todas las personas relevantes que tienen un grado de interés en el proyecto. Los interesados (*stakeholders*), son los individuos y organizaciones que están relacionados activamente en un proyecto de *software*; tienen influencia directa o indirecta sobre los requisitos, o sus intereses se ven afectados por el proyecto (Baar, 2006, Ventura, 2002).

En resumen, son grupos o individuos que están interesados en el producto de *software* que se está desarrollando y necesitarán estar informados acerca del progreso, conflictos, cambios y prioridades del proceso de desarrollo del producto.

Los *stakeholders* se dividen en dos grupos

* **Primarios**. Son aquellas personas indispensables para el correcto funcionamiento de la organización, y tienen una relación económica directa con la empresa. Estos pueden ser sus socios, clientes y accionistas.
* **Secundarios**. Son los entes que no participan directamente de la compañía, pero también son afectados por sus resultados. En este círculo se encuentran los competidores, el mercado y las personas en general.

A continuación, se listan roles más generales de las personas involucradas con sus términos similares, aunque cabe resaltar que existen leves diferencias entre ellos (Sommerville y Sawyer, 2005)

* Líder de proyecto / Administrador de proyecto / Gerente de proyecto.
* Analista / Ingeniero de requisitos.
* Ingeniero de sistemas / Arquitecto.
* Programador / Desarrollador / Ingeniero de *software*.
* Probador / Asegurador de la calidad.
* Administrador de bases de datos.

En la siguiente tabla se presentan los principales roles involucrados en el proceso de ingeniería de requisitos, así como las actividades en las que tienen mayor participación.

**Tabla 2**

*Roles involucrados en la ingeniería de requisitos*

|  |  |
| --- | --- |
| ROL | DESCRIPCIÓN |
| Cliente | Representa a la persona u organización que solicita la creación de un sistema a un área de desarrollo y quien lo paga. Es con quien se negocia el tiempo, costo y alcance del proyecto. Pueden o no ser usuarios del sistema. |
| Usuario | Son las personas que interactuarán con el sistema. Proporcionan información fundamental para el éxito del proyecto, ya que conocen y conviven con los procesos diarios. |
| Líder de proyecto | Por parte del equipo de desarrollo, es el representante ante el cliente. Es la persona responsable de completar el proyecto exitosamente con los recursos dados. |
| Analista | Su labor se enfoca en la ingeniería de requisitos, los identifica, analiza, modela y documenta. Además, establece contacto directo con los usuarios y utiliza diversas técnicas de comunicación y de recopilación de información para lograr su objetivo. |
| Programador | Con base en los requisitos recibidos de parte de los ingenieros de requisitos, el programador realiza la codificación para producir el sistema deseado. |
| Asegurador de la calidad | Garantiza el cumplimiento del proceso y de los estándares del producto. Enfocado en los requisitos, los verifica y valida para imprimir la calidad desde las primeras etapas del desarrollo. Paralelamente, prepara planes de prueba para esos requisitos del sistema. |
| Arquitecto | Es el responsable del diseño de alto nivel y es clave a la hora de precisar los atributos de calidad del producto. |

Nota: Tomado de Ventura (2002)

Algunas de las técnicas que se pueden emplear para llevar a cabo la labor de análisis de los *stakeholders* incluyen entrevistas con los expertos, lluvia de ideas en grupo y lista de chequeo. Se espera que este grupo de personas identifiquen y caractericen a los *stakeholders* objetivamente, por tal motivo es recomendable involucrar a personas de diferentes contextos (Karisen, 2002 citado en Wessinger, 2012).

1. **Matriz de *stakeholders***

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 las estrategias de gestión correspondientes.

* **Importancia de la matriz de *stakeholders* en los proyectos de desarrollo**

En los proyectos de desarrollo, la gestión de los *stakeholders* es de suma importancia para alcanzar el éxito de los proyectos, ya que el proceso de identificación de los involucrados y la definición de sus niveles de interés e influencia en el proyecto, marcarán el punto de partida para desarrollar estrategias que posibilitan obtener el apoyo requerido para alcanzar los objetivos por los que el proyecto es emprendido. Es por ello, que la matriz de *stakeholders* es una herramienta indispensable desde el comienzo del proyecto mismo, ya que proveerá de la información necesaria para gestionar, adecuadamente, las expectativas de los involucrados a lo largo del proyecto, maximizando las influencias positivas y mitigando los impactos negativos potenciales derivados de estos. Además, dado el carácter social de los proyectos de desarrollo, involucrar a la sociedad civil no debe ser solo un ejercicio de comunicación unidireccional sino una oportunidad para lograr su apoyo al proyecto.

* **Proceso de armado de la matriz de *stakeholders***

Para desarrollar la matriz de *stakeholders* es necesario identificar las entradas necesarias que proveerán la información con la que el líder y el equipo de proyecto trabajarán para desarrollar la matriz misma. Tales entradas pueden ser el acta de constitución de proyecto, documentos de adquisición, activos de los procesos y factores ambientales de la organización.

**Figura 1**

*Entradas, herramientas y técnicas, salidas*







* **Descripción de los componentes de la matriz de *stakeholders***

A continuación, se presenta el concepto de cada uno de los componentes que estructuran la matriz de *stakeholders*.

* ***Stakeholder*:** es el nombre con el que se identifica al *stakeholder*.
* **Tipo**: identifica si el *stakeholder* desempeña un rol interno o externo al proyecto mismo. Los *stakeholders* pueden ser internos, como el personal de las unidades ejecutoras, el personal administrativo o ejecutivo de la organización, el personal de las entidades financiadoras con alto nivel de poder e influencia en el proyecto y sus recursos; o externos como los beneficiarios del proyecto, las instituciones del sector o las organizaciones de la sociedad civil, quienes serán de un modo u otro impactados por los resultados del proyecto.
* **Objetivo o resultados**: en este campo se enlistan los objetivos o resultados en los que el *stakeholder* muestra interés o en aquellos en los que puede influir positiva o negativamente con sus acciones. Esta información puede ser suministrada por el acta de constitución de proyectos, la estructura de la organización, la estructura de desglose de trabajo, los diferentes planes que conforman el proyecto, entrevistas a los mismos interesados, etc.
* **Nivel de interés y de influenci**a: grado de interés e influencia que tiene el *stakeholder* en el proyecto. Este grado de interés e influencia puede ser mucho o poco, alto medio o bajo, dependiendo de la escala que se utilice en la matriz de interés vs influencia (figura 2). Cabe mencionar que la determinación de un cierto grado de interés e influencia es una estimación subjetiva que se basa en el juicio de expertos, obviamente respaldada con la información que se conoce del *stakeholder*. Esta información será muy útil al momento de desarrollar las estrategias de gestión de las expectativas de los involucrados.
* **Acciones posibles con impacto positivo / negativo**: son las acciones que puede emprender el *stakeholder* y que pueden influir, negativa o positivamente, en los objetivos del proyecto en los que muestra su interés o en aquellos en los que puede influir debido a su jerarquía, estatus, recursos de los que dispone, entre otros.
* **Estrategias**: es un listado de acciones que se pueden emprender para obtener el apoyo necesario o evitar obstáculos por parte de los *stakeholders* durante la ejecución y conclusión del proyecto. Las estrategias se desarrollan considerando el tipo de *stakeholder*, los objetivos en los que está interesado, el nivel de interés y poder que puede ejercer en el proyecto (figura 1) y las acciones posibles que podría emprender para afectar tanto positiva como negativamente al proyecto.
* **Conclusiones:** es la síntesis sobre puntos clave a considerar para gestionar de manera efectiva las expectativas de los *stakeholders*. Las conclusiones se obtienen de relacionar, analizar y sintetizar toda la información vertida en la matriz de *stakeholders*.



* **Categorización de *stakeholders* y estrategias de gestión de las expectativas**

Como ya se había mencionado anteriormente, la matriz de *stakholders* es una herramienta muy útil que permite clasificar a los involucrados en el proyecto según sus niveles de interés e influencia, priorizando a los más importantes y desarrollando así las estrategias correspondientes para gestionar sus expectativas. De la misma manera, su clasificación puede cambiar durante la vida del proyecto. Así, aquellos que fueron inicialmente identificados con un alto nivel de influencia en el proyecto, pueden ser reclasificados a un nivel más bajo durante otras etapas de la vida del proyecto.

La categorización de los *stakeholders* se lleva a cabo una vez que la información sobre éstos esté completa. Para ello se puede utilizar una matriz de 2x2 en la que se pueda graficar el grado de poder e interés que tiene el involucrado en el proyecto, coadyuvando así a clasificar a cada *stakeholder* dentro del grupo para el cual se definen diferentes estrategias (figura 2).

**Figura 2**

*Ejemplo de matriz interés vs. influencia*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Interés | Mucho | | 3. Satisfacer | | 1. Colaborar |
| Poco | | 4. Observar | | 2. Comunicar |
|  |  | | Poca | | Mucha |
|  | |  | | Influencia | | |

Nota. Tomado de Gardner et al. (1986).

**1.2. Técnicas e instrumentos para elicitar requisitos**

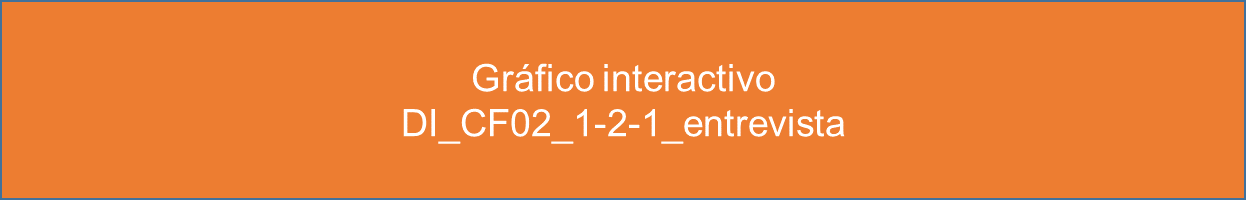
Hay una variedad de técnicas propuestas para ingeniería de requerimientos (Herrera, 2003. p. 12), por lo que es primordial resaltar que estas técnicas pueden ser aplicables a las distintas fases del proceso de la ingeniería de requerimientos (IR), teniendo en cuenta las características propias del proyecto en particular que se esté desarrollándose para aprovechar al máximo su utilidad.

A continuación, se presentará una serie de técnicas destinadas a facilitar la elicitación correcta y efectiva de los requisitos dentro de un proceso de desarrollo.

**1.2.1. Entrevista**

La entrevista es una forma de recoger información de otra persona a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada (Braude, 2003).

En las entrevistas se pueden identificar tres fases: preparación, realización y análisis (Piattini et al. 1996), como se puede observar en el siguiente gráfico.

****

Las entrevistas se pueden clasificar fundamentalmente, en:

* **Entrevista estructurada**

Las preguntas en esta entrevista se deciden, previamente, de acuerdo con el detalle de información requerida.

* + Recoge de forma sistemática y precisa la mayor información sobre los aspectos que quiere explorar.
  + Las preguntas son prefijadas y definidas, las respuestas son esperadas e incluso se le dan al entrevistado en forma de varias opciones.
  + Las etapas son planificadas. d. La interpretación de las respuestas se realiza de acuerdo con unos criterios establecidos.



* **Entrevista semiestructurada**

Esta presenta un grado mayor de flexibilidad que la estructurada, debido a que parten de preguntas planeadas, que pueden ajustarse a los entrevistados. Su ventaja es la posibilidad de adaptarse a los sujetos con enormes posibilidades para motivar al interlocutor, aclarar términos, identificar ambigüedades y reducir formalismos.

* + Las preguntas, desarrollo e interpretación se planifican previamente, pero con un cierto grado de libertad de acción para abordar temas que pueden surgir durante la misma.
  + Se suele utilizar un protocolo para facilitar al entrevistador seguir un modelo preestablecido.
* **Entrevista no estructurada**

Las entrevistas no estructuradas suelen describirse como conversaciones mantenidas con un propósito en mente.

* No se estructura ni planifica previamente.
* Es la más ágil y la que proporciona más información en general, pero requiere un cierto dominio por parte del entrevistador.



**1.2.2. Encuesta**

“Los cuestionarios son herramientas ampliamente utilizadas para recoger datos de sondeos y pueden ser administradas sin la presencia del investigador” (Cohen, 2011, p. 377).

Pueden variar en cuanto a propósito, diseño y apariencia, y consisten en listas de preguntas escritas. Los individuos participantes en la investigación suelen leer los mismos listados de preguntas, por lo que esto permite consistencia y precisión al analizar las respuestas, además de facilitar el proceso. Una de las ventajas más destacadas de los cuestionarios es que simplifican el proceso de la obtención de datos, preguntando directamente a los individuos participantes para obtener datos de forma rápida y directa y se pueden aplicar a un gran número de sujetos.

Los datos que se obtienen a través de los cuestionarios suelen estar clasificados en dos categorías: hechos y opiniones (Denscombe, 2010, p. 156). La información relacionada con los hechos no requiere el juicio o la actitud personal de los sujetos participantes, pero la información obtenida a través de las opiniones implica creencias, puntos de vista y preferencias de los sujetos participantes.

* **Tipos de preguntas**

La distinción más general entre los tipos de preguntas de los cuestionarios, además de hechos y opiniones, es la de preguntas abiertas y cerradas; las preguntas abiertas son aquellas en las que no se especifica ninguna respuesta para elegir y se deja abierta a la elección del participante para que escriba en ella. Las preguntas cerradas son las que ofrecen ya unas respuestas predeterminadas para su elección.

* **Tipos de respuestas**

Las respuestas de escala son las más comunes en los cuestionarios de investigación ya que implican al participante en una valoración o evaluación de las respuestas objetivo por medio de varias opciones en las que tienen que marcar dentro de una escala la importancia de cada una. Esa escala de valoración indica diferentes grados en una categoría y puede ser de diversa naturaleza; por ejemplo, puede valorar una categoría indicando si es “bueno” o “malo”, “frecuente” o “infrecuente”, “importante” o “poco importante” o también pueden valorar opiniones: “completamente de acuerdo” o “en desacuerdo. El número de opciones más común es el de cinco, por ser un número impar, ya que existe una tendencia generalizada a seleccionar la opción intermedia (Dornyei, 2010).



**1.2.3. Observación**

Esta permite la obtención de datos para emprender una investigación de tipo cualitativo, no desde el punto de vista de lo que los sujetos dicen, sino que es la evidencia directa de lo que ve y percibe el investigador en un escenario de primera mano (Denscombe, 2010).

Por su parte, Selltiz (citado por Hernández, Fernández y Baptista, 2006, p. 229), al referirse a la observación, recomienda que para que esta se convierta en una técnica como tal, debe cumplir con cuatro condiciones:

1. Debe servir a un objeto formulado de investigación.

2. Debe de ser planificada sistemáticamente.

3. Debe estar controlada y relacionada con proposiciones generales.

4. Debe ser sujeta a comprobaciones y controles de validez y fiabilidad.

De acuerdo con lo anterior, se puede asumir que la observación:

* Tiene la característica de seguir normas, reglas y procedimientos.
* Permite a los sujetos y objetos establecer relaciones de manera directa.

Para el caso de obtención de requerimientos del *software* la observación nos sirve para estudiar el entorno de trabajo de los usuarios, clientes e [interesados de proyecto](http://www.pmoinformatica.com/2013/01/quinta-edicion-de-la-guia-del-pmbok.html) (*stakeholders*) y para documentar la situación actual de procesos de negocio.

En la siguiente figura, se pueden revisar los tipos de observación.

**Figura 3**

*Tipos de observación*



Ahora bien, para llevar a cabo la observación, el observador puede utilizar como instrumento **la guía de observación,** la cual le permite situarse de manera sistemática en aquello que realmente es objeto de estudio para la investigación; también es el medio que conduce la recolección y obtención de datos e información de un hecho o fenómeno.

Tamayo (2004) define a la guía de observación como:

Un formato en el cual se pueden recolectar los datos en sistemática y se pueden registrar en forma uniforme, su utilidad consiste en ofrecer una revisión clara y objetiva de los hechos, agrupa los datos según necesidades específicas, se hace respondiendo a la estructura de las variables o elementos del problema (p. 172).

Para elaborar la guía de observación se ha de diseñar el contenido de la observación; el cual debe incluir por lo menos los siguientes aspectos:

1. Datos y características de los sujetos a evaluar.

2. Propósitos de la observación o de las observaciones a realizar.

3. Temporalidad de la observación.



**1.2.4. Sesiones grupales**

Es un proceso por el cual se llevan a cabo reuniones en grupo altamente estructuradas que convocan, en una misma sala, a los usuarios de un sistema, los propietarios del sistema y a los analistas durante un amplio periodo de tiempo. Los objetivos de esta técnica son esencialmente los mismos que los de las entrevistas, con la salvedad de necesitar más analistas para llevarlos a cabo.

Dentro de las sesiones de trabajo en grupo se encientran técnicas como la lluvia de ideas, las sesiones JAD y el método Delphi.

* **Lluvia de ideas**

También denominada tormenta de ideas o incluso *brainstorming*. Faickney (1939) investigó sobre diferentes maneras de generar creatividad. Se percató de que la mejor manera de ser creativo en una empresa es a través de la interacción y el trabajo en equipo; todos juntos podían dar sus opiniones y sugerencias sobre un tema determinado. Creó de esta manera la lluvia de ideas.

* **Sesiones JAD (*Joint Application Design*)**

Es un proceso usado para reunir requerimientos en el desarrollo de nuevos sistemas de información para una compañía. El proceso JAD consiste en un taller donde los trabajadores del conocimiento y los especialistas en tecnologías de información se reúnen, algunas veces durante varios días, para definir y revisar los requerimientos de negocio para el sistema. Los asistentes incluyen oficiales de administración de alto nivel, quienes se aseguran de que el producto provea los reportes y la información requerida al final, esto actúa como “un proceso de administración” que permite que los departamentos de servicios de información corporativa trabajen más eficientemente con los usuarios en un marco de tiempo más reducido.

A través de los talleres JAD, los trabajadores del conocimiento y los especialistas en tecnologías de información pueden resolver cualquier dificultad o diferencias entre las posturas referentes al nuevo sistema de información. El taller sigue una detallada agenda para lograr garantizar que todas las incertidumbres entre los grupos sean cubiertas y para ayudar a prevenir cualquier falla en la comunicación, estas fallas de comunicación pueden provocar repercusiones mucho más serias si no se identifican a tiempo. Al final, este proceso resultará en un nuevo Sistema de Información viable y orientado tanto a diseñadores como a usuarios

* **Método Delphi**

Es un método de estructuración de un proceso de comunicación grupal que consiste en la selección de un grupo de expertos a los que se les pregunta su opinión frente a ciertas temáticas.

* **Fase uno.** Formulación del problema: se define el campo de investigación.
* **Fase dos.** Elección de expertos: el experto se elige según su preparación y su capacidad de proyección.
* **Fase tres**. Elaboración de cuestionarios: las preguntas deben hacerse de acuerdo con la temática que se quiere obtener.
* **Fase cuatro.** Desarrollo y explotación de resultados: el cuestionario se entrega a los expertos para ser contestado por ellos.

**1.3.** **Herramientas para captura de requisitos**

Existen varias herramientas para la captura de requisitos potenciales de un nuevo sistema o una actualización de *software*, a continuación, se explican las más utilizadas:

**1.3.1. Diagrama de casos de uso**

Al momento de desarrollar un proyecto se debe pensar en cuáles serán las principales funcionalidades que el *software* debe permitir llevar a cabo y quiénes serán los que podrán ejecutar dichas funcionalidades. La identificación de estos elementos se puede visualizar de manera efectiva a través de la elaboración de diagramas de casos de uso; estos diagramas, que son elaborados durante las etapas iniciales de un proyecto, se convierten en referente para cada una de las etapas siguientes del desarrollo del proyecto.

**Componentes.** En los diagramas de casos de uso, se observan los siguientes componentes.

* **Actor:** se representa mediante un “hombre de palo”. Este se emplea para indicar el tipo de usuario del sistema que podrá ejecutar alguna función.

**Figura 4**

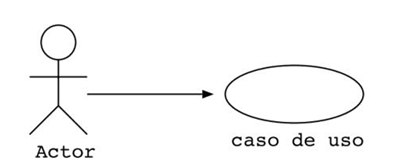
*Nombre del actor*



* **Caso de uso**: se representa mediante un óvalo e indica una función que el sistema debe proveer.

**Figura 5**

*Caso de uso*



Para ejemplificar un proceso se puede emplear un verbo conjugado en infinitivo y que represente la función a realizar (administrar, gestionar, registrar, entre otros). A continuación, se presenta un ejemplo, en el cual se presenta un diagrama de casos de uso de la sistematización de un centro médico.

• Administrar datos pacientes.

• Administrar datos tratamientos.

• Gestionar citas.

• Generar reportes.

* **Representación gráfica**

**Figura 6**

*Caso de uso centro médico*

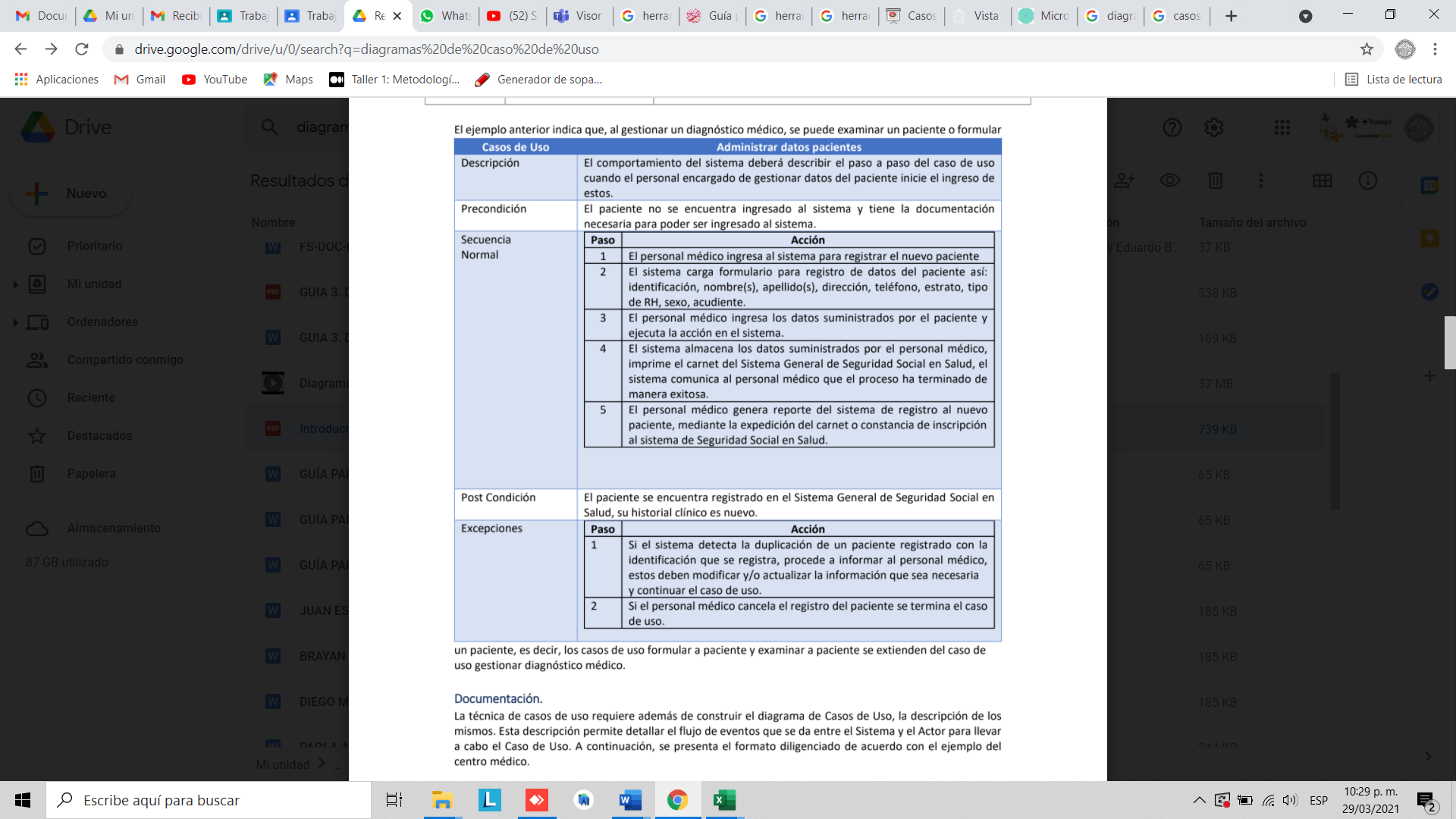




* **Identificación de casos de uso**: en el ejemplo anterior se observan los casos de uso identificados en el sistema, es decir, las funcionalidades que el sistema va a proveer (administrar datos pacientes, administrar datos tratamientos, gestionar citas, generar reportes).
* **Identificación de actores**: los actores son los usuarios que podrán ejecutar los casos de uso, en el ejemplo anterior, se identificaron dos actores (médico y empleado).
* **Documentación:** la técnica de casos de uso requiere además de construir el diagrama de casos de uso, la descripción de estos. Esta descripción permite detallar el flujo de eventos que se da entre el Sistema y el Actor para llevar a cabo el caso de uso. A continuación, se presenta el formato diligenciado de acuerdo con el ejemplo del centro médico.

**Figura 7**

*Documentación caso de uso*



Nota. Tomado de Gutiérrez (s.f.)

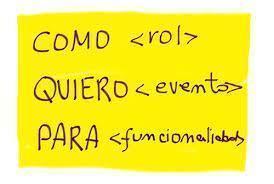
**1.3.2. Historias de usuario**

Las historias de usuario son utilizadas en los métodos agiles para la especificación de requisitos, son una descripción breve de una funcionalidad *software* tal y como la percibe el usuario (Cohn, 2004).

El formato para las historias de usuario Scrum se basan en una regla de tres palabras:

**Figura 8**

*Regla de tres palabras*



* **Como** <rol>
* **Quiero** <eventos>
* **Para** <funcionalidades>

Nota. Tomada de Martin (s.f.)

Así, el **<rol>** que se escoja que va a utilizar la aplicación *software*, requiere de una **<Acción> /<evento>** que ocurra, porque se desea cubrir una **<funcionalidad>.** Corto y conciso, directo y claro.

En las siguientes figuras se presentan ejemplos de historias de usuario.

**Figura 9**

*Ejemplos historias de usuario*



Nota. Tomada de fatto [www.fattocs.com](http://www.fattocs.com) https://i.ytimg.com/vi/Zi9E1aUO\_1U/hqdefault.jpg

**Conversación para explicar mejor la historia de usuario**

Como se mencionó anteriormente, las historias de usuario son una frase sencilla y concisa, sin embargo, eso no impide que se pueda abrir un diálogo (conversación) entre todos los miembros del equipo. De hecho, esta conversación se debe llevar a cabo para explicar mejor la propia historia de usuario y conseguir objetivos como:

* Detallar a mayor nivel como se realizará la solución.
* Clarificar aspectos de valor, funcionamiento y técnicos.
* Resolver las dudas que aparezcan.

Estas conversaciones llevarán a alcanzar acuerdos sobre los distintos puntos tratados, que quedarán reflejados en los criterios de aceptación y que permitirán validar cuando una historia de usuario está terminada.

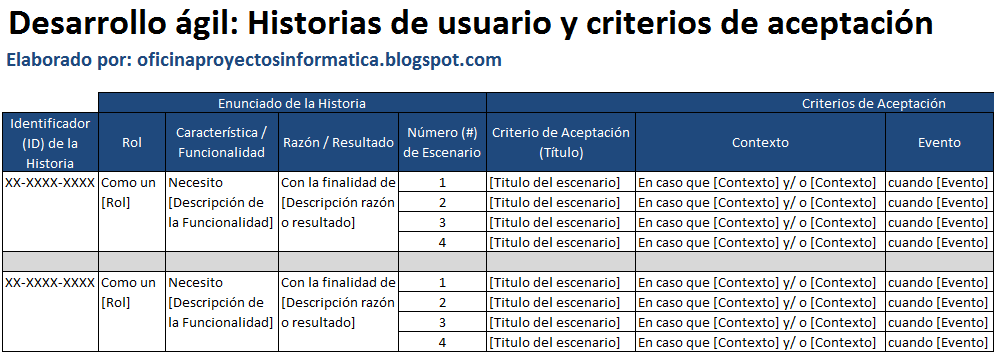
**Confirmación de los criterios de aceptación**

Los criterios de aceptación, es decir, la confirmación. Se trata de criterios claros y específicos que todo el equipo debe comprender y que permitirán avaluar en el futuro si la implementación que se está desarrollando o las pruebas que se realicen están terminadas.

A continuación, un ejemplo de una historia de usuario usando plantilla.

**Figura 10**

*Ejemplos de historias de usuario*



Nota. Tomada de La oficina de proyectos de informática (2012)



**1.3.3. *Storyboard***

Los *storyboards* son un tipo de prototipos muy utilizados, consiste básicamente en ir mostrando en una secuencia de imágenes un proceso, acción o ejercicio que se puede realizar en el sistema una vez terminado, las imágenes van mostrando la evolución del sistema conforme el usuario interactúa con el sistema.

Con esta técnica se pretende crear diferentes vistas del sistema en las primeras etapas de su implementación de la manera más rápida y barata posible [SUT02].

Una forma muy común de ejemplificar los *storyboards* es con las revistas de *cómics*, ya que van mostrando una secuencia de imágenes en cuadros con un orden establecido que permiten entender la línea de la historia contada. La técnica *storyboard* permite generar modelos o esquemas visuales como esbozos de interfaces gráficas de usuario (GUI).

A continuación, se detallan las principales características de los *storyboards*:

* Se preserva el punto de vista del proceso del negocio.
* Se puede validar un escenario.
* Se pueden validar escenarios integradores logrando una visión global.
* Son más fáciles de comprender por el usuario.
* No genera falsas expectativas.
* El usuario sigue trabajando con herramientas conocidas.
* Son fáciles de mantener o adaptar a los cambios.
* Permiten incorporar modificaciones durante la validación.

Las dos figuras siguientes muestran el ejemplo de un *storyboard* que representa un escenario de situación de vendedores de una empresa para explicar el cambio que sufrirá el trabajo, el primero representa la situación actual y el segundo como quedará con la implantación del sistema.

**Figura 11**

*Escenario representado en formato de storyboard que representa una situación típica tal y como se realiza actualmente*

Diagrama

Descripción generada automáticamente

Nota.Tomada de Granollers, Lorés y Perdrix (2002).

**Figura 12**

*Escenario representado en formato de storyboard que representa la misma situación anterior tal como quedará con la implementación del sistema*

Diagrama

Descripción generada automáticamente

Nota. Tomada de Granollers, Lorés y Perdrix (2002).

**1.4. Herramientas de modelado**

Las herramientas de modelado de sistemas informáticos se emplean para la creación de modelos de sistemas que ya existen o que se desarrollarán; estas herramientas permiten crear un "simulacro" del sistema, a bajo coste y riesgo mínimo. A bajo costo porque, es un conjunto de gráficos y textos que representan el sistema.

El Lenguaje Unificado de Modelado (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; UML es un lenguaje gráfico que permite especificar, modelar, construir y documentar los elementos que forman un sistema *software*.

De otra parte, las **herramientas CASE** son un conjunto de programas y procesos “guiados”, que ayudan a los analistas, desarrolladores, ingenieros de software y diseñadores en una o todas las etapas que comprende un ciclo de vida, con el objetivo de facilitar el desarrollo de software. El objetivo general de estas herramientas es acelerar el proceso para el que han sido diseñadas, es decir, para automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas. CASE proporciona un conjunto de herramientas semiautomatizadas y automatizadas que están creando una nueva cultura de ingeniería en muchas empresas. Las herramientas CASE se diseñaron para aumentar la productividad en el desarrollo de software y reducir su costo.

Existen muchas herramientas para modelado tanto en libres como con derechos comerciales; a renglón seguido se listan las herramientas CASE más utilizadas:

* ER win.
* [ArgoUML](https://osluz.unizar.es/aplicacion/argouml).
* Easy Case.
* Oracle Designer.
* Power Designer.
* System Architect.
* SNAP.
* [Gliffy](https://www.gliffy.com/) (https://www.gliffy.com/).
* MagicDraw.
* Lucidchart.
* Papyrus Uml.
* Modelio.
* StarUml.
* Dia.
* Mono Uml.

A continuación, se realizará una descripción del top 5 de las más utilizadas.

* [Gliffy](https://www.gliffy.com/). La aplicación en línea Gliffy es una herramienta de diagramas UML basada en la nube. Apareció por primera vez en 2006 y se trata de una herramienta de modelado que crea todo tipo de diagramas, tales como diagramas de flujo, diagramas de Venn y, por supuesto, diagramas UML. La herramienta en línea fue escrita en HTML5 y es bastante popular gracias a su rapidez de reacción. Es de anotar que antes de que Gliffy pasara por la fase beta en 2007, la empresa homónima cooperó con el grupo de software australiano Atlassian. Ya en 2006, su *software* de colaboración Confluence integró un *plugin* de Gliffy y, más tarde, el equipo de Gliffy desarrolló un *plugin* para Jira. Google Workspace y Drive de Google también integran esta herramienta UML.

**Figura 8**

*Herramienta Gliffy*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

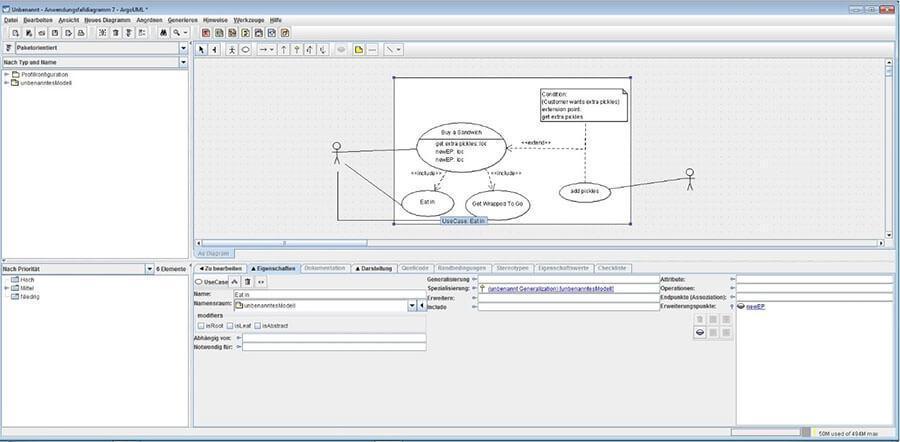
Nota. Tomada de https://www.gliffy.com/

* [ArgoUML](https://osluz.unizar.es/aplicacion/argouml). Ha sido durante mucho tiempo una de las herramientas UML gratuitas de código abierto más populares para el escritorio y aunque ya no se mantiene, muchos modeladores continúan usando el programa para tareas más pequeñas. Su *software* es multiplataforma, cuyo el requisito mínimo es Java 5 ArgoUML soporta todos los tipos de diagramas de la versión 1.4 de UML y perfiles UML. El programa también ofrece algunas formas decorativas que no forman parte del estándar UML.

Además, aunque esta herramienta *UML* está disponible como descarga gratuita, ArgoUML **soporta una amplia gama de lenguajes de programación** cuyo código puede generarse a partir de un diagrama. La ingeniería inversa también es posible para Java, C++, PHP, C# y SQL. El programa reconoce otros idiomas como Delphi o Ruby cuando los agrega como extensiones a la carpeta de archivos ArgoULM*.*

**Figura 9**

*Herramienta de modelado ArgoUML*



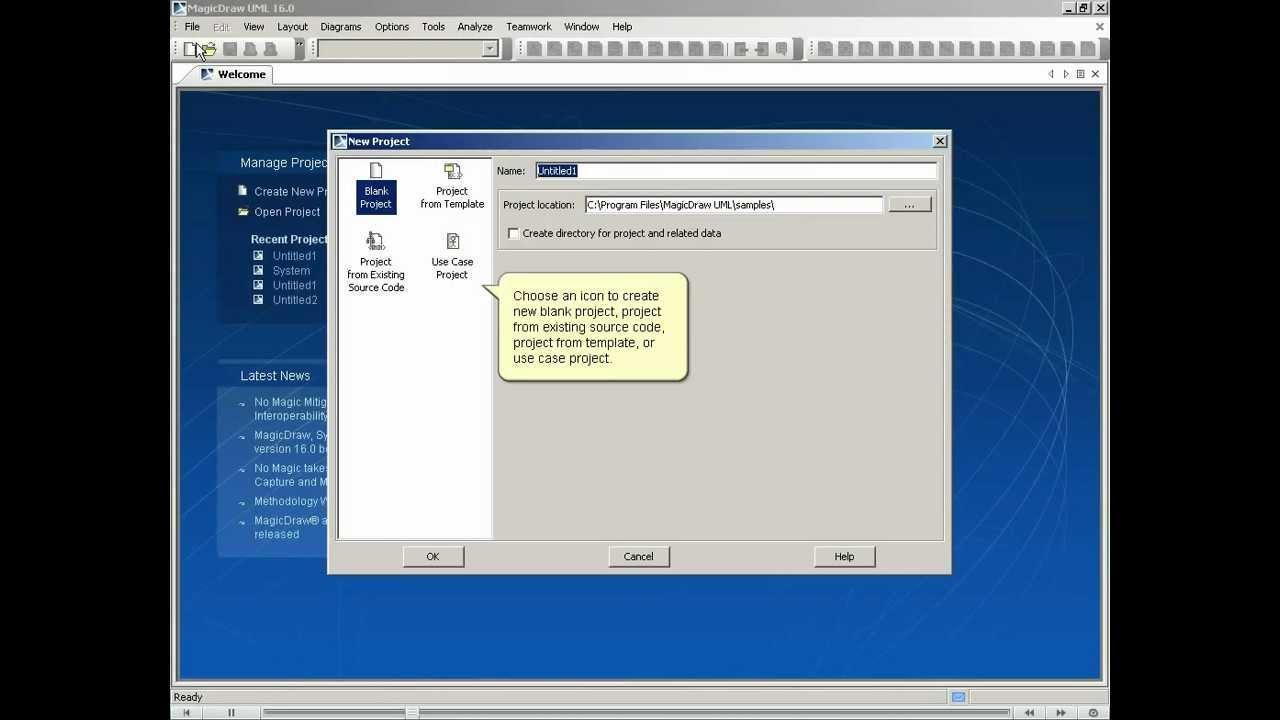
Nota. Tomada de https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/las-mejores-herramientas-uml/

* [MagicDraw](https://www.nomagic.com/products/magicdraw#intro). Esta aplicación de escritorio destaca por su diseño moderno y claro, así como por su variedad de funciones y la facilidad de su uso. Esta herramienta de diagramas UML ofrece además SysML, representación gráfica de procesos de negocio con BPMN (*Business Process Model and Notation*) y el marco de arquitectura UPDM (*United Profile for DoDAF/MODAF*).

MagicDraw también ofrece lenguaje de especificación OCL (*Object Constraint Language*), y *XMI*, que puede usar para exportar diagramas a otros programas sin pérdidas de información.

**Figura 10**

*Herramienta de modelado MagicDraw*



Nota. Tomada de [http://www.magicdraw.com](https://www.youtube.com/redirect?event=video_description&redir_token=QUFFLUhqbVBzUkhOTUhEWnBZU2IzZzdNU0lLYmR3SVZtQXxBQ3Jtc0trTVkxclVUdUhwQzZ5TTBZZ0t0OGE2ck9mZzU4TDZJRjdDMDFpcG5yN3lsX0s5dF9KQWw1M2ExeWRlbE4yclJlZGNVLU1tZHE5ajdKcTgxTG1QLWN6eEtxV0g1SGY2d0RVV1pQY1QxNk9UOW14LTk2NA&q=http%3A%2F%2Fwww.magicdraw.com)​

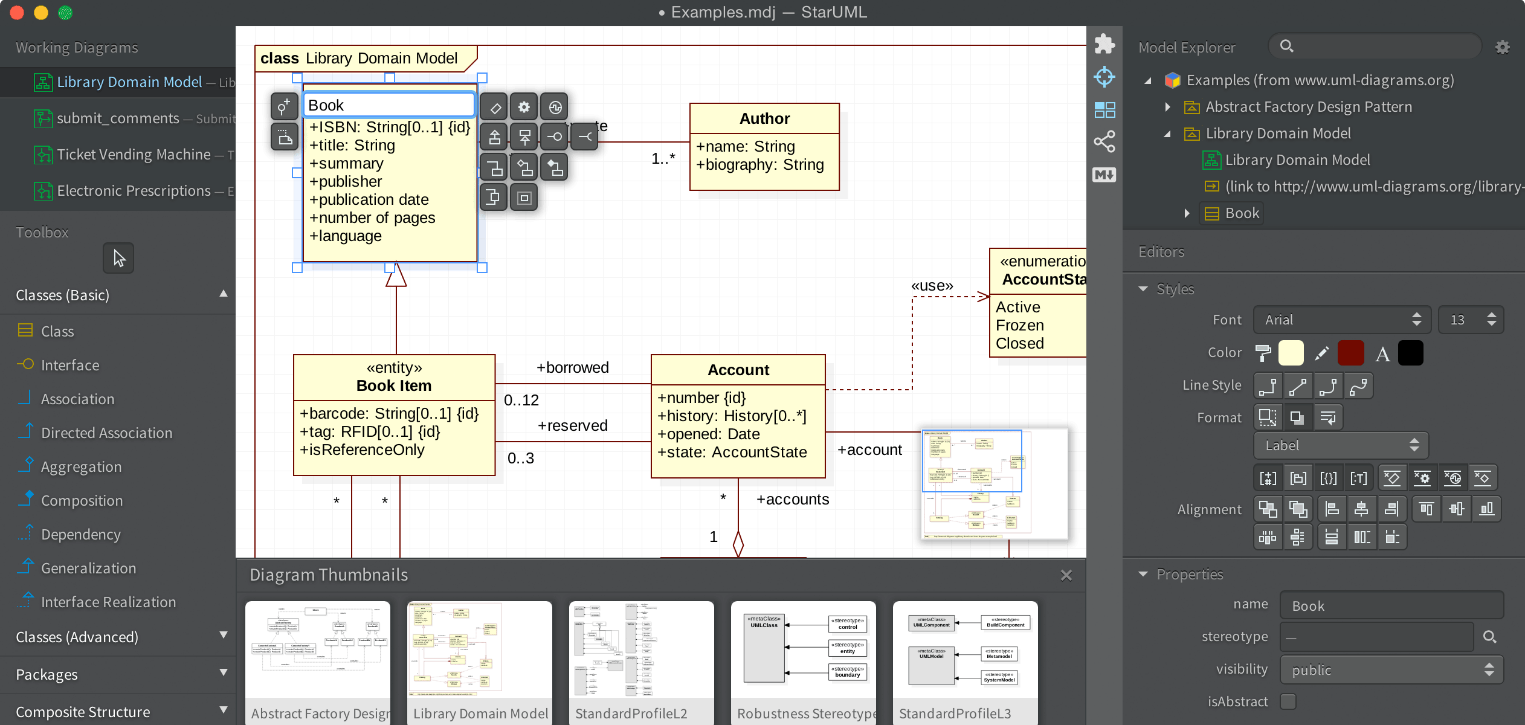
* [StarUML](http://staruml.io/). Es una herramienta para el modelamiento de software basado en los estándares UML *(Unified Modeling Language)* yMDA *(Model Driven Arquitecture).*

Da soporte completo al diseño UML mediante el uso de:

* Diagrama de casos de uso.
* Diagrama de clase.
* Diagrama de secuencia.
* Diagrama de colaboración.
* Diagrama de estados.
* Diagrama de actividad.
* Diagrama de componentes.
* Diagrama de despliegue.
* Diagrama de composición estructural (*UML* 2.0).

**Figura 11**

*Herramienta de modelado StarUML*

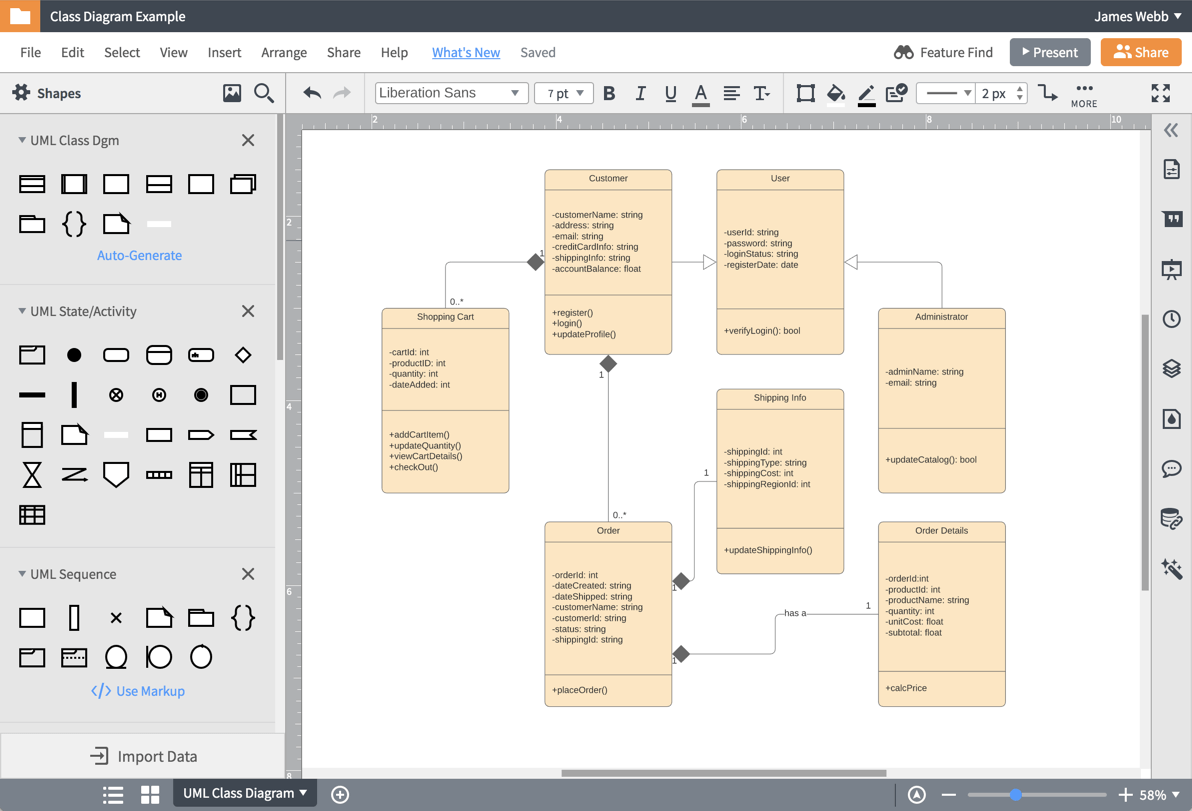


Nota. Tomada de https://staruml.io/

* [*Lucidchart*](https://www.lucidchart.com/pages/es/ejemplos/diagrama-uml?gclid=CjwKCAjwvMqDBhB8EiwA2iSmPPB-G4vkSDDSwMAN0jOdoWmZcoDBlVIcKU1vB4LVPBpZg_k_zUVuuBoCkiQQAvD_BwE&km_CPC_AdGroupID=79594404889&km_CPC_AdPosition=&km_CPC_CampaignId=2090370618&km_CPC_Country=1003669&km_CPC_Creative=376456865961&km_CPC_Device=c&km_CPC_ExtensionID=&km_CPC_Keyword=draw.io&km_CPC_MatchType=e&km_CPC_Network=g&km_CPC_TargetID=kwd-91510626377&km_CPC_placement=&km_CPC_target=&mkwid=sNVxIBevt_pcrid_376456865961_pkw_draw.io_pmt_e_pdv_c_slid__pgrid_79594404889_ptaid_kwd-91510626377_&utm_campaign=_es_tier2_desktop_search_nb_exact_&utm_medium=cpc&utm_source=google). Herramienta para hacer diagramas UML *online* que permite comprender las complejidades en el código de forma más rápida y sencilla, pues automatiza el proceso de generación de un diagrama de clases. Simplemente elabora y personaliza los diagramas de secuencia en línea a partir del texto. Al ingresar el marcado en el diálogo emergente, Lucidchart generará automáticamente un diagrama de secuencia que cumple el estándar de PlantUML

**Figura 12**

*Herramienta de modelado Lucidchart*



Nota. Tomada de www.lucidchart.com

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

Por cada componente formativo se puede proponer **un máximo de dos actividades** que los aprendices puedan realizar una vez han revisado los contenidos presentados y que refuercen la asimilación de los mismos.

**Son actividades que no generan evaluación y que funcionarán independiente del LMS.**

En este ítem deberá diligenciar la siguiente tabla, que especifica las plantillas de diseño de actividad de afianzamiento que entregará para su incorporación en el ambiente virtual.

Para cada actividad debe indicar:

|  |  |
| --- | --- |
| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| Nombre de la Actividad | Técnicas de recolección de información. |
| Objetivo de la actividad | Conocer las diferentes técnicas de recolección de información, así como sus instrumentos y buenas prácticas para ser empleadas durante el proceso de elicitación de requisitos de *software* (integralidad). |
| Tipo de actividad sugerida | Conocer las diferentes técnicas de recolección de información, así como sus instrumentos y buenas prácticas para ser empleadas durante el proceso de elicitación de requisitos de *software* (integralidad). |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo\_Actividad \_didactica 1\_CF002 |

1. **MATERIAL COMPLEMENTARIO:**

Relacionar el material de apoyo o complementario de los temas abordados en este recurso.

|  |  |  |  |
| --- | --- | --- | --- |
| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del Recurso o  Archivo del documento o material |
| Entradas, herramientas y salidas. | Todopmp. (2019). *PMBOK 6 – 49. Procesos y entradas en forma de tarjetas.* https://todopmp.com/cards | Página web | <https://todopmp.com/cards/> |
| Matriz *stakeholders.* | EDAP - Project Business School (2016). *Identificar interesados.* [Video]. YouTube. https://www.youtube.com/watch?v=aUkTxgaajBo | Video | <https://www.youtube.com/watch?v=aUkTxgaajBo> |
| Virtual Training Lteam (2016). *Partes interesadas Stakeholders.* [Video]. YouTube. https://www.youtube.com/watch?v=9AtaIAZEu0c | Video | <https://www.youtube.com/watch?v=9AtaIAZEu0c> |
| Calle, M. (2016). *Análisis de interesados matriz poder interés.* [Video]. YouTube. https://www.youtube.com/watch?v=hDZ0uu0H1wc | Video | <https://www.youtube.com/watch?v=hDZ0uu0H1wc> |
| Ejemplos de entrevistas. | Jibaro X. (s.f.). *Tipos de preguntas en una encuesta.* [Video]. YouTube. https://www.youtube.com/watch?v=mwnQuUi9014 | Video | https://www.youtube.com/watch?v=mwnQuUi9014 |

1. **GLOSARIO:**

|  |  |
| --- | --- |
| TÉRMINO | SIGNIFICADO |
| Ciclo de vida *software* | Se refiere a la aplicación de metodologías para el desarrollo del sistema *software* (AECC, 1986). |
| Método | Indica cómo construir técnicamente el *software*. Se incluyen técnicas de modelado y otras técnicas descriptivas. |
| Metodología | Colección de métodos para resolver un tipo de problemas. |
| Ágil | Comprende un conjunto de tareas o acciones que se utilizan para producir y mantener productos, así como para lograr los objetivos del proceso. La actividad incluye los procedimientos, estándares, políticas y objetivos para crear y modificar un conjunto de productos de trabajo. |
| Requerimiento | El requerimiento se refiere a la petición que se hace de algo que se solicita. |
| Requisito | Es la condición que debe cumplir algo, en general el requisito cumple con lo que se requiere con el requerimiento. |
| *Stakeholders* | Individuo u organización que comparte, reclama o le interesa un sistema o le compete una característica que satisface sus necesidades y expectativas (ISO 29148). |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

Baar, B. (2006). *Using Stakeholder Analysis in Software Project Management*. Universidad de Amsterdam.

Braude, J. (2003). *Ingeniería de software, una perspectiva orientada a objetos*. Alfaomega.

Cohen, L. (2011). *Métodos de investigación educativa*. La Muralla.

Cohn, M. (2004). *User Stories Applied for Agile Software Development.* Pearson Education, Inc.

Cox, K., Niazi, M., y Verner, J. (2009). Empirical study of Sommerville and Sawyer’s requirements engineering practices. *IET Software, 3*(5), 339. https://doi.org/10.1049/iet-sen.2008.0076

Curso de interacción persona-ordenador. (2021). *Storyboarding*. https://mpiua.invid.udl.cat/storyboarding

Denscombe, M. (2010). *The Good Research Guide.* McGraw-Hill Education.

Dornyei, Z. (2010). *Questionnaires in Second Language Research: Construction, Administration, and Processing*. Routledge.

Durán, A., Bernárdez, B., Ruiz, A. y Toro, M. (1999). *A Requirements Elicitatio Approach Based in Templates and Patterns.* <https://www.researchgate.net/publication/2890318_A_Requirements_Elicitation_Approach_Based_in_Templates_and_Patterns>

Gause, C., & Weinberg, G. M. (1989). *Exploring Requirements: Quality Before Design.* Dorset House.

Granollers, T., Lorés, J., y Perdrix F. (2002). *Prototipado*. Capítulo 5: modelo de proceso de la ingeniería de la usabilidad y de la accesibilidad. <https://docplayer.es/13733635-5-6-1->

Gutiérrez, J. (s.f.). *Diagramas UML de casos de uso y de requisitos.* http://www.lsi.us.es/~javierj/cursos\_ficheros/metricaUML/CasosUsoUML.pdf

Hernández, S., Fernández C., y Baptista L. (2006). *Metodología de la investigación.* McGraw Hill.

Herrera J., Lizka J. (2003). *Ingeniería de requerimientos, ingeniería de software.* http://www.monografias.com/trabajos6/resof/resof.shtml

La oficina de proyectos de informática (2012). *PMOinformatica.com.* http://www.pmoinformatica.com/2012/10/plantillas-scrum-historias-de-usuario.html

Piattini M., Calvo-Manzano J., Cervera J., y Fernández, L. (2004). *Análisis y diseño de aplicaciones informáticas de gestión. Una perspectiva de ingeniería de software.* Alfaomega-Rama.

Sommerville, I. (2011). *Ingeniería de software.* Pearson Educación.

Ventura, M. T. (2002). *La ingeniería de requerimientos como factor clave para el éxito de los proyectos de desarrollo de software*. <http://132.248.9.195/ppt2002/0307178/Index.html>

Wessinger, K., (2012) Project Stakeholder Management. *Engineering Management Journal, 14*(84), 19-24.

1. **CONTROL DEL DOCUMENTO**

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

1. **CONTROL DE CAMBIOS**

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

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