MODULO 1 La fase de elicitación de requisitos

La recolección de datos implica el uso de diversas técnicas y herramientas, como entrevistas, encuestas, cuestionarios, observación, sesiones en grupo y visitas a instalaciones. Cada una de estas técnicas cuenta con instrumentos específicos que aseguran un desarrollo profesional y confiable de los sistemas de información.

El material describe detalladamente el proceso de recolección de datos, destacando el uso de diversas técnicas y herramientas por parte del analista para desarrollar sistemas de información. Estas técnicas incluyen entrevistas, encuestas, cuestionarios, observación, sesiones grupales y recolección documental, entre otras. Los analistas combinan varios métodos, como entrevistas, cuestionarios, inspección de registros y observación, para obtener una comprensión completa de una situación. Cada método tiene sus propias ventajas y desventajas, por lo que se suelen usar dos o tres simultáneamente para complementar el trabajo y asegurar una investigación exhaustiva.

La elicitación de requisitos es crucial para entender el problema a resolver y producir una especificación formal del software necesario. Un problema se define como la diferencia entre la percepción actual y el estado deseado (Gause y Weinberg, 1989).

Para lograr una elicitación efectiva, es vital una buena comunicación entre desarrolladores y clientes, lo cual asegura que las necesidades del cliente queden claras. Al final de esta fase, el analista debe tener un conocimiento profundo del dominio del problema.

La elicitación de requisitos, considerada como el primer paso en la ingeniería de requisitos, implica el uso de técnicas para recopilar información. Los objetivos de esta fase son:

1. Comprender el dominio del problema para comunicarse efectivamente con clientes y usuarios.

2. Conocer el sistema actual, ya sea manual o informatizado, identificando sus aspectos positivos y negativos.

3. Identificar las necesidades explícitas e implícitas de clientes y usuarios, así como sus expectativas sobre el sistema a desarrollar.

**1.1. 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.



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

### Resumen: Identificación de Fuentes de Requerimientos

#### Fuentes de Requerimientos en Proyectos de Desarrollo de Software

En los proyectos de desarrollo de software, existen diversas fuentes de requerimientos que se clasifican en:

1. \*\*Gente (People)\*\*

- \*\*Usuarios y Expertos\*\*: Proveen información detallada sobre problemas y necesidades del usuario.

- \*\*Stakeholders (Interesados)\*\*: Incluyen a todos los involucrados relevantes que pueden aportar requerimientos.

2. \*\*Documentos y Sistemas Existentes\*\*

- \*\*Documentación Existente\*\*: Manuales, formularios, reportes y especificaciones previas que ofrecen información sobre la organización y su entorno.

- \*\*Procesos y Sistemas Actuales\*\*: Representan fuentes adicionales de requisitos basados en la operativa vigente.

#### Clasificación de Fuentes de Requerimientos

1. \*\*Fuentes Primarias\*\*

- Proveen información original y sin filtrar.

- Ejemplos: Entrevistas directas, observaciones de usuarios, datos crudos de experimentos.

2. \*\*Fuentes Secundarias\*\*

- Reproducen y reorganizan la información de las fuentes primarias.

- Ejemplos: Artículos de revistas, libros, informes oficiales, bases de datos.

3. \*\*Fuentes Terciarias\*\*

- Contienen referencias a fuentes secundarias, facilitando el acceso a información sintetizada.

- Ejemplos: Bibliografías, guías de referencia, índices.

#### Tipos de Transmisión de Información

- \*\*Orales\*\*: Entrevistas, reuniones, discusiones.

- \*\*Escritas\*\*: Documentos, reportes, manuales.

- \*\*Otros Medios\*\*: Videos, grabaciones, bases de datos digitales.

### Conclusión

Las fuentes de requerimientos son esenciales para recopilar toda la información necesaria en un proyecto de software. Estas pueden ser personas, documentos o sistemas, y es crucial verificar toda la información con los stakeholders para asegurar la validez y relevancia de los requerimientos.



### Identificación de Interesados del Producto

Uno de los primeros pasos en el proceso de desarrollo de software es identificar y analizar a todas las personas y organizaciones interesadas en el proyecto, conocidos como stakeholders. Estos son individuos u organizaciones que tienen una relación directa o indirecta con el proyecto y pueden influir en los requisitos o verse afectados por los resultados del proyecto.

#### Tipos de Stakeholders

1. \*\*Primarios\*\*:

- \*\*Definición\*\*: Indispensables para el funcionamiento de la organización con una relación económica directa.

- \*\*Ejemplos\*\*: Socios, clientes y accionistas.

2. \*\*Secundarios\*\*:

- \*\*Definición\*\*: No participan directamente en la compañía pero son afectados por sus resultados.

- \*\*Ejemplos\*\*: Competidores, el mercado y personas en general.

#### Roles Generales de los Stakeholders

- \*\*Líder de proyecto / Administrador de proyecto / Gerente de proyecto\*\*: Responsable de la dirección y administración del proyecto.

- \*\*Analista / Ingeniero de requisitos\*\*: Encargado de la recopilación y análisis de los requisitos.

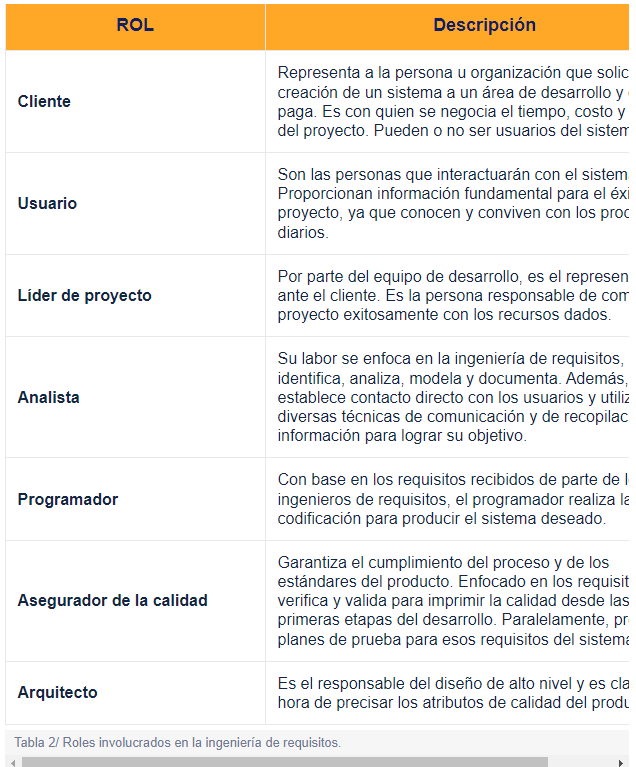
- \*\*Ingeniero de sistemas / Arquitecto\*\*: Diseña la estructura del sistema.

- \*\*Programador / Desarrollador / Ingeniero de software\*\*: Desarrolla el código del software.

- \*\*Probador / Asegurador de la calidad\*\*: Verifica y asegura la calidad del software.

- \*\*Administrador de bases de datos\*\*: Gestiona y mantiene las bases de datos.

La identificación y clasificación de estos roles es crucial para asegurar que todos los interesados estén informados sobre el progreso, conflictos, cambios y prioridades del desarrollo del producto.



Para realizar un análisis de los stakeholders, se pueden emplear varias técnicas, como entrevistas con expertos, sesiones de lluvia de ideas en grupo y listas de chequeo. Es crucial que el grupo encargado de esta tarea incluya a personas de diversos contextos para identificar y caracterizar a los stakeholders de manera objetiva (Karisen, 2002 citado en Wessinger, 2012).

### Matriz de Stakeholders

La matriz de stakeholders es una herramienta que clasifica a los involucrados en el proyecto según sus niveles de interés y poder. Esto ayuda a priorizar a los stakeholders más importantes y desarrollar estrategias de gestión adecuadas.

### Importancia de la Matriz de Stakeholders en Proyectos de Desarrollo

En proyectos de desarrollo, gestionar adecuadamente a los stakeholders es vital para el éxito del proyecto. Identificar a los involucrados y definir sus niveles de interés e influencia desde el principio es esencial para crear estrategias que aseguren el apoyo necesario. La matriz de stakeholders es indispensable desde el inicio del proyecto porque proporciona información crucial para manejar las expectativas de los involucrados, maximizar influencias positivas y mitigar impactos negativos potenciales. Además, involucrar a la sociedad civil debe ser una oportunidad para obtener su apoyo, no solo un ejercicio de comunicación unidireccional.

### Proceso de Armado de la Matriz de Stakeholders

Para desarrollar la matriz de stakeholders, es necesario identificar las entradas necesarias, como el acta de constitución del proyecto, documentos de adquisición, activos de los procesos y factores ambientales de la organización. Con esta información, el líder y el equipo de proyecto pueden elaborar la matriz.



##### 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.

**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.

La matriz de stakeholders es una herramienta esencial para clasificar a los involucrados en un proyecto según su interés e influencia. Esta clasificación permite priorizar a los más importantes y desarrollar estrategias para gestionar sus expectativas. Cabe destacar que esta clasificación puede cambiar a lo largo del proyecto; stakeholders inicialmente con alta influencia pueden ser reclasificados con menor influencia en etapas posteriores.

La categorización se realiza una vez que se tiene información completa sobre los stakeholders, utilizando una matriz 2x2 que grafica el poder e interés de cada uno. Esto facilita la clasificación y definición de estrategias específicas para cada grupo.



\*\*Resumen de Técnicas e Instrumentos para Elicitar Requisitos:\*\*

La ingeniería de requerimientos (IR) utiliza diversas técnicas para recolectar y definir los requisitos de un proyecto. Estas técnicas se aplican a diferentes fases del proceso de IR, dependiendo de las características específicas del proyecto, con el fin de maximizar su efectividad.

\*\*Entrevista:\*\*

Una técnica clave en la elicitación de requisitos es la entrevista, la cual consiste en obtener información a través de una comunicación interpersonal estructurada. Las entrevistas se dividen en tres fases principales:

1. \*\*Preparación:\*\* Planificación de la entrevista, incluyendo la identificación de los objetivos y la elaboración de preguntas.

2. \*\*Realización:\*\* Conducción de la entrevista mediante una conversación estructurada con el entrevistado.

3. \*\*Análisis:\*\* Evaluación y síntesis de la información obtenida durante la entrevista.

Estas fases ayudan a asegurar que la información recopilada sea precisa y relevante para el desarrollo del proyecto.

### Preparación para la entrevista

1. \*\*Documentarse sobre la organización\*\*: Investigar la situación de la empresa y analizar los documentos disponibles.

2. \*\*Minimizar el número de entrevistados\*\*: Evitar entrevistas de cortesía innecesarias.

3. \*\*Analizar el perfil de los entrevistados\*\*: Evaluar las características y cualidades de los participantes.

4. \*\*Definir el objetivo y contenido\*\*: Clarificar el propósito y los temas a tratar en la entrevista.

5. \*\*Planificar lugar y hora\*\*: Elegir un lugar confortable y un horario adecuado para la entrevista.

6. \*\*Enviar cuestionario previo\*\*: Enviar al entrevistado un cuestionario y un documento introductorio sobre el proyecto.

En la realización de entrevistas, según Piattini et al. (1996), se distinguen tres etapas principales:

1. \*\*Apertura\*\*: Presentarse y explicar al entrevistado el motivo de la entrevista.

2. \*\*Desarrollo\*\*: Seguir las reglas del protocolo y acordar cómo se registrará la información obtenida. Durante esta fase se pueden utilizar diversas técnicas:

- \*\*Preguntas abiertas\*\*: También llamadas de libre contexto, permiten una mayor comunicación y evitan respuestas cerradas. Ejemplos: "¿Qué se hace para registrar un pedido?", "Dígame qué se debe hacer cuando un cliente pide una factura" o "¿Cómo se rellena un recibo?".

- \*\*Uso de palabras apropiadas\*\*: Evitar tecnicismos desconocidos para el entrevistado y términos que puedan afectar emocionalmente la comunicación.

- \*\*Mostrar interés constantemente\*\*: Cuidar la comunicación no verbal, como el tono de voz, movimientos y expresiones faciales.

3. \*\*Terminación\*\*: Recapitular la entrevista, agradecer el esfuerzo y dejar abierta la posibilidad de futuros contactos para aclaraciones o nuevas entrevistas.

El análisis de entrevistas en ingeniería de requerimientos implica leer, organizar y contrastar la información obtenida. Se realizan preguntas sobre el sistema actual y el sistema a desarrollar, cuyas respuestas generan los requerimientos del proyecto.

**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.

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.

En el material complementario se pueden revisar ejemplo de entrevistas.

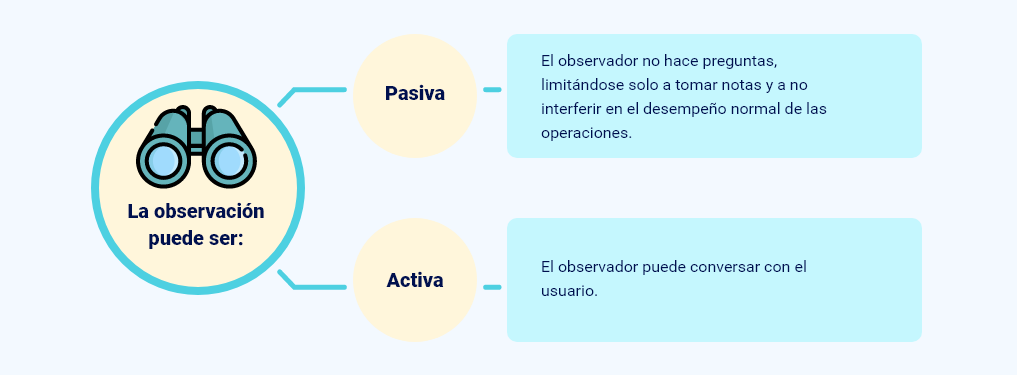
Los cuestionarios son herramientas comúnmente usadas para recolectar datos de sondeos y pueden ser administrados sin la presencia del investigador (Cohen, 2011, p. 377). Varían en propósito, diseño y apariencia, consistiendo en listas de preguntas escritas. Los participantes responden a las mismas preguntas, lo que asegura consistencia y precisión en los datos. Esta metodología simplifica la obtención de datos al preguntar directamente a los participantes, permitiendo alcanzar a un gran número de sujetos rápidamente. Los datos obtenidos se clasifican en hechos y opiniones, donde los hechos no requieren juicio personal, mientras que las opiniones reflejan creencias y preferencias de los participantes (Denscombe, 2010, p. 156).

Existen dos tipos principales de preguntas en los cuestionarios: abiertas y cerradas. Las preguntas abiertas permiten al participante escribir su respuesta sin restricciones, mientras que las cerradas ofrecen opciones predefinidas para elegir.

Las respuestas de escala son comunes en los cuestionarios de investigación. Estas implican que el participante valore o evalúe las respuestas objetivo mediante opciones dentro de una escala. Esta escala puede indicar diferentes grados dentro de una categoría, como "bueno" o "malo", "frecuente" o "infrecuente", o incluso opiniones como "completamente de acuerdo" o "en desacuerdo". Las escalas suelen tener cinco opciones, ya que los números impares tienden a favorecer la selección de la opción intermedia.

La observación en investigación cualitativa consiste en la recopilación directa de datos a partir de lo que el investigador percibe en primera instancia, sin depender únicamente de lo que los sujetos expresan verbalmente. Selltiz plantea que, para considerarse una técnica, debe cumplir cuatro condiciones: estar dirigida a un objetivo de investigación, ser planificada sistemáticamente, estar controlada y relacionada con proposiciones generales, y ser sujeta a comprobaciones de validez y fiabilidad. En el contexto de la obtención de requerimientos de software, la observación se utiliza para estudiar el entorno de trabajo de los usuarios, clientes e interesados en el proyecto y para documentar la situación actual de los procesos de negocio.

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



La observación se lleva a cabo mediante la guía de observación, que facilita la recolección sistemática de datos e información sobre un fenómeno. Tamayo (2004) la define como un formato para recolectar datos de manera uniforme y objetiva, estructurada según las variables del problema. La guía debe incluir datos de los sujetos, propósitos y temporalidad de la observación. Además, en el contexto de sesiones grupales, se utilizan técnicas como la lluvia de ideas, que fomenta la creatividad mediante la interacción y el trabajo en equipo.

Las sesiones JAD son talleres donde trabajadores del conocimiento y expertos en tecnologías de la información se reúnen para definir y revisar los requerimientos de negocio de un nuevo sistema de información. Estos talleres, que pueden durar varios días, cuentan con la participación de altos directivos para garantizar que el producto final cumpla con las necesidades de reportes e información. El proceso sigue una agenda detallada para resolver posibles diferencias y evitar fallas de comunicación, asegurando así la creación de un nuevo sistema orientado tanto a diseñadores como a usuarios.

El Método Delphi es una técnica de comunicación grupal que implica la selección de expertos para obtener opiniones sobre ciertos temas. Tiene cuatro fases: formulación del problema, elección de expertos, elaboración de cuestionarios y desarrollo de resultados a partir de las respuestas de los expertos.

### 1.3. Herramientas para captura de requisitos

Para capturar los requisitos de un nuevo sistema o una actualización de software, se utilizan diversas herramientas. A continuación, se describen las más comunes:

#### 1.3.1. Diagrama de casos de uso

Los diagramas de casos de uso son esenciales para identificar las funcionalidades principales que el software debe ofrecer y los usuarios que podrán ejecutarlas. Estos diagramas se elaboran en las etapas iniciales del proyecto y sirven como referencia en todas las etapas del desarrollo.

\*\*Componentes de un diagrama de casos de uso:\*\*

- \*\*Actor:\*\* Representado por un "hombre de palo", indica el tipo de usuario que interactuará con el sistema.

- \*\*Caso de uso:\*\* Representado por un óvalo, indica una función que el sistema debe proporcionar. Se suele usar un verbo en infinitivo para describir la función (administrar, gestionar, registrar, etc.).

\*\*Ejemplo de un diagrama de casos de uso para un centro médico:\*\*

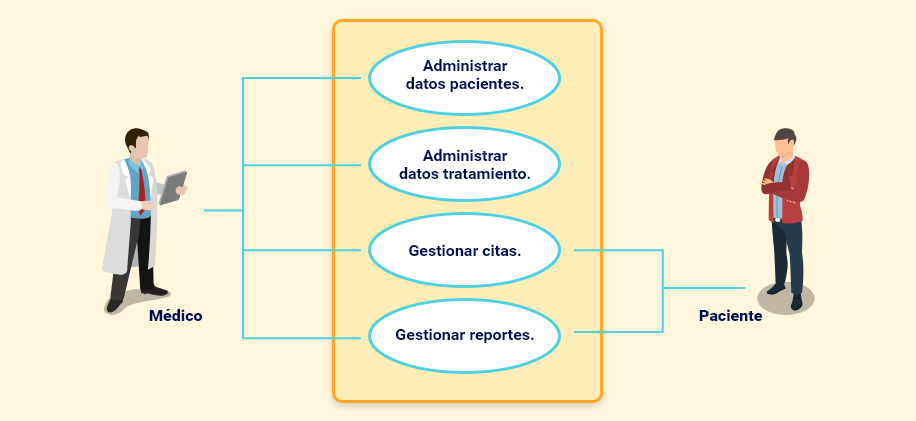
- Administrar datos pacientes.

- Administrar datos tratamientos.

- Gestionar citas.

- Generar reportes.

Representación gráfica



\*\*Identificación de casos de uso\*\*

Se identifican las funcionalidades que el sistema proveerá: administrar datos de pacientes, administrar datos de tratamientos, gestionar citas y generar reportes.

\*\*Identificación de actores\*\*

Se identifican los usuarios que podrán ejecutar los casos de uso: médico y empleado.

\*\*Documentación\*\*

La técnica de casos de uso no solo implica la construcción del diagrama de casos de uso, sino también la descripción detallada de estos. Esta descripción especifica el flujo de eventos entre el Sistema y el Actor para realizar cada caso de uso. Se presenta un formato diligenciado según el ejemplo del centro médico.

##### 1.3.2. Historias de usuario.

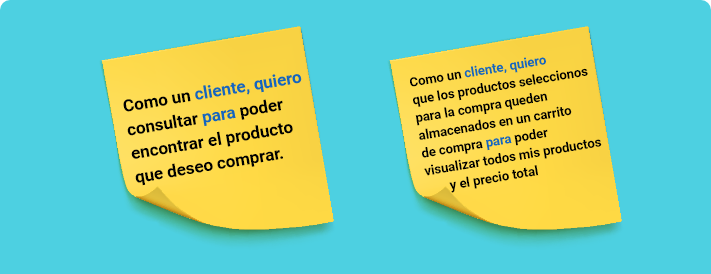
Las historias de usuario son utilizadas en los métodos agiles para la especificación de requisitos, son una descripción breve de una funcionalidad software tal y como la percibe el usuario (Cohn, 2004).

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



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.



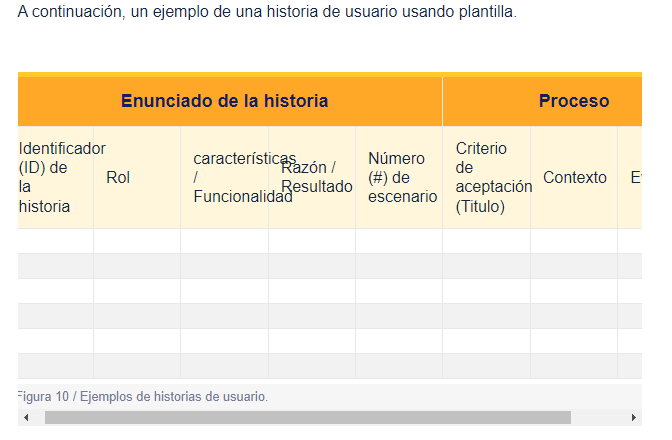
Para entender mejor una historia de usuario, es crucial mantener una conversación detallada entre todos los miembros del equipo. Esta conversación tiene varios objetivos clave:

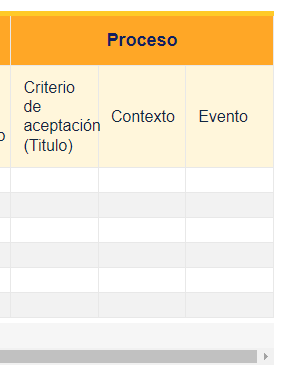
1. \*\*Detallar la solución\*\*: Definir cómo se llevará a cabo la implementación.

2. \*\*Clarificar aspectos\*\*: Aclarar detalles sobre el valor, funcionamiento y aspectos técnicos del proyecto.

3. \*\*Resolver dudas\*\*: Responder a cualquier pregunta que surja.

Estas discusiones ayudan a alcanzar acuerdos sobre diversos puntos, los cuales se reflejarán en los criterios de aceptación. Estos criterios son específicos y comprensibles para todo el equipo, y sirven para evaluar si la implementación o las pruebas están completas.





### ¿Qué es un Storyboard?

Un storyboard es un prototipo que muestra, mediante una secuencia de imágenes, cómo un usuario interactúa con un sistema. Este método es usado en las primeras etapas de implementación para visualizar diferentes vistas del sistema de manera rápida y económica.

#### Principales características de los storyboards:

- \*\*Preservación del punto de vista del proceso del negocio\*\*: Mantiene la perspectiva del flujo de trabajo empresarial.

- \*\*Validación de escenarios\*\*: Permite comprobar la viabilidad de un escenario.

- \*\*Visión global de escenarios integradores\*\*: Ayuda a tener una perspectiva completa del sistema.

- \*\*Comprensión fácil para el usuario\*\*: Los usuarios pueden entenderlos fácilmente.

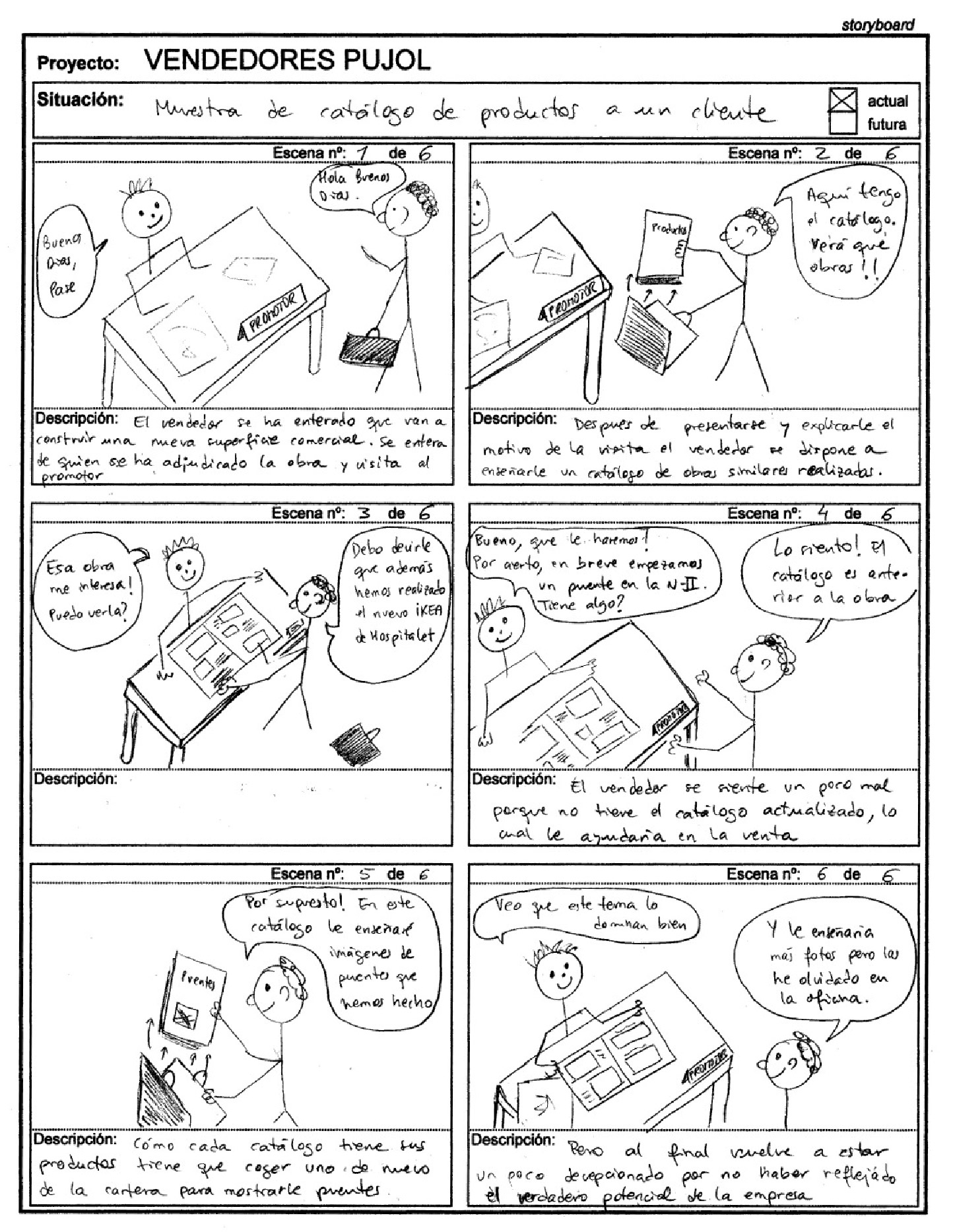
- \*\*No genera falsas expectativas\*\*: Presenta un modelo realista y alcanzable.

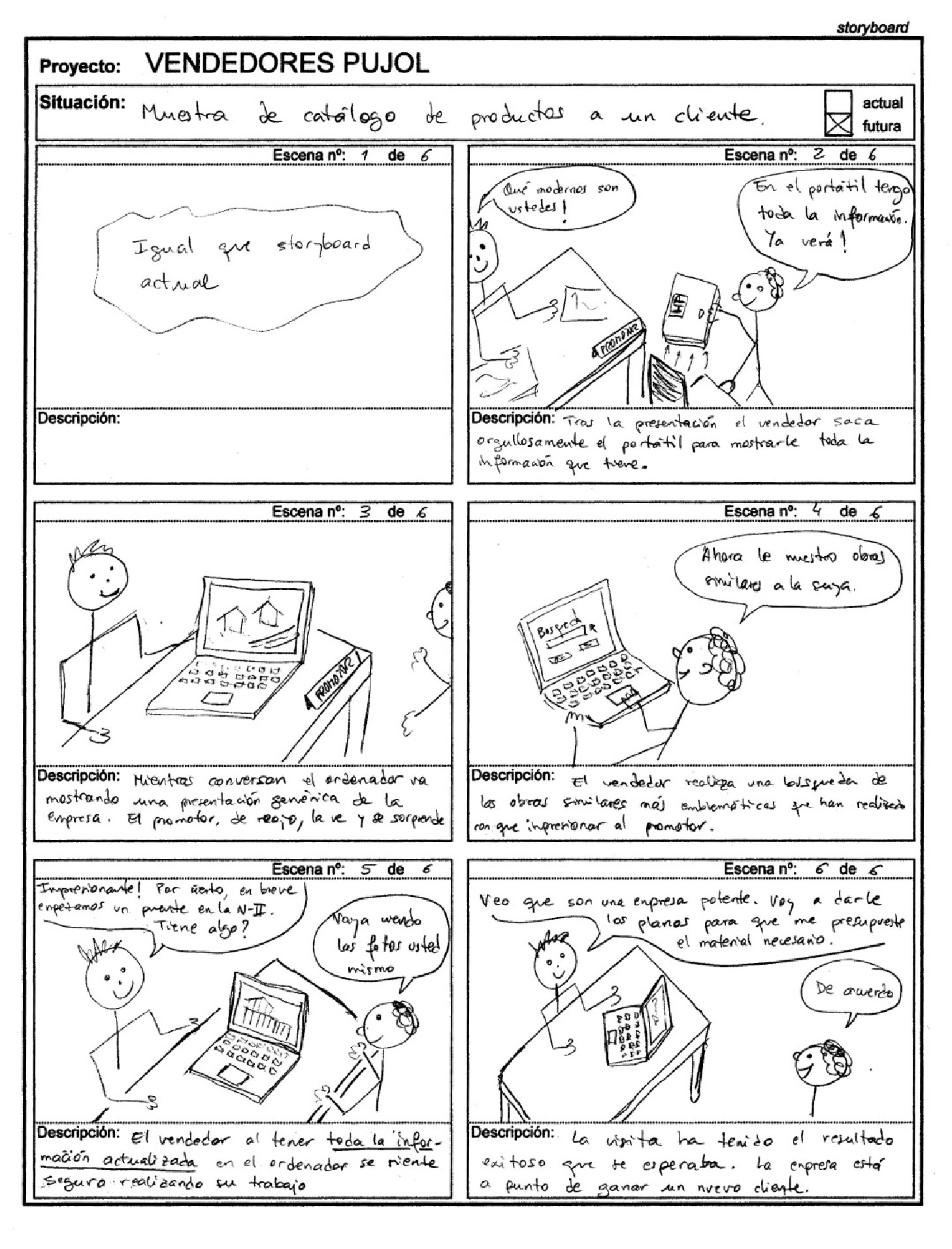
- \*\*Uso de herramientas conocidas por el usuario\*\*: Facilita la adopción y comprensión.

- \*\*Facilidad de mantenimiento y adaptación\*\*: Son fáciles de actualizar según los cambios.

- \*\*Incorporación de modificaciones durante la validación\*\*: Permiten ajustes sobre la marcha.

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.

Figura 12 /Escenario representado en formato de storyboard que representa la misma situación anterior tal como quedará con la implementación del sistema.

### 1.4. Herramientas de Modelado

Las herramientas de modelado permiten crear un "simulacro" del sistema a bajo costo y con mínimo riesgo, utilizando gráficos y textos para representar el sistema.

El Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado, permitiendo especificar, modelar, construir y documentar los elementos de un sistema software mediante gráficos.

Las herramientas CASE son programas y procesos guiados que asisten a los analistas, desarrolladores, ingenieros de software y diseñadores en todas las etapas del ciclo de vida del desarrollo de sistemas. Estas herramientas buscan acelerar el desarrollo de software, automatizando o apoyando diversas fases del ciclo de vida, incrementando la productividad y reduciendo costos.

Existen tanto herramientas de modelado libres como comerciales. Las herramientas CASE más utilizadas son:

- ER win

- ArgoUML

- Easy Case

- Oracle Designer

- Power Designer

- System Architect

- SNAP

- Gliffy

- MagicDraw

- Lucidchart

- Papyrus UML

- Modelio

- StarUML

- Dia

- Mono UML

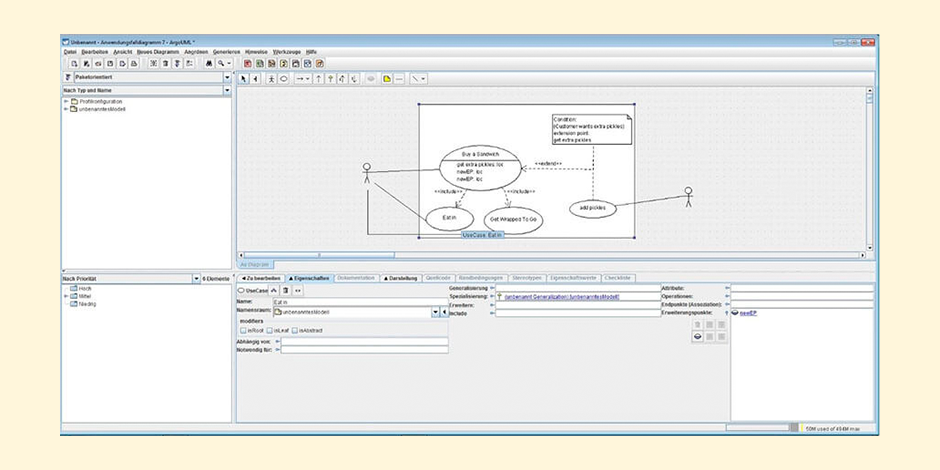
A continuación, se realizará una descripción del top 5 de las más utilizadas.

Gliffy es una herramienta de diagramas UML basada en la nube, lanzada en 2006. Permite crear diversos tipos de diagramas, como diagramas de flujo, de Venn y UML, y está escrita en HTML5, lo que la hace rápida y popular. Antes de su fase beta en 2007, Gliffy colaboró con el grupo de software Atlassian, integrando su plugin en Confluence y desarrollando uno para Jira. Además, Google Workspace y Google Drive también integran esta herramienta UML.

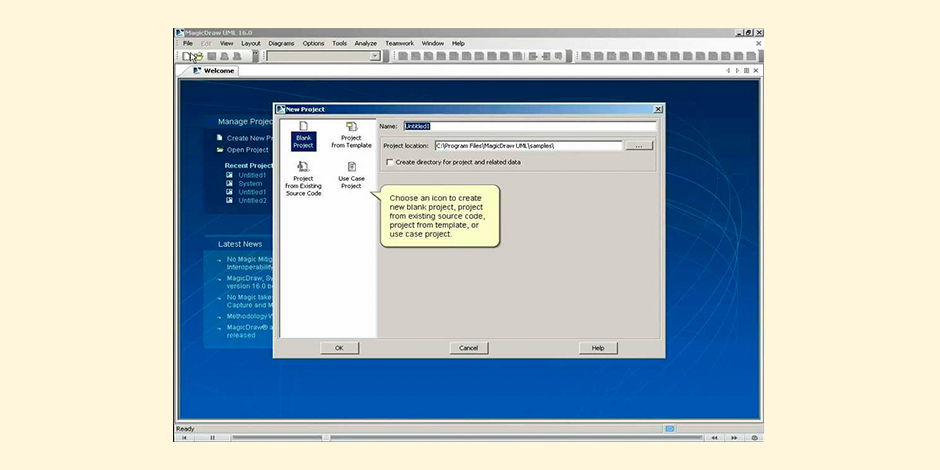


ArgoUML ha sido una de las herramientas UML gratuitas de código abierto más populares para el escritorio, aunque ya no se mantiene. A pesar de ello, muchos modeladores aún la usan para tareas menores. Es multiplataforma y requiere como mínimo Java 5. ArgoUML soporta todos los tipos de diagramas de la versión 1.4 de UML y perfiles UML, además de ofrecer formas decorativas no estándar.

La herramienta está disponible como descarga gratuita y soporta una amplia gama de lenguajes de programación, generando código a partir de diagramas. También permite la ingeniería inversa para Java, C++, PHP, C# y SQL. Además, reconoce otros lenguajes como Delphi o Ruby si se añaden como extensiones en su carpeta de archivos.



Esta aplicación de escritorio se destaca por su diseño moderno y claro, variedad de funciones y facilidad de uso. Ofrece diagramas UML, SysML, BPMN (Business Process Model and Notation) y UPDM (United Profile for DoDAF/MODAF). Además, incluye el lenguaje de especificación OCL (Object Constraint Language) y XMI para exportar diagramas sin pérdida de información.



Es una herramienta para el modelamiento de software que utiliza los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture). Ofrece soporte completo para el diseño UML a través de varios diagramas:

- Diagrama de casos de uso

- Diagrama de clase

- Diagrama de secuencia

- Diagrama de colaboración

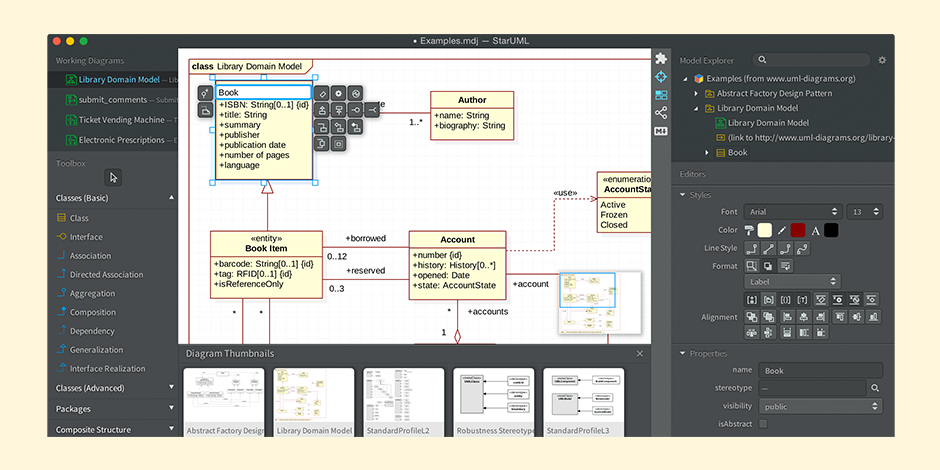
- Diagrama de estados

- Diagrama de actividad

- Diagrama de componentes

- Diagrama de despliegue

- Diagrama de composición estructural (UML 2.0)



Lucidchart es una herramienta en línea para crear diagramas UML que facilita la comprensión del código al automatizar la generación de diagramas de clases. Permite elaborar y personalizar diagramas de secuencia a partir de texto, generando automáticamente un diagrama conforme al estándar de PlantUML al ingresar el marcado en un diálogo emergente.

