**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

| PROGRAMA DE FORMACIÓN | ANÁLISIS Y DESARROLLO DE SOFTWARE |
| --- | --- |

| COMPETENCIA | 220501093 - Análisis de la especificación de requisitos del software | RESULTADOS DE APRENDIZAJE | 220501093-2 - Modelar las funciones del software de acuerdo con el informe de requisitos. |
| --- | --- | --- | --- |

| NÚMERO DEL COMPONENTE FORMATIVO | 7 |
| --- | --- |
| NOMBRE DEL COMPONENTE FORMATIVO | Diagramas para la especificación y análisis de requisitos |
| BREVE DESCRIPCIÓN | En este componente formativo se abordan los saberes de ingeniería de requisitos para el levantamiento de requerimientos que permiten tener una visión del comportamiento del software que se va a desarrollar. La descripción de funcionalidades se realiza con Diagramas y elementos como casos de uso, historias de usuario que describen las interacciones del usuario con el software. |
| PALABRAS CLAVE | Uml, casos de uso, historias usuario, prototipos, requerimientos |

| ÁREA OCUPACIONAL | 2 - CIENCIAS NATURALES, APLICADAS Y RELACIONADAS |
| --- | --- |
| IDIOMA | Español |

1. **TABLA DE CONTENIDO:**

**Introducción**

1. Características lenguaje Unificado de Modelado UML

1.1 Evolución del Lenguaje Unificado de Modelado UML

1.2.Características generales de Lenguaje Unificado de Modelado UML

2. Especificación y análisis de requisitos

2.1 Diagrama de casos de uso

2.2 Componentes de los diagramas de casos de uso

2.2.1 Actor

2.2.2 Relación

2.2.3 Casos de Uso

2.3 Relaciones de Casos de uso

2.4 Especificación de casos de uso

2.5 Casos de uso reales (prototipos)

3. Historias de usuario

3.1 Elementos de una Historias de Usuario

3.2 Ejemplos

3.3 Plantillas

4. Diagrama de actividades

4.1 Elementos de un diagrama de actividades

4.1.1 Actividad

4.1.2 Flujo entre actividades

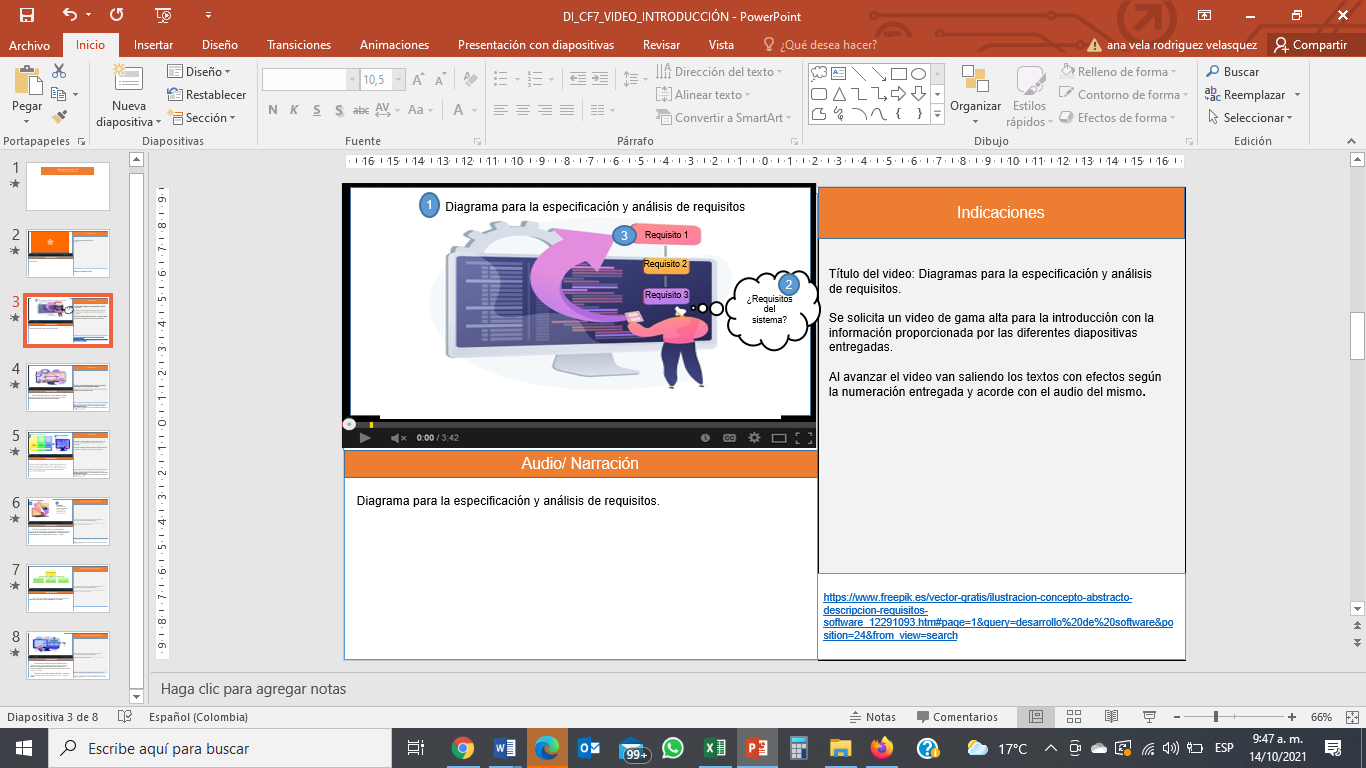
4.1.3 Nodo Inicial

4.1.4 Nodo Final

4.2 Ejemplo diagrama de actividades

1. **DESARROLLO DE CONTENIDOS:**

**Introducción**



**1. Características Lenguaje Unificado de Modelado (UML)**

Con el nacimiento de la Programación Orientada a Objetos (POO) se establecieron una variedad

de lenguajes, métodos, simbologías usadas para detallar el desarrollo de los diferentes sistemas. Indudablemente esta situación generó un desconcierto en los programadores de software; tratando de darle solución a este inconveniente aparece el Lenguaje de Modelado Unificado (*Unified Modeling Language –UML*)

El lenguaje de Modelado Unificado UML proporciona un grupo de herramientas que facilitan realizar el modelado (análisis y diseño) de los sistemas que son orientados a objetos. Indicando una serie de normas técnicas que representan de manera gráfica el camino y la estructura de un sistema determinado a los programadores, analistas, desarrolladores, diseñadores etc.

| El equipo compila un tablero de scrum para objetivos estratégicos Vector Premium | “El UML (Lenguaje Unificado de Modelado) es una de las herramientas más emocionantes en el mundo actual del desarrollo de sistemas. Esto se debe a que permite a los creadores de sistemas generar diseños que capturen sus ideas en una forma convencional y fácil de comprender para comunicarlas a otras personas”(Schmuller,2000, p.24). |
| --- | --- |

A continuación, se menciona la evolución del Lenguaje Unificado de Modelado (UML)

* 1. **Evolución del Lenguaje Unificado de Modelado UML**

Según Booch (2006):

El UML tiene su inicio a finales de los años 80, Inicialmente basado en el método Booch, desarrollado por el diseñador de Software Grady Booch, en el año de 1994 el científico de la computación James Rumbaugh le implementó el modelado de objetos, en 1995 se adicionaron a estas otros principios y otras metodologías, unas de estás adicionadas por Ivar Jacobson.

La primera versión oficial UML 1. Se lanzó en el año de 1997 con la unión y liderato de Booch, Rumbaugh y Jacobson, la cual ofrece una metodología que contiene varios estándares, donde su principal objetivo es que todos los diagramas tengan coherencia. En el año 2005 se divulgó la versión de UML 2.0 cuyo contenido estaba orientado principalmente a la visión Orientada a Objetos, de ahí en adelante se realizan nuevas versiones basadas en UML 2.0.

A continuación, se muestra un recurso de aprendizaje con la evolución mencionada.

| LÍNEA DE TIEMPO INTERACTIVA  DI\_CF7\_1.1.\_EvoluciónUML |
| --- |

**Fuente**: <https://cdn.visual-paradigm.com/guide/what-is-uml/01-uml-history.png>

**1.2 Características generales del Lenguaje Unificado de Modelado (UML)**

Como se mencionó anteriormente UML es un lenguaje capaz de brindar unas reglas que facilitan la comunicación de un sistema por medio de representaciones gráficas de este, indicando como hacer la creación y la lectura de los modelos. Aunque hay una gran variedad de características del UML según ( Booch, 2006) se pueden resumir en las que se nombran a continuación, por lo que le invito a revisar el siguiente recurso de aprendizaje:

Infografía estática

DI\_CF7\_1.2\_CaracterísticasGenerales.

En la siguiente figura se muestran las características básicas del UML, las cuales se mencionaron anteriormente.

*Características del UML*



Las características del *UML* se aplican en las fases de Desarrollo de Software, la primera fase es la de análisis y especificación de requisitos que se detalla a continuación.

**2. Especificación y análisis de requisitos**

La función más importante de la Especificación de Requisitos es servir de intermediario para que los clientes, los analistas de requisitos, desarrolladores y los usuarios puedan comunicarse. En esta es fundamental recolectar tanto los requerimientos de usuario y del cliente como los del software a desarrollar para lograr la satisfacción del cliente, por eso es importante el uso de técnicas que permitan representar de una manera visual e intuitiva estos requerimientos.

Cuando se realiza la recolección de requerimientos de un cliente, *UML* por medio de sus casos de uso permite el modelado de estos requerimientos. Los *stackholders* (actores externos) que están interesados en el producto son incluidos en el modelado con las funcionalidades requeridas (casos de uso) estos a su vez son modelados incluyendo relaciones y asociaciones jerárquicamente divididas entre ellos.

Quienes especifican los requerimientos de un cliente son los casos de uso y sus actores, qué es la expectativa que cada uno tiene del sistema, sin tener en cuenta la funcionalidad que se realizará. Los análisis de requisitos se realizan para diferentes procesos no necesariamente para desarrollos de software. Entre los diagramas más relevantes (y más utilizados) que se encuentran en UML están los diagramas de casos de uso, por lo que a continuación se realiza una definición más detallada de ellos.

**2.1 Diagrama de Casos de Uso**

Los casos de uso permiten la descripción de cada una de las funciones de las aplicaciones a desarrollar desde el punto de vista del usuario. Su utilización radica en expresar las funcionalidades que el producto de software debe efectuar y poder definir las personas que serán las encargadas de ejecutar estas funcionalidades. Es una técnica que consiste en hacer un diagrama con su respectiva notación. El caso de uso representa el comportamiento del software en la interacción con el usuario, para que este alcance un objetivo, por lo que, un caso de uso se compone de esos elementos que hacen parte de su notación.

**2.2 Componentes de los diagramas de casos de uso**

La notación de un diagrama de caso de uso se compone principalmente de 3 elementos que ayudan a representar simbólicamente las funcionalidades, personas involucradas y relación entre estas y los cuales son: los actores, las relaciones y los casos de uso. A continuación, se describe cada uno de ellos:

* + 1. **Actor**

Especifica un rol que ejecutará un usuario, representa a una persona o grupo de personas que interactúan con un software, se representa mediante un hombre de palo “*stick man*” con el nombre del actor (obligatorio). El uso de los actores se realiza para mostrar la clase de usuario que participará en el sistema el cuál será el encargado de realizar una función, por lo que los nombres de los actores normalmente inician con Mayúscula.

Existen varios tipos de actores según (Larman, 2002), los cuales se describen en la siguiente figura:



En la figura que se muestra a continuación, se observa cómo se representa de manera gráfica un actor en un UML, al cual también se le adiciona el nombre del actor correspondiente para dar mayor claridad en el proceso.

**Figura 1**

*Representación gráfica de un actor en UML*

Forma, Círculo

Descripción generada automáticamente

Nota. <https://bit.ly/3BK7Xzp>

Además, se debe tener en cuenta que no necesariamente los actores son personas, pese a que estos estén personificados por formas humanas en el diagrama de casos de uso. Por lo anterior, los sistemas externos también son actores, que pueden requerir alguna información del sistema real o tener alguna interacción con él como una base de datos, por ejemplo.

Al momento de determinar los casos de uso se hace necesario identificar el actor. Este es el nombre que se le da al usuario cuando desempeña ese papel, por lo que ellos llevan a cabo los casos de uso. Teniendo en cuenta lo anterior cada uno de ellos tiene la posibilidad de hacer diferentes casos de uso y viceversa. Para identificar un actor solo es necesario visualizar los roles que interactúan en el sistema.

**Ejemplo 1**

Para explicar cada uno de los elementos de un caso de uso se plantea el siguiente ejemplo de un requerimiento:

| “Una empresa solicita tener un sistema que permita hacer el proceso de generación de factura de venta a un comerciante, la cuál debe ser enviada por medio de un sistema externo que se encargará del proceso de la factura” |
| --- |

Los actores identificados en los requerimientos del ejemplo 1 son: comerciante y sistema externo, los cuales se muestran en la figura a continuación.

**Figura 2**

*Actores*







Nota.https://bit.ly/3k5reVJ

* + 1. **Relación**

Un actor interactúa con un caso de uso y está representado por una relación su símbolo es una línea o una flecha, la cual se muestra en la figura que se observa a continuación:

**Figura 3**

*Representación gráfica de una relación de caso de uso en UML*



A continuación, se describen diferentes tipos de relaciones según los actores mencionados anteriormente:

* La relación entre un actor y un caso de uso se llama **relación de comunicación**, esta se distingue por los actores pasivos o activos.
* El actor activo inicial o dispara la ejecución del caso de uso y la línea o flecha apunta al caso de uso.
* El actor pasivo no inicia el caso de uso y la flecha apunta al actor.
* El tipo de comunicación o la relación entre los casos de uso y los actores forman la asociación directa con el caso de uso.
* La manera de expresar es que el actor ejecuta el caso de uso.

Continuando con el ejemplo 1 se identifican las siguientes relaciones de comunicación:



* + 1. **Caso de Uso**

Los casos de uso se identifican a través de los verbos o acciones que el actor realiza o realizará, estas pueden describir, graficar y nombrar de la siguiente manera:

* Se representa mediante un óvalo e indica una función que el sistema debe proveer.
* El caso de uso debe ser de fácil lectura e interpretación tanto para los desarrolladores como para el cliente para el nombre se sugiere usar un verbo infinitivo con un complemento.

**Figura 4**

*Representación gráfica de un caso de uso en UML*



Los casos de uso que se identifican para el ejemplo 1 son: generar factura y enviar factura, los cuales se ven representados gráficamente en la siguiente figura.

**Figura 5**

*Ejemplos casos de uso*



Generar Factura

Enviar Factura

A continuación, se presenta un ejemplo de la representación gráfica del diagrama de caso de uso con los actores, relaciones y caso de uso del problema del Ejemplo 1 propuesto anteriormente.

**Problema**

| “Una empresa solicita tener un sistema que permita hacer el proceso de generación de factura de venta a un comerciante la cuál debe ser enviada por medio de un sistema externo que se encargará el proceso de la factura” |
| --- |

**Figura 6**

*Ejemplos diagrama de caso de uso venta* 



Por otro lado, los pasos para realizar el diseño del diagrama de casos de uso que se muestra en la figura anterior del sistema de venta se pueden ver en el siguiente recurso de aprendizaje:

| Infografía estática.  DI\_CF7\_2.2.3\_CasoUso |
| --- |

Además de la relación simple de comunicación que se muestra con anterioridad, existen más relaciones en los casos de uso que especifican más detalladamente las acciones entre estos y que se mencionan a continuación.

**2.3 Relaciones de casos de Uso**

En el proceso de desarrollo pueden definirse diferentes casos de uso y por lo que estos no son independientes, ya que se pueden establecer relaciones que dependan uno del otro entre los casos de uso.

Las principales relaciones consideradas por UML son:

* Extensión (extend)
* Generalización (generalization) e
* Inclusión (include)

Los tres tipos de relaciones que son soportados por el estándar UML se encuentran explicados en la siguiente tabla, por lo que se invita a revisar el contenido:

**Tabla 1**

*Relaciones de casos de Uso*

| Relación | Descripción | Notación |
| --- | --- | --- |
| Inclusión (include o use) | Se usa para extraer comportamientos comunes un caso de uso, puede incluir otro. |  |
| Extensión (Extend) | Se utiliza para extender a otro caso de uso, es útil para nuevos requerimientos del sistema. |  |
| Generalización | Relación entre un caso de uso general y uno específico que hereda propiedades del caso de uso base. |  |

Tomando el ejemplo anterior Ejemplo 1, y con el propósito de perfeccionar el diagrama de casos de uso se pueden agregar unos nuevos casos de uso donde se visualicen las relaciones mencionadas anteriormente, la primera de ellas es la relación extends:

**<<extends>>**

Se define esta relación cuando se cuenta con un caso de uso que es muy parecido a otro, pero que sin embargo realiza mucho más. Por ejemplo, en el caso anterior se puede ampliar el caso de uso envía factura por envía factura por correo electrónico. Ambos casos establecen la misma acción, la diferencia es que el caso de uso además de enviar la factura al sistema externo, también envía un correo electrónico con la factura y se representaría de la siguiente manera:

**Figura 7**

*Ejemplos diagrama de caso de uso venta con relación extends*





La relación se lee de la siguiente manera: el caso de uso enviar factura por correo extiende del caso de uso enviar factura.

Otro factor importante que se debe tener en cuenta es el siguiente:

**<< uses o include>>**

Este define la relación cuando una parte del comportamiento o acción es equivalente en más de un caso de uso y no se requiere duplicar la representación de su actuación. Por otro lado, para el Ejemplo 1, en el caso anterior se amplía con el siguiente requerimiento funcional: antes de que el usuario envié la factura puede realizar una búsqueda de facturas para seleccionarla y enviarla haciendo necesario incluir la relación <<include>>. Por lo que se vería así:

**Figura 8**

*Ejemplos diagrama de caso de uso venta con relación include*











| Para profundizar en relaciones de caso de uso puede visitar el enlace que se encuentra en la sección de material complementario con el siguiente nombre: diagrama de Casos de Uso II - 5 - Tutorial UML en español. |
| --- |

**2.4 Especificación de casos de uso**

Cuando se habla de especificación de casos de uso se hace referencia al proceso de descripción textual de cada uno de estos detallando los flujos y eventos que interactúan con el sistema y los actores que participan en él. Usualmente el analista, que es la persona encargada del diseño de los casos de uso a medida es quien va identificando los requerimientos y los va documentando.

En adición la especificación de Casos de Uso busca:

* Realizar la descripción de la interacción de un actor con el sistema
* Narrar el rol desempeñado por los actores en la interacción con el sistema y sus flujos
* Describir los casos de uso.

A continuación, se describe una plantilla donde se indican las partes y las indicaciones básicas para la documentación de casos de uso.

**Tabla 2**

*Plantilla de Casos de Uso*

| DESCRIPCIÓN DE UNA ESPECIFICACIÓN DE CASO DE USO | |
| --- | --- |
| Nombre del caso de uso | Indica el nombre del caso de uso. Normalmente, el nombre expresa el resultado objetivo y observable del caso de uso, como por ejemplo "Imprimir factura" en el caso de un de un sistema de ventas. |
| Autor | Persona que diseña el caso de uso. |
| Fecha | Aquí se diligencia la fecha en la que se realizó el caso de uso. |
| Descripción | Se realiza una corta descripción del caso de uso. |
| Actores | Describe los nombres del actor o actores del caso de uso. |
| Precondiciones | Son aquellas condiciones que deben existir para que se cumpla el caso de uso. |
| Flujo Normal | Son los pasos normales de ejecución del caso de uso. |
| Flujo Alternativo | Flujos alternativos de ejecución de casos de uso. |
| PosCondiciones | Son aquellas condiciones que se cumplen cuando finaliza un caso de uso. |

Nota. https://bit.ly/3k66Sf3

A continuación, se muestra un ejemplo de especificación de casos de uso para el caso de uso “**Crear Foro”** utilizando la plantilla propuesta posteriormente.

**Tabla 3**

*Ejemplo especificación de caso de uso foro*

| DESCRIPCIÓN DE UNA ESPECIFICACIÓN DE CASO DE USO | |
| --- | --- |
| Nombre del caso de uso | Crear foro |
| Autor | Mario Vera Cruz |
| Fecha | 20/09/2021 |
| Descripción | Crea un nuevo mensaje. |
| Actores | Alumno/Docente |
| Precondiciones | Al iniciar el usuario debe estar autenticado en el sistema. |
| Flujo normal | 1. El actor da clic en la opción nuevo mensaje.  2. Se muestra un área de texto para escribir el mensaje.  3. El actor introduce el mensaje.  4. El sistema verifica datos y los almacena.  5. El profesor acepta y publica. |
| Flujo alternativo | 4ª. Se hace una comprobación de los datos si no son correctos muestra mensaje de corrección.  4b. El profesor rechaza el mensaje de modo que no es publicado sino devuelto. |
| PosCondiciones | Se publica y almacena el mensaje satisfactoriamente. |

**2.5 Casos de uso reales (prototipos)**

Su objetivo principal es detallar el proceso de un sistema de información que se describe por medio de un caso de uso incluyendo la interacción de objetos, así como definir las transacciones de las interfaces y clases de los diferentes procedimientos secundarios de diseño. De este modo, cuando se reconozcan las clases que participan internamente de un caso de uso, es fundamental que los entornos que se recolectan del análisis se completen, incorporando las correspondientes clases de diseño sin dejar a un lado las restricciones del ambiente tecnológico.

Los casos de uso reales representan el diseño preciso del caso de uso desde una tecnología específica. Así pues, si se trata de un diseño gráfico para el usuario, se incluirá el diseño de ventanas de este y una descripción de la interacción con los objetos de la interfaz. En resumen, los casos de uso reales brindan diseños de pantalla y explicación de la interacción entre los objetos de esta.

Para aclarar las definiciones anteriores se emplea un ejemplo de un sistema de Ventas que encontrará posteriormente, donde se realiza el diseño del caso de uso real y se hace la descripción de interacción entre los artefactos del diseño, A continuación, se presenta una tabla con los datos de un ejemplo para un Sistema de Venta:

**Tabla 4**

*Sistema de ventas*

| Caso de uso adquirir productos | Adquirir productos |
| --- | --- |
| Actores | Cliente (quien inicia el proceso), Cajero. |
| Tipo | Primario, real. |
| Descripción | En el almacén de ropa el cliente se dirige a la caja registradora con los productos a comprar.  El empleado- cajero hace el registro de los productos que el cliente compra y recibe el dinero en efectivo.  Al finalizar el proceso, el cliente se retira con los artículos comprados. |

Para profundizar su aprendizaje le invito a ver el tema: Diseño del Caso de Uso a continuación:

**Figura 9**

*Ejemplos Caso de Uso reales prototipos*

| *Sistema de Ventas* |
| --- |
| *Cod Producto Cantidad*    *Precio Descripción*    *Total*    *Total Venta*  *Monto* |
|  |

Nota. h[ttps://docer.com.ar/doc/8nn115](https://docer.com.ar/doc/8nn115)

A continuación, se hace la descripción de las interacciones entre objetos en el diseño:

* + - * El inicio de este caso de uso se da cuando el cliente se acerca a la caja registradora con los artículos que desea llevar.
      * Para cada producto el cajero debe ingresar el **código del producto** en **A** de la **ventana-1,** si el usuario tiene más de un producto es opcional registrar la cantidad en **E,** una vez finalice el registro de todos los productos se presiona **H.**
      * El Sistema almacena los productos registrados. La visualización de la descripción y el precio del producto se muestra en **B** y en **F** de la **Ventana-1**
      * Para finalizar el cajero presiona el botón **I para** dar por finalizada la venta.
      * Luego el Sistema hace el cálculo y muestra en **C** el total de la venta.

Como se mencionó anteriormente, otra de las técnicas muy utilizadas para descripción de funcionalidades desde la visión del usuario son las historias de usuario las cuales se detallan a continuación.

**3. Historias de usuario**

Las historias de Usuario son una forma sencilla de representar los requisitos de un sistema de información, son creadas en una o dos oraciones y sobre todo escritas en un lenguaje muy común y legible para el usuario. Son llamadas *User Stories*, convirtiéndose en un estándar muy utilizado en el momento de la definición de requisitos. Estas surgen en XP (1999) y se exaltan concluyentemente con Mike Cohn (en “*User* *Stories Applied*: *For Agile Software* *Development*”, 2004) que establece el patrón para definirlas, siendo su simplicidad la razón de su popularidad, las cuales se muestran a continuación.

*Historia de Usuario*



 Nota. <https://bit.ly/3GMhWrB>

Como se muestra en la figura anterior las historias de usuario están compuestos de 3 elementos que se describen a continuación.

**3.1 Elementos de una historia de Usuario**

Una historia de usuario está compuesta por tres elementos los cuales son esenciales en el proceso:

* **Tarjeta** (*Card*)
* **Conversación** (*Conversation*) y
* **Confirmación** (*Confirmation*)

*Elementos de una Historia de Usuario*

****

A continuación, se hace la descripción de los elementos de una historia de usuario según Cohn, M. (2018):

* **Tarjeta (*Card****)*

Es una frase que sintetiza una necesidad dada en una conversación entre el usuario y el dueño del producto, escrita en una tarjeta, es comúnmente usada en el formato rol-objetivo- beneficio, como se muestro anteriormente, lo más importante es que son escritas teniendo en cuenta la visión el usuario.

Su estructura es:

Como [tipo de usuario] -> quién  
Quiero [necesidad] -> qué  
Para [beneficio esperado] -> para qué

Para ver como se mostraría esta información en la tarjeta se tiene el siguiente requerimiento:

Ejemplo: “Compra una Boleta de cine”:

**Como:** Cliente

**Quiero:** comprar una boleta

**Para:** entrar a Cine

Como se puede observar en el ejemplo con esta simple estructura compuesta por tres líneas:

* Se sabe **quién** (permite identificar los roles de valor),
* El ***que*** *(permite identificar el problema que se resuelve) y*
* **Para qué** (lo que se espera lograr con la construcción de la historia de usuario).
* **Conversación**

Es el proceso en el que se comunica las personas que requieren las necesidades (dueño del producto) con las personas que saben cómo solucionarlas (desarrolladores) esto se hace de la siguiente forma:

* Por medio de preguntas
* Utilizando recursos como: gráficas, prototipos etc.

Lo anterior busca tener claridad y concretar el contenido de las historias de usuario.

Siguiendo con el ejemplo de “Compra una Boleta de cine”: se realizarán las siguientes preguntas:

* ¿Existe un máximo de boletas para la venta?
* Si, solo 3 por cada usuario cuando estén en promoción.
* ¿Qué medios existen para realizar el pago de las boletas?
* Efectivo y Tarjeta ….

Como se mencionó anteriormente la conversación se hace con el fin de entender el problema que se quiere resolver.

* **Confirmación**

Es la verificación del acuerdo con relación a lo que se quiere construir, permite validar si se terminó o no, o si se cumplió con lo esperado. Esta confirmación recibe el nombre de criterios de aceptación.Las funciones que cumplen los criterios de aceptación es tener claro el contexto donde se efectúa la historia de usuario y conocer fácilmente, si una historia realmente está terminada.

Continuando con el ejemplo “Compra una Boleta de cine” los criterios de aceptación que se pueden establecer son:

Ejemplo: “Compra una Boleta de cine”

Criterios Aceptación:

* Cuando se termine la boleta se debe mostrar un mensaje informando que ya no hay disponibilidad para la venta.
* El sistema debe validar la edad para la venta de la boleta según clasificación de la película.
* El sistema debe mostrar el valor de devolución después del pago en efectivo, en el caso de que sobre si no mostrar cero…

Cuando se realizan historias de usuario es muy importante cumplir con las siguientes reglas:

* Ser concisa
* Concreta y
* Realizar criterios de aceptación claros y específicos.

Así como son necesarias estas reglas, también es fundamental una buena redacción de las historias de usuario, por esto se presenta a continuación un método que facilita afianzar la calidad en la escritura de una historia de usuario: el método “Invest”.

Según (Bill Wake 2003): Una buena guía para redactar Historias de Usuario con calidad es seguir el acrónimo **INVEST,** como se muestra a continuación:

**Figura 11**

*Método Invest*

**

Nota. https://bit.ly/3BV4ZIt

Para profundizar en el tema se hace la descripción del método Invest, por lo que se le invita a revisar el siguiente recurso de aprendizaje:

| Slider  DI\_CF7\_3.1\_ElementosUsuario |
| --- |

Utilizando este método se logra asegurar que una historia de usuario no sea dependiente de otra y así poder proporcionar que su planificación y desarrollo no incluya muchos datos técnicos que restrinjan los convenios entre los clientes y los desarrolladores.

**3.2 Ejemplos**

A continuación, se muestran algunos ejemplos prácticos de historias de usuario aplicando el método invest:

Historia de usuario: **“Crear Usuario”**

| Primera Opción |
| --- |

**Como**: cliente de un Supermercado

**Quiero:** hacer compras por medio de una aplicación web

**Para:** no moverme de casa.

Como se muestra en la primera opción de historia de usuario, se identifica el que quiere hacer, pero no el cómo lo va a realizar, no hay ningún detalle del ingreso a las compras en línea que seguramente se aclara en una siguiente reunión.

Esta historia de usuario cumple con las características del método *invest* por lo que es sencilla, clara y satisface una necesidad, es específica lo cual lleva a tener una fácil estimación definiendo el tiempo de desarrollo.

| Segunda Opción |
| --- |

:

**Como**: cliente de un Supermercado

**Quiero acceder** al registro por medio de un formulario que me permita escribir mi correo electrónico y asignar una contraseña.

**Para**: tener seguridad

Si se observa esta historia de usuario se puede notar que hay demasiadas especificaciones en la estructura Quiero, además que no hay claridad en el beneficio.

Un ejemplo más de historias de usuario que no cumplen con las cualidades de Independiente, Estimable y pequeña es el siguiente:

Historia de usuario: **“Crear Aplicación Web”**

**Como**: cliente de un Supermercado

**Quiero**: ingresar a una página web

**Para:** hacer compras en línea.

Esta historia no es independiente ya que describe una historia de usuario muy grande la cual es denominada como **Épicas,** ya que de ella pueden salir muchas historias de usuario como:

* Hacer un registro
* Un ingreso
* Seleccionar productos o
* Pagar, etc.

| Para visualizar una historia del usuario Épica se le invita a ver el siguiente recurso de aprendizaje denominado: ¨Plantillas-Historias¨ | PDF |
| --- | --- |

Por lo anterior, no se cumpliría tampoco con la característica de estimación porque no se sabría cuándo se terminaría la historia. Como se puede notar en la descripción de los anteriores ejemplos el método Invest nos ofrece reglas para que la redacción de las historias de usuario sea óptimas y así lograr cumplir al cliente con lo que realmente necesita.

Para realizar una buena gestión de proyectos, usualmente en las metodologías ágiles se usan una serie de herramientas para crear las historias de usuario como:

* Trello
* Jira
* Taiga o
* Se diseñan o descargan plantillas en Excel.

A continuación, se muestra un ejemplo de uso de una plantilla que mantiene la misma estructura sencilla de las historias de Usuario.

**3.3 Plantillas de Historias de Usuario**

Las plantillas de historias de usuario pueden ser adaptadas según las necesidades requeridas sin perder la flexibilidad y sencillez que ofrece una HU, los equipos de desarrollo trabajan conjuntamente utilizando herramientas en línea para compartir y trabajar en las plantillas. A continuación, se puede visualizar un ejemplo del uso de una plantilla de Excel descargada de la web con un ejemplo de gestión de productos.

**Figura 12***.*

*Ejemplo plantilla historia de usuario*

| ID de la historia | Rol | Funcionalidad | Razón/Resultado | Criterio de aceptación |
| --- | --- | --- | --- | --- |
| Id.01 | Como cliente | Quiero ver una lista de categorías de productos. | Para poder realizar búsquedas de productos por categorías. | Despliegue de listado de categorías a seleccionar con productos asociados. |
| Id.01 | Como cliente | Quiero ver una lista de precios. | Para poder elegir el producto más barato. | Despliegue una lista ordenada de productos por precio de menor a mayor. |

Nota. https://bit.ly/3CJqa13

Ya se han descrito dos de las tres técnicas más utilizadas para la descripción de requerimientos que son los casos de uso y las historias de usuario, a continuación, se hace la descripción de la tercera técnica: los diagramas de actividades.

**4. Diagrama de actividades**

Estos son diagramas de comportamiento que se utilizan para representar una sucesión de actividades, explican el flujo de operaciones desde el punto en que inician hasta el punto final definiendo una variedad de caminos de decisiones en el desarrollo de eventos que abarca una actividad. (Booch, G. (1994). Ellos permiten visualizar un caso de uso específico a un nivel más preciso, ilustrando el flujo de actividades definidas en un sistema.

A continuación, podrá conocer los beneficios del diagrama de actividades.

Beneficios:

* Permite explicar la lógica de un algoritmo.
* Representa cada uno de los procesos realizados en los diseños de casos de uso en UML.
* Permite instruir procesos entre el sistema y los usuarios en un flujo de trabajo o negocios.
* Permite dar claridad optimizando y facilitando los casos de uso complejos.

**4.1 Elementos de un diagrama de actividades**

Un diagrama de actividades contiene fundamentalmente los siguientes elementos:

* Actividad
* Flujo entre actividades
* Nodo inicial y
* Nodo final.

**4.1.1 Actividad**

Una actividad es la descripción o el detalle de una sucesión de conductas parametrizadas, se simboliza como un flujo de acciones ordenadas. El modelado de este flujo se hace por medio de nodos de actividad que se conectan por medio de flujos de control. Las actividades generan jerarquías de peticiones llamando a diferentes actividades o como último recurso solventando operaciones propias.

La representación gráfica de las actividades se hace por medio de un rectángulo que tiene bordes redondeados y dentro del rectángulo especifica el nombre de la actividad.

**Figura 13**

*Representación gráfica de una actividad*



Para nombrar las actividades se usan verbos del modelo de negocio como, por ejemplo:

* Buscar Factura
* Realizar pago y
* Renovar lista, etc.

**4.1.2 Flujo entre actividades**

Este permite realizar los enlaces entre objetos y nodos de una actividad, estos agregan flujos de objetos y flujos de control. Su representación gráfica se hace por medio de una flecha con punta abierta que representa como es el orden en la ejecución de las actividades, muchas veces se adiciona una descripción en la flecha para una mejor comprensión.

**Figura 14***.*

*Representación gráfica de un flujo de actividades*



Nota. https://bit.ly/3nZJwcp

Cuando los flujos de actividades utilizan condiciones para su acción se representan mediante un rombo y son llamados nodos de decisión

**4.1.3 Nodo Inicial**

Como su nombre lo dice el nodo inicial se encarga de empezar un flujo, cuando se llama a una actividad, por cada diagrama existe un solo nodo inicial, el cual se representa con un círculo pequeño.

**Figura 15**

*Representación gráfica de un nodo inicial*







Nota. <https://bit.ly/3wc86du>

**4.1.4 Nodo Final**

También es un nodo de control final. Su función es detener todos los flujos de una actividad, en una actividad pueden existir más de un nodo final. Este se representa con un círculo sólido con un hueco dentro como se puede observar a continuación:

**Figura 16**

*Representación gráfica de un nodo final*





Nota. https://bit.ly/3bJLM1m

| Como parte de su aprendizaje le invito a ver el video denominado: Diagrama de Actividades - 15 - Tutorial UML en español, que se encuentra en la sección de material complementario. |
| --- |

Después de realizada la descripción de los elementos que componen un diagrama de actividades se presenta el siguiente ejemplo de un diagrama de clase sobre un proceso de registro e inicio de sesión.

**4.2 Ejemplo diagrama de actividades**

A continuación, se realiza un ejemplo de un diagrama de actividades que tiene las funcionalidades de iniciar sesión y registrar en un sistema, por lo que se le invita a revisar el siguiente recurso de aprendizaje.

| DI\_CF7\_4.2\_EjemploDiagramasA |
| --- |

Como se muestra en el ejemplo anterior, el flujo inicia con la actividad autenticar usuario y pasa por un nodo de decisión que verifica si el usuario existe no o no, si el usuario existe pasa a un nodo de decisión donde valida el usuario y si no es correcto, muestra un mensaje de error, si es correcto da acceso a la aplicación y muestra el menú inicio.

En caso de no existir el usuario pasa a la actividad de registrar usuario y después ingresa datos, luego sigue a un nuevo nodo de decisión, donde valida datos y da acceso a la aplicación por lo que sigue el flujo de actividad mostrando la interfaz del menú del sistema y finalmente termina la actividad con un nodo final.

1. SÍNTESIS

Con el diagrama de actividades se termina la descripción de las tres técnicas usadas para la especificación y el análisis de requisitos, como se pudo analizar cada una de las técnicas nos brinda características para lograr representar gráficamente y de una forma clara, los requisitos que un usuario necesita poder cumplir satisfactoriamente con las solicitudes de los clientes y usuarios, por otro lado se pueden usar todas las técnicas en una especificación y análisis de requisitos si se ve la necesidad.

A continuación, se muestra un mapa conceptual con los elementos más importantes desarrollados en este componente.

**Figura 17**

*Mapa Conceptual*















































1. **ACTIVIDADES DIDÁCTICAS**

| DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA | |
| --- | --- |
| Nombre de la actividad | Diagramas para la especificación de requerimientos |
| Objetivo de la actividad | Relacionar los conceptos más importantes asociados al proceso de especificación de requisitos con historias de usuario y casos de uso. |
| Tipo de actividad sugerida | Arrastrar y soltar palabras o imágenes. |
| Archivo de la actividad  (Anexo donde se describe la actividad propuesta) | Anexo documento en Word llamado Actividad didáctica 1. |

1. **MATERIAL COMPLEMENTARIO:**

| Tema | Referencia APA del Material | Tipo de material  (Video, capítulo de libro, artículo, otro) | Enlace del recurso o  archivo del documento o material |
| --- | --- | --- | --- |
| 2.3 Ejemplos Include, extends | Nicosiored. (2017, November 22). *Diagrama de Casos de Uso II - 5 - Tutorial UML en español*. [Video]. YouTube. <https://www.youtube.com/watch?v=DUjBnEvIm1M&list=PLM-p96nOrGcaw5dhv8wOA5tVVWEmXtA2F&index=5> | Video | <https://www.youtube.com/watch?v=DUjBnEvIm1M&list=PLM-p96nOrGcaw5dhv8wOA5tVVWEmXtA2F&index=5> |
| 4. Diagrama de Actividades | Nicosiored. (2018, January 31). *Diagrama de Actividades - 15 - Tutorial UML en español.* [Video]. YouTube. <https://www.youtube.com/watch?v=GoYdpOVhDRc&list=PLM-p96nOrGcaw5dhv8wOA5tVVWEmXtA2F&index=15> | Video | [*https://www.youtube.com/watch?v=GoYdpOVhDRc&list=PLM-p96nOrGcaw5dhv8wOA5tVVWEmXtA2F&index=15*](https://www.youtube.com/watch?v=GoYdpOVhDRc&list=PLM-p96nOrGcaw5dhv8wOA5tVVWEmXtA2F&index=15) |

1. **GLOSARIO:**

| TÉRMINO | SIGNIFICADO |
| --- | --- |
| 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. |
| 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. |
| Requerimiento | Se refiere a la petición que se hace de algo que se solicita. |
| Requisito | Es la condición que debe cumplir algo, en general este 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. |

1. **REFERENCIAS BIBLIOGRÁFICAS:**

Booch, G., Rumbaugh, J., Jacobson, I., Molina, J. J. G., & Martínez, J. S. (2006). *El lenguaje unificado de modelado: guía del usuario*. Addison-Wesley.

Booch, G. (1994). Object-oriented analysis and design. Redwood City.

Cohn, M. (2018). User Stories and User Story Examples by Mike Cohn. [online] Mountain Goat Software. <https://www.mountaingoatsoftware.com/agile/user-stories>

Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.

Gutiérrez, J. (s.f.). Diagramas UML de casos de uso y de requisitos. <http://www.lsi.us.es/~javierj/cursos_ficheros/metricaUML/CasosUsoUML.pdf>

Larman, 2002 Larman, C. “Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process”. 2nd Ed. Prentice Hall, 2002

Norris & Rigby. “*Ingeniería de software explicada*”, 1 edición Editorial Megabyte-Noriega editores, México,1994.

Schmuller, J., & Garza Marín, A. D. (2000). *Aprendiendo UML en 24 horas*.

Wake, W.C. 2003 “[Extreme Programming Explored](https://www.amazon.es/Extreme-Programming-Explored-William-Wake/dp/0201733978/ref=sr_1_1?s=foreign-books&ie=UTF8&qid=1553761214&sr=1-1&keywords=Extreme+Programming+Explored)” y “[Refactoring Workbook](https://www.amazon.es/Refactoring-Workbook-Addison-wesley-Object-Technology/dp/0321109295/ref=sr_1_1?s=foreign-books&ie=UTF8&qid=1553761292&sr=1-1&keywords=refactoring+workbook)”

1. **CONTROL DEL DOCUMENTO**

|  | Nombre | Cargo | Dependencia | Fecha |
| --- | --- | --- | --- | --- |
| Autor (es) | Zulema Yidney León Escobar | Experto temático | Regional Cauca, Centro de teleinformática y producción industrial. | Septiembre 2021 |
| Jonathan Guerrero Astaiza | Experto temático | Regional Cauca, Centro de teleinformática y producción industrial. | Septiembre 2021 |
| Ana  Vela Rodríguez Velásquez | Diseñadora  Instruccional | Centro de Gestión Industrial  Regional Bogotá Capital. | Septiembre 2021 |
| Carolina Coca Salazar | Revisora Metodológica y Pedagógica | Regional Distrito Capital- Centro de Diseño y Metrología. | Octubre 2021 |
| Rafael Neftalí Lizcano Reyes | Asesor pedagógico | Regional Santander. Centro Industrial del Diseño y la Manufactura. | Septiembre 2021 |
| Sandra Patricia Hoyos Sepúlveda | Revisión y corrección de estilo | Centro para la Industria de la Comunicación Gráfica - Distrito capital | noviembre de 2021 |

1. **CONTROL DE CAMBIOS**

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

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