



Programas  
Académicos



# **Proyecto Final. – Modelado UML**

## **Ingeniería en Desarrollo de Software**

**Tutor: Eduardo Israel Castillo García**

**Alumno: José Manuel Ramos Vega**

**Fecha: 07 de enero de 2024**



## Índice

Introducción.....	3
Descripción .....	4
Justificación.....	5
Desarrollo .....	6
Diagrama de Actividades.....	6
Diagrama de Componentes.....	7
Conclusión .....	8
Referencias.....	9

## 1- Introducción

El UML provee beneficios significativos para los ingenieros de software y las organizaciones al ayudarles a construir modelos rigurosos, trazables y mantenibles, que soporten el ciclo de vida de desarrollo de software completo.

El Lenguaje Unificado de Modelado (UML) es, tal como su nombre lo indica, un lenguaje de modelado y no un método o un proceso. El UML está compuesto por una notación muy específica y por las reglas semánticas relacionadas para la construcción de sistemas de software.

El UML en sí mismo no prescribe ni aconseja cómo usar esta notación en el proceso de desarrollo o como parte de una metodología de diseño orientada a objetos. El UML soporta un conjunto rico en elementos de notación gráficos. Describe la notación para clases, componentes, nodos, actividades, flujos de trabajo, casos de uso, objetos, estados y cómo modelar la relación entre esos elementos. El UML también soporta la idea de extensiones personalizadas a través elementos estereotipados.

## 2- Descripción

Como se ha revisado en las actividades anteriores modelar los diagramas nos permite identificar elementos que a simple vista no son vistos y se encuentra el vínculo entre los mismos, por tal a continuación se pretende reconocer la importancia de los diagramas de actividad y componentes los cuales muestran cómo los eventos de un caso de uso se relacionan o como se pueden coordinar para el flujo de un sistema y sus funcionalidades.

Diseñar los diagramas de Actividades y Componentes con ayuda de una herramienta de modelado libre o privativa acorde a las actividades anteriores.

Modelar el proceso de negocio es una parte esencial de cualquier proceso de desarrollo de software. Permite al analista capturar el esquema general y los procedimientos que gobiernan el negocio. Este modelo provee una descripción de dónde se va a ajustar el sistema de software considerado dentro de la estructura organizacional y de las actividades habituales.

### 3- Justificación

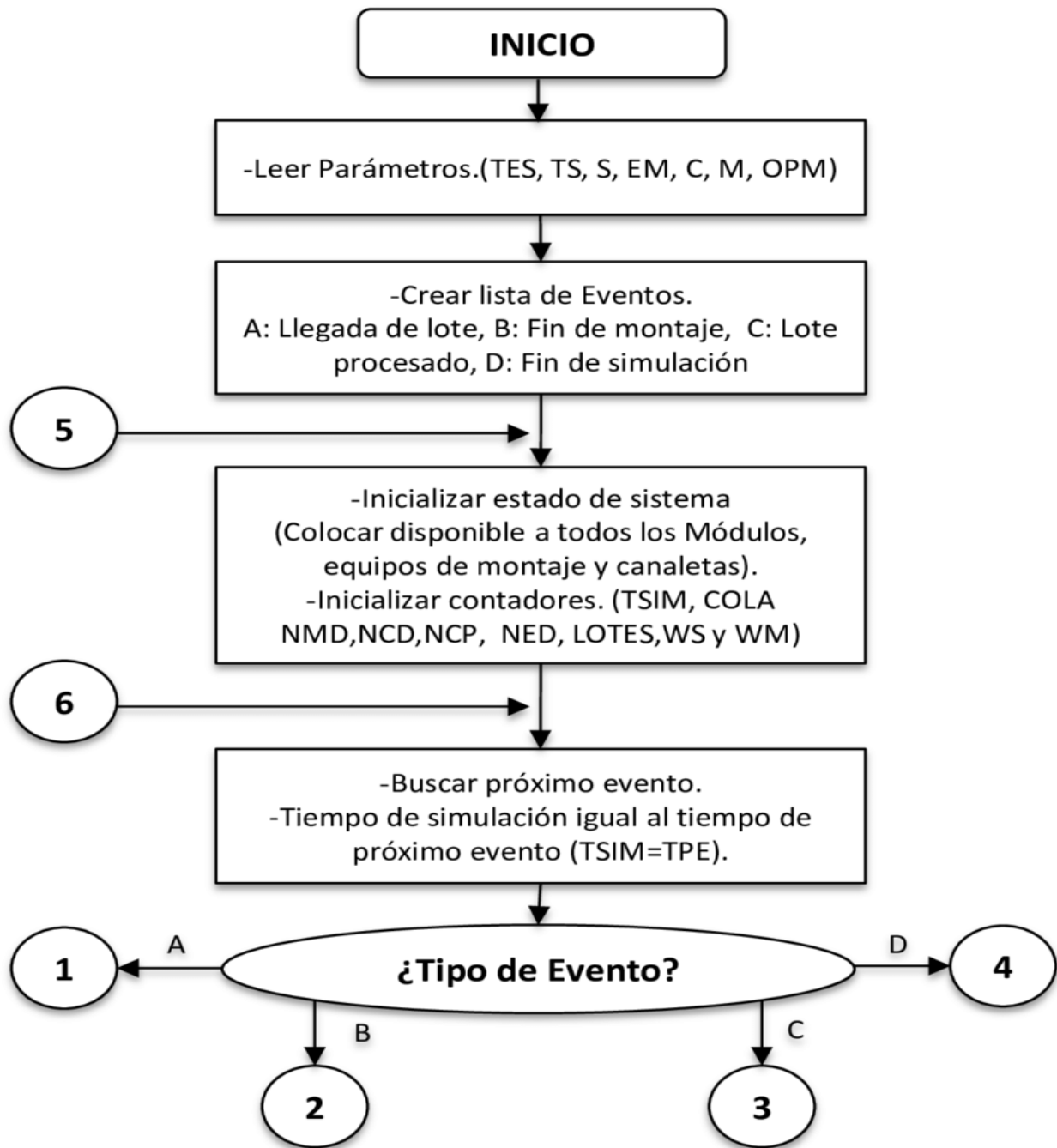
Como un modelo preliminar del negocio, permite al analista capturar los eventos, las entradas, los recursos y las salidas más importantes vinculadas con el proceso de negocio. Es posible construir un modelo completamente trazable mediante la posterior conexión de elementos de diseño (tales como los casos de uso) al modelo de negocio a través de conectores de implementación, desde la generalidad del proceso de negocio a los requisitos funcionales y eventualmente a los artefactos de software que se construirán realmente. Por el hecho de que el modelo de procesos de negocio normalmente es más amplio que la parte de sistema computacional considerada, también permite al analista identificar claramente qué está dentro del alcance del sistema propuesto y qué se implementará de otras formas (por ejemplo: un proceso manual).

Un proceso de negocio es una colección de actividades diseñadas para producir una salida específica para un cliente o un mercado en particular. Esto implica un fuerte énfasis en cómo se realiza el trabajo dentro de una organización, en contraposición con un enfoque del producto en qué se produce.

#### 4- Desarrollo

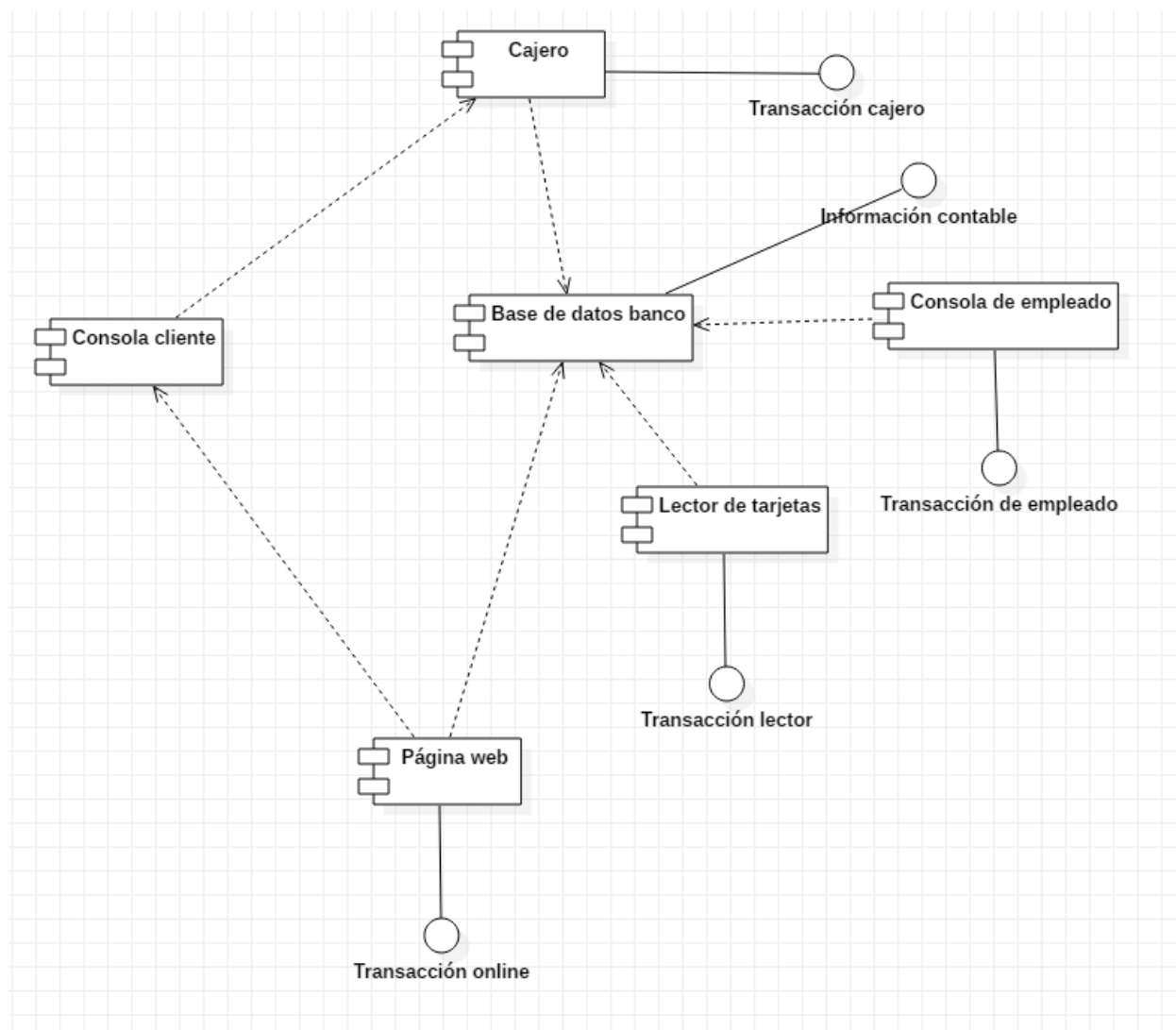
##### -Diagrama de Actividades

En este diagrama describo la actividad de acuerdo al escenario



## -Diagrama de Componentes

En este diagrama el componente de la transacción requerida.



## 5- Conclusión

En conclusión, los modelados permiten analizar y desarrollar software más complejos y eficaces, permitiendo obtener un modelo visual del sistema.

El lenguaje unificado de modelado es de gran importancia en la ingeniería, ayuda a modelar grandes sistemas y es muy utilizado en distintos tipos de software. El UML cada vez es más implementado, ya que ayuda a obtener un mayor entendimiento sobre las funcionalidades del software tanto para el desarrollador y para el cliente. Este modelado no solo permite diseñar, graficar, etc; sino que también facilita la construcción de los sistemas, a través de los diferentes modelos de estructurales y de comportamientos. Muchas veces para desarrollar software complejo es muy difícil describirlo a través de texto, pero si lo expresamos a través de diagrama es más fácil entenderlo. Es por esto que UML presenta tres beneficios: este modelo Estos diagramas son utilizados según la necesidad y el tipo de sistema que se vaya a desarrollar.



## **6- Referencias**

Sinan Si Alhir, UML in a NutShel. ISBN: 1-56592-448-7. Publisher: O'Reilly & Associates, Inc

Doug Rosenberg with Kendall Scott, Use Case Driven Object Modeling with UML. ISBN:0-201-43289-7. Publisher: Addison-Wesley