

## Actividad | 3 | Modelado de UML

### Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Miguel Ángel Rodríguez Vega.

ALUMNO: Uziel de Jesús López Ornelas.

FECHA: 12 de Enero del 2025.

**Índice**

Introducción ..... 1

Descripción..... 1

Justificación..... 2

Desarrollo ..... 3

Diagrama de Actividades ..... 3

Diagrama de Componentes ..... 4

Conclusión..... 4

Referencias..... 6

## Introducción

Para nuestra última actividad que es el proyecto final elaboraremos un diagrama de actividad y componentes en los que elaboraremos la relación que tienen estos dos con la actividad que hemos estado realizando desde el comienzo, para ello es importante explicar de manera sencilla que es un diagrama de actividad y así comprender mejor lo que se quiere transmitir. Un diagrama de Actividad es un caso especial de un diagrama de estados, solo que, en este, todos los estados son actividades que se tienen que desarrollar, muestra acciones o tareas que se ejecutan en cierto orden (y otros elementos adicionales), cubre la vista dinámica de un sistema y es el equivalente en OO a los diagramas de flujo y DFD. Estos se emplean para especificar una operación compleja, un proceso de trabajo(workflow), el proceso de negocio asociada a un caso de uso. A continuación, describiremos en los siguientes apartados las características principales de cada diagrama.

## Descripción

Los Diagramas de Actividades expresan conjuntos de actividades, ¿Qué hacen las actividades?, ¿En qué orden se ejecutan?, ¿Dónde ocurren?, ¿Quién las ejecuta?, ¿Qué insumos requieren?, ¿Qué productos generan? Las actividades se enlazan por transiciones automáticas. Cuando una actividad termina, se desencadena el paso siguiente.

Los **Elementos** de los diagramas de Actividades se dividen en los siguientes:

- **Estados de Acción:** Acciones no interrumpidas de los objetos.
- **Flujo de Acción:** Flechas que ilustran las relaciones entre los estados de acción.
- **Flujo de objetos:** La creación y modificación de objetos por parte de las actividades.

Una flecha desde una acción a un objeto significa que la acción está creando o influyendo sobre dicho objeto. Una flecha desde un objeto a una acción significa que el estado de acción utiliza dicho objeto.

El **Estado inicial** se trata del primer estado de un estado de acción, el **Estado final** se trata del último estado de un estado de acción, la **Ramificación** es un rombo que representa una decisión con caminos alternativos. Las salidas alternativas deben de estar etiquetadas con una condición, la **Sincronización** ayuda a ilustrar la ocurrencia de transiciones paralelas, es decir, las acciones

concurrentes, los **Marcos de Responsabilidad** agrupan las actividades relacionadas en una misma columna.

### **Justificación**

Es importante mencionar las condiciones para crear un diagrama de actividad, para ello se empieza preguntándose qué pasaría en primer lugar, que pasa en segundo lugar, y así sucesivamente, se debe determinar si las actividades se realizan en secuencia o en paralelo, si se ha creado diagramas de flujo de datos físicos, se podrían usar para determinar la secuencia de actividades. Un diagrama de componentes UML es una representación visual de la estructura interna de un sistema de software. Este diagrama muestra los componentes físicos y lógicos que componen el sistema, así como las relaciones y las interacciones entre ellos. Los componentes pueden ser archivos, bibliotecas, clases, módulos o cualquier otra entidad lógica o física que juegue un papel importante en el sistema. Este diagrama es fundamental para la arquitectura de software y proporciona una visión general clara y concisa de cómo se organiza un sistema y cómo interactúan sus componentes.

La **Comunicación Efectiva** actúa como un medio de comunicación efectivo entre el equipo de desarrollo y los stakeholders.

El **Diseño estructurado** el diagrama de componentes permite a los equipos de desarrollo organizar y estructurar sus sistemas de manera efectiva.

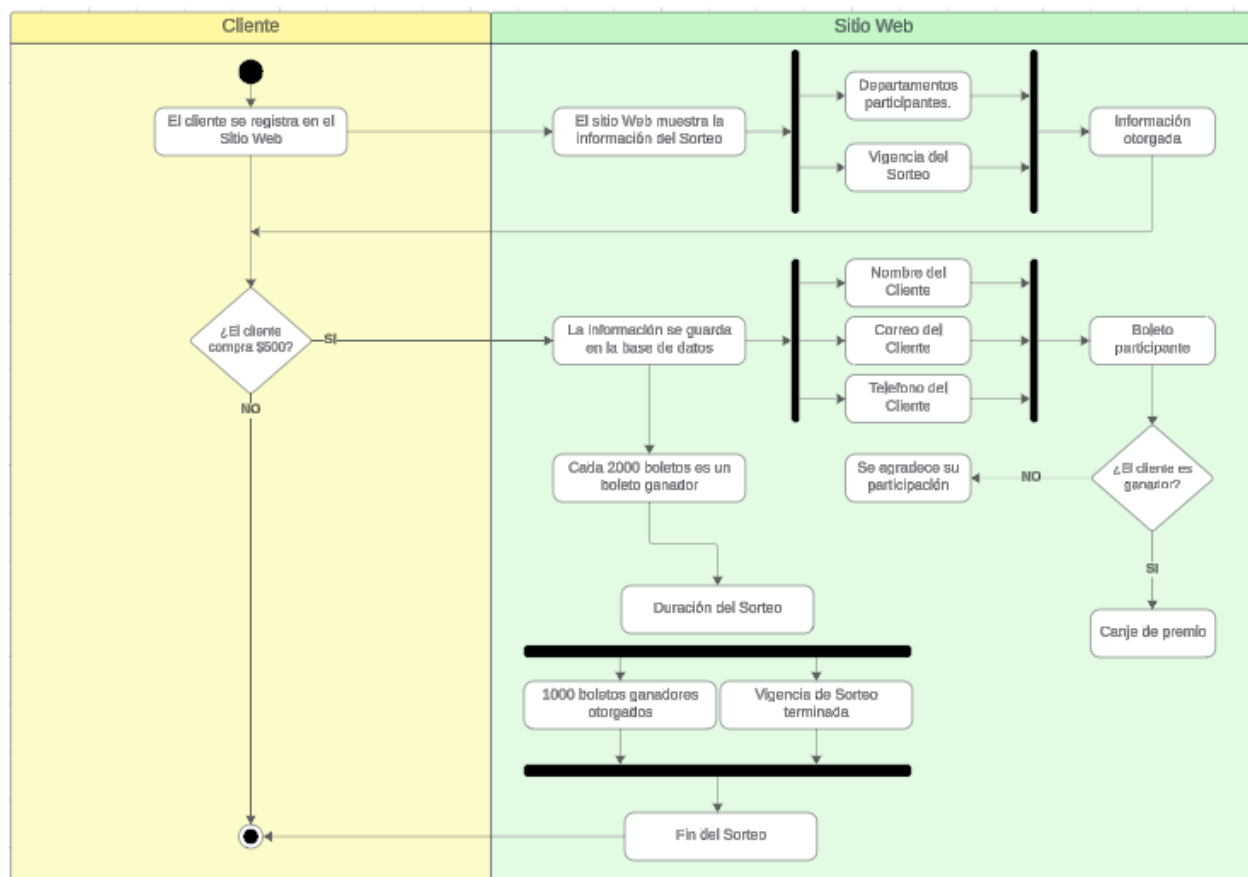
La **Reutilización de componentes** es posible identificar que componentes son reutilizados en otros proyectos.

La **Identificación de dependencias** permite identificar las dependencias entre los diferentes componentes del sistema.

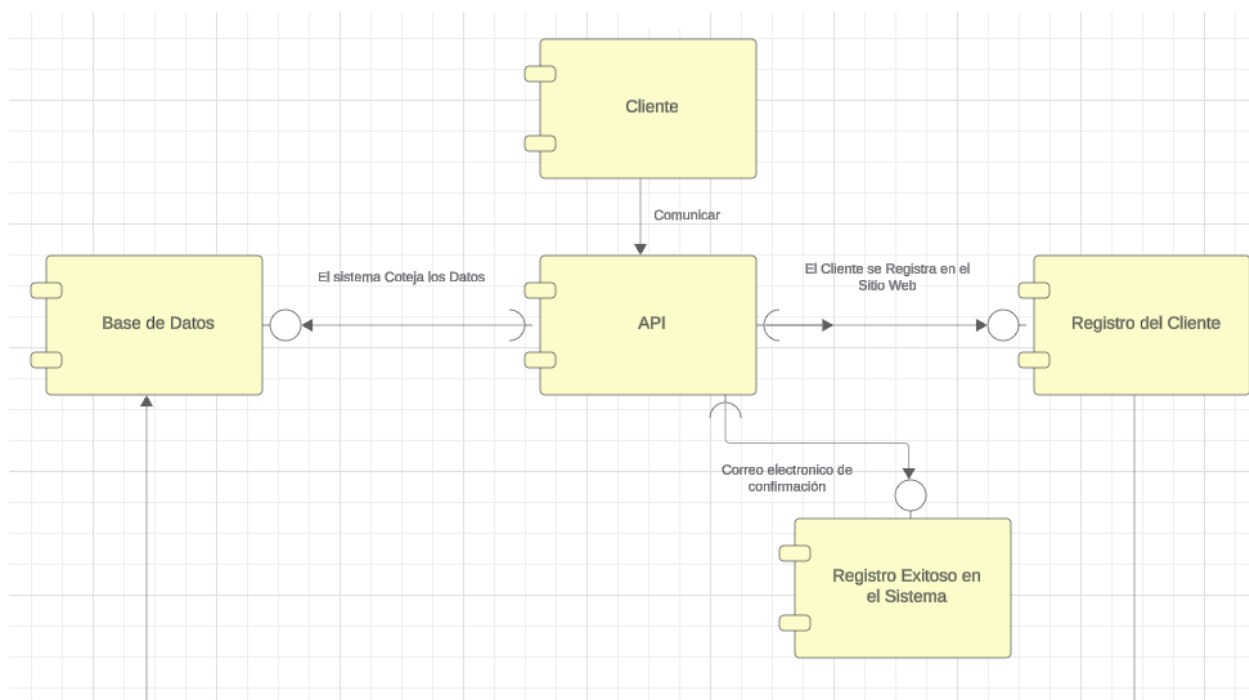
**Facilita el mantenimiento y la evolución** al proporcionar una vista clara de la estructura clara del sistema, facilita la tarea de realizar modificaciones o mejoras en el sistema a lo largo del tiempo.

## Desarrollo

## Diagramas de Actividades:



### Diagrama de Componentes:



### Conclusión

Los diagramas de componentes son una herramienta fundamental para el diseño de y el desarrollo en la arquitectura de Software ya que nos proporcionan ventajas que son necesarias para la representación gráfica de algún proyecto o actividad que se nos presente, la manera en la que se muestra de manera visual los componentes y sus diferentes conexiones nos ayudan a entender con mayor eficiencia el resultado o la ejecución que se plantean en un inicio brindando importancias notorias en su uso, por ejemplo:

- Nos ayuda a mejorar la comprensión que tenemos de nuestro sistema, esto es importante ya que nos da una visión clara de lo que se tiene que realizar con los diferentes elementos que se tienen en la mano y así tomar una acción de acuerdo a lo que es mejor para el proyecto.
- Ayuda a la planificación y el diseño, los diagramas de componentes no son los únicos en su tipo que nos proporcionan una amplia variedad de herramientas y soluciones a la hora de ejecutar o resolver un problema ya que cada diagrama tiene sus características que nos brindan exactitud y precisión.

**Link para GitHub:**

<https://github.com/Leyzu-Ing/Lenguaje-Unificado-de-Modelado.git>

## Referencias

Equipo editorial de IONOS. (2021, 26 mayo). *Diagrama de componentes: modelado eficiente de sistemas con módulos de software*. IONOS Digital Guide.

<https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/diagrama-de-componentes/>