INSTITUTO POLITÉCNICO NACIONAL
SECRETARIA ACADÉMICA
DIRECCIÓN DE EDUCACION MEDIA SUPERIOR
CENTRO DE ESTUDIOS CIENTÍFICOS Y TECNOLÓGICOS No. 13
"RICARDO FLORES MAGÓN"

G U Í A

d e e s t u d i o p a r a

presentar ETS de la

UNIDAD DE APRENDIZAJE

MODELADO DE SISTEMAS

Semestre 2023-2

TURNO VESPERTINO

Integrantes de la academia: Alfredo Campos Guerrero

Fecha de Elaboración: 24/05/2023

FORMATO DE LA GUÍA DE ESTUDIO

Área:	Nombre de la Unidad de Aprendizaje:	Nivel/semestre:
Tecnológica	**Modelado de Sistemas**	**5**

Instrucciones generales de la guía:

- Esta guía no tiene ningún valor sobre la calificación final

- Debes tener instalado el siguiente software:
  - **Star UML ó Visio**
    - Para la elaboración de los diagramas UML en el examen práctico.
    - Hay que recordar que este examen es teórico práctico.
    - El examen práctico abarca los diagramas UML correspondientes a:
    - **Diagramas de Caso**
    - **Diagramas de Clases**

Presentación:

La presente guía de estudio permitirá al estudiante el desarrollo de competencia en la construcción de modelos de un sistema a través del uso del lenguaje unificado de modelado (UML).

Objetivos

Conocer los diferentes tipos de diagramas basados en el lenguaje unificado de modelado:

- Diagrama de clase
- Diagrama de objetos
- Diagrama de caso de uso
- Diagrama de estados
- Diagramas de secuencia
- Diagrama de colaboraciones
- Diagrama de actividades
- Diagrama de componentes

Justificación

El uso del lenguaje unificado de modelado durante el análisis del desarrollo de Sistemas Orientados a Objetos es útil porque nos permite visualizar los requerimientos del usuario así como los recursos que necesitamos para el desarrollo e implementación del sistema.

Estructura y contenidos

Los contenidos que se abordan en la presente guía de estudio son los siguientes:

Modelado Estructural: Efectuar el modelado estructural de sistemas a través de diagramas de clases y objetos.

Modelado del Comportamiento: Efectuar el modelado de comportamiento de un sistema a través de diagramas de casos de uso, interacciones y de estados.

Modelado Arquitectónico: Efectuar el modelo arquitectónico de sistemas a través de diagramas de componentes y despliegue.

Evaluación

No aplica

Bibliografía Básica

- hhtp:\\www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/U ML.pdf
- Aprendiendo UML en 24 hrs, Joseph Smuller, Edit. Prentice Hall

Guía de Estudio

¿Qué es UML?

UML son las siglas de "Unified Modeling Language" o "Lenguaje Unificado de Modelado". Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software.

- **¿Que representan los modelos UML?**

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. ... Es importante remarcar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir.

- **Enumere los diagramas iniciales de UML.**

Diagrama de clases Los diagramas de clase son, sin duda, el tipo de diagrama UML más utilizado. Es el bloque de construcción principal de cualquier solución orientada a objetos. Muestra las clases en un sistema, atributos y operaciones de cada clase y la relación entre cada clase. En la mayoría de las herramientas de modelado, una clase tiene tres partes, nombre en la parte superior, atributos en el centro y operaciones o métodos en la parte inferior. En sistemas grandes con muchas clases relacionadas, las clases se agrupan para crear diagramas de clases. Las Diferentes relaciones entre las clases se muestran por diferentes tipos de flechas.

Diagrama de componentes Un diagrama de componentes muestra la relación estructural de los componentes de un sistema de software. Estos se utilizan principalmente cuando se trabaja con sistemas complejos que tienen muchos componentes. Los componentes se comunican entre sí mediante interfaces. Las interfaces se enlazan mediante conectores.

Diagrama de despliegue Un diagrama de despliegue muestra el hardware de su sistema y el software de ese hardware. Los diagramas de implementación son útiles cuando la solución de software se despliega en varios equipos, cada uno con una configuración única.

Diagrama de objetos Los diagramas de objetos, a veces denominados diagramas de instancia, son muy similares a los diagramas de clases. Al igual que los diagramas de clases, también muestran la relación entre los objetos, pero usan ejemplos del mundo real. Se utilizan para mostrar cómo se verá un sistema en un momento dado. Debido a que hay datos disponibles en los objetos, a menudo se utilizan para explicar relaciones complejas entre objetos.

Diagrama de paquetes Como su nombre indica, un diagrama de paquetes muestra las dependencias entre diferentes paquetes de un sistema.

Diagrama de perfiles El diagrama de perfil es un nuevo tipo de diagrama introducido en UML 2. Este es un tipo de diagrama que se utiliza muy raramente en cualquier especificación.

Diagrama de estructura compuesta Los diagramas de estructura compuesta se utilizan para mostrar la estructura interna de una clase.

Diagrama de actividades Los diagramas de actividad representan los flujos de trabajo de forma gráfica. Pueden utilizarse para describir el flujo de trabajo empresarial o el flujo de trabajo operativo de cualquier componente de un sistema. A veces, los diagramas de actividad se utilizan como una alternativa a los diagramas de máquina del estado.

Diagrama de casos de uso Como el tipo de diagrama de diagramas UML más conocido, los diagramas de casos de uso ofrecen una visión general de los actores involucrados en un sistema, las diferentes funciones que necesitan esos actores y cómo interactúan estas diferentes funciones. Es un gran punto de partida para cualquier discusión del proyecto, ya que se pueden identificar fácilmente los principales actores involucrados y los principales procesos del sistema.

Diagrama de máquina de estados Los diagramas de máquina de estado son similares a los diagramas de actividad, aunque las anotaciones y el uso cambian un poco. En algún momento se conocen como diagramas de estados o diagramas de diagramas de estado también. Estos son muy útiles para describir el comportamiento de los objetos que actúan de manera diferente de acuerdo con el estado en que se encuentran en el momento.

Diagrama de interacción. Los diagramas de interacción incluyen distintos tipos de diagramas:

Diagrama de secuencia Los diagramas de secuencia en UML muestran cómo los objetos interactúan entre sí y el orden en que se producen esas interacciones. Es importante tener en cuenta que muestran las interacciones para un escenario en particular. Los procesos se representan verticalmente y las interacciones se muestran como flechas. Los diagramas de secuencia de UML forman parte de un modelo UML y solo existen dentro de los proyectos de modelado UML.

Diagrama de comunicación El diagrama de comunicación se llamó diagrama de colaboración en UML 1. Es similar a los diagramas de secuencia, pero el foco está en los mensajes pasados entre objetos.

Diagrama de tiempos Los diagramas de sincronización son muy similares a los diagramas de secuencia. Representan el comportamiento de los objetos en un marco de tiempo dado. Si es solo un objeto, el diagrama es directo, pero si hay más de un objeto involucrado, también se pueden usar para mostrar interacciones de objetos durante ese período de tiempo.

Diagrama global de interacciones Los diagramas generales o globales de interacción son muy similares a los diagramas de actividad. Mientras que los diagramas de actividad muestran una secuencia de procesos, los diagramas de interacción muestran una secuencia de diagramas de interacción. En términos simples, pueden llamarse una colección de diagramas de interacción y el orden en que suceden. Como se mencionó anteriormente, hay siete tipos de diagramas de interacción, por lo que cualquiera de ellos puede ser un nodo en un diagrama de vista general de interacción.

- **¿Que representan los diagramas de estado, sustente su respuesta?**

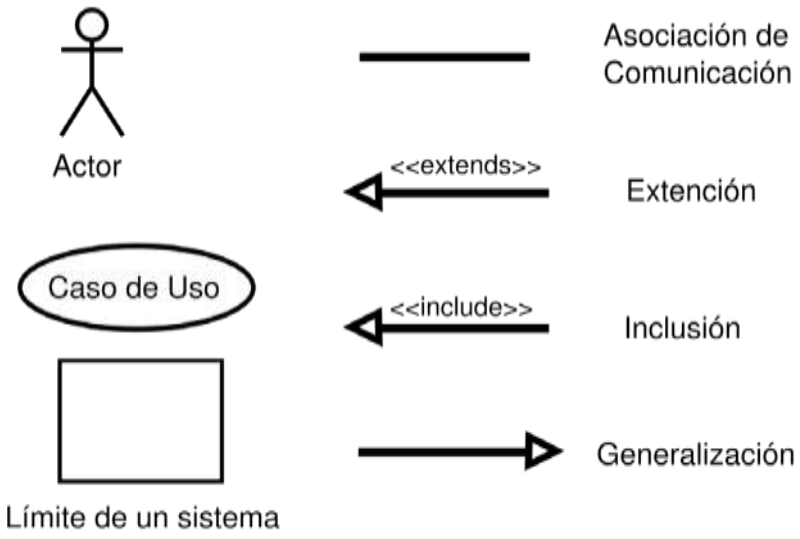
Un diagrama de estado muestra la secuencia de estados que un objeto o una interacción pueden atravesar durante su existencia en respuesta a los estímulos que vayan recibiendo, junto con las correspondientes respuestas y acciones. Los

Diagramas de Estados representan autómatas de estados finitos. Son útiles sólo para los objetos con un comportamiento significativo El resto de objetos se puede considerar que tienen un único estado.

- **¿Que representa un diagrama de casos de uso?**

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema.

- **Enumere los principales componentes de un diagrama de caso de uso.**



- **¿Qué tipos de relaciones se presentan en un diagrama de casos de uso?**

Comunicación: Relación (asociación) entre un actor y un caso de uso. El estereotipo de la relación de comunicación es: <<communicate>> aunque generalmente no se estipula ningún nombre, como podemos apreciar en el siguiente ejemplo de comunicación:

Inclusión: Un caso de uso base incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia. La relación de inclusión sirve para enriquecer un caso de uso con otro y compartir una funcionalidad común entre varios casos de uso, también puede utilizarse para estructurar un caso de uso describiendo sus subfunciones. El caso de uso incluido existe únicamente con ese propósito, ya que no responde a un objetivo de un actor.

Estas relaciones se representan mediante una flecha discontinua con el estereotipo <<include>>. Algunos casos de uso típicos de inclusión son: comprobar, verificar, buscar, validar, autentificar. En principio, no deberíamos abusar de este tipo de relación, para no hacer una descomposición funcional del sistema. A partir de UML 1.3 la relación <<include>> reemplazó al denominado <<uses>>.

Veamos un ejemplo de inclusión entre casos de uso:

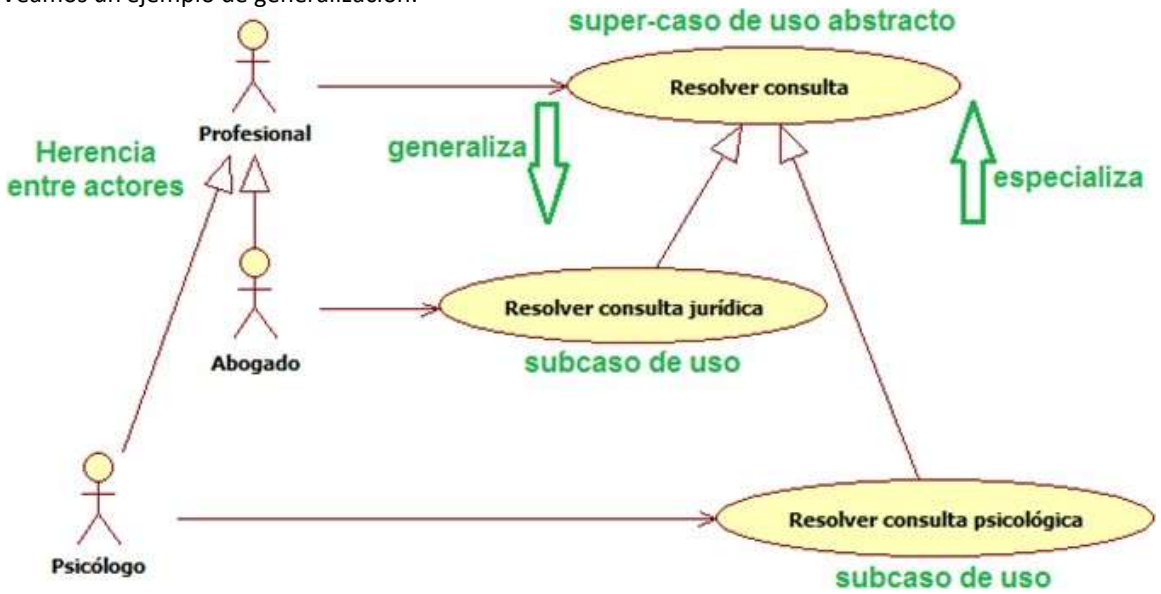
Extensión: Un caso de uso base incorpora implícitamente el comportamiento de otro caso de uso en el lugar especificado indirectamente por este otro caso de uso. En el caso de uso base, la extensión se hace en una serie de puntos concretos y previstos en el momento del diseño, llamados puntos de extensión, los cuáles no son parte del flujo principal. La relación de extensión sirve para modelar: la parte opcional del sistema, un subflujo que sólo se ejecuta bajo ciertas condiciones o varios flujos que se pueden insertar en un punto determinado. Este tipo de relación produce confusión y no debería utilizarse en exceso. Conviene su uso sólo para insertar un nuevo comportamiento no previsto en un caso de uso existente. Estas relaciones se representan mediante una flecha discontinua con el estereotipo <<extend>>.

Veamos un ejemplo de extensión:

En este ejemplo usamos la relación de extensión entre los casos de uso Abrir acción de mejora y Resolver consulta. En este caso tendremos el punto de extensión "resolución retrasada" (en el caso de uso Resolver consulta) debido a que cuando haya pasado un tiempo estipulado por la organización (por ejemplo 3 días laborales) se abrirá una acción de mejora para dejar constancia del retraso y realizar posteriormente las acciones pertinentes, de ahí que digamos que el caso de uso Abrir acción de mejora es una subfunción de uso que puede extender al caso de uso Resolver consulta.

Especialización y generalización de los casos de uso: Un caso de uso (subcaso) hereda el comportamiento y significado de otro, es decir las relaciones de comunicación, inclusión y extensión del supercaso de uso. En muchas ocasiones este supercaso de uso es abstracto y corresponde a un comportamiento parcial completado en el subcaso de uso. O dicho de otra manera, Los casos de uso "hijo" son una especialización del caso de uso "padre". En la medida de lo posible debería evitarse puesto que produce cierta confusión en algunas ocasiones.

Veamos un ejemplo de generalización:



Como podemos ver en este último ejemplo también pueden existir vínculos de generalización o herencia entre actores. Los actores especializados (Abogado y Psicólogo) heredan los casos de uso del actor general (Profesional). La punta de flecha apunta al actor más general. Hay que reseñar que los actores especializados pueden tener otros casos de uso propios que no estarán disponibles para los demás actores. Este tipo de herencia entre actores sí que se usa frecuentemente puesto que nos simplifica considerablemente el diagrama, nos ahorra un número importante de relaciones de comunicación entre actores y casos de uso y nos sirve para esclarecer visualmente la jerarquía entre actores del sistema.

- **Por favor tenga en cuenta como se documentan los casos de uso, si es necesario haga una consulta en internet sobre este tema y elabore un formato para que lo aplique a su proyecto formativo.**

- **Defina el concepto de clase.**

En ingeniería de software, un diagrama de clases en Lenguaje Unificado de Modelado (UML) es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos.

- **¿Que representa un diagrama de clases?**

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de agregación, ya que una clase es una descripción de conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica.

- **¿Qué es un atributo de una clase e indique los tipos de atributos?**

Los atributos se guardan en variables denominadas de instancia, y cada objeto particular puede tener valores distintos para estas variables. Atributo. Las variables de instancia también denominados miembros dato, son declaradas en la clase pero sus valores son fijados y cambiados en el objeto.

- **¿Qué entiende por herencia? Justifique su respuesta.**

La herencia es específica de la programación orientada a objetos, donde una clase nueva se crea a partir de una clase existente. La herencia (a la que habitualmente se denomina subclase) proviene del hecho de que la subclase (la nueva clase creada) contiene los atributos y métodos de la clase primaria. Es cuando una programación recibe la programación antigua y la hace parte de sí.

- **¿Qué entiende por Composición, Agregación, Asociación y dependencia, justifique su respuesta?**

La agregación es un tipo de asociación que indica que una clase es parte de otra clase (composición débil). Los componentes pueden ser compartidos por varios compuestos (de la misma asociación de agregación o de varias asociaciones de agregación distintas).

• ¿Para qué se utiliza un diagrama de secuencia?

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino.

¿Qué tipos de mensajes hay en un diagrama de secuencia y que entiende de cada uno de ellos? •

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino.

¿Que representa un diagrama de comunicación?

Un diagrama de comunicación es una forma de representar interacción entre objetos, alterna al diagrama de secuencia. Es un diagrama de clases que contiene roles de clasificador y los roles de asociación en lugar de solo clasificadores y asociaciones.

• ¿Qué es un diagrama de actividades?

El Diagrama de Actividad es un diagrama de flujo del proceso multipropósito que se usa para modelar el comportamiento del sistema. Los diagramas de actividad se pueden usar para modelar un Caso de Uso, o una clase, o un método complicado.