1. LENGUAJE UNIFICADO DE MODELADO (UML) 2

1.2. LOS OBJETIVOS DE UML SON 3

1.2.1. Visualizar 3

1.2.2. Especificar 3

1.2.3. Construir 3

1.2.4. Documentar 3

1.3. UML ESTÁ COMPUESTO POR TRES BLOQUES DE CONSTRUCCIÓN 3

1.3.1. Elementos 3

1.3.2. Relaciones 3

1.3.3. Diagramas 3

1.4. HAY VARIOS TIPOS DE ELEMENTOS UML 3

1.4.1. Elementos estructurales 3

1.4.2. Elementos de comportamiento 3

1.4.3. Elementos de agrupación 3

1.4.4. Elementos de anotación 3

1.4.5. Elementos de asociación 3

1.4.6. Elementos de agregación 3

1.4.7. Elementos de composición 4

1.4.8. Elementos de generalización 4

2. CLASIFICACIÓN DIAGRAMAS DE UML 4

2.2. DIAGRAMAS DE ESTRUCTURA 5

2.2.1. DIAGRAMA DE PAQUETES 5

2.2.2. DIAGRAMA DE CLASES 6

2.2.3. DIAGRAMA DE OBJETOS 6

2.2.4. DIAGRAMA DE COMPONENTES 7

2.3. DIAGRAMA DE COMPORTAMIENTO 8

2.3.1. DIAGRAMA DE ESTADO 8

2.3.2. DIAGRAMA DE CASOS DE USO 9

2.3.3. DIAGRAMA DE ACTIVIDAD 10

2.4. DIAGRAMAS DE INTERACCIÓN 12

2.4.1. DIAGRAMA DE COMUNICACIÓN 12

2.4.2. DIAGRAMAS DE SECUENCIAS 13

3. VISTAS ARQUITECTURALES 15

3.2. VISTA DE ESCENARIOS 16

DIAGRAMAS DE CASOS DE USO 16

3.3. VISTA LÓGICA 16

DIAGRAMAS DE CLASE, 16

DIAGRAMAS DE SECUENCIA 16

DIAGRAMAS DE COMUNICACIÓN. 16

3.4. VISTA DE PROCESOS 16

DIAGRAMAS DE ACTIVIDAD 16

3.5. VISTA FÍSICA 16

DIAGRAMA DE DESPLIEGUE. 16

3.6. VISTA DE DESPLIEGUE 16

DIAGRAMAS DE PAQUETES 17

DIAGRAMAS DE COMPONENTES 17

4. CONCEPTOS DE MODELO 17

4.2. SISTEMA 17

4.3. MODELO 17

4.4. VISTA (Arquitectural) 17

4.5. DIAGRAMA 17

## Lenguaje Unificado de Modelado (UML)

Es un lenguaje estándar para escribir planos de software centrado en la representación gráfica de un sistema.

### Los objetivos de UML son

#### Visualizar

Permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.

#### Especificar

Permite especificar las características de un sistema antes de su construcción.

#### Construir

A partir de los modelos especificados se pueden construir los sistemas diseñados.

#### Documentar

Los propios elementos gráficos sirven como documentación del sistema des- arrollado que pueden servir para su futura revisión.

### UML está compuesto por tres bloques de construcción

#### Elementos

Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)

#### Relaciones

Relacionan los elementos entre sí.

#### Diagramas

Son colecciones de elementos con sus relaciones.

### HAY VARIOS TIPOS DE ELEMENTOS UML

#### **Elementos estructurales:** Clase, Colaboración, Proceso, Caso de uso etc.

#### **Elementos de comportamiento**: Los elementos de comportamiento son las partes dinámicas de los modelos UML. Estos son los verbos de un modelo, y representan comportamientos en el tiempo y el espacio. Suelen estar conectados semánticamente a elementos estructurales. En total hay tres tipos principales de elementos de comportamiento que se pueden incluir en el modelo UML.

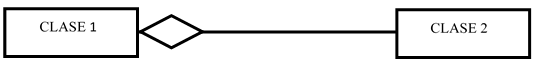
#### **Elementos de agrupación**: Los elementos de agrupación son las partes organizativas de los modelos UML. Estos son las cajas en las que se puede descomponer un modelo. Hay un tipo principal de elementos de agrupación.

#### **Elementos de anotación:** Nota.

#### **Elementos de asociación**

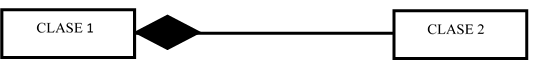
#### **Elementos de agregación**

La agregación Es muy similar a la relación de Asociación solo varía en la multiplicidad ya que en lugar de ser una relación "uno a uno" es de "uno a muchos" Se grafica con un rombo diamante vacío como lo vemos a continuación.



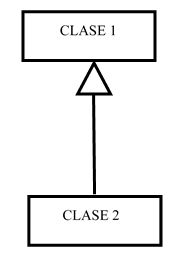
#### Elementos de composición

La Composición es una relación más fuerte y similar a la relación de asociación. Aporta documentación conceptual ya que es una "relación de vida", es decir, el tiempo de vida de un objeto está condicionado por el tiempo de vida del objeto que lo incluye.



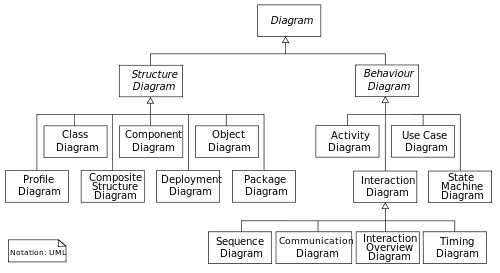
#### Elementos de generalización

Es otro nombre para herencia. Por ejemplo, Honda es un tipo de auto, por lo que la Clase “Honda” va a tener una relación de generalización con la Clase “Auto”. Esta se representa a través de un triángulo vacío hacia arriba.

****

## CLASIFICACIÓN DIAGRAMAS DE UML

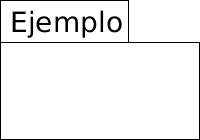
La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo.

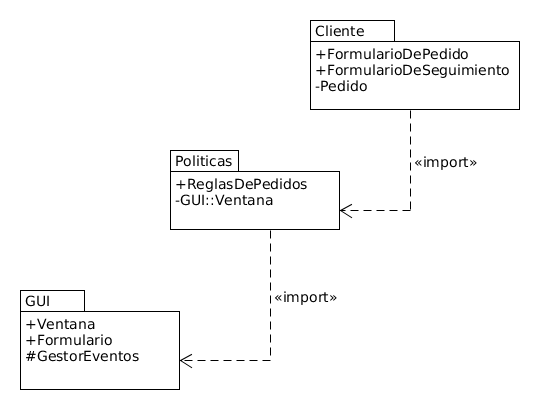


### DIAGRAMAS DE ESTRUCTURA

#### DIAGRAMA DE PAQUETES

Básicamente agrupan diferentes elementos de modelado en una entidad cohesiva. Estos por lo general se representan como una carpeta con el nombre de la misma en la pestaña.





Elementos:

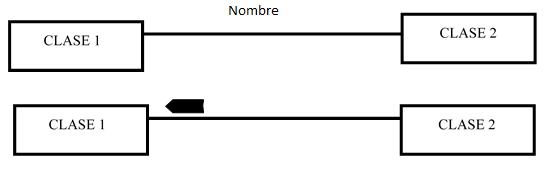
* Paquete: un mecanismo de propósito general para la organización de elementos y diagramas de modelos en grupos. Proporciona un espacio de nombres encapsulado dentro del cual todos los nombres deben ser únicos.
* Clase: una representación de un objeto que refleja su estructura y comportamiento dentro del sistema.
* Interfaz: una especificación de comportamiento. Una clase de implementación debe ser escrita para apoyar el comportamiento de una clase de interfaz.
* Objeto: una instancia de una clase. A menudo se utiliza en el análisis para representar un artefacto u otro elemento.
* Tabla: una clase estereotipada.

#### DIAGRAMA DE CLASES

Los diagramas de clases describen la estructura estática de un sistema.

Elementos

|  |
| --- |
| NOMBRE DE LA CLASE |
| Atributos: Tipo |
| Acciones() |



#### DIAGRAMA DE OBJETOS

Un diagrama de objetos puede ser visto como una instancia de un diagrama de clases. Los diagramas de objetos describen la estructura estática de un sistema en un momento particular y son usados para probar la precisión de los diagramas de clases.

Elementos estructurales de un diagrama de objetos

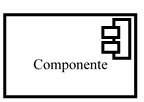
|  |
| --- |
| Nombre Objeto: clase |
| Atributo tipo= ‘valor’  Atributo tipo= ‘valor’ |
| Acciones |

#### DIAGRAMA DE COMPONENTES

Este tipo de diagramas se utilizan para representar entidades reales y componentes de software. Un componente de software puede ser una tabla, un archivo de datos, un ejecutable, documentos etc.

Elementos

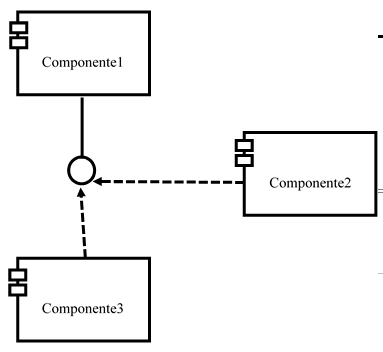
Componente: Es un rectángulo con otros dos rectángulos sobrepuestos. Se puede añadir en su interior, información que indique detalles del componente.

****

Interface: Describe a un grupo de operaciones usada o creada por componentes, estas están representadas con una línea y al final un círculo.

****

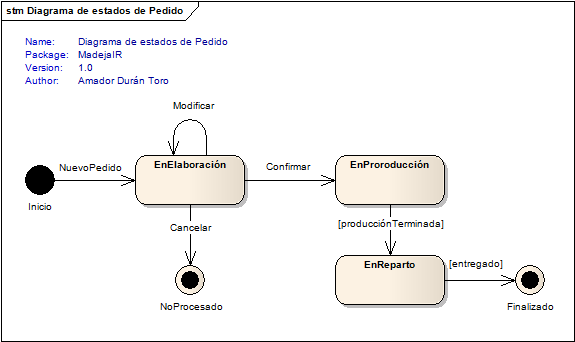
Las dependencias entre los componentes se grafican usando flechas con puntos.

****

### DIAGRAMA DE COMPORTAMIENTO

#### DIAGRAMA DE ESTADO

Es el diagrama que muestra a una máquina de estado, y son útiles para modelar la vida de un objeto



**CONCEPTOS CLAVES**

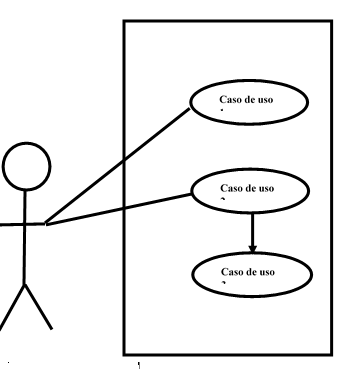
* ESTADO: un estado es una condición o situación en la vida de un objeto durante la cual satisface una condición, realiza alguna actividad o espera algún evento.
* EVENTO: es la especificación de un acontecimiento significativo que ocupa un lugar en el tiempo y en el espacio. Es la aparición de un estímulo que puede o no activar una transición de estado.
* TRANSICION: es una relación entre dos estados que indica que un objeto que está en el primer estado, realizará ciertas acciones y entrará en el segundo estado cuando ocurra un evento especificado y se satisfagan unas condiciones específicas.

Elementos:

* Círculo relleno, señalando al estado inicial.
* Círculo hueco que contiene un círculo más pequeño lleno, lo que indica el estado final (si existe)
* Rectángulo redondeado, lo que denota un estado. Parte superior del rectángulo contiene un nombre del estado. Puede contener una línea horizontal en el medio, por debajo del cual las actividades que se realizan en ese estado se indican
* Flecha, que denota la transición. El nombre del evento (si los hay) que causa esta transición etiqueta el cuerpo de la flecha. Una expresión de guardia puede añadirse antes un "/" y encerrado en corchetes cuadrados (eventName [guardExpression]), denotando que esta expresión debe ser cierto para la transición se lleve a cabo. Si se realiza una acción durante esta transición, que se añade a la siguiente etiqueta un "/" (eventName [guardExpression] / acción).
* Línea horizontal gruesa, ya sea con x> 1 líneas de entrada y 1 línea de salida o 1 línea de entrada y x> 1 líneas de salir. Estos denotan unirse / tenedor, respectivamente.

#### DIAGRAMA DE CASOS DE USO

Modela la funcionalidad de un sistema según lo perciben los usuarios externos, llamados actores.



Elementos

* Actores: las entidades externas al sistema
* Sistema: Rectángulo que representa al sistema y encierra a los casos de uso y las relaciones
* Caso de uso: Acciones de los actores
* Relación: Relacionan casos de uso y casos de uso con entidades externas

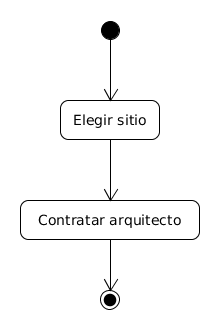
#### DIAGRAMA DE ACTIVIDAD

Es una variante de una máquina de estados, que muestra las actividades de computación implicadas en la ejecución de un cálculo.

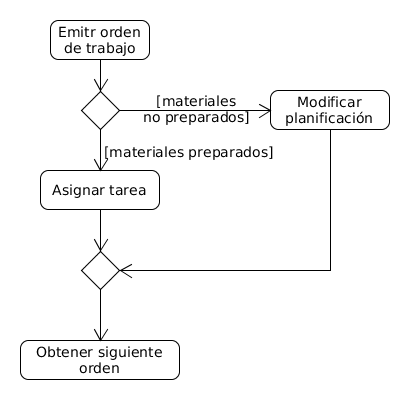


Nodos de actividad: Estos son elementos que encapsulan una serie de pasos en uno solo. A diferencia de las acciones, estos pueden ser divididos en otras acciones, es decir no son atómicos y se representan igual a las acciones excepto con paréntesis al final.

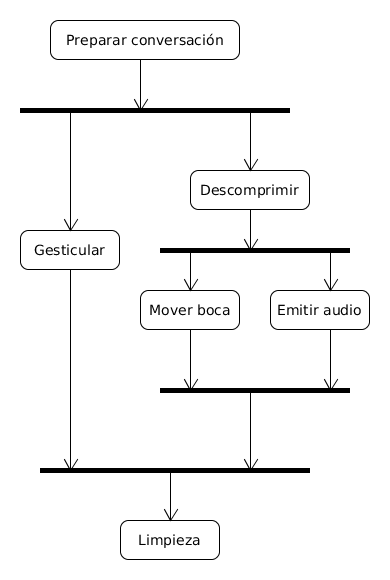
Flujos: Estos permiten describir el flujo u orden en que se ejecutan las acciones en el diagrama de actividad. Se describen con unas flechas y especifican cual acción ocurre después de otra. Para especificar el comienzo del proceso a modelar se coloca un punto relleno y se coloca una flecha saliendo de ella hacia la primera acción, y para especificar el fin se coloca un círculo con un círculo relleno dentro:



Ejemplo:



Otra herramienta útil es la división y unión. La división representan momentos en donde dos o más flujos se ejecutan concurrentemente y la unión representa cuando se deben volver a sincronizar los flujos. Estos se representan con una barra vertical o horizontal.



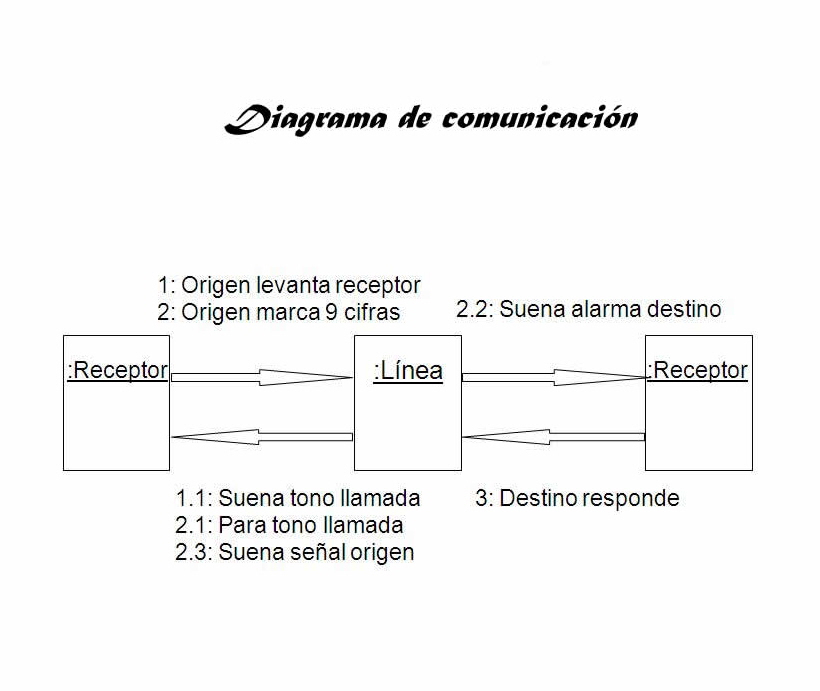
Flujos de objetos: especifican creación y manipulación de objetos en el diagrama de actividad. Estos se representan con recuadros, dentro se debe especificar el nombre del objeto y el estado entre corchetes.

### DIAGRAMAS DE INTERACCIÓN

#### DIAGRAMA DE COMUNICACIÓN

Un diagrama de comunicación es una versión simplificada del diagrama de colaboración de la versión de UML 1.X

Este diagrama describe las interacciones entre los objetos en términos de mensajes secuenciales donde se numera que mensajes ocurren primero que otros.



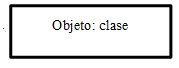
Elementos:

* Instancias de clases: Rectángulo con el nombre de la instancia subrayado.
* Mensajes: Los mensajes se muestran como flechas etiquetadas unidas a los enlaces. Cada mensaje tiene un número de secuencia, una lista opcional de mensajes precedentes, una condición opcional de guarda, un nombre, una lista de argumentos y un nombre de valor de retorno opcional.
* Enlace: Línea que conecta a los objetos que se envían mensajes.

#### DIAGRAMAS DE SECUENCIAS

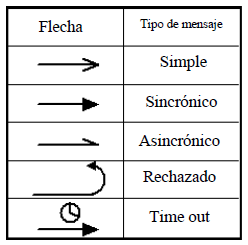
Muestra los conjuntos de mensajes, dispuestos en una secuencia temporal. Cada tol en la secuencia se muestra como una línea de vida es decir una línea vertical que representa el rol por cierto plazo de tiempo

El rol de la clase describe la manera en que un objeto se va a comportar en el contexto. En estas no se listan los atributos del objeto.

****

Los cuadros de activación representan el tiempo que un objeto necesita para completar una tarea.

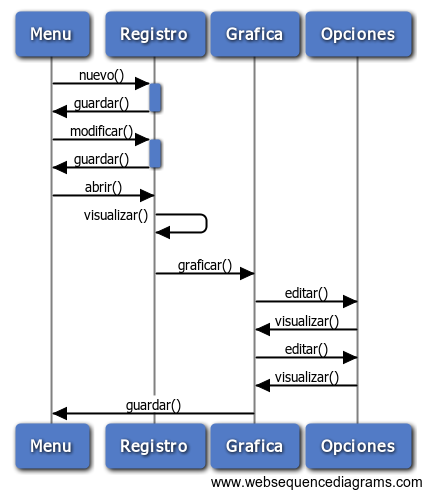
Los mensajes son flechas que representan comunicaciones entre objetos. Las medias flechas representan mensajes asincrónicos. Los mensajes asincrónicos son enviados desde un objeto que no va a esperar una respuesta del receptor para continuar con sus tareas.

****

Las líneas de vida son verticales y en línea de puntos, ellas indican la presencia del objeto durante el tiempo.

Los objetos pueden ser eliminados tempranamente usando una flecha etiquetada "<<destruir>>" que apunta a una X.

Una repetición o loop en un diagrama de secuencias, es representado como un rectángulo. La condición para abandonar el loop se coloca en la parte inferior entre corchetes [ ]. Es un tipo de diagrama usado para modelar interacción entre objetos en un sistema.

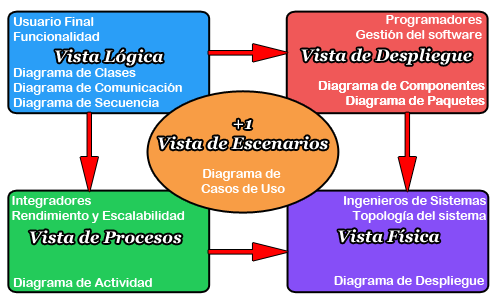


Objetos usados en el diagrama de secuencia:

* Mensaje: pasa de la línea de vida de un objeto a otro.
* Objeto: rectángulos con nombres subrayados, el tiempo se Representa como una progresión vertical.
* Línea de vida activa: el tiempo se representa en forma vertical inicia en la parte superior y avanza a la parte inferior un mensaje que este en la parte superior ocurrirá antes que uno en la parte inferior.

## VISTAS ARQUITECTURALES

Durante el desarrollo de un sistema puede ser visto desde varias perspectivas



### VISTA DE ESCENARIOS

Captura la funcionalidad del sistema tal y como es percibido por los usuarios finales. Los diagramas correspondientes a esta vista son el

#### DIAGRAMAS DE CASOS DE USO

### VISTA LÓGICA

Los elementos que la conforman dan soporte a los requisitos funcionales del sistema. Los diagramas correspondientes son los

#### DIAGRAMAS DE CLASE,

#### DIAGRAMAS DE SECUENCIA

#### DIAGRAMAS DE COMUNICACIÓN.

### VISTA DE PROCESOS

Abarca en especial requisitos no funcionales el rendimiento, la escalabilidad y capacidad de procesamiento. Los diagramas correspondientes son los

#### DIAGRAMAS DE ACTIVIDAD

### VISTA FÍSICA

Se ocupa de la distribución de las partes que forman el sistema de software real. Los diagramas que corresponden es el

#### DIAGRAMA DE DESPLIEGUE.

### VISTA DE DESPLIEGUE

Captura los artefactos que se utilizan para ensamblar y poner en producción el sistema software real. Corresponde en esta vista a los

#### DIAGRAMAS DE PAQUETES

#### DIAGRAMAS DE COMPONENTES

## CONCEPTOS DE MODELO

### SISTEMA

Colección de elementos, posiblemente divididos en subsistemas, organizados para lograr un propósito. Está descrito por un conjunto de modelos.

### MODELO

Simplificación completa y auto consistente de la realidad, creado para comprender un sistema.

### VISTA (Arquitectural)

Proyección de la organización y estructura de un modelo, centrada en su aspecto. Incluye un subconjunto de los elementos en el modelo.

### DIAGRAMA

Representación gráfica de un conjunto de elementos del modelo y sus relaciones.