

5.2. El lenguaje UML

UML (Unified Modeling Language) es un lenguaje gráfico utilizado para visualizar, especificar, construir y documentar los componentes de un sistema de software. Surgió en 1996 y la versión más reciente es la 2.5.1 de 2015.

5.2.1. Tipos de elementos en UML

Elementos estructurales: Son la parte estática de un modelo y representan conceptos o cosas materiales.

1. **Clase:** Descripción de un conjunto de objetos con atributos y métodos comunes.
2. **Interfaz:** Conjunto de operaciones que una clase o componente ofrece.
3. **Colaboración:** Define interacciones entre elementos que cooperan.
4. **Caso de uso:** Describe secuencias de acciones para un usuario.
5. **Clase activa:** Objeto que puede ejecutar concurrentemente.
6. **Componente:** Parte modular del diseño que oculta su implementación.
7. **Artefacto:** Parte física y reemplazable de un sistema que contiene información.
8. **Nodo:** Elemento físico en tiempo de ejecución.

Elementos de comportamiento: Son las partes dinámicas de los modelos UML y representan comportamiento en el tiempo y espacio.

1. **Interacción:** Conjunto de mensajes intercambiados.
2. **Máquina de estados:** Secuencias de estados por los que pasa un objeto.
3. **Actividad:** Secuencia de pasos que ejecuta un proceso.

Elementos de agrupación: Son organizativos de los modelos UML. El principal es el **paquete**, que organiza construcciones de implementación.

Elementos de anotación: Comentarios y notas que se añaden para describir, aclarar y hacer observaciones sobre los elementos.

5.2.2. Tipos de diagramas en UML

Diagramas estructurales:

1. **Diagramas de clases:** Muestran un conjunto de clases y relaciones.
2. **Diagramas de objetos:** Muestran objetos y sus relaciones.
3. **Diagramas de componentes:** Describen la estructura del software.
4. **Diagramas de despliegue:** Muestran nodos de procesamiento y artefactos.
5. **Diagramas de paquetes:** Descomponen el modelo en unidades organizativas.
6. **Diagramas de perfiles:** Extienden UML para su uso en una plataforma particular.
7. **Diagramas de estructura compuesta:** Muestran la estructura interna de un elemento.

Diagramas de comportamiento:

1. **Diagramas de casos de uso:** Ayudan a entender el comportamiento desde la perspectiva del usuario.
2. **Diagramas de actividades:** Muestran el flujo paso a paso de una computación.
3. **Diagramas de estados:** Muestran una máquina de estados.
4. **Diagramas de interacción:** Muestran cómo interactúan los objetos, se dividen en:

- Diagramas de secuencia
- Diagramas de colaboración
- Diagramas de tiempos
- Diagrama global de interacciones

5.3. Clases, atributos, métodos y visibilidad

Una **clase** tiene atributos y métodos. Los atributos son las propiedades y los métodos las operaciones que pueden realizar los objetos de la clase. Se representan en diagramas UML con tres secciones: nombre de la clase, atributos y métodos.

5.4. Relaciones entre clases

Las relaciones entre clases pueden ser de varios tipos:

- **Agregación:** Representa una relación todo-parte. Los componentes pueden existir independientemente del compuesto.
- **Composición:** Similar a la agregación, pero los componentes no pueden existir independientemente del compuesto.

5.4.3. Generalización y especialización

Las relaciones de **generalización-especialización** ocurren cuando se establece una jerarquía de clases y subclases basada en atributos y métodos comunes. En UML:

- **Generalización:** Las subclases heredan atributos y métodos de una superclase.
- **Especialización:** Las subclases añaden atributos y métodos específicos además de los heredados de la superclase.

5.4.4. Asociación

Las **asociaciones** son relaciones entre clases que pueden ser de diferentes tipos:

- **Reflexivas:** Una clase está asociada consigo misma.
- **Binarias:** Entre dos clases.
- **N-arias:** Relaciones de grado mayor que dos (ternarias, cuaternarias, etc.).

Navegabilidad

La **navegabilidad** en UML indica en qué dirección se puede acceder a la información entre las clases asociadas. Se representa mediante una punta de flecha en la línea de la asociación.

Relación de Realización

Una **relación de realización** se da entre una clase **Interface** y las clases que implementan esa interfaz. Permite definir métodos comunes que varias clases pueden implementar.

Tipos de clases de análisis

Durante la fase de análisis, se identifican tres tipos de clases: ![[Pasted image 20240519173220.png]]

1. **Clases de interfaz:** Modelan la interacción entre el sistema y sus actores externos.
2. **Clases de entidad:** Modelan información que persiste en el tiempo.
3. **Clases de control:** Coordinan y controlan otros objetos del sistema.