# Cuestionario UML

**1. ¿Qué es UML y para qué se utiliza?**

El Lenguaje Unificado de Modelado (UML) es el lenguaje que define un sistema, para detallar los artefactos en el sistema, para documentar y construir, ya que es una forma de mostrar visualmente el comportamiento y la estructura de un sistema o proceso. también ayuda a mostrar errores potenciales en las estructuras de aplicaciones, el comportamiento del sistema y otros procesos empresariales.

**2. Nombrar y describir los distintos tipos de diagramas UML**

Existen dos tipos principales de diagramas UML: **diagramas de estructura** y **diagramas de comportamiento** (y dentro de esas categorías se encuentran varios otros).

Estas variaciones existen para representar los numerosos tipos de escenarios y diagramas.

**Diagramas estructurales**

[Los diagramas estructurales](https://support.office.com/en-us/article/create-a-uml-static-structure-diagram-bb11b3df-8830-4e31-a437-332585da7ce8) representan la estructura estática de un software o sistema, y también muestran diferentes niveles de abstracción e implementación. Estos se usan para ayudar a visualizar las diversas estructuras que componen un sistema, como una base de datos o aplicación.

dentro de los diagramas estructurales se encuentran:

* **Diagrama de clases.** Este diagrama, el más común en el desarrollo de software, se usa para representar el diseño lógico y físico de un sistema, y muestra sus clases. Tiene un aspecto similar al del diagrama de flujo porque las clases se representan con cuadros. Este diagrama ofrece una imagen de las diferentes clases y la forma en la que se interrelacionan, y cada clase posee tres compartimientos:

\_Sección superior: nombre de clase

\_Sección central: atributos de clase

\_Sección inferior: métodos u operaciones de clase

* **Diagrama de objetos.** A menudo, este diagrama se usa como una forma de comprobar la revisión de un diagrama de clases para fines de precisión. En otras palabras muestra los objetos de un sistema y sus relaciones, y ofrece una mejor visión de los potenciales defectos de diseño que necesitan reparación.
* **Diagrama de componentes.** También conocido como diagrama de flujo de componentes, muestra agrupaciones lógicas de elementos y sus relaciones. Es decir, ofrece una vista más simplificada de un sistema complejo al desglosarlo en componentes más pequeños.
* **Diagrama de estructura compuesta.** Aunque es similar a un diagrama de clases, adopta un enfoque más profundo, describe la estructura interna de múltiples clases y muestra las interacciones entre ellas.
* **Diagrama de despliegue.** Este diagrama muestra los componentes de hardware (nodos) y software (artefactos) y sus relaciones. Ofrece una representación visual exacta del lugar donde se implementa cada componente de software.
* **Diagrama de paquetes.** se utiliza para representar las dependencias entre los paquetes que componen un modelo. Su objetivo principal es mostrar la relación entre los diversos componentes grandes que forman un sistema complejo.
* **Diagrama de perfiles.** es más similar a un lenguaje que a un diagrama. Un diagrama de perfil ayuda a crear nuevas propiedades y semántica para los diagramas UML al definir estereotipos personalizados, valores marcados y restricciones.

**diagramas de comportamiento**

El enfoque aquí está en los aspectos dinámicos del sistema de software o proceso. En estos diagramas se muestra la funcionalidad de un sistema y se enfatiza lo que debe ocurrir en el sistema que se está modelando.

dentro de los diagramas de comportamiento se encuentran:

* **Diagrama de actividades.**Este representa un proceso paso a paso con un inicio y final claros
* **Diagrama de casos de uso.** Este describe lo que un sistema hace las cosas, pero no la forma en que las hace.
* **Diagrama de descripción general de interacción.** Este diagrama es similar al diagrama de actividad, ya que ambos muestran una secuencia paso a paso de las actividades. Sin embargo este incorpora elementos como la interacción, el uso de la interacción, restricción de tiempo y restricción de la duración.
* **Diagrama de tiempos.** Cuando el tiempo ocupa un lugar central, se usa este diagrama de UML. También conocido como un diagrama de secuencia o eventos, el enfoque aquí está en la duración de los eventos y los cambios que se producen en función de las restricciones de duración. Las principales partes de un diagrama de plazos incluye:

\_Línea de vida: participante individual

\_Línea de tiempo de estado: estados diferentes por los que pasa la línea de vida dentro de una canalización

\_Restricción de duración: tiempo necesario para que se cumpla una restricción

\_Restricción de tiempo: un periodo en el que el participante debe completar una acción

\_Destrucción: cuando finaliza la línea de vida de un objeto.

* **Diagrama de máquina de estados.** También denominado gráfico de estados, este diagrama se aplica cuando el comportamiento de un objeto es complejo y el detalle es esencial. describe la forma en que cambia según los eventos internos y externos.
* **Diagrama de secuencia.** Popular más allá de la comunidad de diseño, este diagrama visualmente atractivo es bueno para mostrar todo tipo de procesos empresariales
* **Diagrama de comunicación.** Un diagrama de comunicación o colaboración es similar a un diagrama de secuencia. Sin embargo, enfatiza la comunicación entre objetos. Muestra la organización de los objetos que participan en una interacción y presenta iteraciones y ramificaciones más complejas.

**3. ¿Cuáles son las ventajas que ofrece?**

* Simplifica las complejidades
* Mantiene abiertas las líneas de comunicación
* Automatiza la producción de software y los procesos
* Ayuda a resolver los problemas arquitectónicos constantes
* Aumenta la calidad del trabajo
* Reduce los costos y el tiempo de comercialización

**4. ¿Cuáles son los elementos estructurales que contiene? Describir cada uno de ellos.**

Estos siete elementos **(clases, interfaces, colaboraciones, casos de uso, clases activas, componentes y nodos)** son los elementos estructurales básicos que se pueden incluir en el modelo UML.

* **Clases:** una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Las clases pueden representar cosas como hardware, software o solo conceptuales y también forman parte de una distribución equilibrada de responsabilidades en el sistema
* **interfaz:** Es una colección de operaciones que especifican un servicio de una clase o componente y representa el comportamiento completo del mismo

.

* **colaboración** Se definen como una interacción y es una sociedad de roles y otros elementos que colaboran para proporcionar un comportamiento cooperativo mayor que la suma de los comportamientos de sus elementos las colaboración representan la implementación de patrones que forman un sistema.
* **caso de uso** Es una descripción de secuencias de acciones que un sistema ejecuta y que produce un resultado observable de interés para un actor particular. Se utiliza para estructurar el comportamiento en un modelo.
* **Clases activas:** Es un objeto que tiene uno o más procesos o hilos de ejecución y por lo tanto pueden dar origen a actividades de control. Son iguales a las clases excepto en que sus objetos representan a elementos cuyo comportamiento es concurrente con otros elementos.
* **componente** Es una parte física y reemplazable de un sistema que conforma con un conjunto de interfaces y proporciona la implementación de dicho conjunto.Representa típicamente el empaquetamiento físico de diferentes elementos lógicos, como clases interfaces y colaboraciones
* **nodo** es un elemento físico que existe en tiempo de ejecución y representa un recurso computacional que dispone de memoria y con frecuencia capacidad de procesamiento.

**5. Explique con sus palabras qué entiende por orientación a objetos.**

la programación **orientada a objetos** busca que representemos nuestro código y algoritmos como una forma más cercana a cómo expresamos las cosas en la vida real; La programación orientada a objetos disminuye los errores y nos permite reutilizar el código ya que lo organiza en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.