**Cuestionario UML**

**1. ¿Qué es y para qué se utiliza 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.

UML sirve para representar visualmente las reglas de creación, estructura y comportamiento de un grupo relacionado de objetos y procesos. Para visualizar de forma eficiente la complejidad de un sistema u organización en un reducido número de diagramas, además de mantener mucho más ágilmente las especificaciones ante los cambios y nuevas actualizaciones de arquitectura.

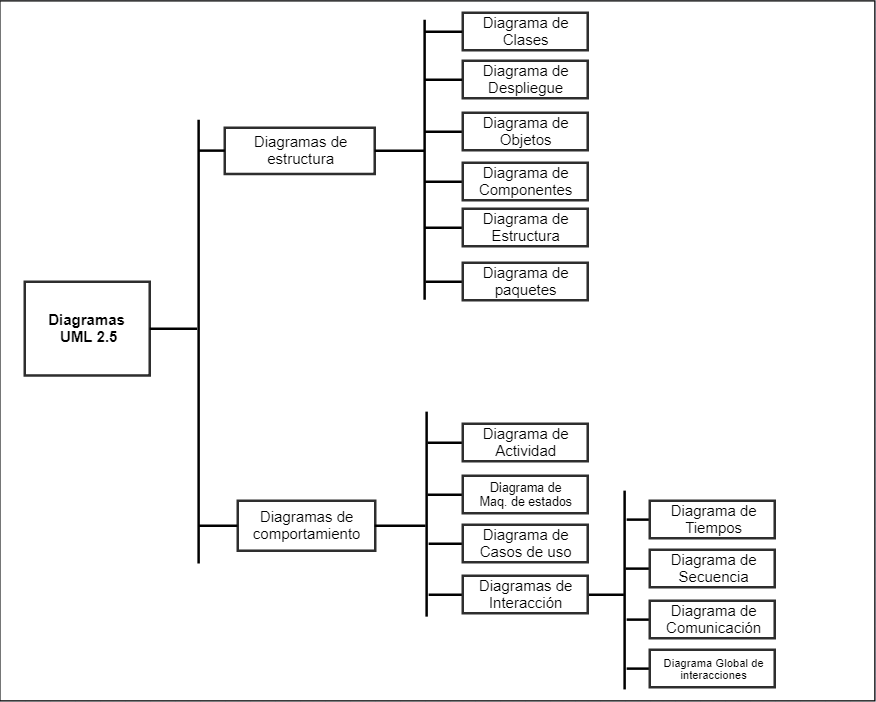
**2. Ventajas de la utilización de UML.**

Las ventajas del uso de UML son:

* Mejores tiempos totales de desarrollo (de 50 % o más).
* Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
* Establecer conceptos y artefactos ejecutables.
* Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
* Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
* Mejor soporte a la planeación y al control de proyectos.
* Alta reutilización y minimización de costos.

**3. ¿Cuántos tipos de diagrama UML existen actualmente?**

Los tipos de diagramas UML existentes a dia de hoy son:

  
**4. Definición de Diagramas UML estructurales y Diagramas UML de comportamiento.**

**Diagramas estructurales**

Los diagramas estructurales muestran la estructura estática del sistema y sus partes en diferentes niveles de abstracción. Existen un total de siete tipos de diagramas de estructura:

**Diagrama de clases**

Muestra la estructura del sistema, subsistema o componente utilizando clases con sus características, restricciones y relaciones: asociaciones, generalizaciones, dependencias, etc.

**Diagrama de componentes**

Muestra componentes y dependencias entre ellos. Este tipo de diagramas se utiliza para el desarrollo basado en componentes (CDB), para describir sistemas con arquitectura orientada a servicios (SOA).

**Diagrama de despliegue**

Muestra la arquitectura del sistema como despliegue (distribución) de artefactos de software.

**Diagrama de objetos**

Un gráfico de instancias, incluyendo objetos y valores de datos. Un diagrama de objeto estático es una instancia de un diagrama de clase; muestra una instantánea del estado detallado de un sistema en un punto en el tiempo.

**Diagrama de paquetes**

Muestra los paquetes y las relaciones entre los paquetes.

**Diagrama de perfiles**

Diagrama UML auxiliar que permite definir estereotipos personalizados, valores etiquetados y restricciones como un mecanismo de extensión ligero al estándar UML. Los perfiles permiten adaptar el metamodelo UML para diferentes plataformas o dominios.

**Diagrama de estructura compuesta**

Muestra la estructura interna (incluidas las partes y los conectores) de un clasificador estructurado.

**Diagramas de comportamiento**

A diferencia de los diagramas estructurales, muestran cómo se comporta un sistema de información de forma dinámica. Es decir, describe los cambios que sufre un sistema a través del tiempo cuando está en ejecución. Hay un total de siete diagramas de comportamiento, clasificados de la siguiente forma:

**Diagrama de actividades**

Muestra la secuencia y las condiciones para coordinar los comportamientos de nivel inferior, en lugar de los clasificadores que poseen esos comportamientos. Estos son comúnmente llamados modelos de flujo de control y flujo de objetos.

**Diagrama de casos de uso**

Describe un conjunto de acciones (casos de uso) que algunos sistemas o sistemas (sujetos) deben o pueden realizar en colaboración con uno o más usuarios externos del sistema (actores) para proporcionar algunos resultados observables y valiosos a los actores u otros interesados ​​del sistema(s).

**Diagrama de máquina de estados**

Se utiliza para modelar el comportamiento discreto a través de transiciones de estados finitos. Además de expresar el comportamiento de una parte del sistema, las máquinas de estado también se pueden usar para expresar el protocolo de uso de parte de un sistema.

**5. Explicar los distintos tipos de modelado para base de datos.**

Los modelos de bases de datos más comunes son:

* Modelo de base de datos jerárquico
* Modelo relacional
* Modelo de red
* Modelo de base de datos orientado a objetos
* Modelo entidad-relación
* Modelo relacional de objetos, que combina los dos que forman su nombre

**Modelo jerárquico**

El modelo jerárquico organiza los datos en una estructura de árbol, en la que cada registro tiene un único elemento o raíz. Los registros del mismo nivel se clasifican en un orden específico. Ese orden se usa a manera de orden físico para almacenar la base de datos.

**Modelo relacional**

Siendo el modelo más común, el modelo relacional ordena los datos en tablas, también conocidas como relaciones, cada una de las cuales se compone de columnas y filas.

**Modelo de red**

El modelo de red se basa en el modelo jerárquico, permitiendo relaciones de muchos a muchos entre registros vinculados, lo que implica registros principales múltiples. Basado en la teoría matemática de conjuntos, el modelo se construye con conjuntos de registros relacionados.

**Modelo de base de datos orientado a objetos**

Este modelo define una base de datos como una colección de objetos, o elementos de software reutilizables, con funciones y métodos relacionados. Hay varios tipos de bases de datos orientadas a objetos:

Una base de datos multimedia incorpora elementos multimedia, tales como imágenes, que no se podrían almacenar en una base de datos relacional.

Una base de datos de hipertexto permite que cualquier objeto se conecte a cualquier otro objeto. Es útil para organizar lotes de datos disímiles, pero no es ideal para análisis numérico.

**Modelo entidad-relación**

Este modelo capta las relaciones entre entidades del mundo real de forma muy similar al modelo de red, pero no está directamente ligado a una estructura física de la base de datos. En cambio, con frecuencia se lo usa para diseñar una base de datos conceptualmente.

**Modelo relacional de objetos**

Este modelo de base de datos híbrido combina la sencillez del modelo relacional con parte de la funcionalidad avanzada del modelo de base de datos orientado a objetos. En esencia, permite a los diseñadores incorporar objetos en una estructura familiar de tablas.