**Cuestionario UML**

Preguntas:

1. ¿Qué es y para qué se utiliza UML?
2. Ventajas de la utilización de UML.
3. ¿Cuántos tipos de diagrama UML existen actualmente?
4. Definición de Diagramas UML estructurales y Diagramas UML de comportamiento.
5. Explicar los distintos tipos de modelado para base de datos.

Respuestas:

1. El UML o lenguaje unificado de modelado es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

Fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. UML tiene aplicaciones más allá del desarrollo de software, p. ej., en el flujo de procesos en la fabricación.

1. Los principales beneficios 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.

1. UML usa elementos y los asocia de diferentes formas para formar diagramas que representan aspectos estáticos o estructurales de un sistema, y diagramas de comportamiento, que captan los aspectos dinámicos de un sistema.

* Diagramas UML estructurales:
* Diagrama de clases: El diagrama UML más comúnmente usado, y la base principal de toda solución orientada a objetos. Las clases dentro de un sistema, atributos y operaciones, y la relación entre cada clase. Las clases se agrupan para crear diagramas de clases al crear diagramas de sistemas grandes.
* Diagrama de componentes: Muestra la relación estructural de los elementos del sistema de software, muy frecuentemente empleados al trabajar con sistemas complejos con componentes múltiples. Los componentes se comunican por medio de interfaces.
* Diagrama de estructura compuesta: Los diagramas de estructura compuesta se usan para mostrar la estructura interna de una clase.
* Diagrama de implementación: Ilustra el hardware del sistema y su software. Útil cuando se implementa una solución de software en múltiples máquinas con configuraciones únicas.
* Diagrama de objetos: Muestra la relación entre objetos por medio de ejemplos del mundo real e ilustra cómo se verá un sistema en un momento dado. Dado que los datos están disponibles dentro de los objetos, estos pueden usarse para clarificar relaciones entre objetos.
* Diagrama de paquetes: Hay dos tipos especiales de dependencias que se definen entre paquetes: la importación de paquetes y la fusión de paquetes. Los paquetes pueden representar los diferentes niveles de un sistema para revelar la arquitectura. Se pueden marcar las dependencias de paquetes para mostrar el mecanismo de comunicación entre niveles.
* Diagramas UML de comportamiento:
* Diagramas de actividades: Flujos de trabajo de negocios u operativos representados gráficamente para mostrar la actividad de alguna parte o componente del sistema. Los diagramas de actividades se usan como una alternativa a los diagramas de máquina de estados.
* Diagrama de comunicación: Similar a los diagramas de secuencia, pero el enfoque está en los mensajes que se pasan entre objetos. La misma información se puede representar usando un diagrama de secuencia y objetos diferentes.
* Diagrama de panorama de interacciones: Hay siete tipos de diagramas de interacciones. Este diagrama muestra la secuencia en la cual actúan.
* Diagrama de secuencia: Muestra cómo los objetos interactúan entre sí y el orden de la ocurrencia. Representan interacciones para un escenario concreto.
* Diagrama de máquina de estados: Similar a los diagramas de actividades, describen el comportamiento de objetos que se comportan de diversas formas en su estado actual.
* Diagrama de temporización: Al igual que en los diagramas de secuencia, se representa el comportamiento de los objetos en un período de tiempo dado. Si hay un solo objeto, el diagrama es simple. Si hay más de un objeto, las interacciones de los objetos se muestran durante ese período de tiempo particular.
* Diagrama de caso de uso: Representa una funcionalidad particular de un sistema. Se crea para ilustrar cómo se relacionan las funcionalidades con sus controladores (actores) internos/externos.

1. Definición de diagramas UML estructurales y diagramas UML de comportamiento.

* Diagramas UML estructurales: representan la estructura estática de un software o sistema, y también muestran diferentes niveles de abstracción e implementación. Estos se usan para ayudarlo a visualizar las diversas estructuras que componen un sistema, como una base de datos o aplicación.
* Diagramas UML de comportamiento: se define formalmente como diagrama que expresa las secuencias de estados por los que pasa un objeto a lo largo de su vida en respuesta a eventos. Hablando en un lenguaje más llano, se trata de diagramas que muestran diferentes estados de un proceso.

1. El UML es popular entre programadores, pero no suele ser usado por desarrolladores de bases de datos. Una razón es sencillamente que los creadores de UML no se enfocaron en las bases de datos. A pesar de ello, el UML es efectivo para el modelado de alto nivel de datos conceptuales y se puede usar en diferentes tipos de diagramas UML. Hay varios tipos de modelado en uml:

* Modelo de base de datos jerárquico: Un modelo antiguo, pero bueno. Los datos de este modelo están organizados en una estructura de árbol. El árbol está compuesto por varios grupos llamados segmentos. Utiliza una relación de uno a muchos. El acceso a los datos también es predecible.
* Modelo de red: Este modelo adopta la forma de un gráfico, donde los tipos de relación son arcos y los tipos de objeto son nodos. A diferencia de otros modelos de bases de datos, el esquema del modelo de red no se limita a una red o jerarquía.
* Modelo de base de datos orientado a objetos: Este modelo utiliza una colección de objetos, o elementos de software reutilizables, con características y métodos asociados. Por ejemplo, una base de datos multimedia podría tener imágenes que no se pueden almacenar en una base de datos relacional. O una base de datos de hipertexto permite establecer vínculos con otros objetos.
* Modelo relacional: Aquí, los datos se estructuran utilizando relaciones o estructuras matemáticas similares a una cuadrícula que tienen columnas y filas. Básicamente, es una tabla.
* El modelo objeto-relacional: Como su nombre lo indica, este modelo es una combinación de los dos mencionados anteriormente. Admite objetos, clases, herencia y otros elementos orientados a objetos, pero también admite tipos de datos, estructuras tabulares y más, como en un modelo de datos relacionales.
* Modelo entidad-relación: Este se compone de tipos de entidad (personas, lugares o cosas). Muestra las relaciones que pueden existir entre ellos. Al definir las entidades, sus atributos y mostrar las relaciones entre ellas, un diagrama ER ilustra la estructura lógica de las bases de datos.
* Modelo de documento: Está diseñado para almacenar y administrar documentos o datos semiestructurados, en lugar de datos atómicos. Tiene una estructura de árbol en la que cada nodo es un objeto que representa una parte del documento.
* Modelo de entidad-atributo-valor: En el EAV o los modelos de esquema abierto, los datos se registran en tres columnas:

1. La entidad (lo que se describe)
2. El atributo o parámetro (por ejemplo, nombre, descripción, tipo de datos)
3. El valor del atributo.

* Esquema de estrella: Esta es la versión más simple de un modelo dimensional, en el que los datos se organizan en dimensiones y hechos. Se utiliza en inteligencia empresarial y almacenamiento de datos, ya que es adecuado para consultar conjuntos de macrodatos.