13. UML

A. ¿Qué es y para que se usa UML?

**Visión general**

**“UML es un lenguaje de modelado de software”**

Suficientemente expresivo para cubrir distintas vistas de la arquitectura del software a lo largo del ciclo de vida.

Mayor nivel de abstracción que un lenguaje de programación, ya que es independiente del lenguaje de programación.

UML es un lenguaje de Modelado. Es un lenguaje estándar para escribir planos de software.

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos.

Tiene una notación gráfica muy expresiva que permite representar en mayor o menor medida fases del proyecto.

Nos indica cómo se pueden crear y leer modelos.

---------------------------------------------------------------------------------------------------

UML puede usarse en las diferentes etapas del ciclo de vida del desarrollo

Objetivos de UML

1-VISUALIZAR

2-ESPECIFICAR software

3-CONSTRUIR software

4-DOCUMENTAR

UML es un lenguaje para VISUALIZAR los elementos de un gran sistema software.

Esto facilita:

– la comunicación entre los desarrolladores

– la comprensión de las soluciones (notación gráfica)

– el mantenimiento de las soluciones conceptuales a lo largo del tiempo (ya que queda como documentación)

UML es un lenguaje para ESPECIFICAR software:

-Se pueden construir modelos precisos (no ambiguos) y completos.

– Cubre decisiones de análisis, diseño, etc.

UML permite especificar las características de un sistema antes de su construcción

UML un lenguaje para CONSTRUIR software

-No es un lenguaje de programación visual, pero sus modelos se pueden conectar de forma directa a una gran variedad de ellos.

– Correspondencias entre UML y lenguajes: Java, C++, etc.

UML es un lenguaje para DOCUMENTAR:

-requisitos, arquitectura, diseño, código fuente, pruebas, ...

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

Diagramas UML

-Diagrama de clases

- Diagrama de casos de uso

- Diagrama de secuencia

- Diagrama de colaboración

- Diagrama de estados

- Diagrama de actividades

- Diagrama de componentes

- Diagrama de despliegue

B. ¿Qué es un modelo? Y podría indicar ¿Para qué modelamos?

***“Un modelo es una abstracción de la realidad”***

Un modelo proporciona un plano de un sistema.

Incluye aquellos aspectos importantes del sistema y omite aquellos elementos menores que no son relevantes.

Cada modelo nos permite fijarnos en un aspecto distinto del sistema.

Un Modelo de Software es una Simplificación de la Realidad.

¿Para qué modelamos?

Construimos modelos para comprender mejor el sistema que estamos desarrollando.

Los Modelos:

Nos ayudan a visualizar cómo queremos que sea un sistema.

Nos ayudan a especificar la estructura o el comportamiento de un sistema.

Nos guían en la construcción de un sistema.

Documentan las decisiones que hemos adoptado.

Construimos modelos de sistemas complejos porque no podemos comprender el sistema en su totalidad.

Hay límites de la capacidad humana para comprender la complejidad.

A través del modelado, se reduce el problema que se esta estudiando, centrándose en un solo aspecto a la vez. (divide y vencerás)

C. ¿Qué entiende por Abstracción? De ejemplos de clases existentes en la vida real.

1-La abstracción es el proceso intelectual a través del cual separamos mentalmente las cualidades particulares de varios objetos para fijarnos únicamente en uno o diversas características comunes.

2-La abstracción es un concepto donde se separa o se aísla toda información que no resulta útil o relevante, y esto se hace a través de un proceso mental. Dicha clasificación ocurre definiendo ciertas características, funciones o atributos de una entidad.

Ejemplos

**Al nombrar un gato**: Si alguien dice la palabra gato, automáticamente la persona oyente sabe lo que es un gato y que este animal maúlla, caza ratones, entre otras cosas. Por lo que cada vez que alguien menciona la palabra gato, no es necesario definir todas sus características, ya que las personas de inmediato realizan un proceso de abstracción definiendo al animal.

**Extraer las características de un objeto como una motocicleta:** Primero se definen que similitudes tienen todas las motocicletas y luego se toman factores como: marca, modelo, tamaño, rendimiento, entre otros.

**Estaciones del año:** Por medio de la abstracción las personas logran diferenciar las características de cada estación del año, agrupando cada una con una serie de conceptos.

**Consultas médicas:** En una consulta médica, el doctor va desechando la información innecesaria que dice el paciente y acumula mentalmente solo aquella que le aporta datos de la enfermedad del paciente.

**Problemas cotidianos**: Al resolver un problema de la vida cotidiana, se separa o se desecha (se abstrae) todo lo que no resulte relevante para su solución, centrando el pensamiento en las cosas que son importantes y que sí ayuden a resolver el problema.

D. Investigue sobre UML y la Ingeniería directa e inversa

<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

**La ingeniería inversa o retroingeniería** es el proceso llevado a cabo con el objetivo de obtener información o un diseño a partir de un producto, con el fin de determinar cuáles son sus componentes y de qué manera interactúan entre sí y cuál fue el proceso de fabricación.

Actualmente (principios del siglo XXI), los productos más comúnmente sometidos a ingeniería inversa son los programas de computadoras y los componentes electrónicos, pero realmente cualquier producto puede ser objeto de un análisis de Ingeniería Inversa.

El método se denomina así porque avanza en dirección opuesta a las tareas habituales de ingeniería, que consisten en utilizar datos técnicos para elaborar un producto determinado. En general, si el producto u otro material que fue sometido a la ingeniería inversa fue obtenido en forma apropiada, entonces el proceso es legítimo y legal. De la misma forma, pueden fabricarse y distribuirse, legalmente, los productos genéricos creados a partir de la información obtenida de la ingeniería inversa, como es el caso de algunos proyectos de software libre ampliamente conocidos.

**Ingeniería Directa** es cuando primero modelas el software y después lo programas. También aplica para el Hardware.

