

## Proyecto Software de comunicaciones.

A continuación se presentan una serie de consideraciones importantes a la hora de realizar el proyecto de software de comunicaciones:

- **Planeamiento (Conceptualizar todo)**

- Descripción del proyecto
- Materiales y componentes a utilizar
- Factibilidad del proyecto
- Viabilidad del proyecto
- Planos
- Localización
- Tamaños y medidas (se propone una escala específica)
- Distribución (cableado, equipos, etc.)
- Protocolos
- Estándares (ISO)

- **Diseño**

- Requerimientos
- Procesos funcionales y no funcionales
- arquitectura
- Interacción con el usuario
- Interfaces (Entradas y salidas)

-Módulos

-Bases de datos

-Web server

Se considera si se utilizan uno o varios servidores.

- **Desarrollo**

-Pruebas (Testing)

-Validación de procesos

-Calidad

-Funcionabilidad

-Estadísticas

- **Sistemas existentes o versiones modificadas.**

-SO utilizados

Se considera si se usa una versión del software disponible o si debe realizarse una modificación.

- **Entregables**

-Reportes según las fases del proyecto

- **Versiones del proyecto**

Considerar modificaciones del proyecto a futuro, ya sean actualizaciones del sistema o modificaciones estructurales.

- **Impacto q genera el proyecto(aportes)**

- **Ciclo de vida del proyecto**

-Sistemas con versiones modificables

-Prototipos

-Alcances

-Limitaciones

“El conocimiento nunca será una limitación”

-Pruebas de rendimiento

- **Documentación**

Se documenta toda la información útil para la realización del proyecto, a partir del planeamiento del proyecto, pruebas de funcionamiento, entrega al usuario y versiones futuras.

Además se documenta todas las pruebas posibles para la realización del proyecto, ya sea por medio de un diagrama, fotos, etc.

# UML

UML: Lenguaje de modelos unificado

Lenguaje: conjunto de ideas para formar un mensaje.

Modelo: forma para representar un estándar que sirve para realizar una nueva tarea.  
Tiene un propósito determinado.

Nota: Un modelo se crea de acuerdo a un público determinado.

## Ejemplos de modelos:

- Mapa de transporte de una ciudad
- Plano de instalaciones sanitarias

El UML puede mostrar los límites de un sistema y sus funcionalidades principales, utilizando casos de uso y actores.

## Descripción análoga

El mapa no es la ciudad si no es una representación de la ciudad.

## Observación directa

Es un concepto abstracto

No se podrán ver datos específicos

## Propósito determinado

Determina la creación del modelo

El mapa sirve para llegar de un lugar a otro

## Abstracción

Lo que modelamos no podría existir

## **Modelos de software**

- EL modelo no es un sistema de software en si si no una representación del mismo
- El software es en sí un modelo
- El modelo más detallado de un producto de software es el código fuente en sí, es decir, que el mejor modelo del edificio es el plano.
- El software es mucho más modificable que otros productos realizados por el hombre.
- Cada producto es un nuevo problema

## **Más de UML**

- Notación de modelo visual

Diagramas para mostrar distintos aspectos del sistema.

Lenguaje de visualización, especificación y documentación del software, basados en 13 tipos de diagramas.

## **UML-¿Es un lenguaje sí o no?**

Si, en cuanto es una herramienta de comunicación formal.

- Serie de construcciones
- Sintaxis definida
- Semántica definida

## **Usos**

- Herramienta de comunicación entre humanos

La mejor forma de desarrollar un UML es mediante la entrevista directa con los usuarios o posibles futuros usuarios del sistema, poniendo atención a cada una de las actividades o pasos que se van a ir desarrollando desde un primer momento hasta un momento final.

## Recomendaciones

1.

Es difícil mantener actualizados cientos de miles de diagramas

2.

La documentación debe ser más útil que abundante

Si se requieren almacenar como documentación se pueden tomar fotografías y mantenerlos en una Wiki.

3.

Los métodos son impulsores del uso del UML

## Otros usos:

### **MDD (Model Driven Develoment)**

-Existen modelos que surgen del análisis y mediante una serie de pasos controlados se llega al código fuente del sistema de forma automática.

- El objetivo del MDD (Model Driven Develoment) es "programar" en un nivel más alto de abstracción.

-Cabe destacar que con esta herramienta se limita la creatividad.

## Los 13 diagramas de UML 2.2

### **Los diagramas estructurales o estáticos:**

- 4 de casos de uso
- 5 de objetos
- 2 de clases
- 1 de paquetes
- 1 de componentes
- 3 de despliegue
- 4 de estructuras compuestas

### **Y diagramas de comportamiento dinámico:**

- 3 de secuencia
- 4 de colaboración
- 3 de estados
- 2 de actividades
- 5 de visión global de la interacción
- 5 de tiempos