**PARADIGMAS Y SOFTWARE**

**METODOLOGÍA Y PARADIGMA**

Una metodología es el elemento que enuncia cuales son las herramientas que se van a usar, quienes realizaran las tareas, cuando son tareas críticas entre otras cosas. Al referirnos a paradigma damos como opciones el estructurado o el orientado a objetos estos no son otra cosa que la forma de desarrollar o ver el problema.

**EL PROCESO DE SOFTWARE**

Estas son las 4 faces características de todo proceso

* Requerimientos
* Construcción
* Evolución

En un proceso de software el modelo nos aportara lo siguiente

* Define las tareas necesarias para el correcto funcionamiento por etapa y en general
* Describe y delimita las faces
* Permite la ubicación del proyecto
* Facilita la estimación de tiempos y costos
* Ubica a los recursos en un momento dado y en una tarea determinada

Básicamente podemos dividir los modelos en 3 tipos:

* Lineales: presentados en cascada cada etapa esta determinada y sigue una secuencia definida esperando resultados anteriores
* Evolutivos: son de diversa índole aunque se destacan el exploratorio y el desechable haciendo hinchapie en el resultado rápido y su confrontación con los deseos del cliente
* Componentes : desarrollo de introducción posterior en el mundo donde es el software aunque es muy conocido entre otras industrias

**MODELO EN CASCADA**

Existe desde hace más de 35 año y sigue siendo el pilar para tener noción sobre cuáles son las etapas que se requiere cumplir antes de tener un entregable de mínima funcionalidad el desarrollo es dividido en faces secuenciales y con metas bien definidas

**ESPECIFICACIÓN**

En esta fase necesitamos definir el dominio del software que vamos a desarrollar comprender sus fines utilidad y limitaciones almenos en grandes rasgos. Para esto recolectaremos información que debe ser fácil de estructurar y almacenar porque será fundamental a lo largo de todo el desarrollo y los problemas o falencias en esta etapa pueden llevar al fracaso

**ANÁLISIS**

Se deben de terminar las áreas que realizará el sistema y sus estructuras procesos críticos y segundarios una vez tengamos una idea acabada del objetivo comenzaremos a diagramar el sistema aplicando alguna de las metodologías de análisis que consideraremos adecuada al proyecto en cuestión. Por sus propiedades las tareas del análisis son denominadas:

* Conceptualización: el objetivo de obtener una vista de alto nivel alto de abstracción se focaliza en el sistema y sus partes teniendo en cuenta los elementos externos
* Análisis funcional: se trata de identificar y describir los procesos internos del sistema
* Análisis de condiciones especificación de las restricciones de nuestro sistema las limitaciones pueden ser de tipo técnico físico económico o humano
* Aplicar la metodología: luego de las etapas anteriores aplicaremos algún proceso de diagramación
* Validación: si nuestro análisis es correcto deberemos pasar a la etapa siguiente pero para asegurarnos tenemos que someter este análisis a una completa validación para evitar inconsistencias

**DISEÑO**

En el diseño aplicamos prácticas de ingeniería para definir de forma técnica los procesos elementos y datos del sistema el diseño profundas y completa la visión del software pero además le agrega los componentes técnicos el resultado de la fase de diseño será la base para el comienzo de la construcción del software. Según las definiciones tradicionales en el diseño se materializan los requerimientos del cliente en general podemos identificar tres etapas

* Diseño de estructuras: el diseño arquitectónico es el eje de la construcción
* Diseño de estructura de datos: se definen y documentan todos los datos que serían utilizados por el sistema
* Diseño de interfaz: se decide desde un punto de vista técnico formal el tipo de interfaz que utilizaremos y cualquier otro elemento con el que deberíamos interactuar incluido el usuario

**CODIFICACIÓN**

En la fase de diseño se eligieron los leguajes con los que sería escrito en esta etapa debemos destinar recursos para que el sistema que ya ha sido analizado y diseñado pueda comenzar a ser una realidad

**PRUEBAS**

la prueba del software es un conjunto que se lleva a cabo para localizar identificar y eliminar errores y mejorar el producto

**MANTENIMIENTO**

Luego de que el sistema haya sido probado y su funcionamiento es el correcto debemos intentar realizar tareas que permitan prolongar su vida útil. Dentro el mantenimiento podemos observar

* Monitoreo: es el conjunto de procesos que evalúan la calidad del control en el tiempo y permiten al sistema reaccionar en forma dinámica
* Mantenimiento correctivo: es la reparación de errores en el momento en el que aparecen
* Mantenimiento preventivo: consiste en la corrección de errores de forma programada o la ejecución de tareas para que estas no aparezcan

**MODELOS EVOLUTIVOS**

Los modelos evolutivos suponen la confrontación de lo desarrollado con los deseos de los clientes

**PROTOTIPOS E INCREMENTAL**

El modelo de prototipado es un modelo evolutivo que establece iteraciones cortas de forma tal de mostrarle los avances al cliente a fin de poder acelerar el desarrollo. El modelo incremental presenta grandes parecidos con el modelo de prototipado se busca satisfacer al cliente desarrollando un conjunto de todos los requisitos que se solicitó al equipo de trabajo

**MODELO EN ESPIRAL**

Es un modelo mixto propuesto por Barry Boehm que conjuga las prácticas del modelo clásico junto a tendencias evolutivas

**MODELO DE DESARROLLO DE COMPONENTES**

El modelo de desarrollo de componentes des un intento de la mejora sobre modelo de construcción en espiral se materializa en la práctica gracias a los paradigmas de análisis diseño y programación orienta da a objetos

**PARADIGMAS DE DESARROLLO**

El desarrollo estructurado a objetos son paradigmas o modelos de desarrollo

**DESARROLLO ORIENTADO A OBJETOS**

El software presenta características particulares que hacen que su complejidad alcance niveles importantes. Detrás de estos paradigmas se encuentran distintas visiones de cómo se puede disminuir la complejidad y obtener software de calidad

**EL MODELO DE OBJETOS**

No solo es una forma de programar sino que nos brinda los cimien tos para dore desarrollar todo tipo de soluciones bajo sus principios muchas de las premisas de este paradigma han sido obtenidas de otras ingenierías y actividades

E modelo orientado a objetos presenta algunos elementos que son denominados fundamentales que son:

* Jerarquía
* Abstracción
* Modularidad
* Encapsulamiento

Si alguno de estos elementos esta ausente no podemos considerar paradigma como modelo de objetos además existen otros elementos obcionales

* Tipos(tipificación)
* Concurrencia
* Persistencia

**JERARQUÍA**

No es más que un ordenamiento de niveles de lo que vamos a presentar

**ABSTRACCIÓN**

Es un proceso de intelectual humano por el cual somos capaces de concentrarnos particularmente en las características que nos interesan para la solución de una situación

**MODULARIDAD**

Este concepto no es propio de la OO sino es uno de los que más se ha desarrollado en la ingeniería del software El objetivo final es la división de un problema complejo en unidades más pequeñas y casi siempre más sencillas

**ENCAPSULAMIENTO**

El encapsulamiento es el ocultamiento de la información de forma que tal que solo esté disponible para interactuar con un objeto sin la necesidad de conocer cómo se comporta internamente

**TIPOS (TIPIFICACIÓN)**

Los tipos permiten representar las abstracciones de forma adecuada es más sencillo ejemplificar el concepto diciendo que so realizamos una suma de 2 números enteros esperamos tener otro número entero y no con un decimal

**CONCURRENCIA**

La concurrencia supone que en la solución de un problema se manejan toda clase de eventos y muchas veces algunos interactúan de manera asincrónica pero también pueden hacerlo simultáneamente

**PERSISTENCIA**

La persistencia es lo que posibilita que trabajemos guardando información durante el tiempo que necesitamos operar con ella

**EL OBJETO COMO BASE**

La orientación a objetos presenta muchas características comunes a otro tipo de prácticas lo que se la distingue es que ponme énfasis a definir y caracterizar de forma clara los componentes del sistema dotándolos de sus capacidades

**PROGRAMACIÓN ORIENTADA A OBJETOS**

Una excelente definición que nos permite solucionar el problema de explicar que es la programación orientada a objetos es la utilizada por los seguidores del desarrollador e ingeniero de software