

# PARADIGMAS Y SOFTWARE

## METODOLOGIA Y PARADIGMA

Antes de comenzar con los contenidos centrales es importante que comencemos algunos temas que suelen prestarse a confusión. Comencemos diciendo que el proceso de creación del software es cíclico. Es así como todo software tiene su propio ciclo y cada proyecto contenido en uno o varios procesos, también. Existe un ciclo de vida del software que puede ser dividido en fases. Cada una de ellas debe ser completada mediante la aplicación de tareas, técnicas y herramientas. Una metodología es el elemento que nos enuncia cuales son las herramientas que se van a usar quienes realizarán las tareas, en que momento y cuales son consideradas tareas críticas entre otras cosas.

## EL PROCESO DE SOFTWARE

El proceso de software es el conjunto de finido, limitado y coherente de actividades que permiten el desarrollo de software. A lo largo de la historia del desarrollo se ha presentado cuatro fases las cuales son:

- Requerimientos
- Construcción
- Validación
- Evolución

Estas etapas representan el ciclo de vida de un proyecto y pueden ser ordenadas de acuerdo a un modelo. Un modelo es una representación simplificada de la realidad. Básicamente podemos dividir los modelos en tres tipos

- Lineales
- Evolutivos
- Componentes

### Modelo en cascada

Cuando nos acercamos por primera vez a los desarrollos se nos enseña el modelo en cascada. Este existe desde hace mas de 35 años 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.

### Especificación

Esta etapa es comúnmente denominada recolección de requisitos o especificación de requisitos, aunque en algunos entornos se le llama ingeniería de sistema. En esta fase necesitamos definir el dominio del software que vamos a desarrollar comprender sus fines, utilidad y limitaciones, al menos en grandes rasgos.

### Análisis

Se debe determinar las tareas que realizarán el sistema y sus estructuras. Una vez que tengamos una idea acabada del objetivo comenzamos a diagramar el sistema, aplicando alguna de las metodologías de análisis que consideramos adecuada al proyecto.

# PARADIGMAS Y SOFTWARE

## Diseño

En el diseño aplicamos prácticas para definir los procesos, elementos y datos del sistema. El diseño profundiza y completa el software pero además le agrega los componentes técnicos. Según las definiciones adicionales en el diseño se materializaran los requerimientos del cliente. Y podemos encontrar 3 capas que son: diseño de estructuras, diseño de estructura de datos y diseño de interfaz.

## Codificación

En esta etapa debemos designar 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 de practicas que se llevan a cabo para localizar, identificar y eliminar errores y mejorar el producto. En general, cuando nos planteamos desarrollos de gran envergadura, un plan de pruebas adecuado puede ser la diferencia entre un software robusto y un software deficiente.

## Mantenimiento

Debemos intentar realizar tareas que permitan prolongar su vida útil. Las operaciones de mantenimiento proponen lograr que el sistema mantenga el nivel de ejecución satisfactorio que obtuvo en las pruebas. Dentro del mantenimiento podemos observar:

- Monitoreo
- Mantenimiento correctivo
- Mantenimiento preventivo

Para facilitar el entendimiento hemos intentado agrupar todas las tareas nos gustaría reflejar aquí una serie completa de las actividades que se realizan habitualmente

- Revisión del sistema
- Toma de requisitos
- Análisis de requisitos
- Reingeniería de procesos
- Diseño
- Programación
- Integración
- Pruebas
- Explotación
- Mejora
- Ampliación

## 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 para contrastar resultados, muchos de los prototipos son construidos haciendo hincapié en características visibles como las interfaces, los reportes o la carga de datos.

## PARADIGMAS Y SOFTWARE

El modelo incremental presenta grandes parecido con el modelo de prototipado se busca satisfacer al cliente desarrollando un conjunto de todos los requisitos que el solicito el equipo de trabajo.

### Modelo en espiral

Es un modelo mixto propuesto por barry boehm que conjuga las practicas del modelo clásico junto a tendencias evolutivas. El resultado es un modelo con el cual se desarrollan versiones de software con mayor funcionalidad por iteración.

Se definen como cuatro actividades:

- Planificación
- Análisis
- Construcción
- Evaluación

### Modelo en desarrollo de componentes

Es un intento de mejorar sobre el modelo de construcción en espiral. Se puede materializar en la práctica gracias a los paradigmas de análisis, diseño y programación orientada a objetos. Con sus características esenciales, los lenguajes orientados a objetos permiten generar componenetes denominados clases.

## PARADIGMAS DE DESARROLLO

Comenzamos al iniciar el capítulo las diferencias entre metodología y paradigma. El desarrollo estructurado y el orientado a objetos son paradigmas a 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 paradigma orientado a objetos plantea que un sistema puede ser visto como objetos que tienen ciertas características y que colaboran entre ellos para realizar una tarea.

No es solo una manera de programar si no que nos brinda los cimientos para poder desarrollar todo tipo de soluciones bajo sus principios muchas de las premisas de este paradigma han sido obtenidas de otras ingenierías y actividades.

### El modelo de objetos

No es solo una manera de programar si no que nos brinda los cimientos para poder desarrollar todo tipo de soluciones bajo sus principios muchas de las premisas de este paradigma han sido obtenidas de otras ingenierías y actividades.

El modelo orientado a objetos presenta algunos elementos que son denominados fundamentales.

- Jerarquía: no es más que la posibilidad de realizar un ordenamiento en niveles de lo que deseamos representar.
- Abstracción: es un proceso intelectual humano por el cual somos capaces de concentrarnos particularmente en las características que nos interesan para la solución de una situación.

## **PARADIGMAS Y SOFTWARE**

- Modularidad: el objetivo final es la división de un problema complejo en unidades más pequeñas y casi siempre más sencilla.
- Encapsulamiento: es el ocultamiento de la información de forma tal que solo esté disponible para interactuar con un objeto sin la necesidad de conocer cómo se comporta internamente.

### **Persistencia**

Es lo que posibilita que trabajemos guardando información durante el tiempo que necesitamos operar con ella.

### **El objeto como base**

Lógicamente la orientación a objetos presenta muchas características comunes a otro tipo de prácticas lo que la distingue es que pone el énfasis en 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 Grady Boach, incluida en su libro análisis y diseño orientado a objetos.



PARADIGMAS Y SOFTWARE  
TECNOLOGICO NACIONAL CEICOM  
CARRERA DE SISTEMAS INFORMATICOS

## PRACTICA 02

### PARADIGMAS Y SOFTWARE

**Materia:** Análisis Diseño Sistemas

**Elaborado por:** Michael Oliver prado morales

**Docente:** ing. Baltazar Llusco Ever Jaime

**Fecha:** 18 de junio de 2018

Cochabamba-Bolivia