

**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE MANABÍ MANUEL FÉLIX LÓPEZ**

**CARRERA DE COMPUTACIÓN**

**SEMESTRE QUINTO PERÍODO MAR 2017 – AGO. 2017**

**TRABAJO DE EXPOSICIÒN**

**TEMA:**

**MODELOS DE PROCESOS DESCRIPTIVOS**

**ESTUDIANTES:**

**CINTYA P. ALVAREZ MOREIRA**

**DENNY M. COBEÑA BRAVO**

**MARJORIE S. KUFFO ZAMBRANO…**

**FACILITADOA:**

**MGTR. AURA D. ZAMBRANO RENDON**

**CALCETA, MAYO 2017**

# MARCO TEORICO

1. **MODELOS DE PROCESOS DESCRIPTIVOS**

# MODELO DE LA CASCADA

Es conocido también como modelo de desarrollo clásico. Este modelo coloca las actividades estructurales una a continuación de la otra siguiendo un flujo descendente en el proceso. (Berenguel, 2016).

Es el enfoque metodológico que ordena rigurosamente las etapas del [proceso para el desarrollo de software](http://es.wikipedia.org/wiki/Proceso_para_el_desarrollo_de_software), de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.( Pressman, 2010)

Ingeniería y Análisis del Sistema

Análisis de los Requisitos

Diseño

Codificación

Prueba

Mantenimiento

Figura 1. Ilustración del modelo de cascadas (Berenguel, 2016).

# ANÁLISIS DE REQUISITOS

En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.

Es importante señalar que en esta etapa se debe consensuar todo lo que se requiere del sistema y será aquello lo que seguirá en las siguientes etapas, no pudiéndose requerir nuevos resultados a mitad del proceso de elaboración del software.

# DISEÑO DEL SISTEMA

Es la fase en donde se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario así como también los análisis necesarios para saber que herramientas usar en la etapa de Codificación.

# CODIFICACIÓN

Es la fase en donde se implementa el [código fuente](http://es.wikipedia.org/wiki/C%C3%B3digo_fuente), haciendo uso de prototipos así como de pruebas y ensayos para corregir [errores](http://es.wikipedia.org/wiki/Error_de_software).

Dependiendo del lenguaje de programación y su versión se crean las bibliotecas y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.

# PRUEBAS

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.

# VERIFICACIÓN

Es la fase en donde el usuario final ejecuta el sistema, para ello el o los programadores ya realizaron exhaustivas pruebas para comprobar que el sistema no falle.

# MANTENIMIENTO

Una de las etapas más críticas, ya que se destina un 75% de los recursos, es el mantenimiento del Software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas.

# MODELO DE PROCESOS INCREMENTAL

Se trata de un modelo de proceso que aplica secuencias lineales en forma escalonada a medida que el calendario de actividades va avanzando. Cada secuencia lineal produce un incremento de software susceptible de entregarse de manera parecida a los incrementos producidos en un flujo de proceso evolutivo. (Pressman, 2010).

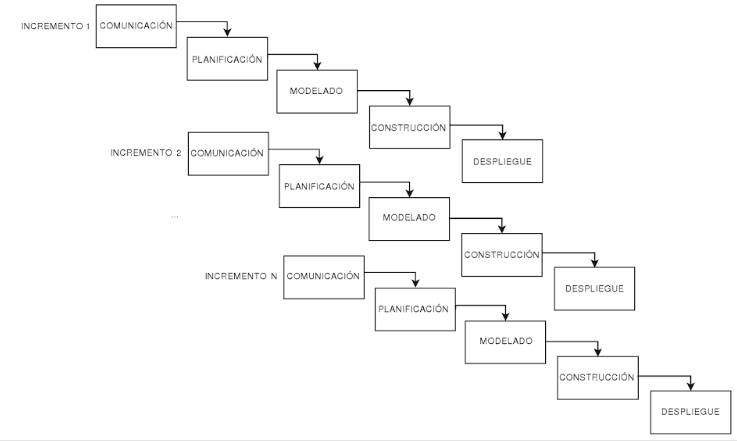


Figura 2. Ilustración del modelo incremental (Berenguel, 2016)

En el modelo incremental subdivide los requisitos del proyecto en diferentes incrementos o entregas para lo que es necesario que todos los requisitos están definidos desde el principio. El software construido en la primera etapa tendrá una funcionalidad limitada que se irá aumentando en las sucesivas entregas. (Berenguel, 2016).

# MODELOS DE PROCESOS EVOLUTIVOS

Este modelo también se lo conoce con el nombre de modelo iterativo dado que el software cambia y mejora en cada una de las iteraciones que se realizan. Se diferencia del modelo incremental porque en este se deben de tener todos los requisitos desde el principio, mientras que en el modelo iterativo prevé el descubrimiento de nuevos requisitos a lo largo del desarrollo (Berenguel, 2016).

Se suele utilizar en proyectos donde no se conoce a fondo todos los requisitos o no está clara alguna funcionalidad futura.

# MODELO DE PROCESOS ESPIRAL

# 

Este modelo mezcla los paradigmas del modelo de cascada y el modelo evolutivo u iterativo. Cada iteración dentro del espiral realiza las fases del modelo de cascada pero en cada una de las iteraciones se pueden desarrollar prototipos para identificar las necesidades del cliente e incorporar nueva funcionalidad en las iteraciones siguientes. (Berenguel, 2016).

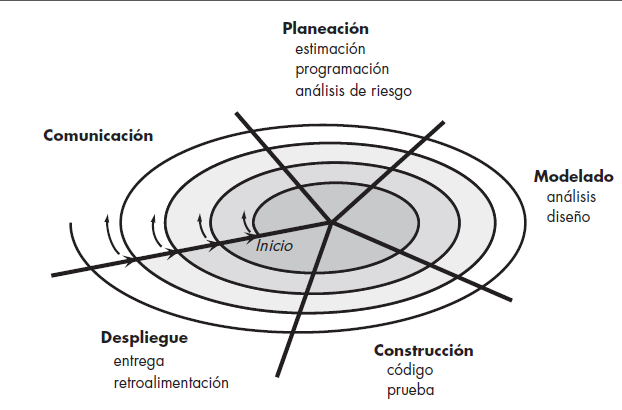


Figura 3. Representación del modelo espiral (Pressman, 2010)

A diferencia de otros modelos de procesos que finalizan cuando se entrega el software el modelo espiral puede adaptarse para aplicarse a lo largo de toda la vida del software de cómputo (Pressman, 2010).

# 

# BIBLIOGRAFÍA

Pressman, R. 2010. Ingeniería de Software un Enfoque Práctico.7 ed. McGraw-Hill Interamericana Editores, S.A. DE C.V. México. P 33- 40.

[Berenguel](https://www.google.com.ec/search?hl=es&tbo=p&tbm=bks&q=inauthor:%22JOS%C3%89+LUIS+BERENGUEL+G%C3%93MEZ%22) J. 2016. Desarrollo de aplicaciones web en el entorno servidor. Paraninfo S.A. España. P 3-6.