



**Nombre del alumno: Johanne Joaquín  
Arriaga Díaz**

**Nombre del profesor: Icel Bernardo Lepe  
Arriaga.**

**Nombre del trabajo: Ensayo de unidad I.**

**Materia: Ingeniería en Software.**

**Grado: Octavo cuatrimestre**

**Grupo: ISC13SDC0119-F**

## DEFINICIÓN Y OBJETIVOS DE LA INGENIERÍA DEL SOFTWARE

La definición de Ingeniería de Software es: “la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo operación (funcionamiento) y mantenimiento del software”. Entonces aplicado a la ingeniería en sistemas podemos tomar la palabra “ingeniería”, la cual aplicamos al software. Su utilidad radica principalmente en el poder analizar, diseñar, programar y aplicar un software de manera correcta y organizada. Gracias a esto mejoramos la calidad del software aumentamos la productividad, utilidad y hacemos más fácil el control en el proceso de desarrollo.

Pero como es de imaginarse el software crece cada día más lo cual provoca que sea más difícil de mantener organizada, y a su vez provoca que la ingeniería tenga que adaptarse a este cambio tan radical.

La diferencia principal entre software y un programa es que un programa es código ejecutable, para propósitos computacionales y el Software es una colección de códigos ejecutables de programación, asociada a las bibliotecas y a la documentación.

## CARACTERÍSTICAS Y APLICACIONES DEL SOFTWARE

Podemos mencionar muchas características aplicables al desarrollo de software pero podemos resumirlas en que el software debe tener la capacidad de mantenerse en entornos constantemente cambiantes lo cual nos lleva a las características principales que son: Modularidad, Sostenibilidad, Flexibilidad y Escalabilidad. Además resumimos “ingeniería en software” como: “una rama de las ciencias de la computación, que usa conceptos de Ingeniería para producir productos software eficientes, duraderos, escalable, y accesibles a tiempo”.

## EVOLUCIÓN HISTÓRICA DEL SOFTWARE

Llamamos “Evolución del Software” al proceso de desarrollo de un producto software usando principios y métodos de Ingeniería de software lo cual abarca el desarrollo inicial, mantenimiento y actualizaciones, hasta que el producto es desarrollado. El proceso para que esto se lleve a cabo es en primera instancia definir los requisitos, crear un prototipo inicial del software, se muestra a los consumidores, tomar los cambios sugeridos, crear actualizaciones y mantenimiento. Y un punto muy importante a tener en cuenta es que las actualizaciones deben ser progresivas ya que es más económico y viable ya que crear el software desde cero nuevamente sería inviable.

## LEYES DE EVOLUCIÓN DEL SOFTWARE

El software según Lehman esta dividió en 3 categorías las cuales son:

'S-type' ('static-type', tipo estático): Funciona estrictamente según se ha definido especificaciones y soluciones. Por ende antes de comenzar a codificar se deben tener bien definidos la solución y el método para conseguirlo.

'P-type' ('practical-type', tipo práctico): Tiene una colección de procedimientos. En este software, las especificaciones se pueden describir, pero la solución no es obvia al instante.

'E-type' ('embedded-type', tipo embebido o empotrado): Se basa en el mundo real. Por ello tiene un alto grado de evolución ya que hay varios cambios en las leyes, impuestos, etc. en las situaciones del mundo real.

## PARADIGMA DEL DESARROLLO SOFTWARE

Aquí se toman en cuenta los requisitos que ayudan a la construcción del producto software. Consiste de y su proceso es: Recoger requisitos, diseñar el Software y finalmente programar.

## PARADIGMA DE DISEÑO DE SOFTWARE

Este paradigma forma parte del desarrollo software y su proceso es: Diseñar, dar mantenimiento y programar.

## PARADIGMA DE PROGRAMACIÓN

Este paradigma incluye: Codificación, pruebas e integración.

## 1.4 PERSPECTIVA GENERAL DE LA INGENIERIA DEL SOFTWARE

Anteriormente el desarrollo de software se realizaba sin ninguna planificación. Avanzo el tiempo y se estableció el software como un producto para una distribución general. Es por ello que existe actualmente el mantenimiento del software.

### LA PROGRAMACIÓN ESTRUCTURADA

Los fundamentos surgieron por Dijkstra, quien propuso construcciones lógicas para realizar cualquier programa; la secuencia, la condición y la repetición. Esto para facilitar la legibilidad del código, las pruebas y el mantenimiento.

### PROCESOS, MÉTODOS Y HERRAMIENTAS

Método de ingeniería del software es un enfoque estructurado para el desarrollo de software esto facilita el producir software de alta calidad de una forma costeable.

Todos los métodos basados en la idea de modelos gráficos de desarrollo de un sistema y en el uso de estos modelos como un sistema de especificación o de diseño. Los métodos incluyen una variedad de componentes diferentes que son:

### MODELO CLÁSICO O LINEAL, MODELO EN CASCADA

El ciclo de vida del desarrollo Software (SDLC en sus siglas inglesas), son pasos a seguir para diseñar y desarrollar software. Y sus pasos son:

Comunicación: Aquí el usuario pide la creación de un producto software por escrito.

Recolección de solicitudes: Se intenta conseguir la máxima cantidad de información posible sobre lo que requiere el usuario, requisitos funcionales y requisitos del sistema.

Estudio de viabilidad: Se idea un plan para procesar el software cubriendo todos los requisitos del usuario y si hay alguna posibilidad de que el software ya no sea necesario. Se investiga si el proyecto es viable a nivel financiero, práctico, y a nivel tecnológico.

Análisis del sistema: En este paso los desarrolladores trazan su plan e intentan crear el mejor y más conveniente modelo de software para el proyecto.

Diseño de Software: Aquí se diseña el producto software con la ayuda de toda la información recogida sobre requisitos y análisis.

Codificación: Es la 'fase de programación'. La implementación del diseño de software empieza con el lenguaje de programación más conveniente, y desarrollando programas ejecutables y sin errores de manera eficiente.

**Pruebas:** Las pruebas de Software se hacen mientras se codifica y suelen hacerlo los desarrolladores y otros expertos evaluadores a varios niveles.

**Integración:** El Software puede necesitar estar integrado con las bibliotecas, Bases de datos o con otro u otros programas. Esta fase del SDLC se focaliza en la integración del software con las entidades del mundo exterior.

**Implementación:** Aquí se instala el software en máquinas de clientes. Se evalúa por su adaptabilidad y su portabilidad.

**Mantenimiento y Funcionamiento:** Esta fase confirma el funcionamiento del software en términos de más eficiencia y menos errores

**Disposición:** Con el paso del tiempo, puede que el software falle en su ejecución. Puede que se vuelva totalmente obsoleto o que necesite actualizaciones. De ahí surge una necesidad urgente de eliminar una parte importante del sistema. Esta fase incluye archivar datos y componentes software requerido, cierre del sistema, planificación de la actividad de disposición y terminación de sistema en el momento final del sistema.

## CONSTRUCCIÓN DE PROTOTIPOS

La construcción de modelos es vital ya que gracias a esto todo el sistema, o algunas de sus partes, se construyan rápidamente esto para comprender y aclarar aspectos para que el desarrollador, el usuario, el cliente estén de acuerdo en lo que se necesita así como también la solución que se propone para dicha necesidad y de esta forma minimizar el riesgo y la incertidumbre en el desarrollo, esto para lograr que las funciones que se agreguen con el tiempo sustituyan a las obsoletas.

## MODELOS EVOLUTIVOS

Los modelos evolutivos son iterativos. Se caracterizan por la forma en que permiten a los ingenieros del software desarrollar versiones cada vez más completas del software.

El software evoluciona con el tiempo. Los requisitos del usuario y del producto suelen cambiar conforme se desarrolla el mismo. Las fechas de mercado y la competencia hacen que no sea posible esperar a poner en el mercado un producto absolutamente completo, por lo que se aconsejable introducir una versión funcional limitada de alguna forma para aliviar las presiones competitivas.

## MODELO ESPIRAL

Es un ciclo de vida de software definido por Barry Boehm en 1988, utilizado mayormente en la ingeniería de software. Fue descrito por Boehm de la siguiente manera: "El modelo de desarrollo en espiral es un generador de modelo de proceso guiado por el riesgo que se emplea para conducir sistemas intensivos de ingeniería de software concurrente y a la vez con muchos usuarios". Las actividades que conforman este modelo forman una espiral, en la que cada bucle o interacción representa un conjunto de actividades. Se tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software.