

A faded, grayscale image of a person's face, likely a man, is visible in the background on the left side of the slide. The person has dark hair and is looking slightly downwards.

# Introducción a la Ingeniería de software

---

# Introducción a la Ingeniería de software

---

## Contenidos

- La evolución del software.
- La evolución histórica del software.
- Las principales características del software.
- Introducción al concepto de Ingeniería del Software.
- Componentes de estudio de la Ingeniería de Software.
- Referencias

# La evolución del software.

---

- La “**evolución del software**” se utiliza para denominar la dinámica de crecimiento del software.
- Se refiere a todas las actividades de programación que se orientan a generar una nueva versión de un software a partir de una versión anterior operativa.
- Se puede ver como la aplicación de las actividades y procesos de mantenimiento del software que generan una nueva versión operativa de un software con una funcionalidad de usuario o propiedades cambiadas a partir de una versión anterior.

*La guía SWEBOK considera que la causa del mantenimiento está tanto en la necesidad de “cambios” como de “evolución” en el software.*

# La evolución del software.

---

## Leyes de Lehman:

- **Cambio continuo:**

- *Un programa que se usa en un entorno real necesariamente debe cambiar o se volverá progresivamente menos útil y menos satisfactorio para el usuario.*

- **Complejidad creciente:**

- *A medida que un programa en evolución cambia, su estructura tiende a ser cada vez más compleja.*
  - *Se deben dedicar recursos extras para preservar y simplificar su estructura.*

# La evolución del software.

---

## Leyes de Lehman:

- **Autorregulación:**

- *La evolución de los programas es un proceso autorregulado.*
- *Los atributos de los sistemas, tales como tamaño, tiempo entre entregas y la cantidad de errores documentados son aproximadamente invariantes para cada entrega del sistema.*

- **Estabilidad organizacional:**

- *Durante el tiempo de vida de un programa, su velocidad de desarrollo es aproximadamente constante e independiente de los recursos dedicados al desarrollo del sistema.*

# La evolución del software.

---

## Leyes de Lehman:

### ► Conservación de la familiaridad:

- *A medida que un sistema evoluciona también todo lo que está asociado a ello, como los desarrolladores, personal de ventas, y usuarios por ejemplo, deben mantener un conocimiento total de su contenido y su comportamiento para lograr una evolución satisfactoria.*
- *Un crecimiento exagerado disminuye esta capacidad, por tanto este incremento promedio debe mantenerse.*

### ► Crecimiento continuo:

- *La funcionalidad ofrecida por los sistemas tiene que crecer continuamente para mantener la satisfacción de los usuarios.*

# La evolución del software.

---

## Leyes de Lehman:

- **Decremento de la calidad:**

- *La calidad de los sistemas software comenzará a disminuir a menos que dichos sistemas se adapten a los cambios de su entorno de funcionamiento.*

- **Retroalimentación del sistema:**

- *Los procesos de evolución incorporan sistemas de retroalimentación multi-agente y multi-ciclo y estos deben ser tratados como sistemas de retroalimentación para lograr una mejora significativa del producto.*

# La evolución histórica del software.

---

## Los primeros años:

- El software se considera un añadido.
- La programación de computadoras era *"un arte"* para el que existían pocos métodos sistemáticos.
- El desarrollo del software se realizaba virtualmente sin ninguna planificación, hasta que los planes comenzaron a descalabrarse y los costes a correr.
- Los programadores trataban de hacer las cosas bien, y con un esfuerzo heroico, a menudo salían con éxito.
- El software se diseñaba a medida para cada aplicación y tenía una distribución relativamente pequeña.
- La mayoría del software se desarrollaba y era utilizado por la misma persona u organización.



# La evolución histórica del software.

---

## Los primeros años:

- La misma persona lo escribía, lo ejecutaba y, si fallaba, lo depuraba.
- El diseño era un proceso implícito, realizado en la mente de alguien y, la documentación normalmente no existía.

## La segunda era:

- Se extienden desde la mitad de la década de los sesenta hasta finales de los setenta.
- La multiprogramación y los sistemas multiusuario introdujeron nuevos conceptos de interacción hombre - maquina.
- Se caracterizo por el establecimiento del software como producto y la llegada de las "casas del software".
- La industria, el gobierno y la academia se aprestaban a "desarrollar el mejor paquete de software" y liderar la industria.

# La evolución histórica del software.

---

## La tercera era:

- ▶ Comenzó a mediados de los años setenta y continuó más allá de una década.
- ▶ Sistemas distribuidos, múltiples computadoras, cada una ejecutando funciones concurrentes y comunicándose con alguna otra, incrementó notablemente la complejidad de los sistemas informáticos.
- ▶ Las redes de área local y de área global, las comunicaciones digitales de alto ancho de banda y la creciente demanda de acceso "instantáneo" a los datos, supusieron una fuerte presión sobre los desarrolladores del software.
- ▶ La conclusión de la tercera era se caracterizó por la llegada y amplio uso de los microprocesadores.
- ▶ El microprocesador ha producido un extenso grupo de productos inteligentes, desde automóviles hasta hornos microondas, desde robots industriales a equipos de diagnósticos.

# La evolución histórica del software.

---

## La cuarta era:

- Se aleja de las computadoras individuales y de los programas de computadoras, dirigiéndose al impacto colectivo de las computadoras y del software.
- Potentes máquinas personales controladas por sistemas operativos sofisticados, en redes globales y locales, acompañadas por aplicaciones de software avanzadas se convierten en la norma.
- La industria del software ya es la cuna de la economía del mundo.
- Las técnicas de la cuarta generación para el desarrollo del software están cambiando en la forma en que la comunidad del software construye programas informáticos.
- Las tecnologías orientadas a objetos están desplazando rápidamente los enfoques de desarrollo de software más convencionales en muchas áreas de aplicaciones.

# La evolución histórica del software.

---

## Algunos problemas persistentes:

- Los avances del software continúan dejando atrás nuestra habilidad de construir software para alcanzar el potencial del hardware.
- Nuestras habilidades y capacidades de desarrollar nuevos programas se queda corta ante la demanda.
- El uso extendido y global de dispositivos de computo ha hecho de la sociedad cada vez más dependiente de la operación fiable del software.
- Cuando el software falla, las consecuencias se pueden evidenciar en diferentes contextos incluido el económico y el social.
- Se pretende construir software con los atributos de alta calidad requeridos.
- La habilidad de soportar y mejorar los programas existentes se ve amenazada por diseños pobres y recursos inadecuados.

# Las principales características del software.

---

## Definiciones

### Software:

- Se refiere a los elementos intangible que hay en un equipo de computo.
  - *Conjunto de programas (código)*
  - *Datos que manipula y almacena.*
- Según *IEEE Software Engineering Standard: glossary of Software Engineering Terminology* el software se entiende como:

*"El conjunto de los programas de computación, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un un ordenador que realiza una función específica o un conjunto de funciones".*

# Las principales características del software.

---

## Definiciones

### Software:

- Según AECC (Asociación Española para la Calidad) el software se entiende como:

*"Programas, procedimientos, reglas y la posible documentación asociada y datos que pertenezcan a la explotación de un sistema de ordenador".*

### Hardware:

- Conjunto de componentes físicos de un equipo de computo.

*La relación de costos entre hardware y software para los equipos de computo han cambiado a lo largo del tiempo con la reducción de los costos de hardware y el aumento de los costos de desarrollo de software.*

# Las principales características del software.

---

- El software es un producto intangible.
- El software es un producto de consumo con un gran peso en la economía.
- El software se desarrolla, no se fabrica en el sentido clásico.
- Los costes del software se encuentran en la ingeniería.
- El software no se desgasta, se degrada.
- Cambios en las fases de mantenimiento
  - *No hay piezas de repuesto para el software*
- Tiene un nivel alto de obsolescencia.
- A pesar de las últimas tendencias, el software se sigue construyendo a medida.

# Las principales características del software.

---

- Los sistemas software son productos complejos:
  - *Gran funcionalidad*
  - *Objetivos diferentes y en ocasiones conflictivos*
  - *En su concepción, desarrollo y mantenimiento interviene un gran número de personas con diferentes perfiles*
  - *Elevado tamaño de sus productos*
  - *Sujeto a cambios continuos referentes a cambios en requisitos, tecnología, etc.*

***La producción de software ha de estar regida por los principios de la INGENIERÍA***

***Debemos pasar de la ARTESANÍA a la INGENIERÍA***



# Las principales características del software.

---

- Escribir software no es una tarea trivial (aunque los lenguajes de programación no faciliten la tarea)

## ***En lenguaje ensamblador x86:***

```
mov ax, num1
mov bx, num2
cmp ax, bx
jg num2MasGrande
;num1>num2
jmp final
num2MasGrande:
;num2>=num1
final:
```

## ***En Python:***

```
if num1 > num2:
    #num1 > num2
else:
    #num2 >= num1
```

# Las principales características del software.

---

- El software se desarrolla con el objetivo de cubrir las necesidades de un cliente u organización concreta.
- El desarrollo de software incluye la programación, pero se refiere al un conjunto de actividades que van mas allá de escribir código para ir desde determinada idea hasta el resultado final del software.
- Dados los costos generados y la calidad requerida de los productos de software es importante desarrollar software en un proceso de ingeniería.

*Aún existe el debate sobre si el desarrollo de software es un arte, una artesanía o una disciplina de ingeniería*

- Desarrollar **software de calidad** con el **mínimo coste posible** es una actividad, en general, **muy compleja**.

# Ámbitos del software:

---

## **Software de sistemas:**

- Son programas escritos para dar servicio a otros programas, como los sistemas operativos o los compiladores.
- Este tipo de programas suelen interactuar directamente con el hardware, de manera que sus usuarios no son los usuarios finales que usan el ordenador, sino otros programadores.

## **Software de aplicación:**

- Son programas independientes que resuelven una necesidad específica, normalmente de una organización, como por ejemplo el software de gestión de ventas de una organización concreta.
- Pueden ser desarrollados a medida o como software de propósito general.

## **Software científico y de ingeniería:**

- Muy enfocados al cálculo y a la simulación, se caracterizan por la utilización de algoritmos y modelos matemáticos complejos.

# Ámbitos del software:

---

## **Software empotrado:**

- ▶ Es el software que forma parte de un aparato, desde el control de un horno hasta el ordenador de a bordo de un automóvil.
- ▶ Se caracteriza por las limitaciones en cuanto a recursos computacionales y por estar muy adaptado al producto concreto que controla.

## **Software de líneas de productos:**

- ▶ Es software diseñado para proporcionar una capacidad específica pero orientado a una gran variedad de clientes.
- ▶ Puede estar enfocado a un mercado muy limitado (como la gestión de inventarios) o muy amplio (como una hoja de cálculo).

# Ámbitos del software:

---

## **Aplicaciones web:**

- Las aplicaciones web, independientemente de que sean un paquete o a medida, tienen una serie de características que las hacen diferentes del resto del software.
- Se caracterizan por unificar fuentes de datos y diferentes servicios en entornos altamente distribuidos.

## **Software de inteligencia artificial:**

- Estos programas usan técnicas, herramientas y algoritmos muy diferentes del resto de los sistemas y, por lo tanto, tienen una problemática propia.
- Pertenecen a esta categoría los sistemas expertos, las redes neuronales y el software de reconocimiento del habla.

# Introducción al concepto de Ingeniería del Software.

---

- Objetivos de la Ingeniería del Software:
  - *Mejorar la calidad de los productos de software.*
  - *Aumentar la productividad y trabajo de los ingenieros desarrolladores de software.*
  - *Facilitar el control del proceso de desarrollo de software.*
  - *Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.*
  - *Convertir el desarrollo de software en un proceso formal.*
  - *Generar resultados predecibles.*
  - *Obtener un producto final de alta calidad.*
  - *Satisfacer las necesidades y expectativas del cliente.*

# Introducción al concepto de Ingeniería del Software.

---

- Diferentes puntos de vista sobre el mismo tema
  - *Diseño, construcción y mantenimiento de grandes sistemas software*
  - *Construcción multipersona de software multiversión*
  - *Conjunto de técnicas que se enfrentan al software como un producto de ingeniería que requiere: planificación, análisis, diseño, implementación, pruebas y mantenimiento*
  - *Aplicación disciplinada de los principios y métodos de la ingeniería, la ciencia y las matemáticas para la producción económica del software de calidad*
  - *Conjunto de teorías, métodos y herramientas para el desarrollo profesional del software*

# Introducción al concepto de Ingeniería del Software.

---

## Definiciones:

- Ingeniería del software es el establecimiento y uso de principios sólidos de ingeniería, orientados a obtener software económico que sea fiable y trabaje de manera eficiente en máquinas reales [Bauer, 1972]
- Tratamiento sistemático de todas las fases del ciclo de vida del software . Se refiere a la aplicación de metodologías para el desarrollo del sistema software [AECC, 1986]
- La construcción de software multiversión por un equipo de varias personas [Parnas, 2011]
- La aplicación disciplinada de principios, métodos y herramientas de ingeniería, ciencia y matemáticas para la producción económica de software de calidad [Humphrey, 1989]



# Introducción al concepto de Ingeniería del Software.

---

## Definiciones:

- Disciplina tecnológica y de gestión concerniente a la invención, producción sistemática y mantenimiento de productos software de alta calidad, desarrollados a tiempo y al mínimo coste [Frakes et al., 1991]
- Aplicación de herramientas, métodos y disciplinas para producir y mantener una solución automatizada de un problema real [Blum, 1992]
- La aplicación de principios científicos para la transformación ordenada de un problema en una solución software funcional, así como en el consiguiente mantenimiento del software hasta el final de su vida útil [Davis, 1993]
- Es la aplicación de herramientas, métodos y disciplinas de forma eficiente en cuanto al coste, para producir y mantener una solución a un problema de procesamiento real automatizado parcial o totalmente por el software [Horan, 1995]

# Introducción al concepto de Ingeniería del Software.

---

## Definiciones:

- La aplicación de métodos y conocimiento científico para crear soluciones prácticas y rentables para el diseño, construcción, operación y mantenimiento del software y los productos asociados, al servicio de las personas [Shaw y Garlan, 1996]
- Según el 24765-2010 - ISO/IEC/IEEE International Standard - Systems and software engineering – Vocabulary , se define la Ingeniería del Software como:
  - *La aplicación sistemática del conocimiento científico y técnico, métodos y experiencia para el diseño, implementación, prueba y documentación del software*
  - *La aplicación de un enfoque sistemático, disciplinado y cuantificable para el desarrollo, la operación y el mantenimiento del software ; es decir, la aplicación de la Ingeniería al software [ISO/IEC/IEEE, 2010]*

# Introducción al concepto de Ingeniería del Software.

---

## Marco conceptual:

- Un sistema software es siempre una parte de un sistema mayor que lo engloba como componente
- La Ingeniería del Software será solamente una parte del diseño del sistema
  - *Los requisitos del software han de ajustarse a los requisitos del resto de los elementos que constituyen este sistema*
  - *El ingeniero del software ha de estar implicado en el desarrollo de los requisitos del sistema completo, comprendiendo el dominio de actividad en su totalidad*
- La Teoría General de Sistemas es el antecedente conceptual en el que se apoya la teoría sobre los Sistemas de Información a los que la Ingeniería del Software intenta aportar soluciones

# Introducción al concepto de Ingeniería del Software.

---

## Marco conceptual:

- ▶ **Proceso:** Define el marco de trabajo y permite un desarrollo racional y oportuno de la Ingeniería del Software
- ▶ **Método:** Indica cómo construir técnicamente el software . Se incluyen técnicas de modelado y otras técnicas descriptivas
- ▶ **Herramientas:** Proporcionan el soporte automático o semiautomático para el proceso y para los métodos
- ▶ **Notación:** Conjunto de reglas gráficas o textuales para la representación de un modelo.
- ▶ **Metodología:** Colección de métodos para resolver un tipo de problemas, descompone el proceso de desarrollo en actividades y proporciona los métodos adecuados para llevar a cabo dichas actividades

# Introducción al concepto de Ingeniería del Software.

---

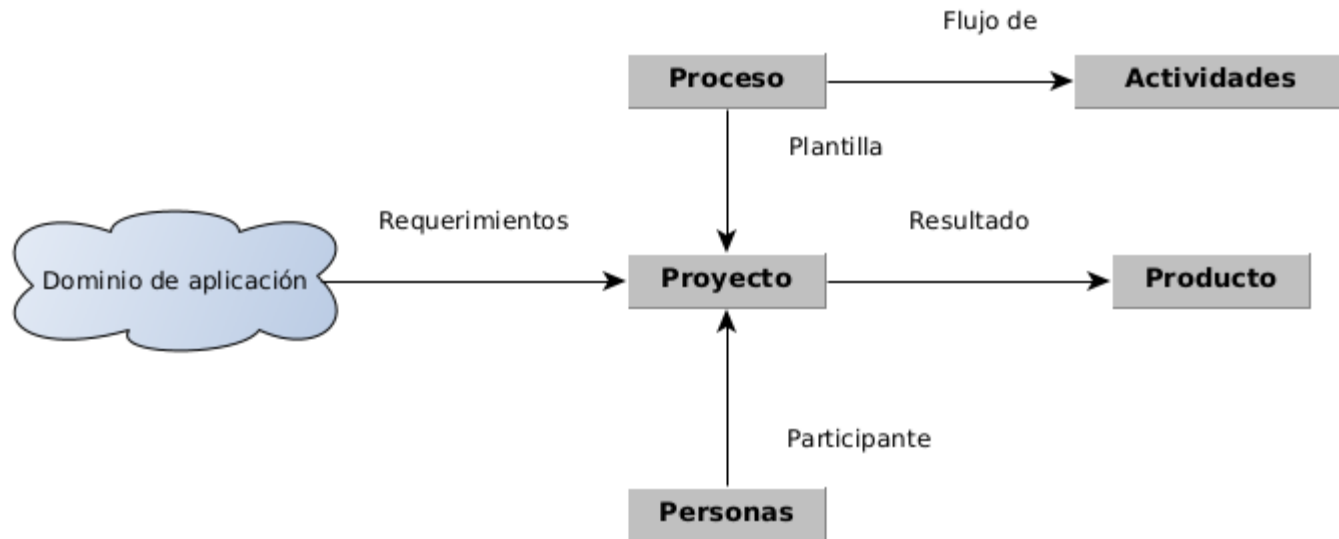
Marco conceptual:



# Introducción al concepto de Ingeniería del Software.

---

## Proceso de desarrollo de software:



# Introducción al concepto de Ingeniería del Software.

---

## Proceso de desarrollo de software:

*"El fundamento de la Ingeniería del Software es la capa proceso. El proceso de la Ingeniería del Software es la unión que mantiene juntas las capas de tecnología y que permite un desarrollo racional y oportuno de la Ingeniería del Software"*  
Pressman

*"Un proceso bien definido es necesario para desarrollar sistemas software de manera repetible y predecible"* G. Booch

*"Permite un negocio sostenible y que puede mejorar en cada nuevo proyecto, incrementando la eficiencia y productividad de la organización"* G. Booch

# Introducción al concepto de Ingeniería del Software.

---

## Disciplinas relacionadas:

- › Ingeniería de computación.
- › Ciencias de la computación.
- › Gestión.
- › Matemáticas.
- › Gestión de proyectos.
- › Gestión de calidad.
- › Ingeniería de Sistemas.



# Introducción al concepto de Ingeniería del Software.

---

## Ingeniería de software vs. Ingeniería de sistemas:

### Ingeniería de sistemas:

- La ingeniería de sistemas se dedica a implementar y mejorar sistemas que ayudan al ser humano a estudiar la realidad; está integrada a otras disciplinas y especialidades.
- Es una aplicación tecnológica. Se considera una de las ramas de la ingeniería que estudia todos los sistemas existentes y por ello no se limita a la informática.
- Aplica ciencias físicas y matemáticas para economizar recursos y materiales en pro de la humanidad.
- La ingeniería de sistemas se dedica a implementar y mejorar sistemas que ayudan al ser humano a estudiar la realidad; está integrada a otras disciplinas y especialidades.

# Introducción al concepto de Ingeniería del Software.

---

## Ingeniería de software vs. Ingeniería de sistemas:

### **Ingeniería de sistemas:**

- Es una aplicación tecnológica. Se considera una de las ramas de la ingeniería que estudia todos los sistemas existentes y por ello no se limita a la informática.
- Aplica ciencias físicas y matemáticas para economizar recursos y materiales en pro de la humanidad.

# Introducción al concepto de Ingeniería del Software.

---

## Ingeniería de software vs. Ingeniería de sistemas:

### Ingeniería de sistemas:

- No construye ningún tipo de producto tangible.
- Es una actividad interdisciplinaria, la cual reúne personas con conocimientos, para conocer la realidad.
- Comprende todo el desarrollo de los sistemas informáticos.
- Trabaja con diferentes ramas y sistemas.

### Ingeniería de software:

- Se apoya en sistemas abstractos a través de metodologías para diseñar y entregar productos tangibles.
- Es una disciplina que comprende los aspectos de producción del software.
- Es parte del proceso de la ingeniería de sistemas.
- Sólo trabaja con sistemas de computación e informática.

# Componentes de estudio de la Ingeniería de Software.

---

El Software Engineering Body of Knowledge (SWEBOK) define 10 áreas de conocimiento de Ingeniería de Software:

- Requerimientos.
- Diseño.
- Construcción.
- Pruebas.
- Calidad.
- Mantenimiento.
- Administración de la configuración.
- Administración de proyectos.
- Procesos.
- Herramientas y métodos.

# Componentes de estudio de la Ingeniería de Software.

---

## Requerimientos de Software:

- Los requerimientos de software expresan las necesidades y restricciones que debe satisfacer un producto para contribuir a la solución de un problema real.
- Esta área de conocimiento considera la obtención, análisis, especificación y validación de los requerimientos, así como el rol que juegan dentro del proceso de desarrollo de software.
- Un especialista en requerimientos tiene conocimiento y experiencia en técnicas para obtener, cuantificar, negociar, clasificar, priorizar, modelar, documentar y validar los requerimientos de software.
- Además debe saber administrar adecuadamente los cambios a éstos.

# Componentes de estudio de la Ingeniería de Software.

---

## Requerimientos de Software:

- › Este componente incluye:
  - › *Fundamentos de los requerimientos del software.*
  - › *Proceso de los requerimientos.*
  - › *Captura de los requerimientos*
  - › *Análisis de los requerimientos.*
  - › *Especificación de los requerimientos.*
  - › *Validación de los requerimientos.*
  - › *Consideraciones prácticas.*

# Componentes de estudio de la Ingeniería de Software.

---

## Diseño de Software:

- El diseño juega un rol clave en el desarrollo de software ya que es donde se generan modelos que sirven como “planos” para la construcción.
  - **Diseño arquitectónico:** *Describe la estructura y organización de alto nivel de un sistema. Identifica los componentes e interfaces entre éstos.*
  - **Diseño detallado:** *Describe individualmente cada componente con suficiente detalle para ser construido.*
- El diseño de software requiere entender a fondo principios como la abstracción, acoplamiento, cohesión, descomposición y encapsulación, ya que son la base para diseñar sistemas robustos.
- Es necesario saber resolver aspectos prácticos como la persistencia de datos, sistemas distribuidos, peticiones concurrentes, manejo de eventos, recuperación a fallas, etc.

# Componentes de estudio de la Ingeniería de Software.

---

## Diseño de Software:

- ▶ Un diseñador que no quiera “reinventar la rueda” cada vez que se le presente un problema, debe estar familiarizado con “patrones”, soluciones documentadas a problemas comunes.
- ▶ Este componente incluye:
  - ▶ *Fundamentos de los diseños de software.*
  - ▶ *Cuestiones claves en el diseño de software.*
  - ▶ *Estructura y arquitectura de software.*
  - ▶ *Análisis y evaluación de la calidad del diseño de software.*
  - ▶ *Notaciones del diseño de software.*
  - ▶ *Estrategias y métodos del diseño de software.*



# Componentes de estudio de la Ingeniería de Software.

---

## Construcción de Software:

- Esta área de conocimiento se refiere a la creación de software útil a través de la programación, depuración (debugging), pruebas unitarias e integración de componentes.
- La construcción lidia con la creación y aplicación de algoritmos para la resolución de problemas, así como su implementación utilizando algún lenguaje de programación.
- Todo esto se debe hacer buscando minimizar la complejidad y cumpliendo estándares para que el código generado sea entendible y extensible por otros, además de que esté optimizado para consumir la menor cantidad de recursos posible.
- Este componente incluye:
  - *Fundamentos de la construcción de software.*
  - *Gestión de la construcción de software.*
  - *Consideraciones prácticas.*

# Componentes de estudio de la Ingeniería de Software.

---

## Pruebas de Software:

- Las pruebas de software consisten en la verificación dinámica del comportamiento real de un programa comparado con su comportamiento esperado en un conjunto finito de casos de prueba seleccionados de un dominio de ejecuciones típicamente infinito.
- Las pruebas se realizan para evaluar la calidad de un producto a través de la detección de fallas en éste.
- Las pruebas de software han evolucionado para dejar de ser consideradas como algo que comienza sólo hasta que la programación termina, con el limitado propósito de detectar fallas.
- Es aceptado que las pruebas deben abarcar el proceso completo de desarrollo, que su planeación comienza durante las primeras etapas del proceso de requerimientos, y que los planes y procedimientos de prueba se deben desarrollar y refinar durante el ciclo completo de desarrollo.

# Componentes de estudio de la Ingeniería de Software.

---

## Pruebas de Software:

- Las pruebas pueden ser de diferentes tipos, ya sea por su alcance (unitarias, integrales, de sistema) o su objetivo (funcionalidad, confiabilidad, desempeño, regresión, aceptación, beta, etc.).
- Se utilizan diferentes técnicas como tablas de decisión, análisis de fronteras, máquinas de estados, y la experiencia misma.
- Para cuantificar la calidad de un producto de software es necesario entender métricas como la densidad de defectos y la evaluación de confiabilidad.
- Este componente incluye:
  - *Fundamentos de las pruebas de software.*
  - *Niveles de prueba.*
  - *Técnicas de prueba.*
  - *Medidas de prueba.*
  - *Proceso de pruebas.*

# Componentes de estudio de la Ingeniería de Software.

---

## Calidad del Software:

- El área de conocimiento de calidad se enfoca en la aplicación de técnicas estáticas para evaluar y mejorar la calidad del software. Esto difiere del acercamiento utilizado en pruebas, donde las técnicas utilizadas son dinámicas, ya que requieren la ejecución del software.
- El área de conocimiento de calidad involucra los subprocesos de aseguramiento de calidad, verificación, validación, revisión y auditoría.
- Considera tópicos como la clasificación de defectos, control estadístico de calidad, modelos de predicción y análisis de tendencias.
- Este componente incluye:
  - *Fundamentos de calidad de software.*
  - *Consideraciones prácticas.*
  - *Procesos de gestión de la calidad de software.*

# Componentes de estudio de la Ingeniería de Software.

---

## Mantenimiento de Software:

- El mantenimiento se refiere a las modificaciones a un producto de software previamente liberado para prevenir fallas (preventivo), corregirlas (correctivo), mejorar su desempeño (perfectivo) o adaptarlo a cambios en el ambiente (adaptativo).
- Históricamente no se le ha dado tanta atención al mantenimiento como al desarrollo de software. Esto está cambiando debido a que las empresas buscan sacar el mayor jugo posible a sus productos existentes.
- Algunos temas clave en el mantenimiento son la reingeniería, análisis de impacto, pruebas de regresión, y outsourcing del mantenimiento.
- Hay que tener en cuenta que el proceso a seguir en el mantenimiento es diferente al que se sigue para el desarrollo, por lo que involucra actividades y artefactos diferentes.

# Componentes de estudio de la Ingeniería de Software.

---

## Mantenimiento de Software:

- › Este componente incluye:
  - › *Fundamentos del mantenimiento de software:*
  - › *Problemas clave en el mantenimiento de software.*
  - › *Proceso de mantenimiento.*
  - › *Técnicas de mantenimiento.*

# Componentes de estudio de la Ingeniería de Software.

---

## Administración de la Configuración del Software (SCM):

- La configuración de un sistema se refiere al conjunto de elementos de hardware y software que lo forman.
- SCM es la disciplina de identificar la configuración en distintos puntos del tiempo con el propósito de controlar los cambios a ésta, manteniendo su integridad y rastreabilidad durante el ciclo completo de vida del software.
- SCM va más allá del simple control de versiones, y requiere saber identificar los elementos de configuración, definir un proceso de control de cambios, auditar y reportar el estatus de la configuración, y administrar la integración y liberación del sistema completo.

# Componentes de estudio de la Ingeniería de Software.

---

## Administración de la Configuración del Software (SCM):

- Este componente incluye:
  - *Gestión del proceso de configuración de software.*
  - *Identificación de la configuración.*
  - *Control de configuración de software.*
  - *Registro del estado de la configuración.*
  - *Auditoría de configuración de software.*
  - *Gestión de lanzamiento y entrega.*



# Componentes de estudio de la Ingeniería de Software.

---

## Administración de la Ingeniería del Software:

- Esta área de conocimiento llamada también Administración de Proyectos o Project Management.
- Consiste en la aplicación de actividades administrativas —como la planeación, coordinación, medición, monitoreo, control y reporte— para asegurar que el desarrollo y mantenimiento de software se lleva a cabo de manera sistemática, disciplinada y cuantificable.
- Los tópicos más importantes de esta área son la planeación de proyectos, estimación de esfuerzo, asignación de recursos, administración de riesgo, manejo de proveedores, manejo de métricas, evaluación, y cierre de proyectos.

# Componentes de estudio de la Ingeniería de Software.

---

## Administración de la Ingeniería del Software:

- Este componente incluye:
  - *Iniciación y alcance.*
  - *Planificación de un proyecto de software.*
  - *Promulgación del proyecto de software.*
  - *Revisión y evaluación.*
  - *Cierre.*
  - *Medidas de la Ingeniería de Software.*

# Componentes de estudio de la Ingeniería de Software.

---

## Proceso de Ingeniería de Software:

- ▶ Cada área de conocimiento considera un proceso para las actividades técnicas y administrativas que deben realizarse para adquirir, desarrollar, mantener y retirar software; éste es considerado como un primer nivel de procesos.
- ▶ Adicionalmente existe un segundo nivel, o meta-nivel, que se enfoca en la definición, implantación, evaluación, mejora y administración del cambio de los procesos de primer nivel. A éste es al que se refiere el área de conocimiento de proceso de ingeniería de software.
- ▶ Este componente incluye:
  - ▶ *Proceso de implementación y cambios.*
  - ▶ *Definición de procesos.*
  - ▶ *Valoración del proceso.*
  - ▶ *Medidas de productos y procesos.*

# Componentes de estudio de la Ingeniería de Software.

---

## Herramientas y Métodos de Ingeniería de Software:

- Las herramientas permiten la automatización de tareas repetitivas y bien definidas, habilitando al ingeniero de software para que se concentre en los aspectos creativos del proceso.
- Existen una gran cantidad de herramientas para asistir todas las áreas de conocimiento, desde la administración de requerimientos hasta las pruebas automatizadas.
- Los métodos de ingeniería de software establecen una estructura para sistematizar las actividades con el objetivo de aumentar las posibilidades de éxito.
- Esta área de conocimiento se enfoca en los métodos que abarcan múltiples KAs, ya que los que son relativos a una sola área de conocimiento se incluyen en el área correspondiente.
- Los métodos pueden aplicar técnicas heurísticas (informales), formales, y basadas en prototipos.

# Componentes de estudio de la Ingeniería de Software.

---

## Herramientas y Métodos de Ingeniería de Software:

- ▶ Este componente incluye para las herramientas de la Ingeniería de Software:
  - ▶ *Requerimientos de las herramientas.*
  - ▶ *Herramientas de diseño.*
  - ▶ *Herramientas de construcción.*
  - ▶ *Herramientas de pruebas.*
  - ▶ *Herramientas de mantenimiento.*
  - ▶ *Herramientas de gestión de configuración.*
  - ▶ *Herramientas de gestión de ingeniería.*
  - ▶ *Herramientas de proceso.*
  - ▶ *Herramientas de calidad de software.*
  - ▶ *Cuestiones de herramientas compuestas.*

# Componentes de estudio de la Ingeniería de Software.

---

## Herramientas y Métodos de Ingeniería de Software:

- Este componente incluye para los métodos de la Ingeniería de Software:
  - *Métodos heurísticos.*
  - *Métodos formales.*
  - *Métodos de prototipado.*

# Referencias

---

- Sicilia, M., Conceptos del Mantenimiento del Software, tomado de: <https://cnx.org/contents/3ak2yyXP@6.1:a6kc6QIh@8/Mantenimiento-del-Software-como-Actividad-de-Ingenier%C3%ADa>
- Chapin, N., Hale, J.E., Khan, K.M., Ramil, J. and Tan, W. (2001) Types of software evolution and software maintenance. Journal of Software Maintenance and Evolution: research and practice, 13, pp. 3-30.
- Lehman, M. M. (1980). «On Understanding Laws, Evolution, and Conservation in the Large-Program Life Cycle». Journal of Systems and Software 1: 213-221. doi:10.1016/0164-1212(79)90022-0
- SWEBOK, [www.swebok.org](http://www.swebok.org)
- Pressman, R. (2010). Ingeniería del software (7ª ed.). México: McGrawHill.