

Ingeniería de *Software*

Guillermo Monroy Rodríguez

Doctor en Ciencias en Computación • gmonroyr@gmail.com

Formación académica y Experiencia profesional del profesor

- Formación académica:
 - ▶ Doctor en Ciencias en Computación por el CINVESTAV, Zacatenco, CDMX
 - ▶ Maestro en Ciencias en Computación por el CINVESTAV, Zacatenco, CDMX
 - ▶ Ingeniero en Computación por la UAM - Azcapotzalco, CDMX
- Experiencia Profesional:
 - ▶ Arquitecto y Desarrollador de *software*
 - ▶ Soporte de aplicaciones para el sector bancario y financiero
 - ▶ Administración de servidores y configuración de ambientes de desarrollo, calidad y producción
- Investigación científica:
 - ▶ Computación ubicua y *Wearable Computing*
 - ▶ *Pervasive healthcare*
 - ▶ Ingeniería de *software*
 - ▶ Compañeros digitales para *healthcare*
 - ▶ Diseño de la interacción (*IxD*)

Objetivo General: Que el alumno sea capaz de utilizar mejores prácticas básicas de ingeniería de *software* en el desarrollo de pequeños sistemas intensivos de *software*

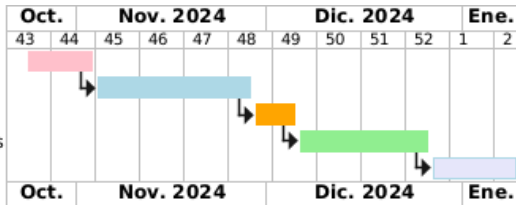
Objetivos Específicos: Que al final de la UEA el alumno sea capaz de:

- ▶ Comprender las características principales del desarrollo de los sistemas intensivos en *software*
- ▶ Conocer los procesos para desarrollar sistemas intensivos de *software*
- ▶ Realizar la administración de proyectos de desarrollo de sistemas intensivos en *software* de tamaño pequeño
- ▶ Realizar el análisis de requerimientos de sistemas intensivos en *software*
- ▶ Realizar la administración de configuración en proyectos de desarrollo de sistemas intensivos en *software*

1. Procesos de desarrollo de software (1.5 semanas)
2. Administración de proyectos (3.5 semanas)
3. Administración de riesgos (1 semana)
4. Administración de requerimientos (3 semanas)
5. Administración de configuración (2 semanas)

Temario para Ingeniería de *Software*

- T1. Proceso de desarrollo de SW
- T2. Administración de proyectos
- T3. Administración de riesgos
- T4. Administración de requerimientos
- T5. Administración de configuración



Aspectos a evaluar:

Cualitativos. Elementos que se deben cumplir para tener derecho a aprobar el curso:

- ▶ Asistencia a las sesiones | 80% de asistencia mínima
- ▶ Participación acertiva en las sesiones | ≥ 1 participación acertiva como mínimo por parcial

Cuantitativos. Elementos que se deben cumplir para aprobar el curso. Se deberán aprobar todos los rubros para que se pueda aplicar el promedio.

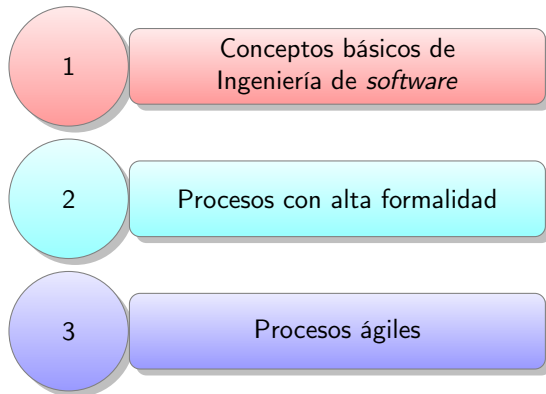
- ▶ Tareas y ejercicios en sesión | 15 %, el número será definido en el transcurso de las sesiones
- ▶ Exposición | 15 %, se hará una exposición como mínimo
- ▶ Exámenes parciales | 30 % , serán como mínimo 3
- ▶ Proyecto: Desarrollo de un sistema de *software* | 40 %

Escala de calificaciones

MB	[8.6 , 10.0]
B	[7.6 , 8.6)
S	[6.0 , 7.6)
NA	[0.0 , 6.0)

Sesión I

Procesos de desarrollo de *software*



Tema 1

Conceptos básicos de Ingeniería de *software*

Conceptos básicos de Ingeniería de *software*

¿Qué es el *software*?

1. Programas de cómputo y su documentación asociada: requerimientos, modelos de diseño y manuales de usuario
2. El software puede ser desarrollado para un cliente en particular o para un mercado general
3. El software puede ser:
 - ▶ Genérico: desarrollado para venderse a múltiples clientes (apps para ofimática como microsoft 365, sistemas operativos, apps móviles, etc.)
 - ▶ A la medida: desarrollado bajo demanda del cliente a un desarrollador específico
4. El software nuevo puede ser creado desarrollando nuevos programas, configurando sistemas de software genérico o reutilizando software existente
5. El buen software debe entregar al usuario la funcionalidad y el desempeño requeridos, y debe ser sustentable, confiable y utilizable

Conceptos básicos de Ingeniería de *software*

Definiciones de *Software*

Pressman

El software es: 1) instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados; 2) estructuras de datos que permiten que los programas manipulen en forma adecuada la información, y 3) información descriptiva tanto en papel como en formas virtuales que describen la operación y uso de los programas.

Sommerville

Programas de cómputo y documentación asociada. Los productos de software se desarrollan para un cliente en particular o para un mercado en general.

Conceptos básicos de Ingeniería de *software*

Clasificación de *software*

- *Software* de sistemas
- *Software* de aplicación
- *Software* de ingeniería y ciencias
- *Software* incrustado
- *Software* de línea de productos
- Aplicaciones web o móviles
- *Software* de inteligencia artificial

Tarea para casa

Realice un resumen, escrito a mano, donde describa cada tipo de sistema de la taxonomía presentada y aporte un ejemplo para cada uno.

Conceptos básicos de Ingeniería de *software*

Software heredado

Programas antiguos, que es frecuente denominar software heredado, han sido centro de atención y preocupación continuas desde la década de 1960

Sin embargo, conforme pase el tiempo será frecuente que los sistemas de software evolucionen por una o varias de las siguientes razones:

- El software debe adaptarse para que cumpla las necesidades de los nuevos ambientes del cómputo y de la tecnología
- El software debe ser mejorado para implementar nuevos requerimientos del negocio
- El software debe ampliarse para que sea operable con otros sistemas o bases de datos modernos
- La arquitectura del software debe rediseñarse para hacerla viable dentro de un ambiente de redes

Conceptos básicos de Ingeniería de *software*

¿Qué es la Ingeniería de *software*?

- Una disciplina de la **Ingeniería** que concierne a todos los aspectos de la producción de *software*
- Los Ingenieros de *Software* deben:
 - ▶ Adoptar un enfoque sistemático para llevar a cabo su trabajo
 - ▶ Utilizar las herramientas y técnicas apropiadas para resolver el problema planteado, de acuerdo a las restricciones de desarrollo y a los recursos disponibles

Conceptos básicos de Ingeniería de *software*

Definiciones de Ingeniería de *software*

Fritz Bauer, 69

La *ingeniería de software* es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica *software* que sea confiable y que trabaje con eficiencia en máquinas reales.

IEEE, 1993

La *ingeniería de software* es: 1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la *ingeniería de software*. 2) El estudio de enfoques según el punto 1.

Sommerville, 2011

La *ingeniería de software* es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción de *software*.

Conceptos básicos de Ingeniería de *software*

Ingeniería de *software*

- La *ingeniería de software* es un enfoque sistemático para la producción de software que toma en cuenta los temas prácticos de costo, fecha y confiabilidad, así como las necesidades de clientes y fabricantes de software
- Como este enfoque sistemático realmente implementado varía de manera drástica dependiendo de la organización que desarrolla el software, el tipo de software y los individuos que intervienen en el proceso de desarrollo, no existen métodos y técnicas universales de ingeniería de software que sean adecuados para todos los sistemas y las compañías

¿Cuál es la diferencia entre Ingeniería de Software y Ciencias Computacionales?

- Las Ciencias Computacionales conciernen a la teoría y fundamentos de cualquier sistema de cómputo (hardware o software)
- La *ingeniería de software* concierne los aspectos prácticos del desarrollo y puesta en marcha de productos útiles de software
- La *ingeniería de software* aún está lejos de ser considerada una ciencia como lo son la química o la física

¿Cuál es la diferencia entre Ingeniería de Software y Ciencias Computacionales?

- La **Ingeniería de sistemas** concierne a todos los aspectos del desarrollo de sistemas basados en cómputo incluyendo hardware, software y la ingeniería de procesos
- La **ingeniería de software** es una parte de este proceso que comprende el desarrollo de software, control, aplicaciones y bases de datos del sistema
- Los **ingenieros de software** son los encargados de la especificación del sistema, del diseño de la arquitectura, la integración y la puesta en marcha

Conceptos clave de Ingeniería de software

¿Cuáles son las actividades fundamentales de la ingeniería de software?

Especificación, desarrollo, validación y evolución del software.

¿Cuál es la diferencia entre ingeniería de software y ciencias de la computación?

Las ciencias de la computación se enfocan en teoría y fundamentos; mientras la ingeniería de software se enfoca en el sentido práctico del desarrollo y en la distribución de software.

¿Cuál es la diferencia entre ingeniería de software e ingeniería de sistemas?

La ingeniería de sistemas se interesa por todos los aspectos del desarrollo de sistemas basados en computadoras, incluidos hardware, software e ingeniería de procesos. La ingeniería de software es parte de este proceso más general.

¿Cuáles son los principales retos que enfrenta la ingeniería de software?

Se enfrentan con una diversidad creciente, demandas por tiempos de distribución limitados y desarrollo de software confiable.

¿Cuáles son los costos de la ingeniería de software?

Aproximadamente 60% de los costos del software son de desarrollo, y 40% de prueba. Para el software elaborado específicamente, los costos de evolución superan con frecuencia los costos de desarrollo.

¿Cuáles son los mejores métodos y técnicas de la ingeniería de software?

Aun cuando todos los proyectos de software deben gestionarse y desarrollarse de manera profesional, existen diferentes técnicas que son adecuadas para distintos tipos de sistema. Por ejemplo, los juegos siempre deben diseñarse usando una serie de prototipos, mientras que los sistemas críticos de control de seguridad requieren de una especificación completa y analizable para su desarrollo. Por lo tanto, no puede decirse que un método sea mejor que otro.

Ejercicio: Diseño de un sistema para gestión de música

Como parte de un sistema que gestiona medios audiovisuales se desea realizar un componente autónomo que funcione como un servicio, dicho componente deberá ser capaz de generar un índice de recursos musicales a partir de un sistema de archivos que contiene estos recursos.

- El índice se realizará a partir de la especificación de los meta-datos de archivos mp3 que contienen pistas de audio musicales, especificado en el estándar ID3 v2.4.
- Los archivos musicales a indizar estarán en formato mp3. Los archivos deberán estar contenidos en un sistema de archivos (carpeta) sin importar su orden o nombre de archivo
- La información indizada deberá persistir en un modelo de datos en su forma de base de datos. Podrá emplear MySQL o MariaDB.
- El servicio deberá proporcionar un mecanismo de consulta para encontrar cualquier archivo o colección basado en los valores de alguno de sus atributos. Por ejemplo, buscar todos los archivos del año 1980, buscar archivos del autor Alfredo Jiménez, buscar los archivos del género Folk o todas las posibles combinaciones