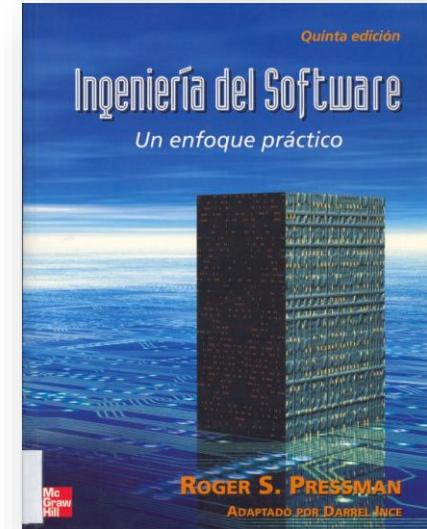


INGENIERÍA DE SOFTWARE I



⊕ ¡Presentándonos!

- ⊕ Nombre(s) y Apellidos
- ⊕ Dirección particular
- ⊕ ¿Convivencia familiar?
- ⊕ ¿Qué hace en su tiempo libre?
- ⊕ ¿Qué expectativas tiene con la asignatura?



ORGANIZACIÓN DE LA ASIGNATURA

- **Tema 1:** Proceso de Software (SW).
- **Tema 2:** Ingeniería de Requisitos.
- **Tema 3:** Estimación y Planificación de SW.
- **Tema 4:** Análisis y Modelado.



FORMAS DE EVALUACIÓN DE LA ASIGNATURA



- Evaluaciones en clases (Orales y escritas).
- Seminario 1 (Semana 2).
- Seminarios 2 y 3 (Semana 11).
- Prueba Parcial (Semana 10).
- Cortes del Trabajo de Curso (Semana 5 y 14).
- Defensa del trabajo de curso (Semana 16).





Actividad 1.

Tema 1. Proceso de Software.

Conferencia 1. Introducción a la
Ingeniería de Software.

OBJETIVO

- Caracterizar la Ingeniería de Software (IS) a partir de los elementos que la conforman.



CONTENIDOS



- + Conceptos y características del Software.
- + Tipos de SW o dominios de aplicación.
- + Conceptos de IS.
- + Las 4 “P” y las capas de la IS.
- + Fundamentos del proceso de desarrollo de software.
- + Marco de trabajo común. Actividades genéricas y sombrilla.

BIBLIOGRAFÍA

- **Pressman, Roger S.; 2010.** “Ingeniería de software. Un enfoque práctico”. 7ma Edición. Parte I. Prólogo de los Capítulos 1 y 2. Páginas 25-26.

- **Pressman, Roger S.; 2010.** “Ingeniería de software. Un enfoque práctico”. 7ma Edición. Parte I. Capítulos 1 y 2. Páginas 25-52.

- **Sommerville, I.; 2011.** “Ingeniería de Software”, 9na Edición. Parte I. Capítulo 2 Epígrafes 2.1-2.4 Pág. 27-56.

¿Qué se define como Software?



- + Conjunto de **programas, instrucciones y reglas informáticas** para ejecutar ciertas tareas en una computadora.



[Diccionario RAE]



El software se forma con:

- Las **instrucciones** (programas de computadora) que al ejecutarse proporcionan las características, funciones y el grado de desempeño deseado.
- Las **estructuras de datos** que permiten que los programas manipulen información de manera adecuada.
- Los **documentos** que describen la operación y el uso de programas.

CARACTERÍSTICAS DEL SOFTWARE



TIPOS DE SW O DOMINIOS DE APLICACIÓN

Líneas de producto

Científico y de Ingeniería

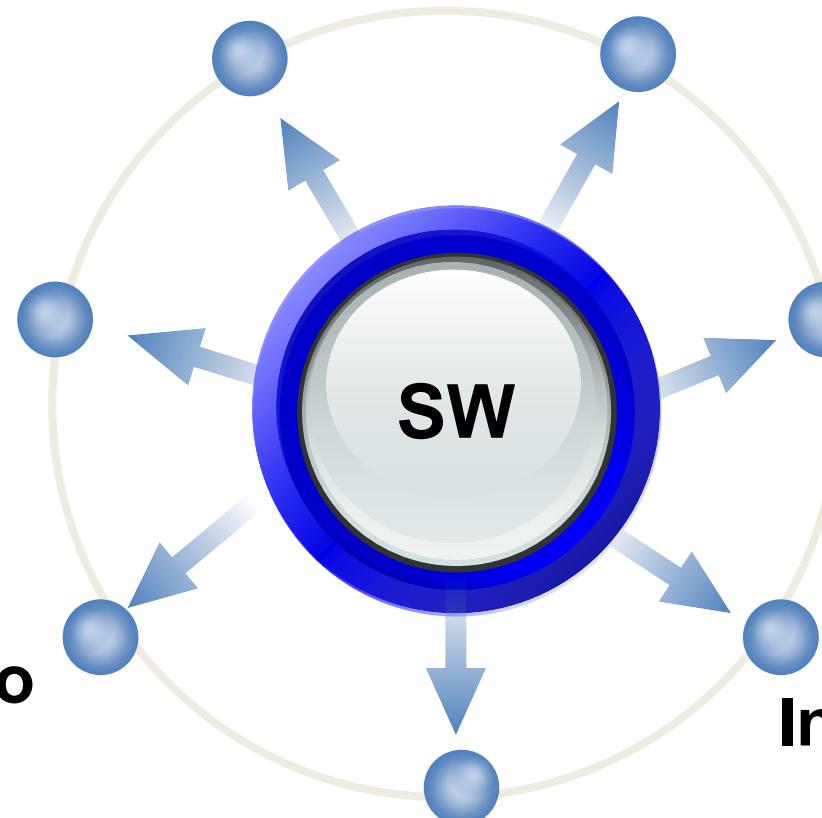
Sistemas

Aplicación

Empotrado

Inteligencia Artificial

Aplicaciones basadas en la web



SOFTWARE DE SISTEMA

- + Colección de **programas** escritos para **servir** a otros **programas**.
- + Procesan **información** determinada: compiladores, editores.
- + Procesan datos indeterminados: componentes de un sistema operativo, controladores, SW de red.



SOFTWARE DE APLICACIÓN

- Programas **independientes** que resuelven una **necesidad** de negocio **específica**.
- Procesan datos **empresariales** o **técnicos** de forma que **facilitan** las operaciones de **negocios** o toma de decisiones técnicas o de gestión.
- **Funciones** de negocio en **tiempo real** (Ej: procesamiento de transacciones en los puntos de venta).

SOFTWARE CIENTÍFICO Y DE INGENIERÍA

- + **Caracterizado por algoritmos "devoradores de números".**

- + Abarca la astronomía, la vulcanología, el análisis de la tensión automotriz, la dinámica orbital de los transbordadores espaciales, la biología molecular, etc.



SOFTWARE EMPOTRADO

- + Reside **dentro** de la **memoria** de sólo **lectura** del sistema.
- + Se **implementan**, **controlan** características y **funciones** para el **usuario final** y el sistema mismo.
- + **Proporcionar** capacidades de **control** y funcionamiento significativo: funciones digitales de un automóvil, control del combustible.



SOFTWARE DE LÍNEAS DE PRODUCTO

- Diseñado para **proporcionar** una capacidad específica y la **utilización** de muchos **clientes diferentes**.
- Se puede **enfocar** en un **nicho** de **mercado** limitado (productos para control de inventarios).



SOFTWARE DE INTELIGENCIA ARTIFICIAL

- + Utiliza **algoritmos no numéricos** en la resolución de problemas complejos que es imposible abordar por medio de un análisis directo.
- + Aplicaciones: la **robótica, sistemas expertos, reconocimiento de patrones (imagen y voz), juegos en la computadora.**



APLICACIONES BASADAS EN LA WEB

- Conjunto de **archivos de hipertextos** ligados que presenta información mediante texto y algunas gráficas.
- Integradas a **bases datos corporativas** y aplicaciones de negocio.
- Engloban un espectro amplio de aplicaciones.



ATRIBUTOS ESENCIALES DEL SOFTWARE

- **Mantenibilidad:** El software debe escribirse de forma tal que pueda evolucionar para cumplir las necesidades de cambios de los clientes. Atributo crítico porque es inevitable.
- **Confiabilidad:** El software confiable no debe causar daños físicos o económicos en caso de fallas (fiabilidad, protección y seguridad).
- **Eficiencia:** No debe existir un desgaste de recursos del sistema (memoria, procesamiento). Tiempos de respuestas mínimos.
- **Usabilidad:** Fácil de usar sin esfuerzo adicional por el usuario. Interfaz de usuario y documentación adecuada.

[Somerville, Capítulo I]

MITOS DEL SOFTWARE

- Creencias acerca del software y de los procesos empleados para construirlo.



MITOS DE LA ADMINISTRACIÓN

*Bajo presión:
presupuestos,
itinerarios y
mejorar la calidad.*



MITOS Y REALIDADES

- **Mito:** Ya se tiene un libro lleno de **estándares** y procedimientos para la **construcción** de SW. ¿Esto **proporcionará** a mi gente todo el **conocimiento** necesario?

- **Realidad:** ¿Se usa? ¿Los encargados de la construcción del SW saben de su existencia? ¿El libro refleja la práctica moderna de la ISW? ¿Es completo? ¿Es adaptable? ¿Está dirigido al mejoramiento del tiempo de entrega sin dejar de enfocarse en la calidad?

MITOS Y REALIDADES

- **Mito:** Si fallamos en la **planificación** podemos añadir **más programadores** y recuperar el tiempo perdido.

- **Realidad:** Ley de Brooks: "Aregar gente a un proyecto atrasado, lo atrasa aún mas".

MITOS DEL CLIENTE



*Persona del
escritorio de al
lado, dpto. de
ventas, compañía
externa...*

MITOS Y REALIDADES

- **Mito:** Una declaración general de los **objetivos** es **suficiente** para comenzar a escribir los **programas**; podemos dar los detalles más adelante.

- **Realidad:** Una **mala definición inicial** es la principal causa del trabajo en vano. Es esencial una descripción formal y detallada del ámbito de la información.

MITOS Y REALIDADES

- **Mito:** Los **requisitos** del **proyecto** **cambian** continuamente pero los cambios pueden acomodarse fácilmente.

- **Realidad:** El **impacto** del **cambio varía** según el momento en el que se introduzca.

MITOS DEL DESARROLLADOR



*Mitos han
permanecido a
través de 50 años
de cultura de
programación*

MITOS Y REALIDADES

- **Mito:** Una vez que **hicimos** el **programa** y **funciona**, nuestro trabajo ha **terminado**.

- **Realidad:** Los datos industriales indican que entre el **50%** y el **70%** de todo el **esfuerzo** dedicado a un **programa** se realizará **después** de que se le haya entregado al **cliente** por primera vez.

MITOS Y REALIDADES

- **Mito:** No hay forma de comprobar la calidad del software hasta que está corriendo.

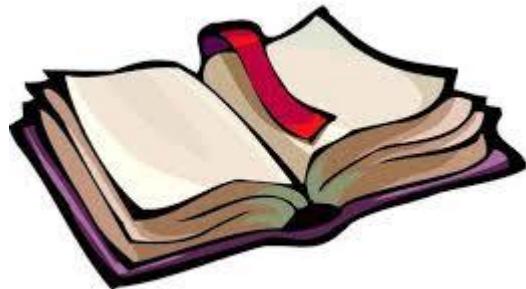
- **Realidad:** Hay técnicas que se pueden aplicar desde el principio, ese es uno de los objetivos de la IS.

**¿Qué se define
como Ingeniería de
Software?**



INGENIERÍA DE SOFTWARE

- + La aplicación de un enfoque **CUANTIFICABLE**, **DISCIPLINADO** y **SISTEMÁTICO** al desarrollo, operación y mantenimiento del software.



[IEEE, 1993]

INGENIERÍA DE SOFTWARE

- La definición y el uso de **PRINCIPIOS** y **MÉTODOS** de Ingeniería para obtener **SOFTWARE ECONÓMICO, CONFIABLE** y que trabaje sobre máquinas reales.



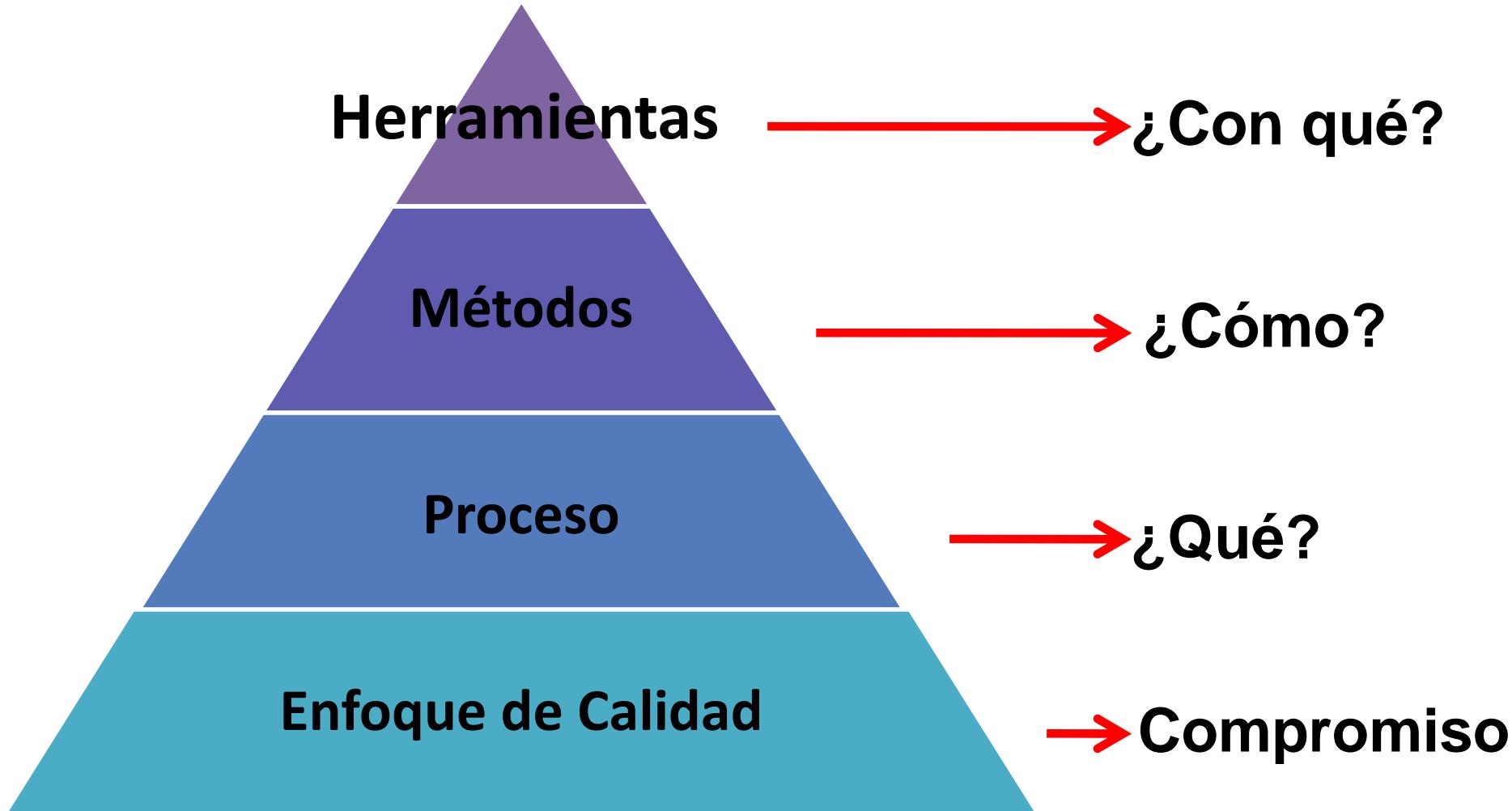
[Bauer , 1968]

Ian Sommerville

- “La IS es la **disciplina** de ingeniería **encargada** de todos los **aspectos** relacionados con la **producción** de **software** desde sus **etapas** más **tempranas** hasta el **mantenimiento** del sistema tras su puesta en marcha”.

IS COMO TECNOLOGÍA ESTRATIFICADA

Roger Pressman



HERRAMIENTAS CASE



Rational Rose



RequisitePro



Enterprise Architect



Visual Paradigm for UML

CARACTERÍSTICAS DE VISUAL PARADIGM

- + Disponibilidad en múltiples plataformas (Windows, Linux).
- + Diseño centrado en casos de uso y enfocado al negocio que generan un software de mayor calidad.
- + Soporte de UML versión 2.1.
- + Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- + Capacidades de ingeniería directa e inversa.
- + Licencia: gratuita y comercial.



¿Por qué surge la Ingeniería de Software?



INGENIERÍA DE SOFTWARE

- Para superar la crisis del software y terminar con los mitos del software existentes hasta ahora.
- Para acabar con las malas prácticas del desarrollo y mantenimiento de software llevadas a cabo por las empresas.
- Crear software de calidad.

... al no tener como base estos principios se continúan incrementando los problemas en el desarrollo de software.

**¿Por qué existen
problemas en el desarrollo
del software?**

**¿Qué se denomina
cómo el fundamento de
la IS?**



PROCESO DE SOFTWARE

- + Un **PROCESO** define: “**QUIÉN**”, “**QUÉ**”, “**CUÁNDΟ**”, y “**CÓMO**” hacer para alcanzar un determinado producto de software.



¿Qué es un proceso de desarrollo de software?



PROCESO DE DESARROLLO DE SOFTWARE

- + Colección de **actividades**, **acciones** y **tareas** que son desarrolladas para crear algún **resultado**, en este caso, software.

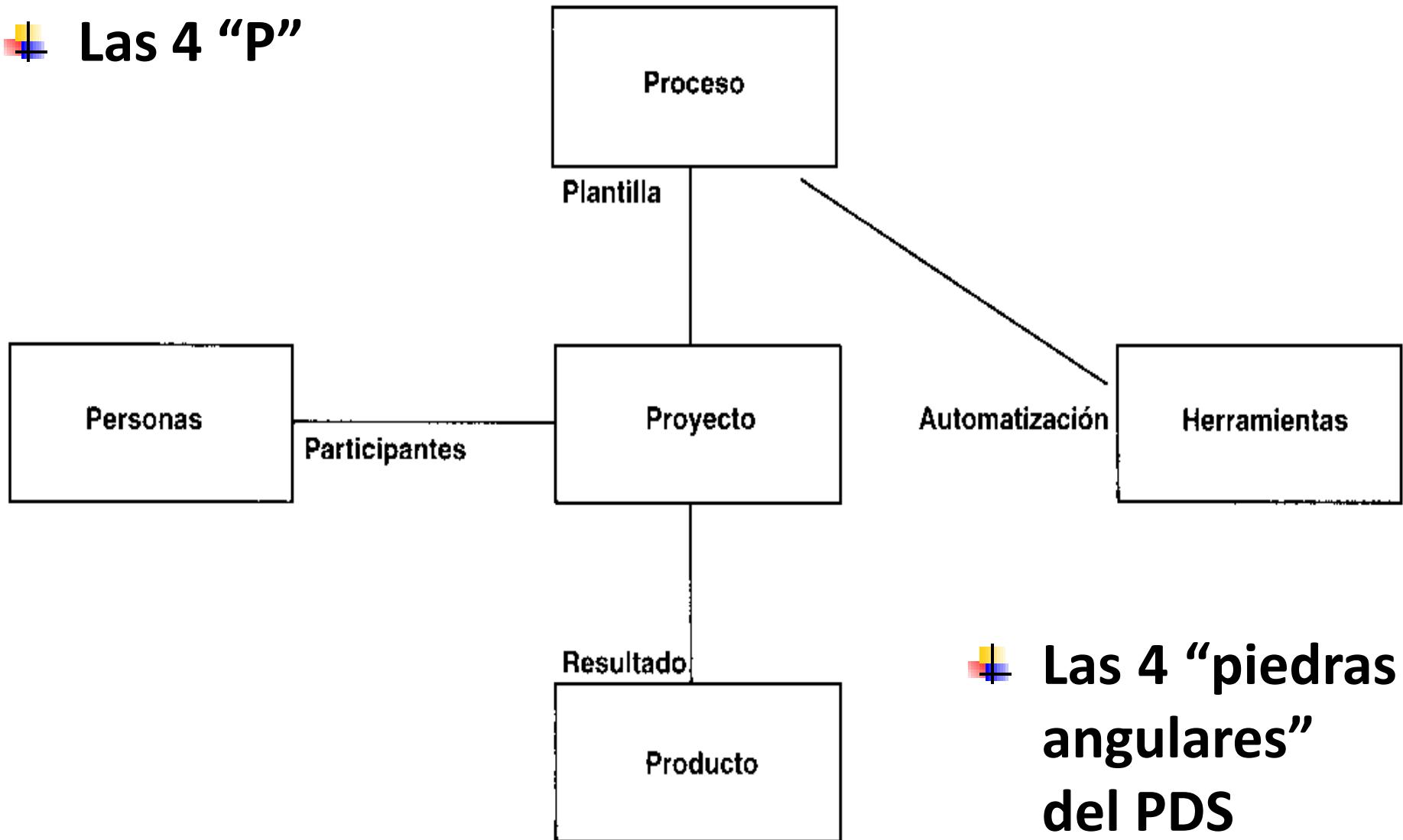


¿Quiénes estarían implicados en un proceso de desarrollo de software?



EN UN PROCESO ESTÁN IMPLICADOS

Las 4 “P”



Las 4 “piedras angulares” del PDS

- Deben estar organizadas en equipos eficientes, motivados para hacer un trabajo de SW de alta calidad y coordinados para lograr una comunicación eficaz.



Gestores ejecutivos.

Usuarios finales.

Gestores (técnicos) del proyecto.

Profesionales.

Clientes.

PRODUCTO

- Abarca cualquier software que se construye, son los artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

PROYECTO

- ✚ Elemento organizativo a través del cual se gestiona el desarrollo de software. El resultado de un proyecto es una versión de un producto.

- Definición de las actividades necesarias para transformar los requisitos de usuario en un producto. Además, representa una plantilla para crear proyectos y/o productos de trabajo.

PUNTOS CLAVE

- + ¿Cuáles son los aspectos más importantes que definen la IS?
- + ¿Cuáles son las capas de la IS y cómo se relacionan entre sí?
- + ¿Qué se denomina como el fundamento de la IS ?
- + ¿Qué otros elementos son imprescindibles a la hora de desarrollar software?



- Al cumplir las actividades de la vida cotidiana, se utilizan un conjunto de acciones y operaciones genéricas que sirven de base a variadas actividades.

- ✓ En la IS, al hacer software, se utilizan un conjunto de **actividades** que son **genéricas** y que sirven para todo tipo de proceso bajo el que realicemos software.

- + A este conjunto de actividades genéricas se le conoce en la Ingeniería como:

MARCO DE TRABAJO COMÚN DEL PROCESO DE DESARROLLO DE SOFTWARE

MARCO DE TRABAJO COMÚN DEL PDS

- Establece la base para un proceso de SW completo, identificando un número de **ACTIVIDADES GENÉRICAS** aplicables a todos los procesos de SW, sin importar su tamaño y complejidad.
- Además, define un conjunto de actividades denominadas **SOMBRIILLA** que se ejecutan a lo largo del PDS.

ESTRUCTURA MARCO DE TRABAJO COMÚN

Marco de trabajo común

Actividades del Marco de trabajo

Conjunto de tareas

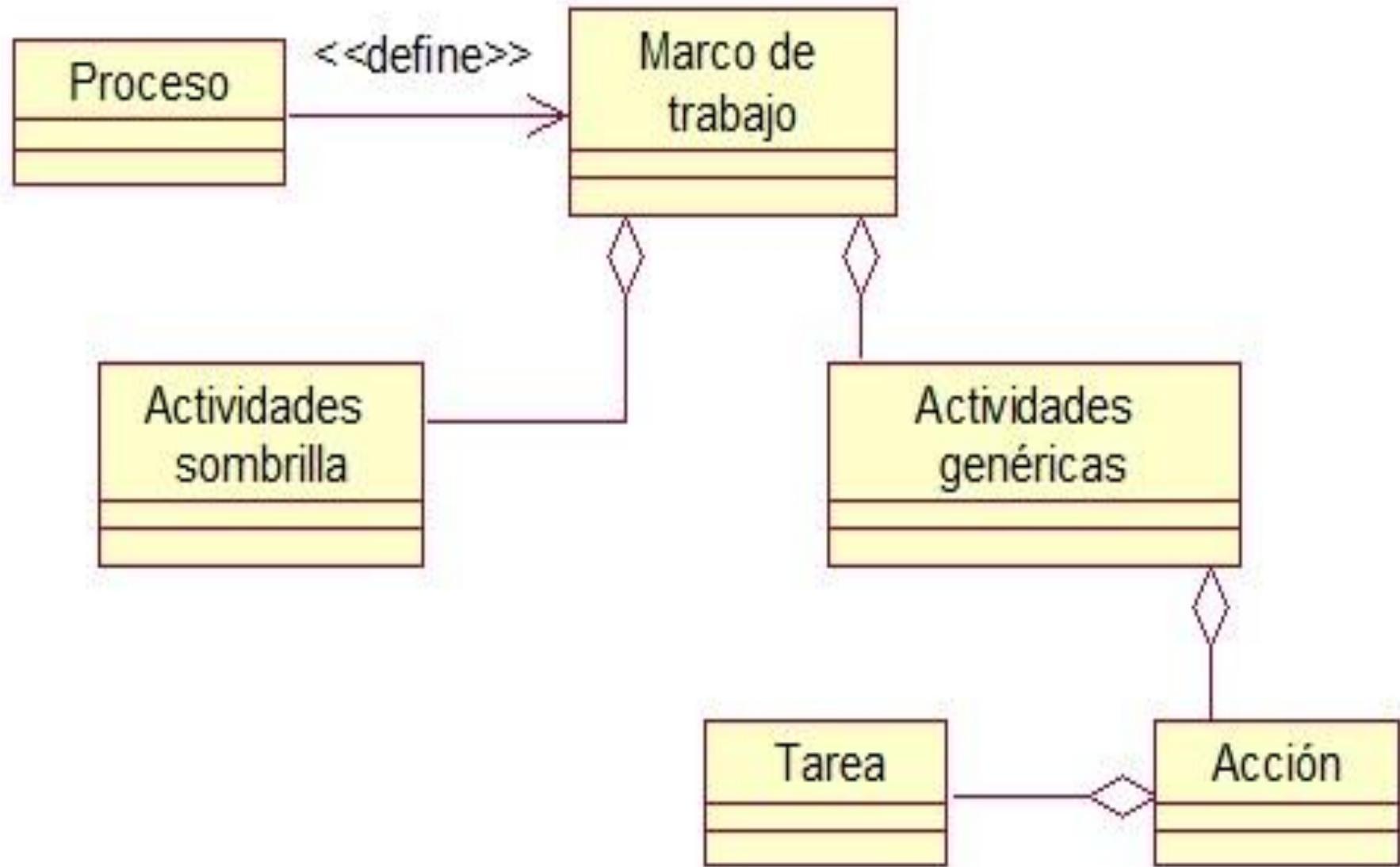
Tareas

Hitos, entregas

Puntos SQA

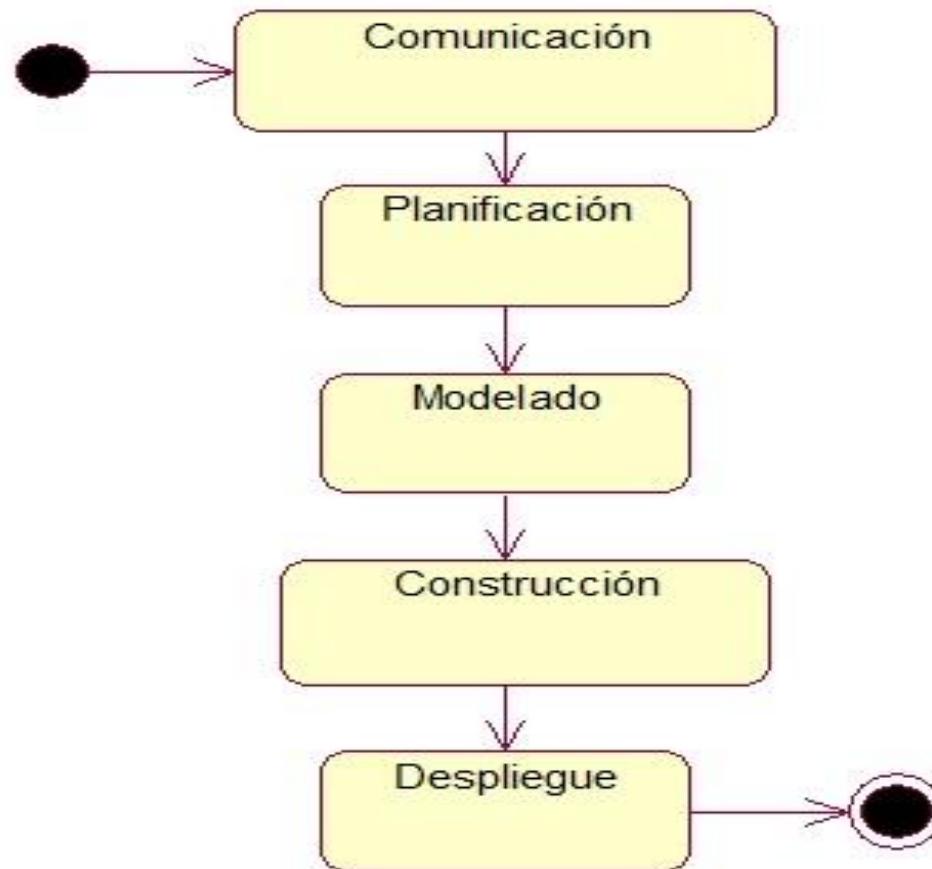
Actividades de protección

MARCO DE TRABAJO COMÚN DEL PDS



ACTIVIDADES GENÉRICAS

- Conjunto de **acciones** de la **IS**, es decir, una serie de **tareas relacionadas** de las que se **obtiene** un **producto**.



ACTIVIDADES GENÉRICAS

Comunicación:

-  Intensa comunicación y colaboración con los clientes de la aplicación.
-  Investigación de requisitos.
-  Definición del entorno y el alcance del proyecto.
-  Determinación de la fiabilidad del proyecto.

ACTIVIDADES GENÉRICAS

■ Planificación:

- Establecer un plan para el trabajo de la IS.
- Descripción de las tareas técnicas que deben realizarse.
- Determinación de los riesgos probables.
- Identificación y cálculo de los recursos necesarios.
- Establecimiento de los productos de trabajo que deberán obtenerse.

ACTIVIDADES GENÉRICAS

Modelado:

-  Construcción de modelos del SW que permitan el entendimiento del cliente y los desarrolladores.
-  Expresión en los modelos de los requisitos del sistema, el entorno de trabajo y el sistema a desarrollar; así como su despliegue físico.

ACTIVIDADES GENÉRICAS

Construcción:

-  Desarrollo en código (manual o automatizado) de los modelos diseñados para el sistema.
-  Realización de pruebas al SW desde los niveles de unidad hasta el nivel de sistema.

ACTIVIDADES GENÉRICAS

Despliegue:

- Instalación del SW resultante (como una unidad completa o en unidades que se incrementan en distintos momentos del desarrollo, por ejemplo la entrega por módulos del resultado total del sistema) en el entorno del cliente.

- Evaluación por el cliente del producto recibido, quien entrega entonces información de la conformidad con el resultado.

ACTIVIDADES SOMBRILLAS

- Ocurren a lo largo del PDS y se enfocan de manera principal en lo relacionado a la **gestión**, el **rastreo** o **seguimiento**; así como al **control** del proyecto.



ACTIVIDADES SOMBRILLAS

- + Seguimiento y control.
- + Administración del riesgo.
- + Aseguramiento de la calidad del producto.
- + Revisiones técnicas.
- + Medición.
- + Administración de la configuración.
- + Administración de la reutilización.
- + Preparación y producción del producto de trabajo.

ACTIVIDADES SOMBRILLAS

✚ Seguimiento y control:

- ✚ Evaluación del proyecto en su ejecución de acuerdo con lo planificado.
- ✚ Permite tomar las medidas necesarias para ajustar el cronograma o mantenerlo en su definición inicial.

ACTIVIDADES SOMBRILLAS

Administración del riesgo:

-  Evaluar los riesgos a los cuales pudiera enfrentarse el proyecto.
-  Permite definir los factores externos e internos del proyecto que pudieran afectar la ejecución del cronograma del proyecto.
-  Diseña la mitigación y la contingencia a dichos riesgos.

ACTIVIDADES SOMBRILLAS

Aseguramiento de la calidad del producto:

-  Se definen las actividades en forma de plan que integradas al plan o cronograma del proyecto garantizarán la calidad del producto.
-  No son actividades aisladas sino que sistémicamente en las áreas de proceso, proyecto y persona, garantizarán la calidad del producto.

ACTIVIDADES SOMBRILLAS

■ Revisiones técnicas:

- Encaminadas a evaluar los productos del trabajo de la IS.
- Están encaminadas a identificar y eliminar los errores, defectos y fallas ante de que estas se propaguen al resto de las fases del desarrollo del SW.

ACTIVIDADES SOMBRILLAS

✚ Medición:

✚ Definen, calculan y recolectan mediciones de todo el proceso, proyecto y producto (entre estas se pueden mencionar tiempo, esfuerzo y recursos) para ayudar al equipo a establecer sus métricas de trabajo y de desarrollo.

ACTIVIDADES SOMBRILLAS

- **Administración de la configuración:**
 - Tiene como objetivo manejar y documentar los cambios que se producen en el proyecto que afectan al producto de forma tal que puedan ser rastreados y corregidos.

ACTIVIDADES SOMBRILLAS

■ Administración de la reutilización:

- Establece los criterios o pautas para la reutilización de partes, unidades, módulos o componentes del producto, sobre las cuales se organiza el trabajo de desarrollo del software.

- Establece a su vez los procedimientos o mecanismos a través de los cuales se desarrollarán estas partes del producto.

ACTIVIDADES SOMBRILLAS

Preparación y producción del producto de trabajo:

-  Establece los criterios o pautas para la producción de los productos de trabajo (modelos, documentos, registros, formatos y listas) los cuales son necesarios para un entendimiento y lenguaje común entre los desarrolladores del producto.

PUNTOS CLAVE

- + Aunque estas actividades son genéricas a todo tipo de sistemas, los detalles del PDS serán muy diferentes en cada caso, aunque las actividades dentro del marco de trabajo permanecerán iguales.

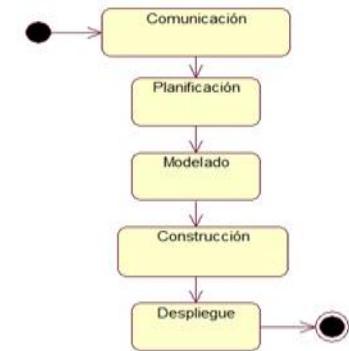


¿QUÉ HACE DIFERENTE AL PDS?

- Flujo global y las interdependencias entre tareas.
- Grado de definición de las tareas.
- Grado de identificación del producto.
- Forma de asegurar la calidad.
- Forma de aplicar el seguimiento y control.
- Grado de descripción del proceso.
- Grado de compromiso del cliente.
- Grado de autonomía del equipo de desarrollo.
- Grado de definición de la organización y las responsabilidades en el equipo.

¿A QUÉ SE DEBEN ESTAS DIFERENCIAS?

- + Diferentes **enfoques** de IGSW.
- + Diferentes **modelos** de desarrollo.
- + Diferentes **metodologías** de desarrollo.
- + Se mantienen invariantes los **principios** del marco de trabajo común del proyecto.



CONCLUSIONES

- + La IS es una tecnología ...
- + Un PDS es ...
- + Además, en un PDS están implicadas ...
- + El marco de trabajo común del PDS propone un conjunto de actividades genéricas, las cuales son ...
- + Las actividades sombrillas ocurren ...



TRABAJO INDEPENDIENTE

1. Recopilar información de un proyecto de los centros de la facultad o universidad a partir de los siguientes elementos a tener en cuenta:

- a. Nombre del centro.
- b. Nombre del proyecto.
- c. Dominio de aplicación que desarrollan. Explicar las características de la selección.
- d. Ejemplificar cómo se llevan a cabo las actividades sombrillas en uno de los proyectos.

2. Consultar en la 7ma edición de Pressman, epígrafe 3.1 (páginas 56-60), así como los epígrafes 4.1 y 4.2 (páginas 83-86) e identificar y caracterizar los **Enfoques de Ingeniería de Software**.

PRÓXIMA ACTIVIDAD

- **Actividad 2.**
- **Tema 1.** Proceso de Software.
- **Conferencia 2.** Enfoques de IS. Modelos y metodologías del PDS.
- **Objetivo.** Caracterizar los enfoques de IS, los modelos y las metodologías de PDS, enfatizando en aquellos elementos que permiten su selección de acuerdo al contexto.





Actividad 1.

Tema 1. Proceso de Software.

Conferencia 1. Introducción a la
Ingeniería de Software.