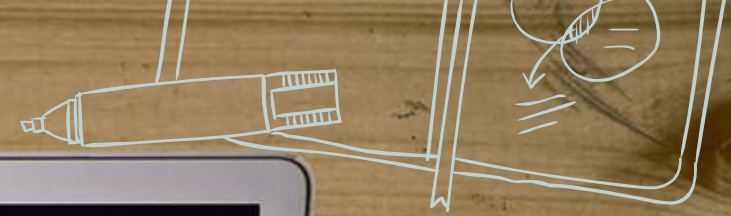
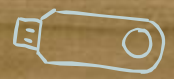
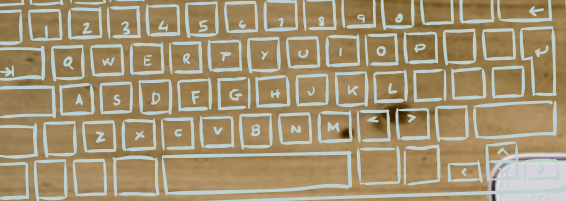




Ingeniería de software I

2024





Docentes

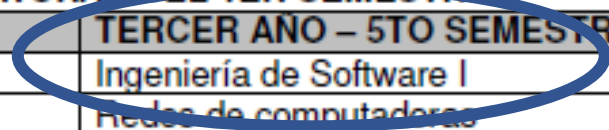
Mgter. Marisa Haderne

Lic Leandro Spadaro

Correlativas



PRIMER AÑO – 1ER SEMESTRE			
LCC01	Algebra		
LCC02	Análisis Matemático I		
LCC03	Introducción a la programación		
LCC04	Geometría Analítica		
PRIMER AÑO – 2DO SEMESTRE			
LCC05	Análisis Matemático II		LCC02, LCC04
LCC06	Matemática Discreta		LCC04
LCC07	Algoritmos y Estructura de Datos I		LCC03
LC008	Introducción a la Tecnología		
SEGUNDO AÑO – 3ER SEMESTRE			
LCC09	Lógica		LCC01
LCC10	Arquitectura de las computadoras		LCC08
LCC11	Métodos Numéricos		LCC03, LCC02, LCC04
LCC12	Algoritmos y Estructuras de Datos II	LCC03	LCC07
SEGUNDO AÑO – 4TO SEMESTRE			
LCC13	Sistemas Operativos		LCC08
LCC14	Paradigmas de programación		LCC03, LCC07
LCC15	Probabilidad y Estadística		LCC01, LCC02
LCC16	Lenguajes formales y computabilidad		LCC09
PARA CURSAR EL QUINTO SEMESTRE SE DEBEN APROBAR TODAS LAS ASIGNATURAS DEL 1ER SEMESTRE			
TERCER AÑO – 5TO SEMESTRE			
LCC17	Ingeniería de Software I		LCC14
LCC18	Redes de computadoras		LCC10
LCC19	Comunicación Técnica I		
LCC20	Teoría de bases de datos		LCC12
TERCER AÑO – 6TO SEMESTRE			
LCC21	Ingeniería de Software II		LCC17

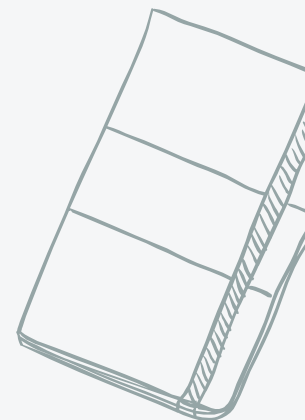




Días de cursado

Martes 18 a 20 hs

Viernes de 17 a 20 hs



Ingeniería del Software I-2023

[Área personal](#) / [Mis cursos](#) / [Grado](#) / [Licenciatura en Ciencias de la Computación](#) / [Cursos 2023](#) / [Ingeniería del Software I-2023](#) / [General](#)

General [Bienvenida](#) [Programa y fechas importantes](#) [Introducción a la ingeniería de software](#) [Noticias](#) [Teoría General de Sistemas. Tr](#)

[Sistemas de información](#) [Ciclos de vida](#) [Trabajo en metodologías ágiles. Complemento con videos](#) [Planificación](#) [Principios que guían la](#)

[Ingeniería de requisitos](#) [Ejercicio de integración.](#) [Lenguajes de especificación. Modelado de los requerimientos.](#) [Modelo de análisis](#) [Estir](#)

[Presentaciones](#) [Material complementario](#) [Actividades](#) [Evaluaciones](#) [Presentaciones de clases en pdf](#) [Recuperatorio de tareas](#) [Entr](#)

 Avisos



Clave IS2024LCC

Área	Espacio curricular	Régimen de cursado	Horas semanales
Ingeniería de Software, bases de datos y Sistemas de información	Ingeniería de Software I	Semestral	5
	Ingeniería de Software II	Semestral	5
	Proyecto I	Anual	4
	Teoría de bases de datos	Semestral	6
	Seguridad Informática	Semestral	5
	Proyecto II	Anual	4

Condiciones de Promoción

Aprobar cada una de las actividades presentes en la plataforma o su instancia recuperatoria con una nota individual igual o superior a 60% ***respetando las fechas de entrega correspondientes.***

Cumplir con el 75% de asistencia.

Aprobar con una nota igual o superior a 60% los 2 parciales o sus recuperatorios.

Condición de regularidad tras el cursado

Regularizarán la materia aquellos estudiantes que hayan aprobado ambos parciales o sus recuperatorios y hayan aprobado por lo menos el 80% de las actividades de la plataforma o su instancia recuperatoria con una nota igual o superior al 60% y hayan cumplido con el 70% de asistencia

Examen final para **alumnos libres**. En caso de alumnos que perdieron la regularidad.

1er parcial: 23 de abril

2do parcial: 31 de mayo

Recuperatorio 1er parcial: 11 junio

Recuperatorio 2do parcial: 14 de junio

01

Conceptos
Crisis del software
TGS y Sistemas de Inf
Los sistemas en la
organización
Principios

02

Proceso software y ciclo de
vida
Desarrollo ágil

03

Ingeniería de requerimiento
Especificación de
requerimientos. El trabajo
con el usuario

04

Lenguajes de especificación.
Modelado de requerimientos
UML

05

Métricas y estimación.
Cómo medimos un producto
software

06

Hacia donde vamos

INGENIERÍA DEL SOFTWARE



Definición y relaciones



“Arquitectos” de sistemas software



El concepto “ingeniería de software”: (1968), nació en una conferencia realizada para discutir la “crisis del software”.

Los enfoques individuales al desarrollo de programas no escalaban hacia los grandes y complejos sistemas de software. No eran confiables, costaban más de lo esperado y se distribuían con demora.



Síntomas de la “crisis del software”.

- Expectativas
- Fiabilidad
- Costo
- Facilidad de modificación
- Plazos
- Portabilidad
- Eficiencia



La ingeniería de software es:

Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software.

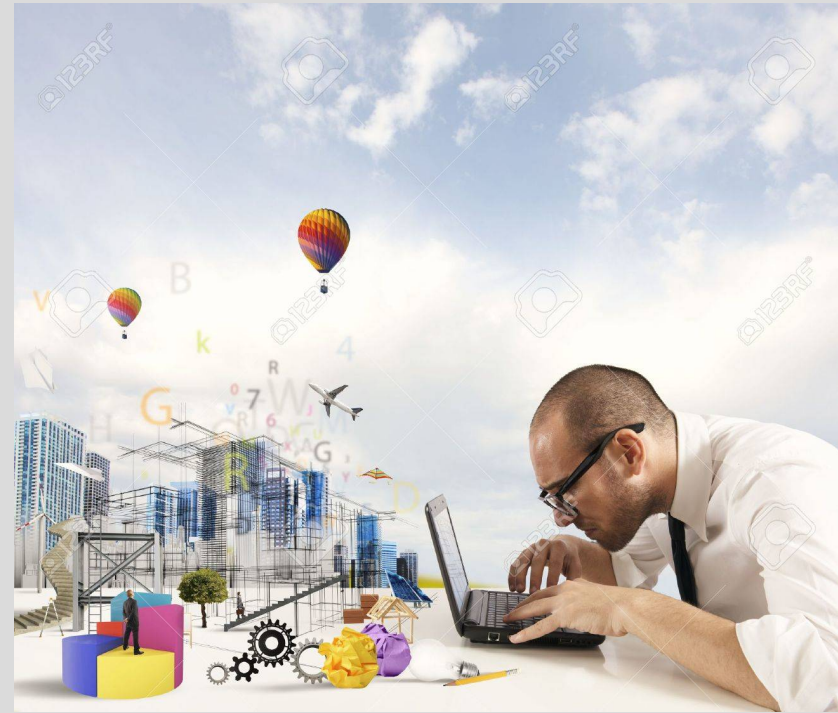
La ingeniería del software ofrece **métodos** o **técnicas** para desarrollar y mantener software de calidad que resuelven problemas de todo tipo, y trata áreas muy diversas de la informática y de las ciencias computacionales.

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.

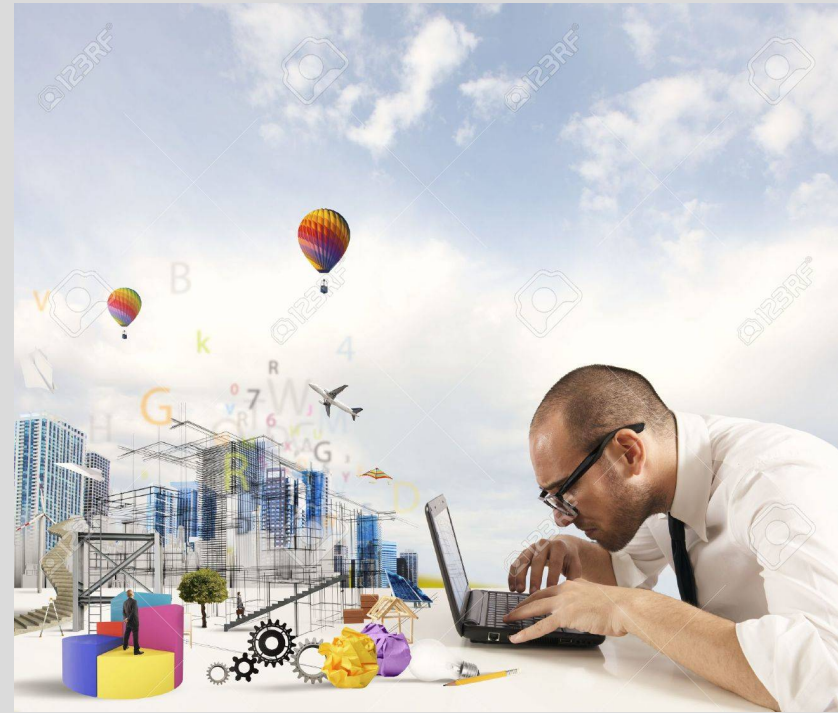


La ingeniería es el “arte” de hacer aplicaciones prácticas de los conocimientos científicos.

Los ingenieros no son científicos, estudian las ciencias y las usan para resolver problemas mediante el diseño creativo.



La ingeniería es el uso de los conocimientos científicos y de sus principios con un propósito práctico... generalmente mediante el diseño de productos útiles.



La ingeniería de software es una **disciplina** de ingeniería que se interesa por **todos los aspectos** de la producción de software, desde las primeras etapas de la especificación del sistema hasta el mantenimiento del sistema después de que se pone en operación.

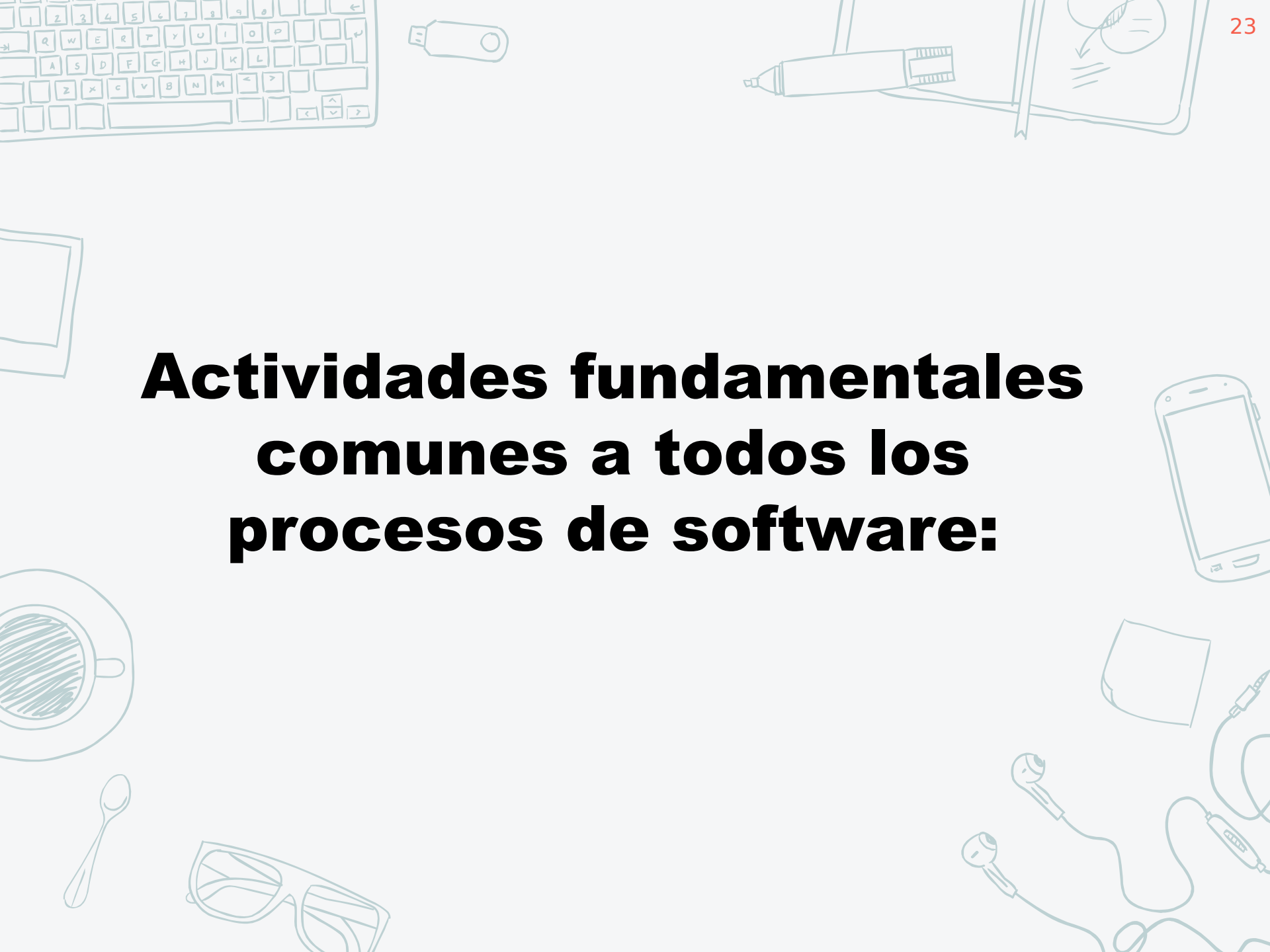


1. **Disciplina** de ingeniería. Los ingenieros aplican teorías, métodos y herramientas de manera selectiva y encontrando soluciones a problemas.

2. **Todos los aspectos** de la producción del software. Se interesa por los procesos técnicos y también por la administración del proyecto de software y el desarrollo de herramientas, así como métodos y teorías para apoyar la producción de software.



**Actividades fundamentales
comunes a todos los
procesos de software:**



1. Especificación del software, donde clientes e ingenieros definen el software que se producirá y las restricciones en su operación.

2. Desarrollo del software, donde se diseña y programa el software.



3. Validación del software, donde se verifica el software para asegurar que sea lo que el cliente requiere.

4. Evolución del software, donde se modifica el software para reflejar los requerimientos cambiantes del cliente y del mercado.



Los ingenieros hacen que las cosas funcionen. Aplican **teorías, métodos y herramientas** donde es adecuado, Reconocen que deben trabajar ante restricciones organizacionales y financieras, de modo que buscan soluciones dentro de tales limitaciones



Teorías, métodos y herramientas

¿Diferencias?



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



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.



La ingeniería de software es una tecnología con
varias capas



Para leer: página 24 de

Sommerville

PUNTOS CLAVE

- La ingeniería de software es una disciplina de ingeniería que se interesa por todos los aspectos de la producción de software.
- El software no es sólo un programa o programas, sino que también incluye documentación. Los atributos esenciales de los productos de software son mantenimiento, confiabilidad, seguridad, eficiencia y aceptabilidad.
- El proceso de software incluye todas las actividades que intervienen en el desarrollo de software. Las actividades de alto nivel de especificación, desarrollo, validación y evolución son parte de todos los procesos de software.
- Las nociones fundamentales de la ingeniería de software son universalmente aplicables a todos los tipos de desarrollo de sistema. Dichos fundamentos incluyen procesos, confiabilidad, seguridad, requerimientos y reutilización de software.
- Existen muchos tipos diferentes de sistemas y cada uno requiere para su desarrollo de herramientas y técnicas adecuadas de ingeniería de software. Existen pocas, si es que hay alguna, técnicas específicas de diseño e implementación que son aplicables a todos los tipos de sistemas.
- Las ideas fundamentales de la ingeniería de software son aplicables a todos los tipos de sistemas de software. Dichos fundamentos incluyen procesos de administración de software, confiabilidad y seguridad del software, ingeniería de requerimientos y reutilización de software.
- Los ingenieros de software tienen responsabilidades con la profesión de ingeniería y la sociedad. No deben preocuparse únicamente por temas técnicos.
- Las sociedades profesionales publican códigos de conducta que establecen los estándares de comportamiento esperados de sus miembros.

¿Qué cambió en estos últimos años?

Costo para almacenar un gigabyte según pasan los años

Florencia Abd
- LA NACION -

1980: US\$ 438.000

1985: US\$ 105.000

1990: US\$ 11.000

2000: US\$ 11

2005: US\$ 1

2012: US\$ 0,1

2014: US\$ 0,03

En 2016: SIN COSTO

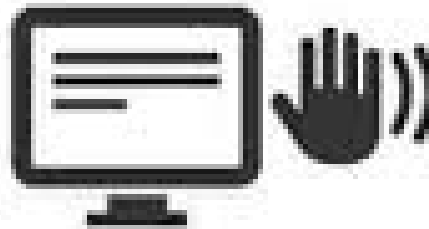
Google
15 GB

Box
10 GB

Amazon
5 GB

“¿QUÉ ES HARDWARE?

El que recibe los golpes cuando falla el **SOFTWARE.**”



Tipos de productos de software:

Productos genéricos

Productos personalizados





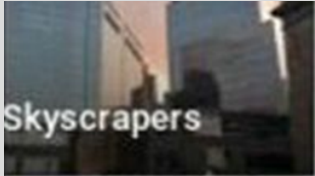
Errores y problemas





OS de la ciudad de
Buenos Aires

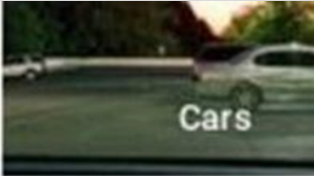




Skyscrapers



Airplanes



Cars



Bikes



Gorillas



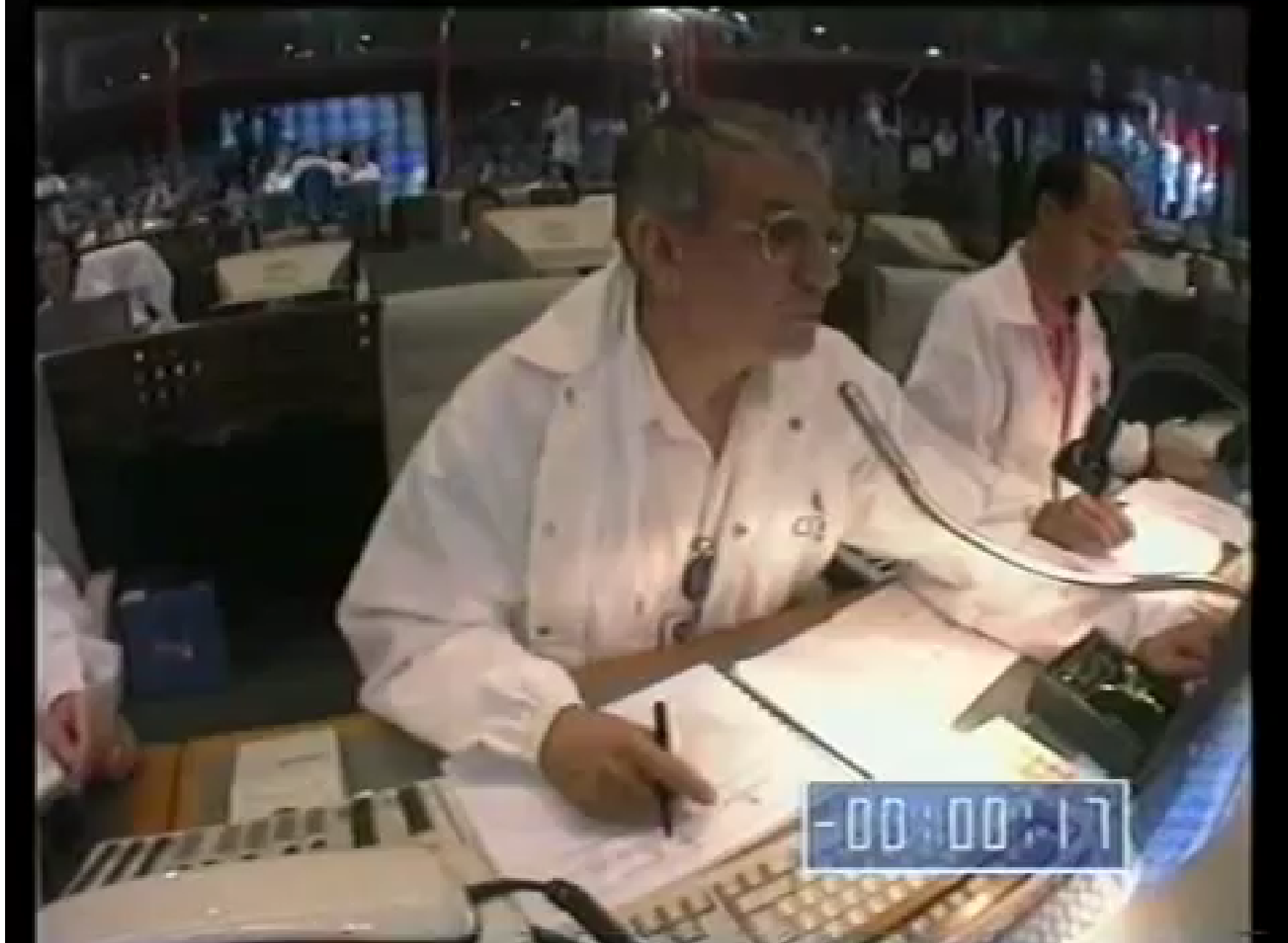
Graduation

En agosto de 2012, un error de programa casi provocó la quiebra de la empresa de inversión Knight Capital. La compañía perdió 500 millones de dólares en media hora debido a que sus ***computadoras comenzaron a comprar y vender millones de acciones sin ningún tipo de control humano.*** Como resultado, el precio de las acciones de Knight Capital cayó un 75% en dos días



En 2013, un ***error de programación*** provocó el caos en la compañía de [aviación](#) American Airlines. La unión de dos sistemas como resultado de la [fusión](#) de varias compañías aéreas originó un fallo en el sistema de reserva de pasajes. Concretamente, el problema surgió con toda probabilidad cuando se intentó unificar plataformas escritas en diferentes lenguajes de programación.





El 4 de junio de 1996 la Agencia Espacial Europea lanzó el cohete Ariane 5. Un error de programación en el módulo de gestión provocó la autodestrucción del cohete 37 segundos después del despegue.



A light blue line-art illustration of a computer keyboard on the left and a notepad with a pencil on the right, serving as a background for the top text.

DIO DE COMER A

A light blue line-art illustration of a laptop screen on the left side of the page.

MUCHOS

A light blue line-art illustration of a smartphone on the right side of the page.

INFORMÁTICOS

A light blue line-art illustration of a cup of coffee on the left side of the page.

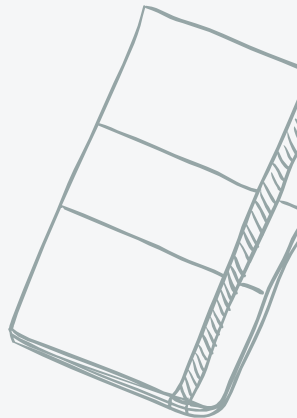
EL PROBLEMA DEL AÑO 2000.

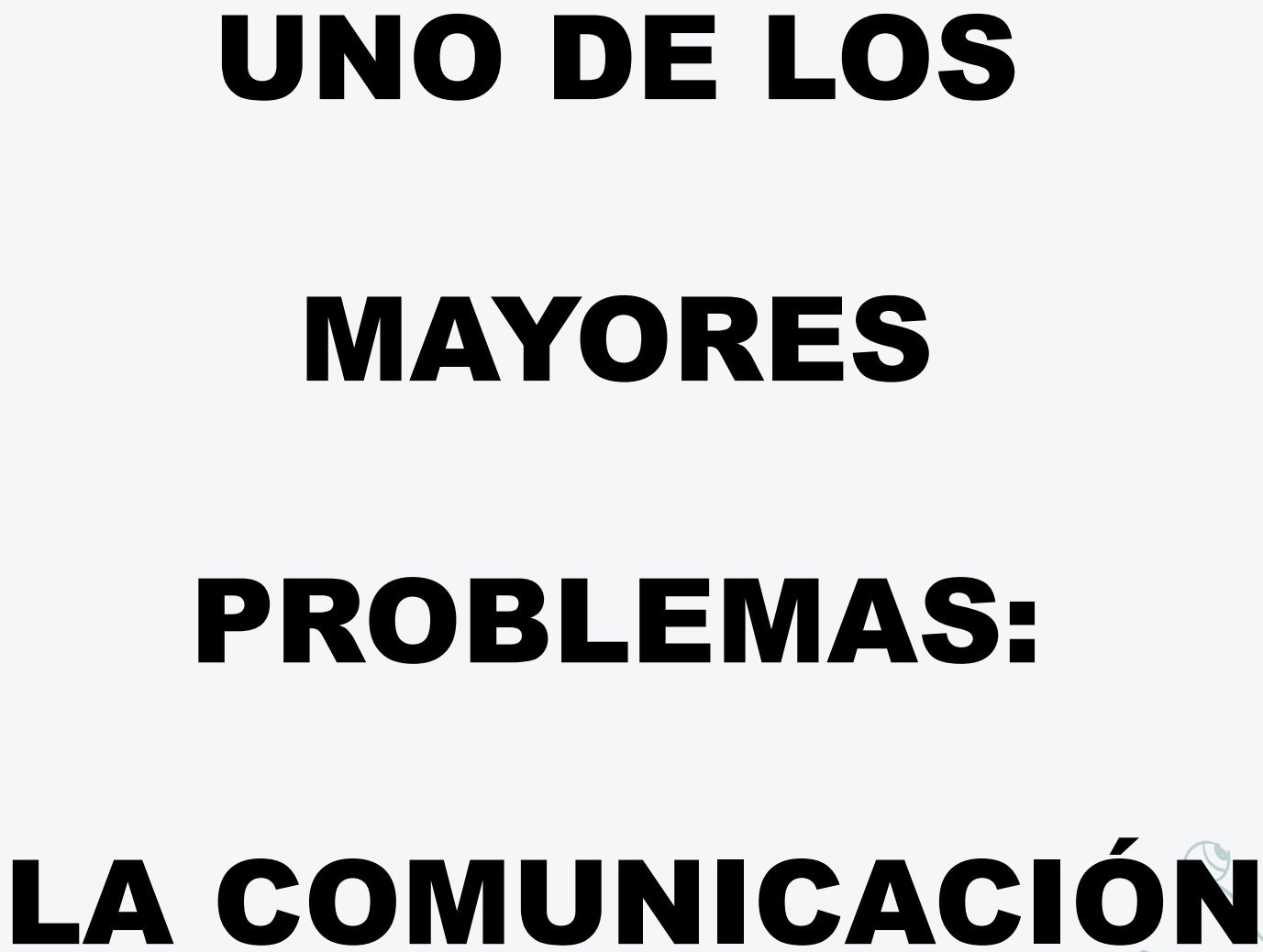
A light blue line-art illustration of a spoon on the bottom left side of the page.



Tragedia en Etiopía

Una “batalla” entre la computadora y los pilotos, posible causa de la caída





**UNO DE LOS
MAYORES
PROBLEMAS:
LA COMUNICACIÓN**



La solicitud del usuario



Lo que entendió el líder del proyecto



El diseño del analista de sistemas



El enfoque del programador



La recomendación del consultor externo



La documentación del proyecto



La implantación en producción



El presupuesto del proyecto



El soporte operativo



Lo que el usuario realmente necesitaba

A trabajar

Búsqueda de software que
haya fallado



Tarea 1

Deben buscar de manera individual errores de software importantes que ocurrieron en algún momento de la historia del software.

