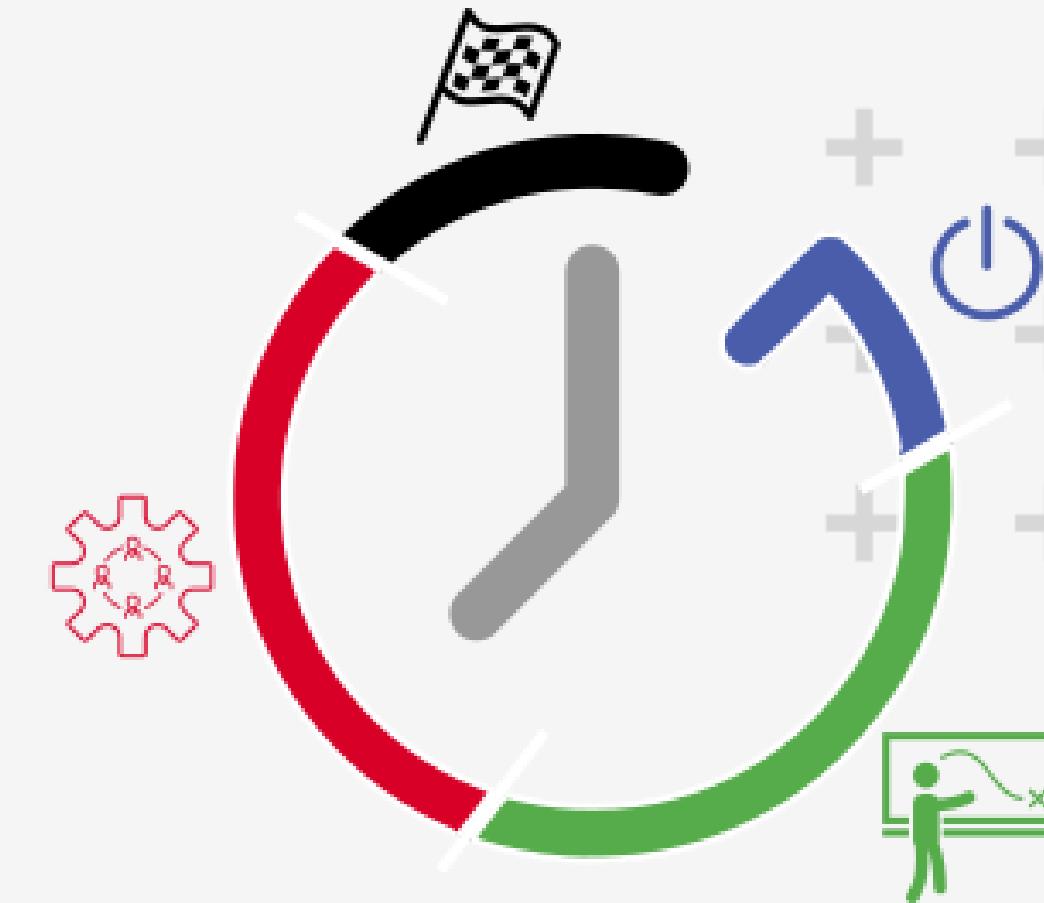




**Mtra. Perla Landero**  
**Seguridad en ingeniería de software**  
**Unidad I**

## Unidad I. Reto colaborativo



[Esta foto de Autor desconocido está bajo licencia CC BY-SA-NC](#)

# Participación

**Unidad I. Actividad de  
aprendizaje**



## **Fecha límite**

[Esta foto de Autor desconocido está bajo licencia CC BY-SA-NC](#)

**Lunes a las 23:55 pm.**

## Unidad I. Definiciones y taxonomía

# ¿Qué es la taxonomía?



Esta foto de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

## **Unidad 1. Definiciones y taxonomía**

**Es el área de seguridad computacional, se refiere a la forma en que puede ser clasificados o caracterizados los diferentes ataques o incidentes que existen hasta el momento.**

**Las taxonomías existentes no se enfocan específicamente en clasificación de ataques,**

[Esta foto de Autor desconocido está bajo licencia CC BY](#)

## **Unidad 1. Definiciones y taxonomía**

**Sino que se basan en vulnerabilidades de sistemas o fallas de seguridad, por ejemplo la taxonomía de activos y pasivos.**

[Esta foto](#) de Autor desconocido está bajo licencia [CC BY](#)

Fuente: [Caracterización de ataques informáticos \(tec.mx\)](#)

## **Unidad 1. Definiciones y taxonomía**



[Esta foto](#) de Autor desconocido está bajo licencia [CC BY](#)

**Ataques pasivos.** Son aquellos ataques que no afectan al estado de la información y/o sistema que la contiene.

Fuente: [Caracterización de ataques informáticos \(tec.mx\)](#)

## **Unidad 1. Definiciones y taxonomía**

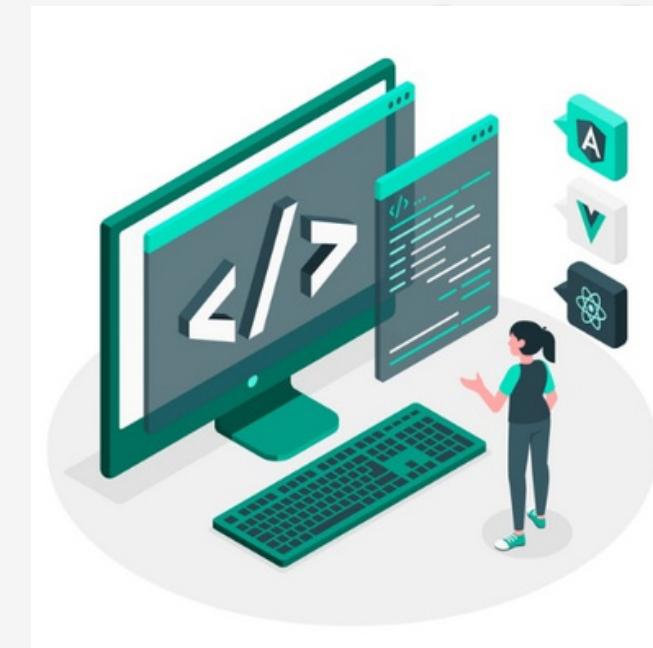


Esta foto de Autor desconocido está bajo licencia CC BY

**Ataques activos.** Se refieren a los ataques que afectan el estado de la información y/o sistema que la contiene.

Fuente: [Caracterización de ataques informáticos \(tec.mx\)](#)

## **Unidad I. Ingeniería de software y de sistemas**

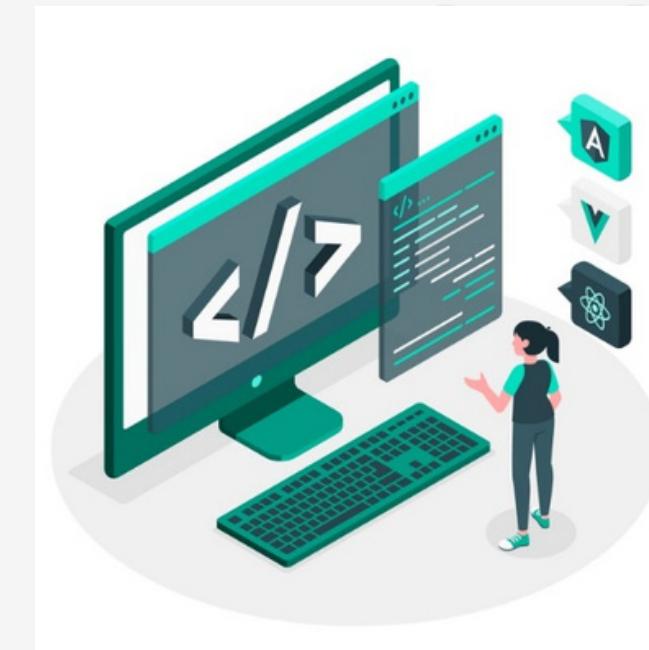


[Esta foto de Autor desconocido](#) está  
bajo licencia [CC BY-SA](#)

# **¿Qué es la ingeniería de software?**

## **Unidad I. Ingeniería de software y de sistemas**

**Es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de programas informáticos, más conocidos como softwares.**

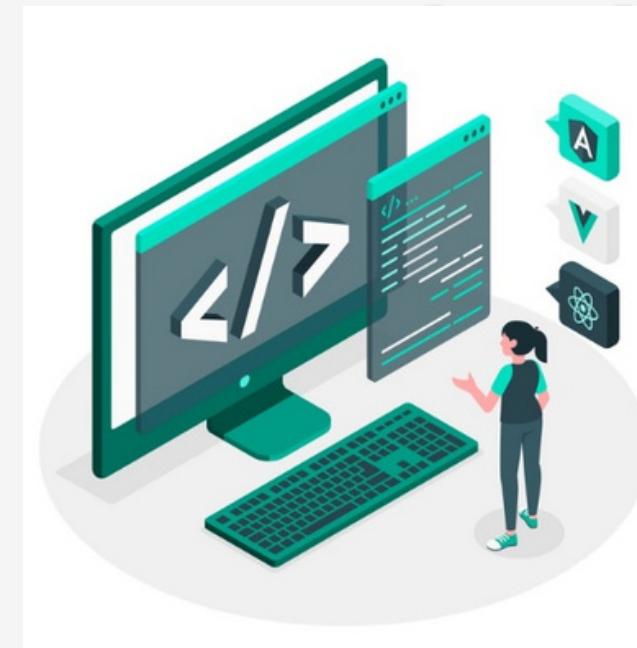


Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [¿Qué es y qué utilidad tiene la Ingeniería de Software? - Teknei](#)

## **Unidad I. Ingeniería de software y de sistemas**

**Engloba toda la gestión de un proyecto. Desde el análisis previo de la situación, el planteamiento del diseño hasta su implementación, pasando por las pruebas recurrentes para su correcto funcionamiento.**



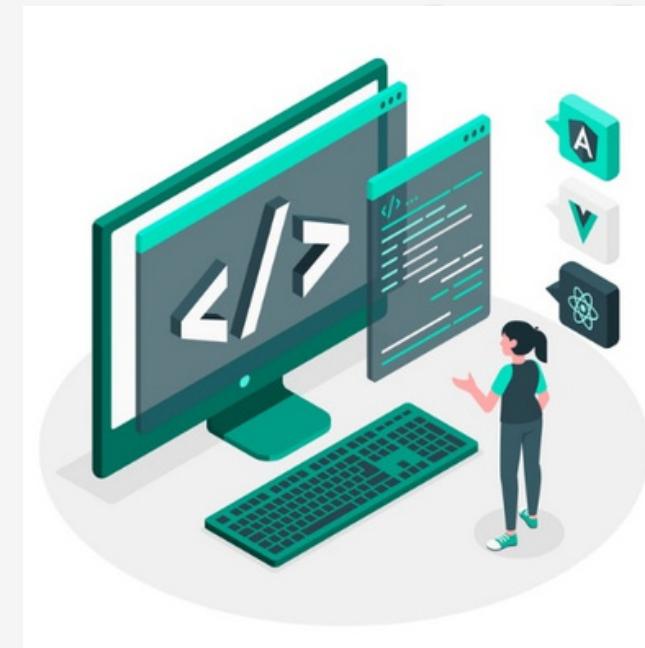
Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [¿Qué es y qué utilidad tiene la Ingeniería de Software? - Teknei](#)

## **Unidad I. Ingeniería de software y de sistemas**

**La ingeniería del software es el continente donde se aloja el contenido, que sería el software en sí.**

Fuente: [¿Qué es y qué utilidad tiene la Ingeniería de Software? – Teknei](#)

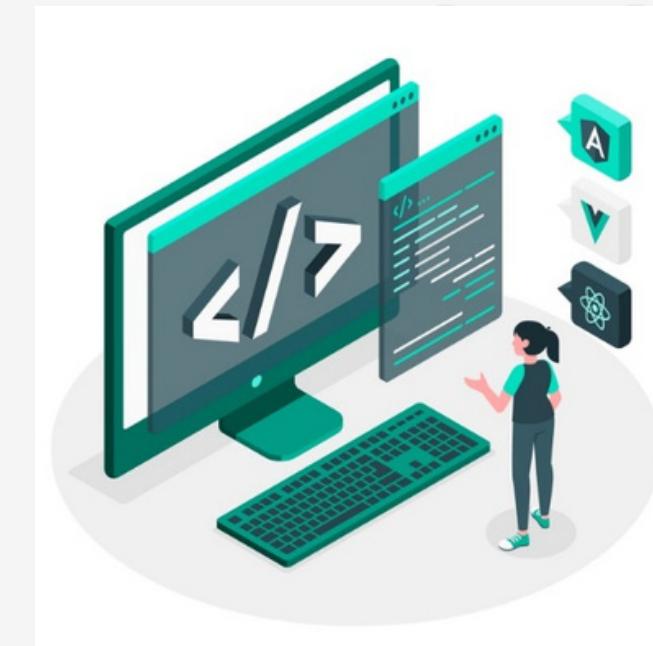


[Esta foto de Autor desconocido](#) está bajo licencia [CC BY-SA](#)

## Unidad I. Ingeniería de software y de sistemas

### Objetivos

**Estar presente en todas las fases del ciclo de vida de un producto.**  
**Estructurar la elaboración de evidencias que comprueben el perfecto funcionamiento de los programas y que se adaptan a los requerimientos de análisis y diseño.**



Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [¿Qué es y qué utilidad tiene la Ingeniería de Software? - Teknei](#)

## Unidad I. Ingeniería de software y de sistemas

### Objetivos

**Incluir procesos de calidad en los sistemas, calculando métricas e indicadores y chequeando la calidad del software producido.**

**Diseñar, construir y administrar bases de datos.**



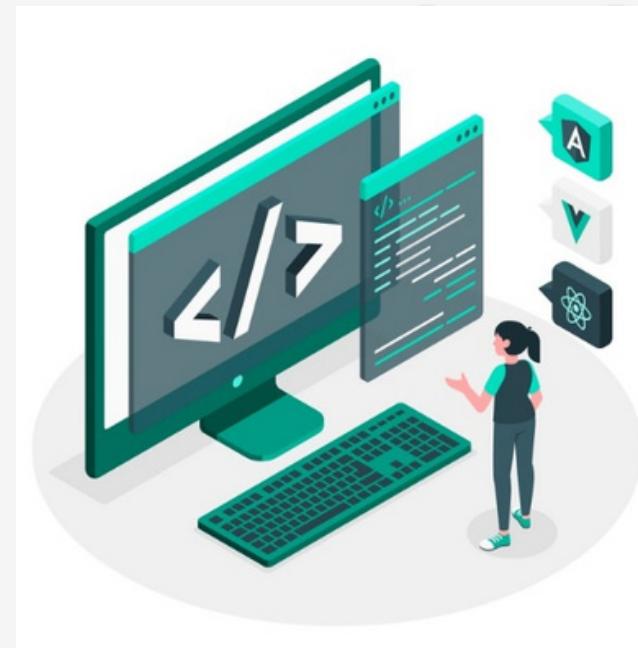
Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [¿Qué es y qué utilidad tiene la Ingeniería de Software? - Teknei](#)

## **Unidad I. Ingeniería de software y de sistemas**

Los cimientos de la ingeniería de software están sustentados en la calidad.

Básicamente, cualquier proceso, tecnología o producto desarrollado con ingeniería debe ser sustentado con calidad. Se puede considerar que el proceso de ingeniería de software es el elemento que mantiene unidas las partes tecnológicas y permite el desarrollo racional y puntual del software.

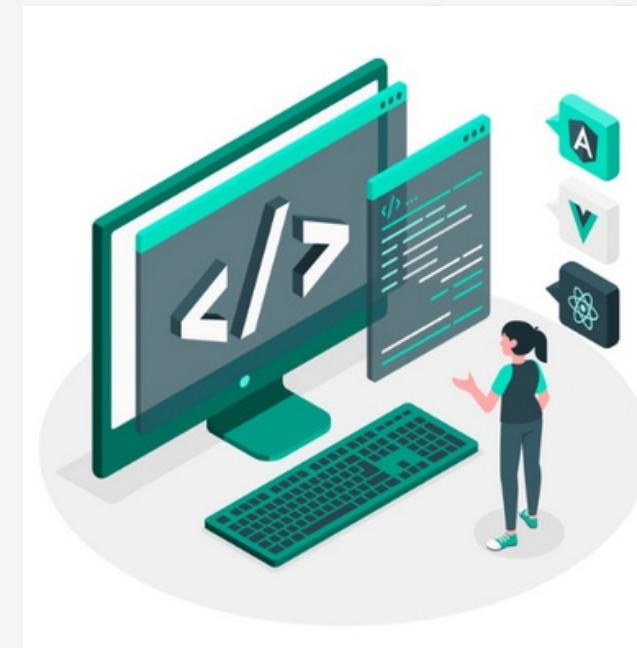


Esta foto de Autor desconocido está bajo licencia CC BY-SA

## **Unidad I. Ingeniería de software y de sistemas**

**Las principales tareas de la ingeniería de software pueden ser descritas de la siguiente forma:**

**Asignar modelos y estándares que agilicen la comunicación interna o externa entre clientes y equipo de desarrollo de software.**



Esta foto de Autor desconocido está bajo licencia CC BY-SA

## **Unidad I. Ingeniería de software y de sistemas**

**Proveer procedimientos, herramientas y métodos que permitan el desarrollo del software de forma eficiente.**

**Designar criterios y parámetros que permitan evaluar la calidad del software.**



Esta foto de Autor desconocido está bajo licencia CC BY-SA

## Unidad I. Ciclo de Vida del Software

**La metodología para el desarrollo de software es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito.**

Esta sistematización nos indica cómo dividiremos un gran proyecto en módulos más pequeños llamados etapas, y las acciones que corresponden en cada una de ellas, nos ayuda a definir entradas y salidas para cada una de las etapas y, sobre todo, normaliza el modo en que administraremos el proyecto.



Esta foto de Autor desconocido  
está bajo licencia CC BY-SA

## Unidad I. Ciclo de Vida del Software

Una metodología para el desarrollo de software son los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que cumplimos el objetivo por el cual fue creado.

Fuente: [01\\_DesarrollosMovNet.qxd \(pbworks.com\)](#)



Esta foto de Autor desconocido  
está bajo licencia CC BY-SA

## Unidad I. Ciclo de Vida del Software

**El ciclo de vida de un software tiene tres etapas claramente diferenciadas.**

**Planificación.** Idearemos un planeamiento detallado que guíe la gestión del proyecto, temporal y económicoamente.

**Implementación.** Acordaremos el conjunto de actividades que componen la realización del producto.



Esta foto de Autor desconocido  
está bajo licencia CC BY-SA

Fuente: [01\\_DesarrollosMovNet.qxd \(pbworks.com\)](#)

## Unidad I. Ciclo de Vida del Software

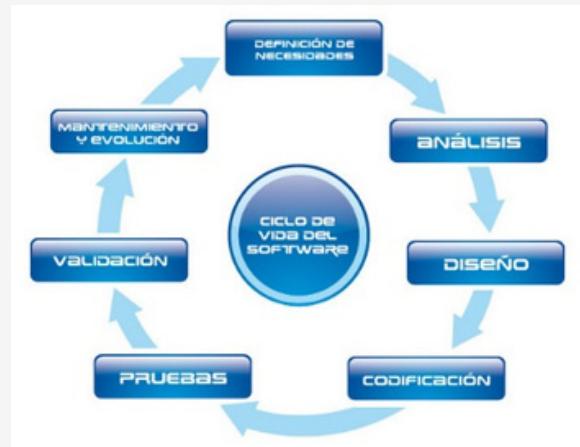


**Puesta en producción.** Un proyecto entra en la etapa de definición, allí donde se lo presentamos al cliente o usuario final, sabiendo que funciona correctamente y responde a los requerimientos solicitados en su momento.

Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [01\\_DesarrollosMovNet.qxd \(pbworks.com\)](#)

# Unidad I. Ciclo de Vida del Software



Esta etapa es muy importante no sólo por representar la aceptación o no del proyecto por parte del cliente o usuario final sino por las múltiples dificultades que suele presentar en la práctica, alargándose excesivamente y provocando costos no previstos.

Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [01\\_DesarrollosMovNet.qxd \(pbworks.com\)](#)

# Unidad I. Ciclo de Vida del Software



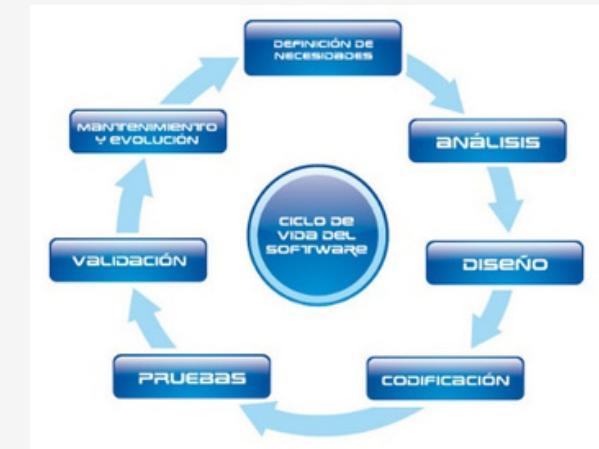
[Esta foto de Autor desconocido](#)  
está bajo licencia [CC BY-SA](#)

**Inicio.** Este es el nacimiento de la idea. Aquí definimos los objetivos del proyecto y los recursos necesarios para su ejecución.  
Hacia dónde queremos ir, y no cómo queremos ir.

Fuente: [01\\_DesarrollosMovNet.qxd \(pbworks.com\)](#)

## Unidad I. Ciclo de Vida del Software

Las características implícitas o explícitas de cada proyecto hacen necesaria una etapa previa destinada a obtener el objetivo por el cual se escribirán miles o cientos de miles de líneas de código. Un alto porcentaje del éxito del proyecto se definirá en estas etapas que, al igual que la etapa de debugging, muchos líderes de proyecto subestiman.



Esta foto de Autor desconocido  
está bajo licencia CC BY-SA

# Unidad I. Ciclo de Vida del Software



**Control en producción. control del producto, analizando cómo el proceso difiere o no de los requerimientos originales e iniciando las acciones correctivas si fuesen necesarias.**

Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [01\\_DesarrollosMovNet.qxd \(pbworks.com\)](#)

## Unidad I. Ciclo de Vida del Software

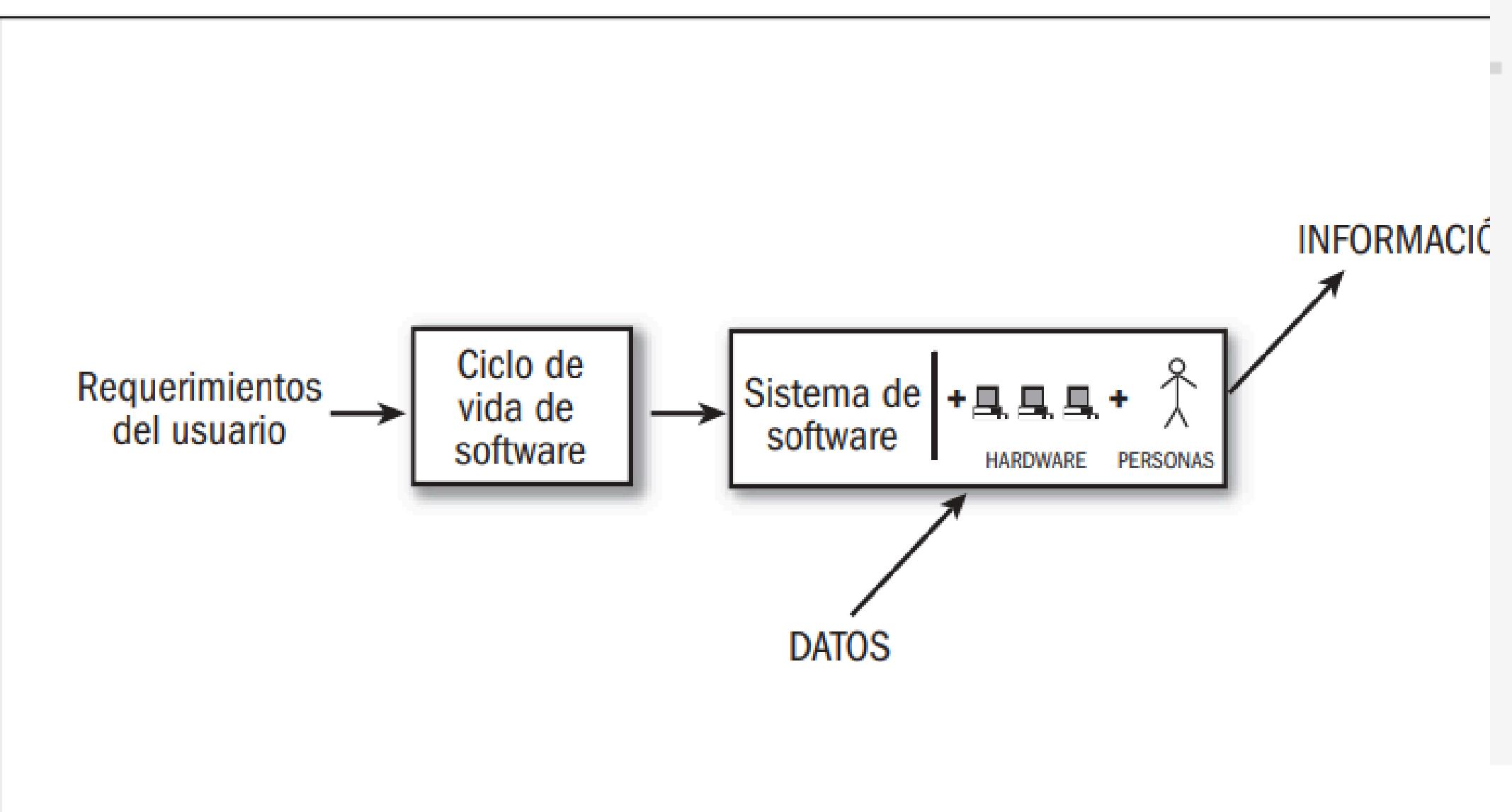
Cuando se dice que hay que corregir el producto, se hace referencia a pequeñas desviaciones de los requerimientos originales que puedan llegar a surgir en el ambiente productivo. Si el programa no realiza la tarea para lo cual fue creada, esta etapa no es la adecuada para el rediseño. Se incluye también en esta etapa el liderazgo, documentación y capacitación, proporcionando directivas a los recursos humanos, para que hagan su trabajo en forma correcta y efectiva.



Esta foto de Autor desconocido  
está bajo licencia CC BY-SA

Fuente: 01\_DesarrollosMovNet.qxd (pbworks.com)

# Unidad I. Ciclo de Vida del Software



## **Unidad I. Ingeniería segura de software**

Hoy en día, la seguridad de las aplicaciones es un aspecto fundamental y prioritario. El desarrollo seguro de software es una metodología cuyo objetivo es considerar la seguridad de las aplicaciones durante todo su ciclo de vida, empezando desde la propia definición de requisitos de las mismas.

Fuente: [Desarrollo seguro de software - ITCL](#)



Esta foto de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

## **Unidad I. Ingeniería segura de software**

**El propósito de esta filosofía es determinar cuanto antes las necesidades de seguridad y las posibles vulnerabilidades que puede tener la aplicación, sin esperar a fases o iteraciones posteriores.**

Fuente: [Desarrollo seguro de software - ITCL](#)



Esta foto de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

## Unidad I. Ingeniería segura de software

### Ventajas

**Aplicaciones más seguras.** El número de vulnerabilidades que tendrán las aplicaciones será menor, al igual que su criticidad. Esto es fundamental puesto que el número de ataques que se aprovechan de vulnerabilidades en las aplicaciones es cada vez mayor y las consecuencias para las empresas y las personas son cada vez más severas.



Esta foto de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

Fuente: [Desarrollo seguro de software - ITCL](#)

## Unidad I. Ingeniería segura de software

### Ventajas

**Optimización de tiempos y costes de ejecución en los proyectos. Los resultados de los proyectos serán mejores, puesto que la corrección de los fallos de seguridad requerirá menores tiempos de desarrollo al detectarse en fases más tempranas, y se evitarán imprevistos de última hora que puedan causar incumplimientos de plazo.**



Esta foto de Autor desconocido está bajo licencia CC BY-SA-NC

Fuente: [Desarrollo seguro de software - ITCL](#)

## **Unidad I. Ingeniería segura de software**



### **Ventajas**

Esta foto de Autor desconocido está bajo licencia CC BY-SA-NC

### **Mayor satisfacción de los clientes.**

Fuente: [Desarrollo seguro de software - ITCL](#)

## **Unidad I. Objetivos del software seguro**

El desarrollo seguro de software es un modelo de trabajo que se basa en la realización de chequeos de seguridad continuos del proyecto en construcción, incluso desde sus fases iniciales y antes de que se escriba una sola línea de código. Estas pruebas se centran en descubrir y corregir cualquier error en una etapa temprana, y comprenden tests de autenticación, autorización, confidencialidad, no repudio, integridad, estabilidad, disponibilidad o resiliencia.



Esta foto de Autor desconocido está bajo licencia CC BY-SA-NC

Fuente: [Metodologías de desarrollo seguro de software | BETWEEN Technology](#)

## **Unidad I. Objetivos del software seguro**



Esta foto de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

**El objetivo es, al fin y al cabo, asegurarnos de que impediremos el acceso al programa y a los datos almacenados por parte de usuarios carentes de permiso.**

Fuente: [Metodologías de desarrollo seguro de software | BETWEEN Technology](#).

## **Unidad I. Propiedades del software seguro**

**Confidencialidad.** Las características, activos y contenidos que administra el software, son accesibles solo a entidades autorizadas.

**Integridad. Resistencia y flexibilidad** a modificaciones no autorizadas del código, o a configuraciones y actualizaciones del software por entidades autorizadas.



Esta foto de Autor desconocido está bajo licencia [CC BY-SA-NC](#)

Fuente: [Diapositiva 1 \(cidecuador.org\)](#)

## **Unidad I. Propiedades del software seguro**



Esta foto de Autor  
desconocido está bajo  
licencia CC BY-SA-  
NC

**Disponibilidad.** El software debe estar accesible y operativo a los usuarios autorizados.

Fuente: [Diapositiva 1 \(cidecuador.org\)](http://cidecuador.org)

## **Unidad I. Propiedades del software seguro**

**Trazabilidad.** incluye el registro de las acciones relevantes relacionadas con la seguridad de una entidad que actúa como usuario, con el objetivo de establecer responsabilidades.

**No repudio.** conlleva prevenir que una entidad o usuario pueda desmentir la responsabilidad de acciones que han sido ejecutadas.



Esta foto de Autor  
desconocido está bajo  
licencia CC BY-SA-  
NC

Fuente: [Diapositiva 1 \(cidecuador.org\)](#)

## Unidad I. Propiedades Fundamentales

Los principios fundamentales de la seguridad en el manejo de la información son la confidencialidad, la integridad y la disponibilidad. Cada elemento juega un papel clave en un buen programa de protección de datos y, por eso, hay que tenerlos todos en cuenta en su definición.

Fuente: [Principios de seguridad en el manejo de la información | EAE](#)



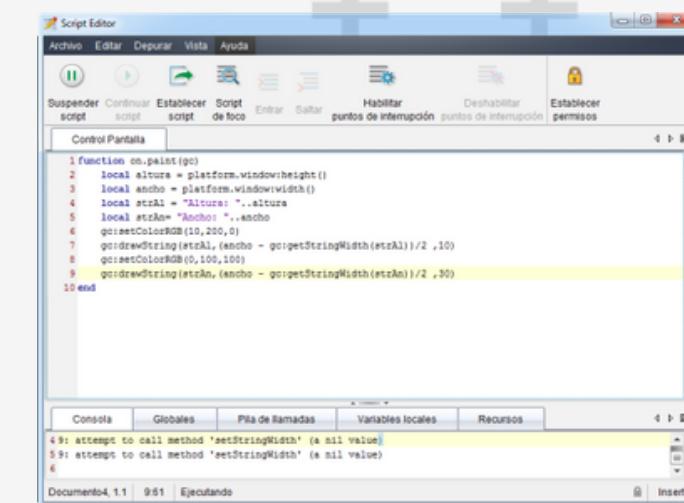
Esta foto de Autor desconocido está bajo licencia CC BY-SA-NC

## Unidad I. Propiedades conducentes

**Las siguientes son propiedades que, si bien no hacen directamente a la naturaleza segura del mismo, permiten caracterizarlo de alguna manera con respecto a la seguridad.**

**Dependability.** Propiedad del software que asegura que éste siempre ha de operar de la manera esperada sin defectos y debilidades.

Fuente: [Microsoft Word - VersionFinal.doc \(core.ac.uk\)](#)

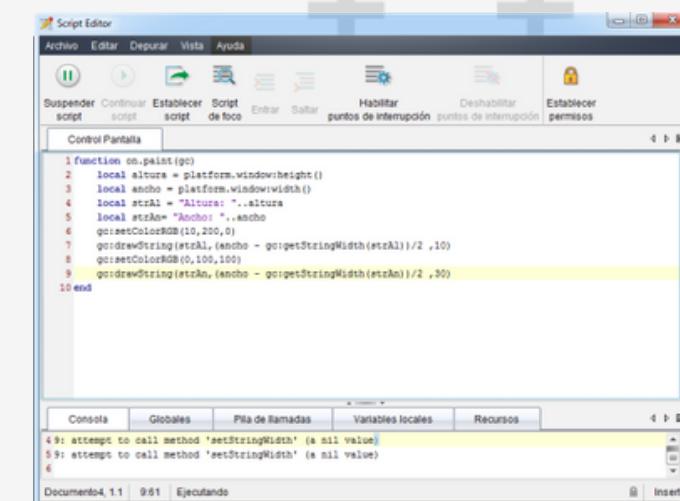


Esta foto de Autor desconocido  
está bajo licencia CC BY-NC

## Unidad I. Propiedades conducentes

**La intencionalidad y el impacto resultante en caso de un ataque determinan si un defecto o una debilidad realmente constituyen una vulnerabilidad que incrementa el riesgo de seguridad.**

**Correcto. Es una propiedad tan importante como las que caracterizan a la seguridad y, en ningún caso, se la debe satisfacer a expensas de otra.**



The screenshot shows a 'Script Editor' window with the following code in the script pane:

```
1 function onPaint(pci)
2     local altura = platform.window.height()
3     local ancho = platform.window.width()
4     local strAll = "Altura: " .. altura
5     local strAncho = "Ancho: " .. ancho
6     gcsetColorRGB(10,200,0)
7     gcdrawString(strAll, (ancho - gcgetStringWidth(strAll))/2 ,10)
8     gcsetColorRGB(0,100,100)
9     gcdrawString(strAncho, (ancho - gcgetStringWidth(strAncho))/2 ,30)
10 end
```

The execution results pane shows two errors:

```
4:9: attempt to call method 'setStringWidth' (a nil value)
5:9: attempt to call method 'setStringWidth' (a nil value)
```

At the bottom, it says 'Documento4.1.1 961 Ejecutando'.

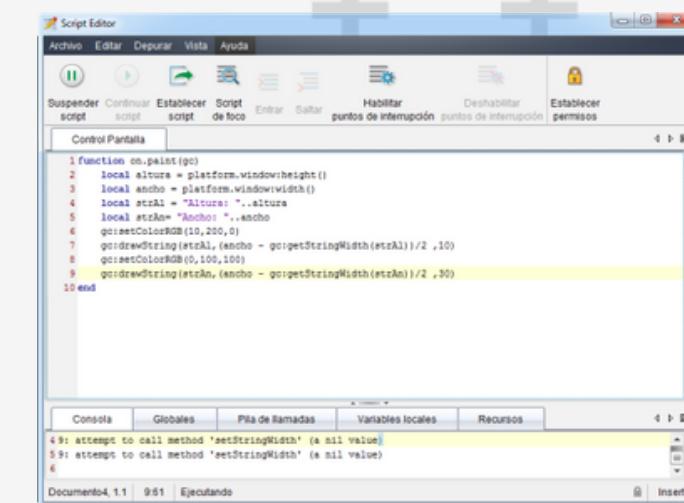
[Esta foto de Autor desconocido](#)  
está bajo licencia CC BY-NC

Fuente: [Microsoft Word - VersionFinal.doc \(core.ac.uk\)](#)

## Unidad I. Propiedades conducentes

**Generalmente su alcance está restringido a las condiciones previstas de operación; pero la seguridad requiere que se**

**la preserve incluso bajo condiciones no previstas, que son las que se presentan cuando el software se encuentra sometido a un ataque.**



The screenshot shows a Windows Script Editor window with the following code:

```
function onPaint(pci)
    local altura = platform.window.height()
    local ancho = platform.window.width()
    local strAll = "Altura: "..altura
    local strAn = "Ancho: "..ancho
    gosetColorRGB(10,200,0)
    gordrawString(strAll, (ancho - ggetStringWidth(strAll))/2,10)
    gosetColorRGB(0,100,100)
    gordrawString(strAn, (ancho - ggetStringWidth(strAn))/2 ,30)
end
```

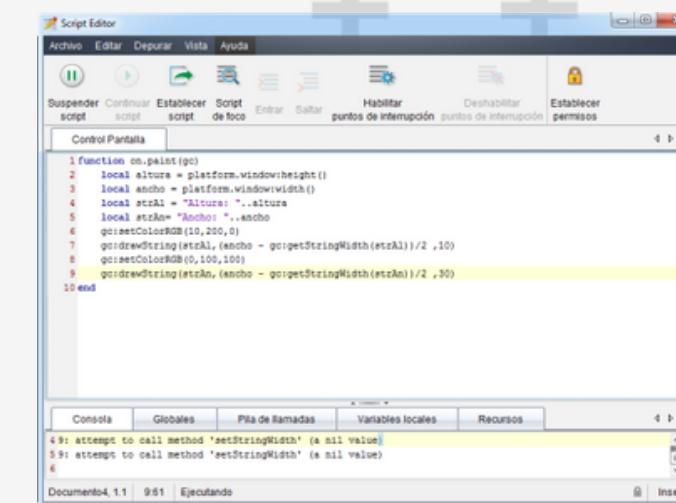
Below the code, the status bar indicates "Documento4.1.1 961 Ejecutando".

[Esta foto de Autor desconocido](#)  
está bajo licencia CC BY-NC

Fuente: Microsoft Word - VersionFinal.doc (core.ac.uk)

## Unidad I. Propiedades conducentes

**Predecible.** Las funcionalidades, propiedades y comportamientos del software siempre serán los que se espera que sean, tal que en todo momento las condiciones bajo las que opera es decir, el ambiente, las entradas que recibe también sean predecibles; así, el software nunca se desviará de su operación correcta bajo condiciones previstas.



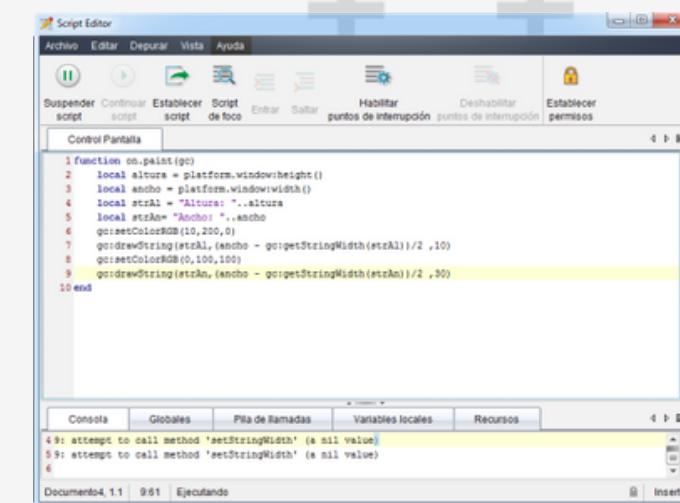
Esta foto de Autor desconocido  
está bajo licencia CC BY-NC

Fuente: Microsoft Word - VersionFinal.doc (core.ac.uk)

## Unidad I. Propiedades conducentes

**Confiable.** Orientada a preservar la ejecución predecible y correcta del software a pensar de la existencia de defectos no intencionales y otras debilidades, y de cambios de estado no predecibles en el ambiente.

**Protección.** Depende de la propiedad anterior y, por lo general, el hecho de no satisfacerla plantea implicancias muy significativas.



Esta foto de Autor desconocido  
está bajo licencia CC BY-NC

Fuente: [Microsoft Word - VersionFinal.doc \(core.ac.uk\)](#)

## **Unidad I. Aseguramiento del software y de la calidad**

**La relación entre SA y Quality Assurance es también un tema de discusión entre las organizaciones involucradas con la seguridad del software, el desarrollo de software y la calidad del software; y está siendo considerada desde dos direcciones**

Fuente: [Microsoft Word - VersionFinal.doc \(core.ac.uk\)](#)



Esta foto de Autor desconocido está bajo licencia CC BY-SA

## **Unidad I. Aseguramiento del software y de la calidad**

**QA para prácticas de desarrollo de sistemas de información y de software seguro; establece la necesidad de respetar durante el SDLC las pautas del aseguramiento de la calidad en las principales actividades relacionadas con la seguridad, a fin de garantizar que los requerimientos de seguridad se definan adecuadamente y se desarrollen los controles para satisfacerlos.**



Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [Microsoft Word - VersionFinal.doc \(core.ac.uk\)](#)

## **Unidad I. Aseguramiento del software y de la calidad**

**Se identifica un conjunto de métricas para la gestión del riesgo de la seguridad, las que se incluyen en el esquema de trabajo de QA en las diferentes fases del Ciclo de vida del desarrollo de software. .**

Fuente: [Microsoft Word - VersionFinal.doc \(core.ac.uk\)](#)



Esta foto de Autor desconocido está bajo licencia CC BY-SA

## **Unidad I. Guías y estándares para el software seguro**

**La creciente dependencia del software para que se hagan trabajos críticos significa que el valor del software ya no reside únicamente en su capacidad para mejorar o mantener la productividad y la eficiencia. En lugar de ello, su valor también se deriva de su capacidad para continuar operando de forma fiable incluso de cara a los eventos que lo amenazan.**



Esta foto de Autor desconocido está bajo licencia CC BY-SA

**Fuente: [Memoria Final Adriana Matei](#)  
(upm.es)**

## **Unidad I. Guías y estándares para el software seguro**

**La capacidad de confiar en que el software seguirá siendo fiable en cualquier circunstancia, con un nivel de confianza justificado, es el objetivo de la seguridad del software.**



Esta foto de Autor desconocido está bajo licencia CC BY-SA

**Fuente: [Memoria Final Adriana Matei](#)  
[\(upm.es\)](#)**

## **Unidad I. Guías y estándares para el software seguro**

**Garantizar el software se ha convertido en algo fundamental porque el aumento espectacular en los riesgos empresariales y de misión ahora se sabe que son atribuibles al software explotable.**

**El principal objetivo de la seguridad del software es la construcción del software de más calidad, más robusto y libre de defectos que sigue funcionando correctamente bajo ataque malicioso**



Esta foto de Autor desconocido está bajo licencia CC BY-SA

**Fuente: [Memoria Final Adriana Matei \(upm.es\)](#)**

## **Unidad I. Guías y estándares para el software seguro**

Mediante el aprovechamiento de los principios de seguridad, un equipo de desarrollo de software se beneficiará de la orientación de destacados profesionales de la industria y se puede aprender a hacer las preguntas correctas de su arquitectura y diseño de software a fin de evitar los errores más frecuentes y graves. Sin estos principios de seguridad, el equipo se reduce a confiar en el conocimiento sobre la seguridad individual de sus miembros más experimentados.



Esta foto de Autor desconocido está bajo licencia CC BY-SA

Fuente: [Memoria Final Adriana Matei \(upm.es\)](#)

## Unidad I. Definiciones y Taxonomía



[Esta foto de Autor desconocido](#)  
[está bajo licencia CC BY-SA-NC](#)

# ¿Alguna pregunta?