

Adrián Tirado Ramos

1º De DAW

Entorno de Desarrollo

HERRAMIENTAS CASE



# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>2</b>
<b>CLASIFICACIÓN DE LAS HERRAMIENTAS CASE .....</b>	<b>3</b>
CONSIDERANDO SU AMPLITUD: .....	3
BASADA EN LAS FASES DEL CICLO DE DESARROLLO QUE CUBREN: .....	3
<b>VENTAJAS Y DESVENTAJAS .....</b>	<b>4</b>
<b>FUNCIONES MÁS USUALES DE LAS HERRAMIENTAS CASE .....</b>	<b>5</b>
<b>DESARROLLO DE SOFTWARE: .....</b>	<b>5</b>
MODELADO VISUAL: .....	5
GENERACIÓN DE CÓDIGO: .....	5
GESTIÓN DE REQUISITOS: .....	5
<b>PRUEBA DE SOFTWARE: .....</b>	<b>6</b>
GESTIÓN DE PRUEBAS: .....	6
SEGUIMIENTO DE DEFECTOS: .....	6
<b>DOCUMENTACIÓN DE CÓDIGO: .....</b>	<b>6</b>
GENERACIÓN DE DOCUMENTACIÓN AUTOMATIZADA: .....	6
MANTENIMIENTO DE VERSIONES: .....	6
<b>BENEFICIOS ADICIONALES DE LAS HERRAMIENTAS CASE: .....</b>	<b>7</b>
COLABORACIÓN EFICIENTE: .....	7
ANÁLISIS DE IMPACTO: .....	7
CUMPLIMIENTO DE NORMATIVAS: .....	7
<b>CONCLUSIÓN .....</b>	<b>8</b>

## Introducción

Las herramientas CASE (Computer Aided Software Engineering) Ingeniería de Software Asistida por Computadora, son software que brindan soporte automatizado a las actividades de ingeniería de software.

Las herramientas CASE se refieren a la aplicación de un conjunto de herramientas y métodos para incrementar la productividad del desarrollo software y reducir costes de tiempo y dinero, obteniendo un software de alta calidad, sin defectos y mantenible.

## Clasificación de las herramientas CASE

Las herramientas CASE pueden ser clasificadas de acuerdo a los siguientes factores:

- Las plataformas que soportan.
- Las fases del ciclo de vida del desarrollo de sistemas que cubren.
- La arquitectura de aplicaciones que producen.
- Su funcionalidad.

### Considerando su amplitud:

- **TOOLKIT:** Es una colección de herramientas integradas que permiten automatizar un conjunto de tareas de algunas de las fases del ciclo de vida del sistema informático: Planificación estratégica, Análisis, Diseño, Generación de programas.
- **WORKBENCH:** Son conjuntos integrados de herramientas que dan soporte a la automatización del proceso completo de desarrollo del sistema informático. Permiten cubrir el ciclo de vida completo. El producto final aportado por ellas es un sistema en código ejecutable y su documentación.

### Basada en las fases del ciclo de desarrollo que cubren:

- **Upper CASE (U-CASE):** ayudan en las fases de planificación, análisis de requisitos y estrategia del desarrollo, usando, entre otros diagramas UML.
- **Middle CASE (M-CASE):** ayudan en la automatización de tareas en el análisis y diseño de la aplicación.
- **Lower CASE (L-CASE):** ayudan a semi-automatizar la generación de código, crean programas de detección de errores, soportan depuración de programas y pruebas.

Además, automatizan la documentación completa de la aplicación.

## Ventajas y Desventajas

### Ventajas:

- **Productividad Mejorada:** automatizan muchas tareas repetitivas y facilitan la creación de diagramas, modelos y documentación, lo que puede aumentar significativamente la productividad del equipo de desarrollo.
- **Mejora de la Calidad:** al proporcionar funciones de verificación y validación, pueden ayudar a reducir errores y mejorar la calidad del software desarrollado.
- **Consistencia y Estandarización:** promueven la consistencia y estandarización en el desarrollo de software.

Los modelos y documentos generados son coherentes y siguen estándares predefinidos.

- **Gestión de Cambios Efectiva:** facilitan el seguimiento de cambios en el código, documentos y modelos, lo que simplifica la gestión de versiones y la implementación de cambios.
- **Facilitan la Colaboración:** Al permitir que varios miembros del equipo trabajen simultáneamente en diferentes aspectos del desarrollo, facilitan la colaboración y la comunicación entre los miembros del equipo.

### Desventajas:

- **Costo Inicial y de Mantenimiento:** a menudo requieren una inversión significativa en términos de licencias, formación y mantenimiento, lo que implica que puede ser un obstáculo financiero para algunas organizaciones.
- **Curva de Aprendizaje:** el aprendizaje y la adaptación a nuevas herramientas pueden llevar tiempo.

Los equipos pueden experimentar una disminución temporal en la productividad mientras se familiarizan con las características y funcionalidades de las herramientas.

- **Personalización Limitada:** Algunas herramientas pueden tener limitaciones en cuanto a la personalización para adaptarse a los procesos específicos de una organización, lo que puede resultar en la necesidad de ajustar los procesos para adaptarse a la herramienta.
- **Dependencia de Proveedores:** algunas organizaciones pueden volverse dependientes de un proveedor específico de herramientas, lo que puede limitar su flexibilidad y libertad para cambiar a otras soluciones en el futuro.
- **Complejidad:** a veces pueden ser demasiado complejas para proyectos pequeños o equipos sin experiencia técnica suficiente, lo que puede llevar a su subutilización.

## Funciones más usuales de las herramientas CASE

### Desarrollo de Software:

Las herramientas CASE ofrecen funcionalidades clave para el desarrollo de software, que incluyen:

#### Modelado Visual:

**Diagramas de Secuencia:** Permite visualizar la interacción entre diferentes partes del sistema a lo largo del tiempo.

Diagramas de Actividad: Facilita la representación gráfica de procesos y actividades, mejorando la comprensión del flujo de trabajo.

#### Generación de Código:

**Plantillas Personalizadas:** Algunas herramientas CASE permiten crear plantillas personalizadas para la generación de código, adaptándose a los estándares específicos del proyecto.

#### Gestión de Requisitos:

**Tracerabilidad:** Ofrece un seguimiento claro y detallado de cómo cada requisito se vincula con los componentes del sistema, mejorando la visibilidad y la gestión del impacto de los cambios.

## Prueba de Software:

En el ámbito de la prueba de software, las herramientas CASE suelen ofrecer las siguientes funciones:

### Gestión de Pruebas:

**Automatización de Pruebas:** Integra la automatización de pruebas, lo que permite la ejecución rápida y repetitiva de casos de prueba, mejorando la eficiencia del proceso.

### Seguimiento de Defectos:

**Integración con Herramientas de Gestión de Proyectos:** Algunas herramientas CASE permiten la integración con herramientas de gestión de proyectos, simplificando la comunicación entre equipos de desarrollo y prueba.

## Documentación de Código:

Para la documentación de código, las herramientas CASE brindan características como:

### Generación de Documentación Automatizada:

**Inclusión de Comentarios:** Permite la generación automática de comentarios en el código, mejorando la legibilidad y facilitando la comprensión del mismo.

### Mantenimiento de Versiones:

**Historial de Cambios:** Proporciona un historial detallado de los cambios realizados en el código y la documentación, lo que facilita la identificación de responsables y la reversión a versiones anteriores si es necesario.



## Beneficios Adicionales de las Herramientas CASE:

### Colaboración Eficiente:

**Colaboración en Tiempo Real:** Facilita la colaboración entre miembros del equipo en tiempo real, permitiendo la revisión y la modificación conjunta de artefactos.

### Análisis de Impacto:

**Análisis de Impacto de Cambios:** Permite evaluar cómo los cambios propuestos afectarán al sistema, minimizando posibles efectos secundarios no deseados.

### Cumplimiento de Normativas:

**Generación de Informes de Conformidad:** Ayuda en la generación automática de informes que demuestran el cumplimiento de estándares y normativas específicas.

## Conclusión

En resumen, las herramientas CASE ofrecen un conjunto diverso de funciones que abarcan el ciclo de vida completo del desarrollo de software, desde la concepción hasta el mantenimiento.

Estas funciones están diseñadas para aumentar la eficiencia y la calidad en todas las etapas del proceso de ingeniería de software.