

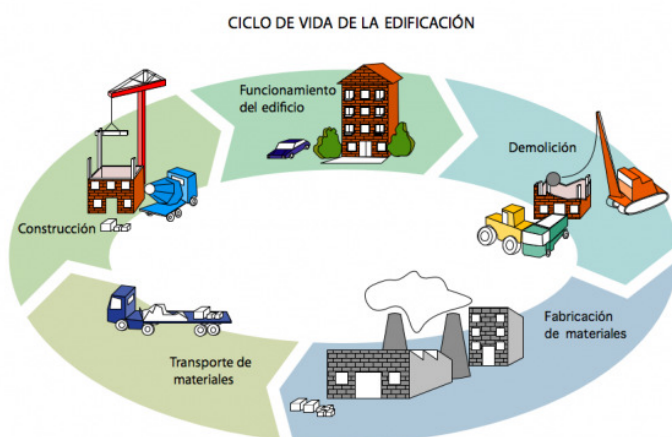
# Sistemas de Información

Todos los sistemas de información, para que puedan ser una realidad y que finalmente sigan persistiendo en el tiempo, requieren dar cumplimiento a una serie de etapas que se conocen como el ciclo de vida del software. Estas fases, que se van sucediendo secuencialmente, son necesarias para seguir adaptando los sistemas a los constantes cambios que sufre desde su entorno. Cambios solicitados por los usuarios, imposiciones del estado (por ejemplo nuevos impuestos), cambios en las regiones (monedas comunes – Euros), etc.



Aquí se muestran esas etapas. El término Ciclo de vida del software determina el desarrollo de un software desde la fase inicial hasta la fase final. El objetivo principal es definir las fases o etapas intermedias que requieren para validar el desarrollo de la aplicación, es decir, garantizar que el software cumpla con los requisitos definidos por los usuarios.

Se lo conoce como ciclo pues ni bien llegamos a la última etapa, nuevos requerimientos o adaptaciones de la aplicación implica volver a empezar y así indefinidamente para siempre mantener los sistemas en forma persistente, cumpliendo con los requerimientos definidos por los usuarios.



En este ejemplo observamos las fases del ciclo de vida de la edificación, al comparar con la anterior podemos decir que:

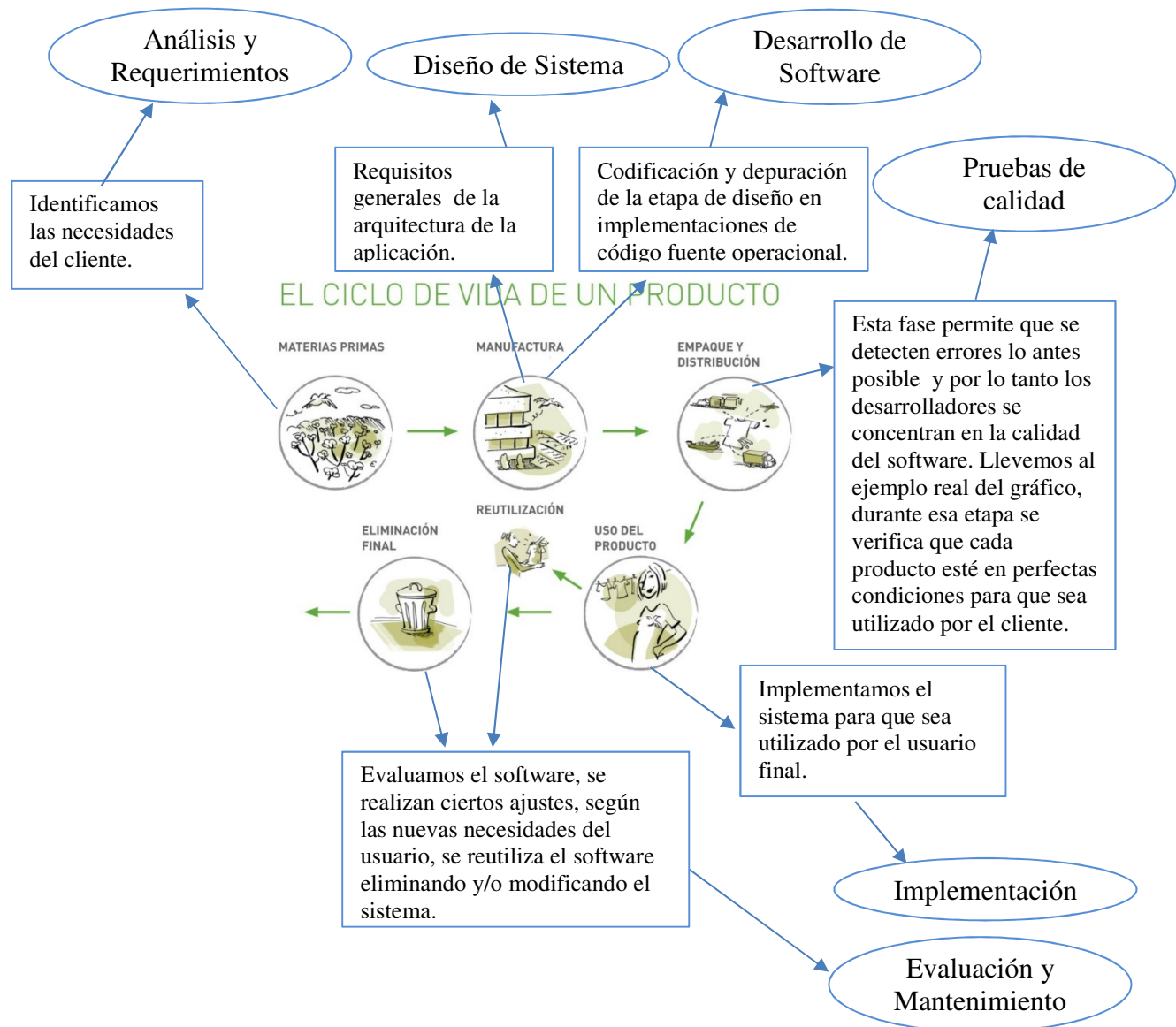
Fase 1: Análisis y Requerimientos = Demolición.

Fase 2: Diseño de Sistema = Fabricación de Materiales.

Fase 3: Desarrollo de Software = Transporte de materiales y Construcción.

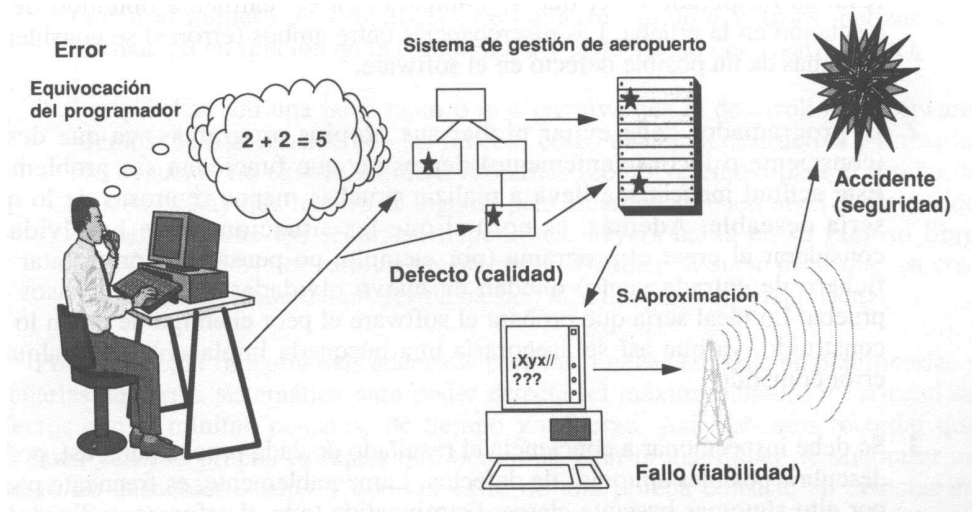
Fase 4: Pruebas de Calidad e Implementación y Evaluación = Funcionamiento del Edificio.

Por consiguiente decimos que el ciclo de vida de un software:



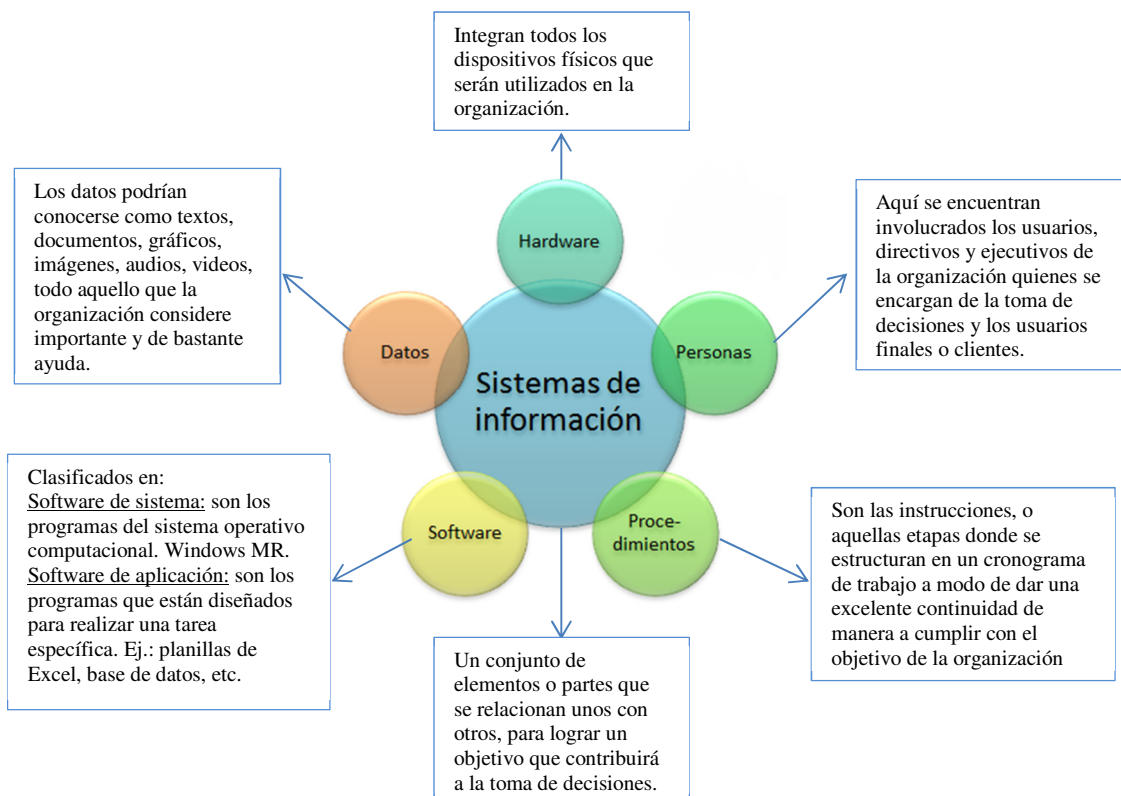
A través de estas presentaciones se obtuvo las fases más importantes del ciclo de vida de un software. En otras investigaciones u otros autores de libros presentan un ciclo de vida con fases más detalladas, es decir, las mismas fases que muestra este apartado pero con la necesidad de volver a dividir en fases más pequeñas según se crea conveniente mencionando que cada ingeniero de software opta y formula sus propios métodos de trabajo.

## Consecuencias de posibles errores



Cuando se manifiesta un error en el sistema, el encargado de resolver esta situación es el programador o comúnmente llamado desarrollador. En este apartado observamos como el programador recorre ciertos caminos buscando el posible punto focal de error lo que deja fallas en el sistema o producen cálculos con operaciones fallidas o interminables.

## Componentes de los sistemas de información

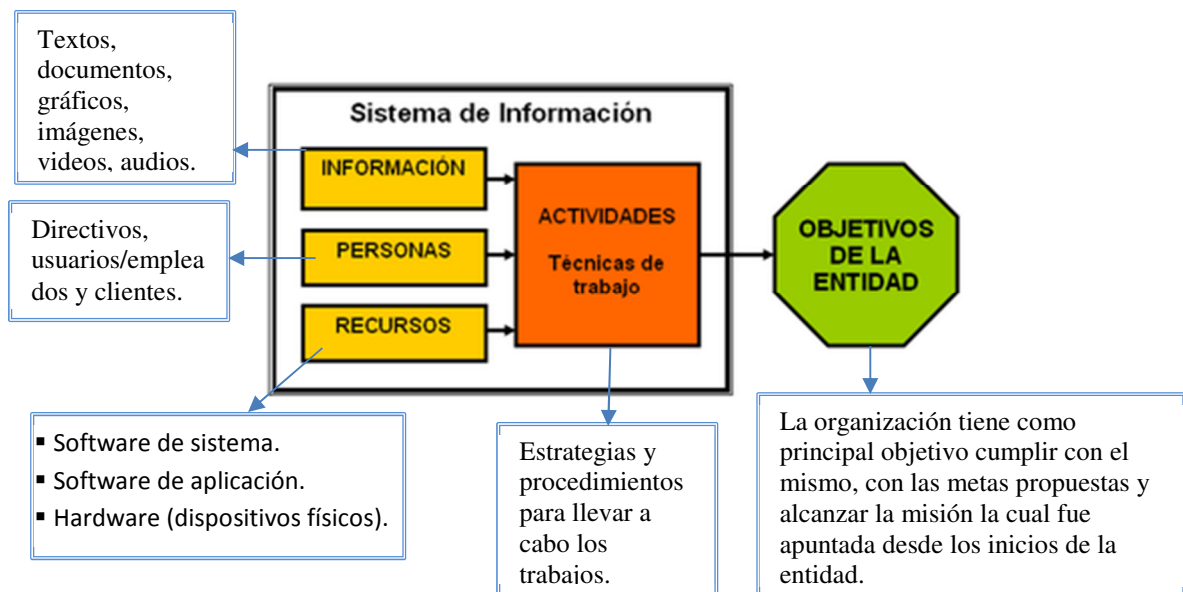


## Los sistemas de información y la influencia de su entorno



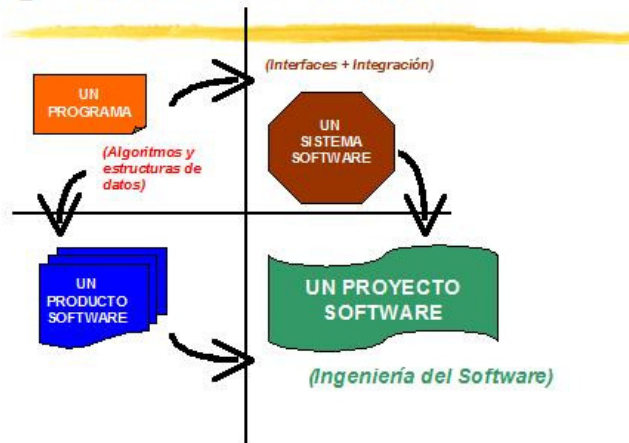
La organización define sus metas, los valores importantes con que trabajarán y la misión a la cual apuntarán; las estrategias se forman a partir del ¿Qué? y ¿Cómo? consolidarán sus debidas obligaciones laborales cada directivo y/o empleado según su área de trabajo. Al referirnos de procesos de negocios decimos que son aquellas tareas que se encuentran relacionadas entre sí y se encargan de proveer los productos y servicios; cuando mencionamos a los indicadores del desempeño apuntamos a medir las actividades que producen la información que los procesos de negocios se encargan de proveer, las mejora y evalúa a través de métricas (clasificar, analizar) de seguimientos.

## Sistemas de información en un esquema tradicional



# Ingeniería del Software

## Ingeniería de software



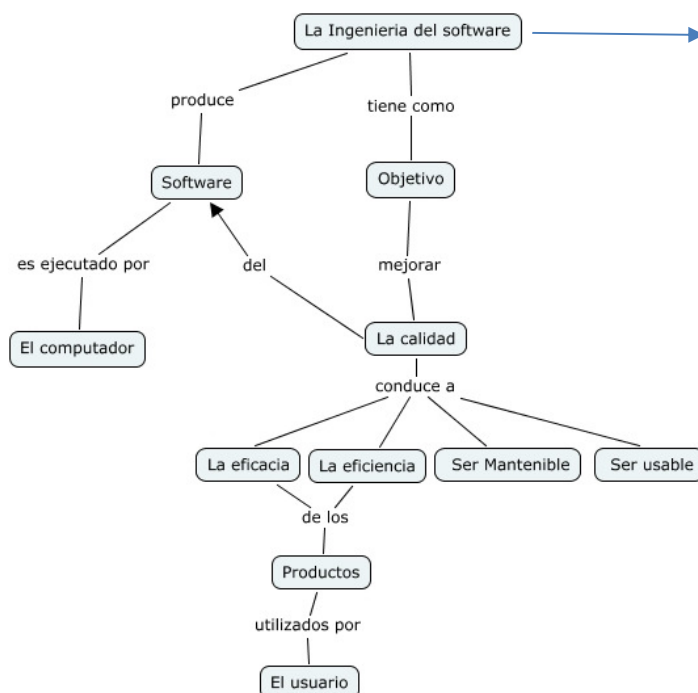
*En un programa o algoritmo*, describimos las instrucciones o secuencias de pasos que permitirán llegar a una solución.

*Un producto software*, podemos nombrar a un conjunto de algoritmos que formarán un sistema el cual será utilizable por un usuario.

*Un sistema software*, es la integración y la presentación gráfica de todos los algoritmos funcionales.

*Un proyecto software*, es el sistema funcional.

Abordar la creación de sistemas informáticos desde el concepto mismo de la Ingeniería del Software nos permitirá dar cumplimiento a tiempo, costos, requisitos, calidad, reusabilidad y todo componente que hace al ejecutable, cuya creación fundada en el uso de la ciencia, sea previsible y duradero en el tiempo.



Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo de la operación. Con ella se puede analizar, diseñar, programar y aplicar un software de manera correcta y organizada cumpliendo con todas las especificaciones o requerimientos del cliente.

Para ello es importante mejorar la calidad de los productos y aumentar la productividad y el trabajo de los ingenieros en software.