

## Material de estudio OBLIGATORIO EJE TEMATICO 1: Metodología en el Desarrollo de Software

Sitio: [Instituto Superior Politécnico Córdoba](#)

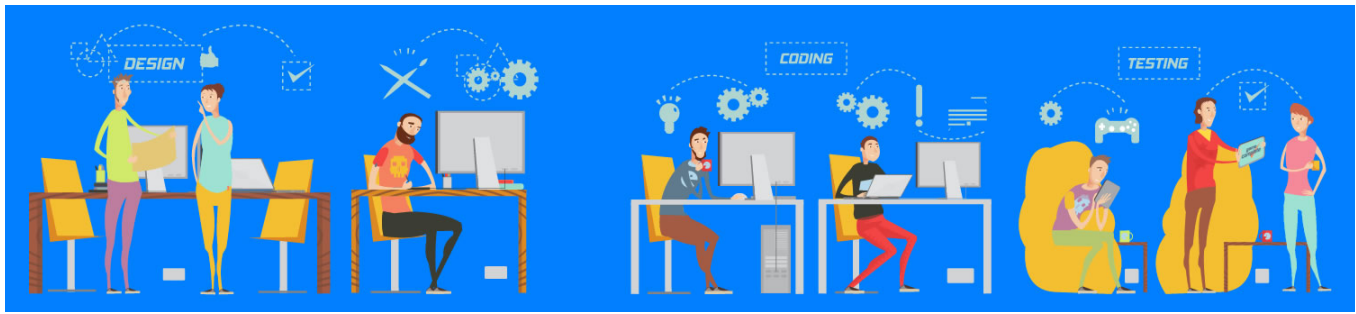
Curso: Programador Web - TSDWAD - 2022

Libro: Material de estudio OBLIGATORIO EJE TEMATICO 1:  
Metodología en el Desarrollo de Software

Imprimido por: Ezequiel Maximiliano GIAMPAOLI

Día: miércoles, 5 abril 2023, 5:42 PM

## Descripción



Vector de desarrollo web creado por macrovector - [www.freepik.es](http://www.freepik.es)

## Tabla de contenidos

### **1. Metodología de Desarrollo de Software**

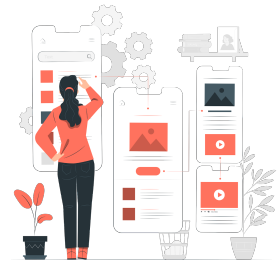
- 1.1. Ciclo de Vida
- 1.2. Procesos/Modelos
- 1.3. Resolución de problemas a través de la computadora
- 1.4. Metodología Tradicional vs Metodologías Ágiles

### **2. Scrum**

### **3. Referencias**

## 1. Metodología de Desarrollo de Software

Según la IEEE, el software es el “**conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados, que forman parte de las operaciones de un sistema de computación**”. Es la parte de un sistema que se puede **codificar** para ejecutarse en una computadora como un conjunto de instrucciones, e incluye la **documentación** asociada necesaria para comprender, transformar y usar esa **solución**. Estos documentos describen la **organización del sistema**, explican al usuario **cómo utilizarlo** y obtener eventuales actualizaciones del producto.



## 1.1. Ciclo de Vida

Un ciclo de vida es el conjunto de **fases [o procesos]** por las que pasa el sistema de software desde que se concibe [o inicio], se desarrolla hasta que se retira del servicio finalizando su uso.

Las fases o procesos están estandarizados, es decir que existe un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, explotación y mantenimiento de un producto de software, abarcando la vida del sistema, desde la definición de requerimientos hasta la finalización de su uso.

*Leer el siguiente material y busquen otros recursos para luego debatir las etapas fundamentales:*

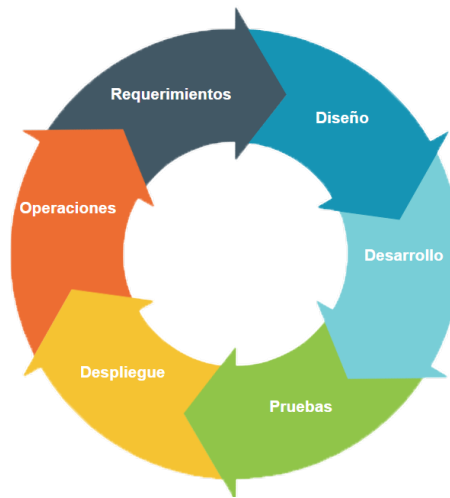
<https://ungoti.com/es/soluciones/desarrollo-de-software/sdlc/>

<https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>

## 1.2. Procesos/Modelos

### Proceso para el desarrollo de software

Define un conjunto completo de actividades necesarias para transformar los requerimientos de un usuario en un producto.



Existen varios procesos o modelos de desarrollo y podríamos catalogarlos en **Secuenciales o Definidos**, y en **Empíricos**. Los primeros responden siempre de la misma forma ante una determinada entrada, mientras que los segundos dependen de la evolución y adaptación de etapas anteriores.

#### Definidos o secuenciales



#### Empíricos

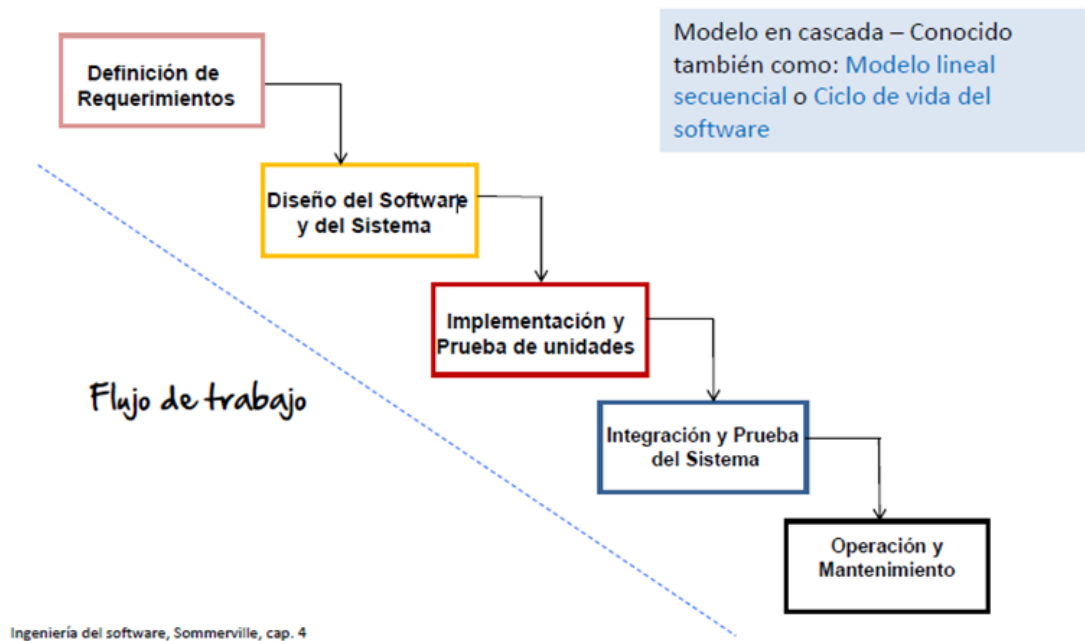


#### Modelo en Cascada

Su principal característica es que cada fase del desarrollo está bien definida y separada, siguiendo un orden secuencial en el que cada etapa depende de la finalización de la anterior y que esta última haya pasado un proceso de validación que apruebe el paso a la siguiente.

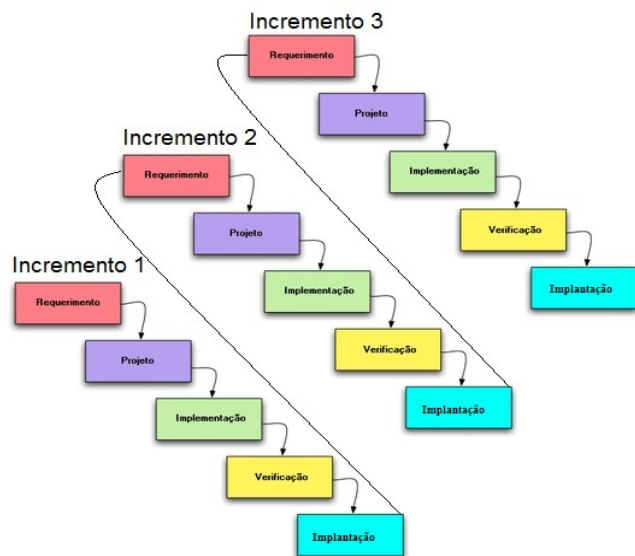
Comúnmente las etapas del modelo son:

- Análisis y definición de los requerimientos.
- Diseño del sistema y del software.
- Implementación y pruebas unitarias.
- Integración y pruebas de sistema.
- Funcionamiento y mantenimiento.



### Metodología Iterativa e Incremental

La metodología iterativa e incremental deriva del proceso de desarrollo en cascada pero, con la diferencia de que aquí se admite que las etapas se solapan en tiempo con la finalidad de flexibilizar el tiempo de desarrollo total y así poder alcanzar resultados funcionales de manera temprana.



### 1.3. Resolución de problemas a través de la computadora

#### Análisis: ¿Qué ...?

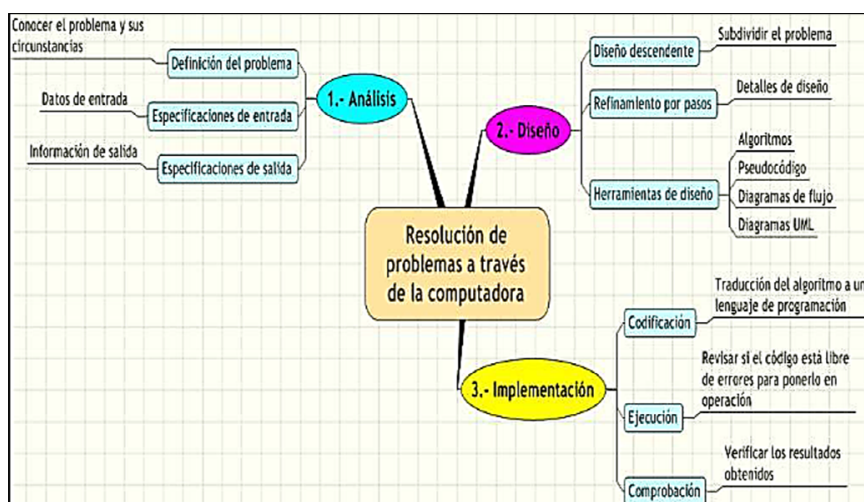
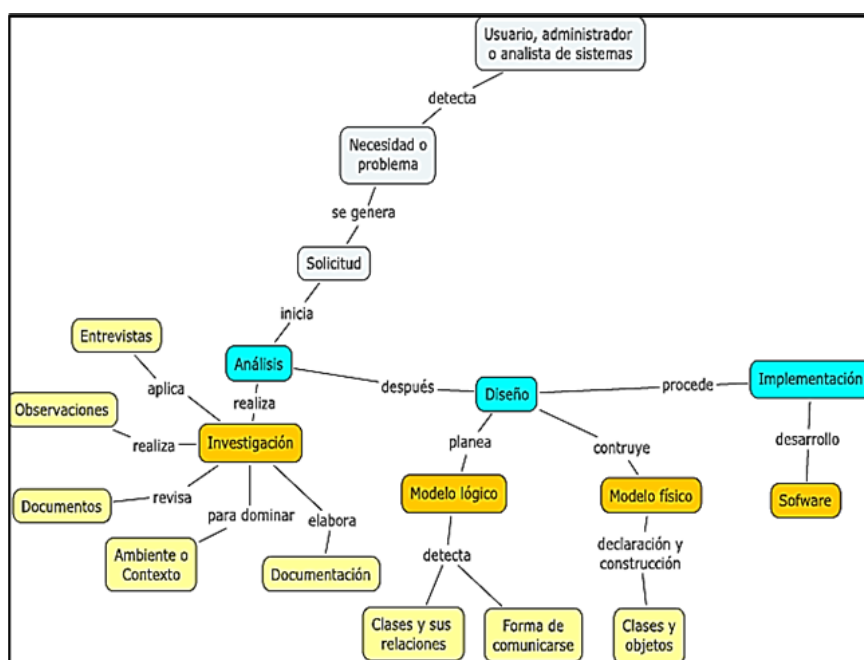
- ¿Qué problema debe resolverse?
- ¿Qué datos se requieren?
- ¿Qué resultados debe arrojar el Sistema?

#### Diseño: ¿Cómo ...?

- ¿Cómo atacar el problema?
- ¿Cómo plantear el modelo de solución?
- ¿Cómo aplicar el modelo de solución?

#### Implementación: ¿Con qué ...?

- ¿Con qué lenguaje se desarrolla el modelo?
- ¿Con qué plataforma de desarrollo?
- ¿Con qué recursos de hardware y software?







## 1.4. Metodología Tradicional vs Metodologías Ágiles

La gestión ágil, es cada vez más popular como una alternativa a las metodologías tradicionales de gestión de proyectos, especialmente entre los equipos de desarrollo de software y TI. Cuando intenta decidir qué metodología de gestión de proyectos utilizar, es importante comprender algunas de las diferencias clave entre los enfoques de gestión de proyectos ágiles y tradicionales, ya que cada uno tiene sus propias ventajas y desventajas.

### Metodologías tradicionales

La gestión de proyectos tradicional se centra en el enfoque lineal siendo la más usada es el modelo en **cascada o waterfall**,

Se caracteriza porque todas las fases se realizan en forma secuencial, es decir, que las etapas se llevan a cabo una detrás de otra, en este enfoque, los requisitos son fijos, y el presupuesto y el tiempo se acuerdan antes. Por esta razón, los equipos a menudo enfrentan problemas de presupuesto y cronograma con este enfoque. No puede utilizar la gestión de proyectos tradicional para desarrollar productos complejos, ya que este enfoque no deja espacio para cambiar los requisitos.

### Metodologías ágiles

En la gestión ágil de proyectos, los proyectos se dividen en iteraciones cortas. La iteración dura un máximo de un mes calendario. Y después de cada iteración, obtendrá un nuevo incremento de producto liberable. La gestión ágil de proyectos se centra más en implementar los comentarios del cliente y revisar el producto periódicamente. La colaboración con el cliente es un factor vital en la agilidad. No sigue un plan a ciegas y responde a los cambios rápidamente, estableciendo un marco que permite acortar los tiempos de desarrollo, eliminar la incertidumbre, mejorar la eficiencia del equipo y obtener el resultado deseado. Entre los avances promovidos por Agile se encuentran la reducción de la burocracia y la microgestión en los proyectos, la rápida adaptación a los cambios de orientación, la reducción del tiempo de comercialización de una funcionalidad, producto o servicio, así como también la eliminación de residuos mediante ciclos frecuentes de pruebas y validación

Por ejemplo, **Scrum, Kanban, LeSS, SAFe y Scrumban** son excelentes ejemplos de métodos populares de gestión de proyectos ágiles.

**Ref.** <https://www.linkedin.com/pulse/metodolog%C3%ADas-tradicionales-vs-%C3%A1giles-jos%C3%A9-gabald%C3%B3n/?originalSubdomain=es>

## 2. Scrum

### Aplicando Scrum en nuestros Proyectos de Software

El **ciclo de Scrum** empieza con una reunión de *stakeholders*, durante la cual se crea la visión del proyecto. Después, el *Product Owner* desarrolla un Backlog Priorizado del Producto (*Prioritized Product Backlog*) que contiene una **lista requerimientos del negocio** y del proyecto por orden de importancia en forma de una historia de usuario. Cada **sprint** empieza con una reunión de planificación del sprint (*Sprint Planning Meeting*) durante la cual se consideran las *historias de usuario de alta prioridad* para su inclusión en el sprint. Un sprint generalmente tiene una duración de una a tres semanas durante las cuales el Equipo Scrum trabaja en la creación de entregables (del inglés deliverables) en incrementos del producto.

Durante el sprint, se llevan cabo *Daily Standups* muy breves y concretos, donde los miembros del equipo discuten el *progreso diario*. Hacia el final del sprint, se lleva a cabo una Reunión de Revisión del Sprint (*Sprint Review Meeting*) en la cual se proporciona una *demonstración* de los entregables al Product Owner y a los stakeholders relevantes.

El Product Owner acepta los entregables sólo si cumplen con los criterios de aceptación predefinidos. El ciclo del sprint termina con una Reunión de Retrospectiva del Sprint (*Retrospect Sprint Meeting*), donde el equipo analiza las formas de mejorar los procesos y el rendimiento a medida que avanzan al siguiente sprint.

Ref. <https://www.linkedin.com/pulse/metodolog%C3%ADas-tradicionales-vs-%C3%A1giles-jos%C3%A9-gabald%C3%B3n/?originalSubdomain=es>

### Scrum

Scrum es un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, a la vez que se entregan productos de forma eficiente y creativa con el máximo valor. En scrum se aplican un conjunto de buenas prácticas para trabajar de manera colaborativa y obtener así el mejor resultado posible en proyectos complejos que demanden constantes cambios y adaptaciones.

### PILARES

Uno de los **pilares de Scrum** es la **transparencia**. Los aspectos significativos del proceso deben ser visibles para todos aquellos que son responsables del resultado. Esto requiere que dichos aspectos sean definidos en base a un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que están viendo. Por ejemplo:

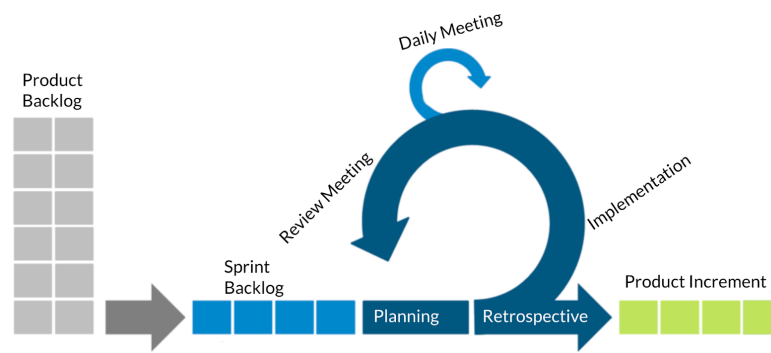
- Deben compartir un lenguaje común todos los participantes para referirse al proceso; y,
- Aquellos que desempeñan el trabajo y quienes inspeccionan el incremento resultante deben compartir una definición común de "Terminado" ("Done").

Otro de los pilares es la **inspección**. Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo para detectar variaciones indeseadas. Su inspección no debe ser tan frecuente como para que pueda interferir en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos en el mismo lugar de trabajo.

Finalmente, otro pilar de la metodología es su **adaptación**. Si un inspector determina que uno o más aspectos de un proceso se desvían de los límites aceptables y que el producto resultante será inaceptable, el proceso o el material que está siendo procesado deben ajustarse. Dicho ajuste deberá realizarse cuanto antes para minimizar desviaciones mayores.

Los **proyectos se llevan a cabo en equipos que trabajan en ciclos temporales cortos y de duración fija**. Cada ciclo se conoce como **Sprint** y tiene que proporcionar un resultado completo, un incremento del producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando este lo solicite. Los componentes del ciclo son:

- **Eventos**: Planificación del sprint, Scrum diario, Revisión del sprint, y Retrospectiva del sprint
- **Roles**: Desarrollador, Propietario del producto, y Scrum master
- **Artefactos**: Pila del producto, Pila del sprint, e Incremento.



### 3. Referencias

Sitios Web consultados:

<https://ungoti.com/es/soluciones/desarrollo-de-software/sdlc/>

<https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>