



# MODELOS DE SOFTWARE Y EL MODELO EN CASCADA

David Flórez Quintero  
Martha Rico Rocha  
Jhon Alexander Corredor



# ¿QUÉ ES UN MODELO DE SOFTWARE?

Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociados entre estas etapas.





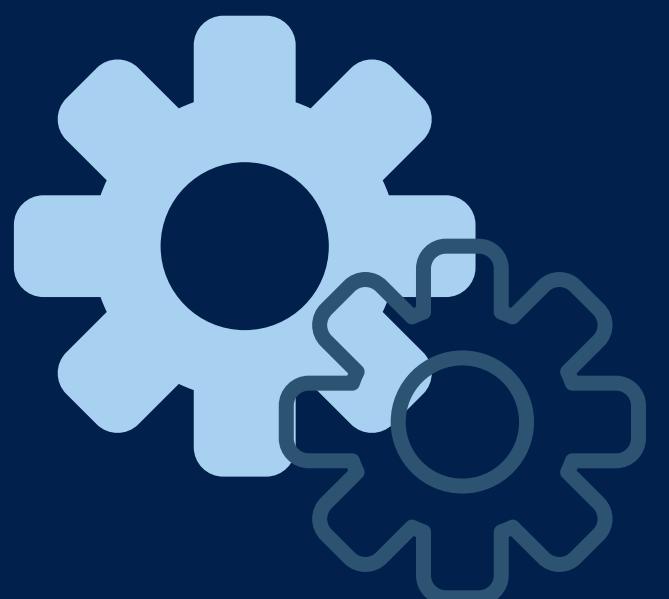
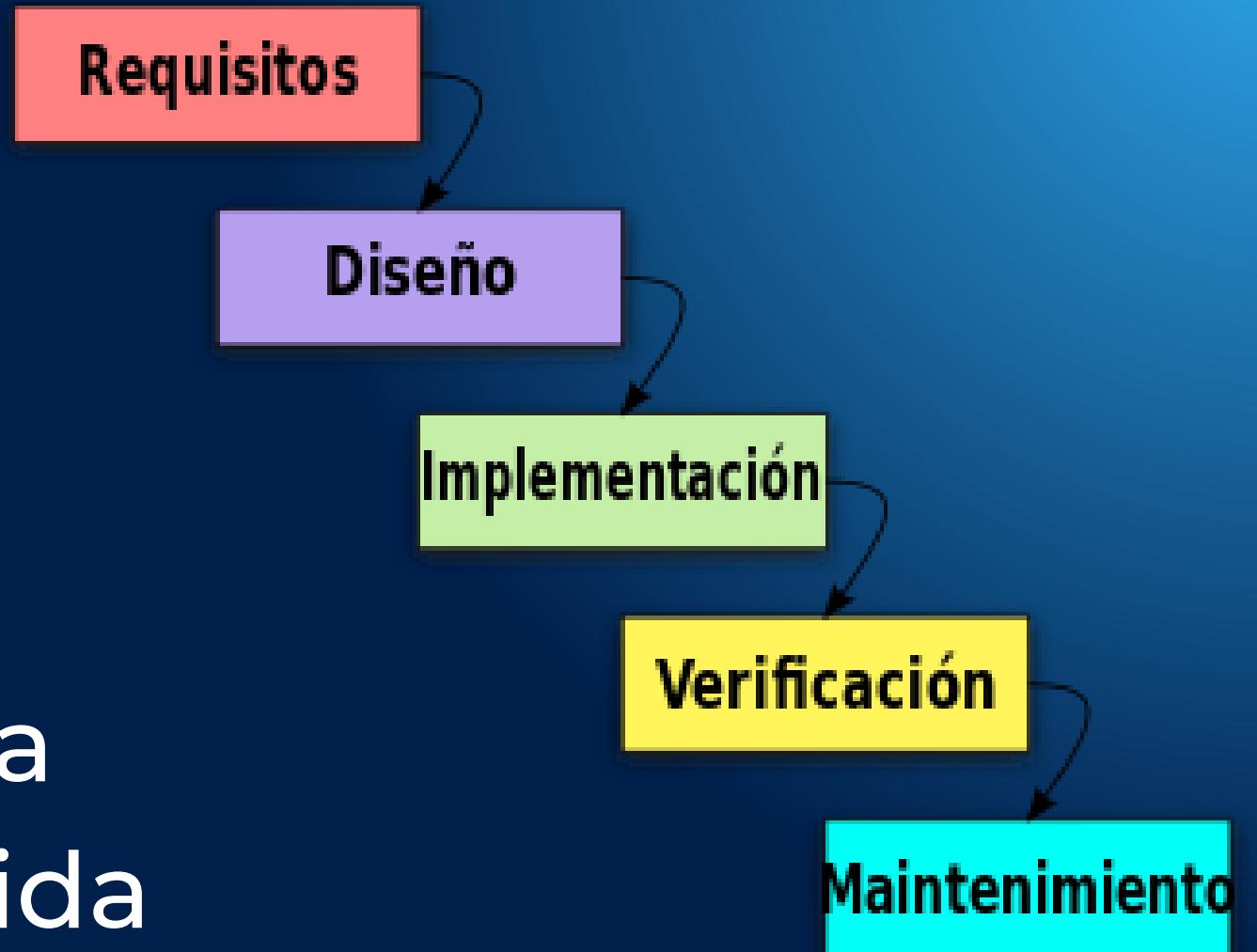
# CARACTERÍSTICAS

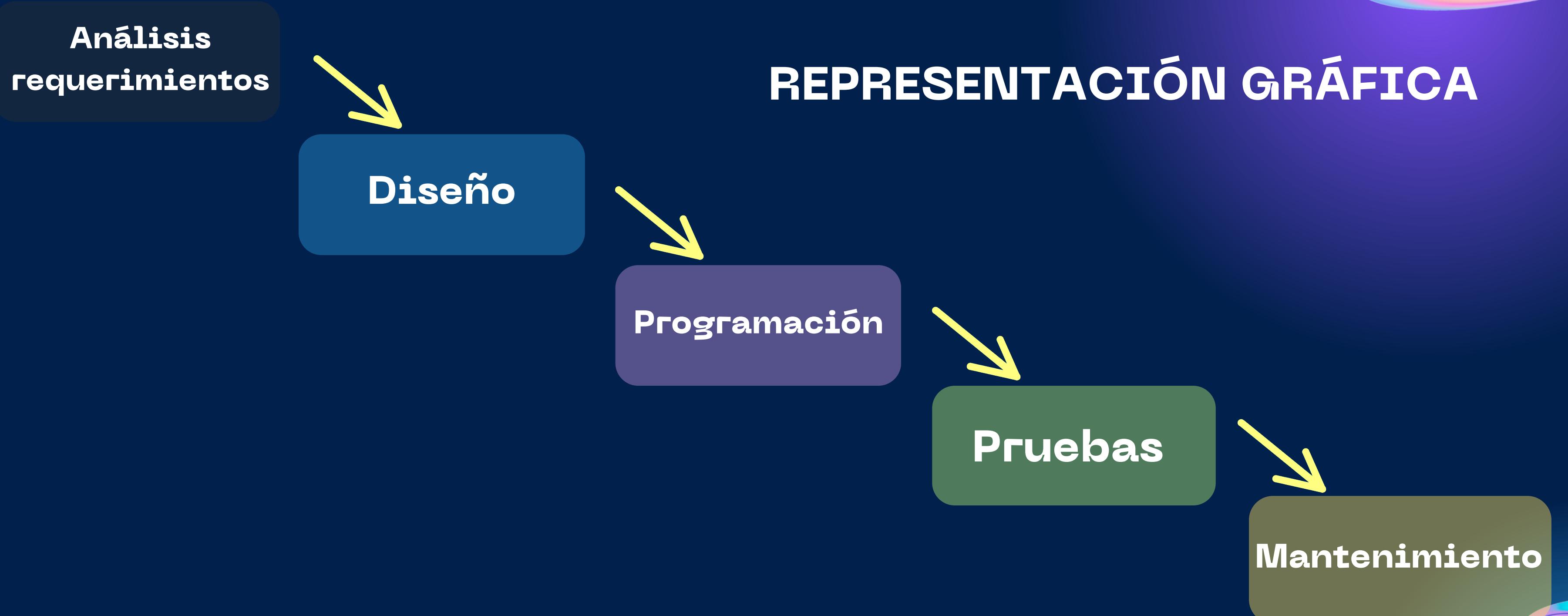
## Un modelo de ciclo de vida del software:

- Describe las fases principales de desarrollo de software
- Define las fases primarias esperadas de ser ejecutadas durante esas fases
- Ayuda a administrar el progreso del desarrollo
- Provee un espacio de trabajo para la definición de un proceso detallado de desarrollo de software

# MODELO DE CASCADA

Es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.





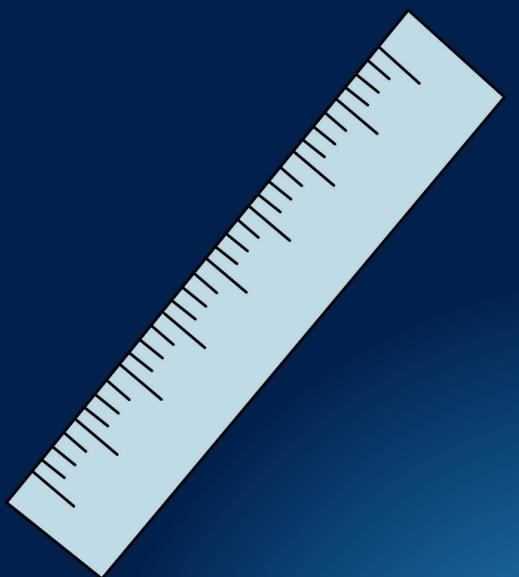
# VENTAJAS DEL MODELO DE CASCADA

- Es apropiado para proyectos pequeños donde los requisitos están bien entendidos.
- Es simple y fácil de utilizar.
- Es fácil de gestionar debido a que cada fase tiene entregables específicos.



# DESVANTAJA DEL MODELO DE CASCADA

- Mala implementación del modelo si el proyecto no tiene una secuencia lineal.
- Es muy restrictivo y no permite movilizarse entre fases .
- Los resultados y/o mejoras no son visibles progresivamente, el producto se ve cuando ya está finalizado.
- Es un modelo pobre respecto a otros para proyectos complejos .



# Caso Práctico - Modelo en Cascada

Imaginemos un proyecto de desarrollo de software para crear un sistema de gestión de inventario para un taller. En el modelo en cascada, el proceso se llevaría a cabo de la siguiente manera:

## 1. Requisitos

En esta fase, se recopilan y documentan todos los requisitos del sistema, como la capacidad de seguimiento de inventario, la gestión de pedidos y la generación de informes.

## 2. Diseño

Con base en los requisitos recopilados, se crea un diseño detallado del sistema, incluyendo la arquitectura de software, la estructura de bases de datos y la interfaz de usuario.

### **3. Implementación:**

El equipo de desarrollo comienza a codificar el software según el diseño especificado. Se crean módulos para el seguimiento de inventario, la gestión de pedidos, etc.

### **4. Pruebas**

Una vez que se ha implementado el software, se llevan a cabo pruebas exhaustivas para garantizar que cumple con los requisitos y funciona correctamente.

### **5. Despliegue**

Después de las pruebas exitosas, el sistema se despliega en el entorno de producción de la tienda minorista.

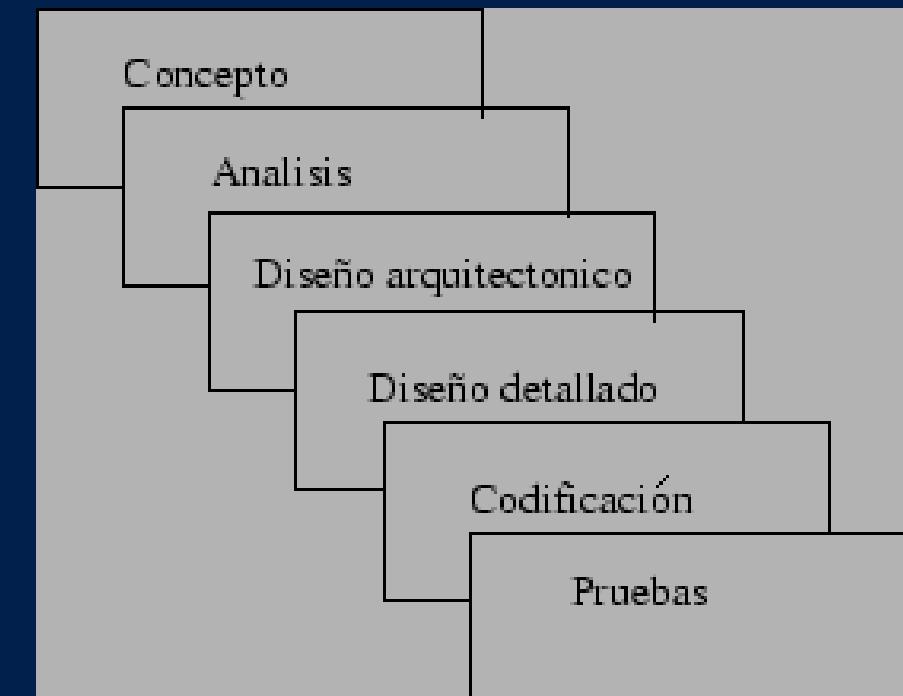
### **6. Mantenimiento**

Finalmente, se realiza el mantenimiento continuo y las actualizaciones según sea necesario.

# Variantes

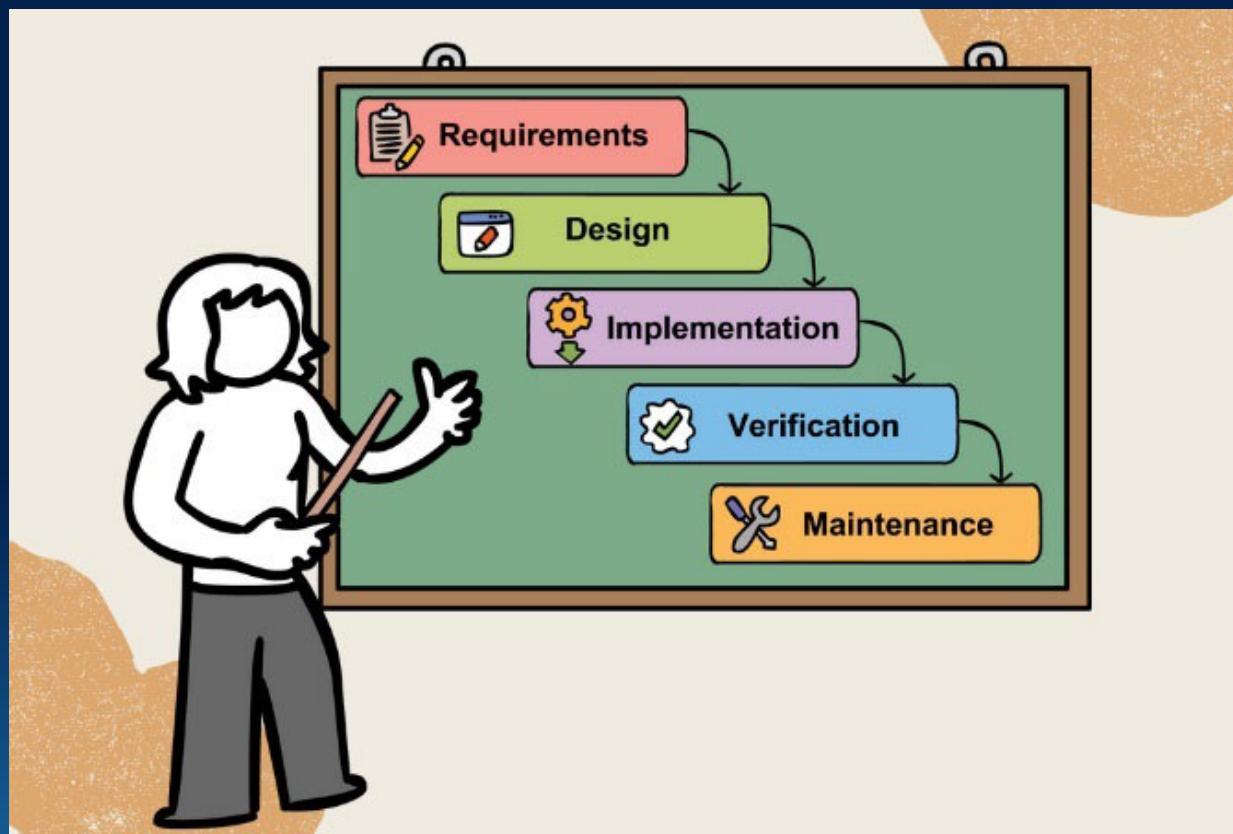
## Modelo Sashimi

El modelo sashimi fue creado por Peter DeGrace. A veces se hace referencia a él como el modelo en cascada con fases superpuestas o el modelo en cascada con retroalimentación. Ya que las fases en el modelo sashimi se superponen, lo que implica que se puede actuar durante las etapas anteriores.



# CONCLUSIÓN

Ha sido ampliamente utilizado en la ingeniería de software a lo largo de los años. Sus ventajas incluyen su simplicidad y organización, lo que lo hace adecuado para proyectos estables con requisitos bien definidos.



Sin embargo, presenta importantes inconvenientes, como su rigidez ante cambios en los requisitos, su falta de visibilidad progresiva de resultados y su dificultad para adaptarse a proyectos no triviales donde los requisitos pueden cambiar con frecuencia.