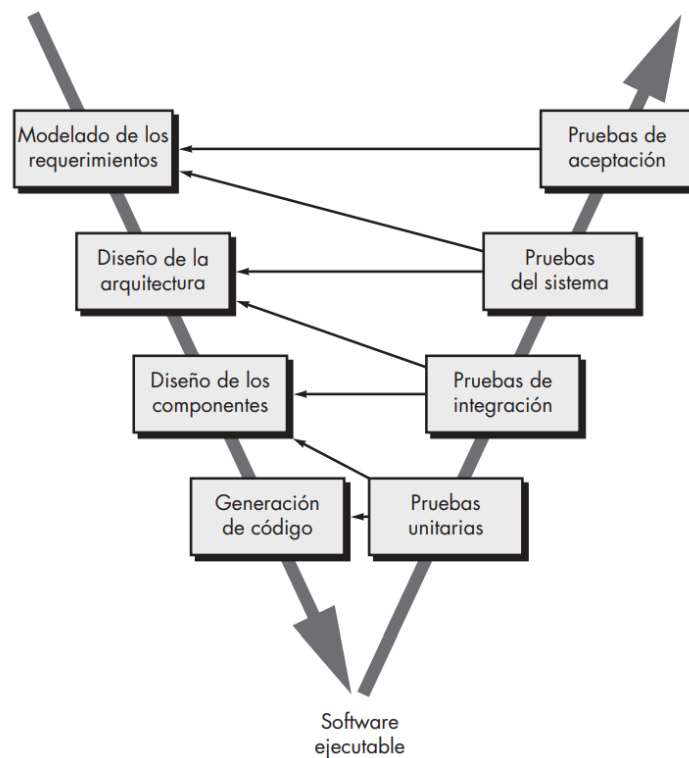


# Paradigma en V

## ¿En qué consiste?

El **paradigma del ciclo de vida en V** es un modelo de desarrollo de software y sistemas que se centra en la **validación y verificación** a lo largo de las diferentes etapas del desarrollo. Este modelo estructura el proceso en forma de una "V" para representar el flujo de las actividades de diseño y pruebas. A diferencia del modelo en cascada, que sigue un enfoque lineal y secuencial, el modelo en V permite realizar pruebas paralelas y continuas durante el desarrollo, promoviendo así la detección temprana de errores.



## Etapas del paradigma del ciclo de vida en V

### Vertiente Izquierda (Fase de Desarrollo)

1. **Requisitos del Sistema:** Se definen los requisitos generales que debe cumplir el sistema. Incluye tanto requisitos funcionales como no funcionales (rendimiento, seguridad, etc.).
2. **Especificación de Requisitos de Software:** Se especifican los requisitos detallados del software. Aquí se desglosan en funcionalidades individuales y sus dependencias, facilitando un diseño más detallado.
3. **Diseño de Arquitectura:** Se establece la arquitectura del sistema, dividiendo el software en módulos y especificando sus interacciones. Este diseño proporciona un esquema general para todo el sistema y facilita su desarrollo modular.

4. **Diseño de Módulos:** Los módulos definidos en la fase de diseño de alto nivel se descomponen en componentes más pequeños y específicos. Este diseño detallado describe cómo cada módulo debe comportarse internamente, incluyendo la lógica de control y los datos específicos.

### **Vertiente Derecha (Fase de Pruebas)**

1. **Pruebas Unitarias:** En las pruebas unitarias se verifica el funcionamiento de cada módulo o componente de forma aislada, asegurándose de que cada uno cumpla con su propósito individual.
2. **Pruebas de Integración:** Se verifica la interacción entre diferentes módulos del sistema para asegurar que trabajan de forma conjunta según lo planeado en la arquitectura.
3. **Pruebas de Sistema:** Se evalúa el sistema completo para verificar que cumpla con los requisitos funcionales y no funcionales especificados.
4. **Pruebas de Aceptación:** Se realizan pruebas finales para asegurar que el sistema cumple con los requisitos del cliente y es aceptado para su implementación.

## **Historia**

El modelo en V tiene sus raíces en la evolución de los modelos de desarrollo de software que buscaban mejorar la **calidad** y **seguridad** de los productos desarrollados. Su desarrollo responde a la necesidad de un enfoque estructurado para proyectos de alta criticidad, y fue una extensión natural del **modelo en cascada**.

### **Orígenes en el Modelo en Cascada**

El **modelo en cascada**, fue el primer enfoque sistemático para el desarrollo de software y estableció las bases para la mayoría de los modelos posteriores, incluido el modelo en V. La idea detrás del modelo en cascada era seguir una secuencia rígida de etapas donde cada fase dependía de la conclusión de la anterior. Sin embargo, este enfoque lineal presentaba una gran desventaja: el proceso de pruebas se dejaba al final, lo que dificulta la corrección de errores descubiertos en fases avanzadas, incrementando el costo y tiempo de desarrollo.

### **Evolución hacia el Modelo en V en los Años 1980**

El **modelo en V** fue desarrollado en los **80 's** como una mejora del modelo en cascada, integrando un enfoque de pruebas paralelo y anticipado. Esto significa que, en lugar de esperar hasta el final del desarrollo para probar todo el sistema, el modelo en V propone diseñar las pruebas desde el inicio y llevarlas a cabo en paralelo a las etapas de diseño. Esto resultó ser particularmente efectivo en **industrias donde la confiabilidad es crítica**, como la aeroespacial y militar, donde se integró en las normas de calidad y desarrollo de sistemas.

### **Expansión en la Industria y Estándares de Calidad**

A partir de los **90 's**, el modelo en V fue adoptado en diversas industrias que requerían sistemas seguros y confiables. La expansión del modelo en V también fue facilitada por la publicación de estándares de calidad como **ISO/IEC 12207** y **ISO/IEC 15288**, que se enfocan en el ciclo de vida del software y sistemas, y establecen marcos en los cuales el modelo en V encajaba perfectamente.

## Uso en Normas de Seguridad Crítica

Desde los años 2000, el modelo en V se consolidó en industrias de alta seguridad, que comenzaron a implementar el estándar **ISO 26262** para sistemas eléctricos y electrónicos de vehículos. Este estándar, derivado del modelo en V, establece una guía clara para verificar y validar sistemas críticos, especialmente para los sistemas de conducción y asistencia al conductor.

## ¿Sigue estando vigente?

El modelo en V es un modelo que sigue estando vigente en la actualidad, aunque su uso se ha visto reducido en detrimento de la aparición de las metodologías ágiles en el ámbito del desarrollo de software.

No obstante, es un modelo que se sigue utilizando en entornos donde el producto ha de cumplir con unos altos niveles de calidad y donde exista una trazabilidad (es el seguimiento de los cambios y avances que se hacen en un proyecto) y validación rigurosa. Es por ello que aunque su uso a nivel de software de consumo o desarrollo de aplicaciones web haya disminuido todavía sigue vigente en el desarrollo de aplicaciones del ámbito militar, aeronáutico o médico.

### Razones por las que sigue vigente:

1. **Validación y verificación:** Este modelo asegura que exista una fase de prueba por cada fase de desarrollo, es decir que tenga su fase de validación.
2. **Documentación:** Dada su rigurosidad ayuda a generar documentación.
3. **Trazabilidad:** El modelo V permite que la trazabilidad se haga de una forma sencilla donde es importante demostrar el cumplimiento de los requisitos y la calidad del software.
4. **Reducción de riesgos:** El modelo permite que se identifiquen problemas de forma temprana y rápida y que estos se puedan corregir.

Aunque no es la primera opción en el enfoque de desarrollo de software, sigue conviviendo con las metodologías ágiles en aquellos entornos donde la validación y rigurosidad premia sobre la flexibilidad y la rapidez.

## Ventajas e inconvenientes

### Ventajas del modelo V:

- Optimización de la comunicación entre las partes involucradas a través de términos y responsabilidades claramente definidos.
- Minimización de riesgos y mejor planificación a través de roles, estructuras y resultados fijos y predeterminados.
- Mejora de la calidad del producto gracias a medidas de control de la calidad firmemente integradas.
- Ahorro de costes gracias al procesamiento transparente a lo largo de todo el ciclo de vida del producto.

## Inconvenientes del Modelo en V:

- **Rigidez:** El modelo es rígido y secuencial, por lo que no se adapta fácilmente a cambios en los requisitos una vez iniciado el proceso.
- A menudo, **es difícil para el cliente enunciar en forma explícita** todos los requerimientos. El modelo V necesita que se expliciten los requerimientos al principio del proyecto, pero muchas veces existen problemas o dificultades para hacerlo por lo que si existe incertidumbre al respecto conforma una limitación.
- **El cliente debe tener paciencia.** No se dispondrá de una versión funcional del programa hasta que el proyecto esté muy avanzado.
- La naturaleza lineal del ciclo de vida clásico llega a “**estados de bloqueo**” en los que ciertos miembros del equipo de proyecto deben esperar a otros a fin de terminar tareas interdependientes, incluso llegando a superar al tiempo dedicado al trabajo productivo.

## ¿Dónde se usa?

El modelo en V es especialmente adecuado para proyectos de desarrollo de software y sistemas en los que la precisión, la estabilidad y la seguridad son prioridades absolutas. Su estructura rigurosa y su enfoque en la verificación y validación en cada etapa hacen que sea particularmente eficaz en entornos donde las fallas pueden tener consecuencias graves.

- **Software crítico:** Estándares de seguridad y regulación estrictos como software médico, aeronáutico y automotriz, sistemas bancarios o de defensa.
- **Sistemas embebidos:** Software que interactúa con un hardware muy específico como el que se usa en medicina o aeronáutica.
- **Proyectos con pocos cambios:** Software que no está sujeto a constantes cambios en los requisitos.

Para proyectos en los que los requisitos cambian constantemente o se espera iterar sobre ellos en función del feedback del cliente, el modelo en V no es la mejor opción. Su naturaleza rígida y secuencial hace que no se adapte bien a estos entornos; en tales casos, metodologías como **Agile**, **Scrum** o **DevOps** suelen ser más eficaces.

## Variantes del Modelo V

### 1. Modelo V Tradicional o Estándar

- **Características:** Es la versión clásica y estándar del Modelo V. Tiene una estructura lineal donde a cada fase de desarrollo le corresponde una fase de prueba.
- **Ámbito:** Su uso se centra en industrias reguladas como la aeroespacial o militar, ya que son industrias donde los procesos deben ser estructurados y muy detallados.

## 2. Modelo V Extendido

- **Características:** Incorpora actividades de retroalimentación y validación adicionales en cada etapa del desarrollo. Esto permite validar continuamente los requisitos a lo largo del ciclo de vida.
- **Ámbito:** Muy común en proyectos complejos y críticos donde es esencial validar constantemente los requisitos y el cumplimiento normativo.

## 3. Modelo V Iterativo

- **Características:** Este modelo permite iteraciones dentro de las fases del ciclo de vida. En esta variante los equipos pueden volver a etapas anteriores para ajustar los requisitos, el diseño, o la implementación en el caso que sea preciso hacerlo.
- **Ámbito:** Proyectos con requisitos cambiantes o en los que exista una incertidumbre alta, ya que brinda más flexibilidad en comparación con el Modelo V estándar o tradicional.

## 4. Modelo V en Espiral

- **Características:** Este tipo de modelo se combina con la metodología en espiral. Esto permite la creación de prototipos, la identificación de riesgos, y la validación continua en ciclos de desarrollo más cortos.
- **Ámbito:** Común en proyectos donde el riesgo es alto o los requisitos no están totalmente definidos desde un principio.

## 5. Modelo V Híbrido (V+Agile)

- **Características:** Es un modelo híbrido que combina el Modelo V con los principios de las metodologías ágiles. El modelo mantiene las clásicas fases de verificación y validación estructuradas, pero integrando iteraciones y feedback rápido para introducir las ventajas que aportan este tipo de metodologías.
- **Ámbito:** Ideal en proyectos regulados que requieren documentación exhaustiva pero que también desean la flexibilidad de los métodos ágiles para iterar en el desarrollo.

## Fuentes

<https://dreamix.eu/insights/what-is-the-difference-between-agile-and-v-model/>

<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/modelo-v/>

<https://appmaster.io/es/glossary/modelo-v-es>

Ingeniería del Software. Un Enfoque Practico