# Pruebas de Integración

## Objetivo

El propósito fundamental de las pruebas de integración dentro del Plan de Verificación y Validación del Software (SVVP) es asegurar la conformidad del software con las expectativas y necesidades detalladas tanto en la especificación del sistema como en los requisitos de software. Este proceso no solo busca evaluar cada componente de manera aislada, sino también cómo interactúan entre sí para formar un sistema cohesivo y funcional. A través de estas pruebas, se identifican y resuelven problemas de compatibilidad e interacción, garantizando que el conjunto del software opere de manera eficaz y eficiente según lo establecido por los criterios de aceptación y las necesidades del usuario final.

Otra meta esencial de las pruebas de integración consiste en identificar y remediar los defectos específicos que emergen durante la unificación de los distintos componentes software en el sistema completo. Estos errores de integración son particularmente insidiosos, ya que pueden no ser evidentes en las pruebas de unidades individuales. Ocurren en las interfaces y en las interacciones entre componentes, donde las discrepancias en la transferencia de datos, problemas de compatibilidad o fallos en las dependencias pueden provocar comportamientos inesperados o fallas en el sistema.

Un objetivo crucial de las pruebas de integración es confirmar que el software no solo opera eficazmente como una entidad aislada sino también dentro del ecosistema más amplio del sistema total. Este enfoque garantiza que cada función del software se desempeñe adecuadamente en armonía con los demás componentes, y que su integración sea fluida y libre de conflictos. La importancia de esta validación radica en el hecho de que un componente puede funcionar perfectamente por sí solo, pero fallar cuando se encuentra en interacción con otros elementos del sistema debido a incompatibilidades o errores en la gestión de las interfaces.

La evaluación de la interoperabilidad representa un aspecto clave en las pruebas de integración, enfocándose en la capacidad del software para interactuar y funcionar armónicamente con sistemas o componentes externos. Este objetivo va más allá de verificar la cohesión interna del sistema, extendiéndose a cómo el software se comunica y coopera con otras aplicaciones, plataformas o infraestructuras. La interoperabilidad es crucial en el entorno tecnológico actual, caracterizado por sistemas heterogéneos y una amplia gama de estándares de comunicación y formatos de datos.

Las pruebas de integración juegan un papel crucial en confirmar la estabilidad y calidad globales del software, con el objetivo de asegurar que este cumpla con los exigentes estándares de calidad, además de ser confiable y robusto. Esta etapa del proceso de desarrollo es vital, especialmente en contextos donde el software se desempeña en entornos críticos para la seguridad, como en operaciones espaciales, donde fallos mínimos pueden tener consecuencias catastróficas.

## Técnicas

Se detallará las estrategias y enfoques metodológicos seleccionados para llevar a cabo pruebas de integración, las cuales están diseñadas para evaluar exhaustivamente la compatibilidad, la comunicación, y la cohesión entre las distintas piezas del software.

* Pruebas de Integración Ascendentes: adoptan un enfoque que inicia con los elementos más elementales del software, integrando progresivamente aquellos de mayor complejidad. Esta metodología facilita la identificación temprana de dificultades relacionadas con la integración a lo largo del proceso de desarrollo. Al centrarse inicialmente en las unidades más simples y avanzar hacia las estructuras más complejas, se establece una base sólida que ayuda a asegurar la cohesión y funcionalidad del sistema en su totalidad desde las etapas iniciales, permitiendo un manejo más eficiente de los errores y anomalías conforme se incrementa la complejidad del sistema.
* Pruebas de Integración Descendentes: Se inician en la cúspide de la complejidad del sistema, integrando de manera gradual los elementos más fundamentales. Esta aproximación facilita el descubrimiento de inconvenientes de integración en las capas superiores del sistema. Empezando por los componentes de mayor complejidad y avanzando hacia los más simples, esta técnica permite asegurar que los niveles superiores del sistema funcionan correctamente antes de proceder a integrar las partes más elementales, optimizando así la detección y corrección de fallos en las etapas iniciales del proceso de integración.
* Pruebas de integración en Sandwich: Combina las pruebas ascendentes y descendentes, trabajando desde ambos extremos del espectro de complejidad hacia el medio, lo que permite una cobertura más amplia en menos tiempo.
* Pruebas de Integración Basada en Componentes: Se enfoca en verificar la funcionalidad y el rendimiento de los componentes previamente integrados en el sistema. Es especialmente útil cuando los componentes son desarrollados por diferentes equipos o vienen de terceros.
* Pruebas de Integración Continua: Se basa en la integración y prueba automáticas de componentes a medida que se desarrollan y se integran al repositorio de código. Favorece la detección temprana de errores y es fundamental en metodologías ágiles y DevOps.
* Pruebas de integración de sistema: Implican una evaluación exhaustiva del sistema en su totalidad, incluyendo la interacción entre todos sus componentes. Este enfoque se centra en verificar tanto la cohesión de los componentes individuales dentro del sistema como la funcionalidad general y el comportamiento del sistema completo. A través de este proceso, se busca asegurar que el sistema funcione de manera integrada y efectiva, cumpliendo con los requisitos y expectativas establecidos.

Estas prácticas son fundamentales para asegurar la entrega de software de alta calidad, funcionando como un sistema cohesivo y eficiente. Adoptar un enfoque metódico y estratégico hacia las pruebas de integración no solo facilita la identificación y resolución temprana de problemas, sino que también refuerza la confianza en la robustez y fiabilidad del software desarrollado.

## Criterios de cumplimiento

Estos criterios no solo definen las expectativas de calidad y funcionalidad, sino que también sirven como un faro guía para el proceso de desarrollo, asegurando que todas las actividades de verificación y validación se alineen con los objetivos finales del proyecto. La determinación precisa de estos criterios es esencial para la realización efectiva de pruebas, la identificación de áreas de mejora y la toma de decisiones informada sobre la aceptabilidad del software.

Los criterios de cumplimiento actúan como un puente entre las expectativas teóricas y la realidad práctica del software desarrollado, facilitando un camino claro hacia la consecución de un producto de software que no solo cumple con los requisitos especificados, sino que también satisface las necesidades y expectativas de los usuarios finales.

Los criterios de cumplimiento son los siguientes:

* Cobertura de Requisitos: Evalúa en qué medida el software cumple con todos los requisitos funcionales y no funcionales especificados en la documentación del proyecto. Esto incluye características, rendimiento, seguridad, y usabilidad.
* Cobertura de Código: Mide el porcentaje del código fuente que ha sido ejecutado y verificado a través de las pruebas, buscando alcanzar una cobertura que garantice la identificación de posibles errores o problemas no detectados.
* Ausencia de Defectos Críticos: Establece que no deben existir errores críticos o bloqueantes al momento de la entrega. Los defectos menores deben ser gestionados y programados para corrección en versiones futuras, según su prioridad.
* Pruebas de Regresión Satisfactorias: Asegura que las nuevas modificaciones o actualizaciones no afecten negativamente las funcionalidades existentes del software. Se deben realizar pruebas de regresión para confirmar que los cambios no introduzcan nuevos errores.
* Tiempo de Respuesta y Rendimiento: Define los tiempos de respuesta aceptables y los parámetros de rendimiento para el software bajo diferentes condiciones de carga. Esto es crítico para aplicaciones que requieren alta disponibilidad o que se utilizan en entornos de tiempo real.
* Estándares de calidad y seguridad: El software debe adherirse rigurosamente a los estándares de calidad y seguridad dictados por la Agencia Espacial Europea (ESA) y otras entidades pertinentes. Este cumplimiento asegura no solo la alta calidad del software sino también su confiabilidad en términos de seguridad, aspectos críticos en entornos operativos donde el margen de error es mínimo. La adherencia a estos estándares involucra una serie de evaluaciones y pruebas detalladas que abarcan desde la codificación y diseño hasta la implementación y mantenimiento, garantizando que el software no solo satisfaga las expectativas de rendimiento, sino que también proteja la integridad de los datos y la seguridad de los usuarios.

Establecer estos criterios de cumplimiento es vital para confirmar que el software cumple con los requisitos necesarios para su funcionamiento en ambientes espaciales y en condiciones críticas de seguridad.