



Universidad
Tecmilenio®



Software Testing

Laboratorio de
pruebas

Semana 8



Los laboratorios de prueba son útiles para que los objetivos del desarrollo se cumplan por completo, buscando que los resultados indiquen si las aplicaciones funcionan correctamente, si cumplen con las expectativas o simplemente saber cómo funciona una determinada herramienta.

Por lo que esta fase de evaluación se vuelve un punto clave para la consecución de sistemas interactivos con máxima usabilidad, a través de la aplicación de técnicas que ingresen la retroalimentación necesaria por parte de usuarios y de evaluadores con experiencia, con el fin de mejorar los procesos interactivos.



Estrategia y tipos de laboratorios de pruebas



Las pruebas se pueden realizar en espacios que se adaptan específicamente para ese fin o en laboratorios bien establecidos.

También, se pueden utilizar salas acopladas, así como emplear el mismo entorno de trabajo en donde los usuarios realizan sus actividades cotidianas.

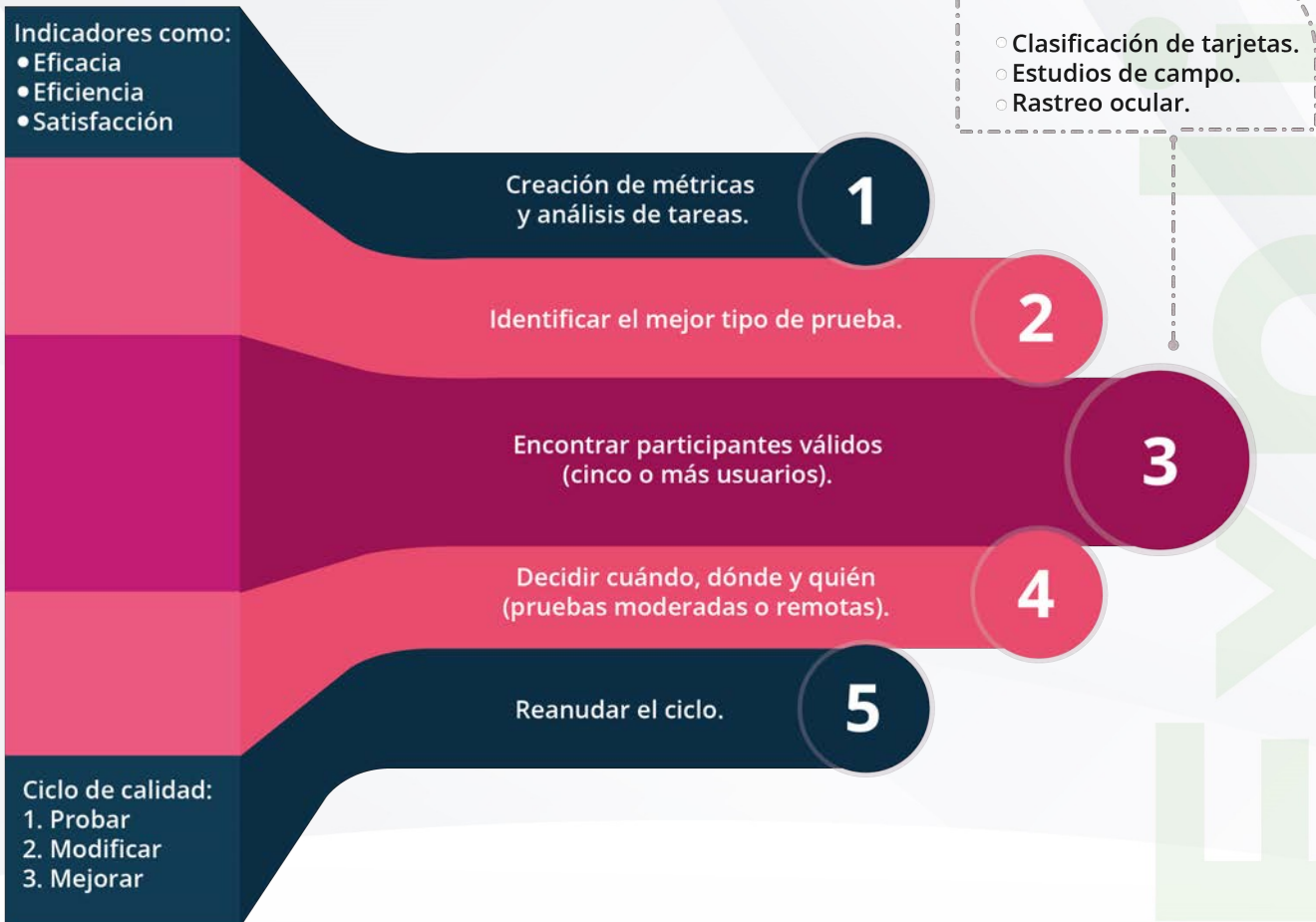
Laboratorios de usabilidad

- Compuestos de dos salas (una para usuarios de prueba y otra para los evaluadores).
- Pueden participar desarrolladores de forma anónima.
- Pueden tener cámaras y hacer grabaciones.
- Se pueden hacer de forma remota, es decir, no siempre deben estar en un mismo lugar físico, puede ser en la nube.

Laboratorios de accesibilidad

- Definidos para contribuir a la mejora de la autonomía.
- Apoyan a mejorar la forma de comunicarse con personas que tienen discapacidad.
- Pueden usarse.

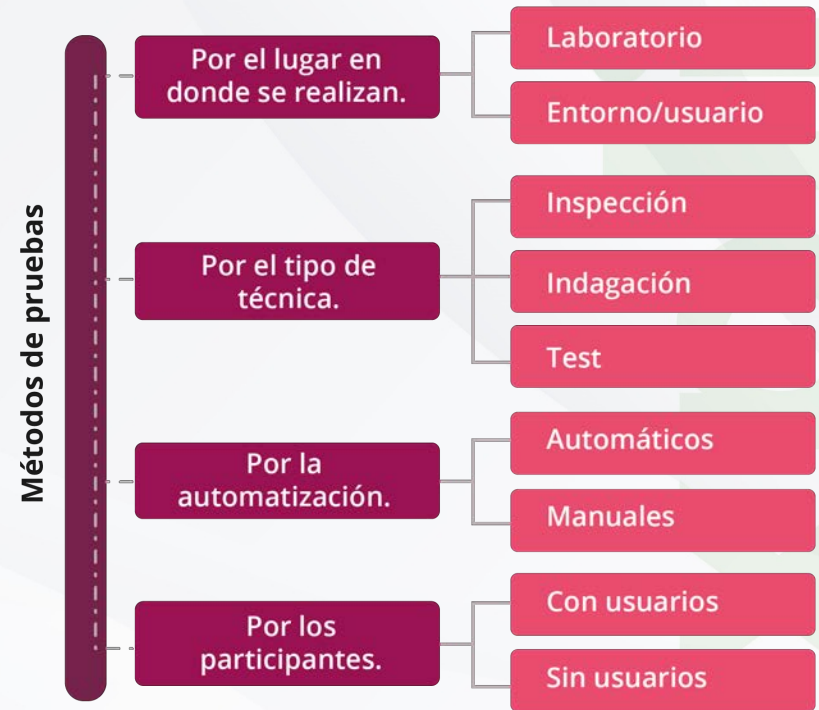
Pasos para ejecutar pruebas de laboratorio



Laboratorios de prueba en sitio y en la nube

Los tipos de laboratorios de prueba son muy variados, cada uno de ellos puede utilizar medios y técnicas diferentes para medir distintos aspectos.

Para poder definir cuál de ellos escoger, será necesario conocer qué es lo que se obtendrá al realizar dicha prueba y cuánto le costará al proyecto.



Fuente: Granollers, T. (s.f.). *Clasificación de los métodos*. Recuperado de <https://mpiua.invid.udl.cat/fases-mpiua/evaluacion/clasificacion/>

Pruebas por el lugar donde se realizan

Laboratorio

Ventajas

- Se pueden usar equipos audiovisuales o computadoras con gran capacidad.
- Los usuarios pueden operar sin que sean interrumpidos por otras actividades.

Desventajas

- El entorno a distancia puede tener algunas interrupciones.
- Puede ser que el contexto de interacción no sea el más adecuado.
- El usuario se puede desconcentrar

Entorno/usuario

Ventajas

- Forma más práctica y eficiente de ver cómo los usuarios trabajan en equipo.
- El análisis de requisitos puede mejorar significativamente si se contemplan cosas que no se habían considerado.

Desventajas

- Se puede extender la prueba debido a que los entornos pueden ser ruidosos.
- No todos los tipos de prueba son para entornos externos, sobre todo cuando es en un lugar poco accesible.

Pruebas por tipo de técnica



Pruebas por automatización

Automáticas	Manuales
Prácticas para hacer pruebas remotas.	Consumen más recursos.
Son objetivas, pero se usan en sistemas terminados.	Útiles para descubrir cosas que se salen del patrón.



Pruebas por los participantes

Con usuarios implicados

- Debe haber usuarios representativos.
- Facilita la intervención de usuarios en el proceso de diseño.

Sin usuarios implicados

- Se realizan por expertos.
- Es más rápido porque no se trabaja con usuarios.
- Para detectar errores funcionales y de consistencia.

Laboratorios de prueba en la nube

La observabilidad es muy utilizada, puesto que se encarga de “medir el estado actual de un sistema a través de los datos generados como registros y métricas” (Ayscom, 2022).



Actividad

¿Cuál es la importancia de los laboratorios de pruebas en los proyectos de *testing*?

¿Qué métodos de pruebas existen y cuándo se deben aplicar?



- Ayscom. (2022). *¿Qué es la observabilidad?* Recuperado de <https://www.ayscom.com/que-es-la-observabilidad/>
- Granollers, T. (s.f.). *Clasificación de los métodos.* Recuperado de <https://mpiua.invid.udl.cat/fases-mpiua/evaluacion/clasificacion/>



Existen diferentes tipos de laboratorios de prueba que pueden ser más convenientes para ciertas etapas en los proyectos, por tanto, su selección dependerá también de la etapa en donde se encuentre el desarrollo de la aplicación.

Por esta razón, los *testers* de los equipos de TI deben conocer a profundidad el tipo de desarrollo que se está haciendo, así como definir de manera correcta las pruebas más apropiadas para recabar la mejor y mayor información posible, con el fin de generar un producto no solo de alta calidad, sino pensando también en el costo-beneficio que ofrece.



Software Testing

Software para gestión
del ciclo de vida de
pruebas



Semana 8



Las herramientas de testing han evolucionado y son cada vez más necesarias, ya que nos permiten realizar ambientes de prueba más amigables a menor costo y con mayor calidad para garantizar un buen ciclo de vida de pruebas del software, asegurando la confiabilidad que el usuario está buscando en un desarrollo final.

Por lo que estas herramientas también ayudan a solucionar vulnerabilidades desde el propio proceso de implementación.



Selenium



Herramienta especializada en aplicaciones web. Está enfocada en realizar la automatización de pruebas funcionales para sistemas en línea y es de software libre.

Componentes

Puede usarse a través de IDE de terceros.

Entorno integrado de desarrollo de pruebas y plug in para navegadores.

API WebDriver para optimización de pruebas de interfaz de usuario.

Uso de Selenium Grid para la reducción de tiempo de pruebas de ejecución.

Uso de ServerHug para la ejecución de pruebas remotas y pruebas de múltiples usuarios en paralelo.

Appium

Herramienta que puede ejecutar *scripts* y probar las aplicaciones nativas, así como las aplicaciones web y las híbridas (las que están desarrolladas en sistemas operativos Android e iOS a través de WebDriver).



Características

Herramienta de uso libre.

Usa licencia "Apache license 2".

Sistema de plataforma cruzada, solo se escribe código una vez y se usa en varias plataformas.

Programada para soportar dispositivos físicos.

Diseñada bajo una arquitectura cliente-servidor.



Katalan

Basada en las herramientas de Selenium y Appium. Tiene como finalidad facilitar los procesos de desarrollo y gestión de pruebas para usuarios que no tengan amplios conocimientos de programación.



Componentes funcionales

Versión gratuita y con licencia.

Compatibilidad con Windows, Linux y MacOS.

Automatización de pruebas de API.

Plantillas para proyectos con librerías de casos de prueba.

Fácil de aprender.

No requiere conocimientos de programación avanzada.

Soporta extensiones para automatización avanzada.

TestComplete



Herramienta de testing de software enfocada en planear, gestionar y ejecutar diferentes pruebas dentro de su amplio catálogo, las cuales pueden ser unitarias, funcionales, de regresión, distribuidas, de rendimiento, de caja blanca y de cobertura.

Componentes funcionales

Facilidad para grabar y reproducir interacción con los usuarios.

Permite la ejecución de pruebas de estrés de sitios web.

Permite realizar pruebas funcionales avanzadas.

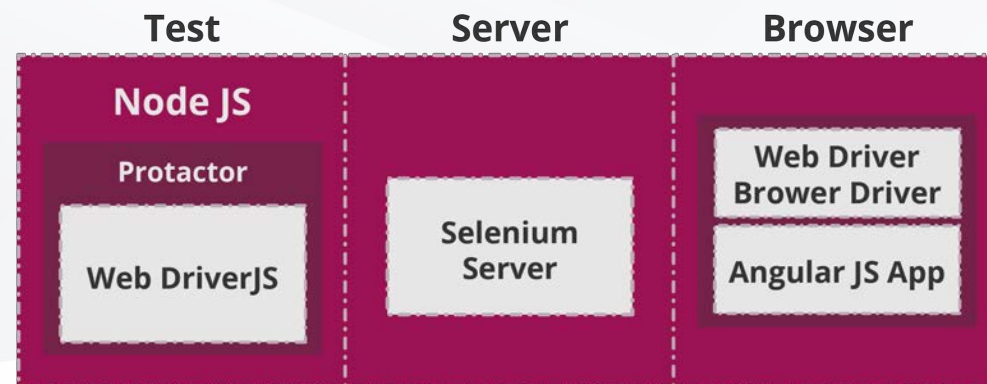
Selección de pruebas manuales o automatizadas.

Integración con sistemas de control de código fuente.

Protractor

La herramienta está basada en Selenium y se considera como un sistema para realizar pruebas punto a punto. Asimismo, tiene la funcionalidad de identificar elementos web de JavaScript y Angular JS. Además, permite gestionar de manera muy eficiente las esperas automáticas.

Debe trabajar en conjunto con Selenium para brindar una plataforma de prueba automatizada que simule la interacción de un usuario con la aplicación angular ejecutada, tal y como se aprecia en la siguiente imagen (Protractortest.org, s.f.):



APACHE JMeter

Herramienta de testing de la empresa Apache Software Foundation, es de código abierto y está programada en Java, misma que funciona bajo licencia Apache License 2.0. Jmeter permite realizar pruebas de carga y estrés, a través de la simulación de peticiones concurrentes con base en un número determinado de usuarios.



Componentes funcionales

Funcionalidad de hilos múltiples.

Resultados en formatos gráficos y tablas.

Pruebas distribuidas en diferentes computadoras.

El uso de complementos es gratuito.

Se integra fácilmente con otras herramientas.

Permite el registro de la actividad de usuarios para reproducir las pruebas.

Uso efectivo de las herramientas

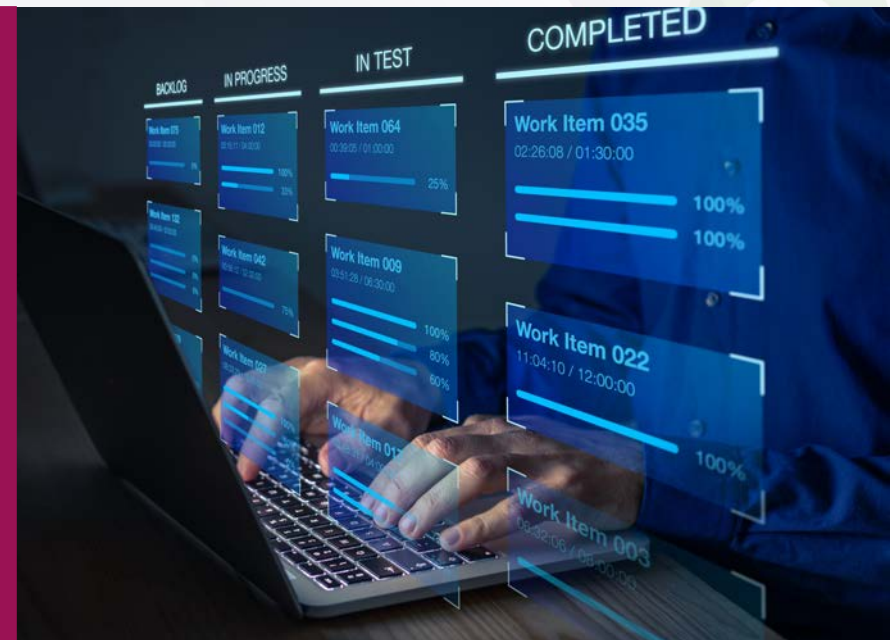
La elección de una herramienta es esencial para garantizar la funcionalidad y la mejor experiencia, en caso contrario, se podrían estar realizando pruebas incompletas.

Cuando se realizan las pruebas ideales, se puede optimizar la aplicación de acuerdo con los parámetros previamente establecidos para obtener las siguientes mejoras (Prasad, 2022):

- Mejorar la funcionalidad de la aplicación.
- Eliminar errores fácilmente.
- Optimizar el rendimiento.
- Garantizar la seguridad de los sistemas.
- Reducción de costos.
- Aumentar la confiabilidad y la estabilidad de la aplicación.
- Optimizar la calidad del resultado final.

¿Cómo puedes usar una herramienta de testing de forma efectiva?

¿Qué herramientas de apoyo para el proceso de testing existen y cuáles son sus características principales?



- Protractortest.org. (s.f.). *How It Works*. Recuperado de <https://www.protractortest.org/#/infrastructure>
- Prasad, D. (2022). *Comprensión de los diferentes tipos de pruebas de aplicaciones*. Recuperado de <https://geekflare.com/es/understanding-application-testing/>

La elección de la herramienta más adecuada para hacer testing de una aplicación determinada será relevante para comprobar la eficacia de la misma en los entornos de prueba que se hayan seleccionado previamente.

Por esta razón, el equipo de testing debe dedicar el tiempo suficiente para seleccionar la herramienta más adecuada, es decir, que cubra todas las condiciones de los módulos a ser probados, ya que, si no se realiza de manera correcta, podrían saltarse errores que posiblemente comprometan la experiencia del usuario final.



Software Testing

Administración de
contratos y
presupuestos

Semana 8





Es importante tener contratos de trabajo que permitan usar las herramientas de testing de software de manera eficaz, es decir, que contemplen todos los elementos: humanos, de sistemas, *back office* y *front office*, así como de mantenimiento y técnicos, con el fin de reaccionar de manera eficiente ante situaciones que pudieran comprometer el éxito del desarrollo.

Por ende, contar con un equipo de trabajo multifuncional que se encargue de revisar y dar seguimiento a los contratos se convierte en una prioridad para comenzar cualquier proyecto de pruebas de software.

Tipos de contratos y esquema de fábrica

Antes de decidir por un proveedor para trabajar con su herramienta de testing, es importante conocer algunas de las características que podrían ser incluidas en el contrato de licenciamiento.

Es fundamental conocer los elementos relevantes que deben ser vigilados en un contrato para garantizar la mejor forma de trabajo “ganar-ganar” (Short y Peláez, 2021):

Formar un equipo de negociación.

Análisis de costos y tarifas.

Términos y condiciones.

Características del proveedor.

Plantilla de puntuación.

Formar un equipo de negociación

Apoyarse en los expertos en temas legales de la empresa. Si no existe ese rol, puede haber un equipo con personas con experiencia en el desarrollo de proyectos.

Elementos a considerar para centrarse en la negociación:

Adquisición.

Seguridad de la
información.

Necesidades de la
empresa.

Desarrollo de
aplicaciones.

Finanzas y
contabilidad.

Aspectos jurídicos
y contratos.

Análisis de costos y tarifas

Software

Precio real neto.

Clase de licencia.

Métricas por usuario.

Requisitos de instalación.

Costo por asistencia técnica.

Mantenimiento y soporte técnico

Duración de la garantía.

Cobertura total.

Descripción de la cobertura.

Descuentos y condiciones.

Servicios y formación

Precios y tiempos de respuesta.

Base de las tarifas.

Métodos de instalación.

Plazos de pago y costos de viaje.



Términos y condiciones

- Se describen los riesgos y las ventajas del contrato.
- Las condiciones pueden variar si se toma en cuenta el método de implementación (puede ser en la nube).

Características del proveedor

- Algunos pueden recabar información.
- Es importante contar con información acerca de la empresa y forma de trabajo para tomar ventajas sobre el proceso de negociación.

Plantilla y puntuación

- Tabla de puntuación para facilitar y tener elementos estructurados de forma eficiente.
- Evaluar los elementos relevantes (precios, costos, tarifas por licencia, costos de capacitación, riesgos, entre otros).

Herramientas para crear, automatizar y dar seguimiento a contratos

ContractPodAi

Juro

Icertis Suite

PandaDoc

ContractBook

Explicación



Administración del presupuesto

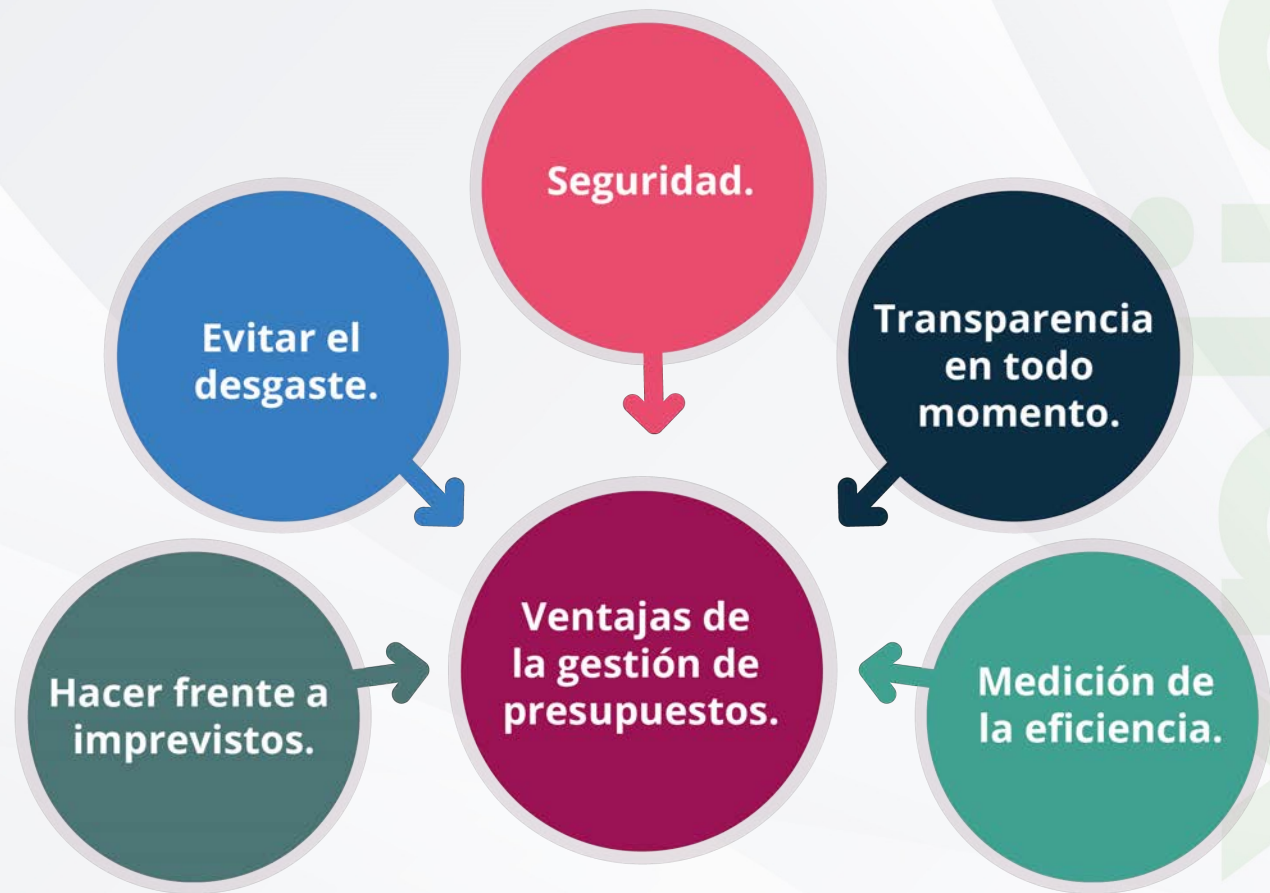
La definición de estos objetivos servirá para establecer los indicadores adecuados para contar con una evaluación a la medida del retorno sobre la inversión (ROI) (SAP Concur, 2021):

- Anticipar políticas para medir los requerimientos.
- Análisis del beneficio a corto y mediano plazo.
- Contemplar índices financieros como el VPN (valor presente neto) y TIR (tasa interna de retorno).
- Considerar excepciones que puedan presentarse.

Se recomienda invertir en:

- Testing de software.
- Procedimientos del testing.
- Separación de ambientes para un mejor testing.
- Asignación de funciones.
- Equipo de software.
- Reservas para contingencias.





Fuente: Hansen, B. (2021). *¿Qué es la gestión de recursos y por qué es importante?*
Recuperado de <https://www.wrike.com/es/blog/que-es-la-gestion-de-recursos-y-por-que-es-importante/>

¿Qué elementos se deben vigilar en un contrato para que ambas partes se vean beneficiadas?



- Hansen, B. (2021). *¿Qué es la gestión de recursos y por qué es importante?* Recuperado de <https://www.wrike.com/es/blog/que-es-la-gestion-de-recursos-y-por-que-es-importante/>
- SAP Concur. (2021). *Presupuesto de tecnologías de la información: ¿cómo optimizarlo?* Recuperado de <https://www.concur.co/news-center/presupuesto-tecnologia>
- Short, T., y Peláez, B. (2021). *6 pasos para conseguir que la negociación del contrato de software tenga éxito.* Recuperado de <https://www.capterra.es/blog/2231/pasos-para-exitosa-negociacion-contrato-de-software>

Es fundamental seleccionar la mejor herramienta de testing que se acople al tipo de proyecto que queremos probar. No obstante, si no tenemos contratos de proveeduría que garanticen el uso correcto de la misma, además de planes de respaldo, atención técnica disponible, contratos saludables y un presupuesto que esté aterrizado a las necesidades de la aplicación, entonces no se contará con un plan integral de gestión del proyecto que ayude a llevarlo a un buen término.

Es sumamente importante considerar todos estos elementos en conjunto para poder generar confianza y eficiencia en la administración del proyecto de testing.

