



Universidad de Buenos Aires
Facultad de Ciencias Económicas



Las Pruebas de las Tecnologías

AR Tema extractado del libro "**Análisis Funcional de Sistemas y Tecnologías de la Información**" de Aníbal M. Mazza Fraquelli - ISBN 978-987-26981-3-3

Presentación del Tema

Las pruebas en proyectos de Tecnologías de la Información (TI) constituyen un conjunto sistemático de actividades orientadas a verificar y validar que un sistema cumple con los requerimientos definidos, opera de manera segura y mantiene niveles adecuados de desempeño. En la gestión moderna de proyectos tecnológicos, las pruebas no representan una etapa secundaria, sino un componente estratégico que impacta directamente en costos, riesgos, continuidad operativa y reputación organizacional.

Desde la perspectiva de la administración, comprender los distintos tipos de pruebas permite dimensionar adecuadamente presupuestos, cronogramas y riesgos asociados a la implementación de sistemas de información. En proyectos como la implementación de un ERP (Enterprise Resource Planning – Planificación de Recursos Empresariales), el desarrollo de una plataforma digital o la integración de sistemas mediante APIs (Application Programming Interface – Interfaz de Programación de Aplicaciones), la ausencia de pruebas adecuadas puede generar pérdidas económicas significativas.

Las pruebas se clasifican según su alcance, objetivo y momento de ejecución. Entre las más relevantes se encuentran: pruebas unitarias, de integración, funcionales, de performance, de seguridad, modulares, de sistemas, estáticas, dinámicas y de carga pico.

Desarrollo

1. Pruebas Unitarias (Unit Testing)

Las pruebas unitarias verifican el correcto funcionamiento de componentes individuales del sistema, tales como funciones, métodos o módulos pequeños.

Objetivo:

- Confirmar que cada unidad de código realiza correctamente la tarea para la cual fue diseñada.

Características:

- Se ejecutan en etapas tempranas.
- Generalmente las realiza el desarrollador.
- Automatizables mediante frameworks específicos.

Ejemplo:

Validar que una función que calcula impuestos aplique correctamente la tasa correspondiente según categoría.

Desde la administración, las pruebas unitarias reducen el costo de corrección temprana. Detectar errores en esta etapa es significativamente más económico que hacerlo en producción.

2. Pruebas de Integración (Integration Testing)

Estas pruebas verifican la interacción entre diferentes componentes del sistema.

Objetivo:

- Confirmar que los módulos funcionan correctamente cuando se comunican entre sí.

Ejemplo:

Verificar que el módulo de ventas actualice correctamente el módulo contable tras emitir una factura.

Importancia administrativa:

- Detecta inconsistencias entre áreas funcionales.

- Reduce riesgo de errores interdepartamentales.
 - Evita fallas críticas en flujos operativos.
-

3. Pruebas Funcionales (Functional Testing)

Evalúan si el sistema cumple con los requerimientos funcionales especificados en el análisis.

Objetivo:

- Verificar que el sistema hace lo que debe hacer.

Ejemplo:

Confirmar que el sistema permite registrar órdenes de compra siguiendo el flujo aprobado.

Desde la gobernanza de TI, estas pruebas validan el alineamiento entre especificación y ejecución.

4. Pruebas de Performance (Performance Testing)

Evalúan el comportamiento del sistema bajo determinadas condiciones de carga.

Miden:

- Tiempo de respuesta.
- Uso de recursos.
- Estabilidad bajo demanda.

Ejemplo:

Medir cuánto tiempo tarda el sistema en procesar 1.000 transacciones simultáneas.

Desde la administración, estas pruebas previenen pérdidas asociadas a lentitud o caída del sistema.

5. Pruebas de Seguridad (Security Testing)

Buscan identificar vulnerabilidades y debilidades que puedan comprometer la confidencialidad, integridad y disponibilidad de la información.

Incluyen:

- Pruebas de penetración (Penetration Testing).
- Análisis de vulnerabilidades.
- Evaluación de controles de acceso.

Ejemplo:

Verificar que usuarios sin permisos no puedan acceder a información financiera sensible.

Impacto administrativo:

- Protege activos críticos.
 - Evita multas regulatorias.
 - Reduce riesgo reputacional.
-

6. Pruebas Modulares (Module Testing)

Las pruebas modulares validan el correcto funcionamiento de módulos completos dentro de un sistema más amplio.

Diferencia respecto a pruebas unitarias:

- La unidad es un componente mínimo.
- El módulo es una agrupación funcional más amplia.

Ejemplo:

Validar que el módulo de gestión de inventario funcione correctamente como conjunto integrado.

7. Pruebas de Sistemas (System Testing)

Evalúan el sistema completo en un entorno similar al real.

Objetivo:

- Verificar el funcionamiento integral.
- Confirmar cumplimiento global de especificaciones.

Ejemplo:

Simular una operación completa desde la venta hasta el registro contable final. Desde la administración, estas pruebas son críticas antes del Go-Live (puesta en producción).

8. Pruebas Estáticas (Static Testing)

Se realizan sin ejecutar el código.

Incluyen:

- Revisión de documentación.
- Inspección de código.
- Análisis estático automatizado.

Objetivo:

- Detectar errores estructurales.
- Verificar cumplimiento de estándares.

Ejemplo:

Revisar si el código cumple con normas de seguridad antes de su ejecución.

Desde la gestión de riesgos, estas pruebas previenen defectos antes de la fase operativa.

9. Pruebas Dinámicas (Dynamic Testing)

Se realizan ejecutando el sistema para observar su comportamiento real.

Incluyen:

- Pruebas funcionales.
- Pruebas de integración.
- Pruebas de rendimiento.

Permiten evaluar:

- Comportamiento ante entradas reales.
- Respuesta ante errores.
- Robustez operativa.

10. Pruebas de Carga Pico (Stress Testing / Peak Load Testing)

Evalúan el sistema bajo condiciones extremas o máximas.

Objetivo:

- Determinar el límite operativo.
- Identificar punto de fallo.
- Evaluar resiliencia.

Ejemplo:

Simular tráfico masivo durante campañas comerciales o cierre fiscal anual.

Desde la perspectiva administrativa, estas pruebas protegen ingresos y reputación ante eventos críticos.

11. Relación entre tipos de pruebas

Las pruebas no son independientes, sino complementarias.

Secuencia habitual:

1. Pruebas estáticas.
2. Pruebas unitarias.
3. Pruebas modulares.
4. Pruebas de integración.
5. Pruebas funcionales.
6. Pruebas de sistemas.
7. Pruebas de performance y carga.
8. Pruebas de seguridad.

Esta estructura progresiva reduce riesgos en cada etapa.

12. Impacto financiero y estratégico

Invertir en pruebas implica:

- Costos adicionales.

- Recursos técnicos.
- Tiempo adicional.

Sin embargo, la ausencia de pruebas puede generar:

- Fallas en producción.
- Interrupción operativa.
- Multas regulatorias.
- Daño reputacional.
- Retrabajos costosos.

Desde la administración, el costo de pruebas es una inversión preventiva.

13. Ejemplo integral

Una empresa implementa un sistema de comercio electrónico:

- Pruebas unitarias: validan cálculo de descuentos.
- Pruebas de integración: validan conexión con sistema de pagos.
- Pruebas funcionales: verifican proceso completo de compra.
- Pruebas de carga pico: simulan alta demanda en fechas especiales.
- Pruebas de seguridad: protegen datos de clientes.

Cada tipo reduce un riesgo específico.

Conclusión

Las pruebas en proyectos de Tecnologías de la Información constituyen un componente estratégico esencial para garantizar calidad, seguridad y desempeño. Desde pruebas unitarias hasta pruebas de carga pico, cada tipo cumple una función específica dentro de un enfoque integral de aseguramiento de calidad.

Desde la perspectiva de la administración, comprender estas pruebas permite dimensionar adecuadamente riesgos, presupuestos y decisiones estratégicas. La correcta planificación de pruebas reduce costos futuros, protege la continuidad del negocio y fortalece la gobernanza tecnológica.

En un entorno organizacional cada vez más digitalizado, la calidad del sistema depende directamente de la rigurosidad del proceso de pruebas implementado.

Preguntas de autoevaluación

1. ¿Cuál es la diferencia entre pruebas unitarias y pruebas modulares?
 2. ¿Por qué las pruebas de integración son críticas en sistemas empresariales?
 3. ¿Qué riesgos se mitigan mediante pruebas de seguridad?
 4. ¿Cuál es la diferencia entre pruebas estáticas y dinámicas?
 5. ¿Por qué las pruebas de carga pico son estratégicas en entornos comerciales digitales?
-

Material de Clases

Compilado por **Aníbal M. Mazza Fraquelli** Doctor de la Universidad de Buenos Aires para el uso de sus clases en la Facultad de Ciencias Económicas de la Universidad de Buenos Aires.

Contenidos de esta página

Los contenidos **aquí incluidos integran desarrollos y escritos propios del autor, así como materiales de terceros (documentos, textos, fragmentos, conceptos, imágenes, esquemas, definiciones u otros recursos)**, los cuales son utilizados a título ilustrativo, explicativo o formativo, respetando la normativa vigente en materia de derechos de autor y citando las fuentes cuando corresponde.

La selección, organización, adaptación pedagógica y contextualización de los contenidos constituye un trabajo original del autor, orientado a facilitar los procesos de enseñanza y aprendizaje.

Este material no persigue fines comerciales y su reproducción, total o parcial, queda limitada al ámbito educativo, debiendo preservarse siempre la mención de la autoría y las fuentes originales.

Autorización de uso

Se permite la reproducción, comunicación pública, distribución y utilización total o parcial de los contenidos de su material, en formato físico o digital, con fines exclusivamente educativos, académicos o de divulgación, siempre que se respete la integridad del contenido y se incluya la correspondiente referencia a la fuente y a la autoría.

Las ideas, opiniones e interpretaciones contenidas en este material corresponden exclusivamente al autor.

Queda expresamente excluido cualquier uso con fines comerciales.