Las pruebas durante todo el ciclo de vida del Software

Para este programa los modelos de ciclo de vida de desarrollo de software los dividiremos en dos:

- **Secuenciales:** Las entregas pueden durar meses o años.
  - Cascada
  - V-Model
- **Iterativos e incrementales:** Las entregas son constantes y rápidas.
  - Proceso unificado racional
  - Scrum
  - Kanban
  - Espiral/Prototipos

Nota: No hay una receta mágica que nos indique que modelo de desarrollo de software debemos utilizar ya que **depende del contexto.**

Algunos criterios que se pueden utilizar para seleccionar el modelo de software son:

- Objetivos del proyecto
- Tipo de producto en desarrollo
- Prioridades comerciales
- Riesgos del producto y proyecto
- Entre otros.

Niveles de prueba

Para este programa tenemos los siguientes niveles:

- Pruebas de componente
- Pruebas de integración
- Pruebas de sistema
- Pruebas de aceptación

	Objetivos	Base de prueba/Productos de trabajo	Objetos de prueba	Defectos y fallos típicos	Enfoques y responsabilidades específicas
Pruebas de componentes	• Reducir riesgo • Confianza en la calidad del componente. • Encontrar defectos • Verificar comportamientos funcionales y no funcionales.	• Diseño detallado • Codigo • Modelo de datos • Especificaciones del componente	• Componentes o modulo • Código • Clases	• Funcionalidad incorrecta • Problemas en flujo de datos • Código y/o lógica incorrecta.	• Desarrollador • Habilidades de desarrollo son requeridas.

	Objetivos	Base de prueba/Productos de trabajo	Objetos de prueba	Defectos y fallos típicos	Enfoques y responsabilidades específicas
Pruebas de Integración Incluyen: ***Pruebas de integración entre componentes*** ***Pruebas de integración entre sistemas***	• Verificar comportamientos funcionales y no funcionales. • Encontrar defectos en interfaces, componentes o sistemas • Prevenir que los defectos se vayan a otros niveles.	• Diagrama de secuencia • Definición de interfaces. • Diseño del software o sistema	• Subsistemas • Base de datos • APIs • Infraestructura • microservicios	• Datos incorrectos o faltantes. • Fallos en comunicación de componentes	• Centrado en integración de componentes y sistemas • Integración entre componentes:**Desarrollador** • Integración entre componentes:**Probador**

	Objetivos	Base de prueba/Productos de trabajo	Objetos de prueba	Defectos y fallos típicos	Enfoques y responsabilidades específicas
Pruebas de Sistema	• Verificar comportamientos funcionales y no funcionales. • Validar sistema completo. • Encontrar defectos	• Requerimientos de sistema • Historias • Diagramas • Manuales	• Aplicaciones • Hardware o software • Sistemas operativos	• Cálculos incorrectos • Comportamientos incorrectos • Fallos acorde a los manuales	• Probadores independientes. • Pruebas de extremo a extremo.

	Objetivos	Base de prueba/Productos de trabajo	Objetos de prueba	Defectos y fallos típicos	Enfoques y responsabilidades específicas (*)
Pruebas de Aceptación	• Verificar comportamientos funcionales y no funcionales. • Validar sistema completo. • **Encontrar defectos NO es un objetivo**	• Procesos de negocio • Requerimientos funcionales y no funcionales • Procedimientos de instalación.	• Sistema sujeto a pruebas (SUT) • Datos de producción existentes • Procesos de negocio (una vez esté todo integrado).	• No cumplen con los requerimientos • Reglas de negocio incorrectas • Fallos no funcionales	• Auditores, probadores, usuarios finales • Más detalle abajo.

(*)

- *Pruebas de aceptación del usuario:* Aptitud para el uso del sistema por parte de los usuarios.
- *Pruebas de aceptación operativa:* Copia de seguridad, instalación,recuperación de desastres.
- *Pruebas de aceptación contractuales y regulatorias:*
  - Contractuales: En base a un contrato - Usuarios independientes o probadores.
  - Regulatorias: Regulaciones. - Usuarios, probadores o auditores.
- *Pruebas de aceptación alfa y beta:*
  - Alfa: Interno en la compañía - Usuarios o probadores independientes.
  - Beta: Externo a la compañía - Clientes potenciales

Tipos de pruebas.

- Para este programa:

	Objetivos
Pruebas funcionales	• Evaluar funcionalidades del sistema • “Lo que el sistema debe hacer” • En todos los niveles • Se utilizan técnicas de caja negra
Pruebas no funcionales	• Rendimiento, usabilidad,etc • “Que tan bien se comporta el sistema” • Se utilizan técnicas de caja negra
Pruebas estructurales (Caja blanca)	• Basadas en estructura interna ○ Código, flujos de trabajo.

Pruebas relacionadas con cambios	• Pruebas de confirmación
	  ○ Confirmar que se ha corregido el defecto.
	• Pruebas de regresión
	  ○ Confirmar que los cambios no han afectado otra parte del sistema.

Pruebas de mantenimiento

- Siempre tendremos que hacer mantenimiento:
- Causas:
  - Cambios en el sistema
  - Cambios en el entorno
  - Retiro de aplicaciones del mercado
- Al hacer mantenimiento esto impactará el sistema o las pruebas.