

# Estrategias y herramientas de pruebas

Escuela de programación Java

Especialidad Java/XML/JEE

[Indra](#) & [Icono Training Consulting](#)

# Las pruebas en el ciclo de vida de diversas metodologías

En lo que sigue, la orientación a objetos puede o no estar presente.

Estructuradas, en "*cascada*"

- Iterativas e incrementales
- Dirigidas por prototipos
- Ágiles
  - Programación Extrema (XP)
    - Desarrollo Dirigido por pruebas (TDD)
      - Desarrollo dirigido por comportamiento (BDD)

# Estructuradas

## Actividades principales

- Viabilidad del sistema
- Análisis de requerimientos
- Diseño del sistema
- Codificación y pruebas unitarias
- Integración y pruebas de sistema (o de integración)
- Despliegue (paso a producción) y mantenimiento

# Iterativas e incrementales

- El sistema se divide en partes más pequeñas, denominadas normalmente iteraciones.
- En cada iteración, se lleva a cabo el ciclo estructurado.
- Por tanto, las pruebas se llevan a cabo al final de cada iteración.

# Dirigidas por prototipos

## Actividades principales

1. Toma de requisitos
  2. Diseño del sistema
  3. Creación o modificación del prototipo
  4. Presentación del prototipo a los usuarios finales (*stakeholders*)
  5. Refinamiento del prototipo
- Los pasos 1..5 se repiten N veces.
  - Implementación del sistema con cualquier metodología, partiendo de cero o reutilizando los prototipos

# Ágiles I

- Ley no escrita de la ingeniería de software el usuario **siempre** cambiará los requerimientos antes de que el proyecto finalice.
- Las metodologías ágiles nacieron para hacer frente a este incremento de la complejidad, y a una tasa de cambios alta mientras el proyecto se está desarrollando.

# Ágiles II

## Características principales

- Siempre se usan iteraciones breves, con **pruebas exhaustivas**
- Se da prioridad a la interacción entre los integrantes del proyecto más que a las herramientas
- Se considera a los clientes miembros del proyecto
- Se le más importancia al software funcionando que a la documentación
- Se presta más atención a responder al cambio con agilidad que a la planificación
- Dentro de estas metodologías, dos muy utilizadas son la

**Programación Extrema (XP)** y **Scrum**

# Tipos de pruebas

- Unitarias: se prueba un componente del sistema de forma aislada
- Integración: se prueba la interacción de varios componentes
- Estrés: verificar el rendimiento de un software
- Regresión: comprobar que modificaciones o nuevos requerimientos no *rompen* el sistema
- Aceptación: similares a las pruebas de sistema, pero incluyendo el matiz de que la aceptación proviene directamente del usuario final, especialmente bajo las metodologías ágiles.
- Sistema: se prueba, a menudo con robots, el sistema como un todo. A menudo se incluyen aquí las pruebas de estrés y de regresión



# Herramientas

- [Objetos simulados](#) (mock objects)
  - [EasyMock](#). Ver [referencia](#) y este [tutorial](#)
- Pruebas unitarias y de integración
  - [JUnit](#). Ver [referencia](#)
  - [Easyb](#). Ver [referencia](#)
  - [Scalatest](#). Ver [referencia](#) y este [tutorial](#)
  - [Specs](#). Ver [referencia](#) y este [tutorial](#)
  - [JBehave](#). Ver [referencia](#), [tutorial1](#) y [tutorial2](#)
- Pruebas de aplicaciones cliente Web
  - [HtmlUnit](#). Ver [referencia](#), [tutorial1](#) y [tutorial2](#).
  - [JWebUnit](#). Ver [referencia](#) y este [tutorial](#)