

## UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

INTEGRANTES: ANDRADE URIEL, MATIAS JEAN, PLUA THOMAS

FECHA: 30/01/2025

CURSO: Fundamentos de Ingeniería de Software

1. La clasificación de las metodologías de desarrollo de software es:

Estructuradas/Orientadas a objetos.

Orientadas a procesos/Orientadas a datos.

Mixtas.

Las metodologías de desarrollo de software han evolucionado junto con la programación: primero fueron estructuradas, luego orientadas a objetos y ahora son ágiles. (*Software*, s. f.)

2. Las metodologías orientadas a procesos se fundamentan en:

El modelo básico de entrada-salida.

El modelo básico del proceso.

El modelo básico de entrada-proceso-salida.

“La Ingeniería del Software se fundamenta en el modelo básico entrada/proceso/salida de un sistema” (García-Peñalvo & García-Holgado, s. f.).

3. El proceso unificado de desarrollo se basa en características que son:

Centrado en el diseño/Centrado en la arquitectura/Dirigido por casos de uso.

Centrado en la arquitectura/Dirigido por casos de uso/Iterativo e incremental.

Centrado en la arquitectura/Dirigido por casos de uso/Exige poca documentación.

Según *García Peñalvo* el proceso usa casos de uso para guiar el desarrollo, se basa en una arquitectura bien definida y avanza de forma iterativa, mejorando el sistema en cada ciclo.

4. Las metodologías orientadas a datos se concentran en:

Entrada-salida.

Actividades de análisis.

Procesos de entrada y salidas.

Según Aricapa las metodologías orientadas a datos se enfocan en actividades de análisis, ya que buscan procesar y entender la información más allá de la simple entrada y salida de datos.

5. Las metodologías orientadas a objetos se fundamentan en:

Datos y procesos.

Procesos de entrada y salidas.

**Conjunto de objetos.**

Según Booch & Jacobson explican que las metodologías orientadas a objetos se centran en la creación y manipulación de objetos, que son instancias de clases que encapsulan tanto datos como comportamientos. Los objetos interactúan entre sí para realizar las funciones del sistema.

6. El diseño y la implementación de las metodologías orientadas a objetos está centrado en:

**El comportamiento y la estructura.**

Los procesos y los datos.

Los objetos.

Larman (2004), describe que el diseño orientado a objetos se enfoca en definir la estructura del sistema (clases, objetos, relaciones) y el comportamiento (métodos, interacciones entre objetos). Ambos aspectos son fundamentales para la implementación de sistemas orientados a objetos.

7. Las metodologías ágiles se clasifican en:

**Agile Unified Process-AUP.**

Rational Unified Process-RUP.

Metodología de Edward Yourdon.

Ambler (2002), introduce el Agile Unified Process (AUP) como una metodología ágil que simplifica el Rational Unified Process (RUP) y se enfoca en la entrega iterativa e incremental de software. AUP es una de las metodologías ágiles más reconocidas.

8. Las metodologías orientadas a datos se clasifican en:

Jackson Structured Programming-JSP-AUP.

Jackson Structured Design –JSD-RUP.

**Desarrollo de sistemas estructurados-Warnierr-Orr.**

Yourdon (1989), menciona que las metodologías orientadas a datos, como el Desarrollo de Sistemas Estructurados de Warnier-Orr, se centran en el diseño de sistemas basados en la estructura de los datos y su flujo a través del sistema. Estas metodologías son especialmente útiles en sistemas donde los datos son el componente principal.

9. Las metodologías ágiles más conocidas son:

OMT, OOSE, PU.

**XP, SCRUM, AUP.**

Método de BOOCH.

Pressman (2010), menciona que la metodología XP (Extreme Programming) es la más utilizada, pero además menciona otras metodologías propuestas y usadas en la industria del software como las siguientes:

- Desarrollo adaptativo de software (DAS)
- Scrum
- Desarrollo esbelto de software (DES)
- Modelado ágil (MA)
- Proceso unificado ágil (PUA o AUP por sus siglas en ingles)

10. Las metodologías orientadas a objetos se clasifican en:

**Object Modelling Technique-OMT/OOSE/PU/BOOCH.**

Jackson Structured Design –JSD-RUP.

“el análisis de requisitos, que es la relación entre la asignación de software al nivel del sistema y el diseño del software, se quedó atrás por lo que empezaron a surgir diferentes métodos [...] Booch, OOSE (Object Oriented Software Engineering) y OMT (Object Modeling Technique)” (UNAM, s. f.)

## REFERENCIAS

Ambler, S. W. (2002). Agile Modeling: Effective Practices for eXtreme Programming and the Unified Process. Wiley.

Aricapa, M. I. L. (2007). MECANISMO DE CONSULTA EN LÍNEA SOBRE PROGRAMACIÓN EXTREMA (XP), SCRUM Y CRYSTAL, METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE, DIRIGIDO A ESTUDIANTES Y PROFESIONALES.

Content.pdf. (s. f.). Recuperado 30 de enero de 2025, de <https://repositorio.grial.eu/server/api/core/bitstreams/1e5798f2-3f7b-42c1-a287-137ff49ecc01/content>

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modeling Language User Guide (2nd ed.). Addison-Wesley.

García-Peñalvo, F. J., & García-Holgado, A. (s. f.). INGENIERÍA DE SOFTWARE I.

Introducción Ingeniería de Software | PDF | Ingeniería de software | Software. (s. f.). Recuperado 30 de enero de 2025, de <https://es.scribd.com/document/462507105/Introduccion-Ingenieria-de-Software>

Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd ed.). Prentice Hall.

Yourdon, E. (1989). Modern Structured Analysis. Prentice Hall.

PRESSMAN. (2010). *Ingeniería en software. Un enfoque práctico*.

<https://www.iberlibro.com/INGENIERIA-SOFTWARE-PRESSMAN-Mcgraw-Hill/30039114600/bd>

*Qué es UML*. (s. f.). <http://profesores.fi-b.unam.mx/carlos/aydoo/intro.html>