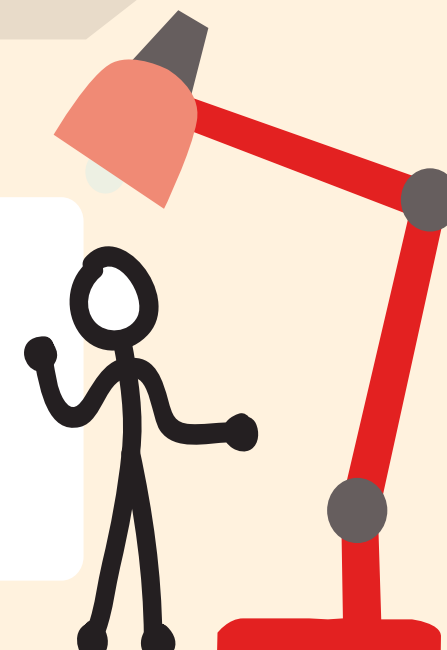


Introducción al UML



Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML





Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML

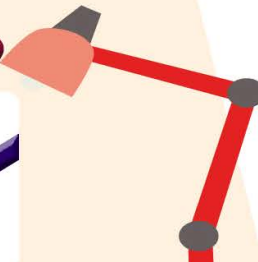


Tabla de Contenido

1. Introducción.....	4
2. Estructura de contenido.....	5
3. ¿Qué es UML?	6
4. Alcance del UML.....	6
5. Introducción a la orientación a objetos	6
5.1. Principios de la orientación a objetos	7
5.1.1. Encapsulación	7
5.1.2. Clase	7
5.1.3. Abstracción	7
5.1.4. Herencia	8
5.1.5. Especialización y generalización	8
5.1.6. Polimorfismo.....	8
5.1.7. Composición y agregación	9
6. Bloques de construcción de UML	9
6.1. Elementos.....	9



Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



6.1.1. Elementos estructurales	9
6.1.2. Elementos de comportamiento	11
6.1.3. Elementos de agrupación	11
6.1.4. Elementos de anotación	11
7. Relaciones	12
8. Ciclos de vida de software.....	12
9. Material de apoyo	14
10. Glosario.....	15
11. Referencias bibliográficas	16
12. Control del documento	17
Créditos	18
Creative Commons	18





Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



1. Introducción

Desde los años 90's, en el desarrollo de sistemas de información se ha venido utilizando el estándar UML como medio de comunicación de modelos de arquitectura dando a entender el funcionamiento y todas las especificaciones del mismo en gran detalle. Tener un medio con el cual poder abstraer la realidad y comunicarla a través de la orientación a objetos de forma clara, precisa, y concisa ha sido y seguirá siendo un elemento clave para arquitectos, diseñadores y desarrolladores de soluciones informáticas a nivel mundial, antes de poder realizar cualquier tarea utilizando el estándar, se hace necesario poder reconocer los elementos y conceptos básicos del mismo para poder comunicar las ideas y los modelos correctamente.

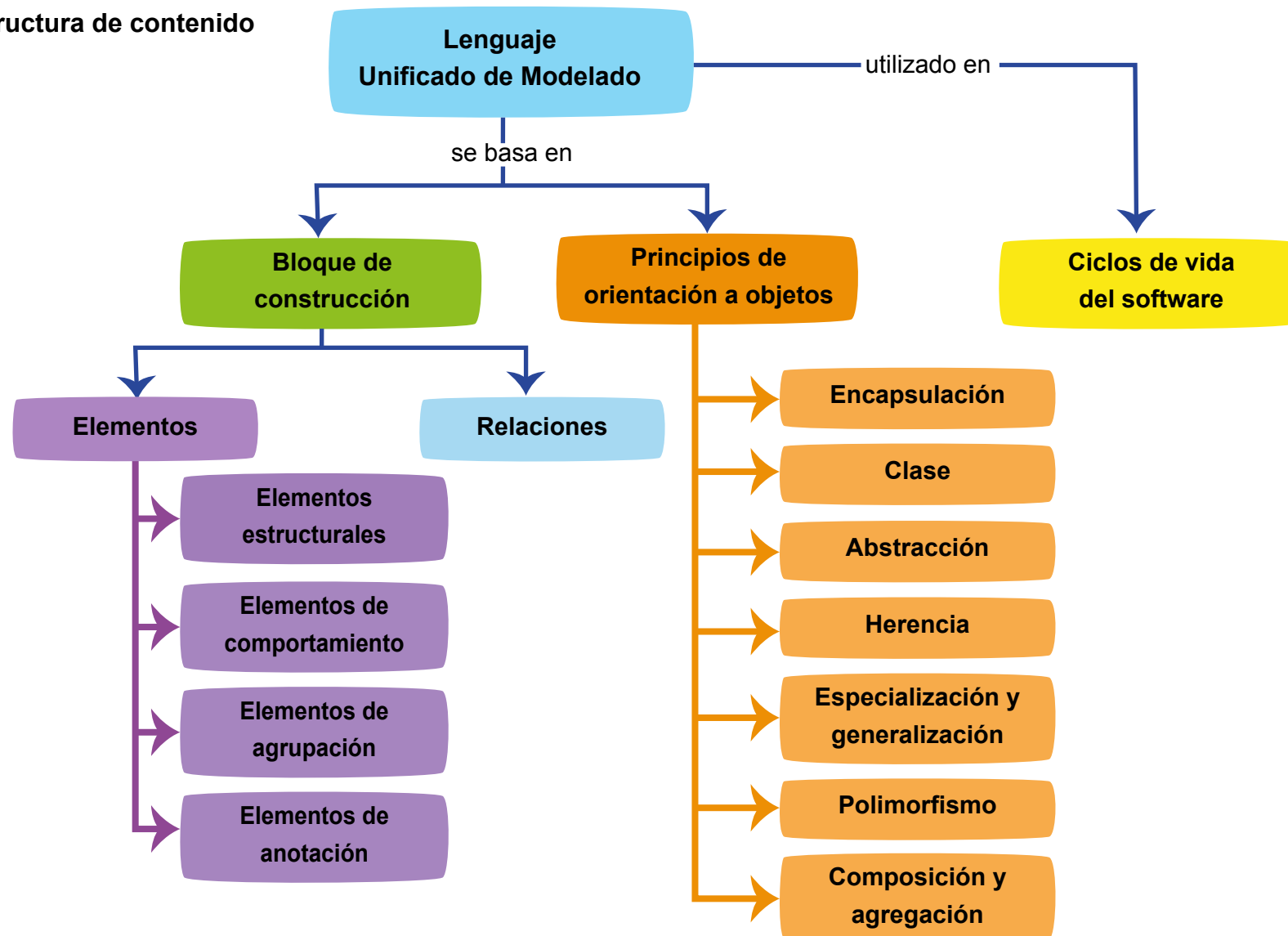




Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



2. Estructura de contenido



Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



3. ¿Qué es el UML?

El lenguaje unificado de modelado UML es un estándar gráfico que permite la especificación, construcción, visualización y documentación conceptual de diferentes tipos de sistemas y sus artefactos ya sean basados o no en software, éste representa una colección de buenas prácticas de ingeniería que pueden permitir el correcto modelado de sistemas tan simples o complejos como pueda imaginar el ser humano en su necesidad.

El estándar es ampliamente utilizado en diferentes procesos de modelado y desarrollo; y con otros tipos de arquitecturas gracias a su amplio alcance y conjunto de elementos, relaciones, bloques y diagramas que posibilitan la extensibilidad, flexibilidad y robustez del lenguaje.

4. Alcance del UML

De acuerdo con la Organización de Administración de Objetos (en inglés, Object Management Group - OMG) el lenguaje unificado de modelado define su alcance de la siguiente manera:

“El objetivo del UML es proveer a arquitectos de sistemas, ingenieros de software y desarrolladores de software con herramientas para análisis,

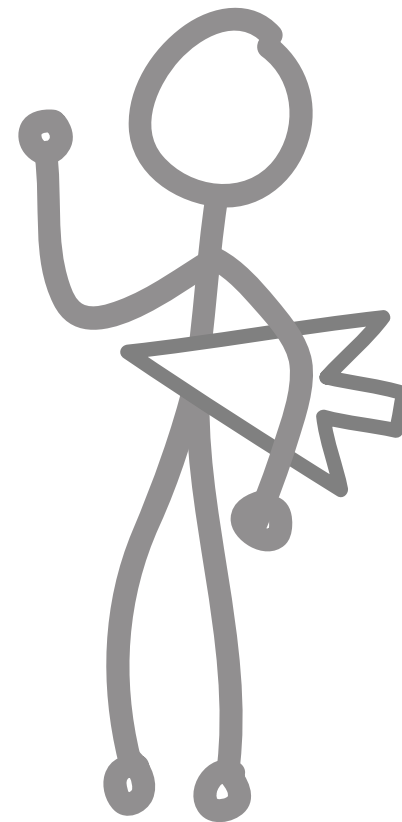
diseño e implementación de sistemas basados en software, modelado de negocios y procesos similares” (Object Management Group, 2015)

5. Introducción a la orientación a objetos

Antes de poder empezar a conocer los elementos base utilizados dentro de las buenas prácticas y el estándar UML como tal, se hace necesario abordar el tema de la orientación a objetos ya que este es más antiguo que el lenguaje (UML), para comprender las diferentes unidades se deberá comprender primero el concepto de “orientación a objetos”.

Para poder comprender el concepto, primero se debe iniciar comprendiendo lo que es un “objeto”, pues bien, en este orden de ideas, un objeto es un elemento claramente identificable dentro del mundo real y fácilmente representable, comprensible y asignable aún cuando se ha aplicado abstracción al mismo, en pocas palabras un objeto puede ser cualquier cosa como por ejemplo un reloj, un teléfono, un libro, una tapa etc.

Al igual que en la realidad, en UML todo objeto posee una serie de atributos que conforman su estructura, por ejemplo un automóvil cuenta con atributos como color, modelo, cantidad de puertas,





Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



ventanas, etc. estos lo describen y le dan un valor representativo y/o tangible; también hay métodos que otorgan un comportamiento o capacidad de realizar acciones que pueden llegar a modificar sus atributos o producir resultados específicos por ejemplo, nuevamente en el automóvil dicho comportamiento puede ser acelerar, frenar, girar, subir o bajar las ventanas entre muchas otras.

Así las cosas, el principio de orientación a objetos en el UML implica conocer que todos los sistemas concebidos están compuestos por objetos que interactúan entre sí mediante las operaciones propias que realizan por su comportamiento propio.

5.1. Principios de la orientación a objetos

La orientación a objetos en el diseño de sistemas requiere de conocer sus principios básicos de funcionamiento para realizar correctamente un modelado que represente la realidad del producto, para esto es necesario entender siete conceptos principales:

5.1.1. Encapsulación

El principio de encapsulación consiste en poder ocultar atributos y métodos de un determinado objeto en relación a otro ya que algunos de estos

se consideran de funcionamiento interno, es decir, consiste en definir qué comportamientos y características de un objeto A pueden ser “visualizadas y utilizadas” por un objeto B.

5.1.2. Clase

Una clase es un “modelo” o “maqueta” de un objeto, a partir de esta se pueden crear los objetos componentes del sistema que se está modelando. La clase contiene todos aquellos atributos y métodos que se consideran comunes entre los objetos del mismo tipo por lo que a partir de esta se crean las instancias de los mismos.

5.1.3. Abstracción

Este principio puede considerarse como uno de los más importantes en el modelado ya que éste consiste en tener la capacidad de crear una representación de la realidad de un objeto o del mundo real teniendo en cuenta únicamente las propiedades y comportamientos del mismo, así las cosas, se puede decir que todos los objetos modelados utilizando el estándar UML son una representación abstracta del mundo real.





Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML

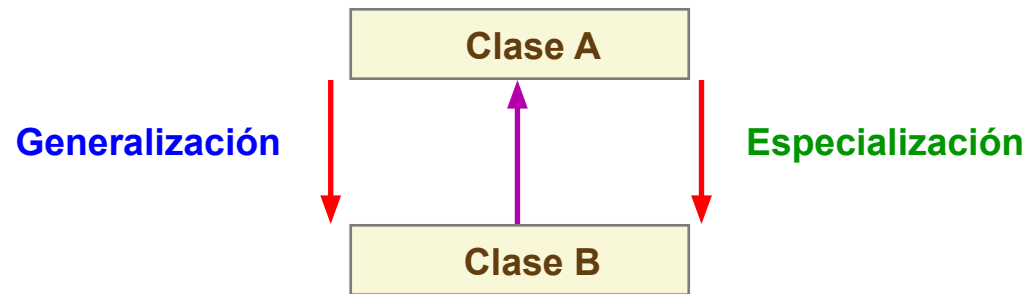


5.1.4. Herencia

Se le conoce como herencia al principio que permite a una o varias clases, beneficiarse de los atributos y métodos de otra, haciendo que estas puedan construirse con los mismos componentes de la clase “padre” (también conocida como súper clase o clase general) y de algunos otros componentes más específicos. La herencia permite tener una jerarquía de clases determinada en la que las del nivel más alto se consideran “clases abstractas” y las de más bajo nivel se les consideran “clases concretas”.

5.1.5. Especialización y generalización

Sabiendo que gracias a la herencia es posible tener una jerarquía de clases, se puede hablar entonces de otro principio importante y es la especialización y la generalización. Este consiste en una relación existente dentro de dicha jerarquía en la que la “especialización” determina que una clase es subclase de otra, mientras que, la “generalización” indica una relación contraria ya que esta muestra que una clase abstracta es padre de una o varias más especializadas en esta.



5.1.6. Polimorfismo

En el mundo real muchos objetos suelen tener diferentes comportamientos completamente semejantes entre sí por ejemplo, se tiene un objeto de tipo cama el cual tiene como propósito esencial el permitir que una persona se acueste y descanse, luego se tiene un objeto de tipo sofá el cual tiene como propósito permitir que una o varias personas se sienten y descanse, pero qué pasaría si se quisiera un objeto que se comporte como los dos a la vez, entonces lo más probable es que se tenga en la mente un objeto de tipo sofá-cama que se puede comportar como cualquiera de los dos anteriores sin dejar de ser un único objeto. Así el polimorfismo es esa capacidad de hacer que un objeto determinado tenga la capacidad de comportarse de múltiples formas.



Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



5.1.7. Composición y agregación

Existen objetos a los que se le denominan complejos ya que estos están compuestos de otros objetos de diferentes tipos, la relación que existe entre estos es lo que se conoce como composición o agregación, para ejemplo de esto se tiene un objeto camisa que puede estar compuesta de los objetos tela, hilo y botón. Cuando se habla de composición se habla de una relación en la que un objeto no puede existir sin el otro, es decir, los objetos componentes no pueden compartirse por lo que la destrucción del objeto compuesto conlleva a la destrucción de los componentes, a esto se le conoce como composición fuerte, mientras que, la agregación o composición débil se le conoce a aquella relación en la que un objeto puede existir sin la necesidad de alguno de sus componentes es decir estos pueden ser compartidos entre objetos complejos y la destrucción del objeto complejo no implica la destrucción de uno o todos sus componentes.

6. Bloques de construcción de UML

UML es un lenguaje robusto, lo que implica que se tenga una definición del mismo claramente delimitada y especificada que permita a éste ser útil en múltiples procesos de desarrollo, para eso el estándar usa tres bloques de construcción de

modelos que se especifican a continuación:

6.1. Elementos

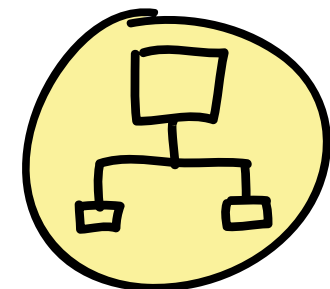
Son componentes del lenguaje que permiten dar estructura a un modelo, describir su comportamiento y delimitar el mismo haciendo que cualquier diseño sea ordenado, claro y legible para cualquiera de los involucrados en el desarrollo.

Estos elementos están divididos en de la siguiente forma:

6.1.1. Elementos estructurales

Son aquellos que por su naturaleza estática pueden representar cosas de nivel conceptual o material estos son:

- Las interfaces: estas definen las operaciones obligatorias que debe realizar una clase para tener un determinado comportamiento mas no especifica cómo debe realizarse dicho funcionamiento. Su símbolo es:

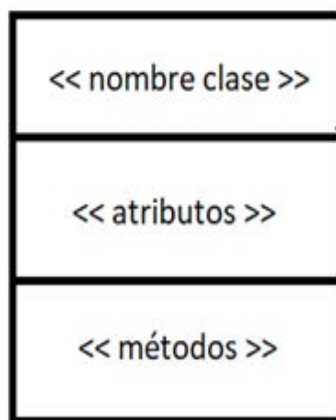




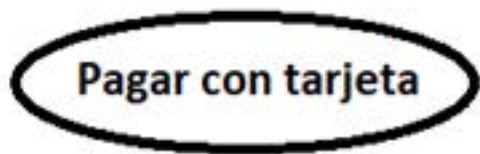
Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



- Las clases: una clase es un “modelo” o “maqueta” de un objeto, a partir de esta se pueden crear los objetos componentes del sistema que se está modelando su símbolo es un rectángulo:



- Los casos de uso: son la descripción de un conjunto de acciones que se realizan en un modelo y que deberán ser ejecutadas por el sistema, este trata las acciones en forma general, es decir no entra en detalle de dichas acciones, se representa por medio de un ovalo marcado con la acción en verbo infinitivo:



- Los componentes: un componente es una representación de una parte física del sistema, es decir, representa un empaquetamiento de diferentes clases, colaboraciones, interfaces y demás elementos lógicos, se representa de la siguiente manera:



- Los nodos: estos son utilizados en los diagramas de despliegue y representan recursos en el modelo, suelen tener capacidades de procesamiento o almacenamientos de datos, estos tienen diferentes formas de representarse, pero la más común suele ser una caja o cubo marcado con el nombre del recurso que se utilizará como nodo:





Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



6.1.2. Elementos de comportamiento

Estos representan la parte dinámica de los modelos de UML, ya que implican comportamiento del modelo a través del tiempo. Estos existen de dos tipos:

- Las interacciones: estas son el mensaje que se puede enviar entre partes del modelo o grupos de objetos dentro de un determinado ámbito, a través de las interacciones se puede alcanzar un propósito específico, su representación es una flecha dirigida desde el emisor hasta el receptor del mensaje marcada en la parte superior de la misma.

enviarMensaje() →

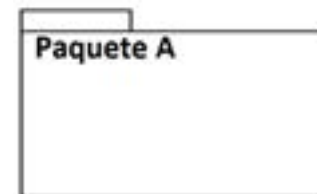
- Las máquinas de estado: es un comportamiento que determina una serie de estados determinados que sufren los objetos de un sistema y las reacciones a estos, se representa por un rectángulo redondeado con el nombre del estado.

Validado

6.1.3. Elementos de agrupación

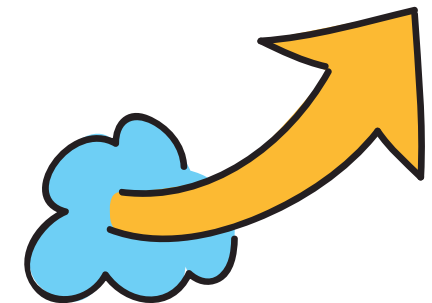
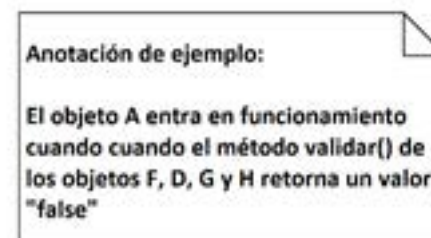
Estos representan la parte organizativa del modelo ya que su propósito es ordenar, delimitar y dar

coherencia al mismo, este grupo está compuesto principalmente por un elemento conocido como paquete y se representa como una carpeta (un rectángulo horizontal con una pestaña en la parte superior izquierda) marcado con el nombre del mismo.



6.1.4. Elementos de anotación

Representan la parte explicativa del modelo en UML se representan mediante una figura de "Sticky Note" llamada anotación en la que se puede describir, calificar, hacer observaciones, y demás pertinentes sobre cualquier elemento del modelo o su funcionamiento.





Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



7. Relaciones

Las relaciones en UML son uno de los pilares base para la creación del modelo estático y se aplican única y exclusivamente a los diagramas de clases, entre las clases e interfaces. Son de los siguientes tipos:

- **Asociación:** es la relación más sencilla en la notación, se utiliza para indicar que dos clases se encuentran vinculadas entre sí, se reconoce como una relación estructural entre los dos elementos que otorga cardinalidad a los mismos. Se representa por una flecha dirigida sólida.
- **Pertenencia:** Esta relación de tipo semántica indica el principio de composición (composición y agrupación) de forma binomial es decir entre solo dos elementos; esta relación, aunque semántica tiene un sentido constructivo dentro del modelo. Su representación está dada por una flecha sólida con un diamante (lleno o vacío) en su origen.



El apartado de relaciones se tomará más a fondo en la unidad de diseño orientado a objetos “Realizando el diseño orientado a objetos”.

8. Ciclos de vida del software

Describe a grandes rasgos cómo se realizará el desarrollo del software desde el inicio hasta el final. La idea del ciclo de vida es una planeación con el propósito de definir las distintas etapas que se requieren para validar el desarrollo de la aplicación y obtener un resultado funcional incluyendo su mejora futura.

Estos ciclos tienen como propósito principal organizar el desarrollo y detectar errores en el sistema antes de pasar a la implementación del mismo, dando a los desarrolladores herramientas para poder hacer un fino trabajo de construcción y mantenimiento de la solución.

Un ciclo de vida normal de software puede contener las siguientes fases básicas sujetas a modificación de acuerdo al estilo de ciclo que de termine el equipo de trabajo y/o el cliente:

- **Definición de objetivos:** en este punto se definen los objetivos y el alcance del proyecto que se desea realizar.
- **Análisis de los requisitos y viabilidad:** se realiza una recopilación de los requisitos del cliente y su posterior análisis de viabilidad dentro del proyecto.





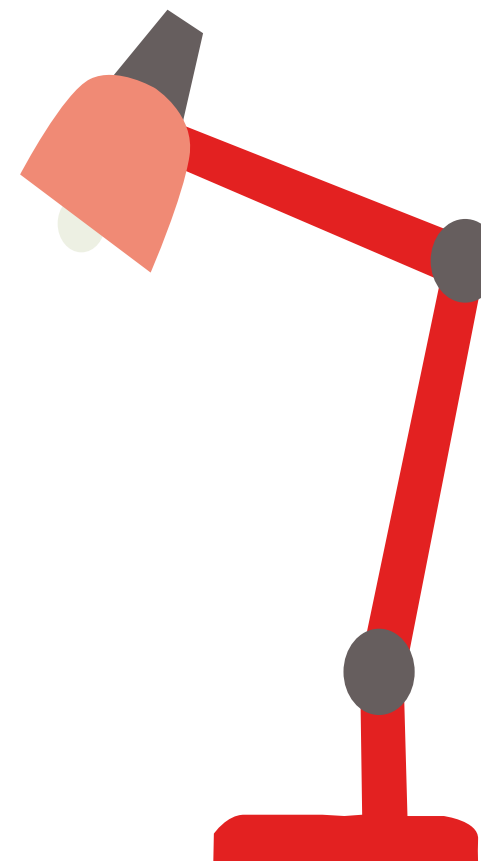
Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



- **Diseño general:** se realiza un primer modelo general del sistema, verificando que los requisitos generales se cumplan y se valida la arquitectura de la aplicación.
- **Diseño en detalle:** se realiza el modelo específico de la solución, aquí se lleva al detalle cada parte modelada ya que este es el punto de partida de la codificación de la solución.
- **Codificación:** se realiza implementación del modelo en un lenguaje de programación siguiendo al detalle el modelo realizar en las etapas de diseño.
- **Pruebas individuales:** se realizan pruebas a cada parte de codificada del modelo en busca de errores para retroalimentación y recodificación de la solución a dichos problemas garantizando que cada una de estas se realizó de acuerdo a las especificaciones.
- **Pruebas de integración:** es la prueba en conjunto para verificar que el producto obtenido corresponde al modelado y los requerimientos del cliente. Garantiza que los diferentes módulos son capaces de convivir armónicamente dentro del sistema final.

- **Validación:** se realiza para garantizar que el software desarrollado cumple con las especificaciones originales.
- **Implementación:** en este punto el desarrollo pasa a producción del cliente y mejora continua.
- **Mantenimiento y mejora continua:** esta fase del desarrollo es el punto final y reinicio de todo el proceso ya que los ciclos de vida ideales son iterativos con el fin de mejorar cada vez más el producto y alargar su tiempo de vida útil. Aquí se realizan mejoras a la funcional ya establecida o se agregan componentes acordes a las nuevas necesidades que surgen.

Durante todo el ciclo de vida del software, y el desarrollo de los sistemas, UML está presente ya sea para el análisis, diseño y modelado de la solución o de guía en la codificación, en todo caso este estándar está comunicando a cada especialista la idea necesaria para que todos alcancen el mismo fin.





Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



9. Material de apoyo



Para complementar los diferentes conceptos adquiridos en el presente documento, se recomienda a los aprendices consultar el siguiente documento: Especificación de sistemas de software en UML.



Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



10. Glosario

Artefacto: según el diccionario de la lengua española es un objeto, especialmente una máquina o un aparato, construido con una cierta técnica para un determinado fin. En el caso del estándar UML, refiere a un objeto o componente construido para un determinado fin en un sistema de información.

Estándar: según el diccionario de la lengua española “Que sirve como tipo, modelo, norma, patrón o referencia”.

Modelo: arquetipo o punto de referencia para imitarlo o reproducirlo.

Notación: sistema de signos convencionales que se adopta para expresar conceptos matemáticos, físicos, químicos, etc.





Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



11. Referencias bibliográficas

Debrauwer, L., & Van Der Heyde, F. (2016). UML 2.5 - Iniciación, ejemplos y ejercicios corregidos. Barcelona, España: Ediciones ENI.

Kimmel, P. (2006). Manual de UML. En P. Kimmel, Manual de UML. Interamericana de España: McGraw Hill.

López, T., Costal Costa, E., Sancho Samsó, D., & Ribera, M. (2003). Especificación de sistemas software en UML. Catalunya: Universitat Politècnica de Catalunya.

Object Management Group. (2000). OMG Unified Modeling Language Specification. OMG.

Object Management Group. (2015). OMG Unified Modeling Language Version 2.5. Object Management Group.



Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



12. Control del documento

Versión	Fecha	Estado	Cambios	Autor
1	31-07-2006	Nuevo		Repositorio Sena
2	01-09-2017	Modificado y/o actualizado	Se modificó y actualizó el contenido al estándar UML 2.5	Mauricio Eduardo Campuzano Méndez



Principios del Análisis y Diseño Orientado a Objetos, Utilizando el Estándar UML



Créditos

Equipo de adecuación gráfica
Centro de comercio y servicios

SENA Regional Tolima

Línea de Producción

Director Regional

Félix Ramón Triana Gaitán

Subdirector de Centro

Álvaro Fredy Bermúdez Salazar

Coordinadora de formación profesional

Gloria Ines Urueña Montes

Senior equipo de adecuación

Claudia Rocío Varón Buitrago

Experto temático

Mauricio Eduardo Campuzano Méndez

Asesor pedagógico

Ricardo Palacio Peña

Guionistas

Genny Carolina Mora Rojas

Jesús Bernardo Novoa Ortiz

Diseño y diagramación

Diana Katherine Osorio Useche

Pedro Nel Cabrera Vanegas

Ismael Enrique Cocomá Aldana

Programadores

Davison Gaitán Escobar

Héctor Horacio Morales García

Ivan Dario Rivera Guzman

Oscar Daniel Espitia Marín



Creative commons

Atribución, no comercial, compartir igual.

Este material puede ser distribuido, copiado y exhibido por terceros si se muestra en los créditos. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

cc creative commons

