PORTADA

***PROGRAMA DE ESTUDIO DE ARQUITECTURA DE PLATAFORMAS Y SERVICIOS DE TECNOLOGÍAS DE LA INFORMACIÓN***

***CONOCIENDO LOS DIAGRAMAS UML***

***CHICLAYO , JUNIO 2023***

I. Introducción

En el mundo actual, los sistemas de tecnología de la información (TI) desempeñan un papel fundamental en el funcionamiento de las organizaciones. Estos sistemas automatizan procesos, gestionan datos, facilitan la comunicación y mejoran la eficiencia operativa. Para desarrollar y mantener estos sistemas, es necesario llevar a cabo un proceso de análisis y diseño adecuado.

El análisis y diseño de sistemas TI es la etapa inicial del ciclo de vida del desarrollo de software. Durante esta etapa, se identifican las necesidades del negocio, se definen los requisitos del sistema y se crea una estructura lógica para la solución propuesta. Esta fase es crucial, ya que sienta las bases para la implementación y el mantenimiento del sistema.

En este contexto, los diagramas de la Lenguaje Unificado de Modelado (UML) desempeñan un papel esencial. UML proporciona un conjunto de notaciones gráficas que permiten visualizar, especificar, construir y documentar los aspectos de un sistema de software. Estos diagramas ayudan a representar de manera clara y precisa los diferentes aspectos del sistema, incluyendo su estructura, comportamiento, interacciones y relaciones entre los componentes.

La importancia de los diagramas UML radica en su capacidad para facilitar la comunicación efectiva entre los miembros del equipo de desarrollo y las partes interesadas. Estos diagramas ofrecen una representación visual comprensible del sistema, lo que permite una mejor comprensión de su funcionamiento y una colaboración más efectiva entre los diferentes actores involucrados en el proceso de desarrollo de software.

El objetivo principal de esta monografía es explorar y analizar el papel de los diagramas UML en el proceso de desarrollo de software. Se analizarán los diferentes tipos de diagramas UML, su utilidad en las distintas etapas del proceso de desarrollo y se proporcionarán ejemplos y mejores prácticas para su creación y utilización efectiva. Además, se evaluarán los beneficios y desafíos asociados con el uso de los diagramas UML en el desarrollo de sistemas de TI.

I. Fundamentos de UML

Es un lenguaje de modelado visual utilizado en el campo del desarrollo de software para representar, especificar, construir y documentar los artefactos de un sistema. Proporciona una notación estándar y un conjunto de reglas y principios para representar gráficamente diferentes aspectos del sistema.

A. Definición de UML

El Lenguaje de Modelado Unificado (UML, por sus siglas en inglés) es un lenguaje estándar de modelado utilizado en el análisis y diseño de sistemas de software. UML proporciona una notación visual y semántica para representar, especificar, construir y documentar los artefactos de un sistema. Estos artefactos incluyen tanto aspectos estáticos como dinámicos del sistema, como su estructura, comportamiento, interacciones y procesos.

UML se basa en un metamodelo, que es una especificación formal de los elementos y las relaciones utilizadas en el lenguaje. El metamodelo define los conceptos fundamentales de UML y las reglas para su uso, lo que garantiza la consistencia y la coherencia en la creación de modelos. Los modelos en UML se construyen mediante la combinación y la interconexión de elementos de modelado, que se representan gráficamente en diferentes tipos de diagramas.

B. Historia y evolución de UML

La historia de UML se remonta a la década de 1990, cuando tres reconocidos expertos en metodología de software, Grady Booch, James Rumbaugh e Ivar Jacobson, se unieron para desarrollar un lenguaje de modelado unificado. Cada uno de ellos aportó sus enfoques y experiencias metodológicas, fusionando sus respectivas metodologías (Booch, OMT y OOSE) en una sola, conocida como UML.

En 1997, UML fue adoptado como estándar por el Object Management Group (OMG), una organización internacional que se encarga de establecer estándares para la industria del software. La adopción de UML como estándar permitió su amplia difusión y su integración en herramientas de modelado y entornos de desarrollo.

A lo largo de los años, UML ha experimentado varias evoluciones y versiones. La primera versión, UML 1.x, estableció los fundamentos del lenguaje y proporcionó una notación para los principales tipos de diagramas. Con UML 2.0, lanzado en 2004, se realizaron mejoras significativas, como la inclusión de perfiles para adaptar UML a dominios específicos y la introducción de diagramas de estructura compuesta y diagramas de interacción mejorados.

Posteriormente, se lanzaron versiones posteriores, como UML 2.1, 2.2 y 2.3, que introdujeron mejoras adicionales y correcciones de errores. Además, se han desarrollado perfiles y extensiones específicas para áreas como el modelado de procesos de negocio (BPMN), el desarrollo de sistemas embebidos (SysML) y el modelado de aplicaciones web (UML for Web Application Development).

C. Ventajas y beneficios de utilizar UML en el análisis y diseño de sistemas

La utilización de UML en el análisis y diseño de sistemas ofrece una serie de ventajas y beneficios significativos:

1. Comunicación efectiva: UML proporciona una notación visual estándar y comprensible, lo que facilita la comunicación entre los miembros del equipo de desarrollo, los stakeholders del proyecto y otras partes interesadas. La notación gráfica de UML permite representar de manera clara y concisa los elementos y las relaciones del sistema, lo que ayuda a evitar malentendidos y confusiones.
2. Abstracción y nivel de detalle adecuado: UML permite representar diferentes niveles de abstracción en el modelado de sistemas. Los diagramas de alto nivel, como los diagramas de casos de uso, brindan una visión general del sistema y sus funcionalidades principales, mientras que los diagramas más detallados, como los diagramas de clases y de secuencia, permiten especificar y diseñar componentes y comportamientos de manera más precisa.
3. Visualización y comprensión del sistema: UML ofrece una amplia gama de diagramas que cubren diferentes aspectos del sistema. Los diagramas de clases muestran las entidades del sistema y las relaciones entre ellas, los diagramas de secuencia representan la interacción entre los objetos a lo largo del tiempo, los diagramas de actividad describen el flujo de trabajo y los procesos, y los diagramas de estado muestran el comportamiento de los objetos en diferentes estados.
4. Documentación y mantenibilidad: UML permite generar documentación automática a partir de los modelos creados, lo que facilita la comprensión y el mantenimiento del sistema a lo largo del tiempo. Los modelos UML actúan como una fuente de información precisa y actualizada sobre el sistema, lo que ayuda a realizar cambios y mejoras de manera más eficiente y controlada. Además, los diagramas UML proporcionan una base sólida para la documentación técnica y la transferencia de conocimiento entre los miembros del equipo.
5. Flexibilidad y adaptabilidad: UML es un lenguaje flexible que se adapta a diferentes metodologías y enfoques de desarrollo de software. Puede utilizarse en diferentes etapas del ciclo de vida del desarrollo, desde la especificación de requisitos hasta la implementación y la documentación. Además, UML puede combinarse con otras técnicas y herramientas, como patrones de diseño y lenguajes de programación específicos, lo que amplía su aplicabilidad y utilidad.
6. Estándar de la industria: UML es un estándar reconocido y ampliamente utilizado en la industria del software. Su adopción garantiza la compatibilidad y la interoperabilidad entre diferentes herramientas y entornos de desarrollo. Además, UML fomenta la reutilización de modelos y componentes, lo que acelera el proceso de desarrollo y reduce los costos.

II. Diagramas de UML

A. Visión general de los tipos de diagramas de UML

UML ofrece una amplia gama de diagramas que se utilizan para representar diferentes aspectos de un sistema. Algunos de los tipos de diagramas más comunes son:

Diagrama de Clases: representa la estructura estática del sistema, mostrando las clases, sus atributos, métodos y relaciones.

Diagrama de Casos de Uso: describe la funcionalidad del sistema desde la perspectiva del usuario, mostrando los actores, los casos de uso y las relaciones entre ellos.

Diagrama de Secuencia: muestra la interacción entre los objetos del sistema a lo largo del tiempo, enfocándose en la secuencia de mensajes intercambiados entre ellos.

Diagrama de Actividad: representa el flujo de actividades y acciones dentro de un proceso o función, mostrando la secuencia de acciones, decisiones y ramificaciones.

Estos son solo algunos ejemplos, pero UML también incluye otros diagramas, como el diagrama de componentes, el diagrama de despliegue, el diagrama de estados, el diagrama de paquetes y el diagrama de objetos, que se utilizan para representar aspectos específicos del sistema.

B. Diagrama de Clases

Concepto y propósito

El diagrama de clases es uno de los diagramas más utilizados en UML y representa la estructura estática del sistema. Se centra en las clases del sistema, sus atributos, métodos y relaciones entre ellas. El propósito principal es modelar la estructura del sistema y definir las clases necesarias para implementarlo.

Elementos y notaciones utilizadas

Los elementos principales en un diagrama de clases son las clases, que se representan como rectángulos divididos en tres secciones: nombre de la clase, atributos y métodos. Las relaciones entre las clases se muestran con líneas y flechas, como la relación de asociación, la herencia y la agregación

Ejemplos de uso y casos prácticos

El diagrama de clases es ampliamente utilizado en el análisis y diseño de sistemas. Se utiliza para identificar las entidades y conceptos clave del sistema, definir las relaciones entre ellas y proporcionar una visión general de la estructura del sistema. También se utiliza como base para generar código y facilitar la comunicación entre los miembros del equipo de desarrollo.

C. Diagrama de Casos de Uso

Concepto y propósito

El diagrama de casos de uso se utiliza para capturar y representar las interacciones entre los usuarios (actores) y el sistema. Describe las funcionalidades del sistema desde la perspectiva del usuario, mostrando los casos de uso, los actores y las relaciones entre ellos. El propósito principal es comprender cómo los usuarios interactúan con el sistema y qué funcionalidades ofrece el sistema para satisfacer sus necesidades.

Elementos y notaciones utilizadas

En un diagrama de casos de uso, los actores se representan como figuras externas al sistema y los casos de uso se representan como elipses. Las relaciones entre los actores y los casos de uso se muestran con líneas sólidas. También se pueden utilizar extensiones y relaciones de inclusión para mostrar relaciones más detalladas.

Ejemplos de uso y casos prácticos

El diagrama de casos de uso se utiliza en la etapa de análisis de requisitos para identificar las funcionalidades que debe proporcionar el sistema desde la perspectiva del usuario. Es útil para establecer una base sólida para el diseño del sistema y garantizar que las necesidades de los usuarios se satisfagan de manera efectiva. También se utiliza como una herramienta de comunicación entre los desarrolladores, los usuarios y otras partes interesadas.

D. Diagrama de Secuencia

Concepto y propósito

El diagrama de secuencia muestra la interacción entre los objetos del sistema a lo largo del tiempo. Se enfoca en la secuencia de mensajes intercambiados entre los objetos y cómo se comportan durante una determinada funcionalidad. El propósito principal es capturar y representar el flujo de eventos y la interacción entre los objetos.

Elementos y notaciones utilizadas

En un diagrama de secuencia, los objetos se representan como rectángulos con sus nombres, y las líneas de vida (líneas verticales) indican la existencia temporal de los objetos durante la secuencia. Los mensajes intercambiados entre los objetos se muestran con flechas y se indican los parámetros y el orden de los mensajes.

Ejemplos de uso y casos prácticos

El diagrama de secuencia se utiliza para modelar y comprender el flujo de eventos y la interacción entre los objetos del sistema. Es útil para identificar los mensajes y las dependencias entre los objetos, así como para detectar posibles problemas de diseño y colaboración. También es utilizado para comunicar y discutir la lógica de interacción entre los diferentes componentes del sistema.

E. Diagrama de Actividad

Concepto y propósito

El diagrama de actividad muestra el flujo de actividades y acciones dentro de un proceso o función. Se utiliza para representar el flujo de control, las decisiones, las ramificaciones y las actividades paralelas en un sistema. El propósito principal es modelar el comportamiento dinámico y los procesos internos del sistema.

Elementos y notaciones utilizadas

En un diagrama de actividad, las actividades se representan como rectángulos con nombres y las transiciones entre actividades se muestran con flechas. También se pueden utilizar decisiones, bifurcaciones y uniones para representar el flujo condicional y paralelo.

Ejemplos de uso y casos prácticos

El diagrama de actividad se utiliza para modelar y comprender los flujos de trabajo, los procesos de negocio y las operaciones internas del sistema. Es útil para identificar los pasos necesarios para completar una tarea, detectar posibles cuellos de botella y optimizar los procesos. También se utiliza para documentar y comunicar los procedimientos y la lógica del sistema.

F. Otros diagramas de UML

Diagrama de Componentes

El diagrama de componentes se utiliza para representar la estructura y las relaciones entre los componentes del sistema. Los componentes son elementos modulares y reutilizables que encapsulan funcionalidades específicas. Este diagrama muestra cómo se organizan los componentes en el sistema y cómo se relacionan entre sí mediante interfaces, dependencias y asociaciones.

Diagrama de Despliegue

El diagrama de despliegue se utiliza para representar la distribución física de los componentes del sistema en el entorno de implementación. Muestra los nodos de hardware, como servidores y dispositivos, y los componentes de software que se ejecutan en cada nodo. Este diagrama es útil para comprender cómo se despliega el sistema y cómo interactúa con el entorno físico.

Diagrama de Estados

El diagrama de estados se utiliza para modelar el comportamiento interno de un objeto o componente y cómo cambia de un estado a otro en respuesta a eventos o condiciones. Representa los diferentes estados posibles y las transiciones entre ellos. Este diagrama es útil para comprender y visualizar el comportamiento dinámico del sistema.

Diagrama de Paquetes

El diagrama de paquetes se utiliza para organizar y mostrar la estructura jerárquica de los elementos del sistema en paquetes lógicos. Los paquetes agrupan elementos relacionados, como clases, componentes o diagramas, y muestran las dependencias y relaciones entre ellos. Este diagrama ayuda a organizar y gestionar la complejidad del sistema mediante la modularidad y la agrupación lógica de los elementos.

Diagrama de Objetos

El diagrama de objetos se utiliza para representar una instancia particular de una clase en un momento específico del tiempo. Muestra los objetos del sistema y las relaciones entre ellos en un escenario concreto. Este diagrama es útil para analizar y comprender la estructura y las relaciones entre los objetos en un contexto específico.

IV. Interacción de UML con el modelado de procesos de negocio

El modelado de procesos de negocio es una disciplina fundamental para comprender, mejorar y optimizar los flujos de trabajo y las operaciones de una organización. Por otro lado, UML (Unified Modeling Language) es un lenguaje de modelado estándar utilizado en el análisis y diseño de sistemas de software.

A través de esta interacción entre UML y el modelado de procesos de negocio, se logra un enfoque integral que combina la representación estructural y comportamental de un sistema, junto con la descripción detallada de los flujos de trabajo y las interacciones entre los actores involucrados. Esto permite comprender y analizar de manera más precisa los procesos de negocio, identificar áreas de mejora y diseñar soluciones eficientes.

A continuación, profundizaremos en la relación entre UML y el modelado de procesos de negocio, explorando cómo UML se aplica para representar los flujos de trabajo y los procesos, así como los beneficios y ventajas que ofrece en el análisis y diseño de sistemas.

A. Relación entre UML y el modelado de procesos de negocio

El modelado de procesos de negocio se enfoca en representar visualmente y analizar los procesos y flujos de trabajo de una organización. Su objetivo es comprender, mejorar y optimizar los procesos, identificando áreas de mejora, detectando cuellos de botella y diseñando soluciones eficientes.

UML, aunque no fue diseñado específicamente para el modelado de procesos de negocio, ofrece una variedad de diagramas y elementos que pueden ser utilizados en conjunto con otras técnicas para representar los flujos de trabajo, las interacciones y los procesos en una organización.

Al aplicar UML al modelado de procesos de negocio, se logra un enfoque integral que combina la representación de los aspectos estructurales y comportamentales de un sistema, con la descripción de los flujos de trabajo y las interacciones entre los actores involucrados.

B. Uso de UML en la representación de flujos de trabajo y procesos

Uno de los diagramas de UML más utilizados en el modelado de procesos de negocio es el diagrama de actividades. Este diagrama permite representar visualmente los pasos y las actividades necesarias para completar un proceso. Puede incluir elementos como acciones, decisiones, bifurcaciones, uniones y bucles, para modelar de manera detallada el flujo de trabajo.

El diagrama de actividades proporciona una representación clara y comprensible de las actividades, sus dependencias y las condiciones que deben cumplirse. Además, se pueden asignar recursos, tiempos y roles a las actividades, lo que permite una planificación más precisa y un análisis de eficiencia.

Además del diagrama de actividades, otros diagramas de UML también pueden utilizarse en el modelado de procesos de negocio:

1. Diagrama de secuencia: Este diagrama se utiliza para representar la interacción secuencial entre los actores y los objetos involucrados en un proceso. Por ejemplo, en el caso de un proceso de compra en línea, se pueden representar las interacciones entre el cliente, el carrito de compras, el sistema de inventario y el sistema de pagos. Esto permite visualizar el orden de las acciones, como la selección de productos, la verificación de la disponibilidad, el pago y la confirmación de la compra.
2. Diagrama de colaboración: Este diagrama se utiliza para representar la interacción entre los diferentes participantes en un proceso de negocio. Por ejemplo, en el caso de un proceso de reclutamiento, se pueden representar las interacciones entre el reclutador, los candidatos y el sistema de gestión de recursos humanos. Esto permite visualizar los mensajes intercambiados entre ellos, como la solicitud de información, la programación de entrevistas y la notificación de resultados.
3. Diagrama de casos de uso: Este diagrama se utiliza para identificar y describir las diferentes funcionalidades y los actores involucrados en un sistema. En el contexto del modelado de procesos de negocio, se puede utilizar para identificar los casos de uso relacionados con los procesos y los flujos de trabajo. Por ejemplo, en un proceso de gestión de pedidos, se pueden identificar casos de uso como "Realizar pedido", "Gestionar inventario" y "Generar factura".

Estos son solo algunos ejemplos de cómo se puede utilizar UML en el modelado de procesos de negocio. La elección de los diagramas y elementos específicos dependerá de la naturaleza y los requisitos del proceso en cuestión.

C. Ejemplos de diagramas UML utilizados en el modelado de procesos de negocio

1. Diagrama de actividades: Este diagrama se utiliza para modelar el flujo de trabajo de un proceso de negocio. Por ejemplo, en el caso de un proceso de solicitud de préstamo, se pueden representar las actividades principales, como la recopilación de información, la evaluación crediticia, la aprobación y la entrega del préstamo. Además, se pueden incluir condiciones y decisiones, como la verificación de la solvencia del solicitante, para determinar el flujo del proceso.

El diagrama de actividades puede ayudar a visualizar las etapas y las acciones necesarias en un proceso, identificar posibles puntos de mejora y optimizar la eficiencia de las actividades involucradas.

1. Diagrama de secuencia: Este diagrama se utiliza para representar la interacción temporal entre los actores y los objetos en un proceso. Por ejemplo, en el caso de un proceso de reservas en un hotel, se pueden representar las interacciones entre el cliente, el sistema de reservas y el personal del hotel. El diagrama muestra el orden de las acciones, los mensajes intercambiados y las respuestas correspondientes.

El diagrama de secuencia permite comprender cómo los actores interactúan entre sí y cómo se llevan a cabo las operaciones dentro del proceso. Esto puede ayudar a identificar posibles cuellos de botella y mejorar la eficiencia del flujo de trabajo.

1. Diagrama de colaboración: Este diagrama se utiliza para representar las interacciones entre los diferentes participantes en un proceso de negocio. Por ejemplo, en el caso de un proceso de compra en línea, se pueden representar las interacciones entre el cliente, el carrito de compras, el sistema de inventario y el sistema de pagos. El diagrama muestra los mensajes intercambiados entre los participantes y las conexiones entre ellos.

El diagrama de colaboración permite visualizar la comunicación entre los participantes y comprender cómo se coordinan y cooperan para llevar a cabo el proceso. Esto puede ayudar a identificar posibles problemas de comunicación y mejorar la colaboración entre los actores involucrados.

V. Notaciones usadas en el modelado de procesos de negocio

A. Visión general de las notaciones comunes

En el modelado de procesos de negocio, existen varias notaciones comunes utilizadas para representar y documentar los procesos empresariales. Estas notaciones proporcionan un lenguaje visual estandarizado para comunicar de manera efectiva la lógica y el flujo de trabajo de los procesos. Algunas de las notaciones comunes incluyen:

BPMN (Business Process Model and Notation)

EPC (Event-driven Process Chain)

UML Activity Diagrams (Diagramas de Actividad UML)

DFD (Data Flow Diagram)

B. BPMN (Business Process Model and Notation)

Concepto y propósito

BPMN es una notación gráfica ampliamente utilizada para el modelado de procesos de negocio. Proporciona una representación visual clara y comprensible de los procesos, permitiendo a los analistas de negocios, diseñadores y usuarios finales colaborar en la comprensión y mejora de los procesos. El propósito principal de BPMN es brindar una notación estándar para modelar y documentar los procesos de negocio de manera precisa y consistente.

Elementos y notaciones utilizadas

BPMN utiliza una variedad de elementos y notaciones para representar los diferentes aspectos de los procesos de negocio. Algunos de los elementos clave incluyen:

Flujos de secuencia: representan el flujo lógico y secuencial del proceso.

Actividades: representan las tareas o acciones realizadas en el proceso.

Eventos: representan eventos que desencadenan cambios en el flujo del proceso.

Puertas de enlace (Gateways): representan puntos de decisión o ramificación en el flujo del proceso.

Artefactos: proporcionan información adicional sobre el proceso, como anotaciones y grupos.

Además, BPMN utiliza símbolos gráficos específicos para representar estos elementos, como círculos para eventos, rectángulos redondeados para actividades y diamantes para puertas de enlace.

Ejemplos de uso y casos prácticos

BPMN se utiliza en diversas situaciones y aplicaciones, como:

Modelado de procesos de negocio: permite a los analistas de negocios representar de manera clara y comprensible los procesos de negocio existentes y propuestos, identificando las actividades, las decisiones y los flujos de trabajo involucrados.

Optimización de procesos: ayuda a identificar cuellos de botella, ineficiencias y oportunidades de mejora en los procesos de negocio, lo que permite tomar decisiones informadas para optimizar y rediseñar los procesos.

Automatización de procesos: sirve como base para implementar soluciones de automatización de procesos, como flujos de trabajo automatizados y sistemas de gestión de procesos empresariales (BPM).

Colaboración y comunicación: al proporcionar una notación común y visualmente comprensible, BPMN facilita la comunicación efectiva entre los diferentes actores involucrados en el análisis y mejora de los procesos de negocio.

VI. Conclusiones

A. Recapitulación de los principales puntos tratados en la monografía

En esta monografía, se ha explorado el análisis y diseño de sistemas TI utilizando los diagramas UML (Lenguaje de Modelado Unificado) y su interacción con el modelado de procesos de negocio. Se han cubierto los siguientes aspectos:

Fundamentos de UML: Se ha definido UML como un lenguaje visual utilizado para modelar sistemas de software. Se ha discutido su historia, evolución y las ventajas de su uso en el proceso de desarrollo de software.

Diagramas de UML: Se han presentado los diferentes tipos de diagramas de UML, incluyendo el Diagrama de Clases, Diagrama de Casos de Uso, Diagrama de Secuencia y Diagrama de Actividad. También se ha mencionado otros diagramas, como el Diagrama de Componentes, Diagrama de Despliegue, Diagrama de Estados, Diagrama de Paquetes y Diagrama de Objetos.

B. Importancia y aplicabilidad de los diagramas UML en el análisis y diseño de sistemas TI

Los diagramas UML son herramientas fundamentales en el análisis y diseño de sistemas TI. Proporcionan una representación visual clara y precisa de los diferentes aspectos de un sistema, como su estructura, funcionalidad, comportamiento y relaciones entre los componentes. Esto facilita la comprensión, comunicación y documentación de los sistemas, así como la detección de posibles problemas de diseño y la toma de decisiones informadas.

Además, los diagramas UML fomentan un enfoque orientado a objetos en el desarrollo de software, lo que permite la reutilización de componentes, la modularidad y la flexibilidad en la implementación de sistemas. También promueven la colaboración entre los desarrolladores, los usuarios y otras partes interesadas, al proporcionar un lenguaje común y comprensible para discutir y visualizar el diseño del sistema.

C. Reflexión sobre la interacción de UML con el modelado de procesos de negocio

Si bien UML se enfoca principalmente en el análisis y diseño de sistemas de software, su interacción con el modelado de procesos de negocio, representado principalmente por BPMN, es crucial. Ambas notaciones proporcionan herramientas poderosas para modelar y documentar los procesos empresariales y los sistemas de información que los respaldan.

El uso conjunto de UML y BPMN permite una visión más completa y holística de un sistema, abarcando tanto los aspectos técnicos como los aspectos empresariales. Esto facilita la alineación entre los objetivos del negocio y la implementación técnica, lo que resulta en sistemas más efectivos y eficientes.

VII. Referencias

A. Bibliografía consultada

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). El lenguaje unificado de modelado (UML). Pearson Educación.

Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Pearson Education.

Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley Professional.

B. Fuentes en línea

DesarrolloWeb.com - UML (<https://www.desarrolloweb.com/articulos/uml.html>)

Aprende UML (<http://www.aprendeuml.com/>)

Universidad de Cantabria - UML (<http://personales.unican.es/junqueraj/uml/>)

AulaVirtualFPC - UML (<https://aulavirtual.fpc.edu.uy/pluginfile.php/153509/mod_resource/content/1/08_ModeladoUML.pdf>)

UML en Español (<https://umlenespanol.blogspot.com/>)