

Universidad Tecnológica del Centro de Veracruz - Innovasolver

Proyecto: Plataforma tecnológica de georecomendación para productos y servicios en modalidad subasta inversa basado en técnicas de cómputo sensible al contexto, análisis de sentimientos, minería de opiniones.

Entregable: (A4.1) Análisis de herramientas para el diseño de diagramas UML



ÍNDICE

| | |
|--------------------------------------------------------------|----|
| RESUMEN..... | 3 |
| INTRODUCCIÓN..... | 4 |
| I. LENGUAJE UNIFICADO DE MODELADO (UML) | 5 |
| II. IMPORTANCIA DE UML | 5 |
| III. HERRAMIENTAS CASE PARA EL DISEÑO DE DIAGRAMAS UML | 6 |
| I. CONCEPTDRAW | 7 |
| II. ENTERPRISE ARCHITECT | 9 |
| III. MAGIC DRAW | 11 |
| IV. VISUAL PARADIGM..... | 13 |
| V. INNOVATOR MODELING PLATFORM..... | 14 |
| IV. CONCLUSIÓN..... | 17 |
| V. BIBLIOGRAFÍA..... | 19 |



RESUMEN

El presente documento contiene un análisis de distintas herramientas de vanguardia utilizadas en el modelado de diagramas UML (acrónimo en inglés de Unified Modeling Language). El lenguaje UML es un estándar reconocido internacionalmente, que es utilizado en la construcción de software para el diseño de los modelos que describen el funcionamiento de la aplicación en los distintos niveles que lo conforman.

En la primera parte del documento se describe el lenguaje UML, así como las ventajas que ofrece dentro del contexto para el desarrollo de una plataforma de georecomendación. En la segunda parte del documento se describen las herramientas CASE (Computer Aided Software Engineering) mejor ponderadas y más populares, con base en la lista de para la generación de modelos UML, y con respecto a las necesidades del proyecto seleccionar la que cubra el 100% de los requerimientos.

INTRODUCCIÓN

Cuando se inicia un desarrollo de software, es importante que el cliente logre definir las necesidades y las comunique correctamente al equipo de desarrollo, quienes se encargarán de diseñar a través de diferentes lenguajes y métodos los componentes necesarios para que el sistema cumpla con dichas necesidades. En la mayoría de las veces, es difícil comunicar al equipo de desarrollo la estructura de un software sobre todo si está es compleja, y es donde UML facilita el proceso a través de la realización de diagramas.

UML permite al equipo de desarrollo mejorar el software, es decir, encontrar oportunidades para que el mismo sea de calidad y permita detectar amenazas para garantizar que el sistema sea seguro. Además de proporcionar la vista estructural del sistema, a través de UML se puede comunicar el comportamiento del mismo. Los diagramas de clases permite representar la vista estructural del sistema, para que el equipo de desarrollo se encuentre bajo el mismo contexto.

La implementación de UML con una metodología de desarrollo de software se realiza de manera natural, debido a la notación y técnicas de diagramación que provee UML, convirtiéndolo en un lenguaje universal, entre desarrolladores. Para el desarrollo de una plataforma web, se pueden implementar metodologías como UWE (UML-based Web Engineering, por sus siglas en inglés), la cual es un enfoque basado en UML presentado por Koch y ampliado posteriormente. Citando a Jacobson, UWE es *un enfoque orientado a objetos, iterativo e incremental basado en el Lenguaje de Modelado Unificado y el proceso de desarrollo de software unificado*. Utilizando una metodología como UWE, el uso de UML para los modelos sería un proceso validado, aunque por su enfoque, puede ser implementado en cualquier metodología de desarrollo.

El presente documento se encuentra estructurado de la siguiente manera; en la segunda sección se presentan las ventajas de usar el Lenguaje Unificado de Modelado (UML). Posteriormente, se tienen las herramientas para el diseño de diagramas UML. En la sección tres se presenta el diseño del diagrama de clases para llevar a cabo el desarrollo de la plataforma.

I. LENGUAJE UNIFICADO DE MODELADO (UML)

UML es un lenguaje estándar¹ de modelado para software, que permite la visualización, especificación, construcción y documentación de distintos artefactos generados en un proyecto de software. UML permite dentro de un equipo de desarrollo, generar esquemas que permitan comunicar de manera estandarizada en funcionamiento del software, así como los resultados esperados de cada uno de los módulos; lo anterior debido a que los símbolos definidos por UML son utilizados en distintos países, dado que es un estándar internacional.

Además de proporcionar una notación gráfica para la especificación de distintos modelados, proporciona un significado, es decir, una semántica. El vocabulario de UML se clasifican en tres categorías: elementos, relaciones y diagramas (Jacobson Ivar, 2007). Los elementos se clasifican en estructurales, de comportamiento, de agrupación y de anotación; las relaciones manejadas en UML son dependencia, asociación y generalización; y los diagramas son casos de uso, clase, objeto, secuencia, colaboración, estados, actividad, componente y despliegue. Los mecanismos de extensibilidad que proporciona UML, que permiten a los usuarios mejorar la sintaxis y semántica, los cuales son estereotipos, valores etiquetados y restricciones (Jacobson Ivar, 2007).

II. IMPORTANCIA DE UML

Dentro del desarrollo de software, es un punto importante organizar el proceso para el desarrollo de software, en donde se definan las actividades de cada uno de los roles operantes dentro del proyecto (analista, clientes, desarrolladores, etc.). Dentro del contexto anterior, UML facilita dicha organización y la representación del sistemas, desde distintos puntos de vista, para poder satisfacer a todas las personas implicadas en el proyecto (Alarcón).

En el contexto de la plataforma “MeAlcanza.com”, debido a la gran cantidad de software que se requiere, UML permite establecer la planificación de proyectos a través del modelado de actividades, requisitos, pruebas, así como su propia arquitectura. Dentro de la plataforma participaran diferentes tipos de usuarios, que intervienen en distintas etapas

¹ Un estándar es un documento establecido por consenso, aprobado por un cuerpo reconocido, y que ofrece reglas, guías o características para que se use repetidamente. (Project Management Institute, 2012)

del proyecto, cada usuario tienen diferentes actividades en distintas etapas dentro del ciclo de vida del proyecto (“se refiere a todas las etapas por las que pasa un proyecto (Alarcón)”).

UML es independiente del ciclo de vida del proyecto, pero puede ser adaptado a la mayoría de las metodologías de software, por la versatilidad de la generación de cualquier modelo aplicable a cualquier proyecto que sea de software. Las ventajas de UML son:

- **Es un lenguaje de modelado de propósito general abierto al público**, UML no tiene propietario o dueño, y se encuentra bajo el común acuerdo de la comunidad tecnológica.
- **UML se puede trabajar con cualquier modelo o metodología de desarrollo existente**, dado que contempla todos los conceptos que se consideran necesarios para modelar un sistema.
- **UML es muy fácil de implementar**, dado que es un lenguaje universal entre los desarrolladores, cuenta con la capacidad para modelar estructuras complejas.
- **UML es un estándar**, y se aplica a cualquier tipo de modelado adaptable a cualquier tipo de industria que así lo requiera, lo cual lo convierte en un lenguaje universal para el modelado para la construcción de prototipos.
- **Emplea operaciones abstractas** como guía para variaciones futuras dentro del desarrollo de software.

III. HERRAMIENTAS CASE PARA EL DISEÑO DE DIAGRAMAS UML

Debido al auge del desarrollo de software especializado en la última década, cada vez son más el número de herramientas que existen para el diseño de diagramas bajo el estándar UML, y es común realizar diferentes tipos de preguntas en la etapa de análisis, como por ejemplo:

- ¿Cuál es la mejor herramienta en la actualidad?
- ¿Cuál se debería usar de acuerdo a las necesidades del proyecto?
- ¿Cuánto cuesta?
- ¿Qué versión de UML soporta?

Para los cuestionamientos anteriores, es conveniente realizar una lista de las herramientas existentes y valorar cual es la más adecuada, aunque algunos autores

sugieren que se debe trabajar con la herramienta más *simple* porque reduce la curva de aprendizaje². En la siguiente lista presentada, se muestran las cinco herramientas más populares, según la lista

- **CONCEPTDRAW**

Es una plataforma para el modelado reconocida internacionalmente, que permite establecer un ciclo de vida para los modelos de UML de manera dinámica. Para la herramienta “ConceptDraw Pro 11”, la compañía ofrece *“una gran cantidad de documentación visual empresarial con más de miles de patrones y cientos de plantillas para un modelado rápido”*.(Odessa, 2017)

- *FUNCIONALIDAD*

Es una plataforma integral para el modelado(diagramas), la cual permite:

- Ejecutar varias tareas de diagramación de bajo-nivel, a través de cientos de plantillas y librerías.
- Modelar presentaciones y diagramas enriquecidos a nivel empresarial.
- Nueva generación de herramientas para el modelado (diagramación), tal como:
 1. RapidDraw
 2. Live Objects
 3. Propiedades definidas por el usuario
 4. PowerEdit

- *INTEGRACIÓN*

ConceptDraw es compatible con las siguientes herramientas:

- MS VISIO: La herramienta permite importar y exportar al formato de un archivo de VISIO nativo [*.vdx, *.vsdx, *.vssx].
- ConceptDraw PROJECT v8
- ConceptDraw MINDMAP v9
- ConceptDraw Office v4

² El concepto de curva de aprendizaje fue introducido por K.J. Arrow en 1962 y se refiere al incremento de productividad con base en la experiencia acumulada.(Fudenberg & Tirole, 1983).

- **CICLO DE VIDA DE DESARROLLO DE SOFTWARE**

UML se encuentra de manera natural ligado con el proceso Unificado³ y cada una de las fases que lo representa; en la ilustración 1 se muestra la alineación entre el proceso unificado y el ciclo de vida de software. El UP es un marco donde se sigue el “proceso de desarrollo de software que describe un enfoque para la construcción, desarrollo y, mantenimiento de software”(Craig, 2002). Las fases de UP son:

- Inicio
- Elaboración
- Construcción
- Transición



Ilustración 1. Proceso Unificado comparado con el ciclo de vida de software(AIGroup, 2017).

- **VERSIÓN UML**

- UML 2.2

- **DIAGRAMAS UML**

- Dentro de ConceptDraw Rapid UML, se contemplan 13 plantillas, que contiene alrededor de 393 objetos de UML. En la imagen ilustración 2 se muestran los diagramas soportados.

³ El proceso unificado de desarrollo, se refiere a un conjunto de actividades para el desarrollo de software, centrado en la arquitectura, dirigido por casos de uso, iterativo e incremental.

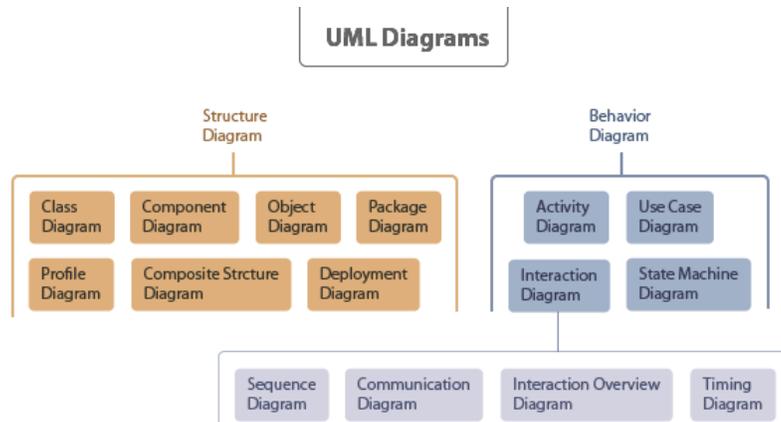


Ilustración 2. Diagramas soportador por ConceptDraw RapidUML.

○ **LICENCIA**

Se estipulan los siguientes tipos de licencia, bajo sus términos EULA:

- Single
- Plan
- Backup
- Multi-user
- Trial y/o demo
- NFR (Not For Resale)
- Académica

• **ENTERPRISE ARCHITECT**

Es una plataforma que además de ofrecer una interfaz eficiente e intuitiva, posee un núcleo sólido con soporte para UML 2.4; adicionalmente ofrece amplio banco de recursos (repositorio) y una gran comunidad de seguidores .

○ **FUNCIONALIDAD**

Es herramienta muy completa para el diseño de modelos UML, dirigida a empresas de todos los tamaños (grandes a pequeñas). *Con uso en diferentes gamas de la industria, como lo es medicina, educación, ingeniería, entre otras (Systems, 2015).*

○ **INTEGRACIÓN**

Enterprise Architect soporte diferentes *Sistemas Gestores de Base de Datos (SGBD, o DBMS por sus siglas en inglés)*, y entre los más destacados se encuentran:

- DB2
- InterBase
- Informix
- MySQL
- Oracle 9i/10g/11g
- PostgreSQL
- MS SQL Server

Enterprise Architect provee una serie de mecanismos para integrar los modelos generados con herramientas de terceros. Estos mecanismos son:

- MDG Link: Permite la interoperabilidad con Visual Studio y/o Eclipse.
- MDG Integration: Permite la edición de los modelos desde los IDEs Visual Studio y/o Eclipse.
- MDG Integration for Microsoft Office: Integración con la suite de Office (Word, Excel, PowerPoint, Visio).
- Capacidad de exportar e importar de diferentes formatos.
- Variedad de formatos XMI (acrónimo de XML Model Interchange).

○ *CICLO DE VIDA DE DESARROLLO DE SOFTWARE*

Dentro del proceso de desarrollo se contemplan diferentes fases que permiten definir cómo se va seguir el desarrollo de software. *Enterprise Architect permite modelar dicha información para facilitar la administración del proyecto, dentro del ciclo de vida de software, permitiendo un modelado del ciclo de vida completo del software (Systems, 2015).*

○ *VERSIÓN UML*

Enterprise Architect soporta el 100% de los modelos y diagramas de UML 2.5. Además de contar con soporte para la notación BPNM (acrónimo de Business Process Modeling Notation) y las especificaciones SysML (acrónimo de System Modeling Language).

- *DIAGRAMAS UML*

Soporte para 14 diagramas: diagrama de clases, componentes, composición, objetos, paquetes, despliegue, actividad, casos de uso, interacción, comunicación, secuencia, estados, profile, y timing.

- *LICENCIA*

Se estipulan los siguientes tipos de licencia:

- Ultimate Edition (Basic y Floating⁴)
- **Systems Engineering Edition (Basic y Floating)**
- **Business & Software Engineering Edition (Basic y Floating)**
- **Corporate Edition (Basic y Floating)**
- Professional Edition
- Desktop Edition
- Multi-user
- Trial y/o demo
- NFR (Not For Resale)
- Académica

- **MAGIC DRAW**

Es una herramienta case muy intuitiva y poderosa. Resaltando que permite el trabajo colaborativo entre desarrolladores a través de un servidor TeamWork.

- *FUNCIONALIDAD*

Magic Draw es una herramienta de modelado, que destaca las siguientes características:

1. Rápido aprendizaje a través de una interfaz intuitiva.
2. *La herramienta de creación de diagramas más rápida del mercado*(Magic, n.d.).
3. Generación de código a partir de los modelos.
4. Generación de diagramas enriquecidos en breves pasos.

⁴ La licencia básica se encuentra limitada a un solo usuario, a diferencia de las flotantes (Floating) en donde pueden realizar la gestión de claves de registro.

5. Ahorro de tiempo y dinero, a través de facilitar la colaboración de equipos.

▪ *INTEGRACIÓN*

A diferencia de una gran cantidad de herramientas de modelado, presenta integración con *Rational Rose* ofreciendo una solución a los usuarios de esta herramienta utilizando el RConverter de Magic Draw. Además de ofrecer integración con entornos *Java*, como Eclipse y Netbeans, para que se puedan sincronizar los modelos con la parte de la programación.

También soporta la generación de código en los siguientes lenguajes: Java, CORBA IDL, C# y C++, así como la generación de los scripts para la creación de la base de datos.

▪ *CICLO DE VIDA DEL SOFTWARE*

En el proceso de desarrollo se contemplan diferentes fases, como se ha mencionado anteriormente, las cuales permiten definir cómo se va seguir el desarrollo de software. Magic Draw se apega completamente al ciclo de vida del software, al proporcionar el 100% de los diagramas de UML 2.5.

▪ *VERSIÓN DE UML*

UML 2.5

▪ *LICENCIAS*

Magic Draw ofrece distintas licencias de software, para que el usuario determine cuál es la edición más acorde a las necesidades del proyecto:

1. Enterprise
2. Architect
3. Professional
4. Standar
5. Personal
6. Reader

También presenta una versión TRIAL para cada una de las ediciones, y se pueda valorar si el producto cumple con las expectativas.

- **VISUAL PARADIGM**

Es una solución integral que permite a un equipo de desarrollo gestionar la comunicación entre los directivos, los desarrolladores y el cliente. Visual Paradigm (VP) contempla que el jefe del proyecto pueda llevar el control de cambios y responder de manera rápida a ellos(Visual Paradigm, 2016).

- *FUNCIONALIDAD*

1. VP ofrece una guía durante el proceso de modelado, a través de instrucciones embebidas, referencias en los campos de entrada(input), y ejemplos.
2. Permite la gestión de proyectos a través de la implementación de estándares abiertos como TOGAF⁵ y/o PMBOOK.
3. Proporciona un entorno colaborativo para gestionar la asignación y la programación de tareas a través del gráfico PERT, generando el informe JIT.

- *INTEGRACIÓN*

1. Permite la generación de código a partir de los modelos UML, manteniendo la sincronización en paralelo.
 - C#, Java, DLL, .NET, VB.NET, PHP, Python, XSD, XML, ActionScript, Objective-C, Ada95, ODL, IDL, Delphi, PERL & Ruby
2. Integración con los siguientes IDE:
 - Eclipse
 - Netbeans
 - IntelliJ IDEA
 - Visual Studio
 - Andriod Studio

⁵ Es un framework de arquitectura para la gestión en la aceptación, producción, uso y mantenimiento de arquitecturas(Josey, Harrison, Homan, & Rouse, 2008)

- *CICLO DE VIDA DEL SOFTWARE*

VP permite el seguimiento del proceso ágil de desarrollo, a través del administrador de tareas que permite gestionar las tareas y los sprints de manera eficaz.

El proceso unificado se encuentra totalmente automatizado, permitiendo colocar las actividades del usuario, como tareas del usuario(Visual Paradigm, 2016).

- *VERSIÓN DE UML*

UML 2.x

- *LICENCIAS*

Visual Paradigm maneja dos tipos de licencias Perpetua y suscripción mensual(1,3,6,12), aplicables en las siguientes ediciones:

1. Enterprise
2. Professional
3. Standar
4. Modeler

También presenta una versión TRIAL por 30 días, y se pueda valorar si el producto cumple con las expectativas.

- **INNOVATOR MODELING PLATFORM**

Es considerado un entorno escalable que permite integrar desde el modelado de diagramas UML hasta el proceso de negocios dentro de una organización; diferenciándose de otras herramientas por su proceso de innovación.

- *FUNCIONALIDAD*

En una sola plataforma, se integran 4 tipos de herramientas, para atender a distintas necesidades:

1. Innovator para Business Analysts.
2. Innovator para Information Architects
3. Innovator para Software Architects(Object)
 - a. Soporte para el desarrollo ágil.
 - b. Fácil manejo para aplicaciones complejas

4. Innovator para Enterprise Architects

- *INTEGRACIÓN*

1. Generación de código JAVA, C#, CORBA IDL, Smalltalk, Visual Basic, Forté TOOL, C y COBOL.
2. Herramienta JAVA Harvester, para la creación de modelos a partir de código existentes JAVA.
3. Compatibilidad con IDE'S como Eclipse.
4. Archivos compatibles para la importación: XML, DDL, XMI.
5. Sistemas Gestores de base de datos compatibles: DB2, ORACLE, Informix, MS SQL Server, entre otras.

- *CICLO DE VIDA DEL SOFTWARE*

Debido a la integración de 4 herramientas en una sola plataforma, la integración de INNOVATOR Enterprise Modelign es completa.

- *VERSIÓN DE UML*

UML 2.5

- *LICENCIAS*

Las licencias dependen del tipo de ediciones que ofrece, pero las ediciones que cuentan con soporte de UML son:

1. INNOVATOR Business
2. INNOVATOR Object

También presenta una versión TRIAL por 60 días, y se pueda valorar si el producto cumple con las expectativas.

Existen más herramientas de acceso abierto, como es ARGO UML. Dicha herramienta *ofrece soporte para el análisis y diseño de sistemas de software*(Ramirez & Tolke, n.d.), lo que lo hace diferente es:

- Acceso abierto
- Soporte de estándares abiertos como: UML, XMI, SVG, OCL, entre otros.
- Multiplataforma

- Adaptable al modelo de cascada, al proceso iterativo de desarrollo de software, y al proceso unificado de desarrollo de software (RUP⁶).

⁶ *El Proceso Unificado de Desarrollo de Software (RUP) presenta un enfoque prescriptivo para estandarizar las mejores prácticas de la ingeniería de software(Torres, 2009).*

IV. CONCLUSIÓN

Dentro de la fase de diseño de la plataforma “MeAlcanza.com”, se contemplan distintos tipos de etapas, para las cuales es importante modelar distintos tipos de artefactos que son necesarios para concluir en los tiempos establecidos el proyecto. Dentro de la fase de diseño se realizaran modelos que permitan identificar la interacción dentro de la plataforma, y el objetivo del documento, es realizar una comparación entre las distintas herramientas CASE aplicables al proyecto.

Existen herramientas libres o de acceso abierto para el desarrollo de modelos UML, al igual existen herramientas de pago, que brindan servicios más especializados con respecto al soporte técnico. Otras de las herramientas mencionadas, cuentan con *trials* de 30 o 60 días para evaluar si es la adecuada para el tipo de proyecto, pero en la tabla 1 se muestra una comparativa más a detalle de las herramientas:

| Plataforma/ Herramienta | Diagramas | | | | | | | | | | | | | | | Licencia | |
|----------------------------|-------------|--------|-------------|-----------|---------|----------|---------|------------|-----------|--------------|-------------|--------------|-----------|--------|---------|----------|---------|
| | Versión UML | Clases | Componentes | Composite | Objetos | Paquetes | Profile | Despliegue | Actividad | Casos de uso | Interacción | Comunicación | Secuencia | Timing | Estados | Trial | Abierto |
| ARGO UML | 1.5 | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ | ✓ | | ✓ | ✓ | | ✓ | | ✓ |
| Enterprise Architect | 2.4 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Innovator | 2.3 | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | ✓ | | ✓ | ✓ | | |
| Magic Draw | 2.4 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | |
| Visual Paradigm | 2.3 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Concept Draw | 2.2 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | | |

Tabla 1. Comparación de las herramientas case(Company, Project, & Vision, 2013) .

Tomando en cuenta la tabla 1, se observa que la herramienta Enterprise Architect es la más completa, además de brindar características adicionales de soporte para el gestor del proyecto, pero implica un costo para adquirir la licencia. Dentro la plataforma MeAlcanza, se aplicará una metodología de desarrollo de software, la cual es fundamental para determinar la herramienta CASE más idónea para el proyecto; lo anterior debido a que algunos diagramas no serán necesarios debido a la naturaleza de la plataforma, y en este sentido, herramientas como Visual Paradigm o ArgoUML pueden cubrir las necesidades del proyecto.

V. BIBLIOGRAFÍA

- Alarcón, R. (s.f.). *Diseño Orientado a Objetos con UML*. Madrid, España: Grupo Eidos.
- Jacobson Ivar, B. G. (2007). *El proceso unificado de desarrollo de software*. Madrid, España: PEARSON.
- Project Management Institute. (2012). *Guía PMBoK. Fundamentos para la dirección de proyectos* (5th edition ed.). (P. M. Institute, Ed.) USA.
- Fudenberg, D., & Tirole, J. (1983). Learning by Doing and Market Performance. *Bell Journal of Economics*.
- Larman, C. (2004). Applying UML and Patterns. *Development* (Vol. 17, p. 960).
<https://doi.org/10.1016/j.nec.2006.05.008>
- AIGroup. (2017). Metodologías. Retrieved from <http://aigroup.com.co/site/>
- Systems, S. (2015). Enterprise Architect 12 Reviewer ' s Guide.
- Josey, A., Harrison, R., Homan, P., & Rouse, M. F. (2008). *TOGAF® VERSION 9.1 – A POCKET GUIDE*. (The Open Group, Ed.) (First edit). United Kingdom: Van Haren Publishing, Zaltbommel.
- Visual Paradigm, I. (2016). Visual Paradigm v 14.1. Visual Paradigm International Ltd.
- Ramirez, A., & Tolke, L. (n.d.). ArgoUML User Manual A tutorial and reference description.
- Torres, F. (2009). Integración del PMBOK al RUP para proyectos de Desarrollo de Software. ... Iberoamericana En Sistemas, Cibernética E Informática ... Retrieved from <http://www.iiis.org/CDs2008/CD2009CSC/CISCI2009/PapersPdf/C690MI.Pdf>
- Company, T. A., Project, O. S., & Vision, C. (2013). osse UML Tool list, 2, 1–2.