CONSTRUCCIÓN DE UNA APLICACIÓN WEB PARA LA ELABORACIÓN DE DIAGRAMAS DE CLASE, ORIENTADA BAJO EL MODELO DE SERVICIOS DE INTERNET CLOUD COMPUTING.

GABRIEL LEONARDO DÍAZ CÁRDENAS

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERIAS  
PLAN DE ESTUDIOS DE INGENIERIA DE SISTEMAS  
SAN JOSÉ DE CÚCUTA  
2012

CONSTRUCCIÓN DE UNA APLICACIÓN WEB PARA LA ELABORACIÓN DE DIAGRAMAS DE CLASE, ORIENTADA BAJO EL MODELO DE SERVICIOS DE INTERNET CLOUD COMPUTING.

GABRIEL LEONARDO DÍAZ CÁRDENAS

Trabajo de grado presentado como requisito para optar por el título de Ingeniero de Sistemas

Director  
MARCO ANTONIO ADARME JAIMES  
Ingeniero de Sistemas  
Magister Scientie en Computación

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERIAS  
PLAN DE ESTUDIOS DE INGENIERIA DE SISTEMAS  
SAN JOSÉ DE CÚCUTA  
2012

TABLA DE CONTENIDO

[INTRODUCCIÓN 7](#_Toc341197280)

[1. DESCRIPCIÓN DEL PROBLEMA 8](#_Toc341197281)

[1.1 PLANTEAMIENTO DEL PROBLEMA 8](#_Toc341197282)

[1.2 JUSTIFICACIÓN 9](#_Toc341197283)

[1.3 OBJETIVOS 11](#_Toc341197284)

[1.3.1 Objetivo General 11](#_Toc341197285)

[1.3.2 Objetivos Específicos 11](#_Toc341197286)

[1.4 ALCANCES Y DELIMITACIONES 12](#_Toc341197287)

[1.4.1 Alcance 12](#_Toc341197288)

[1.4.2 Delimitaciones 12](#_Toc341197289)

[1.5 MARCO TEORICO 13](#_Toc341197290)

[1.6 MARCO LEGAL 13](#_Toc341197291)

[1.6.1 Acuerdos de uso 13](#_Toc341197292)

[1.6.2 Contrato de licencia de código binario, SUN MICROSYSTEMS 14](#_Toc341197293)

[1.6.3 GNU General Public Licence 15](#_Toc341197294)

[1.6.4 PROPIEDAD INTELECTUAL 15](#_Toc341197295)

[2. DISEÑO METODOLÓGICO 16](#_Toc341197296)

[2.1 RECOPILACION Y ANALISIS DE HERRAMIENTAS 16](#_Toc341197297)

[2.1.1 Enterprise Architect 16](#_Toc341197298)

[2.1.2 StarUML 19](#_Toc341197299)

[2.1.3 Draw.IO 20](#_Toc341197300)

[2.1.4 Conclusiones del análisis 20](#_Toc341197301)

[2.2 CARACTERISTICAS PROPIAS DE LA APLICACIÓN EN CONSTRUCCIÓN 23](#_Toc341197302)

[2.3 METODOLOGIA DE DESARROLLO 23](#_Toc341197303)

[2.4 ANALISIS DE LA TECNOLOGIA 23](#_Toc341197304)

[3. INGENIERIA DE REQUERIMIENTOS 25](#_Toc341197305)

[3.1 REQUERIMIENTOS FUNCIONALES 25](#_Toc341197306)

[3.2 REQUERIMIENTOS NO FUNCIONALES 25](#_Toc341197307)

[3.3 MODELO DE CASOS DE USO 25](#_Toc341197308)

[3.4 MODELO DEL DOMINIO 25](#_Toc341197309)

[4. ANÁLISIS 26](#_Toc341197310)

[5. DISEÑO 27](#_Toc341197311)

[6. IMPLEMENTACIÓN 28](#_Toc341197312)

[7. PRUEBAS 29](#_Toc341197313)

[8. DESPLIEGUE 30](#_Toc341197314)

[9. ASEGURAMIENTO DE LA CALIDAD 31](#_Toc341197315)

[RESULTADOS 32](#_Toc341197316)

[CONCLUSIONES 33](#_Toc341197317)

[REFERENCIAS BIBLIOGRAFICAS 34](#_Toc341197318)

[ANEXOS 35](#_Toc341197319)

# INTRODUCCIÓN

**TODO**

# DESCRIPCIÓN DEL PROBLEMA

## PLANTEAMIENTO DEL PROBLEMA

En estudios realizados por el grupo de investigación y desarrollo de software (GIDIS) de la Universidad Francisco de Paula Santander, después convertido en un proyecto de grado[[1]](#footnote-1), se ha determinado que unos de los principales factores involucrados en los altos índices de mortalidad y deserción académica en la carrera de Ingeniería de Sistemas de las universidades de Cúcuta, son la falta de dedicación al estudio por parte de los estudiantes, poca comprensión de temas como abstracción y modelamiento de sistemas, y falta de ambientes prácticos donde los estudiantes tuvieran la posibilidad de llevar a cabo actividades para afianzar los conceptos aprendidos en clase. Teniendo esto en cuenta se hace evidente que se necesitan mecanismos para que los estudiantes ejerciten sus habilidades y de esta manera se disminuya esta problemática que ha afectado considerablemente al programa académico.

Dentro del programa de Ingeniería de Sistemas de la UFPS se tiene un clara tendencia e inclinación hacia el desarrollo de software, es claro para muchos estudiantes que este es uno de los puntos fuertes de la carrera y es por ello que muchos continúan su vida profesional por ese camino; por tal motivo desde los primeros semestres se trata de despertar estas habilidades en ellos; considerando que el modelamiento y abstracción de sistemas son temas de vital importancia, se hace necesario disponer de herramientas automáticas (software) que permitan adquirir competencias en estas áreas y reforzar conceptos aprendidos en clase, pero que solo se consiguen mediante la práctica; además que permitan al estudiante forjar unos cimientos importantes para su vida profesional en aspectos tales como UML, Diagramas de clase, CASE Tools[[2]](#footnote-2) y razonamiento abstracto.

Aun así, se es consciente que actualmente ya existen herramientas muy completas que permiten realizar estas labores y que el programa académico ha venido utilizando desde hace algún tiempo (por ejemplo Enterprise Architect[[3]](#footnote-3), Netbeans UML, StarUML, ArgoUML), además que la universidad dispone de licencias de uso para algunas de ellas (por ejemplo, Enterprise Architect que es software privado), y otras son software libre de modo que pueden ser utilizados de cualquier forma; aun así, la mayoría de ellas, por no decir todas, son software que se instala de manera nativa, de modo que carecen de portabilidad y facilidad de acceso, aspectos que son muy importantes y atractivos hoy en día en los software de aplicación. Otras requieren ciertas características mínimas de hardware que aunque no es muy común, no siempre están disponibles y son suficientes para soportar que estas aplicaciones se ejecuten de manera adecuada.

Por otro lado y no menos importante se ha notado mediante observaciones indirectas y manifestaciones explicitas de algunos estudiantes, sobre todo en primeros semestres, que muchos de ellos aún tienen dificultades en la configuración de estas aplicaciones, otros expresan no tener computadoras propias o simplemente que los recursos hardware son insuficientes para el funcionamiento adecuado de las aplicaciones, aspectos que evidencian la necesidad planteada anteriormente e impulsan a diseñar modelos de servicios accesibles a través internet, los cuales podrían brindar ciertas flexibilidades y beneficios extras para los usuarios que no tienen las aplicaciones nativas tales como facilidad de acceso, portabilidad, autoconfiguración, capacidad de compartir recursos e información.

## JUSTIFICACIÓN

El programa académico de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander, tiene como objeto de estudio la “Construcción y Gestión de Sistemas de Información”[[4]](#footnote-4), por esta razón dentro del pensum existen muchos cursos que se enfocan en programación e ingeniería de software; esto significa que hay una clara tendencia a tratar temas como el modelado de aplicaciones, procesos de desarrollo y otros conceptos que hacen parte de esta área de la ingeniería; de esta manera consideramos que se necesita que los estudiantes pongan en práctica técnicas y métodos para desarrollar y mantener software de calidad, teniendo siempre en mente que la tecnología existe para ser usada como herramienta facilitadora del trabajo.

Lo anterior sugiere la necesidad de disponer de una herramienta que permita realizar prácticas de laboratorio, más específicamente prácticas de abstracción de diagramas de clase, de modo que se puedan aplicar todos los conceptos aprendidos, bajo un ambiente real. Es por ello que nos hemos enfocado específicamente en este aspecto, de tal manera que se pueda obtener una herramienta adicional que sirva de apoyo al proceso de aprendizaje y permita adquirir con seguridad conceptos vitales y que serán de gran ayuda en la vida profesional de los estudiantes.

De igual manera sería ideal que los docentes puedan disponer de un mecanismo de control y supervisión de las actividades realizadas por sus estudiantes, permitiendo de manera fácil y cómoda la revisión de los trabajos de clase o evaluaciones parciales; por tal motivo se pretende construir una aplicación orientada bajo el modelo de servicios “Cloud Computing” [[5]](#footnote-5)(*en español, Computación en la nube*), e incorporar comportamientos como los desarrollados por “Google Docs”[[6]](#footnote-6), en donde los usuarios pueden compartir información con otros de manera fácil y cómoda a través de un click. Aplicando este concepto a la aplicación en construcción, se espera que los estudiantes puedan compartir con su docente los diagramas realizados de modo que el docente pueda consultarlos y corregirlos a través de la aplicación. Además es significativo incentivar el autoaprendizaje, permitiendo a los estudiantes el intercambio de conocimientos e información, de manera que puedan reforzar los conceptos, obviamente mediante algunos mecanismos de control que garanticen un proceso limpio y transparente.

Adicionalmente una motivación importante para el desarrollo de este proyecto es mejorar la infraestructura de servicios y de tecnologías de información disponibles para los estudiantes de la UFPS, mediante la construcción de herramientas innovadoras y que generen conocimientos sobre el desarrollo de aplicaciones web.

Por último, otro aspecto importante es la necesidad de impulsar e innovar en el desarrollo de tecnologías, el compromiso que tiene el plan de estudios de Ingeniería de Sistemas con la universidad y con la comunidad en general, para desarrollar alternativas de solución a problemáticas mediante el uso de la tecnología y generar conocimiento en base a investigación e innovación nos ha llevado a esta iniciativa, esperando propulsar el desarrollo de nuevos productos y herramientas que presten servicios dentro y fuera de la universidad, y que sean realizadas por los mismos estudiantes.

## OBJETIVOS

### Objetivo General

Implementar una aplicación web basada en el paradigma orientado a servicios de Cloud Computing, que permita el diseño y construcción de diagramas de clase UML y que sirva como medio de apoyo para proyectos de software realizados por los estudiantes de Ingeniería de Sistemas de la Universidad Francisco de Paula Santander.

### Objetivos Específicos

* Realizar un estudio de las principales características que poseen las herramientas CASE utilizadas por los estudiantes del programa de Ingeniería de Sistemas, para determinar los servicios que prestan en cuanto al diseño de diagramas de clase.
* Definir las funcionalidades que el software debe poseer, determinando las características específicas, los servicios que debe prestar a los usuarios y los procedimientos que debe llevar a cabo.
* Realizar el modelado del sistema, d2qqqeterminando una solución óptima a los procedimientos definidos, de modo que no se vea afectado el rendimiento y la estabilidad del software.
* Realizar pruebas de funcionamiento que garanticen el comportamiento adecuado del software y que permitan verificar que se cumplan los requerimientos definidos.
* Documentar las pruebas realizadas junto con los procedimientos llevados a cabo y los resultados obtenidos.
* Desplegar la aplicación en un servidor de modo que pueda ser accedido por estudiantes y docentes.

## ALCANCES Y DELIMITACIONES

### Alcance

La ejecución de este proyecto pretende alcanzar lo siguiente:

* Obtener un producto software que permita a los usuarios diseñar de manera visual diagramas de clase UML.
* Disponer de una plataforma online que pueda ser utilizada desde cualquier lugar y a cualquier hora a través de internet.
* Disponer de una herramienta que permita a los docentes asesorar de manera continua y cómoda a sus estudiantes en los proyectos desarrollo de software.
* Un software para realizar modelado de objetos que sirva como soporte de los proyectos de software y permita generar código fuente de manera automática, agilizando el proceso de desarrollo.
* Un documento mediante el cual se pueda conocer el funcionamiento básico del software y los servicios que este presta, que sirva como guía a los usuarios sobre la manera en la cual deben utilizar la herramienta (Manual de usuario).

### Delimitaciones

Con el propósito de determinar la viabilidad y posibilidad de desarrollo del proyecto se han establecido las delimitaciones mencionadas a continuación:

* El software será totalmente independiente del Sistema de Información Académico de la Universidad (SIA), de modo que no se validara que los usuarios registrados estén en la base de datos del mismo. Sin embargo se tendrá en cuenta la información básica de los usuarios como: el código dentro de la Universidad, la facultad y el programa académico al cual pertenecen, entre otros.
* Solo se podrá realizar diagramas de clase UML, la aplicación no abarca el diseño de otro tipo de diagrama.
* El software no dispondrá de un administrador, de modo que no sea necesario disponer de usuarios que realicen configuraciones específicas y únicas en el software.
* El código fuente generado a partir de los diagramas de clase será únicamente en lenguaje Java.
* Considerando que existen varios cursos de programación e Ingeniería de Software dentro del pensum académico actual del programa de Ingeniería de Sistemas, el proyecto no estará dirigido específicamente a un curso en particular, sino que estará disponible para cualquier estudiante en cualquier curso; sin embargo, se tomara una población base de uno o dos cursos para realizar el levantamiento de información. En el diseño metodológico se especificará con más detalle este tema.
* El software no permitirá editar el código fuente generado directamente, para esta labor se hace necesario que el usuario disponga de un editor.

# MARCO REFERENCIAL

TOMAR EL DEL ANTEPROYECTO, REESTRUCTURAR, INCLUIR NUEVOS CONCEPTOS, (**REFLEXION COMPUTACIONAL, entre otros.**)

## 2.1 MARCO TEORICO

### Acuerdos de uso

El software, resultado de la ejecución del proyecto, se regirá bajo los marcos legales fijados por la licencia pública Creative Common (LPCC) que dicta las siguientes condiciones de uso, encontradas más detalladamente en el documento Atribución No Comercial Compartir Igual 2.5 (Colombia)[[7]](#footnote-7):

* Se posee la libertad de compartir la obra, entendiendo compartir como la capacidad de copiar, distribuir, ejecutar y comunicar públicamente la obra.
* Además puede llevar a cabo obras derivadas de la original.
* Es obligatorio reconocer los créditos de la obra de manera especificada por el autor o el licenciante, pero no de manera que sugiera que tiene el apoyo de este último, o que apoya el uso que hacen a su obra.
* No puede ser utilizada para uso comercial.
* Si se altera o transforma, o genera un derivado a partir de esta obra, solo puede ser distribuida bajo una licencia idéntica a la presente.

### Contrato de licencia de código binario, SUN MICROSYSTEMS

Licencia de uso de software concedidos por Sun Microsystem, para la utilización de Java SE, siempre y cuando sean aceptados los términos de licencia, expuestos en el documento SUN MICROSYSTEMS, INC. CONTRATO DE LICENCIA DE CÓDIGO BINARIO[[8]](#footnote-8).

“SUN MICROSYSTEMS, INC. (EN ADELANTE DENOMINADO “SUN”) LE CONCEDE LA LICENCIA DEL SOFTWARE DEFINIDO A CONTINUACIÓN ÚNICAMENTE CON LA CONDICIÓN DE QUE USTED ACEPTE TODOS LOS TÉRMINOS ESTIPULADOS EN EL PRESENTE CONTRATO DE LICENCIA DE CÓDIGO BINARIO Y TÉRMINOS DE LICENCIA ADICIONALES (EN CONJUNTO DENOMINADOS “CONTRATO”). POR FAVOR, LEA EL CONTRATO DETENIDAMENTE. EL USO DEL SOFTWARE SIGNIFICA QUE HA LEÍDO LAS CONDICIONES Y QUE LAS ACEPTA. SI ACEPTA ESTAS CONDICIONES EN REPRESENTACIÓN DE UNA COMPAÑÍA U OTRA ENTIDAD LEGAL, SIGNIFICA QUE ESTÁ EN POSESIÓN DE LA AUTORIDAD QUE LE PERMITE VINCULAR LA ENTIDAD LEGAL A ESTAS CONDICIONES. SI NO ESTÁ EN POSESIÓN DE DICHA AUTORIDAD O SI NO DESEA QUEDAR VINCULADO A ESTAS CONDICIONES, NO DEBE UTILIZAR EL SOFTWARE EN ESTA INSTALACIÓN NI EN NINGÚN OTRO SOPORTE EN EL QUE SE UBIQUE EL SOFTWARE”.

### GNU General Public Licence

La Licencia Pública General de GNU (GNU GPL, por sus siglas en inglés) es una licencia libre y gratuita con derecho de copia para software y otros tipos de obras.

Las licencias para la mayoría del software y otras obras de índole práctica están diseñadas para privarle de la libertad para distribuir y modificar las obras. Por el contrario, la Licencia Pública General de GNU garantiza la libre distribución y modificación de todas las versiones de un programa, a fin de asegurarle dicha libertad a todos los usuarios. En la Fundación para el Software Libre utilizamos la Licencia Pública General de GNU para la mayoría de nuestro software; también se aplica a cualquier otra obra publicada de esta manera por sus autores. Usted también puede aplicarla a sus programas.

Cuando hablamos de software libre, nos referimos a la libertad, no al precio. Nuestras Licencias Públicas Generales están diseñadas para garantizarle a usted la libertad de distribuir copias de software libre (y cobrar por ellas, si así lo desea), obtener el código fuente, o tener la posibilidad de obtenerlo, modificar el software o utilizar partes del mismo en nuevos programas libres, y saber que puede hacer estas cosas[[9]](#footnote-9).

### PROPIEDAD INTELECTUAL

De acuerdo con lo estimulado en el artículo 156 del acuerdo 065 del 26 de 1996, correspondiente al estatuto estudiante de la Universidad Francisco de Paula Santander, el cual dicta de la siguiente manera: “Los trabajos de grado son propiedad intelectual de la Universidad y su uso estará sujeto a las normas que para tal fin estén vigentes”[[10]](#footnote-10).

# DISEÑO METODOLÓGICO

## RECOPILACION Y ANALISIS DE HERRAMIENTAS

Con el objetivo de obtener información importante para la realización del proyecto y con el ánimo de iniciar el proceso de levantamiento de requerimientos se ha realizado un pequeño análisis de algunas de las herramientas utilizadas dentro del plan de estudios de Ingeniería de Sistemas para realizar modelado UML, de modo que se pueda establecer cuáles son las principales características y servicios que estas ofrecen a los usuarios en cuanto al diseño de *Diagramas de Clase* y que puedan ser incluidos en las funcionalidades del software en construcción.

De modo aleatorio se han seleccionado dos de las herramientas más conocidas y que a manera de observación se ha determinado que son ampliamente utilizadas por estudiantes y docentes del plan de estudios, adicionalmente se ha seleccionado una herramienta similar a la que está en construcción, de modo que podamos conocer qué características ofrece a los usuarios y cuales son imprescindibles en una herramienta de modelamiento UML. Las herramientas seleccionadas son: *Enterprise Architect* que es un software comercial conocido a nivel mundial por su gran calidad, *StarUML* que a su vez es open source pero ampliamente utilizado, y por ultimo una herramienta de modelado online llamada *Draw.IO* la cual se asemeja mucho a lo que estamos tratando de lograr.

Al momento de realizar este análisis se tomó la última versión de las aplicaciones que se instalan de manera nativa, Enterprise Architect en su versión 9.0.9 con un periodo de prueba de 30 dias y StarUML en su versión 5.0. En el caso de Draw.IO la librería mxGraph usada para la construcción de esta aplicación se encuentra en la versión 1.10.4.0.

A continuación se describirá cada uno de ellos, resaltando en detalle cada una de sus características:

### Enterprise Architect

Enterprise Architect de Sparx Systems es una herramienta CASE (Computer Aided Software Engineering) para el diseño y construcción de sistemas de software, para el modelado de procesos de negocios, y para objetivos de modelado más generalizados.

EA es una herramienta de análisis y diseño UML comprensivo, que cubre el desarrollo de software desde la obtención de los requisitos, diseño del modelo, pruebas, cambio de control y mantenimiento para la implementación, con completa trazabilidad. EA combina el poder de la última especificación UML 2.4.1 con alta performance, una interfaz intuitiva, para brindar un modelado avanzado a todo el equipo de desarrollo. Es una herramienta visual con multi-usuario con un gran establecimiento de características ayudando a los analistas, testers, administradores de proyectos, personal del control de calidad y desarrolladores alrededor del mundo a construir y documentar, software sostenible.

**Historia**

Sparx Systems es una empresa australiana dedicada a la creación de herramientas para el modelado UML, fundada en 1996 por Geoffrey Sparx y situada en Creswick una pequeña población de Victoria, Australia. La primera versión de Enterprise Architect fue liberada en el año 2000 y fue originalmente diseñada como una herramienta de modelado UML para la versión 1.1 del estándar, de allí en adelante el producto ha evolucionado para soportar otras especificaciones OMG[[11]](#footnote-11) UML como 1.3, 2.0, 2.1, 2.3 y 2.4.1.

Durante el tiempo de vida de la aplicación se han liberado alrededor de 20 versiones, desde la versión 1.0 hasta la más reciente 9.3 liberada en Agosto de 2012, cada una con mejoras y características importantes según los cambios que ha sufrido el estándar UML y la adopción de algunas tecnologías

Actualmente la cantidad de estándares para el diseño y modelado de software y sistemas de negocios que son soportados por Enterprise Architect es considerable, teniendo en cuenta la adopción de UML 2.4.1 como principal herramienta para modelar aplicaciones e incluyendo algunos otros BPMN, WSDL, BPEL y XSD, ha permitido que la herramienta se imponga como uno de los líderes del mercado y que su aceptación por parte de la industria del software haya sido grandiosa.

Enterprise Architect se encuentra disponible en diferentes ediciones, según las necesidades del usuario o de las organizaciones que usan la aplicación, de modo que se pueden adquirir diferentes tipos de licencias para diferentes usos. Actualmente existen dos ediciones y dentro de las cuales se pueden comprar varias versiones del producto:

* Ediciones base
* Corporativa
* Profesional
* Escritorio
* Ediciones Suite
* Ultimate
* Ingeniería de Sistemas
* Ingeniería de Software y negocios

Los precios del producto dependen de la versión adquirida y de la cantidad de licencias que se obtengan de cada uno, siendo la versión de Escritorio la más simple hasta la versión Ultimate que a su vez posee la capacidad de trabajo en múltiples dominios de manera remota y un soporte completo de ingeniería de software y de negocios.

**Características Generales**

Enterprise Architect posee un gran abanico de características, todas encaminadas al desarrollo y mantenimiento de proyectos de software, aunque también brinda soporte para otro tipo de proyectos.

La información aquí consignada fue tomada de la documentación explicita de Enterprise Architect, disponible en la página web, además se usó parte de la información del “Contenido de Ayuda” que viene incluido en la versión del software.

A continuación mencionaremos varias de las características generales de EA:

| **Característica** | **Descripción** |
| --- | --- |
| Plataforma | Diseñado para ser usado bajo sistemas operativos Windows, sin embargo puede ser instalado y utilizado mediante emuladores como CrossOver en sistemas Linux o MAC. CrossOver es una herramienta propietaria basada en una tecnología open source llamada Wine, el cual es una API Win32 re-desarrollada para sistemas operativos basados en Unix como Mac OS X o Linux. |
| Categoría de Software | Software propietario distribuido bajo licencias comerciales, el costo de estas depende de la versión del software y la cantidad de licencias adquiridas. |
| Versión de UML | UML 2.4.1 Liberada en agosto de 2011. |
| Diagramas UML | *Estructurales*: Clases, Objetos, Paquetes, Estructura Compuesta, Componentes, de Perfiles y Despliegue.  *Comportamentales*: Actividad, Casos de Uso, de Estados, de Tiempo, Secuencia, Comunicación e Interacción.  *Adicionales no UML*: Análisis, Requerimientos, Bases de Datos, Mantenimiento, Interfaces de Usuario, Interacción y modelos del negocio. |
| Tecnologías MDG[[12]](#footnote-12) | Soporta tecnologías como ICONIX, ArcGIS, BPMN, SMOF, SoaML, SysML. |
| Fases del desarrollo | Modelamiento del Negocio, Ingeniería de Requerimientos, Análisis, Diseño, Implementación, Pruebas, Mantenimiento, Gestión de la Calidad. |
| Estándar XMI | Soporte completo del estándar en su versión 2.1, permite exportar e importar proyectos en archivos .XMI de modo que puedan ser llevados a otras herramientas de modelado. |
| Proyectos y equipos de trabajo | Permite generar repositorios de servidor DBMS[[13]](#footnote-13), de modo que se pueda trabajar de manera concurrente en proyectos de gran envergadura y con gran cantidad de usuarios.  Adicionalmente permite integración con repositorios de control de versiones como Subversion, CVS y SCC. |
| Importación / Exportación | *Exportación*: Existen varias formas de exportar proyectos, diagramas o cualquier otro tipo de artefacto creado dentro del software, archivos XML, XMI, Código fuente, imágenes planas de los diagramas en formato PNG, JPG, GIF, entre otros.  *Importación*: Desde XML, XMI, archivos CSV[[14]](#footnote-14), Desde código fuente en varios lenguajes, desde archivos binarios como JARs o ensamblados .NET. |
| Ingeniería Directa |  |
| Ingeniería Inversa |  |
| Documentación generada | Soporte completo de WYSIWYG[[15]](#footnote-15) manejador de plantillas para generar archivos RTF. |
| Soporte de Versiones |  |
| Seguridad |  |

Tabla 1. Formato características generales – Enterprise Architect

**Características específicas para diagramas de clase**

En cuanto al diseño, manejo y gestión de diagramas de clase Enterprise Architect posee varias características importantes que lo hacen muy llamativo a los usuarios, brindando facilidad, agilidad y sobre todo calidad. A continuación se mencionan varias de ellas.

|  |  |
| --- | --- |
| **Característica** | **Descripción** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Tabla 2. Formato de características específicas para diagramas de clase - Enterprise Architect

### StarUML

StarUML es un proyecto de código abierto para desarrollar de manera rápida, flexible, extensible y de libre acceso una plataforma UML/MDA[[16]](#footnote-16) bajo arquitectura Windows 32. El objetivo del proyecto StarUML es construir una herramienta para el modelado de software que sea un reemplazo irresistible para otras herramientas UML comerciales como Rational Rose, Together, entre otros.

StarUML está basado en una arquitectura de Plug-in de tal manera que el software puede soportar múltiples funcionalidades configurables dependiendo de las exigencias del usuario, esto hace que el software sea extensible y adaptable para diferentes tipos y categorías de usuarios. La usabilidad es uno de los temas más importantes en el desarrollo de software, StarUML esta implementado para proveer muchas características amigables al usuario final, tal como asistentes, manipulación a través del teclado, diferentes vistas del diagrama, etc. StarUML es escrito en su mayoría en lenguaje Delphi. Sin embargo, cualquier lenguaje de programación puede ser usado para desarrollar StarUML, por ejemplo Java, Visual Basic, JScript, C#, VB .NET, entre otros).

### Draw.IO

Draw.io es una aplicación de diagramado online desarrollada usando la librería mxGraph creada por la compañía JGraph Ltd. Esta empresa fue fundada en el año 2000 y dedicada únicamente a la comercialización y soporte de componentes gráficos de visualización desde el 2004. Actualmente se ha impuesto en el mercado con su producto líder *mxGraph*, el cual comprende una librería javascript que permite dibujar componentes para diagramas en cualquier navegador web usando HTML, CSS y Javascript.

Draw.io es el sitio de demostración para la tecnología desarrollada por la empresa JGraph y su producto mxGraph, el cual permite dibujar diagramas creativos en el navegador, compartir y editar con otros usuarios en tiempo real.

### Conclusiones del análisis

Una vez realizado el análisis de las herramientas y determinadas las principales características que estas ofrecen para la gestión de proyectos de software, podemos generar unas conclusiones en base a estas y especificar cuáles características deben ser implementadas de manera primordial e imprescindible.

Con el objetivo de esclarecer cuales de estos “features” se deben incluir en el software que estamos construyendo se realizó una tabla comparativa, mencionando cada una de las características encontradas y determinando cuales de ellas son susceptibles de implementar y cuales representan el núcleo de una aplicación que realice modelado UML.

A continuación se muestra una tabla comparativa de las herramientas, mencionando algunas características y resaltando cuales de las aplicaciones tienen soporte para estas y especificando si van a ser implementadas o no:

| **#** | **Característica para diagramas de clase** | **Enterprise Architect** | **StarUML** | **Draw.IO** | **UFPS Modeler** |
| --- | --- | --- | --- | --- | --- |
|  | Elementos Soportados: Clases, Interfaces, Enumeraciones, Paquetes, Objetos, Excepciones |  |  |  |  |
|  | Elementos adicionales: Notas, Links, Imágenes, Cuadros de texto, Recuadros, Círculos |  |  |  |  |
|  | Relaciones: Asociación, Agregación, Composición, Herencia, Clase asociación, Realización, Asociación dirigida, Dependencia |  |  |  |  |
|  | Multiplicidad |  |  |  |  |
|  | Navegabilidad |  |  |  |  |
|  | Visibilidad |  |  |  |  |
|  | Atributos y Métodos |  |  |  |  |
|  |  |  |  |  |  |
|  | Generación de código fuente |  |  |  |  |
|  | Plantillas para generar de código fuente en varios lenguajes |  |  |  |  |
|  | Generación de diagramas a partir de código fuente |  |  |  |  |
|  | Sincronización entre diagramas y código |  |  |  |  |
|  | Editar código fuente generado |  |  |  |  |
|  | Exportar imágenes planas de diagramas en diferentes formatos |  |  |  |  |
|  | Soporte del estándar XMI |  |  |  |  |
|  | Personalizar tipos de datos básicos |  |  |  |  |
|  |  |  |  |  |  |
|  | Área de trabajo grafico |  |  |  |  |
|  | Soporte Drag & Drop para crear o mover elementos |  |  |  |  |
|  | Selección múltiple de elementos y conectores (Herramienta Lasso) |  |  |  |  |
|  | Caja de herramientas con los elementos |  |  |  |  |
|  | Panel Árbol del proyecto |  |  |  |  |
|  | Panel Propiedades de Elementos |  |  |  |  |
|  | Panel de Vista Panorámica |  |  |  |  |
|  | Herramienta Zoom |  |  |  |  |
|  | Distribución automática de elementos (Auto Layout) |  |  |  |  |
|  | Redimensión de elementos |  |  |  |  |
|  | Trabajar con múltiples diagramas en paralelo |  |  |  |  |
|  | Autoguardado |  |  |  |  |
|  | Paneles ajustables |  |  |  |  |
|  | Personalizar distribución de paneles |  |  |  |  |
|  |  |  |  |  |  |
|  | Porta papeles |  |  |  |  |
|  | Deshacer, Rehacer (Undo-Redo) |  |  |  |  |
|  | Copiar y Pegar |  |  |  |  |
|  | Cuadricula – Área de trabajo |  |  |  |  |
|  | Regla – Área de Trabajo |  |  |  |  |
|  | Formatos de letra, color y tamaño |  |  |  |  |
|  | Imprimir diagramas en papel (usar la impresora) |  |  |  |  |
|  |  |  |  |  |  |

Tabla 3. Formato de características para diagramas de clase por cada herramienta

## CARACTERISTICAS PROPIAS DE LA APLICACIÓN EN CONSTRUCCIÓN

### Objetivos de la aplicación

* Permitir la creación de proyectos de modelado UML, enfocándose específicamente en el diseño de diagramas de clase.
* Construir de manera gráfica diagramas de clase UML representando todos los elementos disponibles para este tipo de diagrama de manera intuitiva para el usuario, así como las posibles relaciones que se puedan presentar entre estos.
* Generar de manera correcta el código fuente correspondiente a los diagramas construidos de acuerdo a los conceptos y principios inherentes a este tipo de diagramas.
* Gestionar los usuarios que acceden a la aplicación, permitiendo ingresar nuevos, modificar información de los usuarios existentes y gestionar permisos para acceder a unas u otras características del software en caso de ser necesarios.
* Compartir recursos (proyectos, diagramas) entre diferentes usuarios de la aplicación de modo que estos tengan acceso a modificar dichos recursos, definiendo algunos privilegios o permisos según el usuario quien comparte su información defina.
* Permitir a los usuarios valorar los diagramas que les han sido compartidos por otros usuarios, definiendo un mecanismo de valoración consistente y viable para los usuarios, y que no contraste con el proceso de evaluación universitario.
* Ofrecer a los usuarios diferentes mecanismos de importación/Exportación de proyectos o diagramas creados dentro de la aplicación.

### Servicios ofrecidos

**TODO**

* Mencionar las características tomadas de las herramientas anteriormente utilizadas, y que van a ser implementadas.
* Mencionar las características propias y únicas que se pretenden ofrecer.
* Descripción de los usuarios que interactúan con la aplicación (ROLES)
* Objetivos de la aplicación (que se va a desarrollar?)
* Como se va a administrar la aplicación?
* Qué servicios se van a ofrecer?

## METODOLOGIA DE DESARROLLO

**TODO**

* Se menciona el modelo de desarrollo a seguir, con sus fases, actividades y resultados esperados en cada uno de ellos.
* **Es necesario especificar porque se usa esa metodología?**

## ANALISIS DE LA TECNOLOGIA

* Se menciona las tecnologías a utilizar (hablar de cloud computing, desarrollo web, tecnologías html5, css, javafx, jsf, Ajax, web services (puede ser)).

# INGENIERIA DE REQUERIMIENTOS

Una vez realizado el análisis de las herramientas case anteriores se determinaron las características básicas de las cuales debe disponer el software en construcción, algunos tomados de los resultados del análisis y otros añadidos a voluntad dependiendo de las características que se desea que el software tenga una vez terminado. Se establecieron los requerimientos funcionales y no funcionales para la aplicación, a continuación se mencionaran cada uno de ellos:

## 

## REQUERIMIENTOS FUNCIONALES

**TODO**

## REQUERIMIENTOS NO FUNCIONALES

**TODO**

## MODELO DE CASOS DE USO

**TODO**

## MODELO DEL DOMINIO

**TODO**

# ANÁLISIS

**TODO**

# DISEÑO

**TODO**

# IMPLEMENTACIÓN

**TODO**

Que va aquí para el documento?

Código?

# PRUEBAS

**TODO**

* Mencionar Unit Tests (Herramienta TestNG), usado para realizar testing por ejemplo a EJBs métodos que prestan servicios específicos.
* Mencionar Tests Cases. (**usar formato**)
* Pruebas basadas en casos de uso.

# DESPLIEGUE

**TODO**

# ASEGURAMIENTO DE LA CALIDAD

**TODO**

* Hablar de Sonar (Herramienta para verificar la calidad del código)
* Hablar de Coverage (Herramienta para verificar la calidad de los Unit Tests que se van a escribir).

# RESULTADOS

**TODO**

# CONCLUSIONES

**TODO**

# REFERENCIAS BIBLIOGRAFICAS

**TODO**

# ANEXOS

**TODO**

1. AMAYA TORRADO, Yegny Karina y HERRERA ANGARITA, Lady Torcoroma. Identificación de las causas que generan problemas en el aprendizaje de fundamentos de programación de computadores en las facultades de ingeniería de las universidades de la ciudad de Cúcuta. Trabajo de Grado Ingeniería de Sistemas. 2003. Cúcuta N.de S. Universidad Francisco de Paula Santander. Facultad de Ingeniería Disponible en la Biblioteca Eduardo Cote Lamus bajo la signatura TIS 371.334A489i. [↑](#footnote-ref-1)
2. CASE (Computer Aided Software Engineering, Ingenieria de software asistida por computadora), son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo los costos del mismo en términos de tiempo, esfuerzo y dinero. [↑](#footnote-ref-2)
3. Página web oficial de Enterprise Architect, <http://www.sparxsystems.com/> [↑](#footnote-ref-3)
4. Tomado del portal web del programa Ingeniería de Sistemas de la Universidad Francisco de Paula Santander, en la sección Presentación del Programa – Objeto de Estudio.

   Disponible en : <http://ingsistemas.ufps.edu.co/presentacion/objeto-estudio.html> [↑](#footnote-ref-4)
5. Cloud Computing, definición, características, tipos y ventajas, información disponible en la enciclopedia virtual Wikipedia, <http://en.wikipedia.org/wiki/Cloud_computing> [↑](#footnote-ref-5)
6. Google Docs es parte de la nube de servicios proporcionada por la empresa Google para sus usuarios, a través de esta nube los usuarios pueden disponer de un kit de herramientas entre los que se incluye un procesador de texto, hojas de cálculo, diseñador de presentaciones, entre otras. [↑](#footnote-ref-6)
7. Creative Commons Corporation. Código Legal. Atribución no comercial compartir igual 2.5 (Colombia). <http://creativecommons.org/licenses/by-nc-sa/2.5/co/legalcode> [↑](#footnote-ref-7)
8. CONTRATO DE LICENCIA DE CÓDIGO BINARIO. Sun Microsystems, Inc. Java 2 Plataform Standard edition, Runtime Environment 5.0. Disponible: <http://www.java.com/es/download/license_jre5.jsp> [↑](#footnote-ref-8)
9. GNU GENERAL PUBLIC LICENSE Versión 3, 29 june 2007. Disponible: <http://www.gnu.org/licenses/gpl.txt> [↑](#footnote-ref-9)
10. Universidad Francisco de Paula Santander. Estatuto estudiantil. Acuerdo 065, agosto 26, 1996.

    Disponible en : <http://www.ufps.edu.co/ufpsnuevo/archivos/reglamentacion/acuerdo065.pdf> [↑](#footnote-ref-10)
11. El Object Management Group u OMG (o por su nombre en español, Grupo de Gestión de Objetos) es un consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA. Es una organización sin ánimo de lucro que promueve el uso de tecnología orientada a objetos mediante guías y especificaciones para las mismas. El grupo está formado por diversas compañías y organizaciones con distintos privilegios dentro de la misma. [↑](#footnote-ref-11)
12. Model Driven Generation (MDG) [↑](#footnote-ref-12)
13. Data Base Management System (DBMS) o Sistema de gestión de bases de datos, son aplicaciones que permiten el almacenamiento, modificación y extracción de información de una base de datos, también provee herramientas para adicionar, modificar, borrar, acceder y analizar datos almacenados en una localización especifica. [↑](#footnote-ref-13)
14. CSV acrónimo de Comma Separated Values (en español es: Valores separados por coma), son archivos de texto usados para representar datos en forma de tabla, en donde las columnas van separadas por comas y las filas por saltos de línea. [↑](#footnote-ref-14)
15. WYSIWYG es el acrónimo de What You See Is What You Get (en español es: Lo que usted ve es lo que usted obtiene) Se aplica a los procesadores de texto y otros editores de texto con formato que permiten escribir un documento viendo directamente el resultado final. [↑](#footnote-ref-15)
16. La arquitectura dirigida por modelos (Model Driven Architecture o MDA) es un acercamiento al diseño de software, propuesto y diseñado por el Object Management Group. MDA se ha concebido para dar soporte a la ingeniería dirigida a modelos de los sistemas de software. MDA es una arquitectura que proporciona un conjunto de guías para estructurar especificaciones expresadas como modelos. [↑](#footnote-ref-16)