# PI153-10-ZoomableModelTool

Ambiente de visualización y manipulación de grafos con ayuda de interfaces aumentables

Mauricio Alberto Sánchez Franco

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
2015

# PI153-10-ZoomableModelTool
# Ambiente de visualización y manipulación de grafos con ayuda de interfaces aumentables

**Autor:**

Mauricio Alberto Sánchez Franco

MEMORIA DEL TRABAJO DE GRADO REALIZADO PARA CUMPLIR UNO DE LOS REQUISITOS PARA OPTAR AL TITULO DE

MAGÍSTER EN INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

**Director**

Ing. Jaime A. Pavlich-Mariscal, PhD

**Comité de Evaluación del Trabajo de Grado**

Ing. César Alberto Collazos Ordóñez, PhD

Ing. Mario Fernando De la rosa Rosero, PhD

**Página web del Trabajo de Grado**

http://pegasus.javeriana.edu.co/~PI153-10-ZoomableModelTool

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
MAESTRÍA EN INGENIERIA DE SISTEMAS Y COMPUTACIÓN
BOGOTÁ, D.C.
Noviembre, 2015

**PONTIFICIA UNIVERSIDAD JAVERIANA**
**FACULTAD DE INGENIERIA**
**CARRERA DE INGENIERIA DE SISTEMAS**

**Rector Magnífico**

P. Jorge Humberto Peláez Piedrahita, S.J.

**Decano Académico Facultad de Ingeniería**

Ingeniero Luis David Prieto Martínez

**Decano del Medio Universitario Facultad de Ingeniería**

Ingeniero Jorge Luis Sánchez Téllez

**DirectorMaestría en Ingeniería de Sistemas y Computación**

IngenieraÁngela Cristina Carrillo Ramos PhD

**Director Departamento de Ingeniería de Sistemas**

Ingeniero Rafael Andrés González PhD

**Artículo 23 de la Resolución No. 1 de Junio de 1946**

*"La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia".*

# AGRADECIMIENTOS

Primera mente le doy gracias a Dios por haberme dado la oportunidad, la inteligencia y la constancia para desarrollar de forma exitosa mi trabajo de grado.

A mis padres y a mi esposa por el apoyo moral tan grande que me han brindado para mantenerme y seguir luchando hasta el final.

A mi director de trabajo de grado el ingeniero Jaime A. Pavlich-Mariscal por todo su compromiso, empeño y orientación que tuvo en éste trabajo de grado. También especiales agradecimientos al ingeniero Leonardo Flórez por toda su asesoría durante las fases de diseño y análisis del experimento.

Sonmuchas las personas a las cuales les debo agradecer haber llegado a finalizar de forma exitosa mi trabajo de grado, entre ellos se encuentran mis compañeros más cercanos de la maestría, jefes en las empresas en que he trabajado los que me han dado los permisos necesarios para asistir a clase y reuniones en la universidad, los 16 compañeros que participaron en la ejecución del experimento y todas las demás personas que de alguna manera contribuyeron para que ésta meta hoy sea una realidad.

# CONTENIDO

# ÍNDICE DE FIGURAS



# ÍNDICE DE TABLAS

## ABSTRACT

The navigation in software models in the current CASE tools is done through the use of navigation trees and scroll bars. This mechanism introduces a discontinuity between the information displayed on screen which can cause cognitive and mechanicalburdens for the users.

The proposal solution to address the problematicdescribed, consisted in apply the Zoomable User Interfaces paradigm for the visualization of software models. To do this, we designed and built a framework for viewing and navigating graphs representing software models.



## RESUMEN

La navegación de los modelos de software en las actuales herramientas CASE es realizada a través del uso de árboles de navegación, y el desplazamiento en los diagramas es realizado típicamente por barras de scroll laterales. Éste mecanismo introduce una discontinuidad entre la información desplegada que puede causar cargas cognitivas y mecánicas para los usuarios.

La solución propuesta para enfrentar la problemática descrita, consistió en aplicar el paradigma de interfaces aumentables para la visualización y navegación de los modelos de software. Para ello, se diseñó y construyó un framework para la visualización y navegación de grafos que representan modelos de software.

# RESUMEN EJECUTIVO

El modelado de software es una herramienta muy importante para ayudar a manejar la creciente complejidad en los requerimientos de los sistemas de software [1]. El modelado, brinda una visión global del sistema permitiendo entenderlo de una manera fácil y rápida.

De la mano del modelado de software se encuentran las herramientas CASE[1], las cuales son aplicaciones de software que asisten a los Ingenieros de Software en su tarea de modelar los sistemas que posteriormente serán desarrollados. En las actuales herramientas CASE, cada modelo es dibujado mediante múltiples diagramas en lienzos separados, la navegación del modelo es realizada a través del uso de árboles de navegación y el desplazamiento en los diagramas es realizado típicamente por barras de scroll laterales.

Un problema importante que se identifica en el mecanismo de navegación usado porla mayoría de las actuales herramientas CASE, es que la información del modelo se encuentra espacialmente particionada en multitud de diagramasque aumentan a medida que el sistema modelado crece.La navegación a través de múltiples diagramas,introduce una discontinuidad entre la información desplegada en diferentes momentos y lugares que puede causar cargas cognitivas y mecánicas para los usuarios [2]. Los usuarios deben navegar entre diferentes diagramasrecolectando la mayor cantidad de detalles de cada uno y descubriendo las relaciones lógicas existentes entre ellos, para posteriormente unirla mentalmente y poder asimilar la estructura total del espacio de información.

La solución propuesta para enfrentar la problemática descrita, consistió en aplicar el paradigma de interfaces aumentables (en inglés, *Zoomable User Interfaces*) para la visualización y navegación de los modelos de software. Para ello, se diseñó y construyó un framework para la visualización y navegación de grafos que representan modelos de software. El framework fue construido sobre tecnología HTML5 y estuvo enfocado en la visualización y navegación de grafos, funcionalidades de creación y edición no están soportadas hasta el momento.

A partir del framework desarrollado, se diseñó y construyó un prototipo para el modelado de software. El prototipo fue construido con dos principales objetivos: el primero, para validar y mejorar el framework; el segundo, para responder la pregunta de investigación a través de la ejecución de un experimento.

El prototipo construido es una herramienta Web que permite visualizar y navegar modelos de software usando el paradigma ZUI. La pantalla está formada por el área de dibujo donde es renderizado el modelo y el árbol de navegación ubicado en la parte lateral de la pantalla. El prototipo soporta los siguientes diagramas: diagrama de clases, de componentes, de paquetes, de despliegue y de casos de uso. Estos modelos son definidos en archivos de formato JSON que son cargados en la herramienta y renderizados en el área de dibujo.

A partir del prototipo construido, se diseñó y ejecutó un experimento compuesto de tres partes: prueba de competencias, formulario de tareas y encuesta de satisfacción. La prueba de competencias fue realizada para verificar que los participantes contaran con los conocimientos básicos en UML y además para clasificarlos en dos grupos: Novatos y Expertos. El formulario de tareas estuvo compuesto de 7 preguntas, en la ejecución de cada tarea se midió el tiempo y el puntaje obtenido. La encuesta de satisfacción buscaba medir el grado de confort que vivió el participante durante el uso de la herramienta, ésta encuesta brinda resultados cualitativos.

En la ejecución del experimento participaron 16 personas, todos Ingenieros de Sistemas con perfiles de desarrollador de software y líder técnico. A partir de la prueba de competencias, quedaron clasificados 7 participantes como novatos y 9 como expertos. Los 16 participantes fueron divididos en tres grupos: el primer grupo de 5 integrantes realizo el experimento en la herramienta de visualización tradicional; el segundo grupo de 6 participantes realizo el experimento en la herramienta de visualización ZUI; el tercer grupo compuesto por 5 participantes realizo el experimento en ambas herramientas (tradicional y ZUI) y además lo repitió por segunda ocasión 15 días después. La asignación de los participantes en cada grupo se realizó a través de un generador de números pseudo-aleatorios uniformes.

Como conclusión, se puede destacar que basado en los análisis estadísticos realizados sobre los datos capturados del experimento, estos sugieren, que ZUI provee una leve mejora en la precisión de las respuestas entregadas por los participantes, aunque debido a la muestra tan pequeña no son valores que puedan ser considerados estadísticamente significativos.

# INTRODUCCIÓN

El modelado de software es una herramienta muy importante para ayudar a manejar la creciente complejidad en los requerimientos de los sistemas de software [1]. El modelado, brinda una visión global del sistema permitiendo entenderlo de una manera fácil y rápida.

De la mano del modelado de software se encuentran las herramientas CASE[1], las cuales son aplicaciones de software que asisten a los Ingenieros de Software en su tarea de modelar los sistemas que posteriormente serán desarrollados. En las actuales herramientas CASE, cada modelo es dibujado mediante múltiples diagramas en lienzos separados, la navegación del modelo es realizada a través del uso de árboles de navegación y, el desplazamiento en los diagramas es realizado típicamente por barras de scroll laterales.

Un problema importante que se identifica rápidamente por este mecanismo usado en las herramientas CASE, es que la información del modelo se encuentra espacialmente particionada en multitud de diagramas, que aumentan a medida que el sistema modelado crece en tamaño y complejidad. Este mecanismo introduce una discontinuidad entre la información desplegada que puede causar cargas cognitivas y mecánicas para los usuarios [2], quienes al no tener toda la información del modelo en un solo lugar, deben navegar entre diferentes diagramas, encontrando las relaciones lógicas existentes entre ellos y recolectando la mayor cantidad de detalles en cada diagrama, para mentalmente unirla y lograr asimilar la estructura total del espacio de información y su localización dentro de ésta.

En este trabajo se propone desarrollar un framework Web para la visualización y navegación de grafos que representan modelos de software, haciendo uso de un paradigma de visualización alternativo llamado *Zoomable User Interfaces* (ZUI en forma corta) [3]. ZUI permite dibujar todo el modelo en un lienzo ilimitado (limitado solo por el hardware) e implementa para ello dos técnicas de visualización que son: zooming y panning [4].

Además de desarrollar el framework para la visualización de grafos, este trabajo incluye el diseño y ejecución de un experimento piloto que busca validar las interfaces ZUI en la visualización de modelos de software.

El presente documento describe todo el proceso que se llevó a cabo durante la construcción del framework ZMT para la visualización de grafos con soporte de interfaces aumentables. Esta descripción inicia desde la fase de diseño, desarrollo y posterior validación a través de un experimento. El experimento busca responder la pregunta de investigación que dio origen al proyecto.

El presente documento está dividido en 7 secciones donde se explica al lector el desarrollo del trabajo de grado realizado. La primera sección sitúa al lector en el contexto de la problemática abordada y la justificación e importancia que tiene su desarrollo; la segunda sección presenta el marco teórico y conceptos sobre la problemática tratada; la tercera sección explica de forma detallada el diseño y desarrollo del framework ZMT junto con el prototipo implementado; la cuarta sección se

enfoca en el experimento describiendo de forma detallada su diseño y ejecución; la quinta sección describe los resultados y su análisis respectivo; la sexta sección describe las conclusiones obtenidas a partir del desarrollo del trabajo de grado; finalmente la séptima sección presenta todas las referencias consultadas durante el desarrollo del trabajo de grado.

# I - DESCRIPCIÓN GENERAL

## 1. OPORTUNIDAD, PROBLEMÁTICA, ANTECEDENTES

Esta sección pretende situar al lector en el contexto de la problemática a resolver sobre el presente trabajo de grado. A continuación se describe y explica lo que pretende resolver el framework ZMT como librería de software.

### 1.1. Descripción del contexto

En la actualidad, existe una multitud de herramientas CASE para el modelado de software, pero en la mayoría de los casos, sino todos, el manejo de diagramas es bastante difícil [5]. Varias son las razones por las cuales esto sucede: los sistemas de software suelen ser dibujados en diferentes diagramas y lienzos separados para representar diferentes partes del sistema y/o diferentes niveles de abstracción, se crean diferentes vistas para diferentes personas, se tienen representaciones abstractas del mismo problema a nivel estático o dinámico. Los diagramas permanecen visualmente aislados aunque están altamente relacionados de una manera lógica [5][6]. Todo esto lleva a que sea difícil concebir el sistema como un todo [6].

Las herramientas CASE actuales, hacen uso de una técnica para escalar diagramas conocida como Zoom Geométrico [5]. Ésta técnica, permite acercar o alejar elementos mostrados en pantalla, aumentando o reduciendo el espacio que ocupan. Este mecanismo, tiene como problema que al aumentar o disminuir el nivel de zoom, el texto puede llegar a ser ilegible [6], en ocasiones demasiado pequeño para ser procesado por el ojo humano y en otras demasiado grande siendo igualmente ilegible; lo mismo ocurre con las imágenes que representan elementos del modelo, a ciertos niveles de zoom se pueden ver borrosas y distorsionadas, haciendo difícil identificarlas. Una mejor aproximación es usar el Zoom Semántico [5], el cual ofrece la posibilidad de cambiar la apariencia de los elementos dependiendo del nivel de acercamiento, mostrando solo aquella información que es relevante para cierto nivel de detalle [6]. Esta característica, permite lograr una visión global rápida del sistema. En un principio, se puede tener una vista general a un mínimo nivel de detalle (beneficiando la orientación del usuario), luego aumentando el nivel de zoom, el usuario va adentrándose en los elementos del sistema que más le interesen.

Una alternativa de visualización que muestra potencial para resolver los problemas anteriormente descritos, es el paradigma *Zoomable User Interfaces*(ZUI).

ZUI es un paradigma de visualización basado principalmente en el despliegue de información con diferentes niveles de detalle (LODs) [1]. Permite tener toda la información distribuida en un plano limitado de dos dimensiones (limitado solo por el hardware). *"La esencia de este enfoque, es que el usuario se mueva a través del espacio y construya un modelo espacial de la información en su cabeza"*[7], tomando ventaja de la percepción espacial y memoria humana. ZUI ofrece la posibilidad de manipular los modelos de software mediante zoom lo que permite alejarse para obtener una perspectiva general, y acercarse para mostrar los detalles más importantes, facilitando la comprensión por parte del usuario.

ZUI posee tres características esenciales [1]:

a. Utiliza el **zoom** como mecanismo primario de navegación a través de diferentes niveles de abstracción.
b. Utiliza el **zoom semántico** para abstraer detalles en diferentes niveles.
c. Utiliza el **panning** para navegar sobre el lienzo a un mismo nivel de abstracción.

El problema fundamental que el paradigma de visualización ZUI pretende resolver, es que los usuarios requieren interactuar con más información de la que físicamente cabe en pantalla [4][7]. El aprovechamiento del espacio disponible, ZUI lo logra mediante trasformaciones geométricas de los elementos visuales. La representación visual de un elemento cambia con respecto al nivel de zoom en el que se encuentre. La filosofía de ZUI, es mostrar solo aquella información de un elemento que para cierto nivel de zoom es apropiada.

El *zooming* facilita la presentación de información en pantallas de tamaño limitado, debido a que permite mostrar u ocultar información dependiendo de su relevancia, siguiendo el principio de *"no toda la representación, necesita ser presentada simultáneamente"*[8].

A diferencia de las interfaces de Scroll, las cuales son efectivas para espacios pequeños, ZUI desarrolla todo su potencial a medida que el espacio de información crece [4].

Mediante el uso de ZUI, se pretende facilitar la comprensión y evolución de los modelos de software como consecuencia de una navegabilidad más natural. Las técnicas usadas actualmente por las herramientas CASE para visualizar y navegar a través de los diagramas de software, generan una pérdida de contexto del usuario al pasar de una manera cortante de un diagrama a otro. Al acceder a un nuevo diagrama para obtener información adicional, se causan fuertes rupturas en el proceso cognitivo del usuario, obligándoloa ubicarse dentro del nuevo diagrama y encontrar la relación con el anterior. Con ZUI, se evitaría la necesidad de acceder a otro diagrama para ver información en diferentes niveles de abstracción o para ver otras partes del sistema, debido a que en ZUI todo el sistema se encuentra dibujado en el mismo lienzo [2][1][6].

La promesa de ZUI viene en gran medida por las siguientes tres expectativas [7]:

- **Son atractivas.** La animación atrapa la atención visual tomando ventaja de las habilidades de la percepción visual humana.
- **Visualmente ricas.** Hay un mayor grado de libertad para estructurar objetos visualmente.
- **Ofrecen el atractivo de simplicidad.** El hecho de encontrar información buscando por ella en un solo lugar implica la promesa de simplicidad.

### 1.2. Problema a resolver

La pregunta de investigación que dio origen al desarrollo del presente proyecto es: "¿El uso de interfaces ZUI en la visualización y navegación de grafos que representan modelos de software ayuda a los usuarios a comprender de forma más eficiente los sistemas modelados comparado con las herramientas CASE tradicionales?".

Para dar respuesta a esta pregunta de investigación, los esfuerzos fueron enfocados en dos temas principales: el primero consistió en desarrollar un framework Web para la visualización de grafos con ayuda de interfaces aumentables (framework de nombre ZMT); la segunda parte, consistió en el diseño y ejecución de un experimento que permitiera dar respuesta a la pregunta de investigación planteada. El análisis de los resultados obtenidos en la ejecución del experimento, fue realizado a través de la aplicación de técnicas estadísticas.

Como el tema principal del proyecto fue responder la pregunta de investigación, y esta fue planteada claramente en temas de visualización y navegación de grafos, el framework desarrollado se enfocó en implementar funcionalidad referente únicamente a visualización y navegación de grafos que representan modelos de software con soporte de interfaces ZUI, dejando de lado funcionalidad referente a creación y edición de modelos.

Después de tener desarrollado el framework ZMT, se realizó el desarrollo de un prototipo para la visualización de grafos que hace uso extensivo del framework ZMT. El prototipo desarrollado es una herramienta Web que permite cargar modelos de software definidos en archivos JSON y navegar a través de ellos haciendo uso de las técnicas de Zooming & Panning. El prototipo desarrollado es descrito en forma detallada en la sección 3 del presente documento. Este prototipo fue usado por los participantes del experimento para la ejecución de diferentes tareas, cuyo resultado y posterior análisis estadístico busca responder la pregunta de investigación inicialmente planteada.

### 1.3. Justificación

El modelado de software es una herramienta que busca mejorar la comprensión del sistema por parte de los diferentes involucrados, en especial, aquellos que intervienen de forma directa en alguna fase del ciclo de desarrollo, entre ellos se puede nombrar: desarrolladores, diseñadores, analistas y arquitectos de software.

El actual paradigma de visualización usado por las herramientas CASE ha alcanzado el límite de sus posibilidades [8]. Hoy los sistemas son cada vez más

grandes y complejos y por lo tanto requieren mayor cantidad de diagramas para modelarlos con un nivel de detalle aceptable.

La investigación propuesta en el presente proyecto busca desarrollar y evaluar un framework Web con soporte de ZUI para la visualización de grafos que representan modelos de software.

La importancia de este proyecto radica en dos aspectos principales: el primero en proveer un framework Web con soporte ZUI que podrá ser usado por los diferentes roles involucrados en tareas de diseño y desarrollo de software; el segundo, en realizar la validación que permita corroborar que el uso de este paradigma de interfaces ZUI realmente ayuda a los usuarios en la visualización de los modelos en mejor manera que las herramientas CASE tradicionales.

Mediante el framework ZMT desarrollado, se busca proveer a los usuarios de una herramienta que les facilite la comprensión de los modelos de software tomando ventaja del razonamiento espacial y memoria humana [9]. Al facilitar la comprensión del modelo, se mejora la eficiencia del usuario y como consecuencia podría aumentar su productividad y la calidad del producto desarrollado.

Como factor complementario, el desarrollo del framework fue realizado con soporte para la Web y el lenguaje de programación usado fue JavaScript [10]. Además de visualizar el modelo y permitir la interacción con dispositivos apuntadores (como el mouse), esta soportado el uso de la tecnología multi-touch,con la cual el usuario usando sus dedos para realizar los diferentes gestos touch, logra interactuar de igual manera con los modelos.

# 2. DESCRIPCIÓN DEL PROYECTO

A partir de la problemática encontrada y la oportunidad de investigación descritas en la sección anterior, se ha generado la necesidad de desarrollar éste proyecto de investigación con los siguientes objetivos:

## 2.1. Objetivo General

Diseñar, Implementar y Evaluar una herramienta para la visualización de grafos que representan modelos de software soportando la técnica de visualización y navegación conocida como Interfaces Aumentables (en inglés, *Zoomable User Interfaces - ZUI*).

## 2.2. Objetivos Específicos

1. Realizar el estado de arte sobre trabajos relacionados, algoritmos, librerías y framework centrados en interfaces ZUI.
2. Analizar y especificar los requerimientos funcionales y no funcionales que debe cumplir la herramienta.
3. Diseñar la arquitectura de la herramienta.
4. Implementar la herramienta siguiendo la arquitectura definida apoyándose en un framework que soporte la técnica de visualización ZUI.

5. Diseñar y ejecutar un experimento que permita determinar si la técnica de visualización ZUI implementada en la herramienta mejora la comprensión de los modelos de software comparada con las técnicas de visualización usadas actualmente por herramientas CASE.

# 3. METODOLOGÍA

Para el desarrollo del proyecto se usó el Proceso Unificado Ágil (Agile UP) [11]. Éste proceso adopta muchas de las técnicas ágiles de Programación Extrema (XP) [12] y de otros procesos ágiles, incluso conservando algo de la formalidad del Proceso Unificado de Rational (RUP) [13].

La metodología usada durante el desarrollo del presente trabajo de investigación está constituida de 5 fases metodológicas, las cuales son descritas a continuación. Por cada fase establecida se define las actividades principales que fueron desarrolladas en cada una de ellas.

**Fases Metodológicas**

1. Investigación Especulativa
2. Concepción
3. Elaboración
4. Construcción
5. Validación

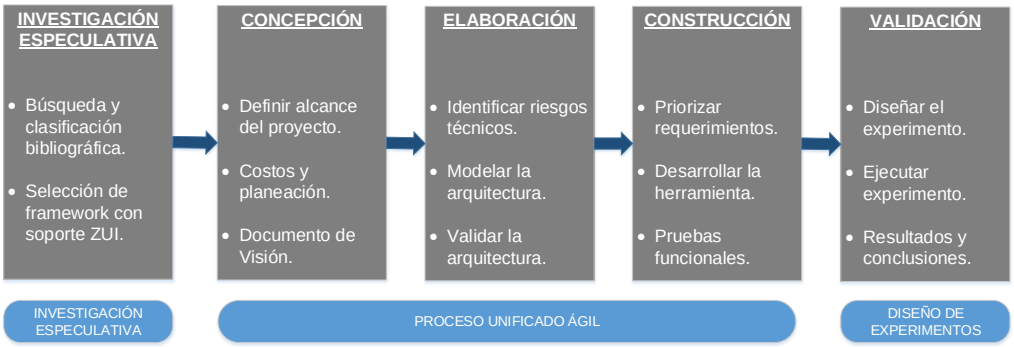


Figura 1: Metodología

Las fases de Concepción, Elaboración y Construcción fueron ejecutadas de forma iterativa e incremental, en ellas se construyó la herramienta *Zoomable Modeling Tool*. Las fases de Investigación Especulativa y Validación fueron ejecutadas una única vez, la primera de ellas en la etapa inicial del proyecto para realizar el estado del arte e investigación previa; la fase de validación en la etapa final del proyecto, para validar mediante el experimento si las interfaces aumentables aplicadas en la visualización de modelos de software mejoran la comprensión de los modelos de software por parte del usuario.

### 3.1. Fase 1: Investigación Especulativa

Durante la fase inicial del proyecto se usó el método de la Investigación Especulativa. Éste es un método que se basa en la construcción teórica a partir de conceptos, preguntas filosóficas y generación de modelos teóricos. La investigación especulativa fue usada para desarrollar el estado del arte del proyecto y la escogencia del framework base con soporte de ZUI.

**Actividades**

a) Búsqueda y clasificación bibliográfica relacionada con técnicas de visualización e interacción intuitiva.
b) Revisión de los frameworks ZUI existentes con soporte para la Web, con el objetivo de seleccionar el más apropiado para ser utilizado en el proyecto.

### 3.2. Fase 2: Concepción

En esta fase se hace uso del Proceso Unificado Ágil (AUP). El objetivo principal de esta fase fue lograr un consenso sobre los objetivos, alcance, estimación y financiación del proyecto.

**Actividades**

a) Definir el alcance del proyecto.
b) Estimar los costos y la planeación.
c) Definir riesgos.
d) Elaborar Documento de Visión.

### 3.3. Fase 3: Elaboración

En esta fase fue desarrollada siguiendo la guía del Proceso Unificado Ágil (AUP). El objetivo principal en esta fase fue confirmar la arquitectura del sistema que sería desarrollado.

**Actividades**

a) Identificar y gestionar riesgos técnicos.
b) Identificar la arquitectura.
c) Modelar la arquitectura.
d) Validar la arquitectura.

### 3.4. Fase 4: Construcción

En esta fase al igual que las dos inmediatamente anteriores se usó el Proceso Unificado Ágil (AUP). Todo el esfuerzo dedicado en esta fase estuvo enfocado en el desarrollo de la herramienta *Zoomable Modeling Tool – ZMT*. Esta fase se dio por finalizada solo en el momento que se cumplieron a cabalidad con cada uno de los requerimientos funcionales y no funcionales definidos en la primeras fases y por tanto la herramienta construida estuviese lista para ser entregada para el uso de los usuarios seleccionados para realizar la validación a través del experimento.

**Actividades**

a) Priorizar los requerimientos de software.
b) Construir la herramienta.
c) Probar y corregir defectos del software.
d) Generar la documentación.

### 3.5. Fase 5: Validación

Durante el desarrollo de ésta fase se usó la metodología de Diseño de Experimentos. El Diseño de Experimentos es una metodología estadística destinada a la planificación y análisis de un experimento. En esta fase se realizó la definición del protocolo, la selección de variables dependientes e independientes, los factores ruido, los factores de control, los recursos necesarios, la viabilidad del estudio, los mecanismos de observación y seguimiento, el perfil de las personas participantes, los escenarios de uso, etc.

El experimento fue diseñado y ejecutado como mecanismo para la validación de la hipótesis generada a partir de la pregunta de investigación formulada al inicio del presente proyecto de investigación.

Para la ejecución del experimento fue seleccionado un grupo de 15 personas como participantes del experimento. Las personas seleccionadas debían cumplir con perfil de Ingenieros de Sistemas o carreras afines y poseer conocimiento en el modelado de software y manejo de herramientas CASE. Dentro del grupo de participantes fue necesaria la existencia de usuarios tanto novatos como expertos en el modelado de software.

A partir de la ejecución de experimento, se logró recolectar medidas directas e indirectas. Las medidas directas, fueron obtenidas a partir de tareas que cada participante ejecutó sobre la herramienta. Las medidas indirectas fueron obtenidas a partir de una encuesta de satisfacción que buscó medir de manera subjetiva el grado de confort que experimento el usuario durante el uso de la herramienta.

**Actividades**

a) Diseño del experimento que permitió dar respuesta a la pregunta de investigación planteada en el presente proyecto.
b) Ejecución del Experimento.
c) Análisis de resultados y generación de las conclusiones del experimento.

## II - MARCO TEÓRICO

Para comprender el desarrollo del framework ZMT y de la investigación ejecutada a su alrededor, es necesario conocer la teoría de las interfaces aumentables, los tra-

bajos e investigaciones previas realizados en esta área y los avances por ellos alcanzados.

Esta sección se divide en dos grandes aspectos: marco conceptual y marcocontextual. El marco conceptual se encarga de presentar conceptos que son de alta importancia para la comprensión de lo expuesto en secciones posteriores. El marco contextual hace un resumen del estado del arte hasta la fecha, describiendo los desarrollos e investigaciones más relevantes en el área de las interfaces aumentables enfocadas en la visualización de diagramas.

# 1. MARCO CONCEPTUAL

A continuación se describe los conceptos más importantes para entender el propósito de la investigación y del trabajo de grado.

Los modelos de software permiten abstraer conceptos clave del sistema ocultando detalles de implementación, en otras palabras, los modelos permiten visualizar el sistema en construcción sin entrar en demasiados detalles. A través de un modelo de software, se puede describir un sistema en su forma estática y dinámica. En forma estática, se puede definir la estructura del sistema a través de: diagramas de clases, diagramas de componentes, diagramas de paquetes, diagramas de despliegue, etc. En forma dinámica, se puede describir a través de: diagramas de casos de uso, diagramas de comunicación, diagramas de secuencia, etc.

El lenguaje para modelado de sistemas software más usado y conocido en la actualidad es UML (lenguaje de modelado unificado)[14], [15]. UML es un lenguaje grafico que permite describir visualmente un sistema software a través de diferentes diagramas. UML cuenta con varios tipos de diagramas usados para describir diferentes aspectos del sistema de software modelado, entre los diagramas están: diagrama de clases, diagrama de casos de uso, diagrama de comunicación, diagrama de despliegue, etc.

Comúnmente los sistemas software son modelados a través del uso de herramientas CASE, las cuales pretenden ser un apoyo en todas las etapas del ciclo de vida del desarrollo de un producto software, buscando reducir la complejidad del sistema y aumentar la calidad del producto. Las herramientas CASE se definen como: *"conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información, completamente o en alguna de sus fases"* [16].

Las actuales herramientas CASE implementan el paradigma tradicional de ventanas, iconos y punteros, conocido como *WIMP*. Las interfaces *WIMP* han alcanzado sus límites en cuanto a tres mayores desafíos[8]:

- El crecimiento exponencial del monto de información con la cual cada usuario debe tratar.
- La distribución de esta información sobre múltiples dispositivos electrónicos.

- Y el creciente rango de usuarios de computadores, con su amplia variedad de habilidades, necesidades y expectativas.

Un paradigma alternativo que muestra potencial para solucionar estos problemas, es el conocido como interfaces de usuario aumentables - ZUI. Las interfaces ZUI[1], proveen un lienzo de tamaño limitado (limitado solo por el hardware) en el cual la información puede ser representada como una jerarquía en un espacio de dos dimensiones. Esto permite a los usuarios recordar posiciones relativas de los objetos en lugar de las posiciones absolutas, siendo similar al reconocimiento de los elementos en un escritorio.

El paradigma ZUI permite filtrar la información mostrada, visualizando solo aquella información relevante según el nivel de acercamiento. ZUI implementa dos técnicas de visualización, las cuales son: *zooming* y *panning*[1],[17],[4]. El *zooming*, permite navegar a través de diferentes niveles de abstracción, acercándose para acceder a un mayor detalle o alejándose para tener una vista más general. El *panning*, permite desplazarse sobre el lienzo sin alterar la distancia al modelo,permaneciendo en el mismo nivel de detalle.

La técnica de zooming implementado por el paradigma ZUI, es el zoom semántico, que a diferencia del zoom geométrico, permite cambiar la representación geométrica de un elemento a medida que la escala de zoom cambia. Éste cambio se realiza para modificar el significado semántico del elemento, permitiendo visualizar un mayor nivel de detalle en caso de ir a un mayor nivel de zoom o proveer un significado más abstracto en caso de alejarse a un menor nivel de zoom [4]. A continuación se muestra dos ejemplos que permiten explicar de forma más clara el concepto de interfaz aumentable y zoom semántico.

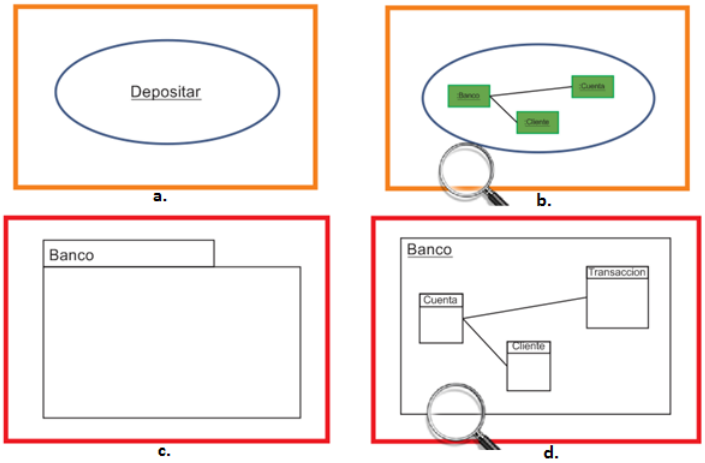


Figura 2: Interfaz Aumentable y Zoom Semántico

El primer ejemplo mostrado en la parte superior de la figura (referenciado con las letras **a.** y **b.**), corresponde al zoom semántico para el caso de uso **"Depositar".** La

parte **a,**muestra el caso de uso en su forma original (sin zoom); la parte **b,**muestra el caso de uso después de una operación de zoom-in, ahoramuestra un mayor nivel de detalley permite observar el diagrama de comunicación de los objetos que realizan el caso de uso.

El segundo ejemplo mostrado en la parte inferior de la figura (referenciado con las letras **c.** y **d.**), corresponde al zoom semántico del paquete **"Banco"**.La parte **c,** muestra el paquete Banco en su estado original (sin zoom); la parte **d.**muestra el paquete después de una operación de zoom-in, ahora el paquete permite observar el diagrama de clases que lo componen.

Después de tener entendidos los conceptos propios de las interfaces ZUI, es necesario definir brevemente algunos conceptos del diseño de experimentos. En la parte final del proyecto de investigación y como mecanismo para realizar la validación de la hipótesis planteada se realizará la ejecución de un experimento. A continuación se definen algunos conceptos básicos que se deben conocer para entender con claridad las fases del diseño, ejecución y análisis del experimento.

Una hipótesis estadística [18],es una afirmación sobre los valores de los parámetros de una población o proceso, que puede probarse a partir de la información contenida en una muestra representativa de la población o proceso.

Una muestra representativa, es un subconjunto de individuos seleccionados adecuadamente de una población, la cual conserva los aspectos y características clave de la población en general [18]. Al decir, "seleccionados adecuadamente", se refiere a que deben ser seleccionados de forma completamente al azar, evitando sesgos durante el proceso de selección.

A partir de los datos recolectados en la muestra representativa, es posible realizar un proceso de inferencia estadística, que permite hacer afirmaciones validas de la población. [18], [19]. Por ejemplo, se toma una muestra representativa de la población A, de la muestra se calcula su Media (X) y su Varianza (S), con estos dos valores observados de la muestra, se puede hacer inferencia acerca de los valores de la media y la varianza poblacional.

Para la validación de hipótesis estadísticas de hace uso de las distribuciones de probabilidad. "*Una distribución de probabilidad de X , relaciona el conjunto de valores de X con la probabilidad asociada con cada uno de estos valores"*[18]. Ésta probabilidad puede ser mostrada a través de tablas o por medio de gráficas dibujadas a partir de una función de probabilidad.

En la validación de las hipótesis planteadas en el presente proyecto, se usa la distribución T-Student, la cual es una de las distribuciones de probabilidad que más se usa en intervalos de confianza y pruebas de hipótesis. La grafica de la distribución t-student al igual que la distribución normal es representada como una campana de Gauss, por encima del eje horizontal y centrada en el valor de 0 sobre el eje vertical.

La distribución t-student es usada cuando la población sigue una distribución normal pero el tamaño de la muestra es menor a 30 individuos (n < 30). La distribución t-student tiende a la distribución normal estándar a medida que el tamaño de la muestra crece, con un tamaño de muestra mayor de 45 (n > 45) las dos distribuciones son prácticamente iguales.

La distribución t-student está definida por tres parámetros: la media (X), la varianza (S) y los grados de libertad (gl). Los grados de libertad se definen con respecto al tamaño de la muestra, y sigue la formula (gl = n-1) donde n es el tamaño de la muestra. Para una muestra de tamaño 10, los grados de libertad son 9 (gl = n -1 = 10 -1 = 9).

# 2. MARCO CONTEXTUAL

El presente trabajo se enmarca en dos grandes áreas de la informática: el Análisis y Diseño de Software y la Interacción Humano Computador – HCI [20],[21],[22]. El Análisis y Diseño de Software, está involucrado por el diseño y construcción del framework ZMT con soporte de ZUI como técnica de visualización y navegación de grafos que representan modelos de software; el área de HCI, está involucrada en el diseño y validación de la interacción de las interfaces aumentables para la visualización de grafos que representan modelos de software.

La problemática expuesta en la sección anterior, hace referencia a una oportunidad de mejora muy específica: el uso del paradigma de visualización ZUI en la visualización de grafos que representan modelos de software, podría mejorar la comprensión del modelo por parte del usuario, ayudándolo a cometer menos errores. Ésta conjetura, es la que se quiere validar durante el desarrollo de la presente investigación.

Teniendo en cuenta la problemática expuesta y las áreas de la informática donde se enmarca esta investigación, existen pocos trabajos que engloben los temas anteriormente mencionados.

Uno de los trabajos en esta área, y la base de la formulación del presente trabajo de investigación, es el paper titulado “Un ambiente de meta-modelamiento y visualización basado en el paradigma de *zoomable user interfaces*”, el cual está muy enfocado en la problemática a manejar y realiza la descripción de un prototipo de herramienta meta-case llamado ZooMEnv, que su funcionalidad principal es la definición e instanciación de sintaxis concretas (notación) de lenguajes visuales de modelamiento [1].

Otro trabajo importante es un paper titulado “ORRIL: a simple building blocks approach to zoomable user interfaces”, el cual es un framework que define 4 componentes básicos: Objetos, Regiones, Relaciones y Lógica de Interface. A través de estos 4 componentes, es posible diseñar cualquier interface ZUI independiente del dominio de aplicación. El objetivo de ORRIL es ayudar en el diseño y construcción de ZUIs. Desde la perspectiva de un desarrollador, ORRIL puede ser usado como el

modelo base para el diseño y construcción de un framework ZUI, además, ORRIL es útil para clarificar el proceso que ocurre dentro de una aplicación ZUI [23].

Otro trabajo a resaltar es un paper titulado “A Zoomable User Interface forPresentingHierarchical Diagrams on Large Screens”. En este paper se presenta el diseño, implementación y evaluación inicial de una interfaz aumentable dedicada a presentar un modelo de diseño jerárquico grande de un sistema mecatrónico complejo. La estructura jerárquica del modelo es ilustrada por medio de notación visual y consiste de más de 800 elementos. La interfaz aumentable construida es usada para realizar presentaciones a una audiencia pasiva, para la presentación se hace uso de pantallas de gran tamaño y de alta resolución. El presentador dispone de funcionalidades para la realización de zooming y panning que le permiten navegar a niveles más detallados de la información[24]. El framework ZMT se diferencia de la interfaz aumentable propuesta en este paper en varios aspectos: primero, la problemática atacada está enfocada en modelos de software; segundo, la herramienta es manipulada directamente por el usuario, no está diseñada como herramienta para realizar presentaciones; tercero, la herramienta ZMT puede ser usada tanto en computadores de escritorio como en pantallas de gran tamaño, además soporta la tecnología multi-touch.

En cuanto a frameworks y librerías construidas con soporte de ZUI se encuentran:

- Pad, la interfaz Pad fue la primera en introducir el concepto de zoom semántico al escritorio y además introdujo el concepto de “plano infinito de información de dos dimensiones”, en el cual cada objeto Pad ocupa un cierto espacio en el sistema [25].
- Pad++ es un toolkit de propósito general para la creación e interacción con información estructurada basado en interfaces aumentables, construida en base a dos premisas: la primera, proveer una animación suave en las transiciones entre diferentes niveles de zoom para mantener a los usuarios orientados en términos de su contexto en el sistema; la segunda, construirlo para que sea relativamente fácil de usar y que terceros puedan construir aplicaciones basadas en él [26].
- Jazz es un toolkit de propósito general escrito en Java para la creación de aplicaciones ZUI usando gráficos 2D, está organizado para soportar animación eficiente, actualizaciones rápidas de la pantalla y fotogramas de alta calidad [27].
- Piccolo es un toolkit basado en Jazz, ofrece una forma revolucionaria de crear robustas aplicaciones gráficas en Java y C#, con llamativos efectos visuales como zooming, animación y múltiples representaciones, soporta el desarrollo de programas gráficos estructurados en 2D en general e interfaces aumentables en particular [28].
- ZVTM es un toolkit implementado en Java diseñado para facilitar la tarea de creación de complejos editores visuales en los que una gran cantidad de objetos se mostrarán o contienen formas geométricas complejas que necesitan ser animados [29].

En cuanto a herramientas específicas para modelado de software con soporte de interfaces aumentables se encuentran:

- Pounamu es una herramienta Meta-CASE implementada en Java usada para construir lenguajes visuales específicos del dominio, permite especificar el meta-modelo que define los constructos de lenguajes de programación visual, y notaciones graficas usadas para visualmente representar la sintaxis del lenguaje en múltiples vistas. Pounamu hace uso del framework Jazz para proveer el soporte de ZUI como alternativa a las interfaces de edición convencionales[30].
- ADORA es una herramienta de modelado que permite al usuario controlar libremente el nivel de detalle en diferentes partes del modelo para reducir el tamaño y complejidad del diagrama que está siendo visualizado. Su concepto de visualización, está basado en la técnica de vistas de ojo de pescado (fisheye views), lo que permite mostrar detalle local y contexto global juntos en una sola vista, liberando al usuario de la tarea de integrar mentalmente detalle y contexto [31]. El framework ZMT se diferencia de ADORA en el uso de zooming y panning como las técnicas de visualización, y también, en la tecnología usada para su desarrollo, ZMT es un framework web, mientras ADORA es un conjunto de plugins para el ambiente integrado de desarrollo de Eclipse.
- SHriMP Views introdujo un nuevo concepto conocido como *vistas intercambiables anidadas,* que consiste en diseñar vistas intercambiables para diferentes niveles de abstracción.SHriMP fue inicialmente diseñado para navegar sistemas software, pero ha sido personalizado a una variedad de dominios de conocimiento, como son: la visualización de ontologías y diagramas de flujo. Para el caso de un sistema software, este puede ser navegado a través de diferentes vistas, la vista gráfica, la vista de código fuente, la vista de javadoc, la vista de arquitectura, la vista de documentación textual, etc. [32]. A diferencia de SHriMP, el framework ZMT está diseñado para navegar a través de grafos que representan modelos de software, implementa el paradigma de visualización ZUI para la navegación y ha sido implementado con tecnología HTML5.

Adicionalmente, en la Pontificia Universidad Javeriana se han desarrollado 2 trabajos de grado relacionados a la problemática tratada:

- FV-GAIA es un framework para la edición y manipulación de grafos que representan modelos de software. Está construido en lenguaje Java y brinda soporte de interfaces aumentables tanto para la edición como para la navegación de los modelos [16]. Ha sido construido en base a los requerimientos establecidos por ZoomEnv [1].
- ZoomTI++ es un framework construido en C++ para la visualización y manipulación de grafos que representan modelos de software, implementa ZUI como paradigma de visualización. Puede llegar a ser considerado el sucesor

de FV-GAIA, adhiriendo soporte para la navegación a través del uso de la tecnología multi-touch [33].

El framework ZMT se diferencia de FV-GAIA y ZoomTI++ en dos principales aspectos: primero, ZMT es implementado sobre tecnología web; segundo, el alcance del proyecto ZMT implica realizar una validación a través de la ejecución de un experimento que permita validar la utilidad del uso de interfaces aumentables en la visualización y navegación de grafos que representan modelos de software.

# III – EL FRAMEWORK ZMT

*Zoomable Modeling Tool - ZMT*es un framework de visualización de grafos que representan modelos de software con un propósito específico: proveer un framework Web de visualización de grafos con soporte ZUI como técnica principal de visualización.

ZMT permite la visualización y navegación de grafos que representan modelos de software con soporte ZUI, implementando el zooming y panning como técnicas de visualización. Por medio del uso del framework, es posible manipular los grafos mediante zoom permitiendo alejarse para obtener una perspectiva general, y acercarse para mostrar los detalles más importantes, facilitando de esta manera la comprensión por parte del usuario.

ZMT al implementar el paradigma de visualización de ZUI aprovecha tres características esenciales [1]:

1. Utiliza el zoom como mecanismo primario de navegación a través de diferentes niveles de abstracción.
2. Utiliza el zoom semántico para abstraer detalles en diferentes niveles.
3. Utiliza el panning para navegar entre entidades a un mismo nivel de abstracción.

El zoom semántico [1] es una característica de ZUI que permite cambiar la forma geométrica de los elementos a medida que el nivel de zoom cambia. Esta técnica permite de forma dinámica y a solicitud del usuario, mostrar u ocultar detalles de los elementos al acercarse y/o alejarse del modelo por medio del uso de zoom-in y zoom-out.

El zoom semántico es mostrado en la *Figura 3.* En ésta imagen se puede apreciar tres niveles de zoom para paquete de nombre ***Banco***. La representación geométrica para el paquete en el nivel de zoom 1, es formado por un rectángulo con una pequeña pestaña en la parte superior donde va definido el nombre del paquete.

El nivel de zoom 2 para el paquete ***Banco***, está formado por el rectángulo ahora sin pestaña y, el nombre del paquete va ubicado en la esquina superior izquierda. Adi-

cionalmente, en este nivel de zoom, se puede ver un mayor detalle del paquete, permitiendo observar que contiene tres clases: ***Cuenta***, ***Cliente*** y ***Transacción***. Para cada una de las clases se muestra el nombre y las líneas que representar asociación entre ellas.

Para el nivel de zoom 3, la forma geométrica del paquete ***Banco*** continúa siendo el rectángulo igual al definido en el anterior nivel de zoom, pero a diferencia de su antecesor, ahora se visualiza la clase ***Cuenta*** a una distancia más cercana logrando observar el nombre y atributos que la forman. A causa del acercamiento sobre la clase ***Cuenta***, las clases ***Cliente*** y ***Transacción*** no son visibles porque se encuentran fuera de los límites de la pantalla, para observarlas, será necesario realizar un desplazamiento (panning) del lienzo arrastrándolo en dirección izquierda de la pantalla (las clases Cliente y Transacción como se puede ver en la *Figura 3* para el nivel de zoom 2 se encuentran a la derecha de la clase Cuenta).

Aunque el nivel de zoom 4 no es mostrado en la *Figura 3*, en ese nivel de zoom la clase ***Cuenta*** mostraría además del nombre y atributos, los métodos por ella implementados.

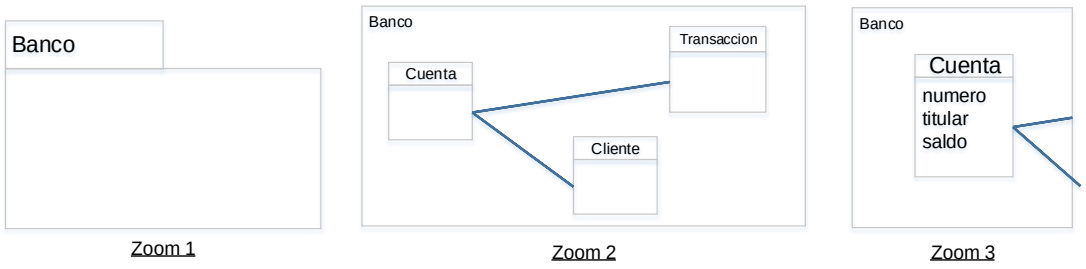


Figura 3: Zoom Semántico

Las siguientes secciones, describen de forma detallada el proceso realizado para la selección del framework base y el diseño e implementación del framework ZMT.

# 1. SELECCIÓN DE FRAMEWORK BASE

La selección del framework base es un paso del proceso de diseño que lleva a tomar una decisión muy importante que puede afectar positiva o negativamente el rumbo del proyecto. Es una decisión de alto impacto para el proyecto que debe ser tomada con la mayor objetividad posible, seleccionando un conjunto de librerías candidatas, evaluándolas en base a un conjunto de criterios, verificando el soporte de cada una de ellas frente a los requerimientos predefinidos para el framework, evaluando el desempeño que puede alcanzar cada librería, etc. A continuación se describe el proceso realizado para la selección de la librería base usada en el desarrollo de la herramienta ZMT.

### 1.1. Librerías Candidatas

Para la selección de librerías candidatas,se realizó una búsqueda de librerías que proveen soporte Web para el renderizado de gráficos vectoriales, en especial que permitan renderizar y manipular grafos (nodos y aristas). Con este criterio de búsqueda, se ha encontrado y preseleccionado las siguientes librerías, en adelante conocidas como librerías candidatas.

- D3 – Data Driven Documents [34]
- jQuery[35]
- Raphael.js[36]
- KineticJS[37]
- HTML5 Canvas[38]

Estas librerías candidatas fueron sometidas a tres elementos de evaluación: **Criterios de Evaluación**, **Soporte de Requerimientos** del framework ZMT y **Desempeño** en el renderizado y animación de elementos tipo nodo.

### 1.2. Criterios de Evaluación

La definición de criterios es realizada en base a las caracteristicas deseables para el ambiente de visualización de grafos con soporte ZUI. La siguiente tabla muesta los criterios utilizados para evaluar el soporte de cada uno de ellos por parte de las librerias candidatas.

| **Zooming** | **Panning** | **Drag & Drop** | **Canvas** | **Gráficos Vectoriales SVG** |
|---|---|---|---|---|
| Técnica para acercar o alejar la información. | Técnica para desplazar el lienzo a través de la pantalla. | Técnica que permite arrastrar y soltar elementos sobre la pantalla. | Lienzo para dibujo. | Una imagen SVG puede ser redimensionada, tanto como se requiera, sin pérdida de calidad de imagen. Esto no es así con un mapa de bits. |
| **Cross-Browser** | **Desempeño** | **Documentación/Ejemplos** | **Selectores CSS3** | **Utilidades del DOM** |
| Debe brindar soporte sobre los diferentes | El desempeño de la librería no debe verse afectado grave- | Debe disponer de amplia documentación y ejemplos de programación. | Permiten realizar busqueda y manipulación | Tener soporte para la manipulación del DOM (Modelo de Ob- |

| | | | | |
|---|---|---|---|---|
| browsers del mercado. | mente por la cantidad de elementos renderizados. | | de objetos en el DOM. | jetos del Documento). |
| **Manejo de Eventos**<br><br>Soportar el manejo de diferentes eventos generados a partir del mouse y touch. | **Animación**<br><br>Soportar animaciones fade-in y fade-out de los elementos dibujados en el canvas | **Soporte Ajax**<br><br>Soporte de Ajax para el manejo asincronico de procesos. Ejemplo: cargar archivos JSON de los modelos. | **Soporte Multitouch**<br><br>Proveer soporte para los diferentes eventos multitouch. | **Facilidad de Programación**<br><br>Cantidad de líneas de código requeridas para implementar la funcionalidad. |

Tabla 1: Criterios de Evaluación

### 1.3. Consolidación y Resultados del Proceso de Selección

Al finalizar el proceso de evaluación de cada una de las librerías candidatas con respecto a los tres elementos: **Criterios**, **Requerimientos** y **Desempeño**, se tienen los siguientes resultados consolidados.

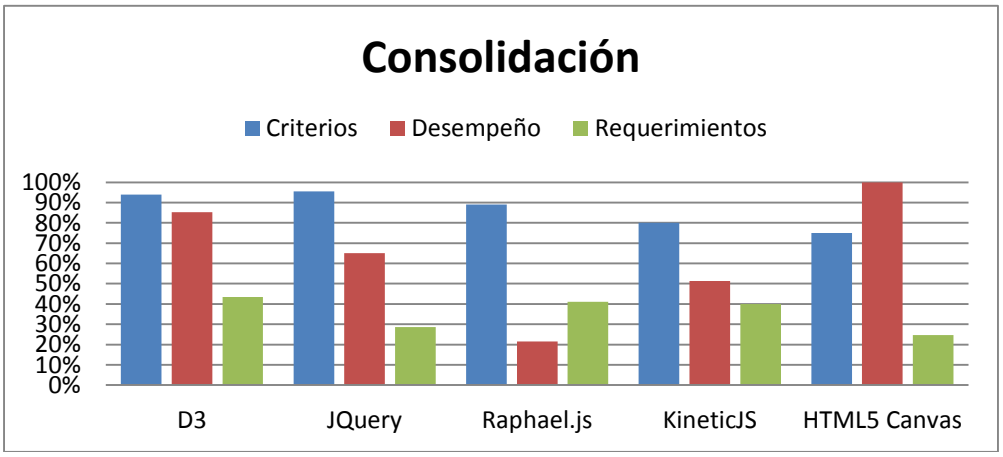


Figura 4: Consolidación resultados librería base

A cada elemento (criterios, desempeño y requerimientos) se le asigna un peso fijo de valor muy similarque representa la importancia que cada elemento tiene para la decisión de la librería base. Los pesos asignados a cada elemento son:

- Criterios: 30%
- Desempeño: 30%

- Requerimientos: 40%

Con estos pesos para cada uno de los tres elementos y los resultados consolidados de cada librería frentre a cada elemento, se realiza un promedio de los tres elementos teniendo en cuenta el valor de peso asignado a cada uno, generando un valor final por cada librería. El resultado del proceso es mostrado en la siguiente imagen:

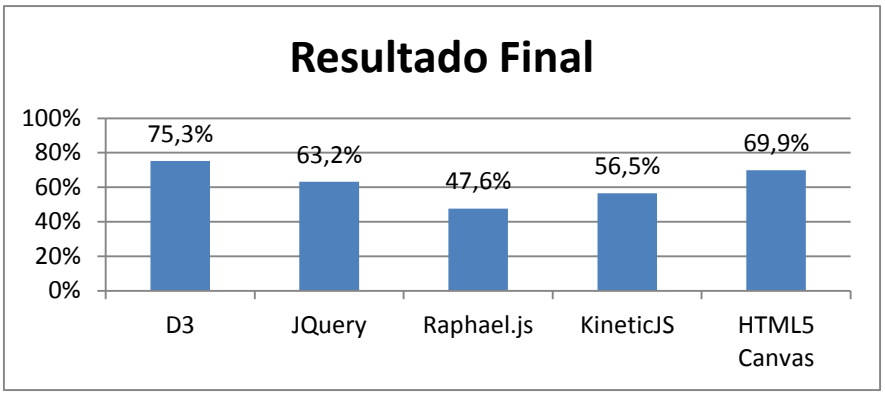


Figura 5: Selección de librería base

Al observar los resultados mostrados en la figura anterior, es claro que la librería D3 ha sido la vencedora alcanzado un porcentaje del 75% de cumplimiento de las evaluaciones realizadas. HTML5 Canvas ha logrado el segundo lugar con un porcentaje del 68,9% de cumplimiento de las evaluaciones realizadas.

Por lo tanto, al finalizar el proceso de selección, habiendo aplicado los criterios establecidos, la evaluación frente a los requerimientos y el desempeño alcanzado de cada librería, se llega a la conclusión de seleccionar la librería D3 - Data Driven Documents como la librería base para el desarrollo del framework ZMT. La librería D3 ha alcanzado la mayor medida en las evaluaciones realizadas, por lo tanto, ofrece el mejor soporte entre las librerias candidatas preseleccionads para el desarrollo del framework ZMT con los requerimientos establecidos.

El documento detallado del proceso de selección puede ser encontrado en el anexo **Selección de Framework Base.**

# 2. DISEÑO DEL FRAMEWORK ZMT

Esta sección describe el diseño realizado para la implementación del framework ZMT. El diseño inicia desde la fase de definición de requerimientos tanto funcionales como no funcionales, el diseño de la arquitectura de software que soportará los requerimientos establecidos, el diseño de clases necesarias para implementar las funcionalidades y, se finaliza con el diagrama de despliegue que muestra la distribución de los diferentes componentes y librerías requeridos para su puesta en marcha en un ambiente productivo.

### 2.1. Requerimientos

Los requerimientos establecidos para el framework ZMT están basados en los requerimientos diseñados para la herramienta ZoomEnv [1]. ZoomEnv es una herramienta de modelado y meta-modelado diseñada por el Ing. Jaime AndrésPavlich-Mariscal, PhD profesor de la Universidad Javeriana Bogotá Colombia.

Haciendo claridad sobre el origen de los requerimientos, a continuación se describe aquellos requerimientos que han sido seleccionados para ser implementados por el framework ZMT. La sección 2.2 describe los requerimientos funcionales que debe implementar el framework. La sección 2.3 describe los requerimientos no funcionales que debe cumplir el framework.

### 2.2. Funcionales

La siguiente tabla describe los requerimientos funcionales definidos en compañía del director del trabajo de grado para ser implementados por el framework ZMT.

| Código del Requerimiento | Nombre | Descripción |
|---|---|---|
| REQ-01 | Primitivas | El framework debe soportar el renderizado de diferentes primitivas graficas haciendo uso de gráficos vectoriales (líneas, rectángulos, óvalos, imágenes y texto). |
| REQ-02 | Elementos gráficos básicos | El framework debe soportar el manejo y manipulación de elementos básicos de grafos (nodos y aristas). |
| REQ-03 | Contenedores | El framework debe soportar que los nodos y las aristas puedan contener otros elementos gráficos recursivamente. |
| REQ-04 | Grupos de Nodos | El framework debe soportar el manejo de nodos que tengan la propiedad de realizar grupos de nodos. |
| REQ-05 | Transformaciones geométricas | El framework debe manejar transformaciones geométricas aplicadas a nodos y aristas. El cambio en las transformaciones geométricas debe ser aplicado como respuesta a cambios en el nivel de zoom. |
| REQ-06 | Diagramas y modelos | El framework debe permitir la visualización y manipulación de diagramas formados a en base a nodos y aristas. |
| REQ-07 | El nivel de detalle varía conforme al | El framework debe permitir mostrar u ocultar información con respecto al nivel de zoom. |

| | | |
|---|---|---|
| | nivel de zoom. | A mayor nivel de zoom, mayor información es mostrada en pantalla; a menor nivel de zoom, menor información es mostrada en pantalla. |
| REQ-08 | Técnicas de Visualización | El framework debe implementar técnicas de visualización de las interfaces aumentables. Estas técnicas son: zoom-in, zoom-out y panning. |
| REQ-09 | Árbol de navegación | El framework debe proveer un árbol de navegación en el cual será definido un elemento por cada nodo pintado en el lienzo. El árbol debe ser una representación jerárquica del modelo de grafos. |
| REQ-10 | Navegación del modelo | El framework debe soportar la navegación a través de:<br>a. Seleccionando un elemento desde el árbol de navegación.<br>b. Doble clic sobre un nodo para centrarlo y llevarlo a tamaño de toda la pantalla.<br>c. Botón para subir niveles en la jerarquía del árbol de navegación. |
| REQ-11 | Manejo de archivos JSON | El framework debe soportar el manejo de archivos JSON para cargar y renderizar la definición de los grafos que representan modelos de software. |

Tabla 2: Requerimientos Funcionales

### 2.3. No Funcionales

La siguiente tabla describe los requerimientos no funcionales definidos en compañía del director del trabajo de grado para ser soportados por el framework ZMT.

| Código del Requerimiento | Nombre | Descripción |
|---|---|---|
| REQ-12 | Enfocadoen la Web | El framework debe ser implementado con tecnología Web. Las herramientas construidas haciendo uso del framework serán aplicaciones basadas en la web y se requiere que el framework las soporte adecuadamente. |
| REQ-13 | Desempeño | El framework debe poder manejar diagra- |

| | | |
|---|---|---|
| | | mas pequeños y medianos sin ver por ello afectado su desempeño. |
| REQ-14 | Mantenibilidad | El framework debe implementar los principios de alta cohesión y bajo acoplamiento para asegurar mantenibilidad en lo referente a la adición de nuevas funcionalidades y modificación de las existentes. |
| REQ-15 | Fiabilidad | El framework debe ser fiable soportando la recuperación a errores, evitando a lo máximo que el usuario se vea obligado a reiniciar la aplicación para solucionarlos. |

Tabla 3: Requerimientos no Funcionales

### 2.4. Arquitectura de Software

La *Figura 6* describe la arquitectura general del framework, la cual comprende tres módulos principales:

- El módulo **ZUI Visualization** se encarga de renderizar en pantalla los grafos que representan modelos de software. Los modelos son definidos en archivos JSON[39]. Para la renderización de los grafos hace uso de la librería Data Driven Documents - D3.js [34], ésta librería permite dibujar los nodos usando gráficos SVG [40] y además provee el soporte de ZUI.

- El módulo **NavigationTree** encargado de crear el árbol de navegación con cada uno de los nodos que forman el modelo.

- El módulo **JSON Proccessing** se encarga de realizar el cargue y procesamiento de los modelos de sistemas software escritos en archivos en formato JSON. Éste módulo interpreta el archivo JSON y crea los objetos JavaScript necesarios para hacer visible el modelo en la pantalla.

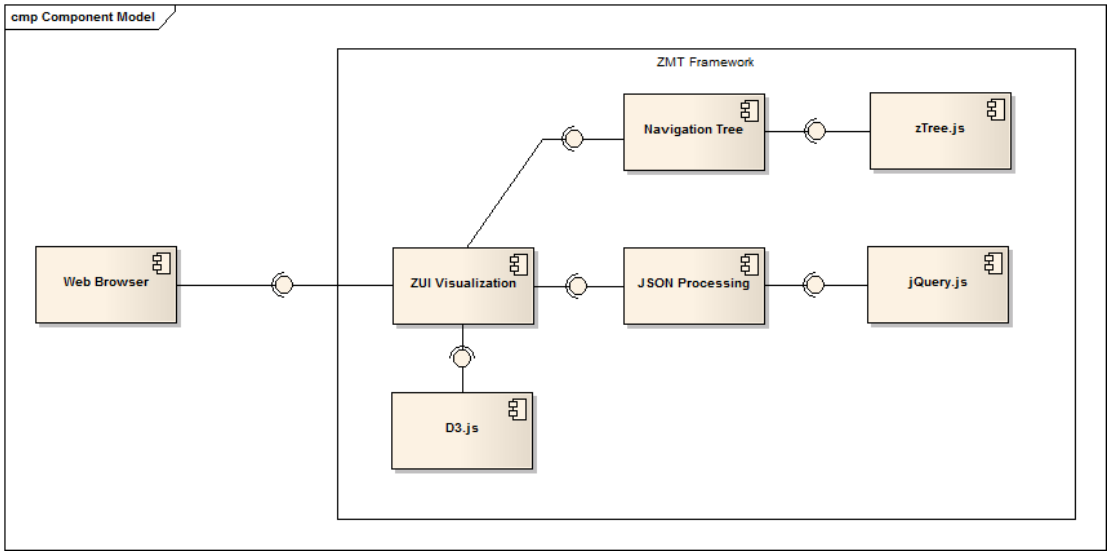


Figura 6: Arquitectura ZMT

El frameworkZMT está construido y soportado por las siguientes librerías: Data Driven Documents - D3 [34], jQuery [35] y zTree [41].

JQuery es una de las librerías JavaScript usadas por el framework ZMT. Esta librería es muy rica en funciones aunque en el frameworkZMT es usada únicamente como librería de soporte. jQuery es distribuido bajo licencia MIT [42].

ZTree es un plug-in de jQuery usado para el manejo del árbol de navegación de la herramienta ZMT. El plug-in es open source y distribuido bajo licencia MIT.

### 2.5. Diagrama de Clases

Las clases diseñadas para implementar la funcionalidad definida para el frameworkZMT son mostradas en el siguiente diagrama. El diseño ha sido pensado siguiendo los principios de bajo acoplamiento y alta cohesión del diseño de software facilitando la mantenibilidad del framework. En el diseño es visible el uso del patrón de diseño de software Decorador [43] el cual es implementado para permitir que cualquier elemento de tipo ***Node*** en el sistema pueda ser decorado con una o más diferentes características enriqueciéndolo con nueva funcionalidad en tiempo de ejecución.

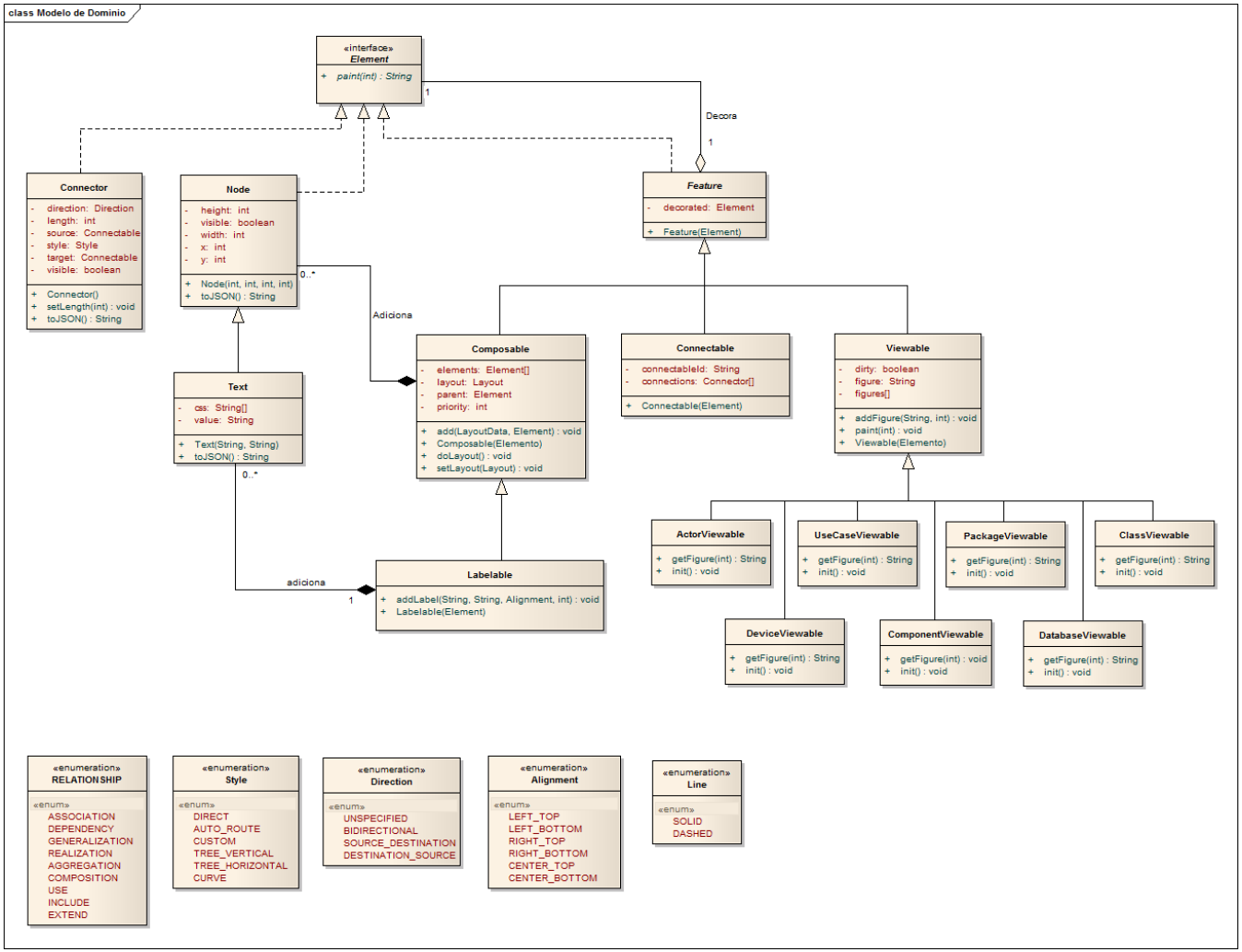


Figura 7: Diagrama de Clases

A continuación se describe cada uno de las clases definidas en el diagrama y se detalla la funcionalidad que cada uno de ellas cumple dentro del sistema.

- **Element**: interface raíz de la cual se desprende todo el diseño.
- **Node**: clase usada para definir un nodo básico en el sistema. La clase define las propiedades básicas de todo nodo, como son: la ubicación en el espacio, representado por los atributos x, y; las dimensiones que ocupa, representadas por los atributos width y height; y una propiedad para indicar si el nodo esta visible en el universo, representada por el atributo visible. Un ***Node*** por defecto no es un elemento gráfico, es decir, no tiene figura geométrica que lo represente visualmente. El ***Node*** puede tomar una forma geométrica y ser visible en pantalla por medio del decorador ***Viewable*** y de alguna de sus especializaciones.
- **Connector**: elemento grafico usado para representar las conexiones entre nodos. El Connector define propiedades para el manejo de: nodo fuente de la conexión y nodo destino de la conexión representadas por source y target, la dirección de la relación representada por el atributo direction, el estilo de la línea que forma la conexión representada por el atributo style.
- **Text**: clase usada para representar un tipo específico de ***Node***, el nodo de tipo texto. Este nodo es usado para representar cada uno de los elementos texto que son visualizados en el lienzo de la herramienta. La clase ***Text*** agrega propiedades value y css específicas para un texto. La propiedad value permite especificar el texto que será renderizado; la propiedad css, define el estilo definido en una hoja de estilos que aplica para el texto donde puede ser definidas el color, el tipo de letra, el tamaño de letra, etc.
- **Feature**: clase que permite decorar elementos ***Node*** y ***Connector*** con diferentes características. Esta clase permite implementar el patrón de diseño de software Decorador [43]. El patrón Decorador, permite agregar comportamiento a un objeto en tiempo de ejecución a diferencia de la herencia que lo hace en tiempo de compilación, logrando incluso ser un mecanismo más poderoso de agregar comportamiento. La clase Feature implementa diferentes especializaciones como son: ***Composable***, ***Connectable*** y ***Viewable***. Para dar soporte a nuevos decoradores en el sistema, basta con crear una clase que sea una nueva especialización de la clase Feature.
- **Composable**: especialización de la clase ***Feature*** que implementa la característica de elemento compuesto. Esta característica permite que el elemento decorado pueda agregar funcionalidades de composición, permitiéndole formar composición con otros nodos. Por medio de esta característica se logra la creación de nodos más grandes y complejos a partir de la composición de varios nodos básicos.
- **Labelable**: especialización de ***Composable*** que adhiere características para decorar un elemento como etiquetable. Al decorar un elemento con esta característica, es posible adicionarle etiquetas (labels) que serán dibujadas cerca al elemento decorado. Normalmente, los elementos decorados con esta característica son los elementos ***Connector.***

- **Connectable**: especialización de la clase ***Feature*** que implementa la característica de elemento conectable. Esta característica permite que el elemento decorado pueda agregar funcionalidades para conectarse y ser conectado por otros nodos. Un Node Connectable permite dibujar una conexión con otro Node Connectable.
- **Viewable**: especialización de la clase ***Feature*** que implementa la característica de elemento visible. Esta característica permite que el elemento decorado pueda tener una representación geométrica y ser visible en pantalla. Existen diferentes formas geométricas para representan un elemento en pantalla, son definidas como clases especializadas de Viewable. Para representar gráficamente un nuevo tipo de elemento en el framework, es necesario la creación de una nueva clase que herede de la clase ***Viewable*** y defina su propia representación geométrica.
  Un aspecto importante a mencionar en este punto, es que un elemento puede tener diferentes representaciones geométricas para diferentes niveles de detalle, es decir, para cierto nivel de zoom el elemento tiene una forma geométrica que lo representa de forma visual, en un nivel de zoom diferente el elemento cambia su representación geométrica haciendo que visualmente se vea diferente.
- **ActorViewable**: especialización de Viewable que define la figura geométrica para representar un Actor en el sistema.
- **UseCaseViewable**: especialización de Viewable que define la figura geométrica para representar un Caso de Uso en el sistema.
- **PackageViewable**: especialización de Viewable que define la figura geométrica para representar un Paquete en el sistema.
- **ClassViewable**: especialización de Viewable que define la figura geométrica para representar una Clase en el sistema.
- **DeviceViewable**: especialización de Viewable que define la figura geométrica para representar un Dispositivo de un diagrama de despliegue en el sistema.
- **ComponentViewable**: especialización de Viewable que define la figura geométrica para representar un Componente en el sistema.
- **DatabaseViewable**: especialización de Viewable que define la figura geométrica para representar la imagen de una Base de Datos en el sistema.

### 2.6. Visualización ZUI

La Visualización ZUI se encarga de pintar en pantalla los grafos que representan modelos de software, utilizando para ello la librería D3 para generar tres tipos principales de vistas: la vista de nodos, la vista de conexiones y la vista de texto.

Las vistas de nodos despliegan la información de las instancias de ***Node*** *(ver*Figura **7***)*. Una vista de nodo puede asumir diferentes formas geométricas y, puede contener otras vistas dentro de ella. Las formas geométricas están soportadas por el uso del patrón de diseño de software Decorador [43]. Mediante el uso de este patrón, es posible instanciar un elemento ***Node*** y decorarlo con el tipo ***Viewable*** requerido, dándole una forma geométrica especifica según el tipo de decorador seleccionado.

Las formas geométricas disponibles para decorar un nodo son: actor, caso de uso, paquete, clase, dispositivo, componente y base de datos.

Las vistas de conexiones despliegan la información de ***Connector*** a través de líneas que asocian nodos en un grafo. Estas vistas pueden contener otros elementos en su interior, tales como texto, lo que permite representar nombres de asociaciones, roles y cardinalidades, entre otras cosas. Los extremos de las líneas pueden desplegar diferentes terminaciones como flechas, rombos, etc., para representar diferentes tipos de relación como asociación, herencia, implementación, composición, agregación, etc.

Las vistas de texto despliegan la información de ***Text*** que es un tipo específico de ***Node***. Por medio de estas vistas es posible representar textos para nodos y conexiones. Los elementos texto son por ejemplo: el nombre de la clase, atributos de la clase, operaciones de la clase, el nombre de la asociación, nombre de rol, cardinalidad, etc.

A partir de estas tres vistas, el frameworkZMT puede renderizar los siguientes diagramas: diagrama de casos de uso, diagrama de comunicación, diagrama de despliegue, diagrama de componentes, diagrama de paquetes y diagrama de clases.

### 2.7. Procesamiento JSON

El módulo de Procesamiento JSON se encarga de realizar el cargue y procesamiento de los archivos JSON donde son definidos los grafos para representar los modelos de software. El procesamiento consiste en crear los objetos JavaScript necesarios para representar de forma gráfica los modelos de software.

La estructura JSON para la definición de un ***Node*** es la siguiente:


Figura 8: JSON para Node

A partir de esta estructura se puede definir un nodo, decorarlo con alguno de los decoradores disponibles y componerlo mediante la agregación de otros nodos. El nodo tiene atributos para la posición en pantalla, las dimensiones de alto y ancho, las características y la composición con otros nodos.La estructura JSON para la definición de una conexión entre dos nodos es la siguiente:


Figura 9: JSON para Connector

La estructura anterior define atributos para el nodo fuente, el nodo destino, el estilo de la relación, la dirección de la línea, el tipo de relación representada y las etiquetas para el nombre y cardinalidad de la relación. La sección de labels puede contener 0 o más elementos de tipo texto. Cada elemento define el texto a mostrar sobre la conexión, el css de la hoja de estilos que se le debe aplicar, la alineación del texto y la prioridad del texto en la conexión. El atributo prioridad define el orden de importancia del elemento sobre la conexión y, es usado para ocultar el elemento de menor prioridad cuando la conexión tiene una longitud corta. La longitud de la conexión cambia en base al nivel de zoom aplicado.

### 2.8. Soporte de ZUI

El frameworkZMT soporta el paradigma ZUI a través de dos características. Primero, las instancias de ***Node*** y ***Connector***pueden ser decorados con la característica ***Composable*** formando una jerarquía de composición *(ver*Figura **7***).* Esta característica permite que tanto nodos y conexiones puedan estar formados por la composición de otros nodos. De esta forma, se puede definir un nodo decorado como paquete compuesto de varios nodos decorados como clase; pero también; un nodo decorado como clase compuesto de varios nodos decorados como texto, donde cada texto dentro de la clase correspondiente al título, atributos y operaciones.

Segundo, el elemento ***Node*** incluye capacidades de zoom semántico a través de la decoración con la característica ***Viewable****(verFigura 7).* El elemento ***Viewable*** define un arreglo de figuras para las diferentes dimensiones del nodo las cuales son alteradas por el nivel de zoom aplicado al lienzo. Cada vez en la que ocurre un cambio en el nivel de zoom, el nodo selecciona de su arreglo de figuras aquella que aplica para el nuevo nivel de zoom alcanzado. En cuyo caso de que para el nivel de zoom el nodo no tenga definida una figura, el nodo será ocultado de forma automática y, solo volverá a estar visible cuando regrese a un nivel de zoom para el que tenga una figura definida.

El arreglo de figuras para un elemento ***Node*** decorado como ***Clase*** es mostrado en la siguiente figura:

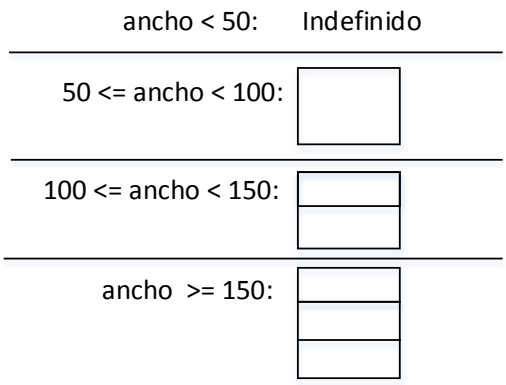


Figura 10: Figuras Nodo Clase

El arreglo para el elemento ***Clase*** contiene 3 figuras diferentes. Cada figura está asociada a un rango de valores de ancho específico. El rango de valores de ancho menor de 50 no cuenta con una figura definida, por lo tanto, cuando el elemento clase se encuentra entre estas dimensiones no será pintado en el lienzo; para un ancho entre 50 y 100, la figura para pintar el nodo clase es un rectángulo con un solo contenedor (titulo); para un ancho entre 100 y 150, la figura es un rectángulo con dos contenedores (título y atributos) y para un ancho mayor de 150 la figura es un rectángulo con tres contenedores (título, atributos y métodos).

Siempre que se cambia el nivel de zoom del lienzo de dibujo, las dimensiones de cada nodo también cambian. Tanto el ancho como el alto son multiplicados por el valor actual de zoom. Para entender el comportamiento se ofrece el siguiente ejemplo: el lienzo se encuentra a nivel de zoom=1, tiene dibujado un nodo con dimensiones de ancho=50 y alto=80; al realizar la operación de zoom-in y alcanzar el nivel de zoom=2, las dimensiones del nodo serán: ancho=100 y alto=160; al alcanzar el nivel de zoom=3, las dimensiones del nodo serán: ancho=150 y alto=240; en caso de aplicar la operación contraria, el zoom-out y alcanzar el nivel de zoom=0.5, las dimensiones del nodo serán: ancho=25 y alto=40.

El algoritmo implementado para la selección de figuras funciona de la siguiente manera:

1. El lienzo notifica al nodo el cambio en el nivel de zoom
2. El nodo actualiza sus dimensiones de ancho y alto según el nivel de zoom aplicado al lienzo
3. El nodo recorre el arreglo de figuras en busca de la figura que aplica para su nuevo valor de la dimensión ancho.
4. El nodo actualiza su representación gráfica en el lienzo con la figura correspondiente. Dado el caso de que la figura encontrada sea *indefinida*, el nodo será ocultado del lienzo automáticamente.

### 2.9. Diagrama de Despliegue

Como se ha descrito en las secciones anteriores, el diseño e implementación del framework ZMT ha sido pensado para ser un framework Web, éste fue uno de los requerimientos principales al inicio del proyecto. Por esta razón, el despliegue de los componentes construidos y librerías que soportan el framework ZMT, requieren ser instalados en un servidor web con soporte para interpretar HTML, JavaScript y CSS.

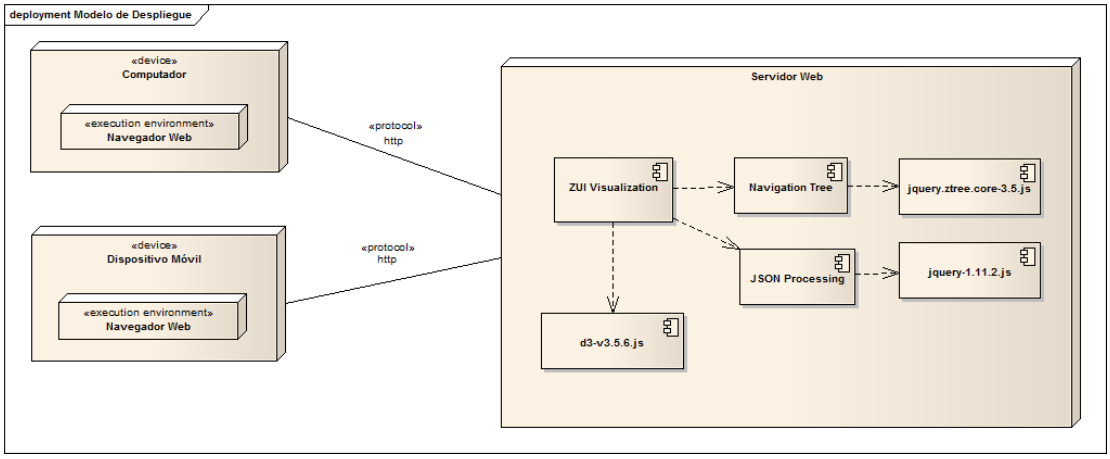


Figura 11: Diagrama de Despliegue

El componente de Visualización ZUI, es la puerta de entrada a los servicios proveídos por el framework ZMT. El componente Visualización ZUI hace uso del componente Procesamiento JSON para cargar y procesar los archivos JSON que definen los modelos de software; y hace uso de las librerías D3, jQuery y zTree para renderizar en el lienzo los modelos y proveer las características de ZUI requeridas por el framework.

En el diagrama se aprecian dos tipos de dispositivo: Computador y Dispositivo Móvil, ambos mediante el uso de un navegador Web y a través del protocolo http pueden acceder al componente de Visualización ZUI.

## 2.10. Prototipo construido con ZMT

Mediante el uso del framework ZMT, se ha construido un prototipo de herramienta Web para la visualización de grafos. La herramienta construida como prototipo tiene una pantalla formada por dos elementos principales: el lienzo para dibujo y el árbol de navegación. El lienzo es el elemento principal y ocupa la mayor cantidad de espacio disponible en pantalla, en él son dibujados los grafos para representar los modelos de software. El árbol de navegación, provee al usuario una vista de árbol de los grafos dibujados en el lienzo, permite además, la navegación a través del modelo por medio de la selección de los nodos definidos en él.

La siguiente imagen muestra la pantalla del prototipo ZMT. En la parte izquierda de la imagen, se visualiza el árbol de navegación; en la parte central, se visualiza el lienzo de dibujo donde se muestra parte de un diagrama de casos de uso y un diagrama de despliegue para un sistema ATM.

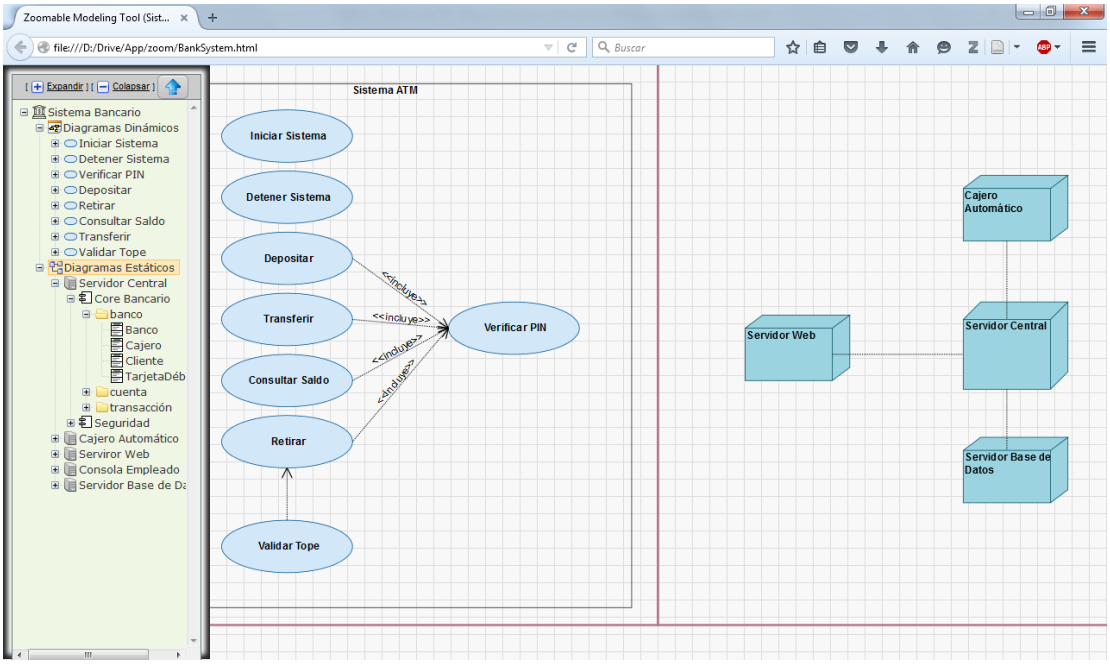


Figura 12: Prototipo ZMT

## 2.11. Características implementadas

El prototipo construido con el uso del framework ZMT ha implementado varias funcionalidades enfocadas todas en visualización de grafos que representan modelos de software. Este prototipo ha sido construido con el objetivo de dar respuesta a la pregunta de investigación que dio origen al desarrollo del presente proyecto. La pregunta de investigación será resuelta a través del diseño y ejecución de un experimento, en el cual un conjunto de personas harán uso del prototipo construido para visualizar modelos de software y responder preguntas sobre ellos.

Las características implementadas por el prototipo son:

**Zoom-In:**permite que el usuario pueda realizar acercamiento al modelo logrando adentrarse a un nivel más detallado de la información. El acercamiento consiste en visualizar el contenido de un nodo, por ejemplo, observar el contenido dentro de un paquete que pueden ser otros elementos de tipo paquete o tipo clase. El efecto de acercamiento se logra aumentando el tamaño de la figura que representa el nodo, a medida que el tamaño aumenta y alcanza espacio suficiente son visualizados los elementos contenidos por el nodo. El zoom-in es realizado por medio de la rueda de desplazamiento del mouse.

**Zoom-Out:**permite que el usuario se aleje del modelo subiendo en el nivel de abstracción. Al alejarse, el contenido de un nodo empieza a ocultarse basado en su nivel de importancia hasta que el nodo alcanza un tamaño suficientemente pequeño para el nodo mismo ser ocultado. El efecto de distanciamiento se logra disminuyendo el tamaño de la figura que representa el nodo, a medida que el tamaño disminuye

se van ocultado los elementos contenidos por el nodo. El zoom-out es realizado por medio de la rueda de desplazamiento del mouse.

**Panning:** permite que el usuario pueda desplazarse por el lienzo llegando a partes lejanas del modelo visualizándolo siempre a un mismo nivel de abstracción. Para realizar panning, el usuario debe mantener presionado el botón principal del mouse y desplazar el puntero sobre el lienzo en la dirección deseada.

**Doble clic sobre nodos:** permite al usuario llevar el elemento seleccionado a tamaño de pantalla completa. Al realizar esta operación, la herramienta le aplica al elemento operaciones de zoom-in de forma secuencial hasta que alcanza el nivel de zoom necesario para llevarlo a pantalla completa.

**Seleccionar Nodo en Árbol:** la herramienta dispone del árbol de navegación el cual define de forma jerárquica cada uno de los nodos que componen el modelo del sistema software dibujado. Cuando el usuario selecciona un nodo del árbol de navegación, la herramienta ZMT se encarga de centrarlo y llevarlo a tamaño de pantalla completa. El efecto de seleccionar un nodo del árbol es igual al efecto generado al realizar doble clic sobre un nodo dibujado en el lienzo.

**Expandir Nodos del Árbol:** permite al usuario mediante el clic en el botón *"Expandir"* ubicado en la parte superior del árbol de navegación, abrir de forma simultánea todos los nodos padre del árbol. Permitiéndole observar todos los nodos de forma simultánea y, además, en caso de estar buscando un nodo específico, encontrarlo de manera fácil y rápida.

**Colapsar Nodos del Árbol:** permite al usuario mediante el clic en el botón *"Colapsar"* ubicado en la parte superior del árbol de navegación, cerrar todos los nodos padre dejando visible únicamente los nodos raíz.

**Subir Nivel en Árbol:** en la parte superior del árbol de navegación se encuentra un botón tipo flecha arriba de nombre *"Subir un Nivel".* Éste botón es usado para alejarse del modelo subiendo un nivel de abstracción en la visualización. Un ejemplo de uso es por ejemplo: el usuario se encuentra visualizando un diagrama de clases, cuando presiona el botón *"Subir un Nivel"* la herramienta realiza la operación de zoom-out alejándose del modelo hasta llegar al nodo padre el cual en el ejemplo sería un diagrama de paquetes.

# IV - EXPERIMENTO

Esta sección describe el proceso de las fases de diseño y ejecución del experimento que fue realizado para responder la pregunta que dio origen al presente trabajo de investigación.

La presente sección se divide en 2 partes principales: la primera parte describe proceso de diseño del experimento lo que incluye la definición del protocolo, el perfil de los participantes, la cantidad de muestras, etc. La segunda parte, describe el proceso realizado durante la ejecución y generación de resultados del experimento.

# 1. DISEÑO

Antes de entrar en el detalle del diseño, es necesario definir que es un experimento: *"Un experimento es un cambio en las condiciones de operación de un sistema o proceso, que se hace con el objetivo de medir el efecto del cambio sobre una o varias propiedades del producto o resultado"*[18].

Con base en esta definición, el objetivo del experimento que se plantea en este trabajo, es observar los efectos que genera el uso del paradigma de visualización ZUI en la visualización y navegación de grafos que representan modelos de software.

En este experimento, la condición de operación que se ha cambiado, es el paradigma de visualización; el sistema, es la herramienta para la visualización de modelos de software; y los efectos que se quieren medir, son el tiempo y la precisión con la cual el usuario encuentra respuestas en el modelo.

A continuación se describen las hipótesis planteadas, las medidas de éxito mínimas requeridas, la viabilidad de ejecutar el experimento tomando en cuenta recursos y condiciones, el protocolo definido y los escenarios de prueba que serán ejecutados para realizar las mediciones.

## 1.1. Hipótesis

Las hipótesis planteadas sobre el uso de ZUI en la herramienta ZMT son las siguientes:

- **a.** La implementación de ZUI permite al usuario describir los componentes que forman el modelo y sus relaciones de forma más rápida comparada con herramientas que usan técnicas de visualización tradicional.
- **b.** La implementación de ZUI permite al usuario encontrar respuestas en el modelo de forma más precisa y en un menor tiempo comparado con herramientas que usan técnicas de visualización tradicional.

## 1.2. Medidas de Éxito

La medida de éxito hace referencia al nivel de precisión estadística con el cual se está dispuesto a analizar los resultados obtenidos. Para entender esta medida de éxito se debe primero definir algunos conceptos básicos de las hipótesis estadísticas.

*"Probar una hipótesis estadística es una decisión probabilística, por lo que existe el riesgo de cometer un error tipo I o un error tipo II. El error tipo I ocurre cuando se rechaza la hipótesis nula (H0) siendo verdadera, y el error tipo II es cuando se acepta la hipótesis nula (H0) siendo falsa"*[18].

La probabilidad de cometer el error tipo I se denota con la letra α y la probabilidad de cometer el error tipo II se denota con la letra ß, como se muestra a continuación:

α = P{error tipo I} = probabilidad de rechazar H0 siendo verdadera

ß = P{error tipo II} = probabilidad de aceptar H0 siendo falsa

El error tipo I es controlado de forma directa por el investigador y es definido desde el inicio de la investigación. El valor dado a α depende del riesgo que se quiera admitir en la conclusión, normalmente se utilizan los valores α = 5% o 1% (α = 0.05 o 0.01). *"El valor de α = 0.05 significa que por cada 100 veces independientes que se aplique el procedimiento y se rechaza H0, se espera que en un promedio de 95 veces, tal decisión sea la correcta"*[18].

El error tipo II es más difícil de controlar porque depende del tamaño de la muestra y de la cantidad de veces que se repita el experimento. Entre mayor sea la muestra y la cantidad de repeticiones, el valor de ß será menor.

Teniendo claras las definiciones anteriores, la medida de éxito predefinida para el análisis de los resultados del presente experimento es del 95%, es decir, se manejará un α = 0.05 (5%). Este valor permitirá aceptar con un grado alto de precisión los resultados obtenidos como verdaderos.

## 1.3. Viabilidad

Esta sección describe la viabilidad para la ejecución del experimento. La viabilidad fue analizada desde la perspectiva de los recursos necesarios (tomando en cuenta el tiempo, las personas participantes, equipos de cómputo, etc.) y los factores ruido que podían afectar una obtención precisa de resultados.

### 1.3.1. Recursos

En este punto se realiza una descripción de los recursos utilizados durante la ejecución del experimento.

- **Tiempo Limitado**. El experimento fue realizado con personas quienes no tienen todo el tiempo disponible para dedicarse al experimento. El tiempo requerido para la realización del experimento con cada participante estuvo alrededor de los 25 minutos.
- **Capital Humano**. Los participantes del experimento fueron profesionales de Ingeniería de Sistemas con conocimientos (básicos o avanzados) en UML.El grado de conocimiento de UML, permitió clasificarlos en dos grupos: novatos y expertos.En total fueron 16 participantes, algunos con perfil de desarrollar de software y otros de líder técnico de desarrollo. Los participantes fueron seleccionados de dos empresas de desarrollo, de Imix Consulting 10 participantes y de Hydra Management 6 participantes, ambas empresas con sede en Bogotá.
- **Equipo de Cómputo**. Durante el experimento se usó el mismo equipo de cómputo para cada participante. La razón fue a causa de dos razones principales: primera, solo se dispuso de una única persona que actuó como observador; segundo, para garantizar que todos los participantes cuenten con similares condiciones durante la ejecución.

- **Internet**. La herramienta construida se encuentra alojada en los servidores de Google Drive y su forma de acceso es a través de internet. Se verificó que el sitio ofrecido por cada compañía para la ejecución del experimento contara con acceso a internet de forma ininterrumpida.
- **NavegadorWeb**. El experimento fue realizado exclusivamente sobre el navegador Firefox en la versión 42.0. De esta forma se evitó el ruido que puede introducir el uso de diferentes navegadores y versiones. Cabe aclarar que la herramienta construida esta soportada para su uso en los navegadores Firefox versión 39 o superior y Chrome versión 44 o superior.
- **Softwarepara Captura de Pantalla.**Fue necesario realizar la captura de la pantalla del computador usado en el experimento para registrar todas las acciones realizadas por el participante sobre la herramienta evaluada; posteriormente estas grabaciones fueron analizadas para la generación de resultados y conclusiones. El software usado para este proceso fue HyperCam 2, creado por la compañía Hyperionics Technology[44].
- **Disco Duro.** Las capturas de pantalla realizadas requieren de espacio en disco duro suficiente para ser almacenadas. El computador usado durante la ejecución del experimento contaba con amplia capacidad disponible de almacenamiento.
- **Cronómetro.** Durante la ejecución del experimento se midió el tiempo usado por cada participante en la realización de cada una de las tareas que le fueron asignadas como parte del experimento. Para tomar los tiempos se usó un iPad que tenía instalado el software Stopwatch creada por Tim O's Studios[45].
- **Libreta de Apuntes.** El observador durante la ejecución del experimento por parte de cada uno de los participantes, estuvo pendiente de tomar apuntes de todos aquellos comportamientos u acciones de los participantes durante el uso de la herramienta. Estas notas fueron usadas durante el análisis de resultados.

### 1.3.2. Ruido

Es este punto se realiza una descripción de todos aquellos factores identificados que podían incluir un ruido durante la ejecución del experimento, y por tanto afectando los resultados obtenidos.

- El observador que hace anotaciones durante el experimento podía llegar a generar cierto sentido de alerta en los participantes, llevándolos quizás a comportarse diferente al sentirse observados y/o analizados. Para evitar o mitigar esta predisposición en los participantes, durante el recibimiento e incluso en la carta de Consentimiento Informado, se les hizo total claridad de que en ningún momento él sería-

caso de evaluación, la única evaluada en el experimento es la herramienta como tal.

- Selección incorrecta de la muestra. Personas participantes del experimento sin experiencia en UML y/o el modelado de software lo cual sea un impedimento para desarrollar de forma efectiva las tareas planeadas en el experimento. Para mitigar este factor, antes del experimento se realizó una prueba de competencias que buscaba medir el conocimiento del participante en los elementos básicos de UML. De los 16 participantes que realizaron el experimento, todos obtuvieron un buen puntaje en la prueba de competencias, lo que permitió corroborar que mínimamente contaban con conocimientos básicos en UML.
- Tamaño de la muestra. Los participantes del experimento pueden ser una muestra insuficiente para representar todos los puntos de vista de la población.
- Los participantes podían llegar a sentirse presionados por responder de forma positiva frente a la experiencia en el uso de la herramienta. Podían tener una experiencia quizás no tan buena e incluso votar o dar una opinión favorable por no sentirse incomodos o hacer sentir incomodo al desarrollador de la herramienta. Para mitigar este factor, de igual forma se le comunicó a cada participante que se sintiera libre de valorar el uso de la herramienta según como a él personalmente le hubiese parecido. La valoración por él dada a la herramienta no genera afectación negativa en los resultados, por el contrario, ayuda a mejorar la herramienta.

### 1.4. Técnicas de Observación y Seguimiento

Durante la ejecución del experimento se realizóla grabación de la pantalla previamente autorizada por el participante a través de consentimiento informado.La grabación fue realizada con el fin de ser reproducida en un momento posterior y extraer información adicional no vista por el observador durante la ejecución en vivo.

Se contó con una sola persona que cumplió el rol de observador. Esta persona estuvo a cargo de tomar notas y resolver dudas del participante respecto al experimento y las tareas asignadas. El observador tuvo prohibidoresolver dudas respecto al uso de la herramienta o la solución de las tareas.

### 1.5. Lugar de Testeo

El lugar de testeo en ambas compañías fue la sala de juntas, la cual tenía buena ventilación, iluminación y privacidad evitando interrupciones externas durante el experimento. En la sala de juntas durante la ejecución del experi-

mento solo estuvo el participante en compañía del observador, no se permitió la presencia de personas adicionales para evitar factores ruidos adicionales.

## 1.6. Diseño del Experimento

En este punto se describe el diseño realizado para el experimento, lo que incluye: factores estudiados, variables de respuesta, factores de control, escenarios de prueba y el protocolo.

### 1.6.1. Factores Estudiados

También conocidas como variables de entrada. Son las variables que se investigan en el experimento y de las cuales se quiere observar su efecto sobre las variables de respuesta. Las variables de interés en el experimento son:

- **a.** Uso de interfaz de usuario aumentable.
- **b.** Uso de interfaz de usuario tradicional.

### 1.6.2. Variables de Respuesta

También conocidas como variables de salida. A partir de estas variables se conoce el efecto que genera cada prueba experimental diseñada. Un cambio en las variables de entrada afecta directamente el valor de las variables de salida. Las variables de salida para el experimento son:

- **a.** Tiempo requerido.
- **b.** Exactitud de la respuesta.

### 1.6.3. Factores de Control

Son variables que no interesa estudiar en el experimento, deben ser controladas para evitar la generación de ruido en los resultados obtenidos de las variables de salida. Los factores de control definidos para el experimento son:

- **a.** En ambas herramientas (con visualización ZUI y con visualización tradicional) fueron dibujados los mismos modelos.
- **b.** La organización y distribución de los elementos dentro de cada diagrama es la misma en ambas herramientas.
- **c.** El layout, el árbol de navegación, el área de dibujo, las formas y colores de cada elemento (paquete, clase, componente, dispositivo, caso de uso, relaciones, etc.) en ambas herramientas es exactamente igual.
- **d.** El cronometro para contabilizar el tiempo requerido en la ejecución de cada tarea es manejado por el observador.
- **e.** El observador no habla con el participante durante la ejecución de las tareas. No responde preguntas referentes al uso de la herramienta. No hace gestos que le puedan indicar al usuario que está resolviendo

correcta o incorrectamente la tarea. Solo habla con el participante para resolver dudas referentes al entendimiento de la tarea pero no sobre la manera derealizarla.

### 1.6.4. Criterios de Evaluación

La siguiente tabla describe los criterios seleccionados para la evaluación de la herramienta *Zoomable Modeling Tool - ZMT*. Cada criterio seleccionado define su significado y forma de medirlo. A partir de estos criterios, se diseñaron los diferentes escenarios de prueba que cada participante completo durante la ejecución del experimento. Estos criterios están basados en los criterios recomendados para la evaluación de una herramienta CASE [46], [47].

| Criterio | Significado | Forma de Medirlo |
|---|---|---|
| Facilidad de Uso, Curva de Aprendizaje [46], [47][48], [49] | La facilidad de aprendizaje se refiere a la velocidad con la cual un usuario que usa por primera vez la herramienta logra identificar y usar cada una de las funcionalidades provistas por la herramienta. | Tiempo que demora el usuario en aprender a usar las funcionalidades de la herramienta. |
| Eficiencia [48], [49] | El tiempo consumido con relación a la certeza con la cual los usuarios logran los objetivos. | ¿Cuánto tiempo le toma a un usuario completar un grupo de tareas específicas? |
| Efectividad [48], [49] | Representa la exactitud con la cual los usuarios alcanzan los objetivos especificados. | La tarea se logró hacer:<br>• Completamente.<br>• De forma parcial.<br>• No se logró. |
| Tasa de Error [48], [49] | Se refiere a los errores que comete el usuario durante el uso de la herramienta y que afectan su productividad. | ¿Cuántos y qué errores hace la gente al ejecutar un grupo de tareas específicas? |
| Sobrecarga Visual [47] | Se refiere a la cantidad de información que es mostrada al usuario en un determinado momento. La cantidad de información mostrada puede afectar la eficiencia del usuario al entender el diagrama. | Contar para cada nivel diferente de zoom la cantidad de elementos que muestra la herramienta con zoom vs la herramienta tradicional. |

| | | |
|---|---|---|
| Satisfacción [49] | ¿Qué tanto le gustó a los usuarios los distintos atributos del sistema? | Cuestionario para medir la satisfacción del usuario en la interacción con la herramienta (tipo QUIS, SUMI, SUS, etc.) |
| Utilidad [48] | ¿El diseño provee las características que el usuario necesita? | Cuestionario para medir la satisfacción del usuario en la interacción con la herramienta (tipo QUIS, SUMI, SUS, etc.) |
| Consistencia [46] | Es la capacidad de utilizar de la misma manera todos los mecanismos, sea cualquiera el momento que se necesite. | Se puede medir usando una máquina de estados de los diferentes elementos. El resultado de una acción debe ser el mismo en todo momento y condición del sistema. |

Tabla 4: Criterios de Evaluación

### 1.6.5. Escenarios

A partir de los criterios definidos en el punto anterior, fueron diseñados los escenarios de prueba que cada participante debió completar durante la ejecución del experimento. Por cada escenario definido, se indica las mediciones que fueron realizadas para la generación de resultados. A continuación, se describe cada uno de los escenarios definidos:

1. Ingresar al modelo de entrenamiento e inspeccionar la herramienta libremente con el objetivo de familiarizarse con su forma de uso y las diferentes funcionalidades que ofrece. El tiempo de esta tarea será cronometrado iniciando desde el momento en que se accede a la URL y finalizando en el momento en que el usuario levanta la mano notificando sentirse preparado para afrontar las tareas diseñadas para el experimento.

   **Medidas**:

   - Tiempo requerido para completar la tarea.

2. Navegar a través del modelo, observar cada uno de sus módulos, las clases que lo forman, atributos, operaciones y las relaciones entre ellas. El objetivo es recordar la mayor cantidad de información posible del modelo. Cuando sienta que ha observado el modelo lo suficiente hágaselo saber al observador. El observador le pedirá ubicar un ele-

mento dentro del modelo. El insumo para la búsqueda será: el nombre y una breve descripción del elemento.
**Medidas:**

- Tiempo requerido para completar la tarea por cada elemento.
- ¿Encontró el elemento?

3. Identifique el paquete o sub-paquete con mayor cantidad de conexiones entre sus elementos directamente contenidos.
   **Medidas**
   - Tiempo requerido para completar la tarea.
   - Exactitud de la respuesta.

4. Identifique el paquete con mayor cantidad de elementos tipo nodo (clases, interfaces, sub-paquetes).
   **Medidas**
   - Tiempo requerido para completar la tarea.
   - Exactitud de la respuesta.

5. Nombre las clases relacionadas de forma directa con la clase A.
   **Medidas**
   - Tiempo requerido para completar la tarea.
   - Exactitud de la respuesta.

6. Encuentre la clase con la mayor cantidad de métodos del sistema.
   **Medidas**
   - Tiempo requerido para completar la tarea.
   - Exactitud de la respuesta.

7. Nombre los casos de uso en los que para su realización participan objetos de la clase **A.**
   **Medidas**
   - Tiempo requerido para completar la tarea.
   - Exactitud de la respuesta.

### 1.6.6. Protocolo

Este punto define el protocolo diseñado para la ejecución del experimento. El protocolo describe de forma clara el paso a paso que se debe seguir desde el momento mismo en que llega el participante al sitio adecuado para el experimento, hasta el momento que finaliza el último de los formularios y abandona el sitio. El observador debe seguir al pie de la letra lo definido en el protocolo, de esta forma, se garantiza que el experimento se pueda repetir con varios participantes en diferentes momentos y siempre bajo similares condiciones.

El protocolo definido para el experimento contempla los siguientes puntos:

- **a.** El observador recibe al participante en el salón adecuado para la actividad dándole la bienvenida y los agradecimientos por su buena voluntad en la participación del experimento.
- **b.** El observador da una breve charla de inducción al participante para explicarle el proceso que se seguirá durante el transcurso del experimento. Le cuenta que el tiempo estimado para la ejecución del experimento es aproximadamente 25 minutos. Le informa que durante este tiempo recibirá la carta de consentimiento informado y tres formularios electrónicos, el primero de clasificación, el segundo de tareas y el tercero una encuesta de satisfacción. Le cuenta que durante el desarrollo del formulario de tareas se usará un software de grabación de pantalla para registrar las acciones realizadas sobre la herramienta.
- **c.** El observador entrega al participante dos copias de la carta de Consentimiento Informado, una copia es conservada por el participante y la segunda la regresa firmada. En esta carta se explica el objetivo del experimento y se deja por escrito la autorización explícita por parte del participante de la captura de pantalla del computador, de su aceptación voluntaria en participar en el experimento y la aclaración de que la única evaluada en el experimento es la herramienta y no su desempeño como participante en la ejecución de las tareas.
- **d.** El observador hace entrega al participante del formulario de competencias. Este formulario contiene preguntas relacionadas con la sintaxis y elementos que hacen parte del Lenguaje Unificado de Modelado – UML. El objetivo del formulario, es clasificar al participante en uno de dos grupos: Novato o Experto, según su conocimiento y nivel de experiencia en diagramas UML. Esta clasificación no tiene implicación en las tareas ni en su nivel de complejidad. Todos los participantes novatos y expertos realizarán las mismas tareas bajo condiciones idénticas.
- **e.** El observador cuenta con tres formularios de tareas todos con preguntas idénticas pero enfocadas a modelos de diferentes sistemas software (todos de similar complejidad). El observador entrega solo uno de los tres formularios al participante. Para determinar cuál formulario entregar, el observador hace uso de un generador de números aleatorios que implementa el método de distribución uniforme. Esta distribución es de carácter continuo y asigna la misma probabilidad de selección para los tres modelos disponibles.
- **f.** El observador brinda las instrucciones finales al participante, inicia el software de grabación de pantalla y le indica al participante que puede arrancar a resolver las tareas definidas en el formulario previamente entregado.

**g.** Al finalizar las tareas propuestas, el observador detiene el software de grabación de pantalla.
**h.** El observador hace entrega al participante del Formulario de Satisfacción. El objetivo de éste último formulario es obtener las impresiones del participante sobre el uso de la herramienta.
**i.** Finaliza el ejercicio y se recoge todo el material desarrollado para su posterior análisis y generación de las conclusiones.

# 2. EJECUCIÓN

Esta sección describe todo el proceso realizado durante la ejecución del experimento. Se describe el proceso desde la asignación de los participantes a los diferentes grupos, la asignación de los modelos software sobre los que cada participante debe trabajar, los diferentes formularios entregados al participante (consentimiento informado, prueba de competencias, formulario de tareas y encuesta de satisfacción).

Todos los formularios exceptuando el consentimiento informado (por motivos de la firma) han sido diseñados como formularios electrónicos de Google Forms [50].

## 2.1. Asignación de Grupos

Para la ejecución del experimento, los 16 participantes fueron divididos en 3 grupos (la asignación del participante a cada grupo fue realizada por medio de un generador de números pseudo-aleatorios uniformes [51]). El primer grupo, encargado de ejecutar las tareas en la herramienta de visualización tradicional; el segundo grupo, encargado de ejecutar las tareas en la herramienta de visualización ZUI; y el tercer grupo, encargado de ejecutar las tareas haciendo uso de las dos herramientas (tradicional y ZUI).

Los tres grupos quedaron formados de la siguiente manera:

- Grupo 1 – Herramienta de visualización tradicional: 5 participantes
- Grupo 2 – Herramienta de visualización ZUI: 6 participantes
- Grupo 3 – Herramienta de visualización tradicional y ZUI: 5 participantes

Los grupos 1 y 2, recibieron un formulario de tareas para ser desarrollado en la herramienta de visualización tradicional y ZUI respectivamente.

El grupo 3, recibió dos formularios cada uno de un sistema software diferente. En primer lugar, realizaron las tareas de la herramienta de visualización tradicional y a continuación, realizaron las tareas en la herramienta de visualización ZUI.

Los grupos 1 y 2, realizaron una sola ejecución del experimento; el grupo 3, realizó dos ejecuciones del experimento. La segunda ejecución fue realizada pasados 15 días a partir de la primera ejecución. El grupo 3 usa ambas herramientas como mecanismo que permite comparar de manera más precisa las medidas obtenidas, donde lo único que cambia entre una ejecución y otra es la herramienta, el participante sigue siendo el mismo. La repetición 15 días des-

pués, se realiza como mecanismo que permite corroborar los resultados obtenidos durante la primera ejecución.

### 2.2. Asignación de Modelos

Para el experimento fueron modelados tres sistemas de software, todos de similar complejidad. El modelo de cada sistema software fue dibujado de forma idéntica en la herramienta de visualización tradicional y en la herramienta de visualización ZUI.

Los tres sistemas software modelados son:

- Sistema de Gestión de Proyectos
- Sistema Bancario
- Sistema de una Tienda Deportiva

Ahora que ya se tenían los participantes distribuidos en los tres grupos, era el momento de asignarles los modelos de los sistemas software sobre los cuales realizarían las tareas.

A cada participante del grupo 1, se le asignó un modelo dibujado en la herramienta tradicional; a cada participante del grupo 2, se le asignó un modelo dibujado en la herramienta ZUI; a cada participante del grupo 3 se le asignó un modelo dibujado en la herramienta tradicional y un modelo de un sistema diferente dibujado en la herramienta ZUI.

La asignación de los modelos sobre los que debía trabajar cada participante, fue realizada a través del uso del generador de números aleatorios uniformes, de forma similar a como fue realizada la asignación de los participantes a cada uno de los diferentes grupos.

### 2.3. Consentimiento Informado

Elconsentimiento informado es un documento que garantiza que el participante voluntariamente ha expresado su intención en participar de la investigación, que conoce los objetivos detrás de ella, los beneficios, las molestias, sus derechos y responsabilidades, los riesgos que trae si hubiese lugar a ellos, la cantidad de tiempo de él requerido y las condiciones bajo las cuales el experimento es realizado.

A cada participante del experimento se le entregó dos copias idénticas del consentimiento informado, una copia la guarda el participante y la segunda la regresa firmada y es guardada por el observador como un documento soporte de la investigación.

El consentimiento informado firmado por cada uno de los participantes del experimento puede ser accedido a través del anexo **Consentimiento Informado**.

### 2.4. Prueba de Competencias

Como mecanismo para validar el prerrequisito de conocimientos básicos de UML por parte de los participantes, y adicionalmente como mecanismo de clasi-

ficación de Novato y Experto, se diseñó una prueba de competencias con preguntas relacionadas a la sintaxis básica de UML.

La prueba de competencias consistió de 17 preguntas separadas en dos secciones: Conceptos Básicos y Conceptos Avanzados.

La sección de nombre **Conceptos Básicos** formada por 11 preguntas. Las preguntas están enfocadas en verificar el conocimiento de elementos básicos, como son: tipos de relación (herencia, implementación, asociación, agregación y composición) y tipo de elemento grafico (clase, paquete, componente, nodo, actor y caso de uso).

La sección de nombre **Conceptos Avanzados** formada por 6 preguntas. Estas preguntas están enfocadas en verificar el conocimiento en elementos más complejos de UML, como son: diagrama de clases, diagrama de casos de uso, diagrama de paquetes, diagrama de despliegue y diagrama de comunicación.

Al finalizar la prueba de competencias, el participante queda clasificado en uno de los dos grupos: Novatos o Expertos. La clasificación en cada grupo depende de la cantidad de respuestas correctas alcanzadas en la prueba, de la siguiente manera:

- **Novato**: respuesta correctas entre 7 y 11 (40% hasta 70%)
- **Experto**: respuestas correctas entre 12 y 17 (70% hasta 100%)

El participante que no logre alcanzar la cantidad mínima de respuestas requeridas para clasificarse en el grupo de Novatos, no puede participar en el experimento, debido a que no cuenta con los conocimientos básicos requeridos para una correcta ejecución.

Los participantes del experimento para la herramienta ZMT fueron en total 16 personas. Los cuales quedaron clasificados según la prueba de competencias de la siguiente manera:

- **Novatos** = 7 participantes
- **Expertos** = 9 participantes

Afortunadamente, todos los participantes superaron el umbral definido para participar del experimento y quedaron clasificados en alguno de los dos grupos.

Esta clasificación de Novatos y Expertos no tiene ninguna implicación en las tareas que realizaron los participantes en la herramienta, todos los participantes sin excepción realizaron las mismas tareas. La clasificación solo tiene efecto para el análisis de los resultados y poder observar el comportamiento/desempeño de usuarios novatos vs expertos.

La prueba de competencias desarrollada por cada participante junto con los resultados obtenidos puede ser accedida a través del anexo **Prueba de Competencias**.

## 2.5. Formulario de Tareas

El formulario de tareas contiene las tareas que se le solicitó realizar a cada participante como método de evaluación de la herramienta ZMT.

El formulario está dividido en dos secciones, entrenamiento y tareas.

La sección de entrenamiento está diseñada con el objetivo de que el participante realice una inspección detallada sobre la herramienta, descubriendo las diferentes funcionalidades implementadas. Esta inspección la realiza de forma autónoma e individual, sin recibir un manual de usuario o ayuda alguna por parte del observador. Al finalizar el entrenamiento, de ser necesario, el observador realiza un entrenamiento guiado para asegurarse de que el participante conoce todas y cada una de las funcionalidades, y por lo tanto, cuenta con el conocimiento para resolver sin problemas las siguientes tareas propuestas en el formulario.

La sección de tareas está formada por 7 tareas. El orden de aparición de las tareas en el formulario electrónico es totalmente aleatorio. La aleatorización es un principio en el diseño de experimentos que aumenta la posibilidad de que el supuesto de independencia de los errores se cumpla, asegurando que las pequeñas diferencias generadas por factores no controlados se repartan de manera homogénea en todas las tareas [18]. Es decir, el rendimiento y entusiasmo del participante puede variar entre las primeras preguntas y las últimas preguntas, causado por cansancio, inconformidad hacia la herramienta u otras razones desconocidas. Por estos motivos, se usa la aleatorización del orden de las tareas logrando que éste efecto se distribuya entre todas.

Las tareas que han sido definidas corresponden a los escenarios de prueba definidos durante la fase de diseño del experimento, descrito en la sección anterior del presente documento.

Durante la ejecución de cada una de las tareas, el observador estuvo alerta para tomar notas y medir el tiempo usado por el participante para llegar a una respuesta.

Los formularios de tareas desarrollados por los participantes al igual que sus respuestas, pueden ser accedidos a través del anexo **Formulario de Tareas**.

## 2.6. Encuesta de Satisfacción

La encuesta de satisfacción busca medir el grado de confort que vivió el participante durante el uso de la herramienta. Es un estudio que brinda valores cualitativos acerca de la percepción del participante en cuanto a la usabilidad, aprendizaje, capacidades del sistema, rendimiento, utilidad, etc.

La encuesta desarrollada consiste de 17 preguntas y una pregunta adicional de comentarios y sugerencias. Las preguntas están clasificadas en las siguientes categorías: reacciones al software en general, aprendizaje, capacidades del sistema y usabilidad.

Las preguntas de esta encuesta de satisfacción fueron extraídas de encuestas existentes en el mercado que han sido rigurosamente probadas y comprobadas, como son:

**Questionnairefor User InteractionSatisfaction - QUIS**[52], es una herramienta desarrollada por un equipo multidisciplinar de investigadores del laboratorio de HCIde la universidad de Maryland. QUIS fue desarrollado para medir la satisfacción subjetiva del usuario con aspectos de la interfaz humano-computador.

**Software UsabilityMeasurementInventory - SUMI**[53], es un método rigurosamente probado y comprobado para medir la calidad del software desde el punto de vista del usuario final.

**MeasuringUsabilitywiththeSystemUsabilityScale - SUS**[54], es un cuestionario para medir la percepción de usabilidad.

El formulario de Satisfacción al igual que las respuestas entregadas por cada participante, pueden ser accedidos a través del anexo **Encuesta de Satisfacción**.

# V – RESULTADOS Y ANÁLISIS

Esta sección describe los resultados obtenidos del experimento y su análisis estadístico.

## 1. TABULACIÓN DE RESULTADOS

El resultado de la ejecución realizada por cada participante fue tabulado a través de la *Tabla 5*, la cual es mostrada a continuación.La tabla muestra cada uno de los escenarios que fueron evaluados. Por cada escenario,especifica el tiempo usado por el participante para completar la tarea, indica si la respuesta fue correcta o incorrecta, el puntaje obtenido (basado en respuestas correctas), el tiempo penalizado y una columna para indicar observaciones, como por ejemplo, los elementos usados de la herramienta para llegar a la respuesta seleccionada.

| **PART-XX** | | | | | |
|---|---|---|---|---|---|
| **ESCENARIO** | **TIEMPO (seg)** | **¿CORRECTA?** | **PUNTAJE** | **TIEMPO PENALIZADO** | **¿QUÉ USÓ?** |
| Entrenamiento | | | | | |
| Conociendo el Modelo | | | | | |
| Buscando Nodo | | | | | |
| Paquete con mayor cantidad de conexiones | | | | | |
| Paquete con la mayor cantidad de nodos | | | | | |
| Nombre las clases relacionadas con la clase A | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| Clase con la mayor cantidad de métodos | | | | | |
| Nombre los CU en los que participa el objeto A | | | | | |
| **Total** | | | | | |
| **Media** | | | | | |
| **Desviación Estándar** | | | | | |

Tabla 5: Plantilla para recolección de resultados por participante

## 2. MECANIMOS DE CALIFICACIÓN

El mecanismo de calificación seleccionado para evaluar las respuestas dadas por los participantes, está diseñado para valorar el esfuerzo del participante por llegar a la solución de la pregunta, incluso si no llega a la respuesta correcta. Es decir, el mecanismo permite valorar las respuestas que sin ser las correctas, son cercanamente correctas. Un ejemplo que permite entender de forma clara el mecanismo es el siguiente: *"Se le pide al buscar la clase con mayor cantidad de métodos",* el participante selecciona como su respuesta la clase X que tiene 9 métodos, pero la respuesta correcta es la clase Y que tiene 10 métodos.

El mecanismo de calificación que se manejó en este caso quiere tomar en cuenta ésta cercanía para valorar el esfuerzo del participante por encontrar la respuesta. Éste mecanismo de calificación es mostrado a continuación.

| **CALIFICACIÓN** | | | |
|---|---|---|---|
| **DESCRIPCIÓN** | **CORRECTA** | **PUNTAJE** | **PENALIZACIÓN** |
| Respuesta totalmente incorrecta | 0 | 0 | 100% |
| La respuesta más correcta | 1 | 5 | 0% |
| La segunda respuesta más correcta | 2 | 3 | 15% |
| La tercera respuesta más correcta | 3 | 2 | 30% |

Tabla 6: Mecanismo de Calificación

Las respuestas generadas por el participante a cada uno de los escenarios evaluados sonclasificadas en una escala de 0 a 3. La clasificación de 0, indica que la respuesta entregada por el participante es totalmente incorrecta o incluso que no entregó una respuesta para dicho escenario. La clasificación de 1, indica que el participante seleccionó la respuesta correcta. La clasificación de 2, indica que el participante seleccionó la segunda respuesta más correcta (la respuesta más cercana a la correcta sin llegar a ser la correcta). La clasificación de 3, indica que el participante seleccionó la tercera respuesta más correcta.

Cada una de estas clasificaciones (0, 1, 2 y 3) están asociadas a un puntaje y a una penalización en tiempo. Una respuesta correcta otorga 5 puntos y una penalización en tiempo del 0%; una respuesta clasificada en 2 (segunda respuesta más correcta) otorga un puntaje de 3 puntos y una penalización en tiempo del 15%; una respuesta clasificada en 3 (tercera respuesta más correcta) otorga un puntaje de 2 puntos y una penalización en tiempo del 30%; cualquier respuesta diferente obtiene 0 puntos y una penalización en tiempo del 100%.

La penalización en tiempo, es manejada con respecto al tiempo usado por el participante para dar la respuesta. Es decir, para el escenario en el cual el participante llega a la respuesta correcta en un tiempo de 25 seg, tiene una calificación de 5 puntos y 0% de tiempo penalizado, por lo tanto, el tiempo total, incluidas penalizaciones, es de los mismos 25 seg; para el escenario en el que el participante llega a la segunda respuesta más correcta en el mismo tiempo de 25 seg, obtiene un puntaje de 3 puntos y una penalización en tiempo del 15%, por lo tanto, el tiempo total incluidas penalizaciones es de 28.75 seg; en el caso de que el participante llegue a una respuesta incorrecta en un tiempo de 25 seg, obtiene un puntaje de 0 puntos y una penalización en tiempo del 100%, por lo tanto, el tiempo total incluidas penalizaciones es de 50 seg.

# 3. RESULTADOS CUANTITATIVOS

En este punto se describen los resultados cuantitativos obtenidos durante la ejecución del experimento. Los resultados cuantitativos están distribuidos en dos categorías: tiempo y puntaje.

## 3.1. Tiempo

La primera medida recolectada durante el experimento, fue el tiempo (medido en segundos) que usó cada uno de los participantes para la solución de los escenarios planteados en el experimento. Los tiempos de cada participante fueron organizados en una hoja de Excel, siguiendo la estructura de la *Tabla 5* mostrada en el presente capitulo.

Después de tener los tiempos tabulados de cada participante, fueron agrupados en la tabla del grupo al que pertenece el participante (tradicional, zui o mixto). La tabla de grupo contiene el tiempo total usado por cada participante.

A continuación, se muestran los resultados obtenidos por cada uno de los tres grupos y su análisis respectivo.

### 3.1.1. Grupo Tradicional

Los participantes que hacen parte del grupo de visualización tradicional obtuvieron los siguientes resultados en tiempo:

| VISUALIZACIÓN TRADICIONAL - TIEMPO (seg) | | | |
|---|---|---|---|
| NOVATOS | | EXPERTOS | |
| PARTICIPANTE | TIEMPO (seg) | PARTICIPANTE | TIEMPO (seg) |

| | | | |
|---|---|---|---|
| PART-03 | 375 | PART-02 | 344,6 |
| PART-07 | 299 | PART-05 | 399 |
| PART-11 | 414 | PART-06 | 383 |
| PART-14 | 610 | PART-09 | 273 |
| | | PART-12 | 273 |
| | | PART-13 | 337 |
| **MEDIA** | **424,50** | | **334,93** |
| **DESV. EST.** | **132,57** | | **53,26** |

Tabla 7: Tiempo Grupo Tradicional

Analizando los resultados obtenidos por los participantes novatos y expertos en el uso de la herramienta tradicional, se puede observar que los expertos lograron una media de tiempo menor que los novatos, y la desviación estándar muestra una dispersión relativamente pequeña e inferior con respecto a la de los novatos.

### 3.1.2. Grupo ZUI

Ahora, para el caso de los participantes del grupo de visualización ZUI, obtuvieron los siguientes resultados en tiempo:

| **VISUALIZACIÓN ZUI - TIEMPO (seg)** | | | |
|---|---|---|---|
| **NOVATOS** | | **EXPERTOS** | |
| **PARTICIPANTE** | **TIEMPO** | **PARTICIPANTE** | **TIEMPO** |
| PART-08 | 383 | PART-01 | 538 |
| PART-10 | 656 | PART-02 | 475 |
| PART-15 | 556 | PART-05 | 575 |
| PART-03 | 273 | PART-06 | 487 |
| | | PART-12 | 233 |
| | | PART-04 | 349 |
| | | PART-16 | 385 |
| **MEDIA** | **467,00** | | **434,57** |
| **DESV. EST.** | **171,59** | | **119,17** |

Tabla 8: Tiempo Grupo ZUI

En esta tabla se puede observar que la media de tiempo requerido por novatos y expertos fue muy similar, y la desviación estándar fue un poco menor para el caso de los expertos. Hasta este punto se ve un comportamiento relativamente normal (los expertos desarrollaron las tareas en un menor tiempo y su desviación estándar fue más pequeña). Pero cuando se observa estos valores (media y desviación estándar) y se compara con los valores de la tabla de visualización tradicional, se pueden observar varias situaciones interesantes.

Para el caso de los novatos, la media del tiempo requerido en ZUI fue levemente superior que la media de tiempo requerida en la tradicional, situación similar ocurrió para la desviación estándar, la cual en ZUI se vio levemente incrementada (tiempos más dispersos).

Para el caso de los expertos, la media de tiempo requerido en ZUI tuvo un mayor incremento, llegando a estar muy cercana de la media requerida por usuarios novatos. En cuanto a la desviación estándar, en ZUI fue bastante mayor que los valores obtenidos por los participantes que trabajaron en la herramienta tradicional.Esto sugiere que los expertos se vieron enfrentados a algo nuevo y que tuvieron que aprender. Mientras algunos lo asimilaron rápido, a otros les tomo un poco más de tiempo. En los novatos no fue tan visible éste comportamiento, lo que podría ser explicado, porque al ser usuarios nuevos en el uso de herramientas de modelado UML y verse enfrentados a otra herramienta igualmente nueva, no se ven tan impactados como en el caso de los expertos.

### 3.1.3. Grupo Mixto

Para el caso de los participantes que hacen parte del grupo mixto, ellos obtuvieron los siguientes resultados en tiempo durante la primera y segunda ejecución:

| GRUPO MIXTO - TIEMPO (seg) | | | | |
|---|---|---|---|---|
| PARTICIPANTE | PRIMERA RONDA | | SEGUNDA RONDA | |
| | TRADICIONAL | ZUI | TRADICIONAL | ZUI |
| PART-02 | 344,6 | 475 | 322 | 336 |
| PART-03 | 375 | 273 | 389 | 496 |
| PART-05 | 399 | 575 | 209 | 339 |
| PART-06 | 383 | 487 | 490 | 443 |
| PART-12 | 273 | 233 | 205 | 274 |
| **MEDIA** | **354,92** | **408,60** | **323,00** | **377,60** |
| **DESV. EST.** | **49,88** | **147,87** | **121,62** | **89,79** |

Tabla 9: Tiempo Grupo Mixto

Los resultados de la primera ejecución muestran que el tiempo medio requerido en el uso de ZUI fue mayor que el tiempo medio requerido por la herramienta tradicional, adicionalmente, la desviación estándar para el caso de ZUI fue significativamente más alta.

Para la segunda ejecución, se puede apreciar una mejora en los tiempos promedio requeridos por ambas herramientas con respecto a la primera ejecución, obteniendo un mejor tiempo la herramienta de visualización tradicional. Al observar la desviación estándar, se puede ver que los participantes durante el uso de ZUI, tuvieron una desviación bastante más pequeña comparada con la primera vez que usaron ZUI e incluso, estuvo por debajo de la

desviación calculada en la segunda ejecución de la herramienta de visualización tradicional. Estos resultados sugieren, que cuando los participantes previamente han conocido y usado ZUI, además de mejorar el tiempo requerido para la ejecución de las tareas, logran desempeñarse de manera bastante similar unos con otros.

## 3.2. Puntaje

La segunda medida recolectada durante el experimento fue el puntaje logrado por los participantes, basado en las respuestas correctas que tuvieron. A continuación se muestran los resultados obtenidos por cada uno de los tres grupos (tradicional, ZUI y mixto) y su análisis respectivo.

### 3.2.1. Grupo Tradicional

Los participantes que hacen parte del grupo de visualización tradicional, obtuvieron los siguientes resultados en puntaje:

| VISUALIZACIÓN TRADICIONAL - PUNTAJE | | | |
|---|---|---|---|
| **NOVATOS** | | **EXPERTOS** | |
| **PARTICIPANTE** | **PUNTAJE** | **PARTICIPANTE** | **PUNTAJE** |
| PART-03 | 28 | PART-02 | 28 |
| PART-07 | 30 | PART-05 | 25 |
| PART-11 | 26 | PART-06 | 28 |
| PART-14 | 21 | PART-09 | 30 |
| | | PART-12 | 25 |
| | | PART-13 | 28 |
| **MEDIA** | **26,25** | | **27,33** |
| **DESV. EST.** | **3,86** | | **1,97** |

Tabla 10: Puntaje Grupo Tradicional

Se puede observar que los expertos lograron una media por encima de la lograda por los novatos y su desviación estándar es menor comparada con la desviación estándar de los novatos. Lo que sugiere que los expertos al usar la herramienta de visualización tradicional alcanzan un mejor desempeño que los novatos.

### 3.2.2. Grupo ZUI

Para el caso de los participantes del grupo de visualización ZUI, obtuvieron los siguientes resultados en puntaje:

| VISUALIZACIÓN ZUI – PUNTAJE | | | |
|---|---|---|---|
| **NOVATOS** | | **EXPERTOS** | |
| **PARTICIPANTE** | **PUNTAJE** | **PARTICIPANTE** | **PUNTAJE** |
| PART-08 | 24 | PART-01 | 30 |

| | | | |
|---|---|---|---|
| PART-10 | 24 | PART-02 | 30 |
| PART-15 | 28 | PART-05 | 27 |
| PART-03 | 30 | PART-06 | 25 |
| | | PART-12 | 24 |
| | | PART-04 | 30 |
| | | PART-16 | 25 |
| **MEDIA** | **26,50** | | **27,29** |
| **DESV. EST.** | **3,00** | | **2,69** |

Tabla 11: Puntaje Grupo ZUI

Al observar la media y la desviación estándar se puede detectar que los expertos tuvieron una media mayor que los novatos. Sin embargo, las desviaciones estándar son bastante cercanas. Analizando estos valores con respecto a los valores obtenidos del uso de la herramienta tradicional, se puede ver que los usuarios novatos tuvieron una desviación estándar similar en el uso de la herramienta tradicional y en el uso de ZUI, y es algo lógico, al ser usuarios novatos en el uso de herramientas de modelado UML no se ven muy afectados al cambiarlos de una herramienta a otra.

Para el caso de los expertos, se puede notar algo muy diferente, mientras en la herramienta tradicional tuvieron una desviación estándar relativamente baja y muy inferior a la de los novatos, en ZUI esta desviación se incrementó llegando a un valor muy similar a la deviación estándar de los novatos. Éstos datos sugieren que los expertos se vieron enfrentados a algo nuevo que tuvieron que aprender y mientras unos lo asimilaron rápido a otros les tomo un poco más de tiempo.

### 3.2.3. Grupo Mixto

En el caso de los participantes que hacen parte del grupo mixto, ellos obtuvieron los siguientes resultados en puntaje:

| **GRUPO MIXTO - PUNTAJE** | | | | |
|---|---|---|---|---|
| **PARTICIPANTE** | **PRIMERA RONDA** | | **SEGUNDA RONDA** | |
| | **TRADICIONAL** | **ZUI** | **TRADICIONAL** | **ZUI** |
| PART-02 | 28 | 30 | 30 | 28 |
| PART-03 | 28 | 30 | 22 | 28 |
| PART-05 | 25 | 27 | 27 | 26 |
| PART-06 | 28 | 25 | 30 | 30 |
| PART-12 | 25 | 24 | 30 | 30 |
| **MEDIA** | **26,80** | **27,20** | **27,80** | **28,40** |
| **DESV. EST.** | **1,64** | **2,77** | **3,49** | **1,67** |

Tabla 12: Puntaje Grupo Mixto

En la primera ejecución se puede notar que ZUI logra una diferencia positiva en la media muy leve respecto a la visualización tradicional, en cambio la desviación estándar es más alta (los puntajes estuvieron más dispersos), lo que indica que algunos participantes se vieron un poco más afectados que otros al verse enfrentados por primera vez a una técnica de visualización diferente de la tradicional.

Para la segunda ejecución se puede notar algo interesante, y es que la media de ZUI se mantuvo por encima con respecto a la media de la tradicional, e incluso, aumentó con respecto a las medias obtenidas durante la primera ejecución; caso similar ocurrió con la desviación estándar, la cual en ésta segunda ejecución de ZUI, fue considerablemente menor que la desviación para la herramienta tradicional, e incluso, fue mucho menor que la desviación estándar obtenida de la primera ejecución de ZUI.

Para complementar la información tabulada y mostrada en la *Tabla 12*, a continuación se muestran dos histogramas donde se puede visualizar la distribución del puntaje de la primera y segunda ejecución para las herramientas Tradicional vs ZUI.

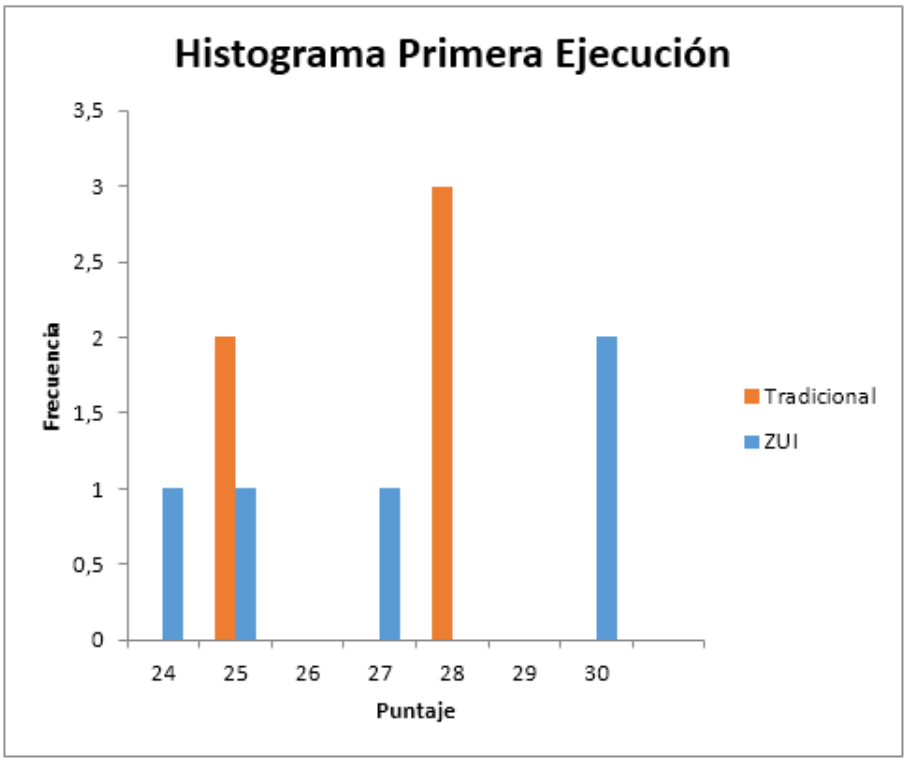


Figura 13: Histograma Mixto Primera Ejecución

Durante la primera ejecución, se puede observar que los participantes al momento de usar la herramienta tradicional estuvieron cercanos, relativamente parejos en los puntajes obtenidos. Cuando los mismos participantes se vieron enfrentados a la herramienta ZUI por primera vez, se puede notar una mayor dispersión, expresada en la diferencia de puntajes obtenidos y en una desviación estándar más alta.

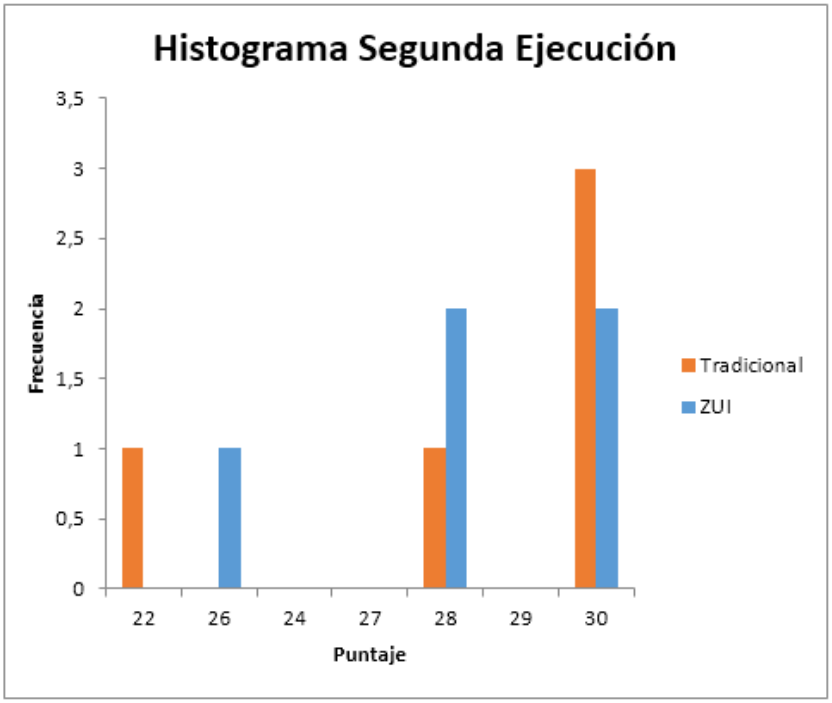


Figura 14: Histograma Mixto Segunda Ejecución

Durante la segunda ejecución,se puede observar que los participantes cuando usaron ZUI lograron una media de puntaje mayor y su desempeño fue más cercano entre ellos, reflejado por una desviación estándar más pequeña.

Estos resultados sugieren que el participante una vez conoce el paradigma ZUI y lo ha incorporado en su base de conocimiento, podría mejorar su desempeño, llegando incluso a superar el desempeño obtenido en herramientas de visualización tradicional.

Estas diferencias observadas en los tres grupos fueron analizadas estadísticamente a través de la distribución t-student [18] con un nivel de significancia predefinido del 5% ($\alpha = 0.05$). Aunque los resultados obtenidos muestran que esta diferencia no alcanza a ser suficiente para ser estadísticamente significativa, habría que verificar si esto persiste con una muestra de mayor tamaño.

Información más detallada con respecto a la ejecución y análisis estadístico puede ser encontrada en el anexo **Ejecución de Experimento**.

## 4. RESULTADOS CUALITATIVOS

El experimento además de resultados cuantitativos (tiempo y puntaje) generó también resultados cualitativos. Los resultados cualitativos fueron obtenidos a través de una encuesta de satisfacción, cada participante completo la encuesta al finalizar el experimento.

La encuesta desarrollada consistió de 17 preguntas y una pregunta adicional de comentarios y sugerencias. Las preguntas fueron clasificadas en las siguientes categorías: reacciones al software en general, aprendizaje, capacidades del sistema y usabilidad.

En general, los resultados de la encuesta muestran una buena aceptación del paradigma ZUI por parte de los participantes. Los resultados de la encuesta de satisfacción pueden ser accedidos a través del anexo **Ejecución Experimento**.

Además, como resultado de las observaciones realizadas por el observador, se logró evidenciar que cuando los participantes trabajan sobre la herramienta de visualización tradicional, muestran una alta dependencia del árbol de navegación, éste comportamiento se pudo evidenciar en cada tarea donde se solicitó buscar un elemento especifico e incluso durante el proceso de navegación. Caso contrario ocurrió en la herramienta de visualización ZUI, donde los participantes en su gran mayoría realizaron cada una de las tareas asignadas a través del lienzo de dibujo, olvidándose casi por completo del árbol de navegación.

Este comportamiento fue incluso más evidente en el grupo mixto, donde se solicita a los participantes realizar el experimento haciendo uso de las dos herramientas. De manera general, cuando se les solicitó buscar un elemento específico dentro del modelo, al momento de usar la herramienta tradicional, usaron el árbol de navegación como índice de contenido; al momento de realizar la misma búsqueda usando la herramienta ZUI, los participantes usaron el lienzo de dibujo y las técnicas de zooming y panning.

Éste comportamiento llama la atención, porque la herramienta ZUI también cuenta con el árbol de navegación de forma idéntica a la herramienta de visualización tradicional, y por alguna motivo, cuando los participantes tuvieron acceso a ZUI usaron el zooming y panning en lugar del árbol de navegación.

# VI– CONCLUSIONES Y TRABAJO FUTURO

## 1. CONCLUSIONES

A partir del experimento realizado y aunque fue con un grupo pequeño (16 participantes), las observaciones sugieren que los usuarios reciben de buena manera

nuevas formas de visualizar y navegar a través de la información. Específicamente hablando de la herramienta ZUI construida para navegar a través de modelos de software, los participantes la entendieron y se familiarizaron rápidamente con su uso, logrando un buen desempeño en la ejecución de las tareas.

En base a los análisis estadísticos realizados sobre los datos capturados del experimento (tiempo y puntaje), sugieren que, aunque inicialmente cuando los participantes usaron ZUI requirieron un tiempo mayor para resolver las tareas, la precisión alcanzada en las respuestas correctas fue mayor comparada con las respuestas correctas logradas a través de la herramienta tradicional. El grupo mixto, quien realizó nuevamente el experimento 2 semanas después de la primera ejecución, sugiere que, una vez el participante conoce el paradigma ZUI y lo ha incorporado en su conocimiento, podría mejorar su desempeño, desarrollando las tareas en un menor tiempo y con una mayor exactitud. Observando la media y desviación estándar obtenidas de la segunda ejecución de ZUI, sus valores sugieren una mejora en el desempeño de los participantes, debido a que la media de puntaje alcanzada fue un poco mayor que la media lograda en la primera ocasión y la desviación estándar fue más pequeña, mostrando una menor dispersión referente al puntaje obtenido por los diferentes participantes.

Aunque las diferencias descritas no lograron ser estadísticamente significativas (analizadas con un nivel de significancia del 5%), la aparente tendencia a favor de ZUI es una razón para continuar investigando sobre el tema, e incluso planear un nuevo experimento de mayores dimensiones que incluya un grupo suficientemente grande y representativo de la población, para que los resultados obtenidos sean mucho más confiables y muestren diferencias que puedan ser demostradas estadísticamente como significativas.

# 2. TRABAJO FUTURO

El framework ZMT en su primera y única versión hasta el momento, estuvo enfocada en la visualización y navegación de grafos que representan modelos de software.Trabajos futuros podrían estar enfocados a enriquecerlo con funcionalidades de creación, edición y persistencia de modelos, incluyendo el soporte de nuevos tipos de diagramas. Además, se podría implementar un nuevo módulo para la definición e instanciación de meta modelos, lo cual permitirá la creación de nuevos lenguajes gráficos y los respectivos modelos a partir de ellos.

En paralelo a esfuerzos que permitan enriquecer el framework, se debe diseñar y ejecutar un experimento más completo (basado en el experimento ya realizado y las lecciones aprendidas), con objetivos más ambiciosos, que incluya un grupo suficientemente grande y representativo de la población, para que los resultados obtenidos sean mucho más confiables y muestren diferencias que puedan ser demostradas estadísticamente como significativas.

Si los resultados futuros validaran de forma irrefutable que ZUI mejora la comprensión de los modelos por parte del usuario, se podría pensar y diseñar nuevas herramientas que brinden soporte en áreas diferentes de la ingeniería de software, que al igual que en la ingeniería para el caso de los modelos de software, presentan problemas similares con respecto a la navegación de grandes volúmenes de información distribuida de forma jerárquica y donde ZUI podría ser de gran utilidad.

# 3. LECCIONES APRENDIDAS

Para obtener unos resultados más exactos en las medidas obtenidas durantela ejecución del experimento, se recomiendarealizar la ejecución sobre el modelo de un único sistema software. Para el caso del experimento realizado en la presente investigación, se usaron los modelos de tres sistemas software diferentes, aunque de similar complejidad, siempre hay un ruido generado por esta diferencia que de cierto modo afecta los resultados obtenidos.

Otro tema a tener en cuenta, es el conocimiento previo de los usuarios no solo sobre herramientas de modelado UML, sino también, en el uso de herramientas que provean un árbol de navegación. Con el experimento ejecutado, se pudo observar que para la herramienta de visualización tradicional, los usuarios dependen mucho del árbol de navegación, toda la navegación y búsqueda la hacen por éste medio, por lo tanto, el conocimiento en ésta herramienta debería ser tenida en cuenta para ser clasificado como novato o experto.

Durante la ejecución del experimento se pudo observar que el orden de los escenarios de prueba debe ser manejado con mucho cuidado. En algunos momentos durante el experimento, el participante llegó a una tarea que tenía relación con la anterior y encontró la respuesta de forma muy rápida, habría generado un resultado diferente, en el caso de que dichas tareas con relación hubiesen sido separadas por más de una tarea.

# VII - REFERENCIAS

[1] J. A. Pavlich-Mariscal, H. D. Veliz-Quispe, S. A. Demurjian, and L. D. Michel, "Un ambiente de meta-modelado y visualización basado en el paradigma de Zoomable User Interfaces," *Ingeniare Rev. Chil. Ing.*, vol. 23, no. 2, pp. 219–234, Apr. 2015.

[2] A. Cockburn, A. Karlson, and B. B. Bederson, "A review of overview+detail, zooming, and focus+context interfaces," *ACM Comput Surv*, vol. 41, no. 1, pp. 2:1–2:31, enero 2009.

[3] L. Good and B. B. Bederson, "Zoomable User Interfaces as a Medium for Slide Show Presentations," *Inf. Vis.*, vol. 1, no. 1, pp. 35–49, Mar. 2002.

[4] N. Schweikardt, H. Reiterer, and T. Büring, "Zooming Techniques," in *Encyclopedia of Database Systems*, L. Liu and M. T. öZsu, Eds. Boston, MA: Springer US, 2009, pp. 3684–3689.

[5] M. Frisch and R. Dachselt, "Benefits of interactive display environments in the software development process," in *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, New York, NY, USA, 2008, pp. 53–56.

[6] M. Frisch, R. Dachselt, and T. Brückmann, "Towards seamless semantic zooming techniques for UML diagrams," in *Proceedings of the 4th ACM symposium on Software visualization*, New York, NY, USA, 2008, pp. 207–208.

[7] B. B. Bederson, "The promise of zoomable user interfaces," *Behav. Inf. Technol.*, vol. 30, no. 6, pp. 853–866, Nov. 2011.

[8] M. Beaudouin-Lafon, "Designing interaction, not interfaces," in *Proceedings of the working conference on Advanced visual interfaces*, 2004, pp. 15–22.

[9] G. Robertson, M. Czerwinski, K. Larson, D. C. Robbins, D. Thiel, and M. van Dantzich, "Data Mountain: Using Spatial Memory for Document Management," in *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA, 1998, pp. 153–162.

[10] "JavaScript," *Mozilla Developer Network*. [Online]. Available: https://developer.mozilla.org/es/docs/Web/JavaScript. [Accessed: 07-Nov-2015].

[11] Scott W. Ambler, "The Agile Unified Process (AUP)." [Online]. Available: http://www.ambysoft.com/unifiedprocess/agileUP.html. [Accessed: 25-Nov-2013].

[12] Don Wells, "Extreme Programming: A Gentle Introduction." [Online]. Available: http://www.extremeprogramming.org/. [Accessed: 25-Nov-2013].

[13] IBM, “Rational Unified Process: Best Practices for Software Development Teams.” [Online]. Available: https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf. [Accessed: 25-Nov-2013].

[14] “Unified Modeling Language (UML).” [Online]. Available: http://www.uml.org/. [Accessed: 07-Nov-2015].

[15] James Rumbaugh, Ivar Jacobson, and Grady Booch, *The Unified Modeling Language Reference Manual*. Addison-Wesley, 2004.

[16] Federico Rodríguez Bravo, “FV-GAIA: Framework para la visualización de Grafos con Ayuda de Interfaces Aumentables,” Pontificia Universidad Javeriana, Bogotá, Colombia, 2012.

[17] W. Chung, “A Survey of Zoomable User Interfaces,” 2006.

[18] Humberto Gutiérrez Pulido and Román de la Vara Salazar, *Análisis y diseño de experimentos*, Segunda Edición. Mc Graw Hill, 2008.

[19] Douglas C. Montgomery, *Diseño y Análisis de Experimentos*, Segunda Edición. LIMUSA WILEY, 2004.

[20] Gary Perlman, “HCI Bibliography : Human-Computer Interaction Resources.” [Online]. Available: http://hcibib.org/. [Accessed: 12-Nov-2015].

[21] Carnegie Mellon University, “Welcome | Human-Computer Interaction Institute.” [Online]. Available: https://www.hcii.cmu.edu/. [Accessed: 12-Nov-2015].

[22] Y. Rogers, H. Sharp, and J. Preece, *Interaction Design: Beyond Human - Computer Interaction*. John Wiley & Sons, 2011.

[23] M. Bennett and F. Cummins, “ORRIL: a simple building blocks approach to zoomable user interfaces,” 2004, pp. 639–644.

[24] C. Geiger, H. Reckter, R. Dumitrescu, S. Kahl, and J. Berssenbrügge, “A Zoomable User Interface for Presenting Hierarchical Diagrams on Large Screens,” in *Human-Computer Interaction. Novel Interaction Methods and Techniques*, vol. 5611, J. A. Jacko, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 791–800.

[25] K. Perlin and D. Fox, “Pad: an alternative approach to the computer interface,” in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993, pp. 57–64.

[26] B. B. Bederson, J. D. Hollan, K. Perlin, J. Meyer, D. Bacon, and G. Furnas, “Pad++: A zoomable graphical sketchpad for exploring alternate interface physics,” *J. Vis. Lang. Comput.*, vol. 7, no. 1, pp. 3–32, 1996.

[27] B. B. Bederson, J. Meyer, and L. Good, "Jazz: an extensible zoomable user interface graphics toolkit in Java," in *Proceedings of the 13th annual ACM symposium on User interface software and technology*, New York, NY, USA, 2000, pp. 171–180.

[28] "Piccolo2D - A Structured 2D Graphics Framework," *A Structured 2D Graphics Framework*. [Online]. Available: http://piccolo2d.org/. [Accessed: 12-Nov-2015].

[29] E. Pietriga, "A toolkit for addressing HCI issues in visual language environments," in *2005 IEEE Symposium on Visual Languages and Human-Centric Computing*, 2005, pp. 145–152.

[30] N. Zhu, J. Grundy, J. Hosking, N. Liu, S. Cao, and A. Mehra, "Pounamu: A meta-tool for exploratory domain-specific visual language tool development," *J. Syst. Softw.*, vol. 80, no. 8, pp. 1390–1407, 2007.

[31] T. Reinhard, S. Meier, R. Stoiber, C. Cramer, and M. Glinz, "Tool Support for the Navigation in Graphical Models," in *Proceedings of the 30th International Conference on Software Engineering*, New York, NY, USA, 2008, pp. 823–826.

[32] M.-A. Storey, C. Best, J. Michaud, D. Rayside, M. Litoiu, and M. Musen, "SHriMP views: an interactive environment for information visualization and navigation," in *CHI'02 extended abstracts on Human factors in computing systems*, 2002, pp. 520–521.

[33] Daniel Rico Hernández and Eduardo Montenegro León, "ZoomTI++: Framework para visualizar diagramas de software mediante Zoomable User Interfaces y tecnología Multi-Touch," Pontificia Universidad Javeriana, Bogotá, Colombia, 2015.

[34] Mike Bostock, "D3.js - Data-Driven Documents," *Data-Driven Documents*, 2015. [Online]. Available: http://d3js.org/. [Accessed: 07-Apr-2014].

[35] The jQuery Foundation, "jQuery." [Online]. Available: https://jquery.com/. [Accessed: 07-Apr-2014].

[36] Dmitry Baranovskiy, "Raphaël—JavaScript Library," *Raphaël—JavaScript Library*. [Online]. Available: http://raphaeljs.com/. [Accessed: 07-Apr-2014].

[37] Eric Rowell, "KineticJS," *KineticJS*. [Online]. Available: http://kineticjs.com/. [Accessed: 07-Apr-2014].

[38] W3C, "HTML5," *HTML5*. [Online]. Available: http://www.w3.org/TR/html5/. [Accessed: 07-Apr-2014].

[39] "JSON," *Introducing JSON*. [Online]. Available: http://www.json.org/. [Accessed: 21-Oct-2015].

[40] W3C, "W3C SVG Working Group," *Scalable Vector Graphics (SVG)*, 2010. [Online]. Available: http://www.w3.org/Graphics/SVG/. [Accessed: 21-Oct-2015].

[41] zTree, “Home [zTree -- jQuery tree plug-ins.],” *zTree -- jQuery tree plug-ins.*, 2010. [Online]. Available: http://www.ztree.me/v3/main.php#_zTreeInfo. [Accessed: 21-Oct-2015].

[42] Opensource.org, “The MIT License (MIT) | Open Source Initiative,” *The MIT License (MIT)*. [Online]. Available: https://opensource.org/licenses/MIT. [Accessed: 21-Oct-2015].

[43] Erich Gama, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed. Addison-Wesley Professional, 1994.

[44] “Hyperionics - the best screen capture software - Free download.” [Online]. Available: http://hyperionics.com/. [Accessed: 21-Sep-2015].

[45] “Stopwatch‰ on the App Store.” [Online]. Available: https://itunes.apple.com/us/app/stopwatch/id505889167?mt=8. [Accessed: 21-Sep-2015].

[46] Lornel Rivas, Maria Pérez, Luis E. Mendoza, and Anna Grimmán, “TOOLS SELECTION CRITERIA IN SOFTWARE-DEVELOPING SMALL AND MEDIUM ENTERPRISES,” vol. 10, no. 1, p. 7, Abril 2010.

[47] M. Eid, A. Alamri, J. Melhem, and A. El Saddik, “Evalutation of UML CASE Tool with Haptics,” in *Proceedings of the 2008 Ambi-Sys Workshop on Haptic User Interfaces in Ambient Media Systems*, ICST, Brussels, Belgium, Belgium, 2008, pp. 4:1–4:5.

[48] Jakob Nielsen, “Usability 101: Introduction to Usability,” *Nielsen Norman Group*, 04-Jan-2012. [Online]. Available: http://www.nngroup.com/articles/usability-101-introduction-to-usability/. [Accessed: 15-Aug-2015].

[49] Walter Ovidio Sánchez and others, “La usabilidad en Ingeniería de Software: definición y características,” 2011.

[50] Google, “Google Forms: Easy-to-create surveys and forms for everyone,” *Google Apps for Work*. [Online]. Available: https://apps.google.com/intx/en_uk/products/forms/. [Accessed: 24-Oct-2015].

[51] Universidad de la República - Uruguay, “Generación de Números Aleatorios Uniformes.” [Online]. Available: http://www.ccee.edu.uy/ensenian/licest/calcnum/material/cap5y6.pdf. [Accessed: 20-Sep-2015].

[52] “Questionnaire For User Interaction Satisfaction.” [Online]. Available: http://www.lap.umd.edu/quis/. [Accessed: 26-Sep-2015].

[53] "SUMI Questionnaire Homepage." [Online]. Available: http://sumi.ucc.ie/. [Accessed: 26-Sep-2015].

[54] "Measuring Usability with the System Usability Scale (SUS): MeasuringU." [Online]. Available: http://www.measuringu.com/sus.php. [Accessed: 26-Sep-2015].