



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**DEPARTAMENTO DE CIENCIAS DE LA
COMPUTACIÓN**

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**TESIS PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS E INFORMÁTICA**

AUTOR: LÓPEZ LUDEÑA ANDRÉS LEONARDO

**TEMA: “INVESTIGACIÓN DESCRIPTIVA DEL LENGUAJE
BASADO EN MODELOS “MARIA” PARA EL ANÁLISIS DE
TAREAS EN EL PROCESO DE INTERACCIÓN HUMANO
COMPUTADOR Y SU APLICACIÓN EN UNA HERRAMIENTA
CASE”**

DIRECTOR: ING. FONSECA, RODRIGO

SANGOLQUÍ, SEPTIEMBRE 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

CERTIFICADO

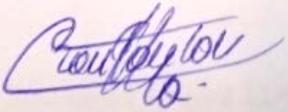
Ing. Rodrigo Fonseca

CERTIFICA

Que el trabajo titulado “INVESTIGACIÓN DESCRIPTIVA DEL LENGUAJE BASADO EN MODELOS “MARIA” PARA EL ANÁLISIS DE TAREAS EN EL PROCESO DE INTERACCIÓN HUMANO COMPUTADOR Y SU APLICACIÓN EN UNA HERRAMIENTA CASE.” realizado por el Sr. Andrés Leonardo López Ludeña, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas – ESPE.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (PDF). Se autoriza al Sr. Andrés Leonardo López Ludeña, que el material se entregue al Ing. Mauricio Campaña, en su calidad de Director de la Carrera.

Sangolquí, Septiembre de 2015



ING. RODRIGO FONSECA
DIRECTOR DE TESIS

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

DECLARACIÓN DE RESPONSABILIDAD

Yo, Andrés Leonardo López Ludeña

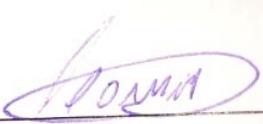
DECLARO QUE:

El proyecto de grado denominado “INVESTIGACIÓN DESCRIPTIVA DEL LENGUAJE BASADO EN MODELOS “MARIA” PARA EL ANÁLISIS DE TAREAS EN EL PROCESO DE INTERACCIÓN HUMANO COMPUTADOR Y SU APLICACIÓN EN UNA HERRAMIENTA CASE.”, ha sido desarrollado con base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme a las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de mí autoría.

En virtud de esta declaración me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, Septiembre de 2015



Andrés Leonardo López Ludeña
C.C: 1104595572

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

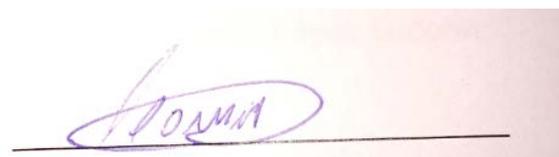
AUTORIZACIÓN DE PUBLICACIÓN

Yo, Andrés Leonardo López Ludeña, autorizo a la Universidad de las Fuerzas Armadas – ESPE la publicación, en la biblioteca virtual de la institución, del trabajo “INVESTIGACIÓN DESCRIPTIVA DEL LENGUAJE BASADO EN MODELOS “MARIA” PARA EL ANÁLISIS DE TAREAS EN EL PROCESO DE INTERACCIÓN HUMANO COMPUTADOR Y SU APLICACIÓN EN UNA HERRAMIENTA CASE.”, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y autoría.

Consecuentemente este trabajo es de mí autoría.

En virtud de esta declaración me responsabilizo del contenido, veracidad y alcance científico del proyecto de grado en mención.

Sangolquí, Septiembre de 2015



Andrés Leonardo López Ludeña
C.C: 1104595572

DEDICATORIA

Dedico este trabajo principalmente a Dios, quien es mi guía espiritual.

A mis padres, quienes han sido mi apoyo durante toda mi vida.

AGRADECIMIENTO

A todas las personas que estuvieron siempre ahí con su apoyo incondicional.

Agradecimiento a mi Mamá por estar siempre a mi lado y darme sus mejores palabras de aliento.

A mi Papá que siempre me motivó a seguir adelante y no dejarme vencer ante ninguna adversidad.

A todos mis familiares que me enseñaron los mejores valores y sus experiencias de vida.

A todos mis amigos y compañeros, que participaron de una u otra manera en esta etapa de mi vida, siempre los recordaré.

Y en especial a mis profesores que hicieron que este trabajo sea posible, al Ing. Danilo Martínez, Ing. Rodrigo Fonseca e Ing. Mauricio Campaña. Que a más de ser buenos educadores son grandes personas.

ÍNDICE DE CONTENIDOS

CERTIFICADO	i
DECLARACIÓN DE RESPONSABILIDAD	ii
AUTORIZACIÓN DE PUBLICACIÓN	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
ÍNDICE DE CONTENIDOS	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
ÍNDICE DE ANEXOS	xii
GLOSARIO	xiii
RESUMEN	xv
ABSTRACT	xvi
CAPÍTULO 1	1
INTRODUCCIÓN	1
1.1. ANTECEDENTES	1
1.2. PLANTEAMIENTO DEL PROBLEMA	4
1.3. JUSTIFICACIÓN	5
1.4. OBJETIVOS	6
1.4.1. Objetivo General	6
1.4.2. Objetivos Específicos	6
1.5. ALCANCE	6
CAPÍTULO 2	7
MARCO TEÓRICO	7
2.1. METODOLOGÍA DE INVESTIGACIÓN	7
2.1.1. INVESTIGACIÓN DESCRIPTIVA	7
2.1.1.1. ETAPAS	7
2.2. RED DE CATEGORÍAS	8
2.3. USABILIDAD Y LA INTERACCIÓN HUMANO COMPUTADOR	9

2.3.1.	Relación entre la Usabilidad y la Interacción humano computador	10
2.3.2.	Influencia en el Ciclo de Vida Software	10
2.3.3.	Técnicas para la mejora de la Usabilidad	11
2.4.	ANÁLISIS DE TAREAS	12
2.5.	MODELADO DE TAREAS	12
2.5.1.	USER ACTION NOTATION (UAN)	13
2.5.2.	GOALS, OPERATOR, METHODS, SELECTION RULES (GOMS)	14
2.5.3.	TASK ACTION GRAMMAR (TAG)	14
2.5.4.	CONCURTASKTREES (CTT)	14
2.6.	LENGUAJES DE MARCADO PARA INTERFACES DE USUARIO	17
CAPÍTULO 3		22
	LENGUAJE DE MODELADO MARIA, SU VINCULACIÓN CON EL DESARROLLO Y LA USABILIDAD DEL SOFTWARE	22
3.1	INTRODUCCIÓN	22
3.2	ORIGEN Y VERSIONAMIENTO DE MARIA	25
3.3	CARACTERÍSTICAS DEL MODELADO DE TAREAS EN MARIA	25
3.4.1.	INTERFAZ DE USUARIO ABSTRACTO	27
3.4.2.	INTERFAZ CENTRADO AL USUARIO ORIENTADO AL ESCRITORIO	30
3.4.3.	INTERFAZ CENTRADO AL USUARIO DE USO VOCAL	32
3.5	INTEGRACIÓN DE LA APLICACIÓN DE MARIA AL DESARROLLO DE SOFTWARE	34
3.6	IDENTIFICACIÓN DE LAS CARACTERÍSTICAS DE MARIA PARA MEJORAR LA USABILIDAD	36
CAPÍTULO 4		38
	DESARROLLO DE PROTOTIPO	38
4.1	INTRODUCCION	38
4.2	PROTOTIPO OPERACIONES MATEMÁTICAS	39
4.2.1	ASIGNACIÓN CONSUMO DEL SERVICIO WEB	42
4.2.2	INTERFAZ ABSTRACTA DE USUARIO	44
4.2.2.1	Presentación	46
4.2.2.2	Funciones externas	50

4.2.3 INTERFAZ CENTRADA AL USUARIO ORIENTADO AL ESCRITORIO	51
4.3 PROTOTIPO CONSULTA INFORMACIÓN AEROPORTUARIA.....	57
4.3.1 INTERFAZ CENTRADA AL USUARIO ORIENTADO AL ESCRITORIO	57
CAPÍTULO 5	60
CONCLUSIONES Y RECOMENDACIONES	60
5.1 CONCLUSIONES	60
5.2 RECOMENDACIONES.....	60
BIBLIOGRAFÍA	62

ÍNDICE DE FIGURAS

FIGURA 1. ASIGNACIÓN DE TÉCNICAS DE USABILIDAD A ACTIVIDADES DEL DESARROLLO (FERRÉ, 2003).....	3
FIGURA 2. RED DE CATEGORÍAS DE LAS VARIABLES DE INVESTIGACIÓN	9
FIGURA 3. TAREA DE ACCESO A UN MUSEO MEDIANTE CTT (ZUMAQUERO, 2008)	17
FIGURA 4. BENEFICIOS AL UTILIZAR EL LENGUAJE MARIA (FAEDO”, 2014)	20
FIGURA 5. DIAGRAMA DE FLUJO DE MARIA.....	24
FIGURA 6. MODELADO DE TAREAS UTILIZANDO NOTACIÓN CTT	27
FIGURA 7. CARACTERÍSTICAS GRÁFICAS DE LA INTERFAZ ABSTRACTA DE USUARIO	30
FIGURA 8. CARACTERÍSTICAS GRÁFICAS DE LA INTERFAZ DE USUARIO EN EL ESCRITORIO.....	32
FIGURA 9. CARACTERÍSTICAS GRAFICAS DE LA INTERFAZ VOCAL DE USUARIO	34
FIGURA 10. VERSIONAMIENTO DE MARIA ENVIRONMENT	38
FIGURA 11. MODELADO DE TAREAS DE LA OPERACIÓN DIVISIÓN	39
FIGURA 12. MODELADO DE TAREAS DE LA OPERACIÓN MULTIPLICACIÓN.....	40
FIGURA 13. MODELADO DE TAREAS DE LA OPERACIÓN RESTA.....	40
FIGURA 14. MODELADO DE TAREAS DE LA OPERACIÓN SUMA	40
FIGURA 15. PRESENTACIÓN EN MARIA ENVIRONMENT	42
FIGURA 16. REGISTRO DE SERVICIO WEB.....	43
FIGURA 17. VISUALIZACIÓN DE LAS OPERACIONES DEL SERVICIO WEB	44
FIGURA 18. CARACTERÍSTICAS FUNCIONALES DEL USUARIO ABSTRACTO.....	45
FIGURA 19. CONTENIDO DE LA DATA QUE PRESENTA LA INTERFAZ DE USUARIO ABSTRACTA.....	45
FIGURA 20. VINCULACIÓN CON LAS FUNCIONES EN LA CAPA DE PRESENTACIÓN	46
FIGURA 21. PROPIEDADES DE LA CAPA DE PRESENTACIÓN.....	47
FIGURA 22. REFERENCIA DE DATOS	47
FIGURA 23. FUNCIONALIDAD DE LA CARACTERÍSTICA TIPO (EVENTO) EN LA CAPA DE PRESENTACIÓN.....	48
FIGURA 24. SCRIPT DEL EVENTO EN LA CAPA DE PRESENTACIÓN ...	48
FIGURA 25. VISUALIZACIÓN DE SCRIPT	49

FIGURA 26. CARACTERÍSTICAS DEL CAMBIO DE PROPIEDAD	49
FIGURA 27. CARACTERÍSTICAS DE LAS FUNCIONES EXTERNAS	50
FIGURA 28. CUADRO DESCRIPTIVO.....	50
FIGURA 29. VISUALIZACIÓN DE LAS FUNCIONES EXTERNAS.....	51
FIGURA 30. CONSUMO DEL RECURSO DEL SERVICIO WEB	51
FIGURA 31. CAJA DE HERRAMIENTAS PARA MODIFICAR LA INTERFAZ.....	52
FIGURA 32. MODIFICACIÓN DEL CAMPO TIPO TEXTO	52
FIGURA 33. VISUALIZACIÓN EN HTML	53
FIGURA 34. EXPORTACIÓN A DIFERENTE INTERFAZ DE USUARIO.....	53
FIGURA 35. REPRESENTACIÓN A LA INTERFAZ DE DISPOSITIVO MÓVIL	54
FIGURA 36. EXPORTACIÓN DE CÓDIGO FUENTE.....	54
FIGURA 37. REPRESENTACIÓN GRÁFICA DEL PROTOTIPO EN JAVA	55
FIGURA 38. VENTANA DE CONSOLA	55
FIGURA 39. REPRESENTACIÓN DE PROTOTIPO CON EL CONSUMO WEB.....	56
FIGURA 40. REPRESENTACIÓN DE PROTOTIPO DEL CONSUMO WEB OPERACIÓN DE LA DIVISIÓN	57
FIGURA 41. MODELADO DE TAREAS INFORMACIÓN DEL AEROPUERTO	57
FIGURA 42. INCLUSIÓN DE IMÁGENES VIDEOS EN LA CAPA DE PRESENTACIÓN	58
FIGURA 43. CAPA DE PRESENTACIÓN EN HTML.....	58

ÍNDICE DE TABLAS

TABLA 1 REPRESENTACIÓN DE LOS LENGUAJES DE MERCADO CON SUS CARACTERÍSTICAS RELEVANTES.....	21
---	-----------

ÍNDICE DE ANEXOS

ANEXOS	64
ANEXO 1: MANUAL DE USUARIO	64

GLOSARIO

B

Back End: Procesadores que no hacen labores de interfaz con el exterior, sino más bien funciones especializadas, como tratamiento de la información en bases de datos, sistemas de control de comunicaciones o cálculos científicos.

C

CTT: Es una notación desarrollada por Fabio Paternò, permite representar relaciones temporales entre las tareas y los usuarios para lograr llevar a cabo las mismas.

D

DTMF: (Dual-Tone Multi-Frequency) es para la señalización de telecomunicaciones sobre líneas telefónicas analógicas en la banda de frecuencia vocal.

H

HCI: (Human-computer interaction) es la disciplina que estudia el intercambio de información mediante software entre las personas y las computadoras.

L

LOTOS: es una notación formal concurrente, usada para poder especificar interfaces de usuario, ya que permite describir comportamientos manejados por eventos y modificaciones de estado (Santoro, 2005).

M

MARIA: (Model-based Language foR Interactive Applications) es un lenguaje basado en modelos, universal, declarativo y de múltiple nivel de abstracción.

MARIA ENVIRONMENT: Herramienta Case para utilizar MARIA

S

Servicios Web: (en inglés, Web Service o Web services) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

X

XHTML: (eXtensible HyperText Markup Language) es una versión más estricta y limpia de HTML.

XML: (eXtensible Markup Language) es un lenguaje que se utiliza para almacenar datos en forma legible.

RESUMEN

A pesar de los problemas que presentan los sistemas informáticos actuales, estos se han convertido en una herramienta de apoyo en la vida cotidiana de las personas, debido a que intervienen en la mayoría de sus actividades. Algunos de los problemas identificados están asociados a la dificultad de uso del sistema, los cuales generalmente son defectos a nivel de sus modelos de **HCI** (de las siglas en inglés Human Computer Interaction) (por ejemplo: de interacción, interfaz gráfica, tareas, etc.). Por lo tanto, a un mediano o corto plazo, se precisa de mayor cantidad de recursos para corregir estos errores y mejorar la funcionalidad de los sistemas. El presente estudio se enfoca en el uso de la técnica de **Análisis de tareas**, que ayuda a obtener una mayor comprensión del problema en la fase de Análisis; así como también a realizar de mejor manera el diseño de interacción de la fase de diseño del desarrollo Software. En este contexto, se pueden identificar varias formas para realizar el **Modelado de tareas**; sin embargo, lo deseable sería utilizar un lenguaje que maximice las características principales del proceso HCI, con la finalidad de que el prototipo resultante se ejecute en varias plataformas y obtener múltiples niveles de abstracción (Xavier, 2003). Al utilizar el lenguaje basado en modelos **MARIA** se pretende aplicar la técnica de modelado de tareas, con cierto nivel de abstracción, cuya finalidad sea desarrollar un **Prototipo** interactivo proporcionando un refinamiento en la descripción de dichas tareas para las diversas plataformas y lenguajes existentes.

Palabras Clave:

MARIA

HCI

MODELADO DE TAREAS

PROTOTIPO

ANÁLISIS DE TAREAS

ABSTRACT

Despite the problems presented by current computer systems, they have become a support tool in the daily lives of people, because involved in most activities. Some of the problems identified are associated with the difficulty of using the system, which generally are defects level models of **HCI** (acronym in English Human Computer Interaction) (eg interaction, graphical interface, tasks, etc.). Therefore, a short or medium term, it needs more resources to correct these errors and improve the functionality of systems. This study focuses on the use of **Task analysis** technique, which helps to gain a greater understanding of the problem in the analysis phase; as well as perform better interaction design phase of software development design. In this context, one can identify several ways to perform the **Modeling tasks**; however, it would be desirable to use a language that maximizes the main features of HCI process, in order that the resulting **Prototype** to run on multiple platforms and get multiple levels of abstraction (Xavier, 2003). The study of a model for modeling tasks, from the perspective of the user interaction with the system. Proposes research-based language models for Interactive Applications **MARIA** (acronym of English language-based Model for Interactive Applications) Paternó Santoro & Spano (2012); and this language, which implements the technique of analysis tasks more emphasis on its operating logic, which aims to develop interactive systems where communication between the user and the system is simple. Using Model-Based MARIA language it is intended to apply the technique of modeling tasks, with some level of abstraction.

Keywords:

MARIA

HCI

MODELING TASKS

PROTOTYPING

TASK ANALYSIS

CAPÍTULO 1

INTRODUCCIÓN

La investigación realizada por el Grupo Standish¹, manifiesta en su último reporte (Chaos Report 2014) que el 31,1 % de los proyectos de software fueron cancelados antes de su finalización, y otros resultados indican que el 52,7% de los proyectos tuvieron un costo del 189% más alto respecto a sus estimaciones originales (Group, 2014); esto indica que sigue existiendo problemas en el desarrollo de producto software, sin que sea aún evidente una solución viable para que se pongan en marcha las empresas que desarrollan software.

Existen varias razones por las que los sistemas podrían tener deficiencias en su interacción. Para Seffah, Vanderdonckt, & Desmarais (2009), uno de los principales problemas es la brecha que existe entre la ingeniería de software y el desarrollo del modelo de (HCI). Otros autores como Sommerville I. (2006), resaltan que la construcción de interfaces de usuarios es parte significativa del proceso de desarrollo de software. Grau (2003) propone que el desarrollo de la interacción usuario computador debe realizarse en forma paralela al desarrollo del software en especial en la fase de análisis y diseño. Para integrar la usabilidad al proceso de desarrollo, es fundamental aplicar varias técnicas de usabilidad a las actividades de desarrollo (Grau, 2003).

1.1. ANTECEDENTES

“La usabilidad es un tema que está cobrando importancia cada vez mayor en el desarrollo de software. A pesar de ello, la Ingeniería del Software sigue

¹ Organización que se encarga de investigar el rendimiento de los proyectos software

enfocándose exclusivamente en atributos del software, como el rendimiento o la fiabilidad. Sin embargo en la actualidad los sistemas software están dirigidos a un público cada vez más amplio y a usuarios cada vez menos expertos en el manejo de sistemas informáticos, la usabilidad está sobresaliendo como atributo fundamental para el éxito de un producto software.” (Ferré, 2003).

La calidad en el producto software va más allá del rendimiento y características técnicas que podría tener, es la satisfacción total que posee el usuario al interactuar con el sistema (Ferré, 2003). Para (Ferré, 2003) la Interacción Persona computador (Human-Computer Interaction HCI) consiste en el intercambio de instrucciones a través del Software. HCI se encarga del diseño, evaluación e implementación de las TIC’s y maximiza el grado eficiencia y eficacia de las tareas computacionales y así su desarrollo sea más productivo.

Aunque el desarrollo de HCI es aceptado normalmente en la ingeniería de software, sus técnicas han dificultado su integración, debido a que sus terminologías, enfoques y perspectivas son distintas. A menudo, son consideradas como análogas ya que se piensa que es exclusivo a la interfaz de usuario, siendo esta una visión errónea, porque está relacionado con la estructura general del sistema y en el concepto base del mismo (Ferré, 2003).

La mayoría de estos problemas son resueltos si se integran las tareas propias del diseño HCI en el proceso de desarrollo de software desde etapas tempranas. De esta manera se pueden vincular las actividades de ambos procesos, como se muestra en la Figura 1. (Xavier, 2003).

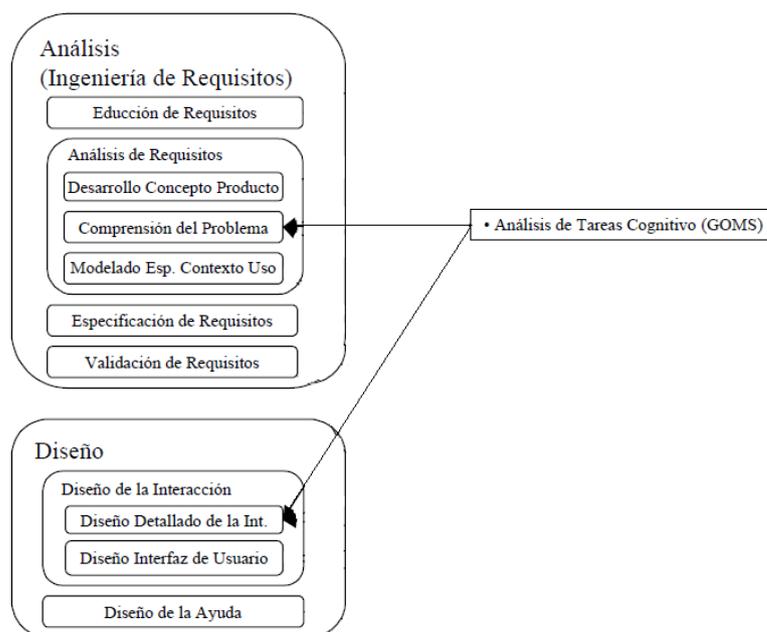


Figura 1. Asignación de técnicas de usabilidad a actividades del desarrollo (Ferré, 2003).

El “análisis de tareas” es una actividad que se ejecuta en ambas fases del proceso de desarrollo software (ver Figura 1.). Sin embargo, el enfoque podría variar. El presente proyecto se centra justamente en el análisis de tareas desde la perspectiva del diseño del modelo de interacción humano computador.

MARIA de las siglas en inglés: (Model-based IAnLanguage foR Interactive Applications) es el lenguaje basado en modelos que ayuda a incrementar el detalle en la fase de análisis de tareas del proceso de desarrollo del modelo HCI (ISTI, 2010). Es un lenguaje con múltiples niveles de abstracción que interactúa con aplicaciones indistintas de su ambiente de desarrollos.

MARIA es el lenguaje caracterizado por ser orientado a servicios, con una amplia interacción con cualquier tipo de dispositivos y necesita de un entorno de desarrollo como MARIA ENVIRONMENT que es la herramienta utilizada para diseñar modelos de interfaz de usuario para el diseño y desarrollo de aplicaciones interactivas basadas en servicios Web (Fabio Paterno', 2008).

1.2. PLANTEAMIENTO DEL PROBLEMA

El desarrollo Software ha mejorado de manera significativa en el transcurso de los años. Sin embargo, la no inclusión de la usabilidad en la fase de diseño y análisis del desarrollo software ha desencadenado serios problemas en la interacción humano computador; de esta manera ha disminuido la calidad y satisfacción del cliente al momento de usar el software (Ferré, 2003).

Debido a que erróneamente se asocia al desarrollo de HCI con la interfaz gráfica de los sistemas, se ha creado una brecha en la integración con la ingeniería de software. Dado el desconocimiento de la aplicación de diversas técnicas de usabilidad, se ha creado aplicativos difíciles de usar y que han necesitado mayor inversión en soporte para satisfacer las necesidades del usuario (Xavier, 2003).

En el proceso de desarrollo de software se realiza el análisis y modelado de tareas pero se orienta a las transacciones, dejando de lado la interacción entre el usuario y el sistema. Este problema desemboca en sistemas que transaccionalmente pueden ser eficientes, pero usablemente son deficientes, lo cual evidencia la necesidad de realizar un modelado de tareas teniendo en cuenta la interacción con los usuarios; es decir, que permitan modelar la interacción que va a existir entre el sistema y el usuario.

El uso de un lenguaje de modelado de tareas orientado a la interacción humano computador podría contribuir a realizar una evaluación y reducción de errores humanos, proporcionando como mínimo un conjunto de técnicas de análisis de tareas, una descripción de los aspectos observables del comportamiento del operador en varios niveles de detalle, junto con algunas indicaciones de la estructura del sistema.

El lenguaje de modelado se centra en expresar la semántica de alto nivel de la interacción: esto se aplica principalmente mediante el uso de modelos de tarea, y asociado a herramientas destinadas a expresar las actividades que los usuarios de la aplicación deben realizar con su interacción (Fabio Paterno', 2008).

1.3. JUSTIFICACIÓN

La realización de un estudio descriptivo del lenguaje de modelado de tareas MARIA y su aplicación en el proceso de desarrollo del de HCI, se justifica en la elaboración de un prototipo que demuestre la práctica del lenguaje en la herramienta MARIA ENVIRONMENT, debido a que es el entorno de desarrollo utilizado para modelar el lenguaje MARIA. En este contexto contar con un estudio acerca de este lenguaje basado en modelos que intervengan en parte del proceso de desarrollo de software, permitirá incrementar la calidad, de productos software desarrollados.

Este lenguaje permite un alto nivel de abstracción, obteniéndose una fase de análisis mucho más detallada, lo cual simplifica el proceso de construcción de software. Además, posee una descripción lógica de interfaz de usuario que permite a los desarrolladores hacer frente a una gran cantidad de detalles de bajo nivel, asociado con los diferentes tipos de lenguajes correspondientes.

MARIA ENVIRONMENT es capaz de modelar tareas e importar automáticamente servicios, realizar anotaciones de descripciones y apoyar a la asociación interactiva de tareas básicas del sistema con las operaciones de servicios Web. También realiza una serie de transformaciones semiautomáticas siendo capaz de explotar la información en este tipo de descripción de servicios independientemente de los dispositivos que se pueden utilizar.

1.4. OBJETIVOS

1.4.1. Objetivo General

Realizar una investigación descriptiva acerca del lenguaje basado en modelos “MARIA” para el análisis de tareas en el proceso de diseño del modelo de interacción humano computadora realizando una aplicación en el entorno MARIA ENVIRONMENT.

1.4.2. Objetivos Específicos

- Realizar una revisión de literatura del lenguaje de modelado de tareas MARIA.
- Realizar el análisis de las características del lenguaje de modelado de tareas MARIA.
- Aplicar el lenguaje de modelado MARIA en un prototipo desarrollado en MARIA ENVIRONMENT.

1.5. ALCANCE

Se realizará un levantamiento de información para crear una revisión de literatura del lenguaje de modelado de tareas MARIA, la cual permitirá realizar un análisis de cada una las características del lenguaje de modelado.

La información necesaria para el desarrollo de tareas HCI se basará en el análisis preliminar de la literatura, en miras de mejorar la integración de este enfoque en el proceso de desarrollo del software.

Como fase final, se elaborará la aplicación de lenguaje de modelado “MARIA” utilizando la herramienta Case “MARIA ENVIRONMENT” en la cual se evidenciará la técnica de análisis de Tareas de HCI.

CAPÍTULO 2

MARCO TEÓRICO

2.1. METODOLOGÍA DE INVESTIGACIÓN

Las técnicas metodológicas de la investigación, son pautas necesarias para desarrollar el proceso de investigación de mejor manera, es necesario orientar cada uno de los procedimientos a cualquier proyecto educativo que se desee realizar (Moreno Flórez, 2005).

Es así como la investigación se centra directamente en un estudio aplicado, teniendo como objetivo principal la resolución de un problema inmediato.

En esta sección se explica en detalle la metodología de investigación necesaria para realizar el estudio del Lenguaje de Modelado MARIA y de esta manera aplicar la técnica de análisis de tareas en el proceso de interacción Humano Computador.

2.1.1. INVESTIGACIÓN DESCRIPTIVA

El Objetivo de la investigación descriptiva es llegar a conocer las situaciones, costumbres y actitudes predominantes a través de la descripción exacta de las actividades, objetos, procesos y personas (Sánchez, 2004).

2.1.1.1. ETAPAS

A continuación se describen las etapas que caracterizan a la investigación descriptiva:

- Examinar las características del problema escogido.

- Definir y formular la hipótesis.
- Enunciar los supuestos en que se basan las hipótesis y los procesos adoptados.
- Elegir los temas y las fuentes apropiadas.
- Seleccionar o elaborar técnicas para la recolección de datos.
- Clasificar los datos, que se adecuen al propósito del estudio y que permitan poner de manifiesto las semejanzas, diferencias y relaciones significativas.
- Verificar la validez de las técnicas empleadas para la recolección de datos.
- Realizar observaciones objetivas y exactas.
- Describir, analizar e interpretar los datos obtenidos en términos claros y precisos.

La investigación descriptiva consiste en indagar propiedades, características y rasgos importantes de cualquier fenómeno que se analice (Hernández, Fernández y Baptista, 2003, p. 119).

Los estudios descriptivos miden de manera más independiente los conceptos o variables a los que se refieren y se centran en medir con mayor precisión (Hernández et al., 2003).

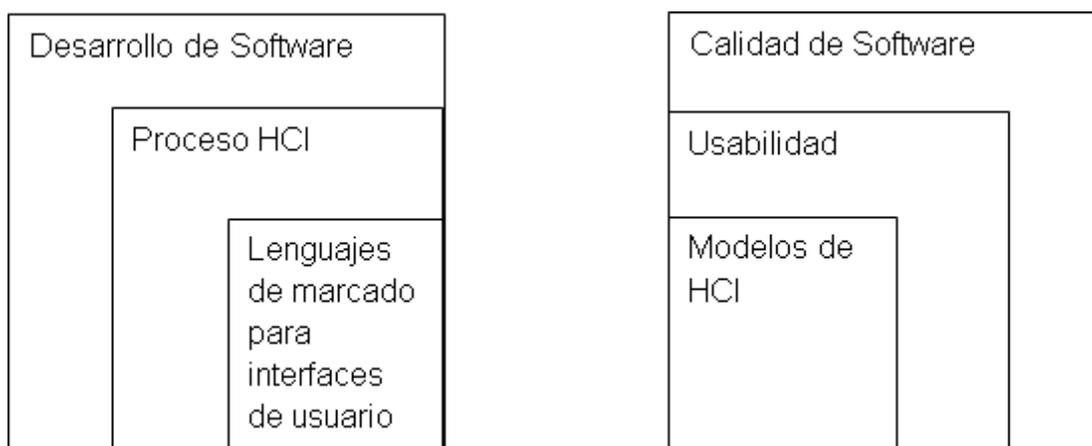
Este tipo de investigación somete a un análisis en el que se mide y evalúa diversos aspectos del problema a investigar.

2.2. RED DE CATEGORÍAS

Para buscar la razón en la fundamentación teórica de la presente investigación, es necesario elaborar una red de las principales categorías que intervienen en el análisis y resolución científica del tema a desarrollarse; se la presenta en la Figura 2.

Variable Independiente

Variable Dependiente



Pasos para aplicar la Técnica

de Análisis de Tareas:

1. Técnicas para la mejora de la Usabilidad
2. Modelado de Tareas
3. CONCURTASKTREES (CTT)

Proceso de investigación del

Lenguaje de Modelado MARIA:

1. Introducción
2. Interfaz de Usuario Abstracto.
3. Interfaz concreta de Usuario de Escritorio
4. Interfaz concreta de Usuario Vocal

Figura 2. Red de Categorías de las Variables de Investigación

2.3. USABILIDAD Y LA INTERACCIÓN HUMANO COMPUTADOR

La presente sección trata sobre la vinculación de la usabilidad dentro del desarrollo software y la importancia de HCI en la misma.

Además se analiza como el modelo de análisis de tareas ayudaría a tener mejores resultados en la captura de los requisitos funcionales de los sistemas software y el desarrollo de la interacción del sistema.

En la actualidad, cada vez hay más presión sobre los desarrolladores para producir sistemas más usables, lo que requiere el uso de métodos apropiados para apoyar a un mejor diseño durante el desarrollo (Xavier Ferre, 2010).

Usabilidad se la define en el estándar ISO 9241 como “el grado en el que un producto puede ser utilizado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un determinado contexto de uso” [ISO98b], y en el estándar ISO 14598-1 [ISO98a] se define calidad de uso de forma análoga (Grau, 2003).

En conclusión se la puede considerar como el nivel de satisfacción que tiene el cliente a un determinado sistema, el grado con el cuál el usuario puede interactuar con el sistema.

2.3.1. Relación entre la Usabilidad y la Interacción humano computador

En el desarrollo software comúnmente se asocia a la usabilidad con características de la interfaz gráfica de usuario, como interfaces basada en ventanas, colores, formas y animaciones. Sin embargo, la usabilidad no está definida en la interfaz gráfica. La usabilidad de un sistema está ligada principalmente a la interacción del mismo, al modo que se realizan las operaciones con el sistema (Grau, 2003).

La usabilidad se ocupa también del entorno del sistema software, ayudando a obtener una documentación mucho más apegada al usuario y al proceso de implementación del sistema (Grau, 2003).

2.3.2. Influencia en el Ciclo de Vida Software

Como se ha mencionado anteriormente, la usabilidad es parte fundamental en el desarrollo software. Es así que los recursos deben ser aplicados para

que el producto software sea cada vez más amigable al usuario, desde el inicio hasta el final de su proceso de elaboración. El uso de la técnica de prototipado hace que los sistemas sean más usables, de esta manera el usuario interviene más en el proceso de desarrollo software. Cuya finalidad es tener un producto que haga al usuario más productivo, aumentando su eficiencia y satisfacción al usarlo (Grau, 2003).

La ingeniería de software, así como también la HCI están preocupados con el diseño del producto software y a pesar que poseen un interés común, tienen orígenes en su teoría totalmente diferentes. En la ingeniería de software hace uso de técnicas de generación de modelos, para mejorar el desempeño del diseño, mientras que en la HCI hace uso de escenarios como prototipos para obtener un diseño más real (Sutcliffe, 2009).

2.3.3. Técnicas para la mejora de la Usabilidad

En esta sección se menciona las técnicas más relevantes que ayudan a conseguir un mayor nivel de usabilidad en un producto desarrollado.

Las técnicas que se mencionan a continuación van a agruparse a las fases del ciclo de desarrollo software:

- Especificaciones: Análisis de usuarios, **análisis de tareas y especificaciones de usabilidad.**
- Diseño: Es el diseño de la interacción, prototipado y participación de usuarios.
- Evaluación: Test de usabilidad y evaluación heurística.

Cada técnica tiene su concepto motivación o procedimiento que debe seguir y su relación si corresponde. La técnica a describirse es Análisis de Tareas ya que es el tema que abarca la presente investigación. Sin embargo previamente se mencionaron las diversas técnicas que existen.

Es preferible usar las técnicas de la usabilidad desde el inicio hasta el final del desarrollo; con ello se logra un producto más efectivo (Simarro, 2007).

2.4. ANÁLISIS DE TAREAS

La definición formal dice que, describe un conjunto de técnicas que se preocupa de como hace la gente para realizar una determinada tarea (Grau, 2003).

La técnica análisis de tareas posee una diferencia desde el punto de vista del desarrollo de interfaces, que de la ingeniería de software, que es necesario una mayor interconexión con el usuario para así obtener un producto más amigable (Xavier, 2003).

Cada técnica de usabilidad tiene un momento adecuado a ser utilizada, en este caso la técnica de análisis de tareas interviene en la comprensión del problema de la fase de análisis y en la construcción detallada de la interacción, establecidas en la fase de diseño de desarrollo de software (Xavier, 2003).

La técnica de análisis de tareas permite una mayor subdivisión y requiere una integración más delicada con las actividades de la ingeniería de software (Xavier, 2003).

2.5. MODELADO DE TAREAS

Los modelos de tareas describen las actividades lógicas realizadas por el usuario cuando interactúe con el sistema.

También es utilizado para el análisis de requisitos de la interfaz, para que la aplicación sea aceptada y brinde un nivel de satisfacción aceptable por el usuario. El modelado de tareas para el diseño y desarrollo de interfaces de usuario consigue un nivel de abstracción superior, haciendo que el desarrollo

sea un proceso de ingeniería, es decir un software de calidad orientado a diseñar interfaces centrados al usuario (Simarro, 2007).

El proceso de análisis de tareas posee dos partes importantes. La primera es la recolección de datos necesarios para comprender las actividades que realiza el usuario. La segunda es la representación de esta información en un modelo. Estas partes nos dan una descripción de acciones que debe realizar el usuario para llegar al objetivo.

Existen varios métodos para el análisis de tareas dependiendo de la finalidad del estudio, los mismos son clasificados en 3 grandes grupos: cognitivos, predictivos y descriptivos (Simarro, 2007).

Los métodos cognitivos representan el tipo de conocimiento que debe tener el usuario para identificar el comportamiento correcto con el sistema. Los métodos predictivos describen secuencias de comportamiento, finalmente permiten hacer una descripción del sistema a partir de la información obtenida (Simarro, 2007).

Algunos ejemplos de métodos de análisis de tareas son: Hierarchical Task Analysis (HTA), Goal-Operations-Methods-Selection (GOMS), Task Action Grammar (TAG), User Action Notation (UAN) y la notación ConcurTaskTrees (CTT) que es la más reciente y en la cual se basa MARIA (Simarro, 2007).

2.5.1. USER ACTION NOTATION (UAN)

Es un tipo de notación de bajo nivel, el mismo que permite una descripción de las acciones realizadas en proceso de interacción usuario computador. Representa el comportamiento del usuario y de la interfaz cuando se realiza una tarea. Por ejemplo pueden ser: arrastrar, apuntar, pinchar determinadas interfaces (Hartson, 1992).

2.5.2. GOALS, OPERATOR, METHODS, SELECTION RULES (GOMS)

Esta técnica sirve para modelar y describir las tareas permitiendo formalizar tanto las actividades físicas, como mentales que intervienen en el proceso de resolución de problemas del ser humano. Cada tarea describe los objetivos del usuario en un lenguaje común, los operadores describen las acciones que el software permite que los usuarios lleven a cabo, métodos y las reglas a utilizar en determinada situación (Card, P., & Newell, 1983).

2.5.3. TASK ACTION GRAMMAR (TAG)

Se utiliza para representar el conocimiento del usuario, descrito para realizar una tarea basada en subtareas con diferentes características. El conocimiento del usuario se puede expresar mediante un esquema que engloba un conjunto de reglas individuales. Los esquemas están formados por una estructura sintáctica definida por un conjunto de características y éstas son atributos que definen la semántica de la tarea (Payne, 1986).

2.5.4. CONCURTASKTREES (CTT)

Es una notación desarrollada por Fabio Paternò, permite representar relaciones temporales entre las tareas y los usuarios para lograr llevar a cabo las mismas. Modela las tareas que un usuario debe realizar en una aplicación interactiva. Caracterizado por tener una estructura jerárquica, sintaxis gráfica muy fácil de interpretar y un gran conjunto de operadores que ayuda a explicar las relaciones.

En CTT, se identifican 4 tipos de tareas, dependiendo del actor que las realiza:

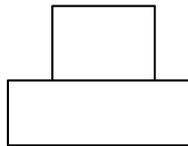
- **Tareas de Usuario:** Son aquellas que son realizadas completamente por el Usuario, que no interactúan con el sistema.

Notación Gráfica



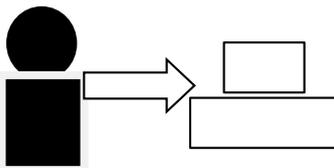
- **Tareas de aplicación:** Son tareas realizadas por el sistema.

Notación Gráfica



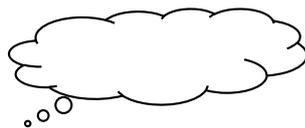
- **Tareas de interacción:** Son tareas definidas por el usuario que interviene con el sistema por medio de cualquier técnica de interacción.

Notación Gráfica



- **Tareas abstractas:** Son aquellas que requieren acciones complejas y que por su complejidad es incierto descubrir donde se van a realizar. Las cuales son descompuestas en subtareas.

Notación Gráfica



Para su descripción se utiliza una serie de operadores existentes en LOTOS², que facilitan las relaciones entre tareas. El uso de estos operadores hace que se facilite la descripción de los comportamientos complejos (Paternò, 2000).

- **T1 ||| T2. Entrelazado (Concurrencia independiente)** Las dos tareas pueden realizarse en cualquier momento, independientemente de su orden.
- **T1 [|]| T2. Sincronización (Concurrencia con intercambio de información).** Las dos tareas tienen que sincronizarse en alguna de sus acciones para intercambiar la información.
- **T1 >> T2. Activar (enabling).** Cuando termina la primera tarea (T1), procede activarse la segunda tarea (T2). De manera secuencial.
- **T1 []>> T2. Activar con paso de información.** Cuando termina la primera tarea (T1) genera algún valor que se envía al T2 antes de ser activada.
- **T1 [] T2. Elección.** Selección alternativa de las dos tareas. Cuando se selecciona una tarea la segunda no puede ser activada por lo menos hasta que termina su ejecución.
- **T1 [> T2. Desactivación.** Cuando se ejecuta la primera acción de la tarea T2, la tarea T1 se desactiva.
- **T1*. Iteración.** La tarea T1 se realiza de manera repetitiva, mientras otra tarea desactive su ejecución.
- **T1(n). Iteración finita.** La tarea T1 se realiza n veces, este número limitado lo define el diseñador.

² LOTOS es una notación formal concurrente, usada para poder especificar interfaces de usuario, ya que permite describir comportamientos manejados por eventos y modificaciones de estado (Santoro, 2005).

- **[T1]. Tarea Opcional.** No es de carácter obligatorio la realización de la tarea.

Como ejemplo en la Figura 3., se evidencia la utilización de los ConcurTaskTrees cuya aplicación sirve para acceder a la información de un museo (Simarro, 2007).

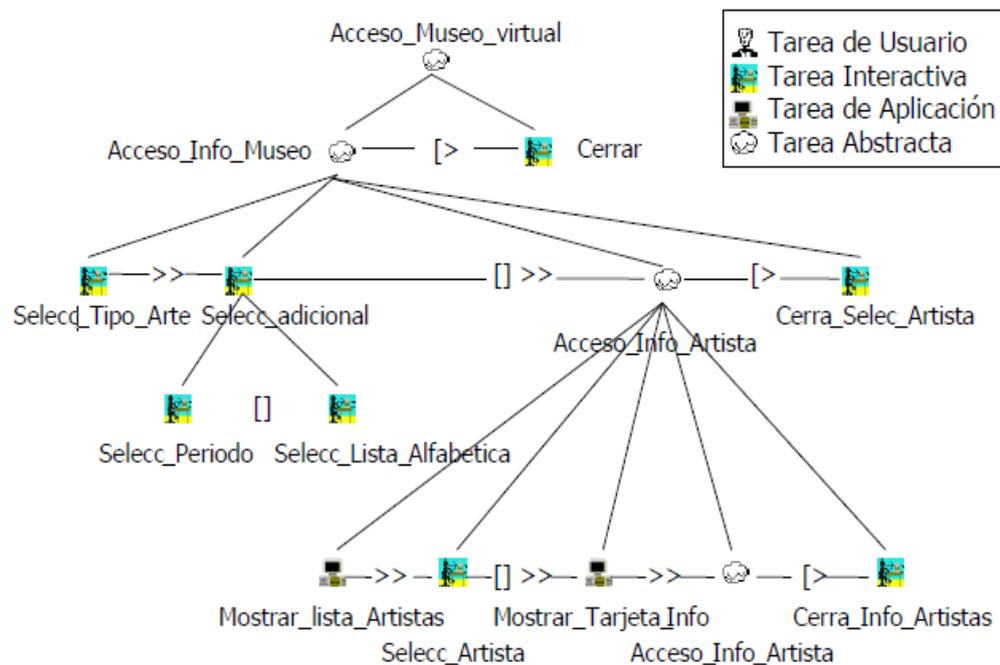


Figura 3. Tarea de acceso a un museo mediante CTT (Zumaquero, 2008)

2.6. LENGUAJES DE MARCADO PARA INTERFACES DE USUARIO

Los lenguajes de marcado de interfaz de usuario permiten construir las interfaces gráficas de usuario y sus controles. Muchos de estos lenguajes son dialectos de XML, es decir son parte del lenguaje XML y dependen de un motor de lenguaje de scripting preexistente.

En esta sección se describe las características de los lenguajes de mayor uso más representativos:

- UIML
- USIXML
- MARIA
- TERESA

UIML

De las siglas en inglés (User Interface Markup Language) (ConstantinosPhanouriou & MarcAbrams, 2002), es un lenguaje XML que permite hacer una descripción declarativa de una interfaz de usuario, de una manera altamente independiente del dispositivo.

Su objetivo es permitir a un documento UIML ser mapeado a cualquier tipo de interfaz de usuario, desde una simple interfaz gráfica de usuario hasta una compleja interfaz hablada.

Lamentablemente UIML no ha llamado mucho la atención a los investigadores, por lo tanto no ha sido tomado en cuenta para su desarrollo (ConstantinosPhanouriou & MarcAbrams, 2002).

USIXML

De las siglas en inglés (User Interface eXtensible Markup Language), es un lenguaje de marcado compatible con XML desarrollado en la Universidad de Louvain-la-Neuve - Bélgica, que tiene como objetivo describir la interfaz de usuario para múltiples contextos de uso. USIXML se descompone en varios metamodelos que describen diferentes aspectos de la interfaz de usuario.

USIXML posee enfoque de la modelización de los contenedores como subtipos de objetos de interacción, los mismos que pueden ser engañosos, ya que su objetivo debe ser más para indicar cómo elaborar elementos en lugar de modelar las interacciones individuales. Los autores utilizan transformaciones gráficas para soportar cambios de modelos, que es un

enfoque académico interesante con algunos problemas de rendimiento (Paterno', Santoro, & Spano., 2009).

TERESA

Tiene un enfoque modular para apoyar la descripción de interfaces de usuario abstractas y concretas. Posee un nivel (interfaz de usuario concreta) representado a través de un número de idiomas dependientes de la plataforma, que son refinamientos del lenguaje abstracto. En TERESA hay diferentes tipos de elementos: interactores (descripción de objetos individuales de interacción), operadores de composición (que indican cómo elaborar las interacciones) y presentaciones (que indican los elementos que se pueden percibir en un momento dado, como los de una página Web).

TERESA destacó la necesidad de proporcionar a los diseñadores mayor control de la producción de la interfaz de usuario, que es una característica importante, sobre todo si tenemos en cuenta el creciente aumento de la flexibilidad y complejidad requerido por las nuevas tecnologías.

Hay una necesidad de un modelo de diálogo y la navegación más flexible. Por ejemplo, con TERESA no es posible apoyar diálogos complejos y entradas paralelas (Paterno', Santoro, & Spano, 2009).

MARIA

Es un Framework que apoya la descripción de interfaces de usuario en niveles concretos y abstractos. El lenguaje abstracto trabaja independientemente de la plataforma de interacción. MARIA trabaja con un número concreto de lenguajes y además proporciona un perfeccionamiento de la descripción del lenguaje abstracto en diversas plataformas (Faedo", 2014).

MARIA posee más características sobre los demás lenguajes descritos anteriormente. MARIA hace referencia a los tipos de interfaces de usuario, las técnicas para desarrollar un modelo basado en tareas, mismos que ayudan a

obtener un prototipo de interfaz de usuario más interactivo (Paternò et al., 2012).

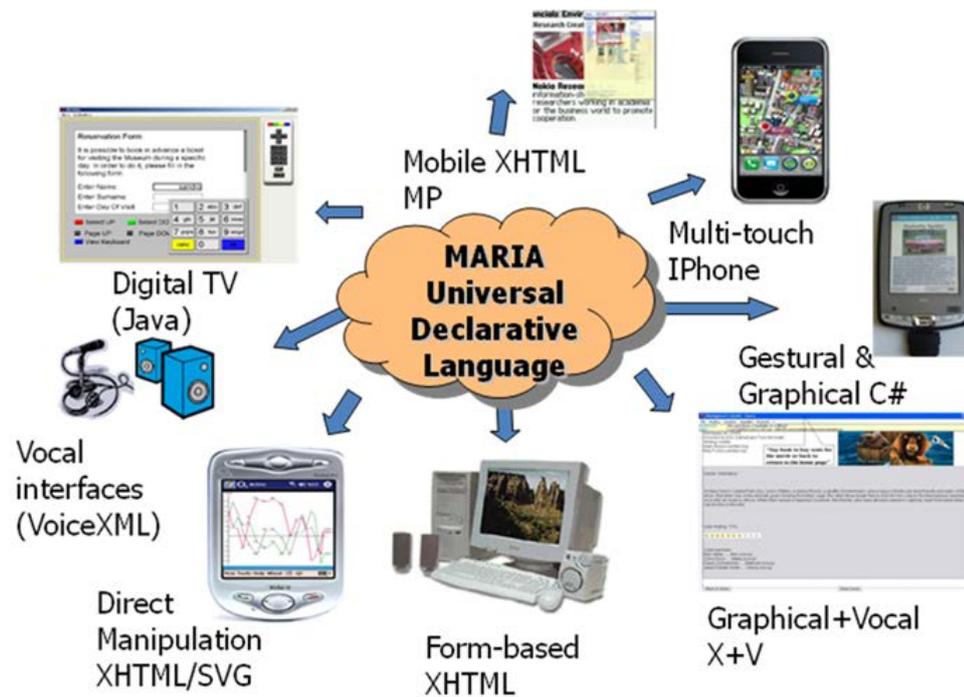


Figura 4. Beneficios al utilizar el lenguaje MARIA (Faedo", 2014)

De tal manera MARIA es el lenguaje que abarca mayor cantidad de características como se puede observar en la Tabla 1.

En el capítulo 3 se describe más en detalle este lenguaje orientado a solucionar la problemática abordada.

Tabla 1

Representación de los lenguajes de marcado con sus características relevantes.

LENGUAJES DE MARCADO PARA INTERFACES DE USUARIO	UIML	USIXML	MARIA	TERESA
Soporta niveles de abstracción		X	X	X
Modelo basado en Tareas			X	
Prototipo de alto nivel de interactividad			X	X
Multipataforma	X		X	

CAPÍTULO 3

LENGUAJE DE MODELADO MARIA, SU VINCULACIÓN CON EL DESARROLLO Y LA USABILIDAD DEL SOFTWARE

3.1 INTRODUCCIÓN

Como se indica en el Capítulo 2, el lenguaje basado en modelos MARIA presenta las características más adecuadas para desarrollar prototipos interactivos partiendo de un modelado de Tareas.

MARIA de las siglas en inglés (Model-based ILanguage foR Interactive Applications) es un lenguaje basado en modelos, universal, declarativo y de múltiple nivel de abstracción. Durante el proceso de modelado, MARIA hace uso del lenguaje XML, de esta manera obtiene aplicaciones interactivas en múltiples ambientes. Es así como MARIA, ha sido escogido como el mejor lenguaje de modelado, sobre UIML, UsiXML y TERESA (Paterno' F. , Santoro, Spano, & D., 2009).

El lenguaje de modelado MARIA, presenta como principal ventaja a los diseñadores de interfaces de usuario de multi-dispositivos, que no precisan aprender los detalles de cada lenguaje de programación al momento de una implementación. Para lo cual MARIA, usa razonamientos abstractos, sin estar atados a una modalidad de interfaz de usuario en particular. De esta manera se puede profundizar el estudio de la semántica, es decir, directamente el estudio de la interacción usuario computador, sin profundizar en los detalles y características especiales de cualquier entorno considerado en particular (Paternò, Santoro, & Spano, MARIA (Model-based ILanguage foR Interactive Applications), 2012).

En la Figura 5. se muestra el flujo de MARIA en el entorno MARIA ENVIRONMENT, representando las principales características al utilizar el lenguaje de modelado.

Como se visualiza en el diagrama de Flujo, se desarrolla el modelado de tareas, que es el pilar fundamental en la construcción del prototipo. Con el modelado de tareas se procede a crear cada una de las interfaces previas a la construcción del prototipo final.

Finalmente el prototipo se exporta en código JAVA, para realizar cualquier tipo de modificación necesaria para su correcto funcionamiento.

Los procesos se realizan en el entorno de trabajo (MARIA ENVIRONMENT) exclusivo para el lenguaje de modelado MARIA. En la siguiente sección se menciona la historia y sus características principales.

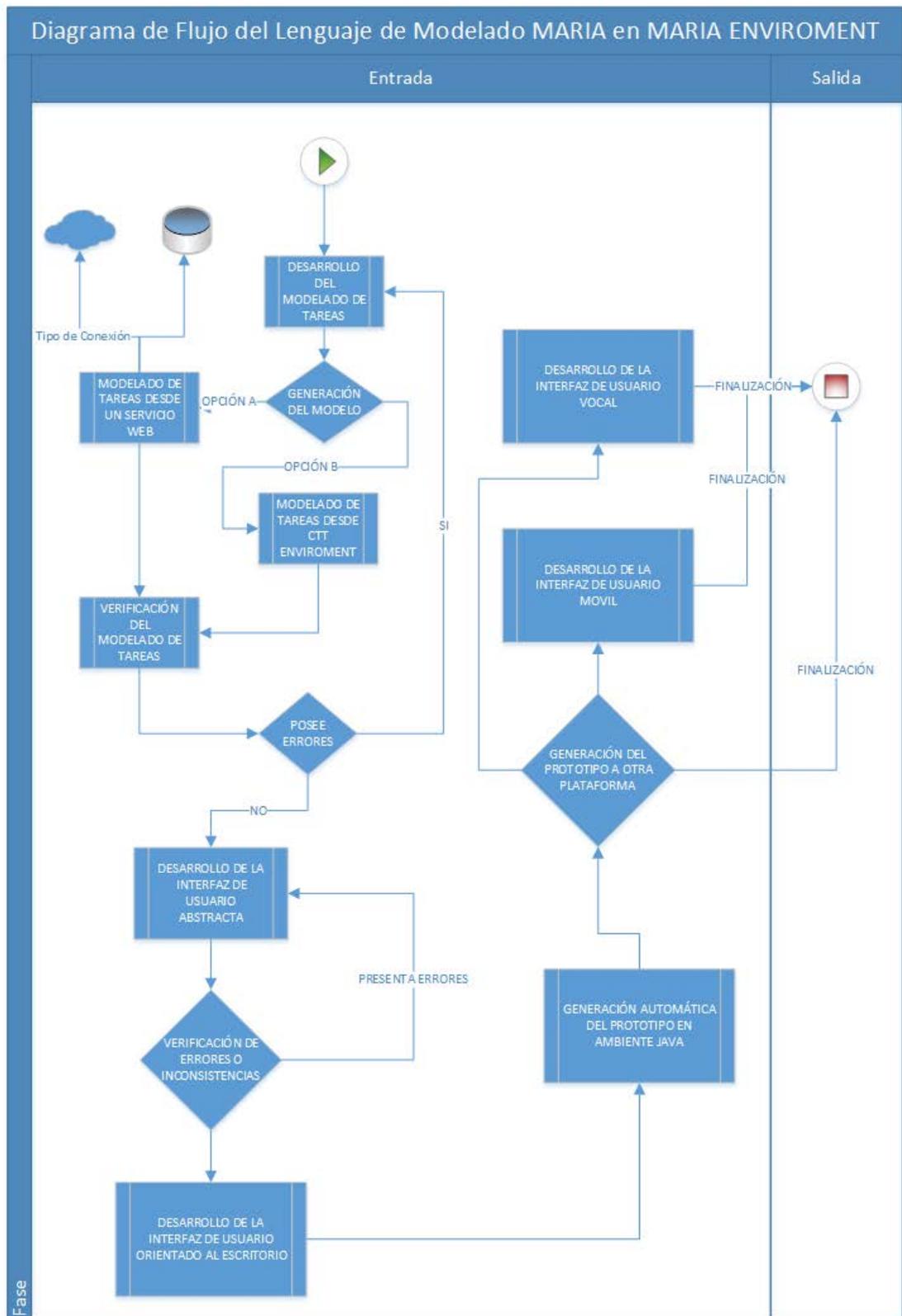


Figura 5. Diagrama de Flujo de MARIA

3.2 ORIGEN Y VERSIONAMIENTO DE MARIA

MARIA es la evolución de TERESA, heredando su enfoque modular. Sin embargo, ha añadido varios cambios que han permitido que MARIA obtenga mayor efecto en el desarrollo de prototipos, que el de su antecesora. MARIA data desde el año 2003, cambiando sus características en el transcurso del tiempo, estos cambios se describen a continuación:

- En su primera versión MARIA permite generar modelos para plataformas basadas en la forma gráfica, TV digital vocal, manipulación directa gráfica, multimodal (gráfica y vocal), para escritorio y móvil y del tipo móvil avanzado (con soporte para multi-touch y acelerómetros, i.n. iPhone) (Paterno' F. , Santoro, Spano, & D., 2009).
- MARIA inicia con la versión 1.0 misma que proporcionaba un mejor apoyo en el desarrollo de interfaces de usuario, ya que utiliza los operadores temporales, tomados de la Notación ConcurTaskTrees (mencionado en la sección del marco teórico), para describir las interacciones paralelas del usuario al computador Jacob et al. [1999].

Actualmente MARIA se encuentra en la versión 1.5.11 permitiendo desarrollar aplicaciones interactivas centradas al usuario en múltiples dispositivos.

3.3 CARACTERÍSTICAS DEL MODELADO DE TAREAS EN MARIA

El Lenguaje de modelado MARIA ofrece soluciones ingeniosas, en cuanto a la explotación de modelos de tareas (representadas en notación ConcurTaskTrees), modelos de interfaz de usuario para el diseño y desarrollo de aplicaciones interactivas basadas en servicios Web.

La herramienta MARIA ENVIRONMENT (basado en MARIA) permite importar automáticamente servicios, anotaciones descriptivas y asociaciones

interactivas de soporte del sistema básico de tareas, con servicios de operación Web. De esta manera, un gran número de procesos están disponibles para explotar la información de servicios y anotaciones.

Si se desea generar una interfaz de usuario abstracta a partir del modelo de tarea, se debe considerar la selección del valor adecuado para el atributo de las “tareas interactivas”. Así mismo, se debe seleccionar el tipo de visualización en el sistema de tareas, para realizar modificaciones en el nivel de presentación (Paternò, Santoro, & Spano, April 2012).

El modelado de tareas en MARIA permite describir no sólo los aspectos de presentación, sino también el comportamiento del modelado de los datos. La definición de la interfaz contiene la descripción de los tipos de datos que se manipulan por la interfaz de usuario. Las interacciones se pueden unir con los elementos del modelado de datos, lo que significa que, en un tiempo de ejecución, la modificación del estado de una interacción también cambiará el valor de los datos de los elementos y viceversa. Este mecanismo permite al modelado la correlación entre los elementos de la interfaz, el diseño condicional de las conexiones condicionadas entre las presentaciones y los valores del formato de entrada. Los datos modelados se definen utilizando las construcciones estándar de una definición de esquema XML (Simarro, 2007).

El modelado de tareas en MARIA es la primera fase para desarrollar en el prototipo. Las tareas son elaboradas con la notación CTT, como se muestra en la Figura 6.

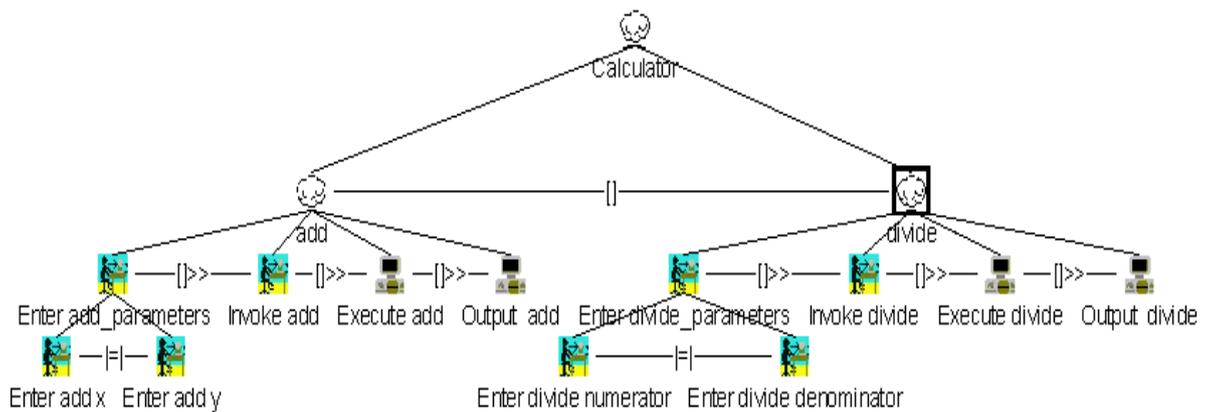


Figura 6. Modelado de tareas utilizando notación CTT

En la Figura 6. se visualiza cada uno de los elementos de la notación CTT, descritos en el marco teórico. Además cada uno de ellos se lo detalla en la elaboración del prototipo.

3.4.1. INTERFAZ DE USUARIO ABSTRACTO

La interfaz de usuario abstracta de las siglas en inglés (Abstract User Interface AUI), describe una interfaz de usuario sólo a través de la semántica de la interacción, sin referirse a una capacidad particular del dispositivo, modalidad de interacción o implementación de tecnología (Paterno', Santoro, & Spano, 2009).

Esta interfaz abstracta está compuesta por varias presentaciones o grupos de elementos del modelo, los cuales son presentados al usuario a la vez. Los elementos del modelo son de dos tipos: usuario que interactúa o composición del usuario que interactúa. Dichas formas representan cada tipo de interacción de usuario, este último agrupa elementos que tienen una relación lógica. De acuerdo a la semántica un usuario que interactúa pertenece a uno de los siguientes subtipos:

- **Selección:** Permite al usuario seleccionar uno o más valores entre los elementos de una predefinida lista, que contiene el valor seleccionado y la información acerca de la lista de elementos conjuntos. Tomando en cuenta el número de valores que son seleccionados en una sola elección o múltiples elecciones.
- **Edición:** Permite al usuario manualmente editar el objeto representado por el usuario que interactúa, el cual puede estar en texto (Edición de texto), un número (Edición numérica), una posición (Edición de posición) o un objeto genérico (Edición de objeto).
- **Control:** Permite al usuario cambiar entre presentaciones (Navegador) o para activar las funciones de interfaz de usuario (Activador).
- **Sólo salida:** Representa la información que se envía al usuario, que no es afectada por las acciones del diseñador. Puede ser un texto de una descripción que representa diferentes tipos de medios de comunicación, una alarma, una evaluación o un objeto genérico.

MARIA XML, permite describir no solamente los aspectos de la presentación sino también el comportamiento del modelo de datos. La definición de la interfaz contiene la descripción de los tipos de datos que son manipulados por la interfaz de usuario. Las interacciones pueden estar ligados con los elementos del modelo de datos, lo cual significa que, en un tiempo de ejecución, modificando el estado de una interacción también cambiará el valor de los elementos de datos ligados y viceversa. Este mecanismo permite al modelamiento de correlación entre los elementos de interfaz de usuario, de las conexiones entre presentaciones, y valores con formato de entrada. El modelo de datos es definido usando el estándar XML Definición de Esquema.

- **Back End Genérico:** La definición de la interfaz contiene un conjunto de Funciones Externas, las cuales representan las funcionalidades explotadas por la interfaz de usuario pero implementadas por una aplicación genérica del apoyo del back-end (ejemplo, servicios web, códigos de biblioteca, base de datos, etc.). Una declaración contiene la

firma de la función externa que especifica su nombre y sus parámetros de entrada/salida.

- **Modelo de evento:** Cada definición de interacción tiene un número de eventos asociados que permiten la especificación de reacción de la interfaz de usuario desencadenada por la interacción de usuario. Dos diferentes clases de eventos han sido identificadas: el Cambio de Evento de Propiedad que especifica el valor de una característica en la interfaz de usuario o en el modelo de datos (con una precondition opcional), y la Activación de Eventos que puede estar elevada por activadores y son destinados a especificar la ejecución de algunas funcionalidades de la aplicación (por ejemplo, la invocación de una función externa).
- **Modelo de diálogo:** Contiene estructuras para especificar la dinámica y comportamiento de una presentación, especificando los eventos que pueden ser activados en un momento dado. Las expresiones de diálogo están conectados mediante operadores CTT con el fin de definir sus relaciones temporales.
- **Actualización continua de campos:** Esto es posible para especificar que un campo dado debe estar periódicamente actualizado invocando una función externa.
- **Conjunto Dinámico de Elementos de Interfaz de Usuario:** El lenguaje contiene construcciones para especificar actualizaciones parciales de presentación (cambiando de forma dinámica el contenido de grupos enteros) y la posibilidad de especificar una navegación condicional entre presentaciones.

En la Figura 7. se puede evidenciar el ejemplo de las características descritas previamente:

En la Figura 7. se visualiza cada uno de los elementos de la INTERFAZ DE USUARIO ABSTRACTO, descritos en la sección anterior. Además cada uno de ellos se lo detallará en la elaboración del prototipo con mucho más detenimiento.

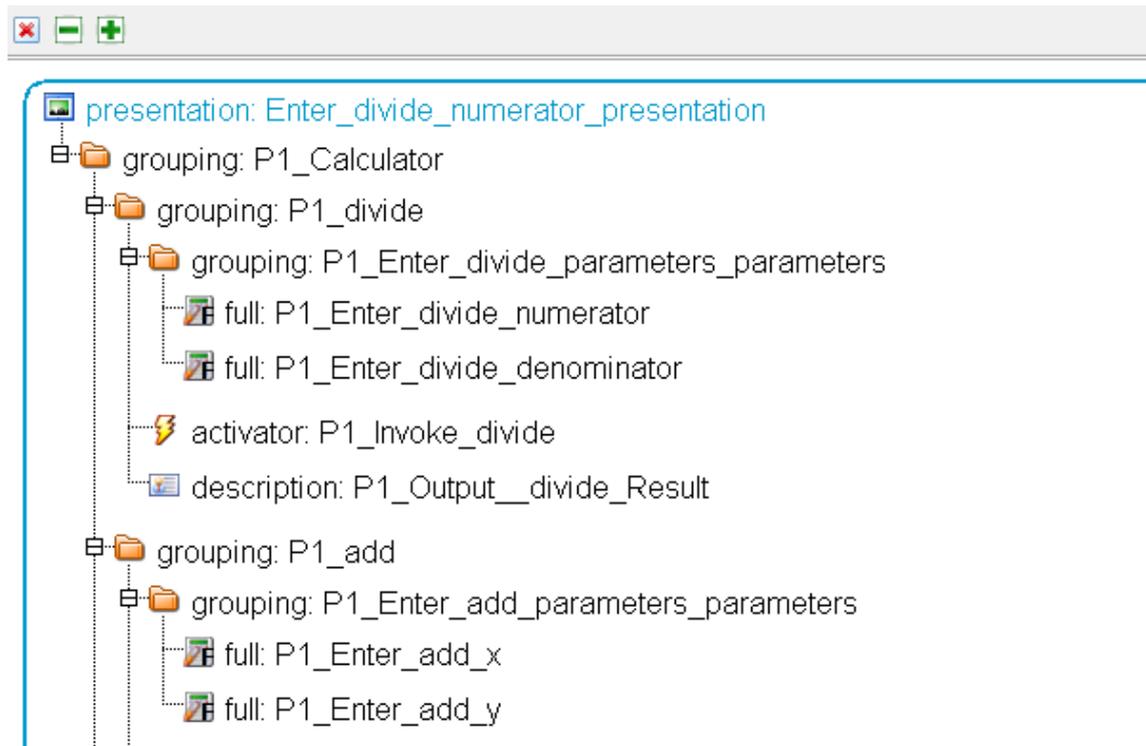


Figura 7. Características gráficas de la interfaz abstracta de usuario

3.4.2. INTERFAZ CENTRADO AL USUARIO ORIENTADO AL ESCRITORIO

Interfaz de Usuario Concreto (IUC) es una meta modelo para una plataforma, es también extensión del meta modelo de Interfaz de Usuario Abstracta (IUA), el cual significa que todas las entidades en la Interfaz de Usuario Abstracta siguen existiendo en la Interfaz de Usuario Concreta. Las extensiones se añaden a la información de plataforma dependiente (pero se sigue implementando un lenguaje independiente) a la estructura del correspondiente modelo de IUA para la misma aplicación de interfaz ya sea agregando atributos o se extiende a través de un mecanismo de herencia de las entidades existentes para la especificación de la posible aplicación concreta de las interacciones abstractas.

En este párrafo se detalla los nuevos atributos utilizados en la meta modelo de IUA para la definición de los Gráficos de Escritorio, mencionados a continuación:

- **Presentación:** Contiene los atributos en los ajustes de presentación, los cuales contienen información sobre el título, fondo (color o imagen) y la fuente utilizada.
- **Agrupamiento:** Contiene los atributos en los ajustes de agrupamiento, los cuales poseen información sobre la técnica de visualización de agrupación (rejilla, fieldset, bala, el color o la imagen de fondo) y si los elementos están relacionados con el pedido o relación jerarquía.

Las clases que han sido extendidas usando la herencia son las siguientes:

- Un Activador puede ser implementado como un botón, un texto_link, imagen_link, imagen_mapa (una imagen con la definición de un conjunto de áreas, cada uno asociado con un valor diferente).
- Una alarma puede ser implementada como un texto (un texto con información de fuente y estilo) o un archivo de audio.
- Una descripción puede ser implementada como un texto, imagen, audio, video o tabla.
- Una elección múltiple puede ser implementada como una caja de verificación o una caja de listas.
- Un navegador puede ser implementado como un link de una imagen, link de un texto, botón, imagen o mapa.
- Una edición numérica completa puede ser implementada como un campo de texto o un cuadro de número (un campo de texto el cual incluya también botones de subida y bajada)
- Una edición de texto puede ser implementada como un campo o área de específica.

En la siguiente Figura 8. se puede evidenciar el ejemplo de las características descritas previamente:

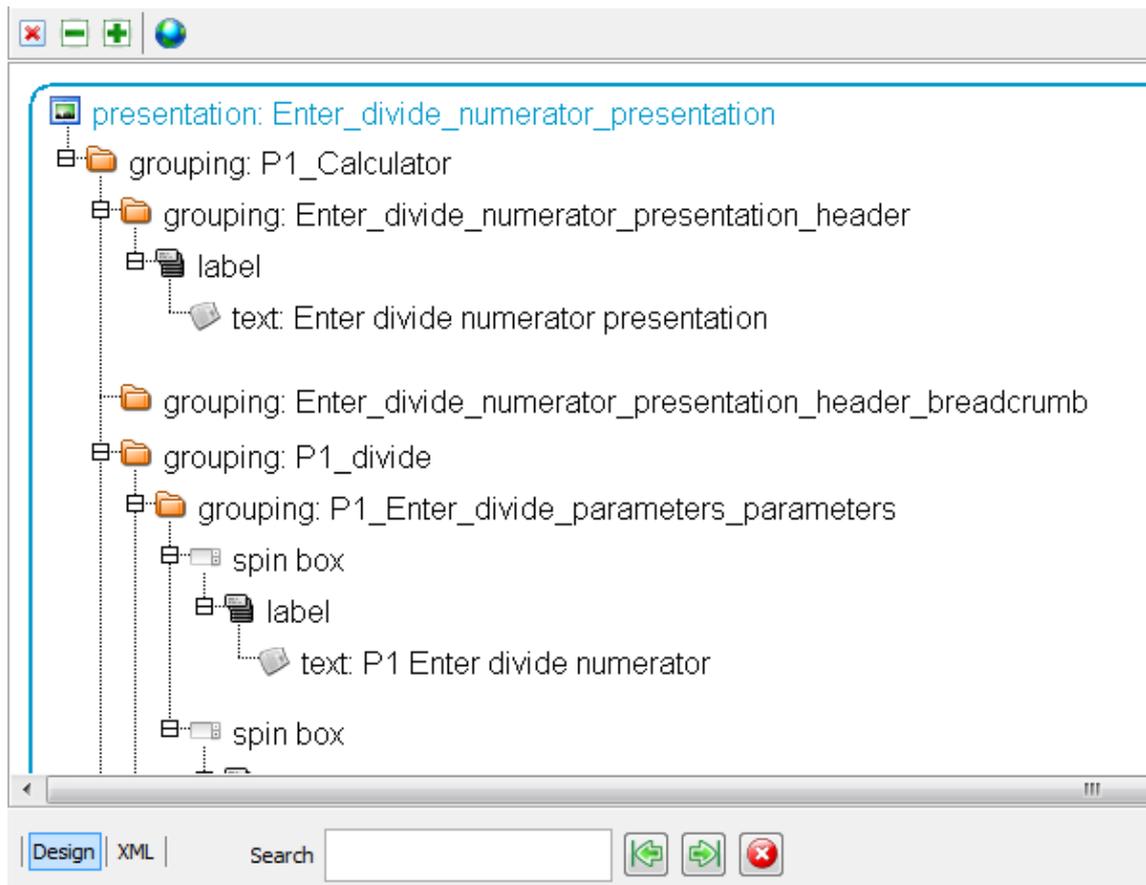


Figura 8. Características gráficas de la interfaz de usuario en el escritorio

Cada refinamiento concreto de las interacciones tiene sus propios atributos para especificar sus disposiciones (bordes, margen, color, etc.) junto con un elemento que especifica la etiqueta, que puede ser texto, imagen o una combinación de ambos (botón, casilla de verificación, elemento elección, la lista desplegable, mapa de imagen, cuadro de lista, icono de e-mail, botón de radio, cuadro de número, área de texto, agrupación, relación, repetidor).

3.4.3. INTERFAZ CENTRADO AL USUARIO DE USO VOCAL

Mientras que en las interfaces gráficas el concepto de presentación pueden ser fácilmente definidas como un conjunto de elementos en interfaces de usuarios perceptible en un momento dado (por ejemplo, una página Web en el contexto), en el caso de interfaces vocales consideramos una presentación

como un conjunto de comunicaciones entre el dispositivo vocal y el usuario, que puede estar considerado como una unidad lógica, por ejemplo un diálogo apoyando a la colección de información sobre un usuario. Al definir el lenguaje vocal hemos refinado el vocabulario abstracto para esta plataforma. Esto significa principalmente, que hemos definido refinamientos vocales de los elementos especificados en el lenguaje abstracto: interactores (elementos de interfaz de usuario), los eventos asociados y sus composiciones.

El refinamiento implica definir algunos elementos que permiten el ajuste de algunas propiedades de presentación. En particular, podemos definir las propiedades predeterminadas de la voz sintetizada (por ejemplo, volumen, tono), el reconocedor de voz (por ejemplo, sensibilidad, nivel de precisión) y los tonos DTMF (Dual-Tone Multi-Frequency).

Sólo las interacciones de producto proporciona una salida para el usuario; la interfaz abstracta los clasifica en texto, descripción, retroalimentación y la salida de alarma. El refinamiento del elemento de texto se compone de dos nuevos elementos: el discurso y el mensaje pregrabado. Discurso define el texto que la plataforma vocal debe sintetizar o la ruta donde la plataforma encuentra los recursos para el texto. Además, es posible establecer una serie de propiedades de voz, como el énfasis, el tono, tasa y el volumen, así como la edad y el sexo de la voz sintetizada. Por otra parte, hemos introducido control del comportamiento en caso de la entrada del usuario inesperado: configurando adecuadamente el elemento llamado barcaza, podemos decidir si el usuario puede detener la síntesis o si la aplicación debe ignorar el evento y continuar. Como se mencionó anteriormente, el otro elemento que refina el texto del resumen pregrabado: define la ruta de los recursos de audio predefinidos que deben ser reproducidos.

Se apoya el caso a la falta de recursos mediante la definición de un contenido alternativo que se puede sintetizar cuando este caso ocurre. Además del discurso y el mensaje pregrabado, el otro elemento de sólo salida es sonido. Estos elementos permiten y definen la trayectoria de un recurso

de audio no vocal que debe desempeñar la plataforma. También es posible insertar una descripción textual del sonido que podría ser utilizado como información adicional sobre el contenido de sonido

En la Figura 9. se puede evidenciar el ejemplo de las características descritas previamente:

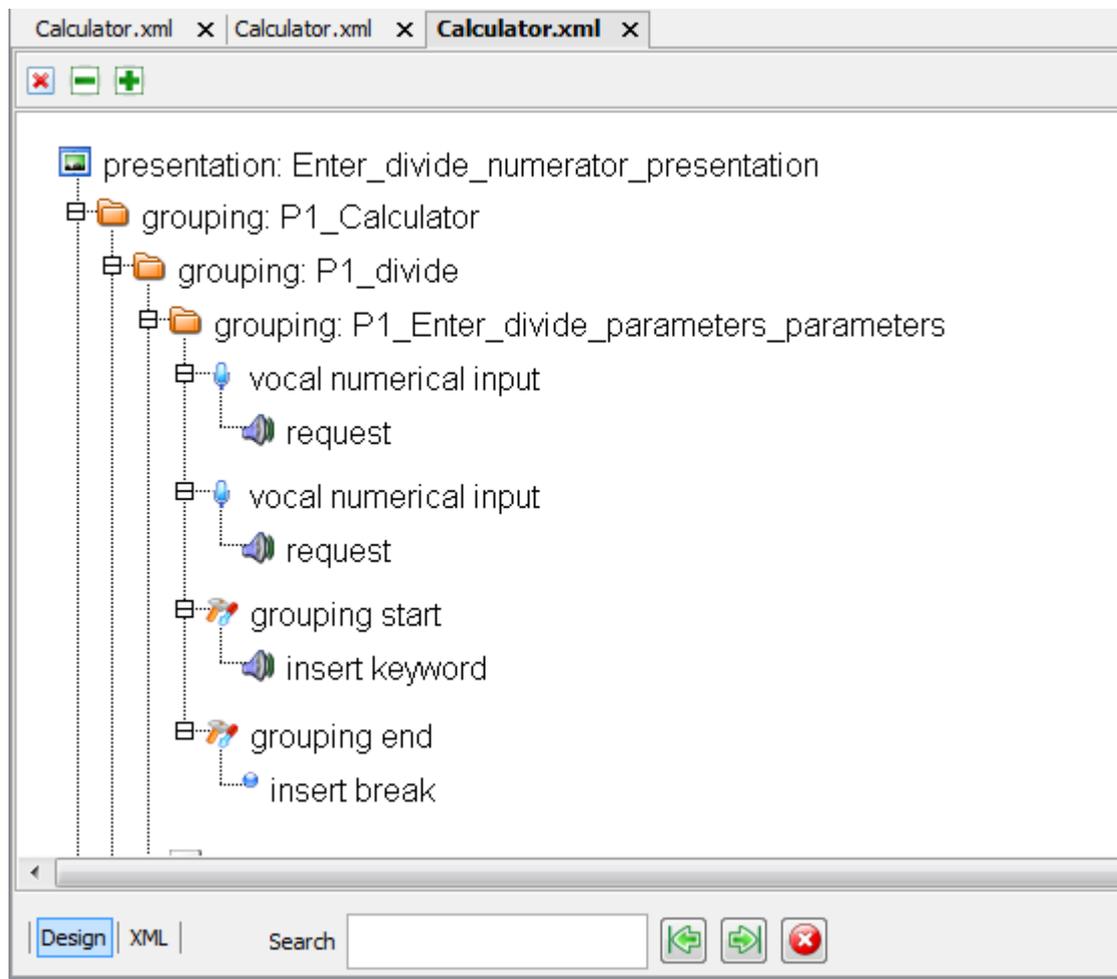


Figura 9. Características graficas de la interfaz vocal de usuario

3.5 INTEGRACIÓN DE LA APLICACIÓN DE MARIA AL DESARROLLO DE SOFTWARE

En esta sección se detalla la integración del prototipo en MARIA ENVIRONMENT, con el desarrollo software.

El desarrollo del modelado de tareas, ayuda a la realización de la fase de análisis de requerimientos del desarrollo software, de tal manera que el modelado de tareas especifica las tareas que va a realizar el usuario con el computador.

La interfaz de usuario abstracta asocia las operaciones y los datos necesarios para el funcionamiento del prototipo, misma que ayuda en el desarrollo a la integración de los datos con la capa de presentación.

Como siguiente, se presenta la interfaz de usuario final, que permite dar el diseño al prototipo que se desea obtener, de esta manera se reduce el tiempo en la modificación de los atributos visuales en el desarrollo software.

Finalmente como utilidad principal se obtiene la posibilidad de generar un prototipo funcional, con su código fuente desarrollado en JAVA, permitiendo al desarrollador realizar cualquier modificación o integración al prototipo generado por MARIA ENVIRONMENT.

Los enfoques basados en modelos son parte fundamental de la HCI. Se basan en una serie de modelos en el que se puede especificar y manipular los aspectos relevantes de una interfaz de usuario, así como para facilitar el trabajo de los diseñadores y desarrolladores. En los últimos años se han construido prototipos en paralelo con el objetivo de hacer frente a los distintos retos planteados por el diseño y el desarrollo de interfaces de usuario en continua evolución y a los ajustes tecnológicos. Podemos identificar una primera generación de enfoques basados en el modelo cuya atención se centró básicamente en abstracciones para las interfaces gráficas de usuario (véase, por ejemplo UÍ [Foley et al., 1994]). En ese momento, los diseñadores de la interfaz de usuario se centraron principalmente en la identificación correspondiente de los aspectos de este tipo de modalidad de interacción.

Entonces, los enfoques evolucionaron a una segunda generación centrándose en expresar la semántica de alto nivel de la interacción: esto era apoyado por el uso de modelos de tarea y asociado a herramientas destinadas

a expresar las actividades que los usuarios de la aplicación desean lograr mientras exista interacción con la aplicación (véase, por ejemplo Adepto [Johnson et al., 1993], GTA [van der Veer et al., 1994], ConcurTaskTrees (CTT) [Paterno, 1999]).

3.6 IDENTIFICACIÓN DE LAS CARACTERÍSTICAS DE MARIA PARA MEJORAR LA USABILIDAD

Una evolución importante en las aplicaciones de software es la propagación de las arquitecturas orientadas a servicios en cualquier tipo de ambiente. Tales ambientes se caracterizan por un amplio conjunto de dispositivos interactivos, con aplicaciones que explotan una serie de funcionalidades desarrolladas de antemano y encapsulados en servicios Web.

En esta sección se mencionan las características principales obtenidas al investigar el uso del lenguaje MARIA en su entorno denominado MARIA ENVIRONMENT.

Como punto principal, MARIA ayuda a la construcción del prototipo partiendo del modelado de tareas, obteniendo una adecuada recolección de datos en la fase de licitación de requerimientos, siendo parte fundamental del desarrollo software. De esta manera obtener una visión mucho más amplia y detallada de lo que se desea elaborar.

Como segundo punto, se integra las operaciones o procesos del servicio web, a cada una de las tareas de interacción obtenidas del modelado de tareas.

En el tercer punto, se exporta el modelado de tareas configurado a la interfaz de usuario abstracta, misma que se encarga de vincular las tareas que realiza el usuario de mejor manera, ya que se especifica cada una de las tareas que van a ser intervenidas por el actor o usuario.

En el cuarto punto, la interfaz de usuario abstracta se exporta a una interfaz más amigable al usuario y diseñador. Misma que se encarga de dar la forma

final al prototipo; integrando color, fondo y material audiovisual que hace que el prototipo sea más intuitivo hacia el usuario. Y es aquí en donde, se enfoca el diseñador a realizar los cambios necesarios para que el prototipo sea visualmente más amigable.

Las conclusiones finales sobre la investigación completa, se las muestran con mucho más énfasis en el capítulo 5 CONCLUSIONES Y RECOMENDACIONES.

CAPÍTULO 4

DESARROLLO DE PROTOTIPO

4.1 INTRODUCCION

En esta sección se presenta la elaboración de dos prototipos, el primero cuya función es realizar las operaciones matemáticas y el segundo consultar información aeroportuaria del mundo, utilizando el modelado de tareas en la herramienta MARIA ENVIRONMENT.

La versión que se encuentra en el repositorio HIIS Laboratory (The Human-Computer Interaction Group) es la 1.5.10, como se muestra en la Figura 10.

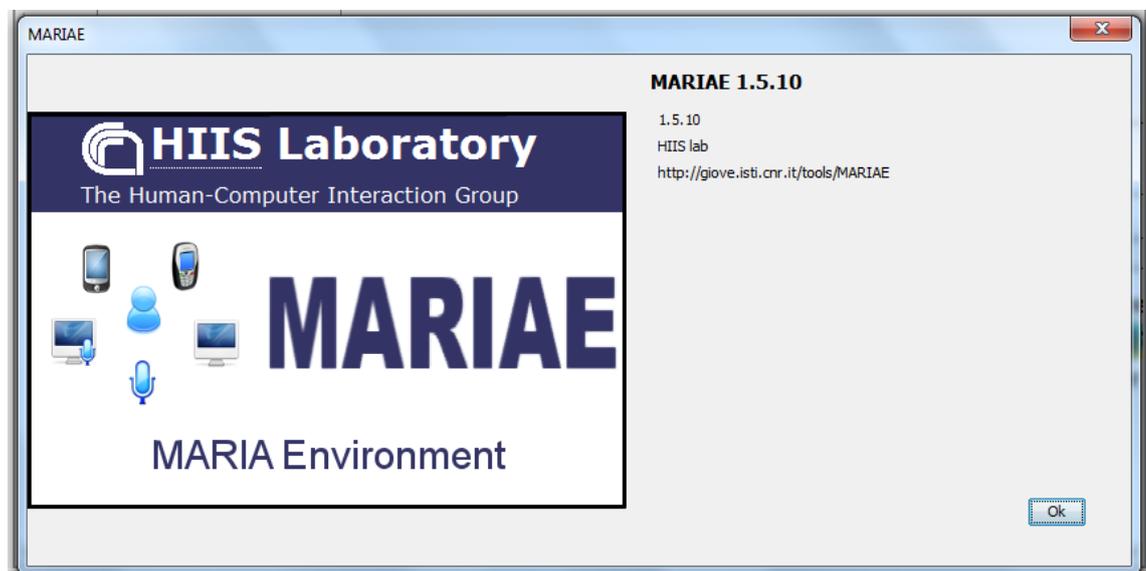


Figura 10. Versionamiento de Maria Environment

La base fundamental para el desarrollo del prototipo, es el modelado de tareas, que se lo puede realizar con la Herramienta ConcurTask Environment CTTE version 2.6.2. El mismo que se encuentra codificado en el lenguaje XML.

4.2 PROTOTIPO OPERACIONES MATEMÁTICAS

Para visualizar de mejor manera el modelado de tareas completo, con cada elemento característico de la notación CTT, se ha procedido a dividir cada una de las ramas, de acuerdo a cada una de las operaciones.

En la Figura 11. se modela la operación de la División.

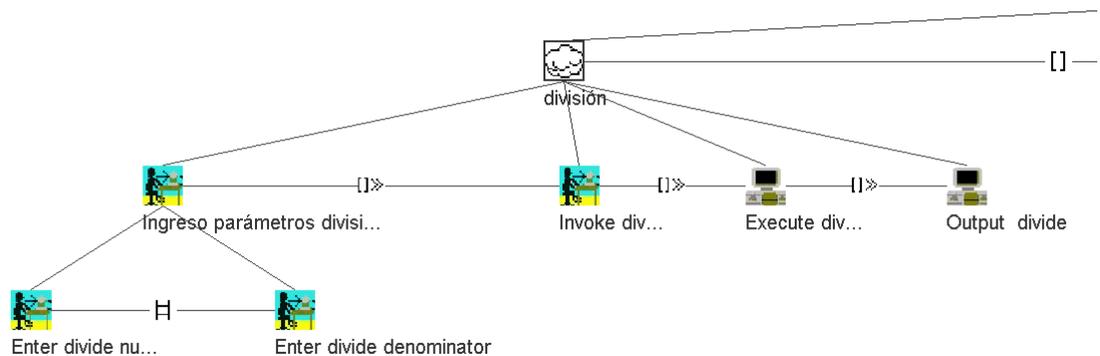


Figura 11. Modelado de tareas de la operación División

El proceso de modelado se lo describe de la siguiente manera, incluyendo cada uno de los operadores. Este ejemplo es detallado con más profundidad para que sirva de ejemplo hacia las demás operaciones.

En la Figura 12. se modela la operación de la Multiplicación.

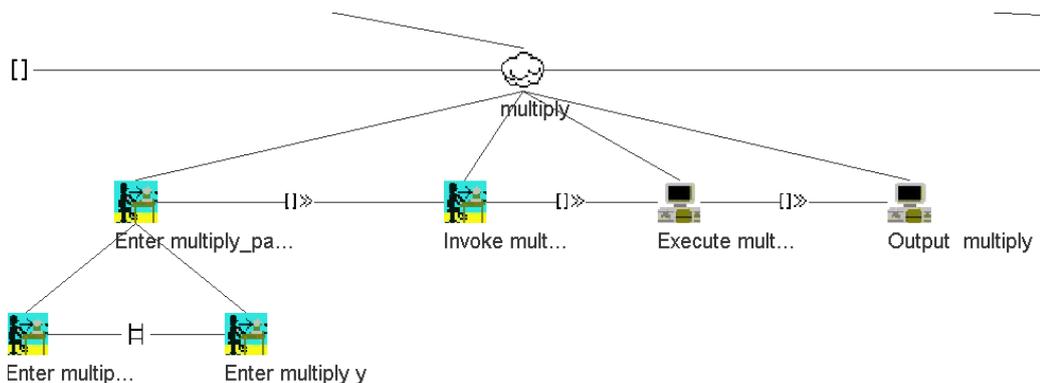


Figura 12. Modelado de tareas de la operación Multiplicación

En la Figura 13. se modela la operación de la Resta.

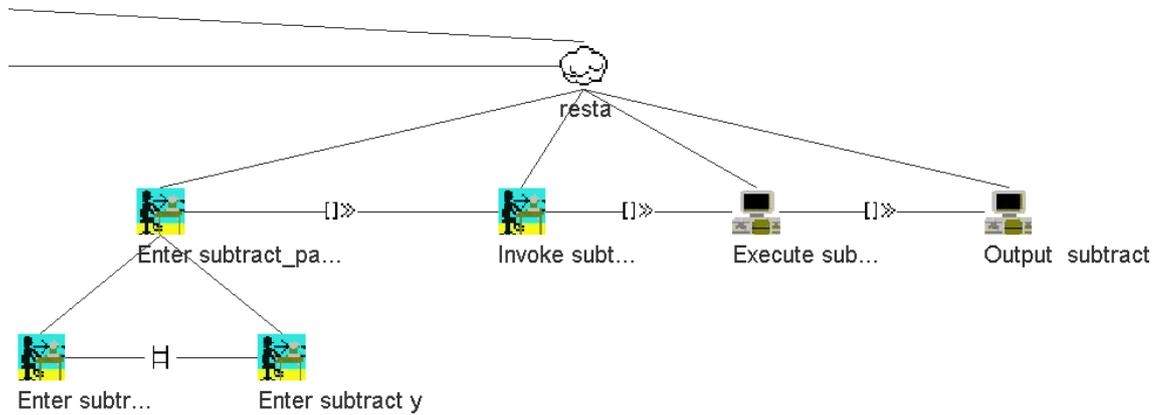


Figura 13. Modelado de tareas de la operación Resta

En la Figura 14. se modela la operación de la Suma

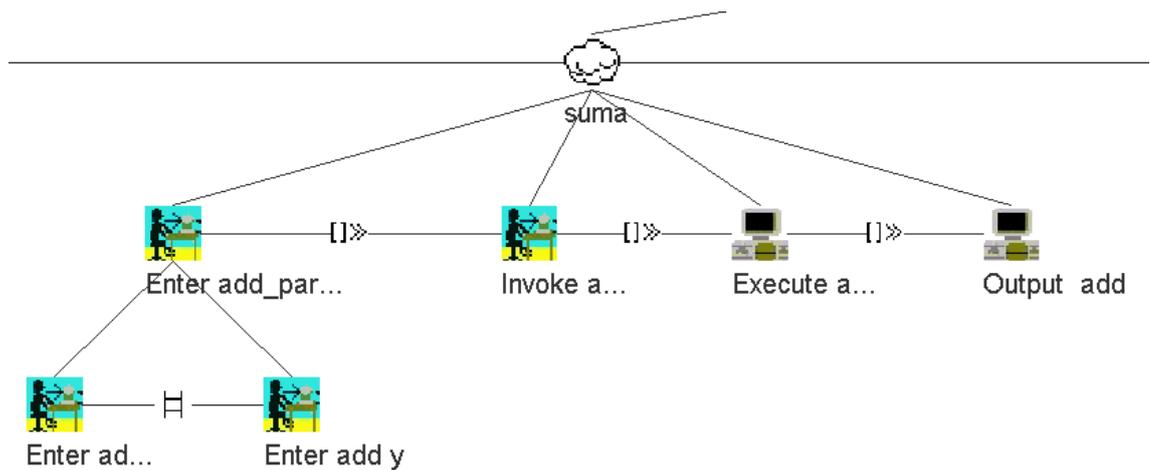


Figura 14. Modelado de tareas de la operación Suma

En la sección anterior se menciona que el modelado de tareas se codifica en lenguaje XML. Sus características principales se las describen a continuación:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

<TaskModel xmlns="http://giove.isti.cnr.it/ctt"
xmlns:coop="http://giove.isti.cnr.it/cttcoop" NameTaskModelID="Calculator">

  <Task Identifier="Calculator" Category="abstraction" Iterative="false"
Optional="false" PartOfCooperation="false">

    <Name>Calculator</Name>

```

Siendo esta la categoría de abstracción, porque se realiza el modelado de las tareas.

En esta sección se define la operación de la División

```

<Task Identifier="divide" Category="abstraction" Iterative="false"
Optional="false" PartOfCooperation="false">

```

```

  <Name>divide</Name>

```

```

  <Type></Type>

```

```

  <TemporalOperator name="Choice"/>

```

```

  <Parent name="Calculator"/>

```

```

  <SiblingRight name="add"/>

```

```

  <SubTask>

```

```

    <Task Identifier="Enter divide_parameters"
Category="interaction" Iterative="false" Optional="false"
PartOfCooperation="false">

```

```

      <Name>Enter divide_parameters</Name>

```

```

      <Type>Edit</Type>

```

```

      <TemporalOperator name="SequentialEnablingInfo"/>

```

```

      <Parent name="divide"/>

```

```
<SiblingRight name="Invoke divide"/>
```

```
<SubTask>
```

```
<Task Identifier="Enter divide numerator"
```

```
Category="interaction" Iterative="false" Optional="false"
```

```
PartOfCooperation="false">
```

Como se visualiza en la codificación no existe ninguna inclusión de una operación matemática en sí. Netamente es una representación gráfica del comportamiento de cada una de las tareas a realizarse.

Una vez importado en MARIA ENVIRONMENT, se presenta en la Figura 15.

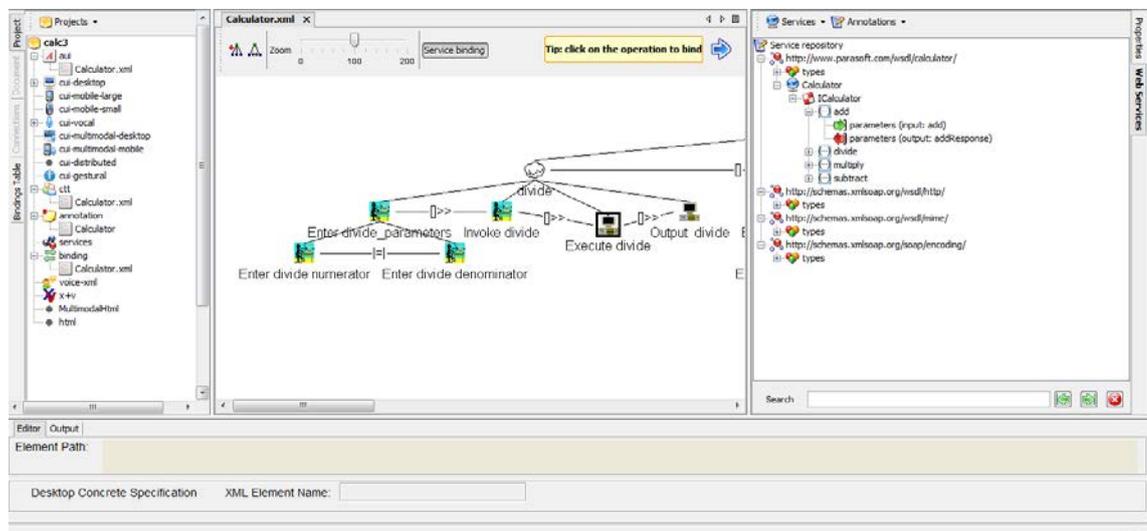


Figura 15. Presentación en Maria Environment

4.2.1 ASIGNACIÓN CONSUMO DEL SERVICIO WEB

En la presente sección se indica cómo realizar el consumo de servicios web y la asignación de éstos servicios a las tareas.

En la Figura 16. se visualiza como se puede realizar la asignación del servicio web. En donde se debe registrar la URL en la que se encuentra disponible el servicio.

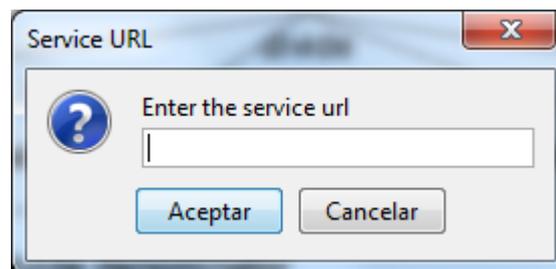
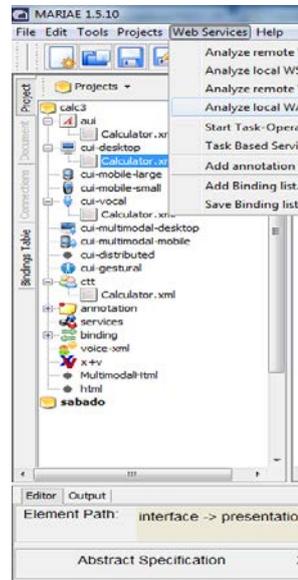


Figura 16. Registro de Servicio Web

Automáticamente se carga en el aplicativo los recursos del consumo web. En la Figura 17. se visualiza como se carga los recursos disponibles para realizar los cálculos matemáticos.

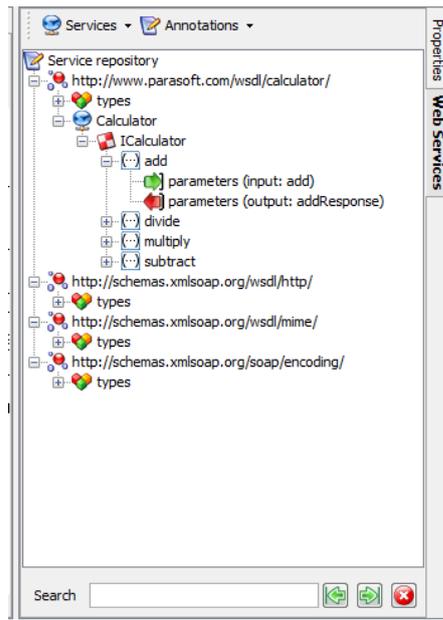


Figura 17. Visualización de las operaciones del servicio web

Estas operaciones son cargadas en las tareas compatibles como las de ejecución.

4.2.2 INTERFAZ ABSTRACTA DE USUARIO

Desde el modelado de tareas se genera la vista preliminar de la interfaz abstracta de usuario, que hereda los tipos de tareas establecidos en el modelado de tareas. Y así genera un atributo especial en cada campo.

En esta fase se evidencia tres secciones totalmente identificadas, que se mencionan a continuación:

- DATA
- Presentación
- Funciones externas del servicio web

En la Figura 18. se las puede visualizar

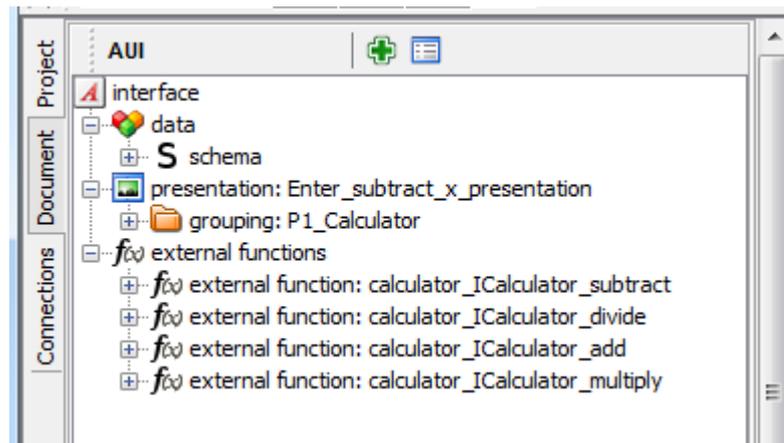


Figura 18. Características funcionales del usuario abstracto

En la Data se encuentran todos los elementos necesarios para hacer las respectivas operaciones como se evidencia en la Figura 19.

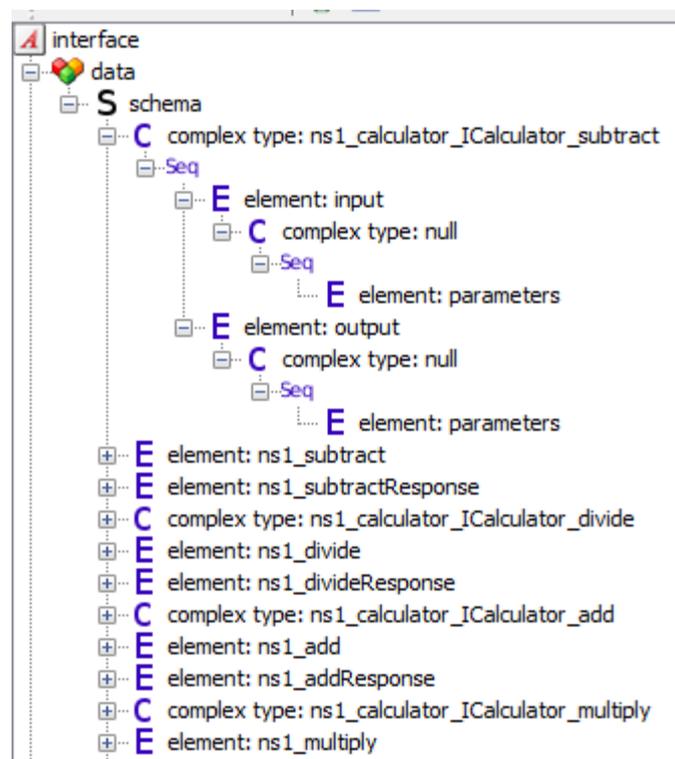


Figura 19. Contenido de la data que presenta la interfaz de usuario abstracta

Se observa cada una de las operaciones, con sus atributos y parámetros necesarios para ejecutar la misma.

4.2.2.1 Presentación

En esta sección se observa cómo se estructura visualmente el prototipo y la vinculación con cada una de las funciones. Como se evidencia en la Figura 20.

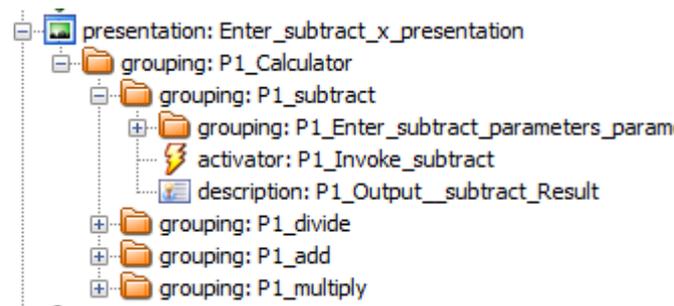


Figura 20. Vinculación con las funciones en la capa de presentación

Cada uno de los elementos tiene su propiedad específica, que va desde el ingreso de datos, hasta la obtención de los resultados como en la Figura 21.

En esta sección la operación de la división se va detallar con mayor profundidad que las otras operaciones para no redundar en su funcionalidad.

A continuación se va explicar cada uno de los elementos que intervienen el proceso de la división con sus respectivas características:

Ingreso Numerador

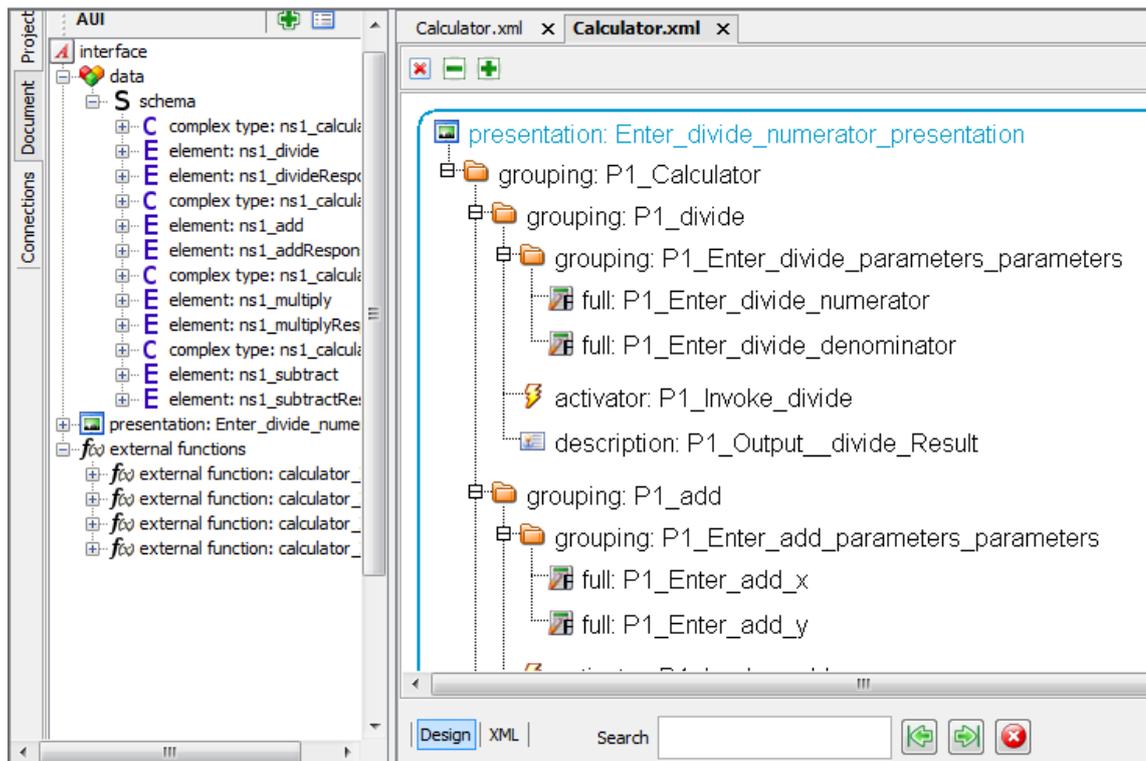


Figura 21. Propiedades de la capa de presentación

Se debe especificar su id y la referencia de los datos de donde son obtenidos. En la Figura 22. se puede identificar.

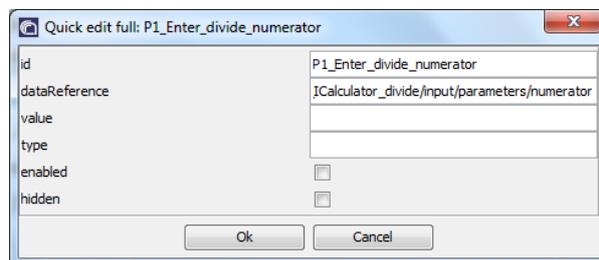


Figura 22. Referencia de datos

Lo mismo sucede con el divisor. Caso particular tiene el activador, cuya función es hacer llamado a la función con sus respectivos parámetros. Además tiene la característica de ser ejecutado por un evento como se indica en la Figura 23.

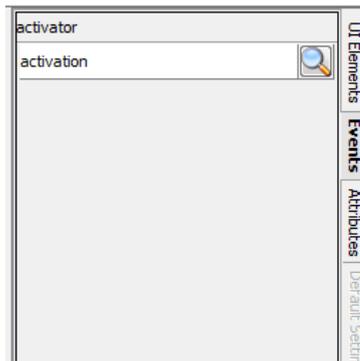


Figura 23. Funcionalidad de la característica tipo (evento) en la capa de presentación

En este caso la configuración se lo realiza mediante un script, como se presenta en la Figura 24.

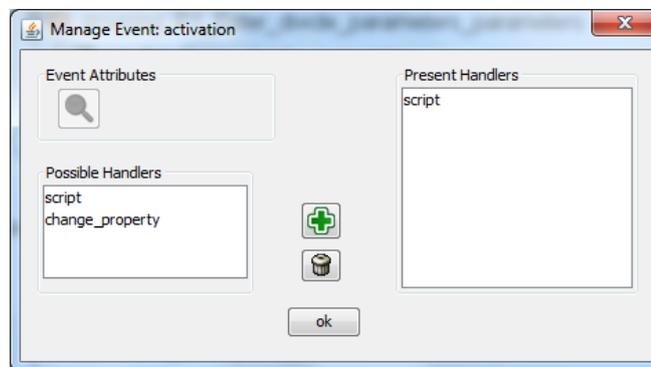


Figura 24. Script del evento en la capa de presentación

Es el script se lo realiza utilizando elementos de la interfaz de usuario como se presenta en la Figura 25.

Se crean elementos del tipo “Change property” ya que captura los valores que se desea procesar como se muestra en la Figura 26.

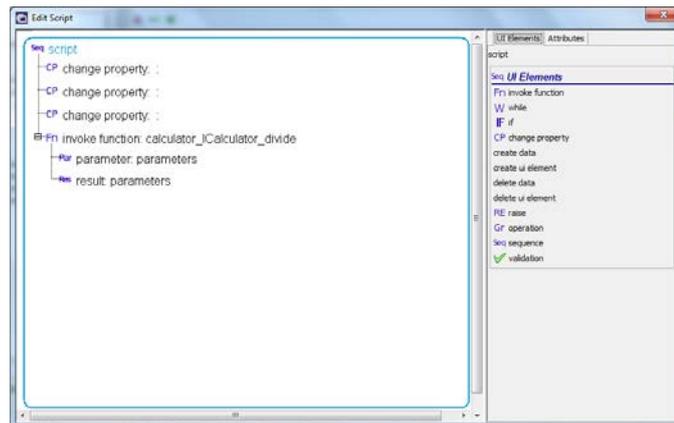


Figura 25. Visualización de script

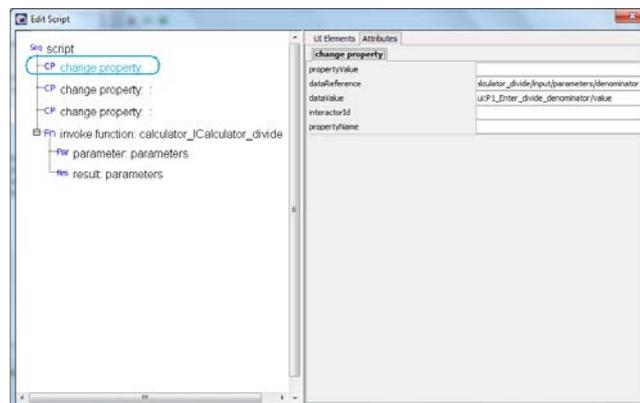


Figura 26. Características del cambio de propiedad

Como siguiente punto se llama a la función externa que va a realizar la operación con sus parámetros y su resultado, como evidencia la Figura 27.

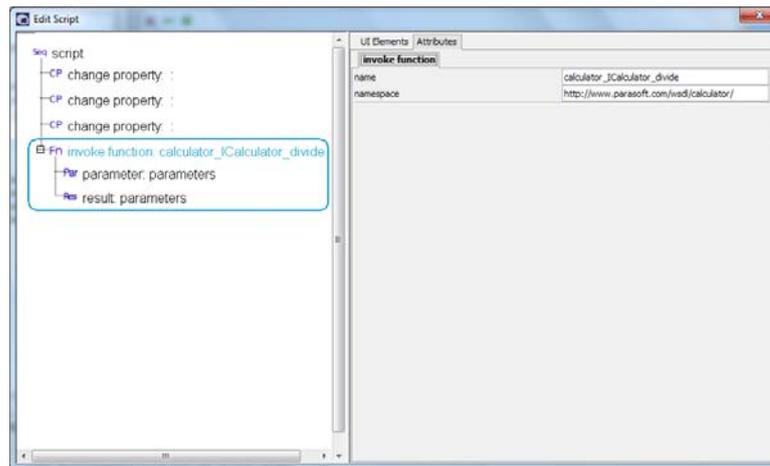


Figura 27. Características de las funciones externas

Una vez concluido la configuración del activador se procede a crear un cuadro en donde se almacenará los resultados como se presenta la Figura 28.

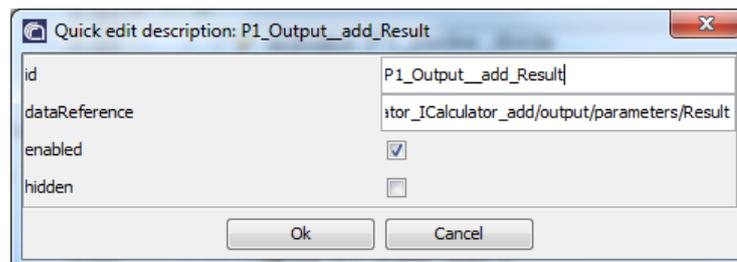


Figura 28. Cuadro descriptivo

4.2.2.2 Funciones externas

Es la sección en donde se establecen las operaciones heredadas del web service, se lo muestra en la Figura 29.

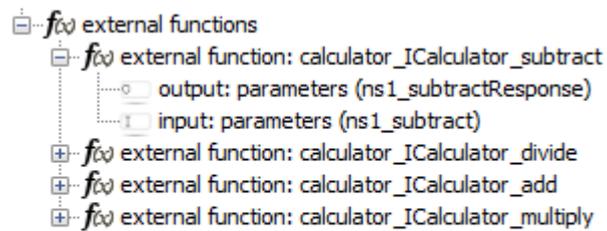


Figura 29. Visualización de las funciones externas

En cada una de las funciones se puede visualizar, de donde obtiene los recursos para el procesamiento de los datos, pueden ser de un servicio web, una base de datos o de una codificación. En este caso es de un servicio web como se indica en la Figura 30.

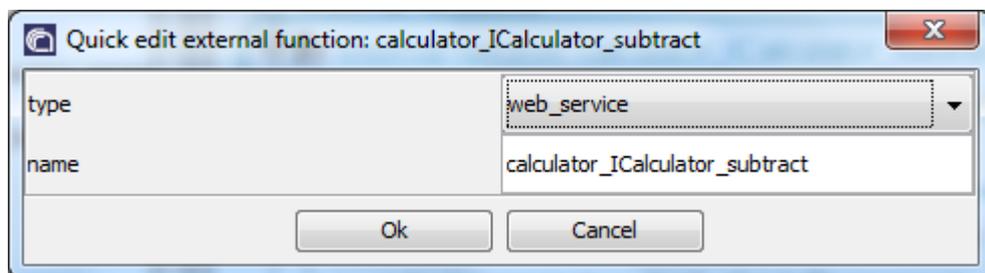


Figura 30. Consumo del recurso del servicio web

4.2.3 INTERFAZ CENTRADA AL USUARIO ORIENTADO AL ESCRITORIO

Esta sección trata la presentación utilizada por el usuario final. Es en esta sección la que se encarga de dar forma al prototipo.

Esta es la fase en la cual se puede configurar cada una de las características gráficas del prototipo, cada elemento posee un conjunto de atributos que pueden ser cambiados para su mejor apreciación.

En la Figura 31. se visualiza la caja de herramientas disponible para esta interfaz en particular.

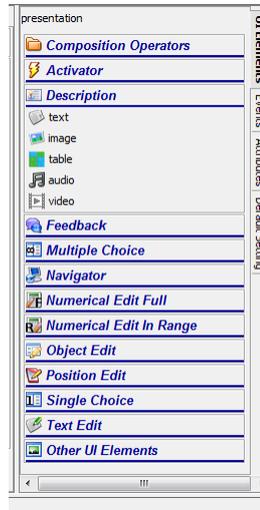


Figura 31. Caja de herramientas para modificar la interfaz

En esta caja de herramientas se puede importar incluso archivos de video, para que la interacción de usuario sea mucho más sencilla e intuitiva.

En los campos de texto se puede realizar modificaciones que, como cambio de tipo de letra, estilos, márgenes, etc. Como se puede visualizar la Figura 32.

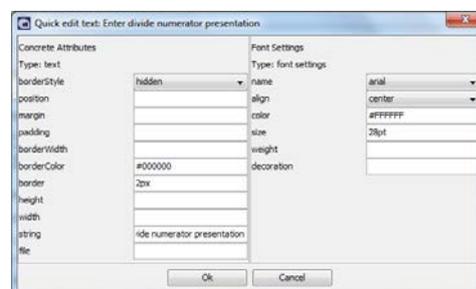


Figura 32. Modificación del campo tipo texto

Además tiene la opción de generar un preliminar en HTML para evidenciar los cambios realizados de mejor manera como se observa en la Figura 33.

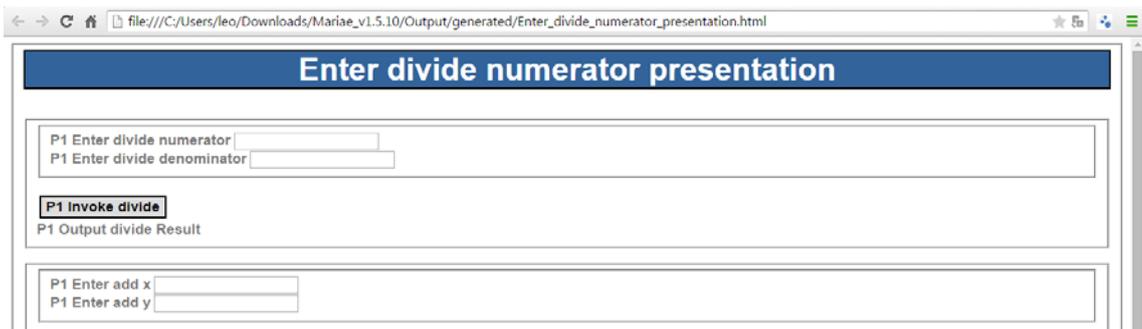


Figura 33. Visualización en HTML

La versatilidad que tiene la herramienta MARIA ENVIRONMENT, es que una vez generado esta interfaz se puede migrar automáticamente a otras interfaces para indistintos dispositivos, como en la Figura 34.

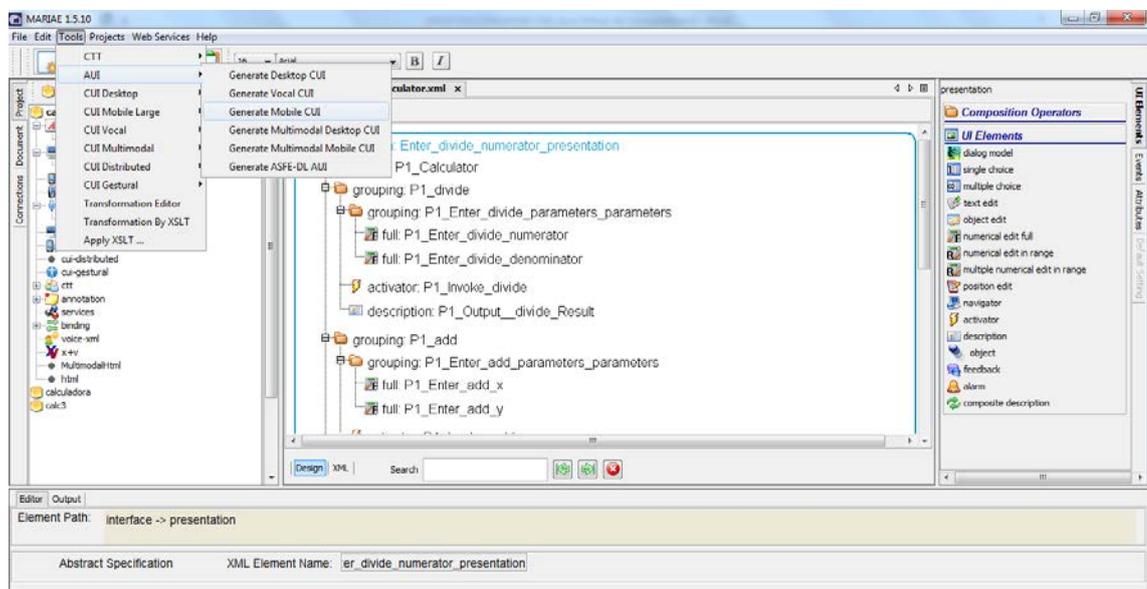


Figura 34. Exportación a diferente interfaz de usuario

Como se puede evidenciar, la transformación parte desde la Interfaz de usuario abstracta a la de dispositivo móvil. Como resultado da la siguiente interfaz representada en la Figura 35.

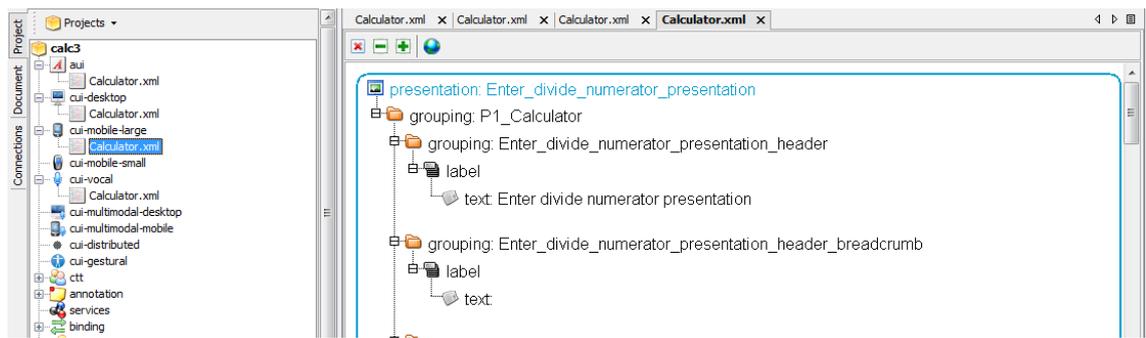


Figura 35. Representación a la interfaz de dispositivo móvil

Una vez concluido la elaboración del prototipo se procede a exportar el prototipo a un proyecto java web, el cual se lo realiza como se presenta en la Figura 36.

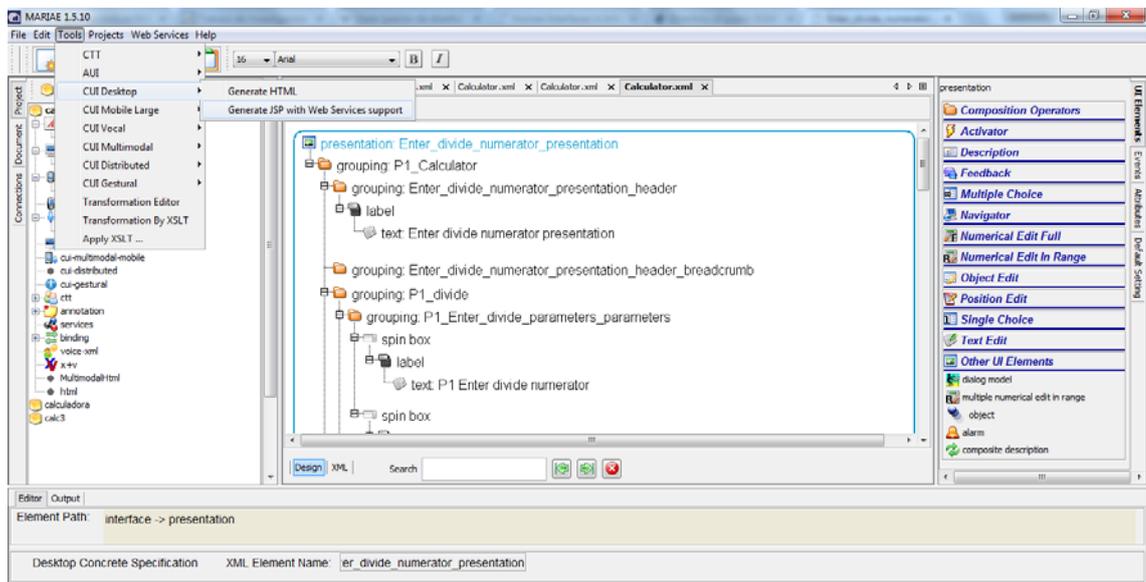


Figura 36. Exportación de código fuente

Este proyecto generado se lo puede ejecutar con cualquier IDE de java, para el presente ejemplo se utilizó el IDE de NETBEANS, como se presenta en la Figura 37. y en la Figura 38.

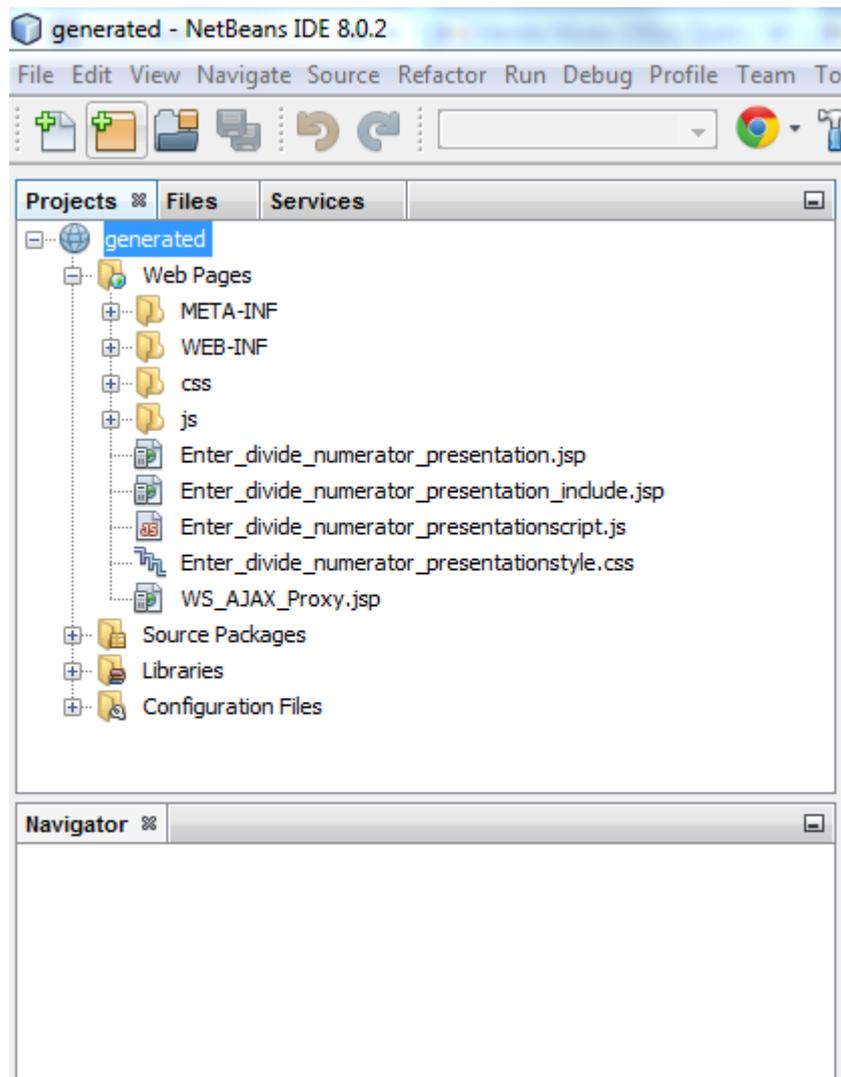


Figura 37. Representación gráfica del prototipo en JAVA

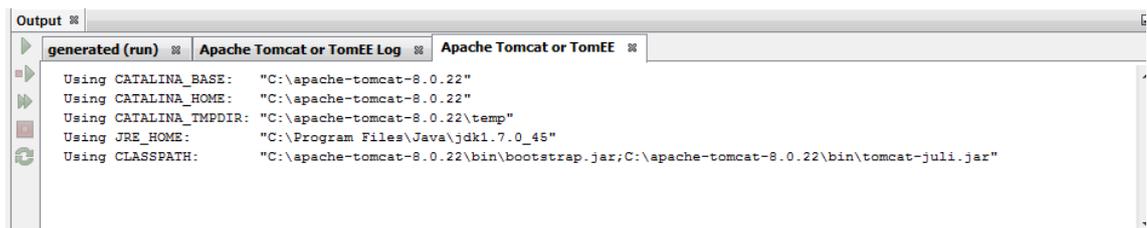


Figura 38. Ventana de consola

En el presente proyecto, se puede agregar codificación, crear una unidad de persistencia o a su vez incluir patrones de diseño para unir las etapas presentes en el proceso de desarrollo software. Y que sus procesos sean totalmente vinculantes, partiendo del mismo modelado de tareas.

Una vez en ejecución el proyecto se puede evidenciar como integra cada uno de los elementos elaborados en MARIA ENVIRONMENT y como hace consumo de los servicios web. Se lo muestra en la Figura 39. y Figura 40.

Enter divide

P1 Enter divide numerator

P1 Enter divide denominator

P1 Invoke divide

P1 Output divide Result 0.0

Figura 39. Representación de prototipo con el consumo Web

P1 Enter divide numerator

P1 Enter divide denominator

P1 Invoke divide

P1 Output divide Result 25.0

Figura 40. Representación de prototipo del consumo Web operación de la división

4.3 PROTOTIPO CONSULTA INFORMACIÓN AEROPORTUARIA

El segundo prototipo presenta el consumo de un servicio web de información acerca de los aeropuertos disponibles.

En la Figura 41. se presenta el modelado de tareas

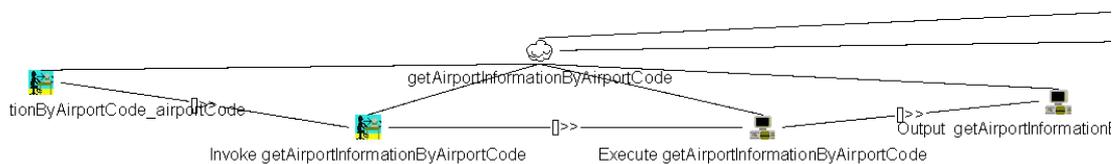


Figura 41. Modelado de tareas Información del Aeropuerto

El proceso de modelado de tareas realiza el consumo de la información más relevante de un aeropuerto.

4.3.1 INTERFAZ CENTRADA AL USUARIO ORIENTADO AL ESCRITORIO

En esta sección se observa las modificaciones de fondo y forma en la interfaz de usuario en mención.

En la Figura 42. se observa la inclusión de material multimedia

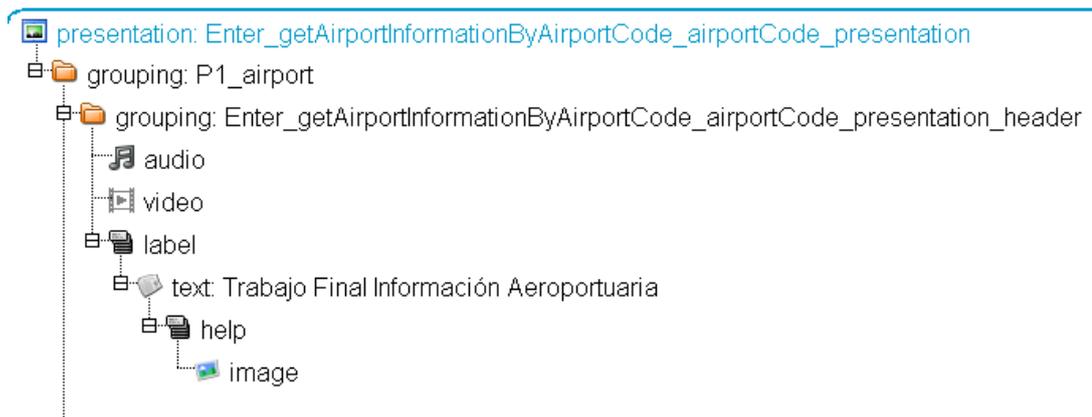


Figura 42. Inclusión de imágenes videos en la capa de presentación

En la Figura 43 se visualiza la capa de presentación finalizada, con la inclusión de material multimedia.

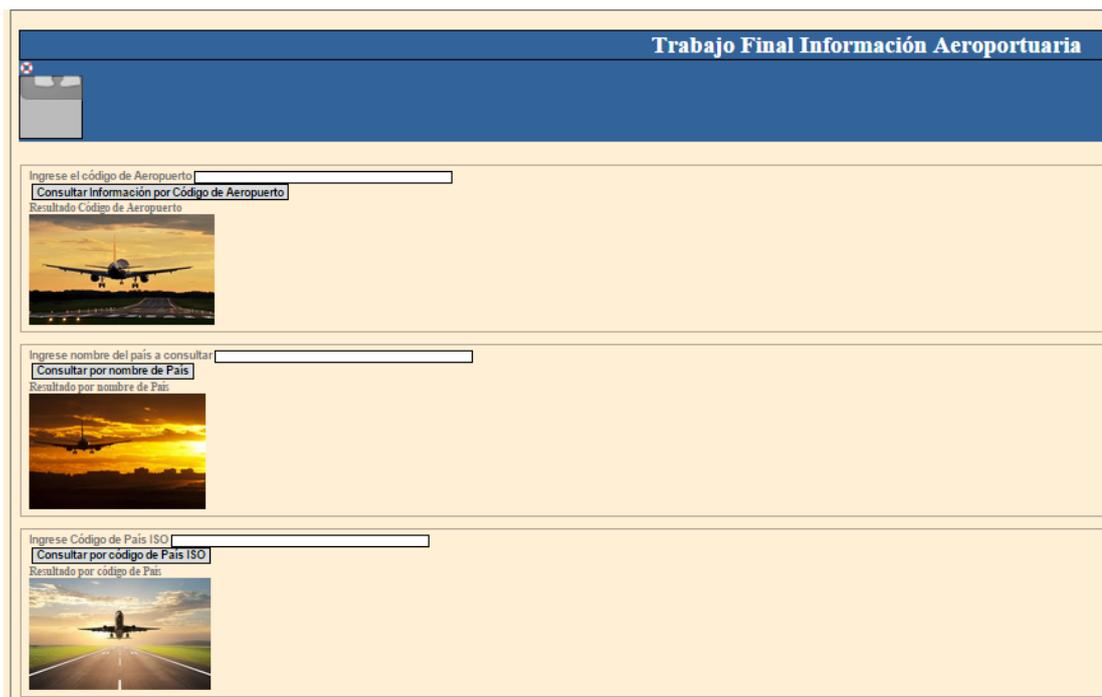


Figura 43. Capa de presentación en HTML

El desarrollo del segundo prototipo, no se menciona características que en el primer prototipo si se lo realizó, debido a que el objetivo del segundo prototipo es resaltar la inclusión de elementos multimedios.

De igual manera el presente trabajo contiene los archivos de los dos prototipos funcionales.

Una vez concluido la elaboración de los prototipos, se procede a realizar el manual de usuario. El manual se encuentra detallado en la sección de ANEXOS.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- Al desarrollar prototipos con el lenguaje MARIA, se obtienen productos y/o servicios en menor tiempo y costo de desarrollo. Utilizando la técnica análisis de tareas se diseñan prototipos tipo multi-plataforma
- El proceso de modelado de Tareas asume un rol protagonista en el diseño de interacción el cuál es desempeñado por el lenguaje MARIA.
- Con el uso del modelado de tareas en la herramienta MARIA ENVIRONMENT, se logra desarrollar prototipos funcionales apoyando a la elicitación de requerimientos y de esta manera realizar una evaluación de usabilidad.
- El Uso del Lenguaje MARIA disminuye la brecha existente entre el proceso de desarrollo de software y el proceso HCI

5.2 RECOMENDACIONES

- Se recomienda la utilización de patrones de diseño para tener prototipos funcionales de alto nivel de forma automática de esta manera ser utilizada para la elicitación de requerimientos y evaluación de la Usabilidad.

- Se propone realizar un banco de patrones de diseño, como servicios web para realizar su consumo en MARIA ENVIRONMENT.
- Se recomienda la utilización de la notación CTT con CTT Environment conjuntamente con MARIA para obtener prototipos basados en análisis de tareas y que tengan una funcionalidad significativa.

BIBLIOGRAFÍA

- Card, S. K., P., T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale: NJ: Erlbaum.
- Comunicación, I. N. (2009). *Guía de Certificación*. Madrid: Laboratorio Nacional de Calidad del Software .
- ConstantinosPhanouriou, & MarcAbrams. (2002). *UIML: An XML Language for Building Device-Independent User Interfaces*.
- Faedo", I. d. (2014, Marzo 18). *HIIS Laboratory*. Retrieved from The Human-Computer Interaction Group: <http://girove.isti.cnr.it/tools/MARIA/home>
- Grau, X. F. (2003). *Principio Básicos de Usabilidad para Ingenieros Software*. Madrid, Boadilla del Monte, España.
- Group, T. S. (2014). *CHAOS*. Project Smart.
- Hartson. (1992). Temporal aspects of tasks in the User Action Notation. In H. & D., *Human Computer Interaction* (pp. 1-45).
- Laboratory, H. (2015, 08 26). *The Human-Computer Interaction Group*. Retrieved from http://girove.isti.cnr.it/tools/MARIAE/MARIAE-how_to_use.html
- Mancha, P. d.-L. (n.d.). *Portal web del proyecto eCLUB*. Retrieved from <http://chico.infcr.uclm.es/eClub>
- Paternò, F. (2000). *Model-Based Design and Evaluation of Interactive Application*. Springer verlag.
- Paternò, F., Santoro, C., & Spano, L. D. (2012). *MARIA (Model-based Language foR Interactive Applications)*. ISTI-CNR.
- Paternò, F., Santoro, C., & Spano, L. D. (April 2012). *ConcurTaskTrees and MARIA languages for Authoring Service-based Applications*. USA: CNR-ISTI.
- Paterno', F., Santoro, C., & Spano, D. (2009). *MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments*. ACM.
- Paterno', F., Santoro, C., Spano, a., & D., L. (2009). *MARIA: A Universal, Declarative, Multiple*. ACM.

- Payne, K. (1986). Infrasonic calls from the Asian elephant (*Elephas maximus*). In K. Payne, *Behavioral Ecology and Sociobiology* (pp. 297-301).
- Santoro, C. (2005). *A Task Model-Based Approach for the Design and Evaluation of innovative User Interfaces*. Louvain la Neuve: Presses universitaires de Louvain.
- Seffah, A., Vanderdonckt, J., & Desmarais, M. C. (2009). *Human-Centered Software Engineering*. London : Springer-Verlag.
- Simarro, B. A. (2007, Septiembre). coCTT: Modelado de tareas en un entorno colaborativo. Castilla, La Mancha, España.
- Sommerville, I. (2005). *Ingeniería del software (Séptima Edición)*. . España: Pearson Educación.
- Sutcliffe, A. G. (2009). Convergence or competition between software engineering and human computer interaction. In A. Seffah, J. Vanderdonckt, & M. C. Desmarais, *Human-Centered Software Engineering* (pp. 71-73). London: Springer-Verlag.
- Xavier Ferre, N. B. (2010). Usability Planner: A Tool to Support the Process of Selecting Usability Methods. Boadilla del Monte, Madrid, España.
- Xavier, F. (2003). Incrementos de Usabilidad al Proceso de Desarrollo. Madrid, Boadilla del Monte, España.
- Zumaquero, S. F. (2008, Noviembre). Aplicación de la notación CTT(ConcurTaskTrees) a la creación de rutas en un museo. Abacete, España.



CARTA DE AUSPICIO

Por medio del presente, Yo Ingeniero Danilo Martínez MSS en calidad de miembro del Grupo de Investigación en Modelos de Producción de Software GRIMPSSOFT de la Universidad de las Fuerzas Armadas ESPE, certifico el auspicio a la tesis: "INVESTIGACIÓN DESCRIPTIVA DEL LENGUAJE BASADO EN MODELOS "MARIA" PARA EL ANÁLISIS DE TAREAS EN EL PROCESO DE INTERACCIÓN HUMANO COMPUTADOR Y SU APLICACIÓN EN UNA HERRAMIENTA CASE" a ser desarrollada por el Sr. Andrés Leonardo López Ludeña.

Existe el compromiso de GRIMPSSOFT con el Tesisista para proporcionarte la guía y la información necesaria para el desarrollo de su trabajo.

Sangolquí, 28 de noviembre de 2014

Atentamente,


Ing. Danilo Martínez Espinoza MSS
INVESTIGADOR
GRIMPSSOFT

CERTIFICADO

Por medio del presente, Yo Dr. Efraín Rodrigo Fonseca Carrera en calidad de miembro del Grupo de Investigación en Modelos de Producción de Software GRIMPESOFT de la Universidad de las Fuerzas Armadas ESPE, certifico la aceptación a la tesis: "INVESTIGACIÓN DESCRIPTIVA DEL LENGUAJE BASADO EN MODELOS "MARIA" PARA EL ANÁLISIS DE TAREAS EN EL PROCESO DE INTERACCIÓN HUMANO COMPUTADOR Y SU APLICACIÓN EN UNA HERRAMIENTA CASE" desarrollada por el Sr. Andrés Leonardo López Ludeña.

Sangolquí, 09 de Septiembre de 2015

Atentamente,



Dr. EFRAÍN RODRIGO FONSECA CARRERA
CC No. 1710979574

HOJA DE VIDA



1. DATOS PERSONALES

Apellidos: López Ludeña

Nombres: Andrés Leonardo

Nacionalidad: Ecuatoriano

Cédula: 1104595572

Fecha de Nacimiento: 11 de Febrero de 1990

Estado civil: Soltero

Domicilio: Inchalillo, Sangolquí

Teléfono: 022080418

Celular: 0984417333

E-mail: leo19902008@hotmail.com

2. FORMACIÓN ACADÉMICA

Educación Básica: Unidad Educativa "La Salle"

Educación Media: Unidad Educativa "La Salle"

Bachiller Físico Matemático

Educación Superior: Universidad de las Fuerzas Armadas

“ESPE”

Ingeniero en Sistemas e Informática

3. FORMACIÓN EXTRA ACADÉMICA

Inglés: Escrito Intermedio

Inglés: Oral Intermedio

Programación: Manejo Intermedio

Lenguajes de programación - C: Seniority Junior

Lenguajes de programación - C++: Seniority Junior

Lenguajes de programación - Java: Seniority Junior

Lenguajes de programación - Visual C/C++: Seniority Junior

Lenguajes de programación - .NET Framework 3.0: Seniority Junior

Bases de Datos - MySQL: Seniority Junior

Hardware - Reparación PC y compatibles: Seniority Junior

Sap BO

4. EXPERIENCIA LABORAR

Nifa jul 2010 - Agosto 2010

Operador, Etiquetador trabajo en bodega.

Ayudante de Servicio Varios

Servicio de Contabilidad

Archivado

3 Personas a cargo.

Jeanveoil julio 2013 – Abril 2014

Programador junior

Danec S.A. Junio 2014 – Julio 2014

Técnico

Heinsohntech. Julio 2014 – Febrero 2015

Consultor Técnico Sap BO

Grupo Siglo. Marzo 2015 – Actualmente

Consultor Técnico Sap BO

5. REFERENCIA

Anita Correa Gerente Jeanveoil 0991215131

Verónica Ríos Jefe de Desarrollo 0987322009

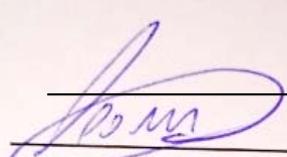
Jenny Cabezas Nifa 022333067

Ramiro Salvador Gerente Predesur

HOJA DE LEGALIZACIÓN DE FIRMAS

ELABORADO POR

LÓPEZ LUDEÑA ANDRÉS LEONARDO



Sr. López Ludeña Andrés Leonardo

DIRECTOR DE LA CARRERA
ING. MAURICIO CAMPAÑA



Ing. Mauricio Campaña

Sangolquí, Septiembre 2015