

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA

MAESTRÍA Y DOCTORADO EN CIENCIAS E INGENIERÍA



“MÉTODO PARA LA GENERACIÓN AUTOMÁTICA DE INTERFACES DE USUARIO
ADAPTIVAS UTILIZANDO UN MOTOR DE ADAPTACIÓN BASADO EN ATRIBUTOS DE
USUARIO, ANÁLISIS DE TAREAS Y REGLAS DE ADAPTACIÓN”

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

PRESENTA:

SERGIO ALBERTO INZUNZA SOBERANES

DIRECTOR DE TESIS:

DR. J. REYES JUÁREZ RAMÍREZ

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA
COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 114
Tijuana, B. C. a 06 de Febrero de 2014

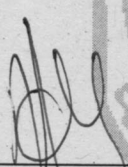
C. Sergio Alberto Inzunza Soberanes
Pasante de: Maestro en Ciencias
Presente

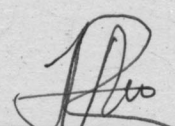
El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

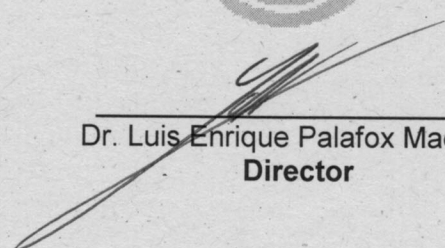
Es propuesto, por el C. Dr. J. Reyes Juárez Ramírez

Quien será el responsable de la calidad de trabajo que usted presente, referido al
tema "METODO PARA LA GENERACION AUTOMATICA DE INTERFACES DE
USUARIO ADAPTIVAS UTILIZANDO UN MOTOR DE ADAPTACION BASADO EN
ATRIBUTOS DE USUARIO, ANALISIS DE TAREAS Y REGLAS DE ADAPTACION ".
el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I.- INTRODUCCION
- II.- FUNDAMENTOS TEORICOS
- III.- METODO PARA LA GENERACION AUTOMATICA DE INTERFACES DE
USUARIOS ADAPTIVOS
- IV.- CASOS DE ESTUDIO
- V.- CONCLUSIONES
- VI.- REFERENCIAS
- VII.- APENDICE


Q. Noemí Hernández Hernández
Sub-Director Académico


Dr. J. Reyes Juárez Ramírez
Director de Tesis


Dr. Luis Enrique Palafox Maestre
Director

Dedicatoria

A mi esposa por darme su apoyo, a mis padres por haberme puesto en este camino y a mi hija por afianzar mi deseo de superación y darme la fuerza para seguir despierto...

Agradecimientos

Gracias a mis padres por haberme puesto en este camino desde muy pequeño, por su apoyo incondicional y consejos. Gracias padre por tu sacrificio en pro de mi bienestar. Gracias madre por haberme dedicado tu tiempo para enseñarme a multiplicar y dividir a mis seis años de edad, y por tus preocupaciones por mis pocas horas de sueño, valió la pena, ¡lo logramos!

Agradezco infinitamente a mi esposa, Adelita, por apoyarme tanto en este trayecto como en cualquier otro que he tomado, pero sobre todo gracias por tratar de entender mis horas de ausencia física y mental. Te agradezco tu interés por ayudarme en lo que te fue posible, y por dejarme solo cuando el nivel de desesperación comenzaba a subir. Tú sabes que sin tu apoyo esto no podría haber sido posible, el logro es ambos.

A mi hija, Sophia, por motivarme a seguir adelante, por ser la chispa de energía que me impulsa a tratar de mejorar. Por comprender cuando no podía atenderte, pero sobre todo por obligarme a levantarme de la silla para jugar contigo.

A mi director de tesis por su guía en la realización de la investigación, y redacción de este documento. Gracias profe por su apoyo para que esto fuera posible, eso no era parte de sus actividades como director. Pero sobre todo gracias por la amistad que me ha brindado desde que coincidimos en el primer proyecto en mis primeros semestres de ingeniería. Y disculpe las develadas.

También agradezco a Tania por apoyarme en lo que te fue posible, por orientarme tanto en la investigación como en los trámites, por escucharme hablar sin parar de cosas sin sentido, pero sobre todo por aguantarme.

Gracias a mis compañeros y amigos de posgrado por acompañarme, apoyarme y recordarme los trámites que tenía que hacer. A mis compañeros de trabajo por escucharme hablar sin entender de lo que hablaba.

Gracias a mis sinodales del comité de tesis por darme esta oportunidad de retomar los estudios, espero no haberlos defraudado.

Gracias a la Universidad Autónoma de Baja California y a la Facultad de Ciencias Químicas e Ingenierías por haberme dado esta oportunidad de superación profesional.

Gracias a CONACYT por el apoyo económico brindado durante los primero tres semestres.

Gracias a los alumnos y compañeros de ingeniería por ayudarme con los experimentos, se agradece su ayuda para que este proyecto finalizara en tiempo. Sobre todo gracias Armando y Jacobo, ay' estuvo.

Gracias a todos ustedes, aprecio su apoyo y guía para que este proyecto fuese posible, espero poder retribuirles en un futuro.

¿Qué, comenzamos de nuevo?

INDICE

INDICE	iv
LISTA DE FIGURAS.....	vii
LISTA DE TABLAS	viii
RESUMEN	x
ABSTRACT.....	xi
1 INTRODUCCIÓN	2
1.1 Planteamiento del problema.....	2
1.2 Objetivos	3
1.3 Metas.....	4
1.4 Hipótesis	4
1.5 Marco de trabajo	5
1.6 Organización del documento de tesis	6
2 FUNDAMENTOS TEÓRICOS.....	8
2.1 Interacción humano computadora.....	8
2.2 Calidad de software.....	9
2.2.1 Modelo de calidad de McCall	10
2.2.2 Modelo de calidad de Boehm	11
2.2.3 Modelo de calidad ISO 9126.....	12
2.3 Usabilidad	13
2.3.1 Atributos de usabilidad.....	14
2.3.2 Usabilidad como atributo de calidad de software.	15
2.4 Ingeniería de usabilidad	16
2.4.1 Ciclo de vida de Ingeniería de Usabilidad.....	17
2.4.2 Modelos de usuario	23

2.4.3	Análisis de tareas	24
2.5	Interfaz de usuario.....	26
2.5.1	Interfaces adaptativas	27
2.6	Motor de adaptación	30
3	MÉTODO PARA LA GENERACIÓN AUTOMÁTICA DE INTERFACES DE USUARIO ADAPTIVAS.....	32
3.1	Elementos principales de MEGAIA	33
3.1.1	Modelo integral de usuario.....	33
3.1.2	Tareas	41
3.1.3	Prototipos de interfaz.....	42
3.1.4	Reglas de adaptación	46
3.1.5	Motor de adaptación.....	53
3.2	Etapas de MEGAIA	55
3.2.1	Etapa 1: Generación de prototipos	55
3.2.2	Etapa 2: Adaptación de la interfaz	56
3.3	Implementación de un motor de adaptación	56
4	CASOS DE ESTUDIO	59
4.1	Caso de estudio con prototipos de interfaz	59
4.1.1	Objetivos del caso de estudio	59
4.1.2	Implementación de un generador automático de prototipos de interfaz	59
4.1.3	Realización del caso de estudio.....	61
4.1.4	Resultados del caso de estudio	63
4.1.5	Conclusiones del caso de uso	70
4.2	Caso de estudio sobre el motor de adaptación	71
4.2.1	Objetivos	71

4.2.2	Desarrollo de una aplicación mediante el método MEGAIA.....	71
4.2.3	Realización del caso de estudio.....	74
4.2.4	Resultados del caso de estudio	80
4.2.5	Conclusiones del caso de estudio.	91
5	CONCLUSIONES	94
5.1	Trabajo a futuro.....	96
6	REFERENCIAS.....	98
7	APÉNDICE.....	106
7.1	Apéndice A: Definición de tareas para realizar prototipos de interfaz	106
7.2	Apéndice B: Gráfica de Snellen para prueba de visión a corta distancia	109

LISTA DE FIGURAS

Figura 1: Aspectos de la IHC.....	8
Figura 2: Ciclo de vida de usabilidad de Mayhew.....	22
Figura 3: Adaptación de una interfaz a diferentes usuarios.....	32
Figura 4: Elementos principales de MEGAIA.....	33
Figura 5: Factores constituyentes del modelo integral de usuario.....	35
Figura 6: Propuesta de aspectos físicos del modelo integral de usuario.....	37
Figura 7: Propuesta de aspectos cognitivos considerados en el modelo integral de usuario.....	38
Figura 8: Aspectos demográficos del modelo integral de usuario.....	39
Figura 9: Atributos de experiencia considerados en el modelo integral de usuario.....	40
Figura 10: Atributos psicológicos considerados en el modelo integral de usuario.....	40
Figura 11: Selección de componentes de interacción para crear prototipos de interfaz.....	45
Figura 12: Ejemplo de una reglas de adaptación en XML.....	50
Figura 13: Elementos del motor de adaptación.....	54
Figura 14: Etapas propuestas en MEGAIA	55
Figura 15: Interfaz de usuario GAPIa.....	60
Figura 16: Prototipo de interfaz para la tarea 1.....	64
Figura 17: Prototipo de interfaz para la tarea 2.....	64
Figura 18: Prototipo de interfaz para la tarea 3.....	65
Figura 19: Prototipo de interfaz para la tarea 4.....	66
Figura 20: Prototipo de interfaz para la tarea 5.....	66
Figura 21: Interfaz normal para tarea 1.....	75
Figura 22: Interfaz normal para tarea 2.....	75
Figura 23: Interfaz normal para tarea 3.....	76
Figura 24: Interfaz normal para tarea 4.....	77
Figura 25: Interfaz normal para figura 5.....	77
Figura 26: Interfaces adaptadas - tarea 3	78
Figura 27: Interfaces adaptadas - tarea 4	79
Figura 28: Aumento en satisfacción de interfaces adaptativas respecto a normales	91

LISTA DE TABLAS

Tabla 1: Criterios considerados en el modelo de usabilidad de McCall.....	10
Tabla 2: Elementos que conforman el modelo de calidad de Boehm.....	11
Tabla 3: Características del modelo de calidad ISO 9126	12
Tabla 4: Usabilidad como atributo en modelos de calidad	16
Tabla 5: Elementos del ciclo de vida de usabilidad según Nielsen	17
Tabla 6: Regla de adaptación para visión excelente	50
Tabla 7: Regla de adaptación para visión normal	51
Tabla 8: Regla de adaptación para visión nula	51
Tabla 9: Regla de adaptación para experiencia con el dispositivo menor a dos meses	52
Tabla 10: Regla de adaptación para experiencia con el dispositivo mayor a dos meses	52
Tabla 11: Ejemplo de componente de interacción de Entrada.....	60
Tabla 12: Ejemplo de componente de interacción de Salida	61
Tabla 13: Ejemplo de componente de interacción de Indicación	61
Tabla 14: Distribución de participantes para diseño de interfaces.	62
Tabla 15: Resultados de evaluación de la completud de los prototipos	67
Tabla 16: Resultados de evaluación de prototipos tarea 1	67
Tabla 17: Resultados de evaluación de prototipos tarea 2.....	67
Tabla 18: Resultados de evaluación de prototipos tarea 3.....	68
Tabla 19: Resultados de evaluación de prototipos tarea 4.....	68
Tabla 20: Resultados de evaluación de prototipos tarea 5.....	69
Tabla 21: Resultados de la evaluación de prototipos tarea 6.....	69
Tabla 22: Guía de preguntas para evaluar el perfil de usuario	72
Tabla 23: Cuestionario de satisfacción	80
Tabla 24: Características de usuarios del grupo GU1.....	81
Tabla 25: Características de usuarios del grupo GU2.....	82
Tabla 26: Resultados de interfaces para la tarea 1 con grupo de usuarios GU1	82
Tabla 27: Resultados de interfaces para la tarea 1 con grupo de usuarios GU2.....	83
Tabla 28: Resultados de interfaces para la tarea 2 con grupo de usuarios GU1	84
Tabla 29: Resultados de interfaces para la tarea 2 con grupo de usuario GU2	85
Tabla 30: Resultados de interfaces para la tarea 3 con grupo de usuarios GU1	85

Tabla 31: Resultados de interfaces para la tarea 3 con grupo de usuarios GU2	86
Tabla 32: Resultados de interfaces para la tarea 4 con grupo de usuarios GU1	87
Tabla 33: Resultados de interfaces para la tarea 4 con grupo de usuarios GU2	87
Tabla 34: Resultados de interfaces para la tarea 5 con grupo de usuarios GU1	88
Tabla 35: Resultados de interfaces para la tarea 5 con grupo de usuarios GU2	89
Tabla 36: Condensado de aumento en eficiencia por tarea.....	89
Tabla 37: Satisfacción de los usuarios con las diferentes interfaces	90
Tabla 38: Satisfacción de usuarios del grupo GU1 con las diferentes interfaces	90
Tabla 39: Satisfacción de usuarios del grupo GU2 con las diferentes interfaces	90

RESUMEN

Método para la generación automática de interfaces de usuario adaptivas utilizando un motor de adaptación basado en atributos de usuario, análisis de tareas y reglas de adaptación.

En esta tesis se propone un método para generar de manera automática interfaces de usuario que en tiempo de ejecución se adapten a las características de cada uno de los usuarios.

La propuesta consta de 4 elementos principales: modelo integral de usuario, prototipos de interfaz, reglas de adaptación y un motor de adaptación. El modelo integral de usuario contiene valores para los atributos de usuario. Los prototipos de interfaz contienen los componentes necesarios para permitirle a un usuario realizar la tarea. Las reglas de adaptación contiene la relación entre atributo de usuario y componentes de interacción. El motor de adaptación es el encargado de realizar las adaptaciones necesarias al prototipo de interfaz.

Estableciendo una relación entre las acciones de la tarea y un componente de interacción, se hace una propuesta para generar de manera automática los prototipos de interfaz, basándose en la descripción de los casos de uso o en el análisis de tareas.

La adaptación de los prototipos de interfaz se realiza mediante reglas de adaptación del tipo IF...THEN. Las comparaciones de las reglas se realizan sobre atributos de usuario y acciones de tarea, y las salidas de las reglas definen que componente de interacción debe ser integrado en la interfaz adaptada.

Se realizaron dos casos de estudio, uno para probar la generación automática de prototipos en la cual se evaluó la completud y efectividad de los mismos, y el segundo para probar el funcionamiento del motor de adaptación así como el aumento que proporciona en cuanto a eficiencia y satisfacción del usuario.

El método propuesto es independiente de leguajes y sistemas operativos, pero la herramienta para generar prototipos se realizó en Windows y pruebas se realizaron sobre plataforma Android.

Palabras Clave: Interfaz adaptiva, prototipo de interfaz, motor de adaptación, regla de adaptación, modelo de usuario.

ABSTRACT

Method for automatic generation of adaptive user interfaces using an adaptation engine based on user attributes, task analysis and adaptation rules.

In this thesis, we propose a method to automatically generate user interfaces capable of adapt to individual characteristics of each user at runtime.

This proposal consist of 4 principal elements: an integral user model, user prototypes, adaptation rules and an adaptive engine. The integral user model contains values for user attributes. Interface prototypes contains the necessary components to allow the user accomplish the task. Adaptation rules contains the relation between user attributes and interaction components, the adaptation engine is in charge of making changes to the interface prototype.

By the establishment of a relation between task actions and interaction components, we are able to automatically generate interface prototypes, using the use case description or task analysis results as input.

The adaptation of the interface prototypes is defined in IF...THEN adaptation rules. Those rules are resolve against user attributes and task actions, the output of the adaptation rules define which interaction component should be integrated in the adapted user interface.

Two cases of study where made, one to prove the automatic generation of prototypes, in this case of study we evaluate the number of component and the kind of each one against an expert generated prototype. The second case of study was made to test the adaptation engine operation and to measure the increase of performance and user satisfaction.

The proposed method in language and platform independent, nevertheless the implemented tools was developed on Windows to generate Android interface prototypes.

Keywords: adaptive interfaces, prototype interface, adaptation engine, adaptation rules, integrated user model.

INTRODUCCIÓN

“Make everything as simple as possible, but not simpler.”

- Albert Einstein -

1 INTRODUCCIÓN

Conforme avanza la tecnología y el uso de computadoras se vuelve más ubicuo, los sistemas computacionales aumentan en funcionalidad, y con ello, la complejidad de las interfaces de usuario (Benyon and Murray 1991; Benyon 1993). A pesar de eso, actualmente las interfaces graficas de usuario son comúnmente desarrolladas asumiendo que serán utilizadas por usuarios típicos, quienes cuentan con las habilidades cognitivas comunes y utilizarán un conjunto típico de dispositivos de entrada y salida. Cualquier desviación de dicha asunción (por ejemplo, temblor en las manos por causa de la edad, falta de alguna extremidad o simplemente una ligera debilidad visual) puede disminuir drásticamente la efectividad del usuario (Gajos, Long, and Weld 2006; Lavie and Meyer 2010).

Sin embargo, hace algunos años, apoyado por el aumento en la capacidad de procesamiento y en la cantidad de sensores presentes en los dispositivos (Soulard 2012), ha surgido una propuesta acerca de interfaces flexibles bajo la rúbrica de *interfaces de usuario adaptivas*, dicha propuesta se basa en modificar los componentes presentes en la interfaz de usuario, así como la organización y características visuales de estos de acuerdo a las características de la persona que está utilizando el sistema (Langley and Fehling 1998).

El principal objetivo este documento es proponer de método para la generación automática de interfaces de usuario adaptivas. Éste consta de los siguientes elementos: un generador automático de prototipos de interfaz basado en el conjunto de acciones de la tarea a para la que se desea genera el prototipo y un motor de adaptación basado en reglas de adaptación, el cual es el encargado de hacer los cambios correspondientes al prototipos de interfaz y así generar una interfaz de usuario específica para cada uno de los usuarios.

1.1 Planteamiento del problema

Los usuarios de los sistemas computacionales difieren en una gran cantidad de variables, incluyendo aspectos demográficos, educación, personalidad, habilidades cognitivas y preferencias. La motivación del usuario, metas y estado de ánimo también varían (Lavie and Meyer 2010).

Actualmente en el desarrollo de sistemas, dicha diversidad en las características de los usuarios es comúnmente ignorada (Gajos, Weld, and Wobbrock 2010) o en el mejor de los casos, atendida mediante un rediseño manual de la interfaz de usuario lo cual limita la escalabilidad del sistema (Bergman and Johnson 1995). También utilizando dispositivos externos para usuarios con discapacidades perceptuales o motoras, pero dichos dispositivos implican costo extra, complejidad logística, configuración y mantenimiento continuo (Dawe 2006).

1.2 Objetivos

El problema descrito anteriormente dio pie a los objetivos de este documento de tesis. A continuación se describe el objetivo general, así mismo se desglosa dicho objetivo en tres más específicos.

Objetivo general:

Definir un método para automatizar la generación de interfaces de usuario adaptivas utilizando un motor de adaptación basado en atributos de usuario, tareas y reglas de adaptación, con la finalidad de aumentar la usabilidad del sistema de software para cada uno de los usuarios del rango de usuarios del sistema.

Objetivos específicos:

O1: Adoptar un esquema de especificación de tareas cuyo nivel más granular sean la acciones de usuario o de sistema.

O2: Definir los atributos representativos de un usuario que guardan relación en la interacción de este con un sistema de software interactivo.

O3: Definir una estructura para reglas de adaptación del tipo IF...THEN, las cuales contengan las acciones a seguir durante el proceso de adaptación de una interfaz de usuario.

O4: Definir un motor de adaptación genérico, basado en reglas de adaptación del tipo IF...THEN (**O3**), que sea capaz de adaptar en tiempo de ejecución una interfaz a los atributos del usuario que está utilizando el sistema.

1.3 Metas

A continuación se presenta una serie de metas, con las cuáles se pretende cumplir con los objetivos descritos en la sección anterior.

M1 (O1): Un análisis las propuestas existentes para la especificación de tareas.

M2 (O2): Una lista de los atributos de usuario que tienen relación con la interacción humano computadora.

M3 (O3): Una propuesta para la representación de reglas de adaptación basadas en atributos de usuario y tareas.

M4 (O4): Un conjunto de componentes que conforman un motor de adaptación.

M5 (Objetivo General): Una análisis de atributos de usabilidad que se ven beneficiados al utilizar adaptabilidad en las interfaces de usuario.

1.4 Hipótesis

Es esta sección se describen las hipótesis sobre las cuales está basado el presente trabajo de investigación.

H1: Una interfaz de usuario puede adaptarse automáticamente a las capacidades y características de los usuarios, utilizando un motor de adaptación basado en atributos de usuario, tareas y reglas de adaptación.

H1.1: Si la interfaz de usuario se crea conforme a las capacidades y características de la persona que está utilizando el sistema en ese momento, la usabilidad del sistema para ese usuario se aumenta significativamente.

H1.2: La generación de prototipos de interfaz de usuario puede ser automatizada estableciendo una relación entre la acción de la tarea y los componentes de interacción que conforman una interfaz gráfica.

1.5 Marco de trabajo

Para desarrollar la propuesta que en este documento se describe se hizo de una extensa investigación acerca del estado del arte de la interacción humano computadora y la usabilidad de software. Conforme se avanzó, la investigación se centró en la usabilidad de las interfaces de usuario y más puntualmente en interfaces de usuario adaptivas.

Debido a que uno de los elementos de nuestra propuesta es el modelo de usuario (véase sección 3.1), se trabajó en conjunto con un alumno de maestría, el ahora M.C. Andrés Mejía, de donde surgió nuestra propuesta del Modelo Integral de Usuario (véase sección 3.1.1), que fue creada en colaboración con el Dr. Reyes (Mejía et al. 2012)

Otro elemento constituyente de esta propuesta son los prototipos de interfaz, que son puntualizados en la sección 3.1.3 Se formularon 5 postulados, 3 de los cuales (postulados 1, 2 y 3) son apoyados por referencias bibliográficas y los 2 restantes son presentados como propuesta.

Los prototipos de interfaz propuestos son un conjunto de componentes de interacción, que pueden ser mapeados a las características de usuario. Dicha propuesta fue presentada en un artículo titulado *Implementing user-oriented interfaces: From user analysis to framework's components* y aceptado en la serie de conferencias *Uncertainty Reasoning and Knowledge Engineering (URKE)* (Inzunza et al. 2011).

Para los experimentos realizados (véase sección 4) se utilizaron 27 estudiantes de la carrera ingeniería en computación, de la Universidad Autónoma de Baja California, Facultad de Ciencias Químicas e Ingeniería, que al momento de la experimentación contaban con conocimientos sobre ingeniería de software, interacción humano computadora, usabilidad y experiencia en desarrollo de sistemas para proyectos escolares. Dichos alumnos diseñaron prototipos de interfaz que fueron comparados con los prototipos desarrollados utilizando nuestra propuesta de generación automática de prototipos de interfaz.

También se utilizó para los experimentos un desarrollador de software con 10 años de experiencia, un estudiante de maestría con experiencia con en uso de dispositivos móviles y una maestra en ciencias del área de usabilidad. Estas tres personas diseñaron los prototipos de interfaz que fueron tomados como versión de expertos y utilizados como punto de referencia para evaluar los prototipos generados utilizando el método propuesto.

Para el segundo experimento se utilizaron a los mismos alumnos pero esta vez eran ellos quienes aplicaban el experimento a personas comunes, elegidas al azar, donde el único requisito que debían cumplir era ser mayor de 20 años de edad. El sexo, profesión, actividad fueron indistintos. Durante experimento, los usuarios debían realizar 5 tareas con dos interfaces de usuario distintas, el sistema se ejecutaba sobre un sistema operativo Android y los dispositivos utilizados fueron diversos, ya que cada equipo encargado de realizar prueba proporcionó su propio dispositivo.

1.6 Organización del documento de tesis

El presente documento consta de siete capítulos por medio que describen el trabajo de investigación desarrollado.

Capítulo 1: En este capítulo se presenta una introducción al trabajo de investigación, se define el planteamiento del problema y se enlistan los objetivos, metas e hipótesis.

Capítulo 2: En este capítulo se establece un marco teórico y se desarrolla brevemente el estado del arte sobre el dominio del problema, que constituye la base fundamental de la propuesta.

Capítulo 3: Es en este capítulo donde se describe el método propuesto, se detallan cada uno de los elementos que la integran y se describe el proceso para generar automáticamente interfaces de usuario adaptivas.

Capítulo 4: En este capítulo se describen los experimentos realizados mediante que prueban el método propuesto. Se presentan objetivos, proceso utilizado, resultados obtenidos y las conclusiones para cada uno de los experimentos.

Capítulo 5: En este capítulo se presentan las conclusiones generales sobre el trabajo de investigación, sobre el método propuesto, los experimentos, así como de hipótesis y objetivos.

Capítulo 6: En este capítulo se enlistan las referencias bibliográficas utilizadas para fundamentar la propuesta.

Capítulo 7: En este capítulo se anexan algunos artefactos utilizados durante la experimentación.

II

FUNDAMENTOS TEÓRICOS

"No man should escape our universities without knowing how little he knows."

-J. Robert Oppenheimer -

2 FUNDAMENTOS TEÓRICOS

2.1 Interacción humano computadora

La interacción humano computadora (IHC) es la encargada del diseño de sistemas informáticos que den soporte a personas, de tal forma que éstas puedan llevar a cabo sus actividades productivamente y con seguridad personal (Preece et al. 1994). Para el diseño de interfaces de usuario, es necesario considerar una gran cantidad de factores (Kumar 2005), para lo cual la IHC hace uso de distintos campos del saber. La **Figura 1** muestra un diagrama de las áreas que apoyan a la IHC. Así, la IHC toma un carácter multidisciplinario que puede proporcionar todos los puntos de vista necesarios para el desarrollo de sistemas informáticos capaces de soportar usuarios con diferentes capacidades y objetivos (Ferré 2005)

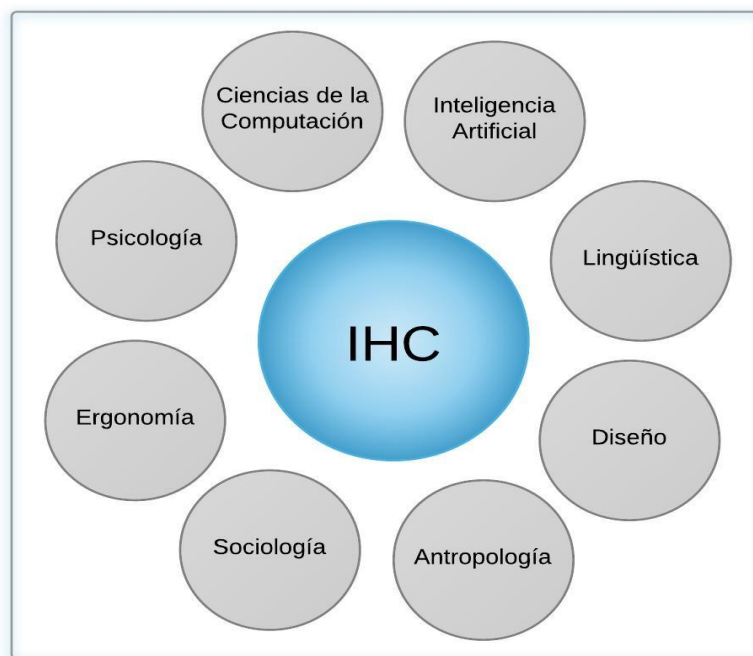


Figura 1: Aspectos de la IHC

El hacer uso de diferentes áreas del saber le permite a la IHC cubrir todos los aspectos que intervienen en el proceso de interacción entre el usuario y el sistema de cómputo, algunas áreas se enfocan en el estudio de los factores humanos (psicología, sociología, antropología y lingüística), otras en factores tecnológicos (ciencias de la computación, inteligencia artificial y

diseño) y existen otras áreas que convergen en ambos factores mencionados anteriormente (ergonomía).

La psicología y las ciencias cognitivas ayudan a entender como los usuarios perciben, piensan, toman decisiones, actúan, y por qué lo hacen de una u otra forma (Gyurky 2006). También nos permite entender las limitaciones cognitivas de los usuarios y sus capacidades de interacción en grupo (Dordević and Rančić 2008). Sociología y antropología aportan el conocimiento necesario sobre el impacto e influencia que la tecnología tiene en los usuarios. Uno de los objetivos principales de las ciencias sociales es el entender la interacción entre un sistema computacional y el usuario durante y después de un evento, por lo tanto, una de las razones por las cuales es necesario incluir las ciencias sociales in la IHC es para obtener una descripción más precisa de la interacción entre usuarios, sus tareas y las tecnologías que el sistema implementa (Lester 2008).

La inteligencia artificial tiene como objetivo diseñar sistemas computacionales que simulen diferentes aspectos del comportamiento humano inteligente, gracias a esto ha sido utilizada, entre otras cosas, para interfaces evolutivas, interfaces de lenguaje natural utilizando la voz y para crear agentes inteligentes que sirvan como guía a los usuarios (Abascal et al. 2001).

Ambos factores, humanos y tecnológicos, convergen dentro del área ergonomía; la cual se encarga de diseñar la parte interactiva del sistema buscando desempeño y eficiencia de una manera cómoda para el usuario, basándose en sus capacidades físicas (M. Nielsen et al. 2004).

2.2 Calidad de software

En (Pressman 2010) se define el término calidad de software como la concordancia con los requisitos explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

Actualmente existen varios modelos de calidad de software, donde se desglosa la calidad de un sistema en un conjunto de atributos y características. Los predecesores de los modelos de calidad actuales son los modelos de McCall (1977) y Boehm (1978).

2.2.1 Modelo de calidad de McCall

Este modelo presentado por (McCall, Richards, and Walters 1977) está enfocado en cerrar la brecha que existe entre los desarrolladores y los usuarios finales de los sistemas de software centrándose en una serie de factores de calidad del software que reflejan tanto los puntos de vista de los usuarios y las prioridades de los desarrolladores (Al-qutaish 2010).

Se divide en 3 categorías que abarcan 11 factores para los cuales se definen 23 criterios (Véase **Tabla 1**: Criterios considerados en el modelo de usabilidad de McCall). Para cada uno de dichos criterios se propone una serie de métricas para su cuantificación (Stavrinoudis, Xenos, and Christodoulakis 1999).

Tabla 1: Criterios considerados en el modelo de usabilidad de McCall

Categoría	Factor	Criterio
Operación	Correctitud	- Rastreabilidad - Completitud - Consistencia
	Confiabilidad	- Consistencia - Exactitud - Tolerancia a Fallas
	Eficiencia	- Eficiencia en ejecución - Eficiencia en almacenamiento
	Integridad	- Control de acceso - Auditoria de acceso
	Usabilidad	- Operabilidad - Entretenimiento - Comunicación
Revisión	Mantenibilidad	- Simplicidad - Corrección
	Capacidad de prueba	- Simplicidad - Instrumentación - Auto-descriptividad - Modularidad
	Flexibilidad	- Auto-descriptividad - Capacidad de expansión - Generalidad

		- Modularidad
Transición	Portabilidad	- Auto-descriptividad - Independencia del sistema - Independencia de la maquina
	Reusabilidad	- Auto-descriptividad - Generalidad - Modularidad - Independencia del sistema - Independencia de la maquina
	Interoperabilidad	- Modularidad - Similitud de comunicación - Similitud de datos

2.2.2 Modelo de calidad de Boehm

Este modelo de calidad es el propuesto por (Boehm 1978), cuyo objetivo es definir cuantitativamente por medio de una lista de atributos y un conjunto de métricas para cada uno de ellos (Véase **Tabla 2**).

Al igual que el modelo propuesto por McCall, el de Boehm está estructurado, y desglosa en características de alto nivel, intermedio y nivel primitivo, donde pueden ser cuantificadas más fácilmente que las de mal alto nivel, de igual manera se puede identificar de manera más directa como contribuyen o afectan a la calidad del sistema de software final.

Tabla 2: Elementos que conforman el modelo de calidad de Boehm

Características de alto nivel	Características de nivel intermedio	Características primitivas
Utilidad	Confiabilidad	- Auto-suficiencia - Precisión - Robustez - Integridad - Consistencia
	Eficiencia	- Responsabilidad - Eficiencia del dispositivo - Accesibilidad
	Ingeniería humana o	- Robustez/Integridad

	usabilidad	- Accesibilidad - Comunicabilidad
Portabilidad	Comprobabilidad	- Independencia del dispositivo - Auto-suficiencia
Mantenibilidad	Entendibilidad	- Responsabilidad - Comunicabilidad - Modularidad - Auto-descriptividad - Estructurabilidad
	Interoperabilidad	- Consistencia - Estructurabilidad - Similitud de datos - Concisión - Legibilidad
	Modificabilidad	- Estructurabilidad - Aumentabilidad

2.2.3 Modelo de calidad ISO 9126

En 1991 ISO publicó el primer consenso internacional sobre características de calidad para la evaluación de productos de software. A dicho estándar se le dio el nombre de Product Software Evaluation - Quality Characteristics and Guidelines for Their Use (ISO 9126). En este modelo se determinan seis características las cuales son sub-divididas en veintitrés para calidad interna y externa del producto de software. Dichas sub-características son un resultado de atributos internos del software y son apreciables externamente cuando el software es utilizado como una parte de un sistema computacional. En la **Tabla 3** pueden observarse las seis principales características así como su sub-división.

Tabla 3: Características del modelo de calidad ISO 9126

Características	Sub-características
-----------------	---------------------

Funcionalidad	- Adecuación - Exactitud - Interpolaridad - Seguridad
Confiabilidad	- Madurez - Tolerancia a fallos - Recuperabilidad
Usabilidad	- Entendimiento - Aprendizaje - Operabilidad - Atracción
Portabilidad	- Adaptabilidad - Facilidad de instalación - Coexistencia - Reemplazabilidad
Mantenibilidad	- Analizabilidad - Cambiabilidad - Estabilidad - Facilidad de prueba
Eficiencia	- Comportamiento en el tiempo - Utilización de recursos - Cumplimiento de eficiencia

Como podemos observar, uno de los atributos de la calidad común entre los diferentes modelos es la usabilidad, la cual se define en la siguiente sección.

2.3 Usabilidad

En (Mayhew 1999) se define usabilidad como una característica medible de una interfaz de usuario que está presente en mayor o menor grado, y representa la facilidad que tiene dicha interfaz para ser utilizada por usuarios novatos y casuales.

En el estándar ISO 9241-11: Guías de usabilidad (Thurnher 2004) define usabilidad como:

“el grado en que un producto puede ser utilizado por usuarios específicos para lograr objetivos específicos con efectividad, eficiencia y satisfacción en un contexto específico de uso”

Efectividad es definida en (Whitney Quesenbery 2001) como la integridad y exactitud con lo cual el usuario logra alcanzar sus metas. Puede ser determinado al observar si el usuario cumplió sus objetivos y si el trabajo realizado es correcto.

Eficiencia se refiere al número de tareas por unidad de tiempo que el usuario puede realizar utilizando el sistema (Ferre et al. 2001).

Satisfacción se refiere a que tan agradable es usar un sistema, este atributo puede ser especialmente importante en sistemas de entretenimiento donde la velocidad es menos importante que el entretenimiento (Jakob Nielsen and Kaufmann 1993).

2.3.1 Atributos de usabilidad

La usabilidad no es un atributo único o unidimensional de la interfaz de usuario, y no podría ser considerada como tal pues goza de un nivel de abstracción demasiado alto, por lo que no podría ser evaluado si es tratado como un elemento único. En (Jakob Nielsen and Kaufmann 1993) se propone descomponer usabilidad en cinco atributos que faciliten su comprensión y cuantificación.

2.3.1.1 Facilidad de aprendizaje

Este atributo hace referencia a la rapidez y facilidad con la que los usuarios se vuelven productivos con un sistema anteriormente desconocido para ellos (Lester 2008). Se mide en tiempo que a un usuario novato le toma aprender a manejar el sistema de tal forma que sea capaz de realizar las actividades principales.

2.3.1.2 Eficiencia

Es el número de tareas realizadas a través de un sistema de software por unidad de tiempo (Ferre et al. 2001). A mayor eficiencia, mayor será el número de tareas que pueden realizarse en el sistema por unidad de tiempo.

2.3.1.3 Fiabilidad

Se refiere al número de errores cometidos por el usuario mientras realiza una determinada tarea (Jakob Nielsen and Kaufmann 1993), a mayor número de errores menor grado de fiabilidad

de dicho sistema. También se refiere a la capacidad del sistema para prevenir y recuperarse de un error una vez que ha sucedido, así como a la utilidad del sistema para permitirle al usuario regresar al estado anterior al error (Whittney Quesenbery 2004).

2.3.1.4 Recuerdo en el tiempo

La capacidad del sistema para ser recordado por un usuario después de un tiempo de no haber utilizado el sistema (Jakob Nielsen and Kaufmann 1993). El sistema debe permitir al usuario recuperar la fluidez de uso sin tener que pasar por el proceso de aprendizaje nuevamente (Whitney Quesenbery 2001).

2.3.1.5 Satisfacción

Se refiere la impresión que el usuario obtiene del sistema, así como las emociones y sentimientos que surgen en él debido a su interacción con el sistema (Norman 1988). Puede considerarse como el atributo más subjetivo de todos, debido a que se basa únicamente en la impresión final del usuario acerca del sistema (Ferre et al. 2001).

2.3.2 Usabilidad como atributo de calidad de software.

Usabilidad no es solo la facilidad de uso de un sistema, o la presentación visual de la interfaz de usuario, también tiene relación con atributos de calidad más allá de la vista o percepción del usuario, aunque el desconocimiento de lo anterior es justificado hasta cierto punto debido a que muchas de las definiciones de usabilidad hacen énfasis en los aspectos visuales del sistema (Seffah and Metzker 2008).

En la sección anterior se dieron definiciones refiriéndose a usabilidad como la capacidad de un sistema para ser utilizado con facilidad. También existen definiciones de usabilidad enfocadas en remarcar la importancia de la usabilidad como una sub-característica de calidad de software.

En (International Organization For Standardization Iso 2001) se define usabilidad desde la perspectiva de calidad de software como:

“Conjunto de atributos de calidad que influyen en uso y la evaluación individual de dicho uso, por un conjunto de usuarios declarados o implícitos”.

En la **Tabla 4** mostrada a continuación, podemos observar la presencia de atributos de usabilidad dentro de los principales modelos de calidad.

Tabla 4: Usabilidad como atributo en modelos de calidad

Modelo	Atributo de Calidad	Sub-atributos
McCall (1977)	Usabilidad	- Operabilidad - Entretenimiento - Comunicación
Boehm (1978)	Ingeniería Humana	- Robustez/Integridad - Accesibilidad - Comunicabilidad
ISO 9126 (1991)	Usabilidad	- Entendimiento - Aprendizaje - Operabilidad - Atracción
Dromey (1995)	Usabilidad	
FURPS+ (1999)	Usabilidad	

2.4 Ingeniería de usabilidad

Ingeniería de usabilidad puede tomarse desde diferentes perspectivas debido a que existen diferentes teorías acerca de su estudio (Seffah and Metzker 2008). Ritter (Ritter et al. 2000) introdujo una clasificación donde propone incorporar los aspectos de IHC dentro de la ingeniería de software.

Como muchos términos dentro las ciencias computacionales, existen diferentes definiciones sobre que es la ingeniería de usabilidad.

Mayhew (Mayhew 1999) en 1999 define el término como:

“Una disciplina que provee métodos estructurados para lograr la usabilidad en el diseño de interfaces de usuario durante el desarrollo del producto.”

Otra definición de usabilidad es la propuesta por Nielsen [Nielsen 1993] donde describe a la ingeniería de usabilidad como un proceso sistematizado para la implementación de la usabilidad durante el proceso de, con la finalidad de producir software usable.

La finalidad de incluir ingeniería de usabilidad en el ciclo de vida del desarrollo de software es el aumento de usabilidad final del sistema, y de esa manera aumentar la productividad del usuario y por ende su satisfacción con el producto de software, pues este le permite llevar a cabo sus tareas de una manera más cómoda (Ferre et al. 2001).

2.4.1 Ciclo de vida de Ingeniería de Usabilidad

Al igual que en ingeniería de software, el ciclo de vida de ingeniería de usabilidad va mano a mano con el software que está siendo desarrollado y tal como existen diferentes métodos y modelos para ingeniería de software, también existen diferentes modelos de ciclos de vida para ingeniería de usabilidad (Thurnher 2004). La mayoría de dichas propuestas de ciclo de vida convergen en tres principales etapas: requerimientos, diseño y evaluación, dichas etapas fueron propuestas inicialmente por (J. Nielsen 1992).

2.4.1.1 Ciclo de vida según Nielsen

Como se mencionó anteriormente Nielsen propone tres principales etapas para el ciclo de vida de ingeniería de usabilidad y a su vez sub-divide dichas etapas en un conjunto de actividades, en la **Tabla 5** podemos observar las tres etapas y actividades de cada una de ellas.

Tabla 5: Elementos del ciclo de vida de usabilidad según Nielsen

Etapas	Actividades
Especificaciones	- Análisis de usuario - Análisis de tareas - Especificaciones de usabilidad
Diseño	- Diseño de interacción - Prototipado - Participación de usuarios

Evaluación	- Pruebas de usabilidad - Evaluación heurística
------------	--

Especificaciones

En esta etapa se definen las especificaciones de usabilidad, se debe realizar un esfuerzo para plasmar en ellas el nivel de usabilidad que se desea alcanzar con el sistema, dichas especificaciones deben ser creadas a nivel de aspectos de usabilidad que se desean cubrir ya que son estas especificaciones junto con el análisis de usuario y de tareas las que darán rumbo al proceso iterativo del desarrollo (Almanza 2012).

Análisis de usuario

Este es un paso clave para lograr la usabilidad de un sistema, se trata de estudiar a los usuario del sistema para conocer sus capacidades, caricaturescas y expectativas a profundidad pues son dichos atributos los que el software debe abarcar para ser usable para dichos usuarios (M. Nielsen et al. 2004).

El tema de análisis de usuario es tratado más ampliamente dentro de una sección específica [2.4.2] debido a su relación con la propuesta que en este trabajo de tesis se presenta.

Análisis de tareas

Si ya se conoce a los usuarios, el siguiente paso es saber ¿Qué es lo que tienen que hacer? ¿Qué es lo que realmente quieres hacer? ¿Qué proceso deben seguir para poder hacerlo?, estas y otras preguntas son las que podemos responder mediante la técnica de análisis de tareas (Crystal and Ellington 2004).

El análisis de tareas es una de las herramientas más poderosas para quienes trabajan en el campo de IHC y tiene aplicación en todas las etapas del ciclo de desarrollo, desde la especificación de requerimientos hasta la evaluación final del sistema (Dan Diaper and Sanger

2006). Esta técnica al igual que el análisis de usuario tiene su propia sección dentro de este documento [2.4.3].

Especificaciones de usabilidad

Esta actividad es realizada antes de iniciar el diseño del sistema, pues durante esta actividad se establecen objetivos cuantitativos de usabilidad que el sistema debe cumplir. De esta actividad se desprenden los requerimientos de usabilidad, en los cuales se describe la interfaz de usuario respecto a aspectos visibles como apariencia, navegabilidad, organización entre otros (Sousa and Furtado 2005).

Los requerimientos de usabilidad representan la facilidad con la que el sistema puede ser utilizado. Los requerimientos de usabilidad deben ser considerados como no funcionales, porque en esencia en ellos no se especifican partes de la funcionalidad del sistema, solo expresan como la funcionalidad debe ser percibida por el usuario (Molina and Toval 2009).

Diseño

Una vez que se tiene la información acerca del tipo de usuarios que van a interactuar con el sistema (análisis de usuario) así como la información acerca de los procesos que los usuarios deben seguir para completar sus tareas, es necesario establecer la forma en que el usuario y el sistema se van a comunicar, los artefactos que el sistema debe contener para obtener y presentar información, el tipo de navegación y el tipo de dispositivo en los cuales el sistema estará presente, esta etapa se descompone en tres actividades descritas a continuación (Gea and Gutierrez 2001).

Diseño de interacción

Durante esta actividad es necesario definir la forma en que el usuario va a interactuar con el sistema, y viceversa de manera que el usuario pueda llevar a cabo las tareas (Jakob Nielsen and Kaufmann 1993). Para esto es necesario considerar los atributos especificados como requeridos en la fase de especificaciones de usabilidad, pues gran parte de dichos atributos aplican y deben ser respetados durante la fase del diseño de interacción. También es necesario tener en cuenta las guías de diseño (si las hay) que el propio fabricante de la plataforma pone a

disposición de desarrolladores y diseñadores para ayudarlos a crear interfaces consistentes con el sistema operativo.

Prototipado

Es en esta actividad donde se realizan bosquejos sobre que componentes necesita incluir la interfaz para que el usuario pueda interactuar con el sistema [Jakob, 1993]. De acuerdo a la propuesta de (Virzi, Sokolov, and Karis 1996), los prototipos pueden dividirse en alta y baja fidelidad, la diferencia radica en el tipo de herramienta utilizado para generarlos, mientras que los de baja fidelidad son creados en papel o herramientas simples, los de alta fidelidad son creados con herramientas de software específicamente diseñadas para ese fin.

Participación de usuarios

La finalidad de esta actividad es involucrar al usuario en el diseño del sistema para obtener retroalimentación de estos, y de esa manera descartar lo que el diseñador puede considerar usable, pero que al usuario no le parece.

Evaluación

Se define como medir o identificar problemas potenciales que pueden afectar los atributos de calidad del sistema y por su tanto su evaluación final (Dumas and Loring 2008).

Pruebas de usabilidad

Consiste en pedirle a usuarios reales que realicen algunas de las tareas para las cuales está diseñado el sistema, para realizarlas deben utilizar el prototipo del sistema y de esa manera obtener retroalimentación del usuario, para lo cual pueden utilizarle muchas y muy diversas técnicas.

Evaluación heurística

Método informal para el análisis de usabilidad donde los evaluadores son personas expertas, se basa en la experiencia que puede llegar a tener los evaluadores, por lo tanto, la experiencia y conocimientos del evaluador son un factor principal.

2.4.1.2 Ciclo de vida según Mayhew

Esta propuesta fue realizada por Mayhew en 1999 (Mayhew 1999), en la cual presenta un ciclo de vida con una visión integral acerca de Ingeniería de usabilidad, las actividades que la conforma y una descripción detallada de los pasos a realizarse en cada actividad. Esta propuesta consta de 3 actividades principales, algunas de las cuales a su vez pueden descomponerse en más actividades.

Ferré resume la filosofía del ciclo de vida en los siguientes puntos (Ferré 2005):

- El diseño de la interfaz de usuario es clave
- La integración de la ingeniería de usabilidad con la ingeniería de software debe ser particularizada
- El análisis de requerimientos vale la pena
- El diseño puede aproximarse en un proceso estructurado de descomposición
- El diseño, las pruebas y el desarrollo deberían ser iterativos
- El ciclo de vida completo puede ser estratificado en subconjuntos de funcionalidad
- Hay una variedad de técnicas para llevar a cabo cada tarea del ciclo de vida
- Las técnicas alternativas hacen que el ciclo de vida sea flexible y adaptable
- Una implementación óptica del ciclo de vida requiere la participación completa de equipos multidisciplinarios

El ciclo de vida propuesto estructura las actividades en tres fases: análisis de requerimientos, diseño/pruebas/desarrollo, e instalación según se muestra en la **Figura 2**.

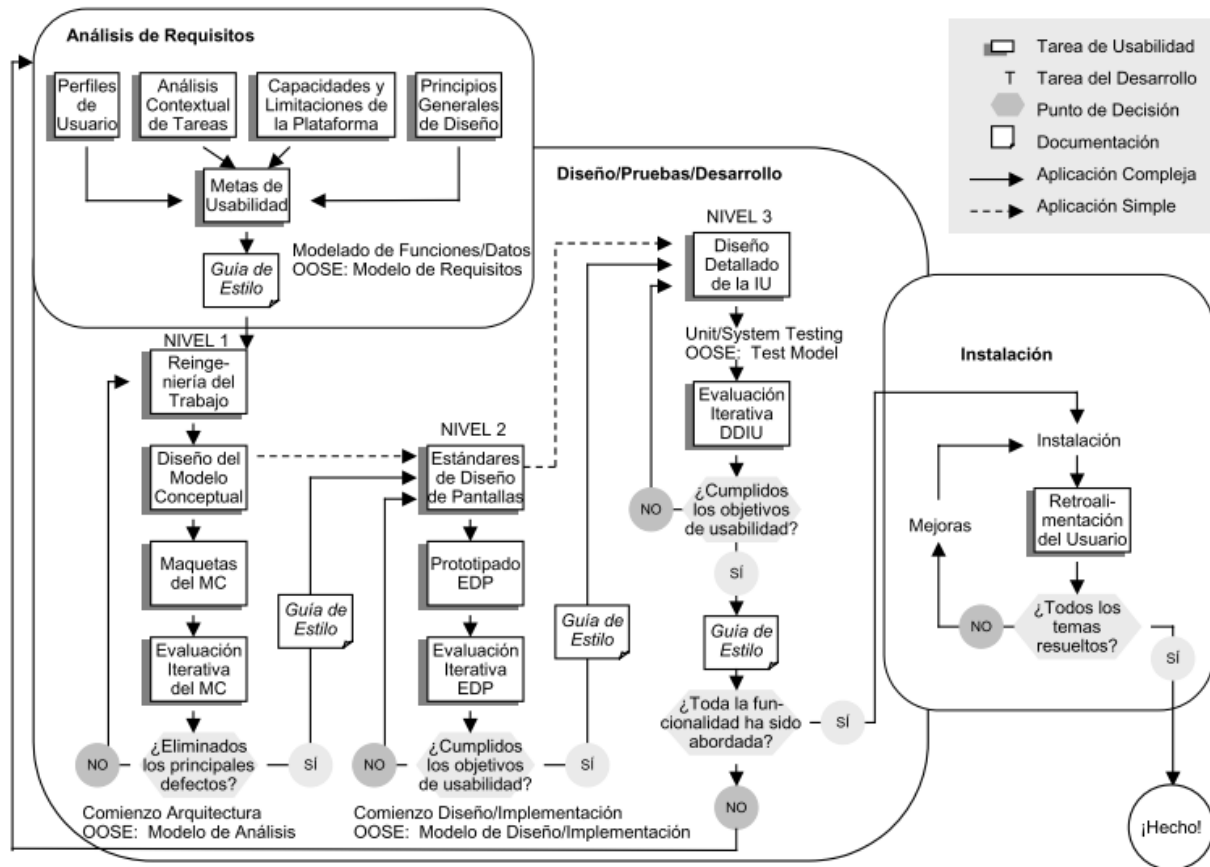


Figura 2: Ciclo de vida de usabilidad de Mayhew

Las tareas de análisis contextual, de la fase de análisis de requerimientos, se centra en proyectos en los cuales el producto ya ha sido identificado, definido y delimitado. También propone el uso de diferentes técnicas durante la fase de análisis con la finalidad de obtener un reporte detallado de las tareas que el usuario debe realizar (Seffah and Metzker 2008).

Las actividades de diseño/pruebas/implementación propuestas están centradas en el diseño y evaluación de la interfaz de usuario. Para lograrlo se divide el diseño en tres niveles: modelo conceptual, estándares de diseño de pantallas, y diseño detallado de la UI. En el nivel más básico se pretende lograr un modelo conceptual, con detalles muy generales sobre el diseño de la interfaz. En el proceso intermedio se busca lograr un diseño utilizando estándares para asegurar la consistencia y simplicidad. En el nivel más detallado se busca la selección de los

controles a utilizar, la ubicación y el formato de cada elemento, así como los tipos de fuente, colores y tamaños (Mayhew 1999).

Durante la fase de instalación se busca no solo entregar y configurar el producto de software con el cliente, si no también obtener retroalimentación de los usuarios que utilizan el sistema, información que puede ser utilizada para mejorar el sistema en una segunda iteración (Thurnher 2004).

2.4.2 Modelos de usuario

El modelado de usuario es un área de creciente importancia, sobre todo para aplicaciones industriales (Vassileva 1994). Un modelo de usuario puede definirse como la representación de sus características dentro de un entorno computacional (Rich 1979).

El objetivo de un modelo de usuario es representar valores cuantificables de diversos factores humanos de los usuarios, para que una aplicación haga uso de dicha información para predecir o aumentar el grado de usabilidad que tendrá el sistema de software (Ritter et al. 2000).

Un factor humano es una característica de usuario que de alguna forma tiene relevancia por su intervención en la interacción entre el usuario y un sistema computacional, el número de factores humanos es muy amplio, y varía en cada propuesta de modelo de usuario (Biswas and Robinson 2010).

Una de las principales diferencias entre los modelos es el tipo de información que representan y/o almacenan y la forma en que actualizan y mantienen al día dicha información. A continuación se presentan tres principales tipos de modelos de usuarios (Weinschenk 2011).

- **Modelo de usuario estático:** La información que representa es recogida y almacenada, el sistema no controla ni se entera de los cambios en la forma que utiliza el usuario para interactuar con el sistema (A. Johnson and Taatgen 2005).
- **Modelo de usuario dinámico:** El sistema controla la forma en que el usuario interactúa con el sistema y el modelo se actualiza como resultado, lo que resulta en un modelo de usuario evolutivo (Hothi and Hall 1998).

- **Modelo basado en un estereotipo:** Es la representación de un usuario ficticio utilizando información estadística de un grupo demográfico de usuarios (Rich 1979).

2.4.3 Análisis de tareas

El análisis de tareas es un elemento clave en la mayoría de los trabajos relacionados a la interacción humano computadora gracias a su preocupación por el desempeño en el trabajo, y es esto lo que distingue al análisis de tareas de otras propuestas. Aunque parezca increíble, los fundamentos del análisis de tareas datan del siglo anterior y una de las razones para su longevidad es que los métodos del análisis de tareas se han adaptado a las necesidades del momento (Dan Diaper and Stanton 2004). La afirmación de que el análisis de tareas es probablemente el método más poderoso disponible para quienes trabajan con IHC y tiene aplicación en todas las etapas del desarrollo de sistemas, desde etapas tempranas como la especificación de requerimientos, hasta etapas finales como es la evaluación final del sistema (D. Diaper et al. 1990).

El concepto de tarea, describe un conjunto de pasos y acciones que usan las personas para realizar una actividad o cumplir un objetivo. De acuerdo a (Mayhew 1999), una tarea es para IHC lo que un caso de uso para el desarrollo de software.

Dentro del análisis de tareas se hace uso de la técnica de análisis de usuario, con ello se logra conocer las actividades que el usuario realiza actualmente para llevar a cabo su trabajo, y el orden de actividades que sigue, teniendo identificadas las actividades que realiza, se puede decidir qué actividades pueden ser asistidas por un sistema computacional.

Existen diferentes propuestas de técnicas para la representación de la información dentro del análisis de tareas, dos de las más importantes son: análisis jerárquico de tareas (HTA por sus siglas en inglés) y método objetivos operadores, métodos y reglas de selección (GOMS por sus siglas en inglés).

Análisis jerárquico de tareas (Annett 2004):

El objetivo principal de esta técnica es la describir las tareas en términos de operaciones y planes, los cuales son vistos desde la perspectiva del usuario.

Operaciones: Es la unidad fundamental del análisis. Está definida por una o más metas por las cuales las personas, las actividades que realiza y las condiciones en las cuales se realiza.

Planes: Es el orden en que deben realizarse cada una de las operaciones y sub-operaciones que conforman la tarea.

La siguiente lista representa los principales pasos recomendados por el autor para llevar a cabo el análisis de tareas jerárquico.

1. Decidir el propósito del análisis
2. Obtener un acuerdo entre los interesados en la definición de las metas de las tareas y criterios de medición de las mismas.
3. Identificar las fuentes de información de las tareas y elegir los medios para adquirir los datos.
4. Adquirir los datos y esquematizar el diagrama de descomposición del proyecto.
5. Evaluar el diagrama con los interesados.
6. Identificar las operaciones significantes de cada una de las actividades.
7. Generar, y de ser posible, crear hipótesis acerca de los factores que afectan la aprendibilidad y desempeño.

Modelo GOMS para el análisis de tareas (Kieras, 2004):

El modelo GOMS es una descripción del conocimiento procedural que el usuario debe tener para poder llevar a cabo las tareas en un dispositivo o en un sistema; es una representación del conocimiento del *cómo hacerlo* que se necesita para completar las tareas. Una descripción breve sobre el modelo GOMS sería decir que el modelo consiste en una descripción de los métodos necesarios para alcanzar las metas. Los métodos son una serie de pasos conformados por operaciones que el usuario realiza. Un método puede necesitar de sub-metas para poder llevarse a cabo, por lo tanto existe una jerarquía dentro de los métodos. Si existe más de un método para una sola meta, entonces las reglas de selección eligen los métodos apropiados dependiendo del contexto.

La forma más simple del modelo GOMS es el modelo a nivel de teclas presionadas (KLM por sus siglas en inglés), propuesto en 1980 por Card, Moran, y Newell, en el cual el tiempo requerido para realizar la tarea es la suma de los tiempos que le toma al usuario realizar cada una de las acciones de la tarea, dichos tiempos son medidos a nivel de pulsaciones de teclas.

2.5 Interfaz de usuario

Una interfaz de usuario es la porción de un sistema computacional interactivo que se encarga de la comunicación entre el usuario y el sistema. El diseño de la interfaz de usuario (UI por sus siglas en inglés) debe incluir cualquier aspecto del sistema que es visible al usuario (Jacob 2000).

En los inicios de la computación, los usuarios eran expertos, y las interfaces de usuario consistían en tarjetas perforadas. Hoy en día una gran variedad de usuarios no especialistas en el área computacional utilizan las computadoras, y las interfaces de usuario son compuestas generalmente por teclado, mouse y pantallas gráficas.

Una buena interfaz de usuario no puede ser implantada en un sistema después que toda la funcionalidad de este fue creada, sino que tiene que ser considerada en el proceso de diseño desde el inicio. Un diseño adecuado de la interfaz de usuario puede hacer una diferencia substancial en el tiempo que le toma al usuario aprender a utilizar el sistema, desempeño del sistema, tasa de errores, así como en la satisfacción del usuario con el sistema (M. Nielsen et al. 2004).

Existen muchas razones por las que la calidad de la interfaz de usuario es crítica para el éxito de un sistema interactivo (Borchers 2001):

- Sistemas críticos requieren interfaces rápidas, libres de errores y que ayuden a realizar las actividades de manera eficiente.
- Aplicaciones de oficina, hogar y de entretenimiento requieren interfaces fáciles de aprender, con una tasa de errores baja y con alto grado de satisfacción para el mercado al que estén dirigidas.

- Sistemas cooperativos, creativos y de exploración necesitan cumplir con muy altas expectativas de los usuarios, por lo que un buen diseño de la interfaz de usuario es clave en este tipo de sistemas.

2.5.1 Interfaces adaptativas

Crear una interfaz de usuario que se acomode a las necesidades de los diferentes usuarios, mientras que se mantiene el nivel de usabilidad ha sido desafío por largo tiempo, aun entre un grupo de usuarios homogéneos, debido a que aún las más pequeñas diferencias entre ellos puede hacer que el sistema sea más usable para un usuario que para otro (Peissner, Schuller, and Spath 2011). Como se menciona en (Gajos, Weld, and Wobbrock 2010), las propuestas que intentan solucionar dicho predicamento, como *universal design* (inglés para diseño universal), *inclusive design* (inglés para diseño inclusivo) y *design for all* (inglés para diseño para todos) intentan crear tecnologías que tienen propiedades adecuadas a tantos usuarios como sea posible. Aunque esto sea una meta plausible, las propuestas anteriores son imprácticas en muchos casos, particularmente cuando los sistemas de software complejos (Bergman and Johnson 1995).

Una única interfaz de usuario apropiada para todos los usuarios no es tarea fácil, y en muchos de los casos es casi imposible (Gajos, Weld, and Wobbrock 2010), es por esto que surge la técnica de interfaces adaptativas como solución al mencionado problema (Lavie and Meyer 2010; Peissner, Schuller, and Spath 2011).

Una interfaz de usuario adaptativa es un artefacto de software que mejora su habilidad para interactuar con los usuarios mediante la construcción de un modelo de usuario, y adaptando la interfaz a las características de usuario presentes en el modelo del usuario. Esta definición deja claro que una interfaz adaptativa no existe en aislamiento, sino que es diseñada para interactuar con el usuario, con la finalidad de mejorar la experiencia de uso del usuario con el sistema, y en general la usabilidad general del mismo (Langley and Hirsh 1999).

De acuerdo con (Schneider-Hufschmidt et al. 1993), las interfaces adaptativas fueron inicialmente desarrolladas para hacer frente a cuatro principales preocupaciones:

- Un sistema de software es utilizado por usuarios con diferentes requerimientos.
- Un sistema es utilizado por un usuario con requerimientos cambiantes.

- Un usuario trabaja en un entorno de sistema cambiante.
- Un usuario trabaja en diferentes entornos de sistema.

Las adaptaciones en una interfaz de usuario pueden ocurrir en tres áreas del sistema (Brusilovsky and Maybury 2002):

- 1.- La selección del contenido que debe ser mostrado o recomendado al usuario
- 2.- La presentación de la información, incluyendo colores, tamaño de fuente y disposición.
- 3.- Navegación dentro de la cual se definen las posibles rutas que el usuario puede tomar dentro de la aplicación con la finalidad de acceder a cierta información o funcionalidad.

Desde la perspectiva del usuario final, el utilizar interfaces adaptables puede ser benéfico y resultar en adaptaciones efectivas, sin embargo desde la perspectiva de los desarrolladores del sistema, el implementar la adaptabilidad no es tarea sencilla (Peissner, Schuller, and Spath 2011). El principal desafío es que los beneficios de una correcta adaptación deben sobre pasar los costos y los efectos negativos en la usabilidad (Findlater and Gajos 2009).

En (Peissner, Schuller, and Spath 2011) se marcan como los principales requerimientos que debe cumplir una interfaz adaptable:

Adaptar contenido, navegación y presentación: El perfil de usuario sobre el cual se basan las adaptaciones debe cubrir percepción, cognición, habilidades motoras y deficiencias de los usuarios. Como consecuencia los tres puntos mencionados anteriormente deben estar sujetos a adaptaciones.

Modularidad y extensibilidad: Un diseño modular es necesario para manejar la gran cantidad de posibles perfiles de usuario y las respectivas interfaces que soporten dichos perfiles.

Aprender y adaptar en tiempo de ejecución: Interfaces de usuario adaptativas deben ser lo suficientemente inteligente y evitar configuraciones tediosas, para lo cual el sistema debe encargarse de recolectar la información necesaria durante la ejecución de la aplicación.

Consistencia y significancia en las interfaces de usuario: Las interfaces adaptivas para interfaces de usuario multi-dimensionales pueden incurrir en inconsistencias fácilmente. Por lo que utilizar reglas y mecanismos efectivos para adaptaciones adecuadas son fundamentales para alcanzar un nivel de consistencia adecuado.

Lavie en (Lavie and Meyer 2010) propone una serie de aspectos que deben ser considerados en el diseño del sistema para alcanzar un nivel mínimo de adaptabilidad:

- *La tarea que el usuario debe completar:* es indispensable considerar la tarea que el usuario intenta realizar, debido a la interfaz de usuario es la que debe permitirle realizar dicha tarea.

- *El usuario y sus características:* el usuario junto con sus características deben ser tomadas en consideración para las interacciones a realizarse dentro del sistema.

- *Rutina vs no rutina:* Identificar cuáles de las tareas y/o acciones serán realizadas de manera rutinaria.

- *Nivel de adaptación que se desea lograr:* Es necesario identificar si se desea una adaptación manual, totalmente adaptable, totalmente adaptiva o cualquier nivel intermedio que se desee obtener.

Un concepto que está muy ligado al de interfaces adaptivas, y que continuamente se usan como sinónimo, aun cuando no lo son, es el de interfaces adaptables. Una interfaz adaptable es aquella que el desarrollador o diseñador adapta en la etapa de diseño y/o implantación de manera manual. Mientras que una interfaz adaptiva se adapta a sí misma automáticamente basada en un perfil de usuario dinámico y en información obtenida del contexto en el cual se encuentra el usuario (Holzinger, Geier, and Germanakos 2010).

El término *contexto* dentro de interfaces adaptivas tiene muchos significados diferentes e incluye la experiencia del usuario, habilidades especiales de los usuarios, necesidades y sentimientos en el momento de uso, así como factores ambientales (luz ambiental, ruido externo, ubicación geográfica, etc.) y el estado actual del sistema o dispositivo (Andrina Grani 2007).

2.6 Motor de adaptación

Si bien existen una gran cantidad de propuestas para sistemas con interfaces adaptativas, en la gran mayoría de ellas se profundiza en los motivos que llevaron a utilizar interfaces adaptativas, y en los beneficios obtenidos de ellas (Balint, 1995; Lavie & Meyer, 2010; Stephanidis et al., 1998; Stephanidis, Paramythis, et al., 1998; Zarikas, 2007), en muy pocas propuestas de las que pudimos analizar se explica la fase o componente encargada de realizar dicha adaptación.

A dicho componente se le dio el nombre de *motor de adaptación*, el cual representa un punto clave de la propuesta presentada en este, puesto que es el encargado de realizar, en tiempo ejecución, las adaptaciones pertinentes a la interfaz de usuario con la finalidad de adaptar está a las características del usuario. Diferentes autores se refieren a este componente de diferentes maneras.

Vogel & Giese en (Vogel and Giese 2013) utilizan el término *motor de adaptación* para referirse al componente encargado de controlar las adaptaciones subyacentes que serán realizadas a la interfaz de usuario, en el que se describe la adaptación mediante el uso de modelos de tiempo de ejecución que representan los aspectos relevantes del sistema adaptable, tales como análisis y planeación de las operaciones a realizar en tiempo de ejecución.

Henricksen en (Henricksen and Indulska 2001) utiliza el término *mecanismo de adaptación* para al proceso mediante el cual se realizan adaptaciones a la interfaz gráfica de usuario.

En (Smits and De Bra 2011) se utilizan de igual manera el término *motor de adaptación* para referirse al componente de su propuesta encargado de realizar cambios a la interfaz de usuario. También describen algunos aspectos del motor de adaptación que son clave para nuestra propuesta. Describen al motor de adaptación como un componente que toma parte en tiempo de ejecución, basado en configuraciones realizadas por los programadores en la fase de implementación, o utilizando configuraciones dinámicas como reglas de adaptación (Smits and De Bra 2011).

III

MÉTODO PARA LA GENERACIÓN AUTOMÁTICA DE INTERFACES DE USUARIO ADAPTIVAS.

"I want to put a dīng in the universe."

- Steve Jobs -

3 MÉTODO PARA LA GENERACIÓN AUTOMÁTICA DE INTERFACES DE USUARIO ADAPTIVAS.

Si se desea obtener una interfaz que sea usable para un grupo de usuarios con diferentes capacidades, experiencias y expectativas, una opción es diseñar interfaces para cada estereotipo de usuario, otra opción es utilizar la técnica de interfaces adaptativas, en la cual una interfaz es capaz de adaptarse a diferentes usuarios. En este trabajo se propone un Método para la Generación Automática de Interfaces de Usuario Adaptativas (**MEGAIA**).

La propuesta consiste en una serie de pasos derivados de ingeniería de software así como de ingeniería de usabilidad y de un motor de adaptación basado en reglas, para automatizar la adaptabilidad de interfaces de usuario.

El objetivo es aumentar la usabilidad de los sistemas de software interactivos mediante la adaptación de la interfaz a las capacidades y preferencias del usuario, sin dejar de lado la naturaleza de la tarea. La **Figura 3** muestra una representación esquemática de cómo un prototipo de interfaz es adaptado a diferentes perfiles de usuario para crear una interfaz de usuario específica para cada perfil.

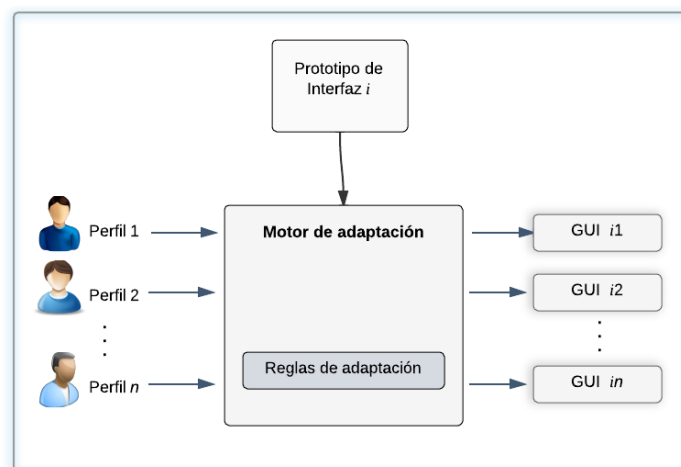


Figura 3: Adaptación de una interfaz a diferentes usuarios.

3.1 Elementos principales de MEGAIA

Como parte de la propuesta se plantean una serie de postulados y esquemas. También se propone hacer uso de técnicas de ingeniería de usabilidad, como son el análisis de usuario y análisis de tareas, que en conjunto con un motor de adaptación permite crear interfaces de adaptadas a las características del usuario.

Los elementos principales que conforman la propuesta son: un modelo integral de usuario, tareas, prototipos de interfaz, reglas de adaptación y un motor de adaptación.

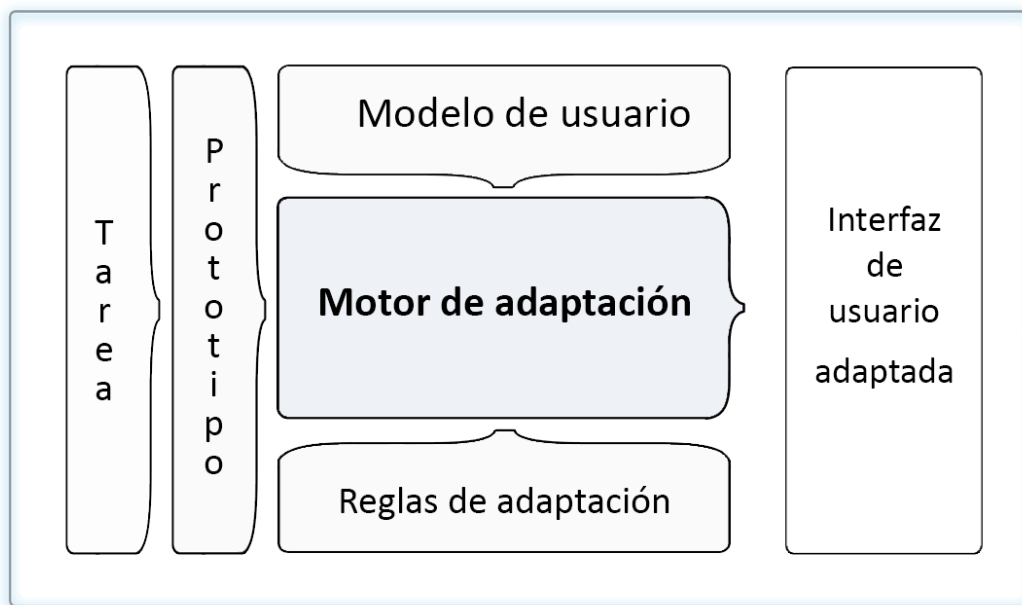


Figura 4: Elementos principales de MEGAIA

3.1.1 Modelo integral de usuario

Como se menciona en la sección 2.3, el primer paso en el ciclo de vida de ingeniería de usabilidad es estudiar a los usuarios con la finalidad de conocer las características de las personas que estarán utilizando el sistema (Jakob Nielsen and Kaufmann 1993), para almacenar la información resultante de dicho análisis, se utiliza la técnica de modelos de usuario.

Existen diferentes propuestas de modelos de usuario, cada una de ellas se centran en aspectos específicos del usuario, por ejemplo; existen propuesta centradas solamente en aspectos cognitivos de la persona, también existen otras propuestas centradas solo en aspectos

psicológicos. Pero a nuestro conocimiento no existe un modelo que integre todas las características de usuario que tienen o puedan tener relación en el proceso de interacción del humano con la computadora.

Lo anterior dio pie a realizar la propuesta de un Modelo Integral de Usuario (MIU) (Mejía et al. 2012) basado en el modelado de las características generales del usuario, que se integra a partir de trabajos anteriores (Zhang et al, 2004; Biswas et al, 2005; Johnson et al, 2005; Weinschenk, 2011; Stuart et al, 1983).

La propuesta fue probada en (Andres Mejía 2013), donde por medio de experimentación se llegó a la conclusión que:

- El modelo integral de usuario es implementable como parte de una aplicación.
- Que al ser implementado correctamente dentro de una aplicación, ayuda al incremento de la usabilidad general de la misma, pues en él se contienen la información necesaria para interactuar con el usuario de acuerdo a sus capacidades y características.

El modelo integral de usuario está compuesto por cinco principales factores: físicos, cognitivos, demográficos, experiencia y psicológicos (Figura 5).

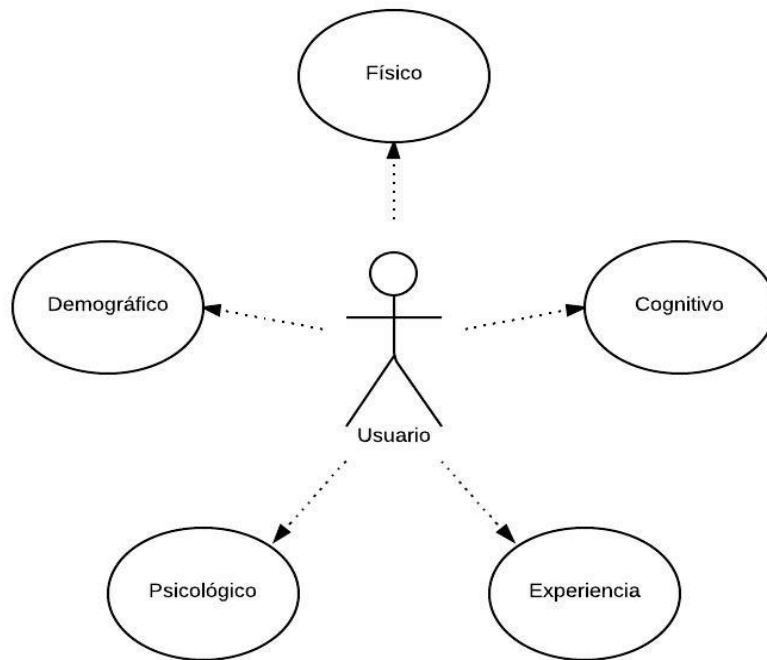


Figura 5: Factores constituyentes del modelo integral de usuario.

La idea general detrás de esta propuesta es incluir la mayor cantidad de atributos de usuario que intervienen en el proceso de interacción humano computadora, con la finalidad de tener un modelo de usuario lo más aproximado posible a la persona que está usando el sistema, y de esa manera poseer toda la información necesaria para tomar decisiones de usabilidad que ayuden a facilitar la interacción del usuario con el sistema y de manera general mejorar su experiencia y percepción del mismo.

La información del usuario necesaria para alimentar el modelo integral de usuario puede ser obtenida utilizando el análisis de usuario, técnica perteneciente a ingeniería de usabilidad.

3.1.1.1 Físico

En esta sección se consideran las partes físicas del usuario que tienen o pueden, en un futuro dado, tomar parte en el proceso de interacción del usuario con un sistema de software interactivo.

Dentro de los factores físicos se encuentran las partes del cuerpo, así como los 5 sentidos que poseemos los seres humanos. Como se mencionó en el párrafo anterior, es posible que actualmente no todos los atributos considerados tomen partido en el proceso de interacción humano computadora, como por ejemplo el olfato, o la capacidad de sentir dolor, pero se han enlistado igualmente para ser tomados en cuenta previendo el avance de la tecnología.

Como puede observarse en la **Figura 6**, para las partes del cuerpo se consideran tres propiedades:

- Presencia: para denotar si el usuario posee o carece de dicha extremidad.
- Movimiento: para expresar si la extremidad posee la capacidad de movimiento
- Control: para reflejar si la capacidad de movimiento de la extremidad es voluntaria.

También podemos observar la relación obvia existente entre un sentido y la parte del cuerpo sobre la que se ejecuta dicho sentido.

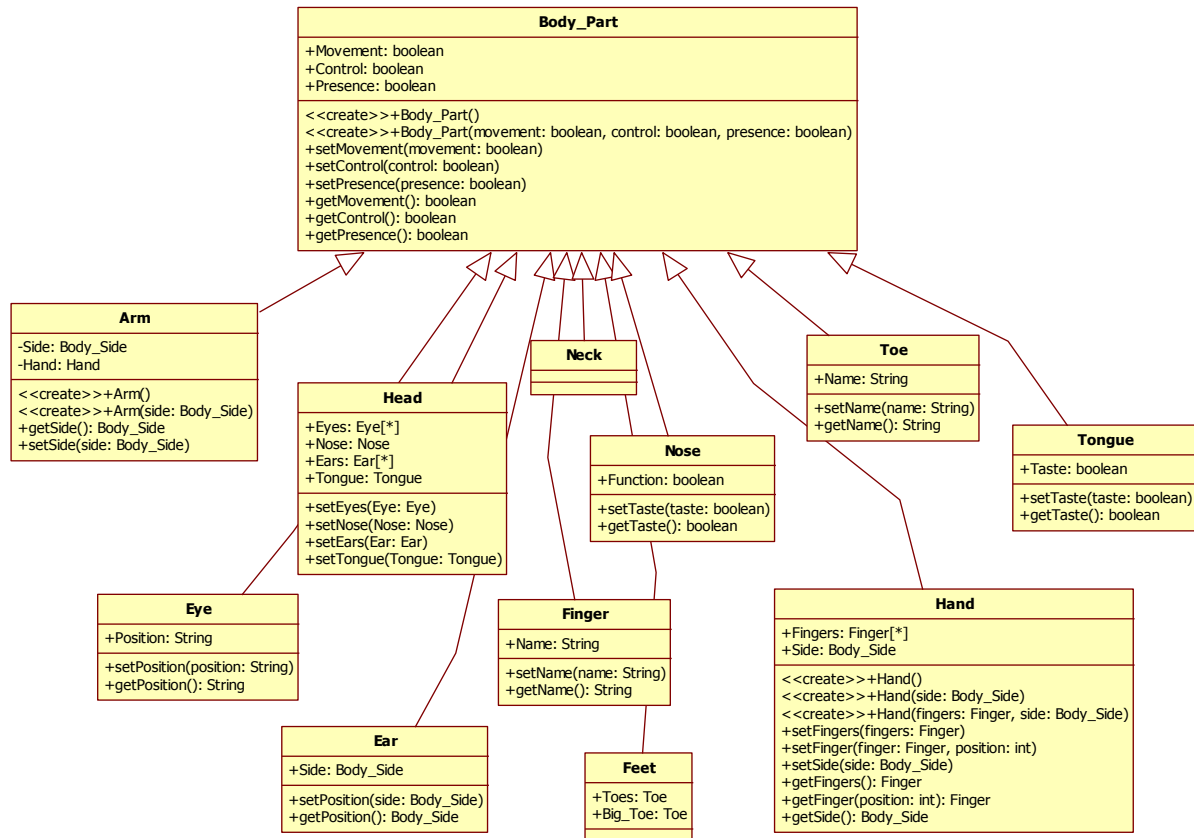


Figura 6: Propuesta de aspectos físicos del modelo integral de usuario

3.1.1.2 Cognitivo

Dentro de los factores cognitivos son consideradas habilidades cognitivas y de inteligencia (**Figura 7**), como la memoria y la atención, debido a que dichos atributos toman partido en el proceso de interacción del usuario con el sistema y por lo tanto están indirectamente relacionados con los atributos de usabilidad, por ejemplo: la memoria a corto plazo de un usuario puede afectar la percepción hacia el sistema.

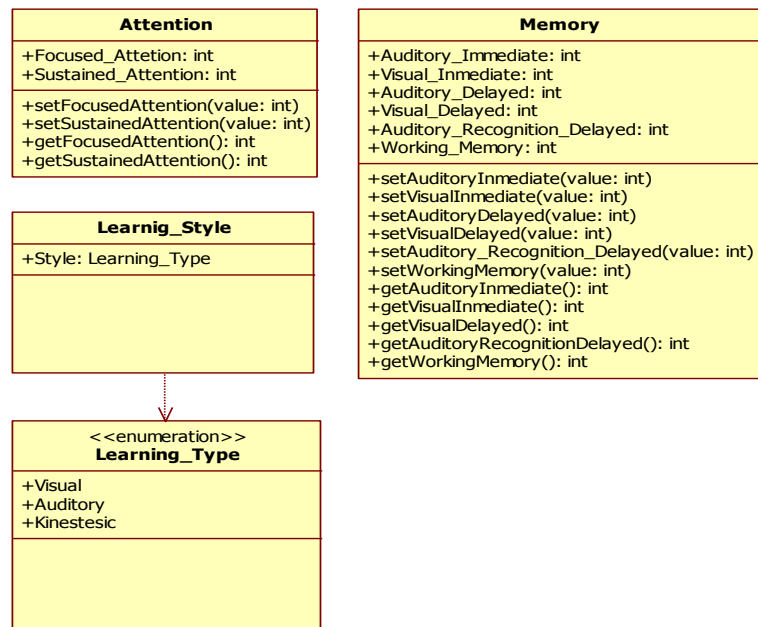


Figura 7: Propuesta de aspectos cognitivos considerados en el modelo integral de usuario

3.1.1.3 Demográfico

En esta sección se considera información estadística del usuario como edad, sexo, historial educativo, país de nacimiento entre otros. Esta información puede ser utilizada para adaptar la interfaz del usuario en base a preferencias culturales, entre otras. Por ejemplo los colores utilizados dentro de la interfaz pueden tener diferente significado dependiendo de la cultura del usuario. La **Figura 8** muestra algunos de los elementos considerados dentro de esta sección.

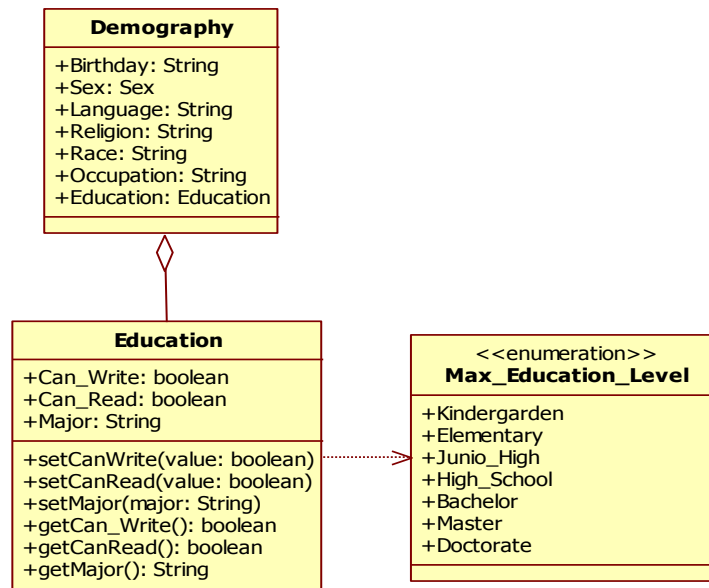


Figura 8: Aspectos demográficos del modelo integral de usuario

3.1.1.4 Experiencia

Toma en cuenta atributos de usuario respecto a experiencias con dispositivos, experiencia laboral, conocimiento del dominio del problema, software utilizado, etc. (**Figura 9**).

Dichos atributos son útiles debido a que el usuario crea modelos mentales de lo que espera de un nuevo sistema de software baso en experiencias previas con software o dispositivos similares, por lo que es importante contar con la estructura capaz de almacenar información que refleje las expectativas que el usuario puede tener para con el sistema (Designer, To, and About n.d.; J. Johnson 2010).

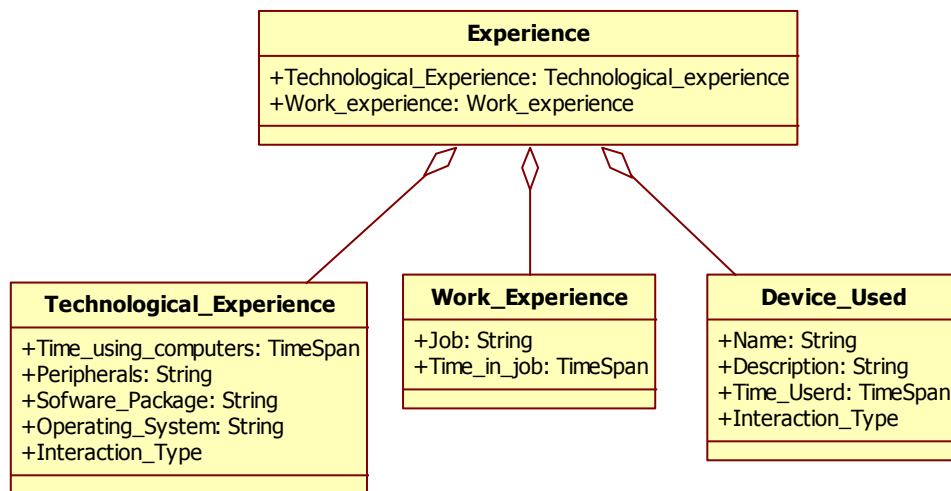


Figura 9: Atributos de experiencia considerados en el modelo integral de usuario.

3.1.1.5 Psicológico

Como su nombre lo indica, en esta sección se consideran atributos psicológicos del usuario, aquellos que puedan tener relación con la conducta de la persona, preferencias personales y emociones.

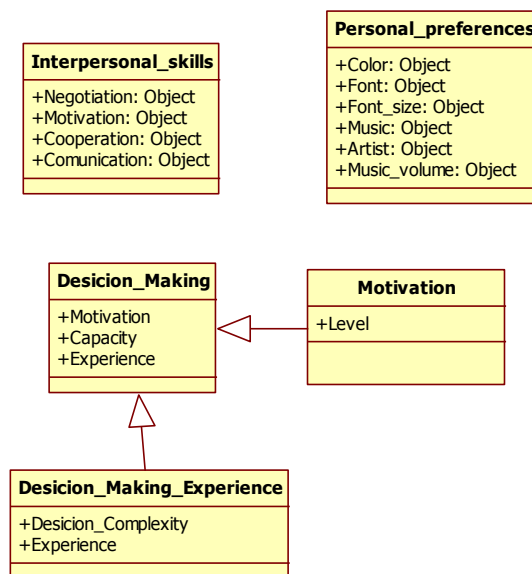


Figura 10: Atributos psicológicos considerados en el modelo integral de usuario.

3.1.2 Tareas

Una tarea dentro del contexto de ingeniería de software es una actividad compuesta por una secuencia de sub actividades, que el usuario ejecuta dentro de un entorno computacional para lograr un objetivo específico. Un ejemplo sería: *iniciar sesión*, donde el objetivo de la tarea es ser autenticado dentro un sistema de software como un usuario válido.

Para este trabajo se adopta la propuesta realizada en (Juárez-Ramírez et al. 2011) donde se proponen tres niveles de detalle para una tarea:

- Tarea principal
- Subtarea
- Acción

De la lista anterior podemos observar que el nivel más detallado propuesto para definir una tarea son las acciones que se realizan entre el usuario y el sistema. Dichas acciones son clasificadas en base al manejo de la información en tres tipos:

- **Acciones de entrada:** Son aquellas cuyo objetivo es que el usuario introduzca datos al sistema. Por ejemplo: *El usuario ingresar su fecha de nacimiento.*
- **Acciones de salida:** En las cuales el sistema es el que presenta información al usuario. Por ejemplo: *El sistema presenta al usuario su edad.*
- **Acciones de indicación:** Cuando el usuario comunica al sistema una decisión tomada, o le pide al sistema realizar algo. Por ejemplo: *El usuario selecciona una opción de una lista.*

Para obtener el conjunto de acciones de una tarea, el tipo de acción y el tipo de dato de cada una de ellas, podemos utilizar actividades pertenecientes a ingeniería de software o actividades de ingeniería de usabilidad.

En ingeniería de software podemos obtener el conjunto de acciones utilizando los flujos de los casos de uso. Mientras que en Ingeniería de usabilidad podemos hacer uso de la técnica de análisis de tareas.

Al igual que en la propuesta realizada por (Juárez-Ramírez et al. 2011), en el análisis de tareas jerárquico (Annett and Duncan 1967) se descompone una tarea en sub-tareas, las cuales a su vez están formadas por un conjunto de acciones. En el modelo de estructuras de conocimiento de las tareas (del inglés Task Knowledge Structure) presentado en (P. Johnson 1989) cuyas estructuras taxonómicas representan las propiedades de los objetos de transición de las tareas y las acciones asociadas a estos. Como se demuestra en los trabajos anteriores, uno de los elementos más comunes utilizado para detallar la composición de una tarea son el conjunto de acciones. Basado en lo anterior, a continuación se proponen dos postulados.

Postulado 1: Una tarea está formada por un conjunto no vacío de acciones.

Postulado 2: El conjunto de acciones son un elemento indisoluble de una tarea.

Como se describe al inicio de esta subsección, las acciones de *entrada* son en las que el usuario proporciona datos al sistema, y en las de *salida* es el sistema el que comunica datos al usuario (Bentley et al. 1997; Juárez-Ramírez et al. 2011). Tomando como base lo anterior, y el hecho de hecho que a cada dato, al ser representado dentro de un sistema computacional le corresponde un tipo dependiendo de la información que este represente, a continuación se propone el tercer postulado.

Postulado 3: A cada acción de entrada o salida le corresponde un tipo de dato, dicho dato obtiene o comunica al usuario respectivamente.

3.1.3 Prototipos de interfaz

A lo largo de este trabajo de tesis se utiliza el término *prototipo de interfaz* para referirse a aquellas interfaces diseñadas comúnmente, donde el diseñador selecciona, posiciona y estiliza los componentes que cree necesarios para el usuario lleve a cabo las tareas. En este trabajo se les menciona como prototipos pues la interfaz gráfica contenida en dichos archivos no necesariamente representa a la interfaz final que será presentada al usuario, puesto que el prototipo de interfaz pasará por un proceso de adaptación donde será ajustado a las características del usuario antes de ser presentado en pantalla.

Como se define en el párrafo anterior, prototipo de interfaz es el nombre que se le da a una interfaz de usuario que será ajustada a las características del usuario en un proceso de adaptación, por lo cual podemos utilizar el término prototipo de interfaz para referirnos a los prototipos utilizados en ingeniería de software los cuales solo contienen componentes pero carecen de diseño visual, así como también podemos referirnos a una interfaz de usuario totalmente desarrollada, que esta lista para presentarse al usuario, pero se ha optado por incluirla en el proceso de adaptación.

Un prototipo de interfaz está compuesto por un conjunto de componentes (Williams and Probert 2002) los cuales permiten al usuario realizar las acciones que conforman la tarea, a estos componentes se les conoce como *componentes de interacción*.

3.1.3.1 Componente de interacción

Un componente de interacción es un elemento gráfico de interfaz de usuario, predefinidos por los entornos de programación, y pensado para permitir a un usuario realizar tareas comunes como mostrar texto en pantalla, seleccionar elementos de una lista o ingresar datos al sistema. Se propone el **Esquema 1:** como una estructura para la representación de los componentes de interacción.

Esquema 1: Sea $\Phi (\eta, \delta, \alpha, \delta, \rho)$ el conjunto de elementos que definen a un componente de interacción.

Donde:

η : define el nombre del componente.

δ : representa una descripción del componente.

α : representa en tipo de acción que realiza el componente.

δ : representa el tipo de dato maneja el componente.

ρ : describe el sistema operativo en el cual puede implementarse el componente.

Partiendo del **Postulado 1:**, donde se denota que una tarea es conformada por un conjunto de acciones, y aunando a esto la definición que se da en el punto 3.1.3.1 para un componente de interacción, se proponen los dos siguientes postulados.

Postulado 4: *Una acción de una tarea se realiza haciendo uso de uno o más componentes de interacción.*

Postulado 5: *Un prototipo de interfaz está compuesto por un conjunto de componentes de interacción, cada uno de los cuales está asociado a una acción del conjunto de acciones de la tarea.*

3.1.3.2 Generación automática de prototipos de interfaz

La construcción automática de un prototipo de interfaz es posible gracias a la relación existente entre *tarea*, *acción* y *componente de interacción*, gracias a la cual es posible obtener el conjunto de componentes de interacción necesarios para permitir al usuario llevar a cabo las acciones de la tarea. A continuación se proponen una serie de acciones mediante las cuales podemos automatizar la generación de prototipos de interfaz.

1. **Crear un catálogo de componentes:** Es necesario integrar un catálogo con los componentes de interacción disponibles en la plataforma para la cual se desea realizar el prototipo de interfaz.
2. **Listar las acciones de la tarea:** Debemos contar con el conjunto de acciones que conforman la tarea, así como el tipo de acción y el tipo de dato de cada una de las acciones.
3. **Mapear cada acción a un componente de interacción:** Durante esta actividad es necesario seleccionar un componente de interacción que permita al usuario llevar a cabo cada acción, respetando la correspondencia entre el tipo de la acción con el tipo de acción que el componente es capaz de realizar, así como la correspondencia entre el tipo de dato correspondiente a la acción con el tipo de dato que maneja el componente.
4. **Crear el prototipo de interfaz:** Unir el conjunto de componentes de interacción seleccionados en una interfaz de usuario para formar el prototipo de interfaz.

La **Figura 11** es una representación esquemática del proceso de selección de los componentes de interacción para formar un prototipo de interfaz.

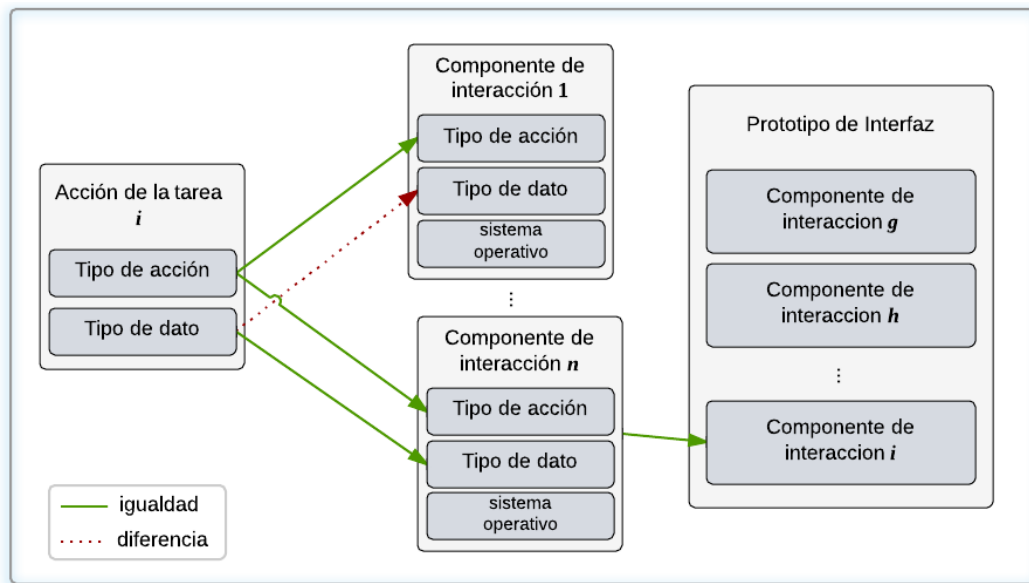


Figura 11: Selección de componentes de interacción para crear prototipos de interfaz

La selección de los componentes de interacción para construir un prototipo de interfaz está dada por la regla:

$$\text{IF } \Phi_{\alpha} \text{ is } X_{\alpha} \text{ and } \Phi_{\delta} \text{ is } X_{\delta} \text{ then } \Phi \in A$$

Donde:

A: es el conjunto de componentes de interacción que conforman el prototipo de interfaz

Φ : es el componente de interacción

X : es la acción de la tarea

α : es el tipo de acción

δ : es el tipo de dato

3.1.4 Reglas de adaptación

Las reglas de adaptación, son un conjunto de medidas que marcan las acciones específicas a seguir en un punto dado del proceso de adaptación de la interfaz de usuario. Como se define en (Vogt and Meier 2010), las reglas de adaptación se utilizan para mapear interfaces de usuario abstractas a diferentes modalidades y optimizar la interfaz para el usuario final.

Basados en la propuesta de reglas de adaptación en (Almanza 2012), proponemos una estructura de reglas del tipo *IF...THEN*, que utilizan como entrada atributos de usuario y acciones de las tareas para producir una salida.

3.1.4.1 Elementos de entrada de las reglas

1. Atributo(s) de usuario

Este elemento de entrada puede constar de un solo atributo de usuario o de un conjunto de ellos. El valor para cada uno de los atributos debe estar dentro del rango de valores permitidos en el modelo de usuario para el atributo en cuestión.

2. Tipo de acción

Se refiere al tipo de acción que el usuario ejecutará, los tipos de acciones fueron detallados en la sección 3.1.2 de este documento.

3.1.4.2 Elemento de salida de la regla

Como elemento de salida de las reglas obtenemos un componente adaptivo de interacción adaptado a las características del usuario, mediante el cual se puede llevar a cabo el tipo de acción marcado en la entrada de la regla. Dicho componente es descrito a continuación y representados por el **Esquema 2:**.

Componente adaptivo de interacción (CADI)

Un componente adaptivo de interacción es un elemento de interfaz, basado en un componente de interacción, cuyas propiedades de presentación (colores, tamaños, posición) y/o método de interacción con el usuario (puntero, táctil, dictado) han sido adaptadas a las

características específicas del usuario que está utilizando el sistema, para permitirle realizar una acción (entrada, salida o indicación) de una manera usable.

La estructura propuesta para un componente de interacción está definida por el siguiente esquema:

Esquema 2: Sean $\Upsilon(\sigma_{\Delta}, \delta_{\Delta}, \lambda)$ el conjunto de los elementos que describen a un componente adaptivo de interacción.

Donde:

σ_{Δ} : es el nombre del componente.

δ_{Δ} : es el método recomendado para la interacción del componente con el usuario.

λ : es el conjunto de propiedades de diseño del componente, cuyo esquema está representado por el **Esquema 3:**.

Esquema 3: Sean $\lambda (v_1, v_2, \dots, v_n)$ en conjunto de pares ordenados, donde cada v_n es un par ordenado utilizado para representar una propiedad de diseño y su valor.

Donde:

$v_n: \langle \sigma, \delta \rangle$

σ : nombre de la propiedad (color, tamaño de letra, etc.).

δ : valor para la propiedad de diseño.

Por lo tanto, puede expresarse como:

$\lambda (\langle \sigma, \delta \rangle, \langle \sigma, \delta \rangle, \dots, \langle \sigma, \delta \rangle)$

3.1.4.3 Definición formal de las reglas de adaptación

A continuación se propone una definición formal para reglas que tienen como entrada con un solo atributo de usuario, y otra para reglas que tienen como entrada un conjunto de atributos.

Regla con un solo atributo:

IF **aux_i** is **r_j** AND **accion** is **A₁** THEN **componente** is **v**

Regla con dos o más atributos:

IF **aux_i** is **r_j** ... **aux_{i+1}** is **r_{j+1}** AND **accion** is **A₁** THEN **componente** is **v**

Entradas:

aux_i = nombre del atributo de usuario

r_j = valor específico para el atributo de usuario

A₁ = tipo de acción

Salidas:

v = componente adaptivo de interacción (**Esquema 2:**)

Ejemplo: Si tenemos una sub-tarea para la cual la acción es del tipo *input* donde el usuario debe ingresar su nombre, dentro de una aplicación para la plataforma Android utilizando como dispositivo un teléfono inteligente (Smartphone), los datos que tenemos para la regla son:

Entradas: Como entradas se tiene que el atributo a expresar en la regla es *visión*, atributo para el cual el usuario tiene un valor de 20/40 (0.5 decimal), y el tipo de acción para el que se escribe la regla es *input*.

aux_i = visión

r_j = 0.5 (20/40)

A₁ = input

Salidas: Como salida tenemos un CADI correspondiente a la plataforma Android, cuyo nombre es *textbox* (inglés para cuadro de texto), como método de interacción con el componente se recomienda *interacción táctil* (en inglés touch) debido a que la debilidad visual del usuario no es tan grande para evitarle leer y escribir texto en un cuadro de texto, y las propiedades del componente son: aumentar en tamaño de la letra a 22px obtenido mediante una interpolación utilizando como parámetros de referencia el tamaño de letra indicado en la gráfica de Snellen (des Auges, Snellen, and Landolt 1874) para un nivel de visión 20/20, y el ultimo parámetro expresado en la regla es el color de la fuente en negro (#000000 en hexadecimal).

v = $\Upsilon(\sigma_{\Delta}, \delta_{\Delta}, \lambda)$ donde:

σ_{Δ} : textbox.

δ_{Δ} : touch.

λ : $\cup \{ \langle \sigma_0 = \text{Tamano_Letra}, \delta_0 = 22\text{px} \rangle, \langle \sigma_1 = \text{Color_Letra}, \delta_1 = \#000000 \rangle \}$

3.1.4.4 Implementación de las reglas de adaptación

La implementación de la reglas de adaptación se utiliza el formato XML, y se crear un archivo XML por cada una de las reglas de adaptación siguiendo la estructura propuesta en el punto anterior. El separar cada una de las reglas nos da la granularidad necesaria para que cualquiera de ellas pueda ser cambiada o modificada en producción ya sea por un sistema evolutivo, por preferencias del usuario o por múltiples motivos, sin afectar a las demás reglas.

La **Figura 12** muestra un ejemplo de una regla de adaptación expresada en formato XML.

```

<?xml version="1.0" encoding="UTF-8"?>
<rule>
  <attribute>vision</attribute>
  <comparison>
    <lessthan>
      <value>.9</value>
    </lessthan>
  </comparison>
  <action>input</action>
  <component>
    <name>EditText</name>
    <method>SpeechDictation</method>
    <properties>
      <property>
        <name>Font_Size</name>
        <value>24</value>
      </property>
      <property>
        <name>Font_Color</name>
        <value>#00000000</value>
      </property>
    </properties>
  </component>
</rule>

```

Figura 12: Ejemplo de una reglas de adaptación en XML

Con la finalidad de probar que la estructura propuesta para las reglas de adaptación es implementable y que las reglas pueden ser soportadas por estándar XML, se crearon cinco reglas de adaptación (descritas a continuación) para plataforma Android tomando en cuenta los atributos de usuario: visión y experiencia con el dispositivo.

Para las reglas de adaptación basadas en el atributo de usuario *visión* la adaptación que se proponen es adecuar el tamaño de letra según el nivel de visión del usuario. Los valores respectivos para el tamaño de fuente según el nivel de visión se obtuvieron realizando una interpolación de la prueba de visión de Snellen (des Auges, Snellen, and Landolt 1874) donde se propone el tamaño de fuente óptimo para un nivel de visión 20/20 (1.0 expresado en números decimales).

Tabla 6: Regla de adaptación para visión excelente

Regla 1: Visión excelente	
Entradas	
Atributo	vision

Valor del atributo	> 0.97
Acción	Input
Salida (componente CADI)	
Nombre	Edittext
Método de interacción	Touch
Propiedades	<font_size, 9> <text_color,#000000>

Tabla 7: Regla de adaptación para visión normal

Regla 2: Visión normal	
Entradas	
Atributo	vision
Valor del atributo	>= 0.9
Acción	Input
Salida (componente CADI)	
Nombre	Edittext
Método de interacción	Touch
Propiedades	<font_size, 17> <text_color,#000000>

Tabla 8: Regla de adaptación para visión nula

Regla 3: Visión nula	
Entradas	
Atributo	vision
Valor del atributo	< 0.3
Acción	Input
Salida (componente CADI)	
Nombre	Edittext
Método de interacción	speech dictation
Propiedades	

Para las reglas referentes a la experiencia con el dispositivo la adaptación que se propone en los siguientes ejemplos, es la de utilizar una etiqueta para describir el dato que el usuario debe introducir en cada capo de texto o utilizar la opción de la plataforma Android que es la de poner sobre el campo de texto vacío la sugerencia (*text hint*) del texto que se debe ingresar, de esta manera se ahorra espació necesario sobre todo en pantallas pequeñas.

Tabla 9: Regla de adaptación para experiencia con el dispositivo menor a dos meses

Regla 4: Experiencia con el dispositivo menor a 2 meses	
Entradas	
Atributo	device_experience
Valor del atributo	< 2
Acción	Input
Salida (componente CADI)	
Nombre	Edittext
Método de interacción	touch label
Propiedades	

Tabla 10: Regla de adaptación para experiencia con el dispositivo mayor a dos meses

Regla 5: Experiencia con el dispositivo mayor a 2 meses	
Entradas	
Atributo	device_experience
Valor del atributo	>= 2
Acción	Input
Salida (componente CADI)	
Nombre	Edittext
Método de interacción	touch nolabel
Propiedades	

3.1.5 Motor de adaptación

El motor de adaptación es la entidad encargada de realizar el proceso adecuación de los prototipos de interfaz a las características del usuario, para lo cual se toman los elementos descritos anteriormente como entrada para construir como salida una interfaz adaptada al usuario que está utilizando el sistema, y de esa manera permitirle llevar a cabo la tarea de manera usable.

El proceso de adaptación por el que es sometido un prototipo de interfaz para ser adecuado al usuario, consta de seleccionar el componente de interacción más adecuado a las características del usuario para cada una de las acciones de la tarea y construir una interfaz gráfica con el conjunto de componentes seleccionados.

Las decisiones del motor de adaptación son tomadas en base a las reglas de adaptación ya que es en ellas donde se expresa que componente adaptivo de interfaz debe presentarse al usuario, según el tipo de acción que se desee realizar, así como el valor que el usuario tenga para un atributo en específico.

Con la finalidad que el motor de adaptación sea genérico, es decir que pueda adaptarse a diferentes aplicaciones, sin la necesidad de reconstruirlo o modificarlo, proponemos que las reglas de adaptación sean independientes e intercambiables fácilmente, de esa manera el motor puede ser aplicado a diferentes aplicaciones solo creando o modificando reglas de adaptación.

La ejecución del motor de adaptación se realiza en tiempo de ejecución, lo que significa que cada vez que se desee adaptar una interfaz para un usuario, el motor procederá a leer el conjunto de reglas de adaptación para tomar decisiones y crear la interfaz óptima para el usuario. Esto permite que los cambios que se hayan realizado a las reglas, prototipos de interfaz o a los valores del modelo de usuario, sean aplicados la próxima ejecución del motor.

El motor de adaptación está compuesto por cinco secciones que trabajando en conjunto producen como elemento de salida la interfaz adaptada. La **Figura 13** muestra la interconexión entre los componentes, así como su relación con los elementos de entrada y salida.

- **Acceso a datos del modelo de usuario:** se encarga de obtener los valores de los atributos almacenados en el modelo integral de usuario.

- **Lector de prototipos:** es la sección encargada de leer los archivos de prototipos de interfaz para pasarlos al núcleo.
- **Lector de reglas:** es la encargada de leer los archivos con las reglas de adaptación y pasarlas al núcleo del motor para ser interpretadas.
- **Núcleo:** es la sección encargada de interpretar las reglas de adaptación, comparar el valor del atributo de usuario al que hace referencia la regla contra el valor del mismo atributo que el usuario en ejecución tiene, comparar el tipo de acción de cada uno de los componentes de interacción presentes en el prototipo de interfaz, y finalmente modificar o intercambiar el componente de interacción para convertirlo en un componente adaptivo de interacción.
- **Creador de interfaz:** es el encargado de aplicar lógica de organización (si se requiere) para posicionar cada uno de los componentes adaptivos de interacción creados por el núcleo, en una interfaz de usuario que será mostrada al usuario.

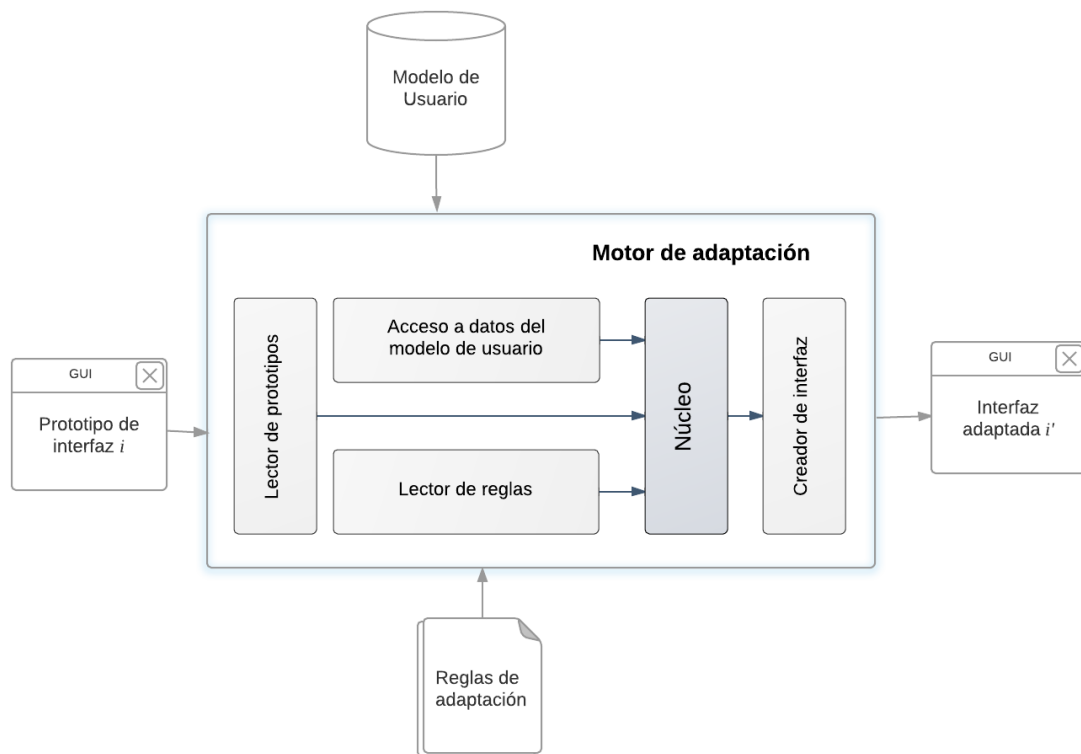


Figura 13: Elementos del motor de adaptación

3.2 Etapas de MEGAIA

Nuestra propuesta de método para la generación automática de interfaces de usuario adaptivas consta de una serie de pasos que se integran dentro del ciclo de vida del desarrollo de software, dichos pasos pueden ser agrupados en dos principales etapas, las cuales son descritas a continuación.

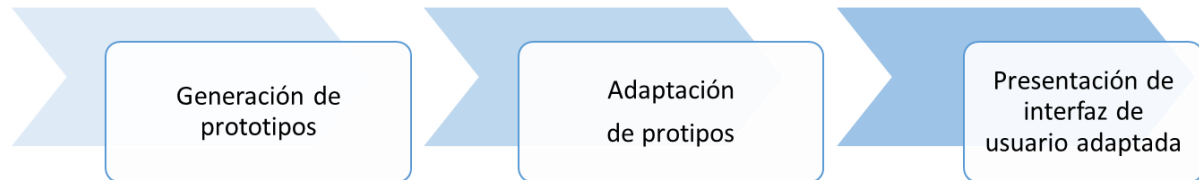


Figura 14: Etapas propuestas en MEGAIA

3.2.1 Etapa 1: Generación de prototipos

Esta etapa comprende desde el inicio del ciclo de desarrollo de software hasta el final de la fase de diseño. Y tiene como finalidad producir los prototipos de interfaz que permitan al usuario interactuar con el sistema de software.

El primer paso dentro de esta etapa consiste en aplicar la técnica de análisis de usuario mediante la cual se obtienen los valores para los atributos correspondientes al modelo integral de usuario.

El siguiente paso consiste en realizar un análisis de tareas para obtener la lista de tareas principales del sistema, y profundizar en cada una de ellas para identificar el conjunto de acciones que la conforman.

El último paso de esta etapa consiste en elaborar los prototipos de interfaz que contengan los componentes de interacción necesarios para permitir al usuario realizar cada una de las acciones de la tarea.

3.2.2 Etapa 2: Adaptación de la interfaz

Esta etapa comprende desde el inicio de la fase de desarrollo de software y continúa hasta la ejecución del sistema por parte del usuario, pues es en este momento cuando se lleva a cabo el proceso de adaptación de la interfaz de usuario.

El primer paso de esta segunda etapa consiste en crear las reglas de adaptación necesarias para guiar al motor de adaptación en el proceso de adecuación de la interfaz de usuario, este paso es de gran importancia para la calidad de la adaptación de la interfaz ya que la exactitud del motor de adaptación está ligada a la cantidad y calidad de las reglas de adaptación.

El segundo paso consiste en implementar dentro del sistema de software, el motor de adaptación, y plasmar en código las llamadas al motor de adaptación indicando el prototipo que se desea adaptar.

Por último, en tiempo de ejecución, el motor de adaptación realizará los ajustes necesarios al prototipo de interfaz, basándose en las reglas de adaptación, y en los atributos del usuario. De esta manera se crea y presenta al usuario una interfaz adaptada a sus características.

3.3 Implementación de un motor de adaptación

En esta sección se describe la implementación de un motor de adaptación genérico basado en un modelo de usuario, reglas de adaptación y prototipos de interfaz de usuario para aplicaciones Android siguiendo la propuesta presentada en la **sección 3.1.5**.

Para implementar el motor se optó por utilizar Android como plataforma móvil principalmente por contar con una gran variedad de dispositivos que varían tanto en prestaciones como en precio se espera que en un futuro se la plataforma móvil más utilizada y también debido a que las interfaces de usuario para este sistema operativo están representadas mediante archivos XML (Google Inc. 2013a) que son interpretados en tiempo de ejecución lo cual permite al motor de adaptación leer fácilmente la estructura de la interfaz gráfica contenida en dichos archivos.

Su principal funcionalidad consiste generar en tiempo de ejecución interfaces adaptadas a las características de cada uno de los usuarios del sistema, para lo cual toma como entrada: un prototipo o una interfaz de usuario, un conjunto de reglas de adaptación basadas en el esquema

propuesto en la **sección 3.1.4** de este documento, así como valores para los atributos de usuario enlistados en modelo integral de usuario descrito en la **sección 3.1.1**.

Durante la ejecución de la aplicación, es la misma aplicación la cual debe realizar la llamada para adaptar la interfaz y presentarla en pantalla, de esta manera es el programador quien tiene la decisión de que interfaces se presentan de manera común o cuales serán adaptadas antes de ser presentadas en pantalla.

Para la generación de la interfaz de usuario adaptada, el motor de adaptación no hace diferencia sobre el tamaño del dispositivo sobre el cual se está ejecutando, si no que sigue los principios de diseño presentados por Google (Google Inc. 2014a) para generar interfaces de usuario para pantallas de teléfonos inteligentes, y que podrían presentarse sin ningún problema en dispositivos de mayor tamaño como tabletas.

El lenguaje utilizado para desarrollar el motor fue Java, pues es el lenguaje oficial soportado por Android para el desarrollo de aplicaciones, como entorno de desarrollo se utilizó Eclipse versión 4.2.1 (Eclipse 2010) con el módulo ADT (Android Developer Tools, ingles para herramientas para desarrolladores de Android) versión 22.0.1 (Google Inc. 2014b), utilizando un computadora con Windows 8, procesador Intel i7 2.20 GHz con 6 GB de RAM.

El motor de adaptación está basado en tres principales módulos; reglas de adaptación, modelo de usuario y mecanismo de adaptación, implementados en el esquema de paquetes de Java, lo que permite al motor de adaptación integrarse de manera modular a cualquier aplicación Android implementada en Java.

IV

EXPERIMENTOS REALIZADOS

*“No amount of experimentation can ever prove me right; a single
experiment can prove me wrong.”*

- Albert Einstein -

4 CASOS DE ESTUDIO

En esta sección son descritos los casos de estudio que se realizaron con el objetivo de comprobar o desaprobar las hipótesis propuestas al inicio de este documento, así como el modelo que en este se propone.

Los casos de estudio se dividen en dos etapas; caso de estudio sobre los prototipos de interfaz y caso de estudio sobre el motor de adaptación. A continuación se describen más a detalle los casos de estudio así como el resultado obtenido de cada uno de ellos.

4.1 Caso de estudio con prototipos de interfaz

Este caso de estudio tiene como objetivo comprobar la hipótesis **H1.2** y la propuesta de prototipos de interfaz presentada en la sección 3.1.3 Prototipos de interfaz. A continuación se describen los objetivos del caso de estudio, la metodología utilizada, resultados obtenidos y conclusiones.

4.1.1 Objetivos del caso de estudio

OEP1: Generar automáticamente prototipos de interfaz de usuario basados en un conjunto de acciones derivados de los flujos de los casos de uso.

OEP2: Evaluar el grado de completud de los prototipos de interfaz generados automáticamente, comparándolos con versiones generados por expertos en el área.

4.1.2 Implementación de un generador automático de prototipos de interfaz

En este caso de estudio se desarrolló la aplicación generador automático de prototipos de Interfaz Android (**GAPIa**), para escritorio utilizando el lenguaje C# .net versión 4.0.

El objetivo de GAPI es la de generar automáticamente prototipos de interfaz para la plataforma Android versión 2.0 en adelante, tomando como entrada el tipo de acción de la tarea y el tipo de dato de la acción.

GAPIa utiliza los postulados de este documento así como en un catálogo de componentes de interacción definidos por el **Esquema 1**: como reglas de producción para generar un archivo XML que contiene la interfaz de usuario

La salida es en formato XML debido a que este es el formato utilizado por Android para los archivos de interfaz de usuario, este archivo producido por GAPIa está listo para ser leído por un editor de interfaces de Android o para ser presentado al usuario.

En la **Figura 15** puede observarse la interfaz de usuario de GAPIa para capturar la lista de acciones, el tipo de acción, el tipo de dato y algunas propiedades gráficas opcionales.

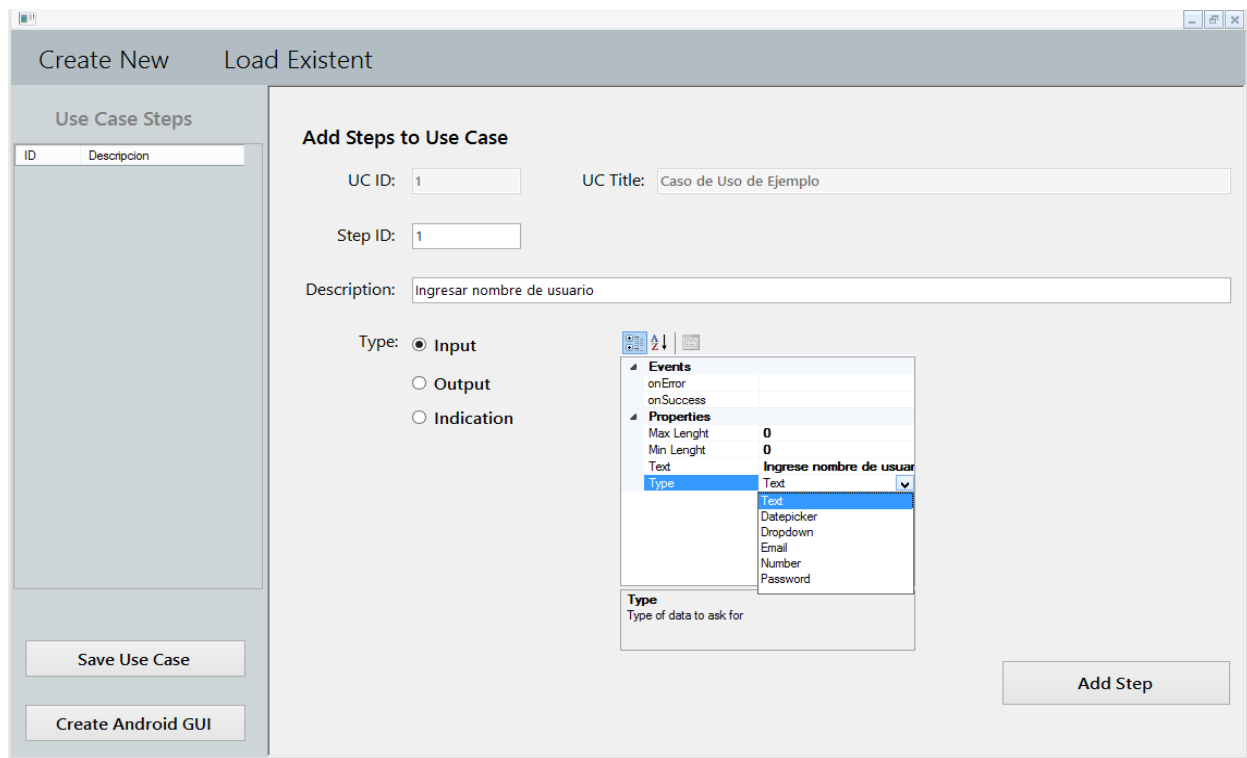


Figura 15: Interfaz de usuario GAPIa

4.1.2.1 Ejemplos de componentes de interacción

A continuación se muestra un ejemplo de componente de interacción para cada tipo de acción.

Tabla 11: Ejemplo de componente de interacción de Entrada

Nombre	EditText
Descripción	Permite al usuario ingresar texto al sistema
Tipo de acción	Entrada

Tipo de dato	String
Sistema operativo	Android

Tabla 12: Ejemplo de componente de interacción de Salida

Nombre	TextView
Descripción	Permite al sistema desplegar texto en pantalla
Tipo de acción	Salida
Tipo de dato	String
Sistema operativo	Android

Tabla 13: Ejemplo de componente de interacción de Indicación

Nombre	Button
Descripción	Representa un botón presionable en pantalla
Tipo de acción	Indicación
Tipo de dato	(No aplicable)
Sistema operativo	Android

4.1.3 Realización del caso de estudio

La realización del caso de estudio consta de cinco principales actividades:

1. Definición de cinco tareas base
2. Selección de alumnos para diseñar interfaces de usuario.
3. Diseño de interfaces de usuario para las tareas base.
4. Generación de prototipos de interfaz para las tareas base utilizando GAPIa.
5. Evaluación de prototipos generados utilizando GAPIa.

Definición de las tareas base: se seleccionó un conjunto de cinco tareas, a las que se les conoce en este documento como tareas base (**tab**), dichas tareas están pensadas principalmente para que el usuario introduzca datos. La complejidad de las tareas incrementaba de una a otra, donde la

menor consistía en ingresar solamente un nombre de usuario y una contraseña (tarea de iniciar sesión) y la que tenía mayor complejidad (requería un mayor número de datos) fue dar de alta un nuevo empleado, la cual implicaba 12 datos del empleado, dentro de los cuales se requería ingresar fechas, y seleccionar el tipo de empleado de un catálogo.

El documento con el detalle de las tareas puede encontrarse en el apéndice A.

Selección de los diseñadores: Se seleccionaron alumnos de dos grupos durante el curso de Diseño de Interacciones, de la carrera Ingeniería en Computación impartida en la Facultad de Ciencias Químicas e Ingeniería en la Universidad Autónoma de Baja California, periodo 2013-2.

Características de los alumnos.

- Total de 27 Alumnos.
- Cursaban semestres entre el 5to y el 6to de la carrera.
- Tenían conocimiento del tema de usabilidad.
- Con menos de 6 meses de experiencia desarrollando aplicaciones, todas del tipo inter-clase.

Diseño de las interfaces normales: Para el diseño de las interfaces de usuario se separaron los participantes en dos grupos; grupo A y grupo B. Los pertenecientes al grupo A fueron organizados en equipos de dos personas. La **Tabla 14** muestra la distribución de 26 participantes del caso de estudio. El participante no incluido en los grupos fue el seleccionado para utilizar el GAPIa para generar los prototipos de interfaz.

Tabla 14: Distribución de participantes para diseño de interfaces.

Grupo	Cantidad de equipos	Integrantes por equipo	Total de integrantes
A	9	2	18
B	8	1	8

Se les entregó un documento con la lista de posibles componentes de interfaz para la plataforma Android y se les dio un día para estudiarlo, al día siguiente se les entrenó en utilizar la herramienta que Google ofrece para el desarrollo de aplicaciones Android. La herramienta cuenta con un editor gráfico del tipo WYSIWYG.

Posteriormente se les entregó el documento con la definición de las tareas base y se les pidió a los 17 equipos generar las 5 interfaces de usuario correspondientes a las 5 tareas base, para lo cual se les dio un tiempo de 3 días como fecha límite para entregar los XML correspondientes a las interfaces de usuario.

Generación de prototipos automáticos: Al participante seleccionado para la generación de prototipos de interfaz mediante el GAPIa se le entrenó en el uso de GAPIa. Posteriormente se le entregó el documento con la definición de las tareas base y se le pidió generar los prototipos de interfaz para cada una de las tareas base.

Evaluación de prototipos GAPIa: Con la finalidad de evaluar la completud de los prototipos de interfaz generados por GAPIa, se le pidió a un equipo de 2 expertos en diseño de interfaces de usuario, leer el documento con la descripción de las tareas base y posteriormente diseñar las interfaces para las 5 tareas.

4.1.4 Resultados del caso de estudio

Tomando como base el diseño realizado por los expertos se evaluaron los prototipos de interfaz comparando la cantidad de componentes y el tipo de cada uno de ellos, los resultados de dicha evaluación se presentan a continuación.

La **Figura 16**, **Figura 17**, **b**, **Figura 19** y **Figura 20** son capturas de pantallas de los prototipos resultantes para cada una de las tareas, tomadas ejecutando los prototipos de interfaz sobre un teléfono inteligente Samsung Galaxy S3 con una pantalla de 4.8” con resolución de 1280x720.

Pruebas de Interfaz

Usuario

Password

Aceptar Cancelar

Figura 16: Prototipo de interfaz para la tarea 1

Pruebas de Interfaz

Número de cuenta

Banco

Numero de Cheque

Cuenta Contable
1000-34 ▼

Cancelar Guardar

Figura 17: Prototipo de interfaz para la tarea 2

Pruebas de Interfaz

Razón del cliente

RFC del cliente

Dirección del cliente

Concepto de la factura

Cantidad

Importe

Figura 18: Prototipo de interfaz para la tarea 3

Pruebas de Interfaz

Tipo de cliente
Escuela ▼

Nombre del cliente:

Nombre del Contacto:

Numero del Contacto:

Correo:

RFC:

Fecha de Vencimiento:

+	+	+
Feb	05	2014
-	-	-

Día de contrarecibo.

Lunes ▼

Hora del contrarrecibo:

+	+	
2	47	AM
-	-	

Cancelar Guardar Datos

Figura 19: Prototipo de interfaz para la tarea 4

Pruebas de Interfaz

Tipo de empleado
Administracion ▼

Nombre del empleado

Apellidos del empleado

Fecha de ingreso

+	+	+
Feb	05	2014
-	-	-

Fecha de nacimiento

+	+	+
Feb	05	2014
-	-	-

RFC

Teléfono

Dirección

Número de Licencia

Fecha de vencimiento de Licencia

+	+	+
Feb	05	2014
-	-	-

Cancelar Guardar

Figura 20: Prototipo de interfaz para la tarea 5

La **Tabla 15** representa los resultados de evaluar la cantidad de componentes de interacción presentes en cada uno de los prototipos generados por el GAPIa, respecto a los prototipos diseñados por el equipo de expertos.

Tabla 15: Resultados de evaluación de la completud de los prototipos

Tarea	Cantidad de componentes	Concordancia en cantidad de componentes
T1	4	100.00%
T2	6	100.00%
T3	10	100.00%
T4	11	100.00%
T5	12	100.00%
		100.00%

Las siguientes tablas muestran los resultados de comparar el tipo de cada uno de los componentes de interacción de los prototipos generados por el GAPIa respecto al tipo de componentes que los expertos seleccionaron para cada acción de la tarea.

Tabla 16: Resultados de evaluación de prototipos tarea 1

Tarea	Acción	Componente	Concuenda Tipo
T1	T1A1	EditText	Si
T1	T1A2	EditText[Password]	Si
T1	T1A3	Button	Si
T1	T1A3	Button	Si
			100.00%

Tabla 17: Resultados de evaluación de prototipos tarea 2

Tarea	Acción	Componente	Concuenda Tipo
T2	T2A1	EditText	Si
T2	T2A2	EditText	Si

T2	T2A3	EditText	Si
T2	T2A4	Spinner	Si
T2	T2A5	Button	Si
T2	T2A5	Button	Si
			100.00%

Tabla 18: Resultados de evaluación de prototipos tarea 3

Tarea	Acción	Componente	Concuerda Tipo
T3	T3A1	EditText	Si
T3	T3A2	EditText	Si
T3	T3A3	EditText	Si
T3	T3A4	EditText[Number]	Si
T3	T3A5	EditText[Decimal]	Si
T3	T3A6	Button	Si
T3	T3A6	Button	Si
			100.00%

Tabla 19: Resultados de evaluación de prototipos tarea 4

Tarea	Acción	Componente	Concuerda Tipo
T4	T4A1	Spinner	Si
T4	T4A2	EditText	Si
T4	T4A3	EditText	Si
T4	T4A4	EditText	Si
T4	T4A5	EditText[E-mail]	Si
T4	T4A6	EditText	Si
T4	T4A7	DatePicker	Si
T4	T4A8	Spinner	Si
T4	T4A9	TimerPicker	No
T4	T4A10	Button	Si
T4	T4A10	Button	Si
			90.91%

Tabla 20: Resultados de evaluación de prototipos tarea 5

Tarea	Acción	Componente	Concuerda Tipo
T5	T5A1	Spinner	Si
T5	T5A2	EditText	Si
T5	T5A3	EditText	Si
T5	T5A4	DatePicker	Si
T5	T5A5	DatePicker	Si
T5	T5A6	EditText	Si
T5	T5A7	EditText[Phone]	No
T5	T5A8	EditText[Postal Addres]	No
T5	T5A9	EditText	Si
T5	T5A10	DatePicker	Si
T5	T5A11	Button	Si
T5	<u>T5A11</u>	Button	Si
			83.33%

Los resultados de las tablas, podemos observar que surge una posible relación entre la cantidad de acciones de la tarea y la disminución de la efectividad del generador automático de prototipos de interfaz. Para disprobar esta posible relación se decidió hacer una prueba más, donde se aumentó drásticamente el número de acciones de la tarea. El resultado se muestra en la tabla de a continuación.

Tabla 21: Resultados de la evaluación de prototipos tarea 6

Tarea	Acción	Componente	Concuerda Tipo
T6	T6A1	EditText	Si
T6	T6A2	EditText	Si
T6	T6A3	Spinner	Si
T6	T6A4	Spinner	Si
T6	T6A5	EditText	Si
T6	T6A6	Spinner	Si
T6	T6A7	EditText	Si

T6	T6A8	EditText[Number]	Si
T6	T6A9	EditText[Number]	Si
T6	T6A10	EditText	Si
T6	T6A11	DatePicker	Si
T6	T6A12	DatePicker	Si
T6	T6A13	Spinner	Si
T6	T6A14	Spinner	Si
T6	T6A15	Spinner	Si
T6	T6A16	TimePicker	Si
T6	T6A17	DatePicker	Si
T6	T6A18	EditText	Si
T6	T6A19	EditText	Si
T6	T6A20	EditText[Phone]	Si
T6	T6A21	EditText[Phone]	Si
T6	T6A22	EditText[E-mail]	Si
T6	T6A23	EditText[E-mail]	Si
T6	T6A24	Spinner	Si
			100%

4.1.5 Conclusiones del caso de uso

De este caso de estudio podemos concluir que sí es posible generar automáticamente los prototipos de interfaz tomando como base el flujo de los casos de uso o utilizando la descripción de la tarea, y lograr un alto grado de efectividad y sobre todo de completud con lo cual comprobamos la hipótesis **H1.2**.

Comprobamos también que la aparente relación entre la cantidad de acciones de la tarea y la disminución en la efectividad del GAPIa no era tal, y que el defecto se debía a que en las tareas con más acciones se utilizaron componentes que no estaban considerados dentro del catálogo de componentes de interacción del generador GAPIa, por lo que bastó con renovar el catálogo de componentes del GAPIa para que fueran considerados dentro del prototipo generado para la tarea 6.

Debido a que la selección de los componentes que integran el prototipo son elegidos de un catálogo para ese fin creado, el prototipo obtenido ayuda a los diseñadores inexpertos a crear

prototipos basados en los principios de diseño de la plataforma, que en caso de los experimentos aquí realizados fue el sistema operativo Android.

Otro aspecto que cabe destacar sobre la generación automática de prototipos es la disminución en el tiempo que toma, en comparación con la creación de prototipos de manera tradicional, constituye una disminución considerable, sobre todo cuando los diseñadores son novatos para la plataforma en la que se desea implementar los prototipos.

4.2 Caso de estudio sobre el motor de adaptación

En esta sección se describe el caso de uso llevado a cabo para probar las hipótesis **H1** y **H1.1**, así como el funcionamiento del motor de adaptación descrito en las secciones y **3.3** de este documento.

4.2.1 Objetivos

OEM1: Probar la capacidad del motor de adaptación implementado en la sección **3.1.5**, para crear interfaces adaptadas dinámicamente a diversos usuarios.

OEM2: Medir el aumento en la eficiencia y satisfacción del usuario al utilizar interfaces adaptadas a sus características individuales, en contraste con interfaces generadas de manera tradicional.

4.2.2 Desarrollo de una aplicación mediante el método MEGAIA

Para probar el motor de adaptación se optó por crear una aplicación Android a la que se le dio el nombre de Pruebas de interfaz Android (**PIA**). La finalidad de esta aplicación fue la de probar el método propuesto en este trabajo de tesis, para lo cual PIA presenta ante usuarios reales dos tipos de interfaces diferentes para cada una de las tareas base; interfaces no adaptivas e interfaces adaptadas por el motor de adaptación propuesto en este documento.

Dentro de PIA se incluyen todos los aspectos que se proponen para el método MEGAIA; los prototipos de interfaz, las reglas de adaptación, el modelo de usuario, así como el motor de adaptación.

Interfaces normales: De las interfaces de usuario desarrolladas por los alumnos para las cinco tareas base, se seleccionó la mejor de las interfaces para cada tarea. Las cinco interfaces se incluyeron dentro de la aplicación para ser mostradas al usuario. A dichas interfaces se les dio el

nombre de interfaces normales. Dichas interfaces representan las interfaces que generalmente se incluyen en una aplicación, que son diseñadas durante el proceso de desarrollo del software y son presentadas al usuario sin modificaciones.

Prototipos de interfaz: Dentro de la aplicación también se incluyen los prototipos de interfaz generados por el GAPIa para cada una de las tareas base. Estos prototipos serán adaptados por el motor de adaptación antes de presentarse al usuario.

Reglas de adaptación: Se incluyen las cinco reglas de adaptación descritas en el punto 3.4.4 dentro la PIA, dichas reglas son sobre las que se basa el motor de adaptación para tomar las decisiones acerca de los cambios o ajustes que es necesario realizar a los prototipos de interfaz.

Modelo de usuario: También se incluye en PIA el modelo integral de usuario descrito en la sección 3.1 de este documento. Debido a que se incluyen solo reglas de adaptación para atributos de usuario visión y experiencia con el dispositivo, no se consideró necesario realizar un análisis a profundidad de cada uno de los usuarios de PIA para obtener los valores para cada atributo considerando en el modelo de usuario, y se optó por obtener los valores para los dos atributos considerados en las reglas, así como algunos datos personales del usuario.

Para obtener los valores de los atributos del usuario se realizó un cuestionario a cada usuario, dicho cuestionario se realiza utilizando la propia aplicación PIA de tal manera que los datos obtenidos del usuario van directamente a alimentar el modelo de usuario. El cuestionario consta de las siguientes preguntas:

Tabla 22: Guía de preguntas para evaluar el perfil de usuario

Pregunta	Posibles respuesta
Edad	(valor numérico)
Nivel de visión*	(valor numérico de 0 al 20)
Utiliza lentes	- Si - No
Capacidad de ver colores	- Colores - Solo blanco y negro - No azul ni amarillo - No rojo ni verde
Ha utilizado anteriormente un dispositivo Android	- Si - No
Cuanto tiempo <i>(solo si la respuesta a la</i>	- Menor a dos meses

<i>pregunta anterior es positiva)</i>	- Igual o mayor a dos meses
---------------------------------------	-----------------------------

Para obtener el nivel de visión se le realiza al usuario la versión de corta distancia de la prueba de Snellen (des Auges, Snellen, and Landolt 1874) en la cual se presenta al usuario una gráfica de Snellen [véase apéndice B] a una distancia de 40cm del rostro medidos desde la parte externa del ojo a la superficie de la hoja. Con la finalidad de evitar la mayor cantidad de ruido posible en los datos obtenidos, se instruyó a los alumnos para la realización de dicha prueba de visión y se les entregó a cada uno de ellos la gráfica de Snellen, así como un cordón de 40 cm de longitud con la finalidad de evitar los más posible ruidos en el valor del nivel de visión.

Motor de adaptación: PIA cuenta con el motor de adaptación propuesto en el punto 3.5 de este documento, donde su finalidad es la de adaptar los prototipos de interfaz a las características del usuario en tiempo de ejecución de la aplicación. El funcionamiento del motor de adaptación se divide en cinco pasos los cuales son enlistados y descritos a continuación.

- Se lee el prototipo de interfaz que se desea adaptar, el prototipo está contenido en un archivo XML, que puede ser generado utilizando el GAPIa o por cualquier editor de interfaces de Android. En el caso de PIA, los archivos son los correspondientes a las cinco tareas base, los cuales fueron generados utilizando la propuesta GAPIa.
- Se leen los archivos XML que contienen las reglas de adaptación, para el caso de la aplicación pruebas de interfaz Android, son las cinco reglas descritas en el punto 3.4.4, las cuales se enfocan solo en los atributos *visión* y *experiencia en el dispositivo*.
- Se compara el atributo de usuario al que hace referencia cada una de las reglas de adaptación, con que usuario tiene para el atributo en cuestión. En caso de concordancia en los valores se almacena la regla en una lista de reglas aplicables para el usuario.
- Se aplican al prototipo de interfaz las adaptaciones contenidas en la lista de reglas aplicables para el usuario.

- Una vez que se han aplicado las adaptaciones, se presenta en pantalla la interfaz de usuario adaptada a las características del usuario.

4.2.3 Realización del caso de estudio

Posterior a desarrollo de la aplicación PIA para la plataforma Android, la siguiente fase del caso de estudio consistió en presentar la aplicación con usuarios reales. A continuación se describe el proceso llevado a cabo.

Identificación del usuario: El primer paso fue identificar al usuario antes de que este comenzara a realizar las pruebas para posterior análisis de los datos. Para esto la pantalla inicial de la aplicación es la de registrar el nombre del usuario, o seleccionarlo de una lista en caso de que este haya sido registrado anteriormente.

Perfil de usuario: Se aplica al usuario el examen rápido para obtener el nivel de visión, así como el resto de las preguntas del cuestionario descrito en la **Tabla 22**, de esta forma se obtienen los valores para los atributos de usuario que se van a utilizar para las adaptaciones.

Interfaces no adaptivas: Se le pide al usuario que realice cada una de las cinco tareas base utilizando las interfaces normales, destacando desde un inicio que procederá a ejecutar una aplicación con cinco interfaces normales. En la **Figura 21**, **Figura 22**, **Figura 23**, **Figura 24** y **Figura 25** se muestran capturas de pantalla para cada una de las interfaces no adaptivas presentadas a usuarios reales durante el experimento, dichas imágenes fueron ejecutadas sobre un dispositivo Samsung Galaxy S3 con una pantalla de 4.8” con resolución de 1280x720.

Pruebas de Interfaz

Nombre de usuario

Ingresar contraseña

Figura 21: Interfaz normal para tarea 1

Pruebas de Interfaz

Nombre del Banco:

Numero de Cuenta:

Numero de Cuenta Contable:

Numero de Cheque:

Figura 22: Interfaz normal para tarea 2

Pruebas de Interfaz

Razon Social

RFC

Direccion

Concepto

Cantidad

Importe

Guardar

Figura 23: Interfaz normal para tarea 3

Pruebas de Interfaz

Tipo de Cliente

Nombre del Cliente

Nombre de Contacto

Fecha de Vencimiento

Correo de Contacto

No. de Contacto

Dia de contra-recibo

Hora de Contra-Recibo

Figura 24: Interfaz normal para tarea 4

Pruebas de Interfaz

Tipo:

Nombre

Apellidos

RFC:

Número de teléfono:

Dirección:

Número de licencia:

Fecha Ingreso:

Fecha Nacimiento:

Vencimiento Licencia:

Figura 25: Interfaz normal para figura 5

Interfaces adaptativas: El siguiente paso durante la ejecución del caso de estudio consiste en pedir al usuario que realice las cinco tareas base utilizando esta vez las interfaces adaptadas en tiempo real a sus valores de visión y experiencia del usuario. Se le hace notar al usuario que procederá a utilizar una aplicación distinta a la anterior donde las interfaces serán adaptadas a sus características. A continuación se muestran ejemplos de interfaces de usuario resultantes para ciertos valores *visión (Vs)* y *experiencia con el dispositivo (Ed)*.

Interfaces adaptadas para
Vs: 20/25
Ed: Menor a dos meses

Pruebas de Interfaz

Número de cuenta

Banco

Numero de Cheque

Cuenta Contable
1000-34 ▼

Cancelar Guardar

Interfaces adaptadas para
Vs: 20/25
Ed: Mayor a dos meses

Pruebas de Interfaz

Número de cuenta

Banco

Numero de Cheque

Cuenta Contable
1000-34 ▼

Cancelar Guardar

Interfaces adaptadas para
Vs: 20/60
Ed: Mayor a dos meses

Pruebas de Interfaz

Número de cuenta

Banco

Numero de Cheque

Cuenta Contable
1000-34 ▼

Cancelar Guardar

Figura 26: Interfaces adaptadas - tarea 3



Figura 27: Interfaces adaptadas - tarea 4

Mediciones: Para obtener los datos se utilizaron dos técnicas; mediante un registro de eventos y utilizando un cuestionario post-ejecución.

- **Registro de eventos:** La propia aplicación PIA realizó un registro de eventos sobre las actividades de los usuarios, de forma tal que nos proporcionó los datos para identificar el número de intentos que le tomó a un usuario realizar satisfactoriamente cada una de las tareas, el tiempo transcurrido desde el inicio de la tarea hasta su terminación y el número de ocasiones que el usuario abandonó la tarea antes de terminarla.

Con estas mediciones se pretende obtener valores que soporten el atributo de usabilidad *eficiencia*, que representan directamente el efecto en usabilidad que tiene el motor de adaptación dentro de un sistema de software.

- **Cuestionario post-ejecución:** Para conocer la impresión del usuario con cada tipo de interfaz, se le pidió contestar dos cuestionarios una vez terminada la ejecución. Antes de contestar cada uno de los cuestionarios se le pidió al usuario centrarse solo en su experiencia con el tipo de interfaz a evaluar.

Con dichos cuestionarios se pretende obtener valores para el atributo de usabilidad *satisfacción*, de tal forma que se puedan evaluar dos atributos de usabilidad para el motor de adaptación.

Los cuestionarios constan de siete afirmaciones cada uno, los cuestionarios son similares, pero enfocados en las distintas interfaces (normales y adaptivas). Los cuestionarios se hicieron utilizando la plataforma *Google Forms* (Google Inc. 2013b). Para cada afirmación el usuario debía seleccionar un valor del 0 al 4, donde el cero significa *totalmente en desacuerdo* y el 4 *totalmente de acuerdo*. La tabla que se presenta a continuación contiene las afirmaciones que se le presentaron al usuario.

Tabla 23: Cuestionario de satisfacción

Código	Afirmación
AS1	Fue fácil aprender a utilizar la aplicación
AS2	Fue fácil identificar lo que se debía hacer en cada una de las tareas
AS3	Fue fácil utilizar el sistema para realizar la tarea
AS4	Fue fácil lograr que el sistema hiciera lo que debería hacer
AS5	Fue fácil leer el texto en pantalla
AS6	Fue agradable el uso del sistema
AS7	Creo que utilizar el sistema aumentaría mi productividad

4.2.4 Resultados del caso de estudio

Resultados de evaluación en uso: A continuación se presentan una serie de tablas correspondientes al análisis de resultados, mediante esta parte del caso de estudio se midió la eficiencia de cada uno de los usuarios respecto a cada tarea, comparando la eficiencia obtenida de las interfaces normales contra la eficiencia obtenida utilizando interfaces adaptadas a las características de los usuarios.

Las siguientes tablas muestran las características de los usuarios con quienes se probaron las interfaces. Los usuarios fueron separados en dos grupos:

GU1: Usuarios sin conocimiento en el dominio del problema. Son los usuarios que no han utilizado un sistema empresarial ERP.

Tabla 24: Características de usuarios del grupo GU1

Usuario	Edad	Visión	Lentes	Visión de color	Experiencia previa	Tiempo de uso
UP1	25	20	Si	Colores	Si	mayor a 2 meses
UP2	43	44	No	Colores	Si	mayor a 2 meses
UP3	20	18	No	Colores	Si	mayor a 2 meses
UP4	20	18	No	Colores	Si	mayor a 2 meses
UP5	21	22	No	Colores	No	menor a 2 meses
UP7	45	20	No	Colores	No	menor a 2 meses
UP8	26	80	Si	Colores	Si	mayor a 2 meses
UP9	29	20	Si	Colores	Si	menor a 2 meses
UP10	21	15	Si	Colores	Si	mayor a 2 meses
UP12	22	18	No	Colores	Si	menor a 2 meses
UP14	51	32	Si	Colores	No	menor a 2 meses
UP15	23	36	No	Colores	Si	mayor a 2 meses
UP16	37	40	Si	Colores	Si	mayor a 2 meses
UP17	18	18	No	Colores	Si	mayor a 2 meses
UP18	21	20	No	Colores	Si	mayor a 2 meses
UP19	26	20	No	Colores	No	menor a 2 meses
UP20	24	20	No	Colores	Si	mayor a 2 meses
UP21	25	20	No	Colores	Si	mayor a 2 meses
UP22	32	32	No	Colores	Si	mayor a 2 meses
UP23	20	17	Si	Colores	Si	menor a 2 meses
UP24	18	20	No	No Azul/Amarillo	Si	mayor a 2 meses
UP25	30	20	No	Colores	No	menor a 2 meses
UP26	30	160	No	Colores	Si	mayor a 2 meses
UP27	53	36	No	Colores	No	menor a 2 meses
UP28	23	48	Si	Blanco/Negro	No	menor a 2 meses
UP29	31	36	Si	Colores	Si	mayor a 2 meses
UP30	26	30	Si	Colores	No	mayor a 2 meses
UP31	20	31	No	Colores	Si	mayor a 2 meses
UP32	24	28	No	Colores	Si	mayor a 2 meses
UP33	32	27	Si	Colores	Si	mayor a 2 meses
UP35	57	80	Si	Colores	No	menor a 2 meses

GU2: Usuarios con conocimiento en el dominio del problema. Aquellos usuarios que tienen experiencia utilizando sistemas ERP.

Tabla 25: Características de usuarios del grupo GU2

Usuario	Edad	Visión	Lentes	Visión de color	Experiencia previa	Tiempo de uso
UP36	32	24	No	Colores	No	menor a 2 meses
UP37	24	32	No	Colores	Si	mayor a 2 meses
UP38	35	32	No	Colores	Si	mayor a 2 meses
UP39	33	24	No	Colores	Si	mayor a 2 meses
UP40	24	24	Si	Colores	No	menor a 2 meses
UP41	34	80	Si	Colores	Si	mayor a 2 meses
UP42	36	32	No	Colores	No	menor a 2 meses
UP43	29	24	No	Colores	No	menor a 2 meses
UP44	37	32	No	Colores	No	menor a 2 meses
UP45	41	40	No	Colores	No	menor a 2 meses
UP46	35	56	Si	Colores	Si	menor a 2 meses
UP47	45	32	No	Colores	No	menor a 2 meses
UP48	44	56	Si	Colores	No	menor a 2 meses
UP49	52	56	Si	Colores	No	menor a 2 meses
UP50	34	24	No	Colores	Si	mayor a 2 meses

Como se describe anteriormente, durante la evaluación en uso se le presenta a cada usuario dos interfaces diferentes para cada una de las tareas base. Dentro de las siguientes tablas de resultados, la columna *A1* representa a los resultados obtenidos para interfaz adaptiva de la tarea 1, y la columna *N1* representa los de la interfaz normal (no adaptiva) de la tarea 1. Los resultados son presentados en segundos correspondientes al tiempo que le tomó al usuario completar la tarea. La columna *aumento en eficiencia* representa la diferencia en segundos entre *N1* y *A1*.

Tabla 26: Resultados de interfaces para la tarea 1 con grupo de usuarios GU1

Usuario	A1	N1	Aumento eficiencia	Porcentaje
UP1	00:09	00:15	00:06	40.00%
UP2	00:39	01:08	00:29	42.65%
UP3	00:14	00:23	00:09	39.13%
UP4	00:43	01:20	00:37	46.25%
UP5	00:13	00:33	00:20	60.61%
UP6	00:30	00:41	00:11	26.83%
UP7	00:15	00:22	00:07	31.82%
UP8	00:13	00:34	00:21	61.76%
UP9	00:17	00:47	00:30	63.83%

UP10	00:23	00:49	00:26	53.06%
UP12	00:16	00:46	00:30	65.22%
UP13	00:48	00:55	00:07	12.73%
UP14	00:25	00:38	00:13	34.21%
UP15	00:18	00:30	00:12	40.00%
UP16	00:12	00:25	00:13	52.00%
UP17	00:08	00:09	00:01	11.11%
UP18	00:07	00:10	00:03	30.00%
UP19	00:16	00:26	00:10	38.46%
UP20	00:18	00:46	00:28	60.87%
UP21	00:22	00:27	00:05	18.52%
UP22	00:08	00:13	00:05	38.46%
UP23	00:13	00:23	00:10	43.48%
UP24	00:09	00:19	00:10	52.63%
UP25	00:37	00:43	00:06	13.95%
UP26	00:08	00:19	00:11	57.89%
UP27	00:16	00:23	00:07	30.43%
UP28	00:11	00:19	00:08	42.11%
UP29	00:13	00:15	00:02	13.33%
UP30	00:02	00:03	00:01	33.33%
UP31	00:42	01:02	00:20	32.26%
				39.56%

Tabla 27: Resultados de interfaces para la tarea 1 con grupo de usuarios GU2

Usuario	A1	N1	Aumento eficiencia	Porcentaje
UP32	00:19	00:32	00:13	40.63%
UP33	00:15	00:28	00:13	46.43%
UP34	00:09	00:25	00:16	64.00%
UP35	00:10	00:23	00:13	56.52%
UP36	00:05	00:24	00:19	79.17%
UP37	00:08	00:19	00:11	57.89%
UP38	00:17	00:21	00:04	19.05%
UP39	00:15	00:27	00:12	44.44%
UP40	00:13	00:28	00:15	53.57%
UP41	00:19	00:31	00:12	38.71%
UP42	00:19	00:30	00:11	36.67%
UP43	00:12	00:33	00:21	63.64%
UP44	00:22	00:37	00:15	40.54%
UP45	00:20	00:35	00:15	42.86%

UP46	00:11	00:28	00:17	60.71%
				49.65%

Tabla 28: Resultados de interfaces para la tarea 2 con grupo de usuarios GU1

Usuario	A2	N2	Aumento eficiencia	Porcentaje
UP1	00:09	00:15	00:06	40.00%
UP2	00:39	01:08	00:29	42.65%
UP3	00:14	00:23	00:09	39.13%
UP4	00:43	01:20	00:37	46.25%
UP5	00:13	00:33	00:20	60.61%
UP6	00:30	00:41	00:11	26.83%
UP7	00:15	00:22	00:07	31.82%
UP8	00:13	00:34	00:21	61.76%
UP9	00:17	00:47	00:30	63.83%
UP10	00:23	00:49	00:26	53.06%
UP12	00:16	00:46	00:30	65.22%
UP13	00:48	00:55	00:07	12.73%
UP14	00:25	00:38	00:13	34.21%
UP15	00:18	00:30	00:12	40.00%
UP16	00:12	00:25	00:13	52.00%
UP17	00:08	00:09	00:01	11.11%
UP18	00:07	00:10	00:03	30.00%
UP19	00:16	00:26	00:10	38.46%
UP20	00:18	00:46	00:28	60.87%
UP21	00:22	00:27	00:05	18.52%
UP22	00:08	00:13	00:05	38.46%
UP23	00:13	00:23	00:10	43.48%
UP24	00:09	00:19	00:10	52.63%
UP25	00:37	00:43	00:06	13.95%
UP26	00:08	00:19	00:11	57.89%
UP27	00:16	00:23	00:07	30.43%
UP28	00:11	00:19	00:08	42.11%
UP29	00:13	00:15	00:02	13.33%
UP30	00:02	00:03	00:01	33.33%
UP31	00:42	01:02	00:20	32.26%
				39.56%

Tabla 29: Resultados de interfaces para la tarea 2 con grupo de usuario GU2

Usuario	A2	N2	Aumento eficiencia	Porcentaje
UP32	00:41	01:07	00:26	38.81%
UP33	00:32	00:57	00:25	43.86%
UP34	00:24	00:43	00:19	44.19%
UP35	00:25	00:41	00:16	39.02%
UP36	00:51	01:18	00:27	34.62%
UP37	00:18	00:41	00:23	56.10%
UP38	00:34	00:58	00:24	41.38%
UP39	00:41	01:06	00:25	37.88%
UP40	00:36	00:56	00:20	35.71%
UP41	00:27	00:45	00:18	40.00%
UP42	00:24	00:53	00:29	54.72%
UP43	00:37	00:58	00:21	36.21%
UP44	00:38	01:21	00:43	53.09%
UP45	00:36	01:12	00:36	50.00%
UP46	00:36	00:57	00:21	36.84%
				42.83%

Tabla 30: Resultados de interfaces para la tarea 3 con grupo de usuarios GU1

Usuario	A3	N3	Aumento eficiencia	Porcentaje
UP1	00:51	01:18	00:27	34.62%
UP2	01:28	01:57	00:29	24.79%
UP3	01:42	02:34	00:52	33.77%
UP4	01:47	02:03	00:16	13.01%
UP5	02:01	02:32	00:31	20.39%
UP6	01:55	02:19	00:24	17.27%
UP7	00:12	00:32	00:20	62.50%
UP8	01:10	01:35	00:25	26.32%
UP9	00:59	01:41	00:42	41.58%
UP10	01:05	01:24	00:19	22.62%

UP11	00:53	01:14	00:21	28.38%
UP12	00:48	01:37	00:49	50.52%
UP13	01:35	03:24	01:49	53.43%
UP14	00:52	01:20	00:28	35.00%
UP15	00:50	02:36	01:46	67.95%
UP16	01:31	02:35	01:04	41.29%
UP17	00:39	00:53	00:14	26.42%
UP18	00:18	00:29	00:11	37.93%
UP19	00:18	00:41	00:23	56.10%
UP20	02:00	02:34	00:34	22.08%
UP21	00:47	01:21	00:34	41.98%
UP22	00:21	00:30	00:09	30.00%
UP23	00:31	00:46	00:15	32.61%
UP24	00:49	01:28	00:39	44.32%
UP25	01:23	01:54	00:31	27.19%
UP26	00:15	00:36	00:21	58.33%
UP27	00:32	00:43	00:11	25.58%
UP30	00:02	00:09	00:07	77.78%
UP31	01:42	03:03	01:21	44.26%
				37.86%

Tabla 31: Resultados de interfaces para la tarea 3 con grupo de usuarios GU2

Usuario	A3	N3	Aumento eficiencia	Porcentaje
UP32	00:41	01:05	00:24	36.92%
UP33	00:29	00:57	00:28	49.12%
UP34	00:21	00:49	00:28	57.14%
UP35	00:22	00:53	00:31	58.49%
UP36	00:27	00:58	00:31	53.45%
UP37	00:32	01:02	00:30	48.39%
UP38	00:25	00:56	00:31	55.36%
UP39	00:32	01:10	00:38	54.29%
UP40	00:29	00:51	00:22	43.14%
UP41	00:25	00:56	00:31	55.36%
UP42	00:37	01:04	00:27	42.19%
UP43	00:43	01:12	00:29	40.28%
UP44	00:42	01:09	00:27	39.13%
UP45	00:39	01:02	00:23	37.10%
UP46	00:21	00:47	00:26	55.32%
				48.38%

Tabla 32: Resultados de interfaces para la tarea 4 con grupo de usuarios GU1

Usuario	A4	N4	Aumento eficiencia	Porcentaje
UP1	00:52	01:40	00:48	48.00%
UP2	02:55	04:20	01:25	32.69%
UP3	01:49	02:26	00:37	25.34%
UP4	02:34	03:12	00:38	19.79%
UP5	01:02	03:06	02:04	66.67%
UP6	01:55	03:38	01:43	47.25%
UP7	00:50	01:28	00:38	43.18%
UP9	01:53	02:35	00:42	27.10%
UP10	02:12	02:54	00:42	24.14%
UP11	01:19	02:37	01:18	49.68%
UP12	01:03	01:18	00:15	19.23%
UP13	02:20	04:12	01:52	44.44%
UP14	01:04	02:23	01:19	55.24%
UP15	01:03	02:34	01:31	59.09%
UP16	02:35	03:12	00:37	19.27%
UP17	00:39	01:00	00:21	35.00%
UP18	00:28	00:58	00:30	51.72%
UP19	01:21	02:13	00:52	39.10%
UP21	00:43	01:16	00:33	43.42%
UP22	00:44	01:07	00:23	34.33%
UP23	00:37	01:16	00:39	51.32%
UP24	01:47	02:26	00:39	26.71%
UP25	01:37	02:33	00:56	36.60%
UP26	01:28	02:16	00:48	35.29%
UP27	00:51	01:13	00:22	30.14%
UP28	00:23	01:06	00:43	65.15%
UP30	00:01	00:06	00:05	83.33%
UP31	01:46	03:03	01:17	42.08%
				41.26%

Tabla 33: Resultados de interfaces para la tarea 4 con grupo de usuarios GU2

Usuario	A4	N4	Aumento eficiencia	Porcentaje
UP32	00:51	01:25	00:34	40.00%
UP33	00:42	01:12	00:30	41.67%
UP34	00:37	01:15	00:38	50.67%

UP35	00:49	01:32	00:43	46.74%
UP36	00:58	01:36	00:38	39.58%
UP37	00:43	01:27	00:44	50.57%
UP38	01:01	01:35	00:34	35.79%
UP39	00:43	01:23	00:40	48.19%
UP40	00:45	01:11	00:26	36.62%
UP41	00:49	01:38	00:49	50.00%
UP42	00:33	01:27	00:54	62.07%
UP43	00:57	01:35	00:38	40.00%
UP44	00:47	01:41	00:54	53.47%
UP45	00:46	01:37	00:51	52.58%
UP46	00:43	01:12	00:29	40.28%
				45.88%

Tabla 34: Resultados de interfaces para la tarea 5 con grupo de usuarios GU1

Usuario	A5	N5	Aumento eficiencia	Porcentaje
UP1	02:04	03:26	01:22	39.81%
UP2	02:03	03:14	01:11	36.60%
UP3	02:09	03:21	01:12	35.82%
UP5	01:29	03:00	01:31	50.56%
UP6	02:16	03:29	01:13	34.93%
UP7	00:41	01:10	00:29	41.43%
UP8	01:17	01:54	00:37	32.46%
UP9	01:18	02:02	00:44	36.07%
UP10	01:55	03:11	01:16	39.79%
UP11	01:27	02:23	00:56	39.16%
UP12	01:15	01:50	00:35	31.82%
UP13	01:04	03:20	02:16	68.00%
UP14	01:04	02:31	01:27	57.62%
UP15	01:34	02:47	01:13	43.71%
UP17	01:02	01:53	00:51	45.13%
UP18	01:11	02:47	01:36	57.49%
UP19	00:54	01:38	00:44	44.90%
UP21	01:10	02:07	00:57	44.88%
UP22	01:30	02:43	01:13	44.79%
UP23	00:56	01:51	00:55	49.55%
UP24	01:03	01:30	00:27	30.00%
UP25	02:04	03:04	01:00	32.61%

UP26	01:44	02:37	00:53	33.76%
UP27	00:34	00:58	00:24	41.38%
				42.18%

Tabla 35: Resultados de interfaces para la tarea 5 con grupo de usuarios GU2

Usuario	A5	N5	Aumento eficiencia	Porcentaje
UP32	00:50	01:37	00:47	48.45%
UP33	01:01	01:34	00:33	35.11%
UP34	00:46	01:17	00:31	40.26%
UP35	00:51	01:37	00:46	47.42%
UP36	00:46	01:17	00:31	40.26%
UP37	00:42	01:15	00:33	44.00%
UP38	01:02	01:39	00:37	37.37%
UP39	00:52	01:31	00:39	42.86%
UP40	00:48	01:24	00:36	42.86%
UP41	00:54	01:38	00:44	44.90%
UP42	00:48	01:37	00:49	50.52%
UP43	00:57	01:34	00:37	39.36%
UP44	00:56	01:47	00:51	47.66%
UP45	01:04	02:03	00:59	47.97%
UP46	00:52	01:41	00:49	48.51%
				43.83%

Para tener una visión consolidada del porcentaje de aumento en eficiencia para cada tarea, a continuación se presenta la **Tabla 36**. En dicha tabla podemos observar que el aumento en eficiencia es mayor para usuarios con experiencia en el dominio del problema (**GU2**), lo que no demerita la ganancia en eficiencia para usuarios sin experiencia en el dominio del problema.

Tabla 36: Condensado de aumento en eficiencia por tarea

Usuarios	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5	Promedio
GU1	39.56%	35.62%	37.86%	41.26%	42.18%	39.30%
GU2	49.65%	42.83%	48.38%	45.88%	43.83%	46.12%
Promedio general de aumento en eficiencia						42.71%

Como ya se mencionó en la descripción del caso de estudio, la segunda parte de este, consistió en medir la satisfacción del usuario con las interfaces que fueron adaptadas de acuerdo a sus características, para esto se le aplicó a cada usuario que participó en el caso de estudio un cuestionario inmediatamente después de haber interactuado con las interfaces, los resultados de dichos cuestionarios son mostrados a continuación.

La primera del conjunto de tablas de esta parte del caso de estudio representa el promedio de satisfacción de los usuarios para las cinco tareas base, la primera fila describe la satisfacción con interfaces normales, y la segunda fila, la satisfacción con interfaces adaptivas. En la tercera fila se presenta el aumento en satisfacción de las interfaces adaptivas respecto a las normales.

Tabla 37: Satisfacción de los usuarios con las diferentes interfaces

	AS1	AS2	AS3	AS4	AS5	AS6	AS7	Promedio
Interfaces normales	52.40%	54.79%	57.88%	53.08%	48.97%	50.34%	52.74%	52.89%
Interfaces adaptivas	85.96%	86.64%	88.01%	85.96%	89.04%	80.48%	83.90%	85.71%
	33.56%	31.85%	30.14%	32.88%	40.07%	30.14%	31.16%	32.83%

Para una mejor apreciación de la satisfacción de los usuarios, a continuación se presentan por separado la satisfacción de usuarios sin experiencia en el dominio del problema (GU1) y los usuarios con experiencia (GU2).

Tabla 38: Satisfacción de usuarios del grupo GU1 con las diferentes interfaces

	AS1	AS2	AS3	AS4	AS5	AS6	AS7	Promedio
Interfaces normales	50.00%	51.29%	54.31%	48.28%	48.28%	44.83%	46.55%	49.08%
Interfaces adaptivas	76.29%	77.16%	78.45%	74.57%	78.02%	69.83%	74.57%	75.55%
	26.29%	25.86%	24.14%	26.29%	29.74%	25.00%	28.02%	26.48%

Tabla 39: Satisfacción de usuarios del grupo GU2 con las diferentes interfaces

	AS1	AS2	AS3	AS4	AS5	AS6	AS7	Promedio
Interfaces normales	33.33%	36.67%	40.00%	40.00%	21.67%	41.67%	41.67%	36.43%

Interfaces adaptivas	83.33%	83.33%	86.67%	90.00%	91.67%	83.33%	80.00%	85.48%
	50.00%	46.67%	46.67%	50.00%	70.00%	41.67%	38.33%	49.05%

Apoyados en las tablas anteriores, podemos decir que en todas las preguntas del cuestionario existe un aumento en satisfacción con las interfaces adaptivas respecto a las interfaces normales. Lo anterior puede observarse más fácilmente en las siguientes gráficas.

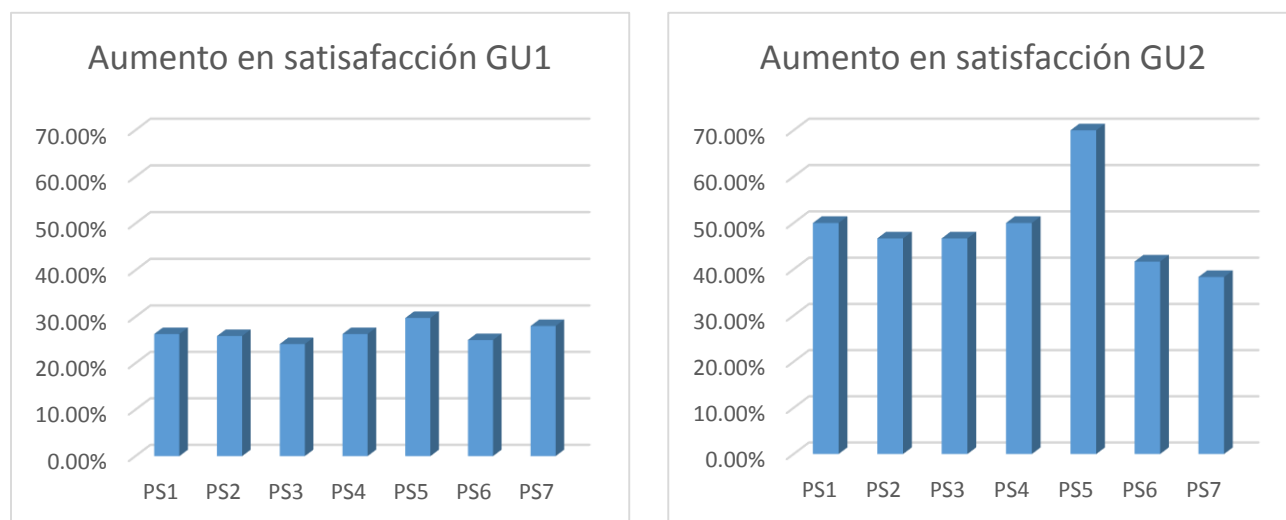


Figura 28: Aumento en satisfacción de interfaces adaptivas respecto a normales

En las gráficas anteriores podemos observar claramente la diferencia en el aumento de satisfacción entre usuarios sin experiencia en el dominio del problema, y usuarios con experiencia. Otro fenómeno que cabe recalcar es que la pregunta que obtuvo mejores resultados en ambos grupos de usuarios es la referente a la legibilidad del texto en pantalla (PS5).

4.2.5 Conclusiones del caso de estudio.

De la primera parte del caso de estudio podemos concluir que el utilizar interfaces adaptadas a las características del usuario tiene un impacto positivo en su eficiencia, y dicho impacto es aún mayor en usuarios con experiencia en el dominio del problema.

De la segunda parte del caso de estudio, donde se evaluó la satisfacción de los usuarios con las interfaces adaptivas, podemos concluir que el adaptar las interfaces a las características del usuario tiene un impacto positivo en la satisfacción que el usuario obtiene con las interfaces adaptadas debido a que le es más fácil ingresar datos, visualizar los datos que el sistema le presenta, y en general le facilita la realización de la tarea.

Como conclusión general y apoyados en los datos que revelan un aumento en eficiencia del 42% y en la satisfacción del 32% en promedio, podemos concluir que la propuesta de crear un motor de adaptación basado en reglas y en un modelo de usuario no solo es factible, sino que además tiene un impacto positivo en la usabilidad de aplicaciones empresariales.

CONCLUSIONES

“I never did anything by accident, nor did any of my inventions come by accident; they came by work.”

- Thomas A. Edison -

5 CONCLUSIONES

El objetivo principal que en este trabajo de tesis se desarrolló fue el de generar un modelo que permita automatizar la generación de interfaces de usuario capaces de adaptarse a las características específicas de cada uno de los usuarios del sistema. Como medio para lograr dicho objetivo se realizó la propuesta de *método para la generación automática de interfaces de usuario adaptivas* **MEGAIA**.

Dicha propuesta está integrada por cuatro elementos principales; un modelo de usuario, prototipos de interfaz, reglas de adaptación y motor de adaptación.

Como modelo de usuario, se utilizó una propuesta de modelo integral de usuario en la cual se busca considerar todos los atributos de usuario que toman o pueden tomar parte en el proceso de interacción humano computadora. Dicha propuesta fue realizada en conjunto con otro estudiante de maestría. La propuesta de modelo integral de usuario no fue tratada en este trabajo ya que fue desarrollada su aplicabilidad en otra tesis y discutida en una conferencia internacional en el año 2011.

Se propone que un prototipo de interacción está conformado por un conjunto de componentes de interacción, y se presentan 5 postulados para demostrar la relación existente entre una tarea y un componente de interacción. Para la representación formal de los componentes de interacción se propuso un esquema.

Para la generación automática de prototipos de interfaz se proponen dos reglas del tipo IF...THEN las cuales permiten seleccionar de un catálogo de componentes de interacción un componente que permita a un usuario realizar una acción de una tarea.

Se propuso una estructura para describir las reglas de adaptación y se eligió utilizar XML como formato de archivos para almacenar dichas reglas. Se describen dos reglas formales del tipo IF...THEN mediante las cuales pueden ser evaluadas las reglas de adaptación, así como también se proponen dos esquemas para representar el componente adaptivo de interacción el cual es el elemento de salida de las reglas de adaptación.

Para ejemplificar las reglas de adaptación, se presentan 3 reglas de adaptación para diferentes valores del atributo de usuario *visión* y 2 reglas para el atributo *experiencia con el dispositivo*.

Se propone utilizar un motor de adaptación como mecanismo encargado de aplicar las reglas de adaptación al prototipo de interfaz para de esa manera generar la interfaz de usuario de acuerdo a las características específicas de cada uno de los usuarios, proceso el cual es realizado en tiempo de ejecución de la aplicación.

Con la finalidad de probar el método propuesto y sus elementos constituyentes, se realizaron dos casos de estudio, uno para probar la factibilidad de la generación automática de prototipos de interfaz y el segundo para probar el motor de adaptación junto con las reglas de adaptación.

Los resultados obtenidos de este experimento apoyan la propuesta realizada y demuestran su factibilidad, demuestran que es de gran utilidad especialmente si los diseñadores no tienen experiencia con la plataforma para la que están desarrollando. También permitieron medir el nivel de efectividad y completud.

El segundo experimento demostró que la estructura propuesta para las reglas de adaptación es factible de implementar, y que el formato de archivo seleccionado soporta dicha estructura. También en este caso de estudio demuestra que es posible automatizar la generación de interfaces adaptivas y que el resultado de implementar dicho tipo de interfaces en aplicaciones empresariales para dispositivos móviles que utilizan plataforma Android tiene un impacto positivo en atributos de usabilidad como son la eficiencia y la satisfacción del usuario, con lo cual se cumple el objetivo principal de este trabajo de tesis y se prueban las primeras dos hipótesis en este presentadas.

Como conclusiones generales podemos decir que nuestra propuesta de un motor genérico de interfaces de software es implementable y tiene un impacto positivo en la usabilidad general de la aplicación, y que es posible generar automáticamente prototipos de interfaz de usuario basado en el detalle de la tarea o descripción del caso uso.

5.1 Trabajo a futuro

Si bien el resultado de la propuesta fue satisfactorio, eso no significa que el trabajo de investigación en el tema propuesto termine aquí, aún es posible realizar mejoras al método propuesto, para lo cual se requiere más investigación.

Aun es necesario realizar más pruebas sobre el método y las herramientas aquí propuestas, así como integrar el método en un ambiente real, para de esa manera obtener retroalimentación sobre debilidades o fallas que pueda presentar el método y realizar las mejoras necesarias.

Una de las principales mejoras que consideramos debería realizarse en la fase de generación de prototipos, es la de generar derivar automáticamente los prototipos de interfaz directamente de la descripción de los casos de uso.

Sobre la fase de adaptación de la interfaz sin duda sería de mucha utilidad desarrollar un módulo que permita al usuario realizar adaptaciones a la interfaz en tiempo de ejecución, así como también sería un gran avance el incluir a esta propuesta un módulo inteligente que le permitiera al sistema aprender acerca del usuario según los tiempos y errores resultantes del uso prolongado del sistema.

VI

REFERENCIAS

“Both the man of science and the man of action live always at the edge of mystery, surrounded by it.”

- J. Robert Oppenheimer -

6 REFERENCIAS

- Abascal, Julio, Ignacio Aedo, Jose Cañas, Miguel Gae, and Ana Belen Gil. 2001. *La Interacción Persona-Ordenador*. Primera. ed. Jesús Lores. Lleida. <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:La+interacción+persona-ordenador#0> (December 24, 2013).
- Almanza, Tania E. 2012. “Modelo Basado En Esquemas y Relaciones Entre Tareas, Perfil de Usuario y Patrones de Usabilidad Para Diseñar Interfaces Gráficas de Usuario Adaptivas.” Universidad Autónoma de Baja California.
- Al-qutaish, Rafa E. 2010. “Quality Models in Software Engineering Literature : An Analytical and Comparative Study.” *Science* 6(3): 166–75.
- Andrina Grani, Jelena Naki. 2007. “Meeting User Individual Characteristic Throun Adaptive Interface of an e-Learning System: An Empitical Studi Desing.” 333–38.
- Annett, John. 2004. “Hiierarchical Task Analysis.” In *The handbook of task analysis for human-computer interaction.*, eds. Dan Diaper and Neville Staton. Mahwah, New Jersey: Lawrence Erlbaum associates, 67–83.
- Annett, John, and Keith D Duncan. 1967. “Task Analysis and Training Design.”
- Des Auges, Die Functionsprufungen, Herman Snellen, and Edmund Landolt. 1874. “Capitel I.” *Handbuch der gesamten Augenheilkunde* 1.
- Balint, Lajos. 1995. “Adaptive Interfaces for Human-Computer Interaction: a Colorful Spectrum of Present and Future Options.” In *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century.*, *IEEE International Conference on*, , 292–97.
- Bentley, Richard, Wolfgang Appelt, Uwe Busbach, Elke Hinrichs, Douglas Kerr, Klaas Sikkell, Jonathan Trevor, and Gerd Woetzel. 1997. “Basic Support for Cooperative Work on the World Wide Web.”
- Benyon, David. 1993. “Accommodating Individual Differences through an Adaptive User Interface.” *HUMAN FACTORS IN INFORMATION TECHNOLOGY* 10: 149.
- Benyon, David, and Dianne Murray. 1991. “Applying User Modelling to Human-Computer Interaction Design.” 1–30.
- Bergman, Eric, and Earl Johnson. 1995. “Towards Accessible Human-Computer Interaction.” *Advances in human-computer interaction* 5: 87–113.

- Biswas, Pradipta, and Peter Robinson. 2010. "A Brief Survey on User Modelling in HCI." In *Proc. of the International Conference on Intelligent Human Computer Interaction (IHCI) 2010*,.
- Boehm, B W. 1978. *Characteristics of Software Quality*. Second. North-Holland Pub. Co. <http://books.google.com.mx/books?id=Cdm0AAAAIAAJ>.
- Borchers, Jan O. 2001. "A Pattern Approach to Interaction Design." *AI & Society* 15: 359–76.
- Brusilovsky, Peter, and Mark T Maybury. 2002. "From Adaptive Hypermedia to the Adaptive Web." *Communications of the ACM* 45(5): 30–33.
- Crystal, Abe, and Beth Ellington. 2004. "Task Analysis and Human-Computer Interaction : Approaches , Techniques , and Levels of Analysis." *Americas The* (August): 1–9. <http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1967&context=amcis2004>.
- Dawe, Melissa. 2006. "Desperately Seeking Simplicity: How Young Adults with Cognitive Disabilities and Their Families Adopt Assistive Technologies." In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, , 1143–52.
- Designer, Every, Needs To, and Know About. *100 THINGS*.
- Diaper, D., D. Gilmore, G. Cockton, and B. Shackerl. 1990. "Task Analysis: The Oft Missed Step in the Development of Computer-Human Interfaces: Its Desirable Nature, Value and Role." *Human-computer Interaction: Interact*: 1051–54.
- Diaper, Dan, and Colston Sanger. 2006. "Tasks for and Tasks in Human–computer Interaction." *Interacting with Computers* 18(1): 117–38. <http://www.sciencedirect.com/science/article/pii/S0953543805000603>.
- Diaper, Dan, and Neville Stanton, eds. 2004. *The Handbook of Task Analysis for Human-Computer Interaction*. Mahwah, New Jersey: Lawrence Erlbaum associates.
- Dordević, N, and D Rančić. 2008. "Testing Cognitive Characteristics of Users in Interaction with Computer." *WSEAS Transactions on Information Science and Applications* 5(3): 300–305. <http://www.scopus.com/inward/record.url?eid=2-s2.0-48849093353&partnerID=40&md5=fe052bc29075d143ab5759a0bca07461>.
- Dumas, Joseph S, and Beth A Loring. 2008. *Moderating Usability Tests: Principles and Practices for Interacting: Principles and Practices for Interacting*. Morgan Kaufmann.
- Eclipse. 2010. "Eclipse Standard." <https://www.eclipse.org/downloads/packages/eclipse-standard-431/keplersr1>.
- Ferré, Xavier. 2005. "Marco de Integración de La Usabilidad En El Proceso de Desarrollo de Software." Universidad politécnica de Madrid, Facultad de Informática.

- Ferre, Xavier, Natalia Juristo, Ana M Moreno, M Isabel Sánchez, Facultad De Informática, Universidad Politécnica De Madrid, and Boadilla Monte. 2001. "A Software Architectural View of Usability Patterns." *Architectural Design* (1). <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.6084&rep=rep1&type=pdf>.
- Findlater, Leah, and Krzysztof Z Gajos. 2009. "Design Space and Evaluation Challenges of Adaptive Graphical User Interfaces." *AI Magazine* 30: 68–73. <https://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/2268>.
- Gajos, Krzysztof Z., Jing Jing Long, and Daniel S. Weld. 2006. "Automatically Generating Custom User Interfaces for Users with Physical Disabilities." *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility - Assets '06*: 243. <http://portal.acm.org/citation.cfm?doid=1168987.1169036>.
- Gajos, Krzysztof Z., Daniel S. Weld, and Jacob O. Wobbrock. 2010. "Automatically Generating Personalized User Interfaces with Supple." *Artificial Intelligence* 174(12-13): 910–50. <http://linkinghub.elsevier.com/retrieve/pii/S0004370210000822> (August 8, 2013).
- Gea, M., and L. Gutierrez. 2001. El diseño *La Interacción Persona-Ordenador*. Granada, España: In J. Lores.
- Google Inc. 2013a. "Android Design and Accesibility Patterns." <http://developer.android.com/design/index.html>.
- . 2013b. "Google Forms." 2001. <http://www.google.com/google-d-s/createforms.html>.
- . 2014a. "Desing Principles." <http://developer.android.com/design/get-started/principles.html> (January 11, 2014).
- . 2014b. "Developer Tools." <http://developer.android.com/tools/index.html> (January 10, 2014).
- Gyurky, Szabolcs Michael. 2006. *The Congnitive Dynamics on Computer Science*. New York, New York, USA: Wiley-Interscience.
- Henricksen, Karen, and Jadwiga Indulska. 2001. "Adapting the Web Interface: An Adaptive Web Browser." In *Australian Computer Science Communications*, , 21–28.
- Holzinger, Andreas, Michael Geier, and Panagiotis Germanakos. 2010. "ON THE DEVELOPMENT OF SMART ADAPTIVE USER INTERFACES FOR MOBILE E-BUSINESS APPLICATIONS Towards Enhancing User Experience – Some Lessons Learned."
- Hothi, Jatinder, and Wendy Hall. 1998. "An Evaluation of Adapted Hypermedia Techniques Using Static User Modelling." *2nd WS on Adaptive Hypertext and Hypermedia, USA*.

- International Organization For Standardization Iso. 2001. "ISO/IEC 9126-1." *Software Process: Improvement and Practice 2*: 1–25. <http://ebookbrowse.com/iso-iec-9126-1-2001-pdf-d72715451>.
- Inzunza, Sergio, A Mejia, R Juarez-Ramirez, and Manuel Gomez-Ruelas. 2011. "Implementing User-Oriented Interfaces: From User Analysis to Framework's Components." In *Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference on*, , 107–10.
- Jacob, R.J. 2000. "User Interface." In *Encyclopedia of computer science*, eds. E.D. Reilly and D. Hemmendinger. London: Nature Publishing Group, 1821–26. <http://www.cs.tufts.edu/~jacob/papers/encycs.pdf>.
- Johnson, Addie, and Niels Taatgen. 2005. "User Modeling." In *Handbook of human factors in web design*, Erbaum, N.J.: Nahwah, 424–38.
- Johnson, Jeff. 2010. *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Johnson, P. 1989. "Supporting System Design by Analyzing Current Task Knowledge." In *Task Analysis for Human-Computer Interaction*, ed. D. Diaper. Chichester, England: Ellis Horwood Limited, 160–84.
- Juárez-Ramírez, Reyes, Manuel Gomez-Ruelas, Alan A Gutierrez, and Pavel Negrete. 2011. "Towards Improving User Interfaces: a Proposal for Integrating Functionality and Usability Since Early Phases." In *Uncertainty Reasoning and Knowledge Engineering (URKE), 2011 International Conference on*, , 119–23.
- Kumar, Rajendra. 2005. *Human Computer Interaction*. New Delh: Firewall Media.
- Langley, Pat, and Michael Fehling. 1998. "The Experimental Study of Adaptive User Interfaces." *Institute for the Study of Learning and Expertise: Palo Alto, CA. Teehnieal Report*: 93–98.
- Langley, Pat, and Haym Hirsh. 1999. "User Modeling in Adaptive Interfaces." *Courses and lectures-international centre for mechanical sciences*: 357–70.
- Lavie, Talia, and Joachim Meyer. 2010. "Benefits and Costs of Adaptive User Interfaces." *International Journal of Human-Computer Studies* 68(8): 508–24. <http://linkinghub.elsevier.com/retrieve/pii/S1071581910000145> (March 8, 2013).
- Lester, Cynthia Y. 2008. *First International Conference on Advances in ComputerHuman Interaction Advancing the Multidisciplinary Nature of Human Computer Interaction in a Newly Developed Undergraduate Course*.

- Mayhew, Deborah J. 1999. *The Usability Engineering Lifecycle A Practitioners Handbook for User Interface Design* *The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design*.
http://books.google.co.uk/books/about/The_Usability_Engineering_Lifecycle.html?id=q_9tlu__DgC.
- McCall, Jim A., Paul K. Richards, and Gene F. Walters. 1977. "Factors in Software Quality." *Nat'l Tech. Information Servicel* 1, 2 and 3.
<http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA049055>.
- Mejía, Andres. 2013. "Modelo Para La Conservación de Los Atributos de Calidad de Mantenibilidad y Flexibilidad de La Arquitectura Del Software Con La Integración Del Modelo de Usuario." Universidad Autónoma de Baja California.
- Mejía, Andrés, Reyes Juárez-Ramírez, Sergio Inzunza, and Rocio Valenzuela. 2012. "Implementing Adaptive Interfaces: a User Model for the Development of Usability in Interactive Systems." In *Proceedings of the CUBE International Information Technology Conference*, New York, NY, USA: ACM, 598–604.
- Molina, Fernando, and Ambrosio Toval. 2009. "Integrating Usability Requirements That Can Be Evaluated in Design Time into Model Driven Engineering of Web Information Systems." *Advances in Engineering Software* 40(12): 1306–17.
<http://www.sciencedirect.com/science/article/pii/S096599780900043X>.
- Nielsen, J. 1992. "The Usability Engineering Life Cycle." *Computer* 25.
- Nielsen, Jakob, and Morgan Kaufmann. 1993. "Usability Engineering." *Usability Engineering* 44: 362. <http://www.useit.com/jakob/useengbook.html>.
- Nielsen, Michael, Moritz Störring, Thomas B. Moeslund, and Erik Granum. 2004. "A Procedure for Developing Intuitive and Ergonomic Gesture Interfaces for HCI." In *Proc. of Gesture Workshop 2003*, , 409–20. http://link.springer.com/chapter/10.1007/978-3-540-24598-8_38.
- Norman, Donald A. 1988. *The Psychology of Everyday Things*. Basic books.
- Peissner, Matthias, Andreas Schuller, and Dieter Spath. 2011. "A Design Patterns Approach to Adaptive User Interfaces for Users with Special Needs *." 268–77.
- Preece, Jenny, Helen Rogers, David Benyon, Simon Holland, and Tom Carey. 1994. *Human-Computer Interaction: Concepts And Design (ICS)*. 1st ed. Addison Wesley.
http://www.amazon.com/Human-Computer-Interaction-Concepts-And-Design/dp/0201627698/ref=sr_1_fkmr0_1?ie=UTF8&qid=1377056065&sr=8-1-fkmr0&keywords=j.+reece+human+computer+interaction+1994.
- Pressman, Roger. 2010. *Software Engineering: A Practitioner's Approach*. 7th ed. New York, NY, USA: McGraw-Hill, Inc.

- Quesenbery, Whitney. 2001. "What Does Usability Mean : Looking Beyond ' Ease of Use '." In *Proceedings of the 48th Annual Conference, Society for Technical Communication*, , 1–8.
- Quesenbery, Whittney. 2004. "Balancing the 5Es of Usability." *Cutter IT Journal* 17(2): 4–11.
- Rich, Elaine. 1979. "User Modeling via Stereotypes." *Cognitive science* 3(4): 329–54.
- Ritter, Frank E., Gordon D. Baxter, Gary Jones, and Richard M. Young. 2000. "Supporting Cognitive Models as Users." *ACM Transactions on Computer-Human Interaction* 7: 141–73.
- Schneider-Hufschmidt, Matthias, Uwe Malinowski, and Thomas Kuhme. 1993. *Adaptive User Interfaces: Principles and Practice*. New York, NY, USA: Elsevier Science Inc.
- Seffah, Ahmed, and Eduard Metzker. 2008. *Adoption-Centric Usability Engineering: Systematic Deployment, Assessment and Improvement of Usability Methods in Software Engineering*. 1st ed. Springer Publishing Company, Incorporated.
- Smits, David, and Paul De Bra. 2011. "GALE: a Highly Extensible Adaptive Hypermedia Engine." In *Proceedings of the 22nd ACM conference on Hypertext and hypermedia*, , 63–72.
- Soulard, Olivier. 2012. "Development of an Adaptive User Interface for Mobile Indoor Navigation." Technische Universität München.
- Sousa, K, and E Furtado. 2005. "From Usability Tasks to Usable User Interfaces." *4th International Workshop on Task Models and Diagrams, TAMODIA 2005*: 103–10.
- Stavrinoudis, Dimitris, Michalis Xenos, and Dimitris Christodoulakis. 1999. "RELATION BETWEEN SOFTWARE."
- Stephanidis, C, A Paramythis, M Sfyraakis, A Stergio, N Maou, A Leventis, G Paparoulis, and C Karagiannidis. 1998. "Adaptable and Adaptive User Interfaces for Disabled Users in the AVANTI Project." *Architecture* 1430: 153.
<http://www.springerlink.com.offcampus.lib.washington.edu/content/pna66jeq2dvuu5kd>.
- Stephanidis, C, a Paramythis, M Sfyraakis, a Stergiou, N Maou, a Leventis, G Paparoulis, and C Karagiannidis. 1998. "Adaptable and Adaptive User Interfaces for Disabled Users in the AVANTI Project." *Architecture* 1430: 153.
<http://www.springerlink.com.offcampus.lib.washington.edu/content/pna66jeq2dvuu5kd>.
- Thurnher, Bettina. 2004. *Usability Engineering*. Vienna, Austria: Institute of Software Technology and Interactive Systems.
- Vassileva, Julita. 1994. "A Practical Architecture for User Modeling in a Hypermedia-Based Information System." In *4-th International Conference on User Modeling*, , 115–20.

- Virzi, Robert A., Jeffrey L. Sokolov, and Demetrios Karis. 1996. "Usability Problem Identification Using Both Low- and High-Fidelity Prototypes." In *Proceedings of the SIGCHI conference on Human factors in computing systems common ground - CHI '96*, , 236–43. <http://portal.acm.org/citation.cfm?doid=238386.238516>.
- Vogel, Thomas, and Holger Giese. 2013. *Model-Driven Engineering of Adaptation Engines for Self-Adaptive Software*. Potsdam: Universitätsverlag Potsdam.
- Vogt, Joël, and Andreas Meier. 2010. "An Adaptive User Interface Framework for eHealth Services Based on UIML 2 Use Case : Patient Visit." (2009): 409–22.
- Weinschenk, Susan. 2011. *100 Things Every Designer Needs to Know About People*. Pearson Education.
- Williams, Alan Webber, and Robert L Probert. 2002. *Software Component Interaction Testing: Coverage Measurement and Generation of Configurations*. University of Ottawa.
- Zarikas, V. 2007. "Modeling Decisions Under Uncertainty in Adaptive User Interfaces." *Universal Access in the Information Society* 6(1): 87–101. <http://www.scopus.com/inward/record.url?eid=2-s2.0-34548038372&partnerID=40&md5=9ac86e89750e3005f1ea3c93819f99e0> (April 30, 2013).

VII

APÉNDICE

“I worked with such concentration and focus and I had hundreds of obscure engineering or programming things in my head. I was just real exceptional in that way.”

- Steve Wozniak -

7 APÉNDICE

7.1 Apéndice A: Definición de tareas para realizar prototipos de interfaz

A continuación se adjunta el documento entregado a los usuarios durante el caso de estudio 1 (véase sección 4.1) en el cual se describen una serie de tareas para las cuales los usuarios debían realizar un prototipo de interfaz.

Prototipos Android

Objetivo: Generar los prototipos de interfaz para las siguientes tareas, considerando que su rango de usuarios es de 50 personas, de las cuales 10 tienen una debilidad visual, y 20 de sus usuarios nunca han utilizado un dispositivo Android.

Tarea 1

Título de la Tarea: Iniciar sesión

Información de la tarea: El usuario quiere iniciar sesión en el ERP de la empresa

Pasos:

1. Ingresar el Nombre de usuario
2. Ingresar la Contraseña
3. Seleccionar la opción
 - a. *Aceptar*
 - b. *Cancelar*

Tarea 2

Título de la tarea: Agregar Cuenta Bancaria

Información de la tarea: En el ERP de la empresa, se va a agregar una nueva cuenta bancaria al catálogo de cuentas

Pasos:

1. Ingresar el Número de cuenta
2. Ingresar el Nombre del banco
3. Ingresar el Numero de cheque
4. Seleccionar el Número de Cuenta Contable (la lista de cuentas son las capturadas en el catálogo de cuentas contables)
5. Seleccionar la opción
 - a. *Aceptar*

- b. *Cancelar*

Tarea 3

Título de la tarea: Crear Factura

Información de la tarea: En el ERP de la empresa, se va a crear una nueva factura

Pasos:

1. Ingresar la Razón Social del cliente
2. Ingresar el RFC del cliente
3. Ingresar la Dirección del cliente
4. Ingresar el Concepto de la factura
5. Ingresar la Cantidad
6. Ingresar el Importe
7. Mostrar el Subtotal de la factura
8. Mostrar el Total de la factura
9. Seleccionar la opción
 - a. *Aceptar*
 - b. *Cancelar*

Tarea 4

Título de la tarea: Agregar nuevo cliente

Información de la tarea: En el ERP de la empresa, es necesario agregar un cliente al catálogo de clientes, los pasos se enlistan a continuación

Pasos:

1. Seleccionar el tipo de cliente (maquiladora, iglesia, primaria, secundaria, preparatoria)
2. Ingresar el nombre del cliente
3. Ingresar el nombre de contacto
4. Ingresar el número de contacto
5. Ingresar el correo del contacto
6. Ingresar el RFC
7. Ingresar la fecha de vencimiento
8. Seleccionar el día de contra-recibo (días de la semana)
9. Ingresar la hora del contra-recibo
10. Seleccionar la opción
 - a. *Aceptar*
 - b. *Cancelar*

Tarea 5

Título de la tarea: Agregar Empleado

Información de la tarea: En el ERP de la empresa, es necesario agregar un empleado al catálogo de empleados de la empresa

Pasos:

1. Seleccionar el tipo de empleado (administración, operador, taller, supervisor)
2. Ingresar el Nombre
3. Ingresar Apellidos
4. Ingresar fecha de ingreso
5. Ingresar fecha de nacimiento
6. Ingresar RFC
7. Ingresar Número de teléfono
8. Ingresar dirección
9. Ingresar el Numero de licencia
10. Ingresar fecha de vencimiento de la licencia
11. Seleccionar la opción
 - a. *Aceptar*
 - b. *Cancelar*

7.2 Apéndice B: Gráfica de Snellen para prueba de visión a corta distancia

A continuación se incluye la gráfica de Snellen utilizada para realizar la prueba de visión al usuario durante el segundo caso de estudio.

Near Vision Test Card

Hold at a distance of 16 inches (40.6 centimeters).

160 in.	E O P Z T L C D F	4.0 m
80 in.	T D P C F Z O E L	2.0 m
56 in.	D Z E L C F O T P	1.4 m
48 in.	F E P C T L O Z D	1.2 m
40 in.	P T L F C Z D E O	1.0 m
32 in.	E L Z T C O F P D	80 cm
24 in.	D Z E C I P T O E	60 cm
20 in.	L O P P E R I O T	50 cm
16 in.	D I T C F A R C E	40 cm