

# **Universidad de las Ciencias Informáticas**

**Facultad 5, Laboratorio de Gestión de Proyectos**

**Facultad 3, Centro de Informatización de la Gestión de Entidades**



**Procedimiento para el aseguramiento y evaluación de la usabilidad basado en patrones arquitectónicamente sensibles para los sistemas de gestión del Centro de Informatización de la Gestión de Entidades.**

Trabajo final presentado en opción al título de  
Máster en Gestión de Proyectos Informáticos

**Autor: Ing. Iliannis Pupo Leyva**

**Tutores: (MSc) Michael González Gorrín**

**(MSc) Nemury Silega Martínez**

**Ciudad de La Habana, junio de 2012**

**DEDICATORIA Y AGRADECIMIENTOS**

*A la persona que me brindó su cariño, dedicación y amor e irá conmigo donde quiera que esté. Que sirva la presente, como un pequeño homenaje a mi abuela Mirella Cabayero*

*A mis padres queridos, que hoy soy el resultado de su educación y que le debo toda mi vida*

*A mi hermanita de alma, gracias por estar siempre a mi lado y a su novio Nelson.*

*A mi tía Ana por quererme y ayudarme como si fuera su hijita.*

*A toda mi familia que son maravillosos y estoy muy orgullosa de tenerlos a Consy, Oda, Mongo, Anabel, Vivian y a Oscarito por acogerme como su hija.*

*A mi esposo David por ser la persona más maravillosa que he conocido, y ser en estos momentos más que esposo: amigo y tutor*

*A mis tutores Nemu y el Micha, muchas gracias*

*A todos mis amigos en especial a Gise, Leo, Day, Javi, Victor, Yisel y Danaysa*

*A todos los que permitieron que un sueño se hiciera realidad*

**DECLARACIÓN JURADA DE AUTORÍA**

Declaro por este medio que yo Iliannis Pupo Leyva del registro de inscripción de nacimiento, con carné de identidad 84080421634, soy el autor principal del trabajo final de maestría “Procedimiento para el aseguramiento y evaluación de la usabilidad basado en patrones arquitectónicamente sensibles para los sistemas de gestión del CEIGE”, desarrollada como parte de la Maestría en Gestión de Proyectos Informáticos y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Y para que así conste, firmo la presente declaración jurada de autoría en Ciudad de La Habana a los días del mes de junio del año 2012.

---

**Firma**

**RESUMEN:**

La presente investigación describe un procedimiento para aseguramiento y evaluación de la usabilidad de los sistemas de gestión del Centro de Informatización de la Gestión de Entidades (CEIGE) que cuenta de tres fases que son las propuestas por la Ingeniería de la Usabilidad y que coinciden con el modelo de desarrollo del CEIGE.

La propuesta realizada tiene un importante aporte práctico, su aplicación es muy efectiva para ayudar a detectar problemas de usabilidad temprano en el ciclo de desarrollo, cuando son más fáciles y menos costosos de solucionar. Esta aproximación difiere de la idea tradicional de medir y mejorar la usabilidad una vez finalizado el sistema.

Se realiza un análisis de los resultados obtenidos de la aplicación del procedimiento que se propone en los proyectos del sistema CEDRUX, mostrándose la efectividad de la aplicación de la misma mediante los indicadores definidos.

***Palabras Claves: Arquitectura software, diseño software, usabilidad.***

**ABSTRACT**

The usability is one of the key quality attributes of software development. This research, which has three phases, describes a procedure for assurance and usability evaluation of management systems Infomatization Center Management Entities (CEIGE). The phases of the procedure are the same proposed by the Usability Engineering, which coincide with the center's development model CEIGE (Analysis, Design and Release).

The proposal is an important practical contribution; its application is very effective in helping to detect usability problems early in the development cycle, when they are easier and less costly to fix. This approach differs from the traditional idea of measuring and improving the usability when the system is complete.

The analysis of the results obtained applying the proposed procedure in CEDRUX system projects, showing the effectiveness of the application behind the indicators defined.

## INDICE

<b>CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION</b>	<b>1</b>
1.1 Introducción	1
1.2 Análisis bibliométrico	1
1.3 Diferentes enfoques del término usabilidad	2
1.3.1 Consideraciones sobre las diferentes definiciones de usabilidad	3
1.4 Evaluación de la usabilidad	3
1.4.1 Consideraciones sobre los métodos y técnicas de evaluación de la usabilidad	4
1.5 Diseño enfocado a la usabilidad	4
1.5.1 Diseño centrado en el usuario	4
1.5.2 Diseño orientado al producto	5
1.5.3 Consideraciones sobre los modelos de diseño enfocado en la usabilidad	6
1.6 Calidad y Arquitectura del Software	7
1.7 Relación entre atributos, propiedades y patrones de usabilidad	8
1.8 La importancia de la usabilidad en la arquitectura de software	10
1.9 Consideraciones sobre los modelos y normas de usabilidad	12
1.9.1 Modelos nacionales	12
1.9.2 Modelos y normas internacionales	13
1.9.2.2 Patrones de Usabilidad: Arquitectura de software que soporta la usabilidad (STATUS) y Escenarios de Software de Usabilidad (USE)	13
1.9.2.3 Escenario Nivel arquitectura basada en análisis de usabilidad (SALUTA)	13
1.9.2.4 ISO 13407: Human-centred design processes for interactive systems	13
1.9.3 Conclusiones de los modelos y normas de usabilidad a nivel internacional	14
1.10 Conclusiones parciales	17
<b>CAPITULO 2: DESCRIPCION DEL PROCEDIMIENTO</b>	<b>18</b>

2. Introducción	18
2.1 Objetivos del procedimiento.	18
2.2 Estructura del procedimiento.	18
2.3 Descripción de las fases del procedimiento	21
2.3.1 Fase de análisis	21
2.3.1.1.1 Actividad: Análisis del perfil del usuario	21
2.3.1.1.2 Actividad: Análisis de tareas	25
2.3.2 Fase de diseño	26
2.3.2.2 Actividades de la fase de diseño	27
2.3.2.2.1 Actividad: Evaluar prototipos	27
2.3.2.2.2 Actividad: Determinar escenarios y patrones de usabilidad a partir de los atributos de usabilidad.	32
2.3.2.2.3 Actividad: Rediseño	34
2.3.3. Fase de Lanzamiento o retroalimentación del usuario	35
2.3.3.2 Actividades de la fase lanzamiento o retroalimentación del usuario	35
2.3.3.2.1 Actividad: Determinar el grado de usabilidad del sistema.	35
2.4 Beneficios identificados de la aplicación de las actividades de aseguramiento y evaluación de la usabilidad en el CEIGE	36
2.6 Conclusiones parciales	37
<b>CAPITULO 3: ANALISIS DE LOS RESULTADOS</b>	38
3.1 Introducción	38
3.2 Indicadores de medición	38
3.3 Características de la muestra	39
3.4 Evaluación	39
3.4.1 Análisis del soporte de la usabilidad en la arquitectura del software	39

3.4.1.1 Análisis de los atributos de usabilidad, números de patrones y propiedades de usabilidad empleados	39
3.4.2 Análisis del indicador calidad del procedimiento	50
3.4.3 Ejecución del pre-experimento	52
3.5 Análisis del impacto económico de la propuesta	55
3.5.1 Análisis costo- beneficio	55
3.5.1.1 Explicación del modelo utilizado para el cálculo del costo-beneficio	55
3.5.1.2 Modelo aplicado al costo-beneficio de la propuesta	58
3.5.2 Disminución de los costes de desarrollo	62
3.6 Conclusiones parciales	63
<b>Conclusiones</b>	<b>64</b>
<b>Recomendaciones</b>	<b>64</b>
<b>Bibliografía</b>	<b>65</b>
<b>Anexos</b>	<b>70</b>

### Introducción

A partir del surgimiento de los grandes sistemas informáticos en el mundo, se fue incluyendo al mismo tiempo y por necesidad, el proceso de pruebas, usado sistemáticamente junto al desarrollo de cada aplicación, para verificar la calidad de los productos de software y contribuir que los mismo cumplieran con ciertos atributos como son: confiabilidad, la eficiencia y la usabilidad. (IEEE, 1990)

La usabilidad es un atributo de calidad de los productos de software, que se ha convertido paulatinamente en una necesidad y casi obligación para las empresas dedicadas al desarrollo de software. Cuestiones tales como si un producto es fácil de aprender, de usar, o si es sensible al usuario y si el usuario puede de manera eficiente completar las tareas, en gran medida puede afectar la aceptación de un producto y el éxito en el mercado. En un futuro, los usuarios se vuelven más críticos, una usabilidad pobre puede convertirse en una barrera importante para el éxito de las nuevas aplicaciones de software comercial. Por lo tanto, las organizaciones de desarrollo de software están prestando más atención a garantizar la usabilidad. Un software que no cuente con pruebas enfocadas en sus usuarios finales cierra su proyección de venta en un gran porcentaje lo que implica pérdidas grandes de dinero, es preferible realizar una inversión en pruebas de usabilidad a tiempo para poder lograr los resultados esperados. Un ejemplo real de la aplicación de esta teoría es: Clare-Marie Karat de la IBM, quienes publicaron un reporte donde demuestran las ventajas de aplicar la ingeniería de la usabilidad, compensaron los costos de la inversión inicial; IBM utilizó la usabilidad en sus aplicaciones, de esta manera lograron una vuelta en la inversión de cien veces para un producto de software particular. En ese caso, se invirtieron \$60.000 en la ingeniería de la usabilidad a través del desarrollo lo que dio lugar a ahorros de \$6.000.000 en el primer año solamente. (Vivas, 2008)

En el estándar ISO/IEC 25010 dentro del cual se identifican características de la calidad del software entre las cuales se encuentra la usabilidad, esta se define como: “la capacidad de un producto de software para ser entendido, aprendido, utilizado y atractivo hacia el usuario, cuando se usa bajo condiciones específicas”. Es decir, que la usabilidad comprende a su vez una serie de atributos del software relacionados con el esfuerzo necesario para su uso, y la valoración individual de tal uso; los atributos que se mencionan tienen que ver con: (León, 2010)

- **Tiempo de aprendizaje:** está relacionado con cuán rápido y fácil los usuarios pueden empezar a hacer trabajos productivos con un sistema el cual es nuevo para ellos.

- **Eficiencia en el uso:** se refiere al número de tareas por unidad de tiempo que un usuario puede desempeñar usando el sistema.
- **Confiabilidad en el uso:** como los errores hechos durante el uso del sistema y cuán fácil es recuperarse de ellos.
- **Satisfacción:** es el atributo más elusivo, y es completamente dependiente de una opinión subjetiva de los usuarios.

En los años 80 y 90 asumieron que la usabilidad era primariamente una propiedad de presentación de la información. Por lo tanto, separar la presentación y la aplicación hacía fácil modificar tal presentación después de una prueba de usuario. Sin embargo, tal supuesto probó que no era suficiente para alcanzar la usabilidad en los sistemas. (Casanovas, 2006)

De esta forma la creencia que existía a finales de los noventas es que la usabilidad se ve grandemente afectada por la funcionalidad del sistema. Aún si la presentación del sistema y la funcionalidad son diseñadas extremadamente bien, la usabilidad del sistema puede estar comprometida en gran medida si la arquitectura que se encuentra por debajo no soporta aspectos de usabilidad. Así, mientras más tarde sea detectado un problema en el ciclo de vida, más caro será repararlo, por lo que se evidencia la gran relación que existe entra la usabilidad y la Arquitectura del Software. (Casanovas, 2006)

Tradicionalmente se defiende la idea a nivel internacional de medir y mejorar la usabilidad una vez finalizado el software. Las investigaciones a profundidad sobre el tema han demostrado que esto no es cierto, algunos proyectos han fracasado por no tener en cuenta la usabilidad desde la creación del sistema. (Casanovas, 2006)

Estudios de Pressman y Landauer plantea que una parte importante de los costes de mantenimiento de los sistemas de software se gastan en resolver problemas de usabilidad. Estos altos costos se deben porque algunos de los requisitos de usabilidad no se especificaron hasta que el software fue implementado o instalado. (Pressman, 1992), (Landauer, 1995)

Además entre otras causas se encuentran:

- Los requisitos de usabilidad son a menudo pobremente especificados.
- Las técnicas de la ingeniería presentan una capacidad limitada para la captura de los requisitos de usabilidad
- Los requisitos de usabilidad pueden cambiar durante el desarrollo del producto.

Muchos de los cambios necesarios para asegurar la usabilidad de un sistema, no pueden ser fácilmente soportados por la arquitectura de software. Algunos de los cambios que pueden

mejorar la usabilidad requieren un alto grado de modificación. Por ejemplo, los cambios que se relacionan con las interacciones que tienen lugar entre el sistema y el usuario, tales como deshacer los cambios a una determinada función o en todo el sistema. (Pressman, 1992), (Landauer, 1995)

A nivel internacional son muchas las empresas que han tenido problemas con la usabilidad de sus productos, en el caso de la empresa *Creative Good* tuvieron malas experiencias con el cliente, el cual tuvo efecto devastador en los ingresos, el 39 % de los compradores fracasó en sus intentos de comprar porque los sitios eran muy difíciles de navegar. Además, el 56 % de los intentos de búsqueda fracasaron. La empresa *Jared Spo* en un estudio de 15 grandes sitios de usuarios comerciales sólo se podía encontrar la información requerida en el 42 % de ellos. (UsabilityNet, 2003)

La Universidad de las Ciencias Informáticas (UCI), no está ajena a estos problemas con la usabilidad de sus productos. En una entrevista realizada a 10 centros de la universidad y en específico al Centro de Informatización de la Gestión de Entidades (CEIGE), con el objetivo de diagnosticar el estado actual de la usabilidad arrojaron los siguientes resultados:

- Aunque a la usabilidad se le confiere una importancia alta, en varios proyectos no se especifican los requisitos no funcionales, y si lo realizan, en el caso específico de la usabilidad, la describen como las características que debe brindar la interfaz de usuario(IU), que la aplicación debe ser amigable, la facilidad de uso de sistema, la estandarización de los formularios, las etiquetas de la IU, la ubicación, guía al usuario y el acceso a los datos, sin embargo en ningún proyecto se percibe que elementos de la usabilidad impactan sobre parte funcional del sistema, además no existe una forma de medir cuan usable es el sistema, por ejemplo ¿Cómo medir la facilidad de uso o si es amigable el mismo ?
- Se percibe que la usabilidad tiene un gran impacto en la satisfacción del cliente, la calidad del producto y el éxito del proyecto y un menor impacto en la funcionalidad y la arquitectura del software, esto denota el escaso conocimiento que existe en torno a la usabilidad.
- La evaluación de la usabilidad solo en un caso se identificó que debía realizarse durante el diseño y análisis arquitectónico, el resto defendió la idea que debía medirse en el levantamiento de requisitos, las pruebas de software y en la aceptación del producto, esta idea ha fracasado, la usabilidad debe medirse en todo el proceso de desarrollo.

- Se identificó la fase de aceptación del producto la que podría originar problemas de usabilidad, demostrándose que no tenían en cuenta la usabilidad en otras partes del ciclo de vida del producto.
- Durante el proceso de captura de requisitos, se confunden los escenarios de usabilidad con requisitos funcionales del sistema, cuando se debería tener en cuenta, que los requisitos funcionales son el centro de las necesidades del cliente, mientras que los de usabilidad, giran alrededor de estos para garantizar que el usuario los realice de una forma más cómoda y sencilla.

El CEIGE obtuvo como resultado la liberación de la versión 1.0 del sistema CEDRUX por parte del Centro para la Excelencia en el Desarrollo de Proyectos Tecnológicos (CALISOFT). A partir de julio del pasado año se comenzó con las pruebas de aceptación con el cliente en la empresa consultora y certificadora de software CONABANA.SA. Con el objetivo de diagnosticar el grado de usabilidad del producto, se le aplicó una prueba y un test de usabilidad a la especialista principal de consultoría de dicha empresa, siguiendo el procedimiento para la evaluación de la usabilidad de sistema de gestión que propone CALISOFT, mediante una lista de chequeo con 99 indicadores, de los cuales 38 fueron evaluado de positivos, que representa un 38 %, arrojando un resultado de un grado de usabilidad insatisfactorio del producto según el estándar ISO 14598.

Esto se debe a que en el proceso de desarrollo del sistema CEDRUX se priorizaron otros atributos de calidad como fue el de funcionalidad restándole importancia a la usabilidad, por lo que no se aseguró, ni se evaluó en todas las fases del desarrollo del sistema.

Por lo descrito anteriormente se identificó el siguiente **problema científico**: La deficiente toma de decisiones con respecto a la priorización de los atributos de calidad trae consigo bajo soporte de la usabilidad en la arquitectura de los productos de software del CEIGE.

Enmarcándose el **objeto de estudio**: La usabilidad en la arquitectura en los productos de software.

Se establece como **objetivo general**: Desarrollar un procedimiento para el aseguramiento y evaluación de la usabilidad basado en patrones arquitectónicamente sensibles que contribuya al soporte de la usabilidad en la arquitectura de los productos del CEIGE.

Del objetivo general se desglosan los siguientes **objetivos específicos**:

1. Definir el marco teórico de los métodos, procedimientos, modelos y normas de la usabilidad a nivel nacional e internacional.

2. Diseñar un procedimiento para el aseguramiento y evaluación de la usabilidad que contribuya al soporte de la usabilidad en la arquitectura de los productos de software del CEIGE.

3. Validar el procedimiento diseñado a través de del método de evaluación SALUTA.

Como **campo de acción**: La usabilidad en la arquitectura de los sistemas de gestión del CEIGE.

**Tipo de investigación**

La investigación es descriptiva, se especifican las propiedades más importantes de la usabilidad y se estima el impacto de la misma en el proceso de desarrollo de software, en específico en la arquitectura.

Se plantea la siguiente **hipótesis**: Si se desarrolla un procedimiento para el aseguramiento y evaluación de la usabilidad basado en patrones arquitectónicamente sensibles se logrará el soporte de la usabilidad en la arquitectura de los productos de software del CEIGE.

**Operacionalización de las variables dependientes e independientes.**

Variable independiente: Procedimiento para el aseguramiento y evaluación de la usabilidad basado en patrones arquitectónicamente sensibles

Variable dependiente: Soporte de la usabilidad en la arquitectura.

**Tabla 1: Operacionalización de las variables**

<b>Variables</b>	<b>Indicadores</b>	<b>UM</b>
Soporte de la usabilidad en la arquitectura.	Número de atributos de usabilidad identificados y priorizados	Cantidad de atributos de usabilidad priorizados para el diseño arquitectónico.
	Número de propiedades de usabilidad identificados y priorizados	Cantidad de propiedades de usabilidad que brinda el sistema.
	Números de patrones de usabilidad empleados	Cantidad de patrones de usabilidad empleados para el diseño arquitectónico.
Procedimiento para el aseguramiento y evaluación de	Fases, tareas y actividades definidas. (existencia)	Aplicabilidad del procedimiento

la usabilidad basado en patrones arquitectónicamente sensibles	Elementos conceptuales. (existencia)	
	Roles que intervienen. (existencia)	
	Técnicas y herramientas definidas (existencia)	
	Calidad de la propuesta	Capacidad de ser generalizado
		Homologación con modelos internacionales y nacionales

**Métodos y técnicas a utilizar**

Los métodos teóricos a utilizar para estudiar las características del objeto de investigación son:

**Histórico lógico:** En la investigación se realiza un estudio del estado del arte con el fin de analizar la evolución que han tenido los diferentes temas asociados a la problemática (modelos, metodologías y otros elementos relacionados con la arquitectura y la usabilidad) y lograr un mayor entendimiento de los mismos, establecer las conexiones y dependencias fundamentales entre ellos.

**Hipotético deductivo:** Se sigue este método para dar solución al problema planteado, definiéndose una hipótesis a partir de la cual, siguiendo reglas lógicas de deducción, se llega a nuevos conocimientos y predicciones que posteriormente son sometidos a verificaciones empíricas.

Los métodos empíricos utilizados para estudiar las características de los fenómenos originados o que involucran al objeto son:

**Observación:** Se hará uso de la misma durante varios momentos de la investigación, pues permite conocer de manera directa el comportamiento del objeto de estudio y como esta permite investigar el fenómeno en su manifestación externa, sin llegar a la esencia del mismo.

**Medición:** Se sigue este método con el objetivo de obtener información numérica acerca de una propiedad o cualidad del objeto, donde se comparan magnitudes medibles y conocidas. Se hará uso de este método en los resultados que brinden las distintas validaciones a la que se sometan los resultados de la investigación.

**Muestreo**

**Población y muestra**

La población a analizar está constituida por los proyectos del sistema CEDRUX para un total de 10 proyectos. La muestra se seleccionó de forma intencionada teniendo en cuenta que todos los elementos de la población tienen características muy similares, además se seleccionaron los proyectos en los cuales se pudieran obtener los datos necesarios en el período antes de aplicar la propuesta, se escogieron 7 proyectos, que representa el 70 % de la población.

**Diseño de investigación**

Para la validación de la investigación se realizará un pre-experimento, donde se tomarán varios grupos experimentales a los cuales se les aplicará el procedimiento, en este caso estarán formados por 7 proyectos: Inventario, Costo Proceso, Estructura Composición, Banco, Caja, Capital Humano y Configuración. Para la validación de la presente investigación se hará uso de procesamiento estadísticos apoyados en el test:

**Test de Wilcoxon** (Comparación de dos muestras apareadas)

Así como la herramienta informática SPSS para la ejecución del test.

Para la ejecución del pre-experimento se seguirá el siguiente diseño de la figura 1.

G1	O <sub>1</sub>	X	O <sub>2</sub>
G2	O <sub>1</sub>	X	O <sub>2</sub>
G3	O <sub>1</sub>	X	O <sub>2</sub>
⋮	⋮	⋮	⋮
G7	O <sub>1</sub>	X	O <sub>2</sub>

**Figura 1. Diseño de las pruebas para el análisis estadístico del pre-experimento**

Simbología del diseño de las pruebas

- G1,G2,G3....G77      Grupo de participantes
- X                      Aplicación del tratamiento
- O1                      Observación inicial
- O2                      Observación final

Se realizará un estudio inicial de cómo actualmente en estos proyectos se comportan los indicadores de las variable dependiente e independiente antes de la aplicación del procedimiento y una observación final del impacto que tuvo el procedimiento en la muestra seleccionada.

**Instrumentos**

Para la obtención de los datos, es decir el comportamiento de los indicadores definidos se diseñó un instrumento de captación de datos y se aplicó en forma de encuesta a los arquitectos y analista de los proyectos antes de comenzar el pre-experimento y después, el instrumento aplicado se encuentra en el Anexo 2.

### **Aporte Práctico de la investigación.**

La propuesta realizada es efectiva para ayudar a detectar problemas de usabilidad temprano en el ciclo de desarrollo, cuando son más fáciles y menos costosos de solucionar.

Como parte de la investigación se obtienen los siguientes resultados:

- Procedimiento para el aseguramiento y evaluación de la usabilidad basado en patrones arquitectónicamente sensibles para los sistemas del CEIGE.
- Vista de presentación del expediente de arquitectura de la universidad.
- Catálogo de patrones de Interfaz de usuario.
- Listas de chequeo para evaluar el grado de usabilidad en sistema de gestión.

### **Listado de publicaciones, eventos y avales de la investigación**

1. Pupo, Iliannis. (2011) Plan para la verificación y validación de requisitos en el proyecto ERP-Cuba, Informática 2011.
2. Pupo, Iliannis. (2011) Un nuevo paradigma en los entornos de desarrollo de software: Los entornos sociales. Informática 2011.
3. Pupo, Iliannis. (2010) El aseguramiento y evaluación de la usabilidad para los sistemas de gestión del CEIGE. Fórum de Ciencia Técnica
4. Pupo, Iliannis. (2010) Constructor de aplicaciones WEB. Serie Científica UCI.
5. Pupo, Iliannis. (2010) Herramienta para el desarrollo colaborativo, una forma de gestionar el conocimiento. Informática 2011.
6. Pupo, Iliannis. (2010) Ide Caxtor. Uciencia 2010.
7. Pupo, Iliannis. (2012) Procedimiento para el aseguramiento y evaluación de la usabilidad. Uciencia 2012.
8. Pupo, Iliannis. (2012) Asegurando la usabilidad en el proceso desarrollo de software. Uciencia 2012.
9. Pupo, Iliannis. (2012) Definición de un procesos de pruebas para los productos del CEIGE. Uciencia 2012.
10. Pupo, Iliannis. (2012) Integración de técnicas de usabilidad en el proceso desarrollo del CEIGE. V Conferencia científica el patrimonio cultural de los pueblos iberoamericanos.

11. Pupo, Iliannis. (2012) La usabilidad y la arquitectura de software. XVI Fórum de Ciencia y Técnica.
12. Pupo, Iliannis. (2011) Herramienta para el desarrollo colaborativo, una forma de gestionar el conocimiento. Informática 2011

### **Estructura del trabajo:**

**Capítulo 1:** Se realiza un estudio del estado del arte y se analizan algunos temas como: Ingeniería de usabilidad, escenarios de usabilidad, arquitectura de software, modelos nacionales e internacionales relacionado con la usabilidad, entre otros temas. Se hace énfasis en el análisis de cada uno de los temas como se relaciona con la usabilidad.

**Capítulo 2:** Se describe la solución propuesta.

**Capítulo 3:** Se hace un análisis de los resultados obtenidos en la aplicación de la propuesta en los proyectos del CEIGE.

**CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION**

**1.1 Introducción**

En éste capítulo se exponen los temas fundamentales que sustentan la investigación. Se definen conceptos, importancia, técnicas y métodos de evaluación de la usabilidad. Se aborda la relación que existe entre la arquitectura del software y la usabilidad, la importancia de la misma y los patrones de usabilidad que se relacionan con estos, se analizan los diferentes modelos de la usabilidad.

**1.2 Análisis bibliométrico**

Esta sección se relaciona las bibliografías consultadas y referenciadas en la investigación:

	Últimos 5 años	Años anteriores
Libros y monografías		6
Tesis de doctorados	3	
Tesis de maestrías	4	
Artículos en Revistas referenciadas en Web of Science, SCOPUS	3	
Memorias de eventos	2	
Artículos publicados en la web	15	10
Reportes técnicos y conferencias		2

Muchos son los autores: Hackel de 1991, Nielsen de 1993, Preece de 1994, Wixon y Wilson, 1997, Shneiderman de 1998, ISO 9.126-1, etc que han estudiado la usabilidad. La mayoría de estos autores se centran en encontrar y definir el conjunto óptimo de los atributos que componen la usabilidad, el desarrollo de pautas y heurísticas para mejorar el diseño de las interfaces de usuarios. Existen varias técnicas que se puede utilizar para evaluar la usabilidad de los sistemas. Sin embargo, ninguno de estas técnicas se centra en la relación esencial de la usabilidad con la arquitectura de software.

Se evidencia que el 60 % de la bibliografía empleada en la investigación es de los últimos cinco años.

### 1.3 Diferentes enfoques del término usabilidad

La usabilidad se compone de dos tipos de atributos: Atributos cuantificables de forma objetiva, como son la eficacia o número de errores cometidos por el usuario durante la realización de una tarea, y eficiencia o tiempo empleado por el usuario para la consecución de una tarea. Atributos cuantificables de forma subjetiva, como es la satisfacción de uso, medible a través de la interrogación al usuario, y que tiene una estrecha relación con el concepto de usabilidad percibida (Hassan Montero, 2004)

#### ➤ **Shackel**

Uno de los primeros autores en el campo para reconocer la importancia de la ingeniería de usabilidad y la relatividad del concepto de usabilidad es (Shackel, 1991). Shackel define un modelo donde la aceptación del producto es el más alto concepto. La usabilidad se define como:

"la usabilidad de un sistema es la capacidad humana en términos funcionales que deban utilizarse con facilidad y eficacia por el rango especificado de usuarios, recibir una formación específica de apoyo, para cumplir con el rango especificado de tareas, dentro del rango especificado de escenarios "(Shackel, 1991).

#### ➤ **ISO 9241-11**

Las normas ISO 9241-11 relativas a los requisitos ergonómicos como el hardware y el medio ambiente, por el contrario, han sido ampliamente adoptados por la industria. Estas normas han dado lugar a las directrices para interfaces de software y la interacción basada en la investigación realizada por Macleod (Macleod, 1994). La ISO 9241-11 proporciona la definición de la usabilidad que se utiliza más a menudo en las normas ergonómicas. La usabilidad se define como: "el grado en que un producto puede ser utilizado por determinados usuarios para conseguir objetivos específicos con efectividad, que es el grado en que las metas previstas por el usuario se logran, la eficiencia, relacionada con los recursos que tienen que ser invertidos para lograr los objetivos y la satisfacción; como la medida en que el usuario encuentra el uso del producto aceptable, en un contexto de uso específico ".(Macleod, 1994)

#### ➤ **ISO 9126**

La comunidad de ingeniería de software siempre ha asociado usabilidad con el diseño de interfaz. ISO 9126 para definir usabilidad como una contribución relativamente independiente a la calidad del software asociado con el diseño y evaluación de la interfaz de usuario y la interacción. La usabilidad se define como: "un conjunto de atributos de software que realizan

el esfuerzo necesario para su uso y la evaluación individual de tal uso por una declaración explícita o implícita por un conjunto de usuarios".(Macleod, 1994)

### **1.3.1 Consideraciones sobre las diferentes definiciones de usabilidad**

Se puede concluir que las diferentes definiciones de usabilidad se solapan. Sin embargo, aunque exista una cantidad considerable de solapamiento, algunos atributos son diferentes cuando se examina en un nivel inferior. Por otro lado también hay similitudes en un nivel inferior para los atributos que parecen diferentes. Por ejemplo, el número de errores cometidos durante una tarea, o el tiempo para aprender una tarea, son indicadores medibles para el atributo facilidad de aprendizaje. Por otro lado, el número de errores cometidos durante una tarea es una indicación de los errores de Nielsen, sino también en la norma ISO 9216 es un atributo de eficiencia. Los atributos de errores y de la eficiencia se encuentran estrechamente relacionados, aunque los autores lo ponen en diferentes lugares en la jerarquía de la composición de usabilidad. Se requiere de un análisis adicional para determinar qué se mide en cada atributo.

### **1.4 Evaluación de la usabilidad**

En cada fase de desarrollo, se necesita algún tipo de evaluación del sistema, para poder identificar tan rápidamente como sea posible cuando el proceso de diseño se desvía hacia un mal camino. (Marín, 2003)

Existen tres tipos de métodos de evaluación de la usabilidad:

#### ➤ **Pruebas de usabilidad**

En la realización de las pruebas de usabilidad se requiere de los usuarios para trabajar en tareas típicas utilizando el sistema o prototipos. Las mismas se realizan al modelo del prototipo final del producto y a los atributos del producto final, incluso si aún no está listo, simplemente el modelo del prototipo se prueba. Los evaluadores utilizan los resultados para ver cómo la interfaz de usuario soporta a los usuarios que deben realizar sus tareas. Los métodos de prueba son los siguientes: medida de las prestaciones, *thinking aloud* (Pensando en voz alta), Interacción constructiva, método del conductor.

#### ➤ **Inspección**

El término inspección aplicado a la usabilidad aglutina un conjunto de métodos para evaluar la usabilidad en los que hay expertos conocidos como evaluadores que determinan el grado de usabilidad de un sistema basándose en la inspección de la interfaz del mismo.

Existen varios métodos que se enmarcan en la clasificación de evaluación por inspección. Los más importantes son: heurística, recorrido de la usabilidad, recorrido cognitivo y estándares.

### ➤ **Indagación**

El proceso de indagación trata de llegar al conocimiento discurrendo o por conjeturas y señales. En los métodos de evaluación realizados por indagación hay un gran trabajo de hablar con los usuarios y observarlos detenidamente usando el sistema en trabajo real (no para un test de usabilidad) u obteniendo respuestas a preguntas verbalmente o por escrito a los usuarios. Los principales métodos de evaluación por indagación son: Observación de campo, Grupo de Discusión Dirigido y entrevistas.

#### **1.4.1 Consideraciones sobre los métodos y técnicas de evaluación de la usabilidad**

Aunque existe una amplia variedad de técnicas de evaluación de la usabilidad, y que pueden ser utilizadas durante el diseño, no hay técnicas que pueden utilizarse durante la fase de diseño arquitectónico. Las técnicas discutidas en la Ingeniería de la usabilidad como la inspección de la usabilidad y las técnicas de pruebas de usabilidad, requieren una interfaz de usuario o de un prototipo de interfaz para su evaluación. La indagación de la usabilidad se centra en la evaluación de los sistemas puestos en prácticas. La mayoría de estas técnicas evalúan el sistema para los requisitos y especificaciones de la usabilidad que pueden realmente ser medidas para sistemas ya terminados. Tales métodos de evaluación son bastante ineficientes en el diseño de un nuevo sistema desde sus inicios. Además la mayoría de los problemas de usabilidad no solo dependen de la interfaz, también de la funcionalidad, por ejemplo, la funcionalidad de deshacer. Por lo tanto en la interfaz o el sistema, las técnicas basadas en la evaluación tal como se presenta en la ingeniería de la usabilidad no son suficientes para la propuesta de solución.

Se necesitan técnicas de evaluación centradas en evaluar la arquitectura, además de la interfaz o el sistema.

#### **1.5 Diseño enfocado a la usabilidad**

Hay dos enfoques para el diseño de la usabilidad identificados :(Keinonen, 1998).

- El enfoque orientado a procesos o diseño centrado en el usuario.
- El enfoque orientado al producto o centrado en los conocimientos de diseño.

##### **1.5.1 Diseño centrado en el usuario**

El diseño centrado en el usuario es un enfoque basado en procesos orientados hacia el diseño de la usabilidad. Están estrechamente relacionados con los principios de evaluación de la usabilidad y técnicas que se describen en la ingeniería de la usabilidad. Todo el proceso de

diseño para la usabilidad, las pruebas de usuario se denomina diseño centrado en el usuario. Uno de sus principales ventajas es que se consolidan las relaciones con los usuarios y disminuye la resistencia al cambio en las organizaciones. En el diseño centrado en el usuario, se utilizan numerosas técnicas, tales como: intercambio de ideas, análisis de tareas, la observación directa, cuestionarios, entrevistas, grupos focales, los paneles de usuario, el modelado de tareas, modelado de usuario, prototipado y etc.

### **1.5.2 Diseño orientado al producto**

El enfoque orientado al producto considera la usabilidad como un atributo del producto por nombrar ejemplos de las propiedades del producto, del sistema o las características que influyen en la usabilidad. El conocimiento de diseño consiste en una colección de propiedades y cualidades que han demostrado tener un efecto positivo en la usabilidad. Este enfoque puede ser dividido en tres categorías:

#### ➤ **Directrices de las interfaces**

Estas pautas proporcionan sugerencias y recomendaciones de componentes de bajo nivel de interfaz, por ejemplo: orientaciones y pautas para los iconos, ventanas, paneles, botones, campos, etc.

#### ➤ **Diseño - heurística y principios de usabilidad**

El diseño heurístico y los principios de usabilidad indican las propiedades y los principios que tienen un efecto positivo en usabilidad. Los principales principios se mencionan a continuación: Coherencia, tareas coincidentes, control del usuario, manejo de errores, orientación y apoyo, etc.

#### ➤ **Patrones de usabilidad**

Los patrones son formas de describir las mejores prácticas, buenos diseños, y capturar la experiencia de una manera que es posible que otros puedan reutilizar esta experiencia. Aunque los patrones vienen de desarrollo de software, puede ser utilizado para cualquier diseño incluyendo el diseño de la interfaz de usuario. Los patrones se definen como: "cada patrón es una regla de tres partes, que expresa una relación entre un determinado contexto, un problema, y una solución." (Alexander, 1977)

Un patrón de usabilidad no es lo mismo que un patrón de diseño conocido en la ingeniería de software. Los patrones de diseño se centran específicamente en los detalles de implementación y su efecto sobre un atributo particular de calidad, mientras que los patrones de usabilidad solamente expresan una relación específica con la usabilidad. Una de las cosas

## CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION

que comparten los patrones de usabilidad con los patrones de diseño es que su objetivo es capturar la experiencia de diseño en una forma que pueda volver a utilizarse de manera eficaz por los diseñadores de software a fin de mejorar la usabilidad de un software, sin tener que abordar cada problema desde el principio. Otro aspecto compartido con los patrones de diseño es que un patrón de usabilidad no debe ser la solución a un problema específico, sino que debe ser capaz de ser aplicado para resolver una serie de problemas relacionados en una serie de diferentes sistemas, de conformidad con el principio de la reutilización del software. La relación entre la arquitectura de software y la usabilidad se basa en el concepto de patrón de usabilidad arquitectónicamente sensibles. La aplicación de un patrón de usabilidad es una modificación que puede resolver un determinado problema de usabilidad en un contexto específico, pero que puede ser muy difícil de poner en práctica después, porque este patrón puede tener implicaciones arquitectónicas. Un "patrón de usabilidad arquitectónicamente sensibles" se refiere a una técnica o mecanismo que debe aplicarse al diseño de la arquitectura de un sistema de software con el fin de hacer frente a una necesidad identificada por una propiedad de la usabilidad en la fase de requisitos. (Folmer, 2005)

### **1.5.3 Consideraciones sobre los modelos de diseño enfocado en la usabilidad**

Hay técnicas que permiten recoger los requisitos que hacen que el software más usable, pero no existe un proceso explícito de que estos requisitos se traduzcan en soluciones arquitectónicas específicas. El enfoque actual para el diseño de usabilidad se refiere a las instrucciones de diseño muy detallado; a las directrices de la interfaz, como sugiere el diseño de iconos, y una gran variedad de principios usabilidad. Estos principios son muy útiles, pero son generalmente difíciles relacionar a la arquitectura de software.

La comunidad de la ingeniería usabilidad ha recogido y desarrollado varias soluciones de diseño tales como patrones de usabilidad que se pueden aplicar para mejorar la usabilidad.

La evaluación de la usabilidad o el diseño para el nivel de arquitectura consiste sólo en una lista de escenarios arquitectónicos. Es necesario que los requisitos relacionados con la usabilidad puedan ser capturados en una forma que pueda ser usado para informar al diseño arquitectónico, lo cual permite la ingeniería de usabilidad desde el principio en el proceso de análisis.

### 1.6 Calidad y Arquitectura del Software

Muchas veces se tienen que realizar cambios profundos en la funcionalidad de una aplicación después de haber detectado problemas de usabilidad. Dick Berry, en su analogía del Iceberg de la usabilidad, explica que los aspectos relacionados con la presentación sólo afectan en un 40% a la usabilidad. El 60% restante está influenciado por lo que él llama “modelo del usuario”, que está constituido por los objetivos que el usuario quiere alcanzar con sus tareas. (Casanovas, 2004)

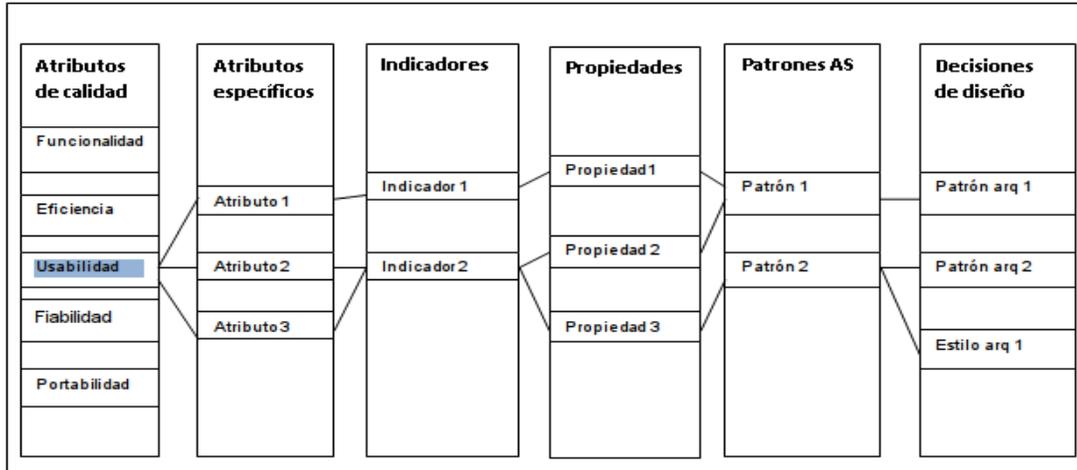
En las últimas décadas ha quedado claro que la tarea más difícil para un arquitecto de software no es sólo el diseño para la funcionalidad requerida, sino también centrarse en el diseño de los atributos específicos, tales como rendimiento, seguridad, mantenimiento o usabilidad, que contribuyen a la calidad de software (Bosch, 2002). La calidad no es algo que se puede agregar fácilmente a un sistema durante la última etapa, tiene que ser incorporado en el sistema desde el principio (Svahnberg, 2000).

Durante el diseño de la arquitectura, las decisiones de diseño se deben realizar en el principio en un proyecto ya que estas son muy costosas de revocar, sin embargo, los ingenieros de software en general, disponen de pocas técnicas para predecir los atributos de calidad en un sistema de software antes de que el mismo esté disponible (Bengtsson, 2002). Por lo tanto los sistemas se han desarrollado y la calidad siempre se mide sólo cuando el sistema se ha completado.

Diseñar significa tomar decisiones, y tomar las decisiones correctas. El diseño de la arquitectura debe incluir:

- Hacer concesiones entre los atributos de calidad: la mayoría de las decisiones de diseño tiene una influencia considerable sobre las cualidades de un sistema como la fiabilidad, el rendimiento, y la usabilidad. Estas cualidades no son independientes, una decisión de diseño puede tener un efecto positivo en un atributo, pero puede ser muy negativo para otros (Bengtsson, 2002). Algunos atributos con frecuencia entran en conflictos, por ejemplo, las decisiones que mejoren la usabilidad a menudo afectan negativamente el rendimiento.
- Mantener una integridad conceptual es decir, saber cuándo dejar de añadir complejidad o mantener la simplicidad en el diseño.

A continuación en la figura 2 se muestra la relación que existe entre la calidad y la arquitectura del software, como los atributos de calidad, los indicadores y las propiedades permiten relacionar los patrones sensibles de la arquitectura con la calidad del software.



**Figura 2: Relación entre Calidad y Arquitectura de Software**

Estas relaciones son de utilidad para los diseñadores poder analizar cuando uno o varios patrones arquitectónicamente sensibles deben ser implementados para cumplir con este ciertos requisitos de calidad.

### **1.7 Relación entre atributos, propiedades y patrones de usabilidad**

Las propiedades de usabilidad se derivan de la heurística del diseño y los principios del diseño, tales como proporcionar retroalimentación en cualquier momento o proporcionar consistencia. Están directamente relacionadas con las decisiones de diseño de software. Las propiedades de usabilidad pueden no tener una relación estrictamente uno a uno con los atributos de usabilidad. Por ejemplo un patrón asistente es una solución que consiste en la división de una tarea compleja en un conjunto de pequeñas sub-tareas. Cuando los usuarios se ven obligados a seguir el orden de las tareas, los usuarios tienen menos probabilidades de perder lo datos importantes y por lo tanto, se cometen menos errores. Un patrón asistente implementa la propiedad de usabilidad de Orientación que guía al usuario a través de la compleja tarea. La orientación puede disminuir el tiempo necesario para aprender una tarea. El tiempo para aprender una tarea es un indicador de la facilidad de aprendizaje. La facilidad de aprendizaje es un atributo de la usabilidad. (Folmer, 2005). Por lo tanto el patrón asistente tiene una relación positiva con la facilidad de aprendizaje, mientras una relación negativa con el rendimiento o eficiencia.

A continuación se ilustra estas relaciones a partir del patrón asistente (ver Figura 3).

## CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION

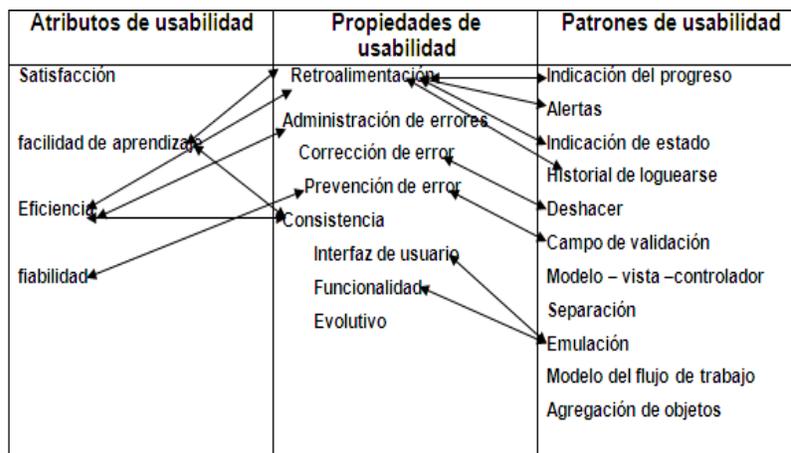


**Figura 3: Relación entre la facilidad de aprendizaje y el patrón asistente**

En resumen, se puede clasificar las propiedades de usabilidad en:

- ✓ Proporcionar información: el sistema proporciona información continua sobre el funcionamiento del mismo al usuario.
- ✓ Gestión de errores: incluye la prevención y recuperación de errores.
- ✓ Consistencia: la coherencia tanto de la interfaz de usuario y del funcionamiento del sistema.
- ✓ Orientación: orientación en línea sobre el funcionamiento del sistema.
- ✓ Minimizar la carga cognitiva: el diseño del sistema debe reconocer las limitaciones cognitivas y la memoria a corto plazo
- ✓ Accesibilidad: incluye la internacionalización y el apoyo a las personas con discapacidad.

En la figura 4 se muestra una colección de atributos, propiedades y patrones y cómo estos se vinculan para demostrar la relación entre la usabilidad y la arquitectura de software.



**Figura 4: Relación entre atributos y patrones de usabilidad**

Los atributos de usabilidad se pueden medir en un sistema finalizado, mientras que una propiedad de la usabilidad existe en el dominio del problema, y forman parte de los requisitos que impulsan el proceso de diseño arquitectónico. Un patrón de usabilidad es la relación entre los dominios de problema y la solución, que proporciona un mecanismo para cumplir con los requisitos. Después que los componentes se han diseñado e implementado, el sistema resultante puede ser sometido a pruebas con respecto a sus atributos de usabilidad.

### **1.8 La importancia de la usabilidad en la arquitectura de software**

Los sistemas siguen creciendo en tamaño y complejidad. Una arquitectura bien definida se puede utilizar como una herramienta para manejar este tamaño y complejidad. Aunque hay muchas definiciones de la arquitectura de software, una definición comúnmente aceptada de la arquitectura de software es la siguiente: "La arquitectura de software es la organización fundamental de un sistema incorporado en sus componentes, sus relaciones entre sí y con el entorno y los principios que rigen su diseño y evolución (IEEE, 1998)".

Dentro de la comunidad de ingeniería del software es comúnmente aceptado que los atributos de calidad de un sistema como la eficiencia o el rendimiento son, en gran medida, limitado por su arquitectura. Esta restricción también se aplica a la usabilidad.

Las modificaciones a nivel de diseño detallado se puede observar por cambios en el código fuente ya existente. Estos cambios tienen un impacto a pequeña escala (Bosch, 2000), que sólo influye en un módulo o un objeto y no afecta a la arquitectura de software. Las modificaciones en el nivel de diseño de la arquitectura sin embargo, tienen un impacto estructural. Si algunas partes del sistema ya se han implementado en el momento en que los cambios se hacen, es probable que la modificación afecte a muchas partes del código fuente existente, que es muy costoso modificar. Los ingenieros de software tienen como objetivo reducir al mínimo la frecuencia de los cambios con impacto a nivel arquitectónico.

La comunidad de ingeniería de software se ha dado cuenta del papel crucial que juega la arquitectura de software en el cumplimiento de los requisitos de calidad de un sistema. (Casanovas, 2004). Por lo tanto, es de gran importancia que los requisitos de calidad más esenciales para el éxito del sistema de software debería conducir el diseño, especialmente en el nivel de arquitectura de software. El diseño actual de la usabilidad no lograr este objetivo. El problema principal es que los sistemas demuestran ser inflexibles. Ciertas modificaciones en la mejora de la usabilidad que son sólo descubiertas durante la implementación o después de un diseño inicial, o porque los requisitos de usabilidad han cambiado durante el desarrollo, no

## CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION

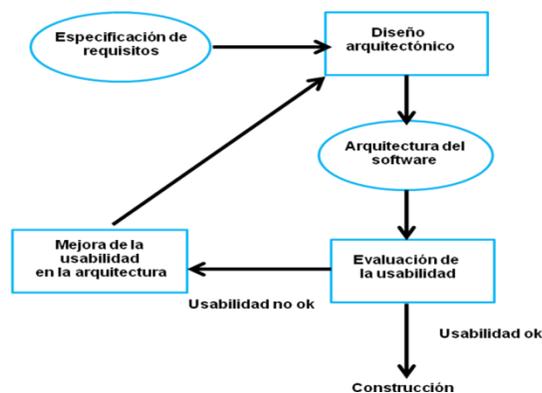
pueden ser implementados. Es demasiado costoso implementar las modificaciones debido a su impacto estructural. Por lo que ser capaz de evaluar la arquitectura durante el diseño para que soporte la usabilidad podría revelar aquellas cuestiones de usabilidad para las cuales el sistema no es flexible.

Ser capaz de diseñar y evaluar la usabilidad de forma iterativa a nivel arquitectónico, como se muestra en la Figura 5, mejora la usabilidad de los sistemas, no sólo durante el diseño, sino también después de la implementación.

Cuanto más tarde se detectan los problemas, más cuesta arreglarlos. Si al diseñar una interfaz, se quieren crear interacciones y diálogos en el entorno tecnológico, como que el usuario pueda visualizar el progreso de sus peticiones, que pueda deshacer acciones, que pueda disponer de un entorno multilingüe, cancelar una operación que lleva mucho tiempo en espera, y reutilizar información que ha introducido anteriormente, esto no va a ser posible. La respuesta es muy sencilla, una vez hecho el diseño de la arquitectura del software, no se pueden realizar estos cambios a nivel de interfaz para mejorar la usabilidad, porque esto implicaría la modificación de casi el sistema completo.

Si se analizan los escenarios de interacción que no pudieron ser cambiados, a simple vista se puede ver que la causa de que no se puedan implementar es que hay que tener en cuenta al usuario desde el inicio del diseño del sistema, es decir, en la Arquitectura del Software. (Casanovas, 2004)

Para lograr este enfoque de diseño, que es un enfoque diferente a la ingeniería de la usabilidad es necesario que se base en un enfoque de diseño general de los atributos de calidad como se muestra en la figura 5.



**Figura 5: Método de evaluación del diseño de la Arquitectura de Software.**

Este diseño se evalúa de nuevo y el mismo proceso se repite, si es necesario, hasta que todos los requisitos funcionales se hayan cumplido a la medida de lo posible. En general, algunos

concesiones son necesarios con respecto al conflicto de los requisitos no funcionales. El proceso de diseño depende de dos requisitos:

- Se requiere para determinar cuando el proceso de diseño de software ha finalizado. Por lo tanto, las técnicas de evaluación son necesarias para proporcionar información cuantitativa o cualitativa, para determinar si la arquitectura cumple con los requisitos no funcionales.
- Desarrollo o identificación de las decisiones de diseño arquitectónico para mejorar la usabilidad.

### **1.9 Consideraciones sobre los modelos y normas de usabilidad**

#### **1.9.1 Modelos nacionales**

En la universidad se han propuesto varios procedimientos para la evaluación de la usabilidad entre los que se encuentran:

- “Propuesta de Modelo de Evaluación de Usabilidad en las Aplicaciones Web Educativas en la UCI”
- “Procedimiento para la Evaluación de la Usabilidad en los Software de Gestión sobre Plataforma Web en la Facultad 2”

Como principal desventajas de estas propuestas es que no aseguran la usabilidad en todo el proceso de desarrollo, y si lo realizan solo se basan en las técnicas y principios de la ingeniería de la usabilidad, dejando fuera el impacto de este atributo en la arquitectura del software.

En el caso del Centro para la Excelencia en el Desarrollo de Proyectos Tecnológicos (CALISOFT), el cual se creó con el objetivo de verificar el cumplimiento de las políticas de calidad. El Departamento de Pruebas de Software (DPS) propone un procedimiento donde las etapas del ciclo de vida en que se puede hacer una evaluación de la usabilidad, son: después del diseño cuando se disponga de los prototipos no funcionales, y durante el desarrollo según los hitos que decida el equipo de proyecto, a través de una lista de chequeo que mide los atributos de usabilidad mediante criterios de experto.

Este procedimiento no se ajusta completamente a los productos del CEIGE, pues cuando una no conformidad de usabilidad detectada en la evaluación requiere para su solución de cambios significativos en la arquitectura y funcionalidades del sistema, no se podría llevar a cabo o bien sería ineficiente porque afectaría los plazos y costo del proyecto. Además la evaluación de la usabilidad no debe ser una simple prueba para dar un resultado, sino debe ser parte inherente del proceso de desarrollo.

### 1.9.2 Modelos y normas internacionales

#### 1.9.2.1 El Modelo de Proceso de la Ingeniería de la Usabilidad.

Propone un enfoque donde se aplican aspectos del Diseño Centrado en el Usuario (DCU) a las fases de desarrollo de software, con el fin de mejorar los productos a través de flexibilidad, entendimiento y fácil manejo por los usuarios pero desde las etapas de desarrollo de software y no desde el producto final como se ha venido haciendo; de esta manera el modelo agrega 3 componentes esenciales o pilares básicos el usuario que interactúa en cada etapa de desarrollo: análisis, diseño (prototipo) y la evaluación del producto. A pesar de ser un modelo que toma en cuenta aspectos de desarrollo de software y de diseño centrado en el usuario, su enfoque radica en el uso de la usabilidad dentro de las fases del proceso de desarrollo de software. (MPlu+a, 2007)

Wixon y Jones (1995) en un estudio de la aplicación de la ingeniería de usabilidad en un producto logró que los ingresos fueran de un 80% superior a la de la primera versión desarrollada sin ingeniería de la usabilidad, y 60% por encima de las expectativas del proyecto (Wixon, 1995)

#### 1.9.2.2 Patrones de Usabilidad: Arquitectura de software que soporta la usabilidad (STATUS) y Escenarios de Software de Usabilidad (USE)

Este modelo presenta una aproximación para mejorar la usabilidad de un sistema software aplicando un proceso específico de diseño para usabilidad. Se relaciona con el desarrollo de técnicas y procedimientos para mejorar la usabilidad, desde momentos tempranos del desarrollo de software. Esta aproximación difiere de la idea tradicional de medir y mejorar la usabilidad una vez finalizado el sistema. (Sánchez-Segura, 2008)

#### 1.9.2.3 Escenario Nivel arquitectura basada en análisis de usabilidad (SALUTA)

Es el primer método desarrollado para evaluar arquitecturas desde la perspectiva de la usabilidad del sistema, siendo el resultado de los estudios de Folmer y Gulp. Es un modelo de evaluación que ayuda a los arquitectos de software en el diseño de una arquitectura que soporte la usabilidad, basado en la evaluación de escenario de uso. Un escenario de uso se define como "una interacción (tareas) entre los usuarios y el sistema en un contexto de uso específico". Un escenario de uso no especifica nada acerca de la usabilidad que requiere el sistema. Con el fin lograrlo, el escenario de uso se relaciona con cuatro atributos de la usabilidad (satisfacción, eficiencia, facilidad de aprendizaje y fiabilidad) (Folmer, 2005)

#### 1.9.2.4 ISO 13407: Human-centred design processes for interactive systems

## CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION

El estándar ISO 13407: *“Human-centred design processes for interactive systems”* constituye un marco que sirve de guía para conseguir el desarrollo de sistemas interactivos usables incorporando el DCU durante el ciclo de vida del desarrollo. El estándar describe las siguientes cuatro actividades que se necesitan: (1) Entender y especificar el contexto de uso, (2) Especificar los requisitos de los usuarios y organizativos, (3) Diálogo simple y natural y (4) Producción de soluciones de diseño. (Bevan, 2007)

### 1.9.3 Conclusiones de los modelos y normas de usabilidad a nivel internacional

**Tabla 2. Comparación cualitativa de los diferentes Modelos de Usabilidad**

Crterios	Ingeniería de la Usabilidad	Patrones de Usabilidad(STAT US y USE)	ISO 13407	SALUTA
Ámbito de Aplicación	Genérico. Puede ser aplicado a cualquier tipo de proceso de desarrollo de software.	Se centra en la etapa de diseño del producto.	Genérico. Se aplica a cualquier ciclo de vida y a la construcción de sistemas iterativos.	Una vez especificado la arquitectura, pero antes de implementarla
Integrable al Proceso de Desarrollo	Si. Se acopla en las diferentes Etapas genéricas del ciclo de vida.	No. Su integración se basa solo en la fase de diseño.	No. Su aplicación se realiza en el diseño y depende del ambiente y estado del proceso de diseño.	No. Su integración se basa solo en la fase de diseño
Procesos	Estructura Propia. Cuenta con características y actividades definidas para	No se definen. Cuenta con una serie de Escenarios y patrones que se aplican en el diseño.	No se definen. Describe unos principios básicos sin estipular métodos específicos	Define una serie de pasos para la evaluación de la arquitectura para que soporte a la usabilidad.

**CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION**

	cualquier proceso de desarrollo software.			
Validación	Los proyectos cuentan con “Métricas Predictivas de la Usabilidad” para su ponderación cualitativa.	No cuenta con un modelo que evalúe el nivel de usabilidad del diseño gracias a la utilización de los patrones.	No define modelo de Validación.	Evalúa los escenarios de uso, relacionando los mismo mediante cuatros atributos de usabilidad.
Objetivo	Se enfoca en desarrollar sistemas iterativos teniendo en cuenta la Ingeniería de la Usabilidad	Mejorara la usabilidad de un Software aplicando un proceso específico de diseño para la Usabilidad.	Desarrolla sistemas iterativos a través de la incorporación del DCU en el proceso de desarrollo Software.	Mejorar la usabilidad realizando un análisis de la arquitectura.
Representación	Plana e Iterativa. Cuenta con cuatro fases que iteran después de un prototipado y su evaluación.	Mejoramiento Continuo de la Arquitectura hasta que cumpla con criterios de usabilidad y posteriormente es plana hasta su entrega.	Plano e iterativo. Consta de cinco fases iterativas hasta alcanzar los resultados esperados	Cuenta de cuatros fases iterativas hasta alcanzar los resultados esperados.

La ingeniería de usabilidad cuenta con varias fases, relación que muestra la integración de la misma con la ingeniería de software, puesto que aparecen las cuatro fases principales del

## CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION

modelo de proceso clásico: análisis, diseño, implementación e instalación. Una de las primeras aproximaciones hacia la ingeniería de usabilidad considera a la misma como una característica principalmente de la presentación de la información, o sea la interfaz de usuario. Si la arquitectura usada separa la interfaz de usuario de la aplicación que la utiliza, como la arquitectura modelo-vista-controlador, la usabilidad puede ser asegurada. Si la usabilidad necesita ser mejorada, la interfaz se puede cambiar y aplicar fácilmente los cambios después de las pruebas de usuario, y estos no afectan a la funcionalidad de la aplicación. Este enfoque es considerado ingenuo hoy en día por los expertos. La mayoría de problemas de usabilidad no dependen de la interfaz, pero sí de la funcionalidad, por ejemplo, la función "deshacer". A los expertos que apoyan este enfoque se les llama la comunidad de ingeniería de la interfaz. La comunidad de ingeniería de interfaz trata la usabilidad a un nivel de diseño detallado, como resultado se han obtenido estándares de interfaz de usuarios, línea base de diseño, maquetación de objetos, colores de interfaz y la semiología de los botones, entre otros. (Folmer, 2005). Es decir en la fase del diseño del sistema solo cubre parte del análisis del sistema vinculado al modelo conceptual y a los prototipos de Interfaz de usuario, dejando fuera el diseño de clases, componentes y la arquitectura de software.

En las investigaciones realizadas se comprobó que la usabilidad de un sistema debe ser medida desde la creación del mismo, es decir desde el momento en que se está desarrollando la arquitectura del software, son varios los modelos que abordan esta relación, como son: STATUS, USE y SALUTA. Estos enfoques argumentan esta relación, por el vínculo que existe entre atributos, propiedades y patrones de usabilidad, los dos primeros modelos proponen escenarios arquitectónicamente sensibles que deben ser implementados para que soporte a la usabilidad. Estos modelos tienen como desventajas de forma general que solo se integran en la fase de diseño y no tienen en cuenta el análisis de los prototipos de Interfaz de usuario.

SALUTA propone una serie de pasos que deben ser aplicados una vez especificado la arquitectura para la evaluación de los escenarios de uso, este método no resuelve el problema de la especificación tardía de los requisitos de usabilidad, y necesita de una buena documentación de la fase de análisis, como son los requisitos de usabilidad para poder ser aplicado. Es decir que a menudo son débiles la especificación de los requisitos, lo que hace difícil crear un perfil de uso representativo.

En SALUTA los escenarios de uso se utilizan para evaluar una arquitectura de software una vez especificada. Un mejor enfoque sería el diseño de la arquitectura basada en el perfil de uso que

## CAPITULO 1. FUNDAMENTO TEORICO DE LA INVESTIGACION

propone la Ingeniería de la usabilidad, es decir tener en cuenta estos aspectos en el momento del diseño de la arquitectura.

Es notable la creciente preocupación sobre el papel que puede jugar la usabilidad para el éxito de un producto, son diferentes los modelos y enfoques, los cuales se encuentran dispersos, no existe un procedimiento que integre dichos modelos. No hay integración de las técnicas de Ingeniería de Software con las técnicas IPO (Interacción-Persona-Ordenador) en el diseño arquitectónico. Debido a que el diseño de interfaces a menudo es aplazado hasta las últimas etapas de desarrollo corre el riesgo de que muchos supuestos puedan ser incorporados en el diseño de la arquitectura que, sin saberlo puede afectar el diseño de interfaces, y viceversa. La arquitectura del software es vista como un producto intermedio en el proceso de desarrollo, pero su potencial con respecto a la evaluación de la calidad no se aprovecha plenamente.

La problemática es esencial, entender la relación entre la arquitectura de software y la usabilidad para asegurar que el sistema últimamente la alcance, de tal forma que es importante que tanto ingenieros de software como especialistas en usabilidad entiendan dicha relación, por lo que se hace evidente la necesidad de un procedimiento que integre ambos enfoques.

### 1.10 Conclusiones parciales

#### Se concluye que:

- La revisión de los principales métodos, modelos y procedimiento para la evaluación de la usabilidad, demostró la existencia de dificultades en la adecuación de estos para su aplicación en los proyectos del sistema CEDRUX por diversos factores explicados en el epígrafe 1.9.3.
- El enfoque actual para el diseño de usabilidad se refiere a las instrucciones de diseño muy detallado; a las directrices de la interfaces, como sugiere el diseño de iconos, etc., o te ofrece una gran variedad de principios usabilidad. Estos principios son muy útiles, pero son generalmente difíciles relacionar con la arquitectura de software.
- Los elementos analizados en las secciones anteriores apuntan la necesidad de definición y especificación de una procedimiento que asegure y evalúe la usabilidad en todas las fases de desarrollo de los productos del CEIGE.

### CAPITULO 2: DESCRIPCION DEL PROCEDIMIENTO

#### 2. Introducción

En el presente capítulo se darán a conocer los elementos que conforman la propuesta, sus características, implicaciones y restricciones. Se abordan los elementos distintivos del mismo, los pasos para su aplicación, la conceptualización de cada uno de los elementos que se utilizan, así como los roles involucrados en la puesta en práctica de la propuesta.

#### 2.1 Objetivos del procedimiento.

El procedimiento para el aseguramiento y la evaluación de la usabilidad de los sistemas de gestión del CEIGE contará de 3 fases, propuesta por la Ingeniería de la Usabilidad, además estará basado en los modelos que proponen los proyectos STATUS, USE y en el método de evaluación SALUTA.

Con el procedimiento se persiguen los siguientes objetivos:

1. Garantizar la usabilidad en todas las fases del desarrollo de los productos del CEIGE.
2. Lograr un mejor uso del sistema a través de las decisiones de diseño incorporadas en la arquitectura de software.

#### 2.2 Estructura del procedimiento.

La usabilidad implica muchos aspectos distintos que dependen del tipo de sistema a construir y de las características particulares de los usuarios que utilizarán el sistema. Por esta razón la usabilidad no puede definirse como un simple atributo de un sistema, sino que deben existir un conjunto de técnicas que permitan atacar al ciclo de desarrollo de software desde sus diferentes fases para lograr incluir los atributos de usabilidad en el producto.

La estructura del procedimiento es la siguiente: está compuesto por las siguientes fases:

- **Análisis:** Se caracteriza al usuario y al producto, tiene entre sus principales salidas las propiedades, atributos y beneficios de usabilidad.
- **Diseño:** Se determina los patrones y escenarios a tener en cuenta en el diseño arquitectónico, tiene entre sus principales salidas el grado de usabilidad de los prototipos de IU y el soporte de la usabilidad en la arquitectura.
- **Lanzamiento:** Se evalúa el grado de usabilidad con el usuario una vez implementado el sistema.

### **CAPITULO 3: ANALISIS DE LOS RESULTADOS**

Cada fase la integran actividades con los artefactos de entrada o de salida, las herramientas y técnicas empleadas, así como los roles que participan en cada actividad.

Además el procedimiento se integra a las fases Análisis, Diseño y Pruebas internas del modelo de desarrollo del CEIGE, mediante actividades de aseguramiento y evaluación de la usabilidad como se muestra en la figura 6. No se integra en la fase de implementación porque esta corresponde exactamente con la que se describe en la Ingeniería del Software clásica, y no es objetivo de la investigación el desarrollo de esta fase, puesto que la Ingeniería de la Usabilidad no trata de cómo programar un producto interactivo, sino la metodología para conseguir un producto usable.

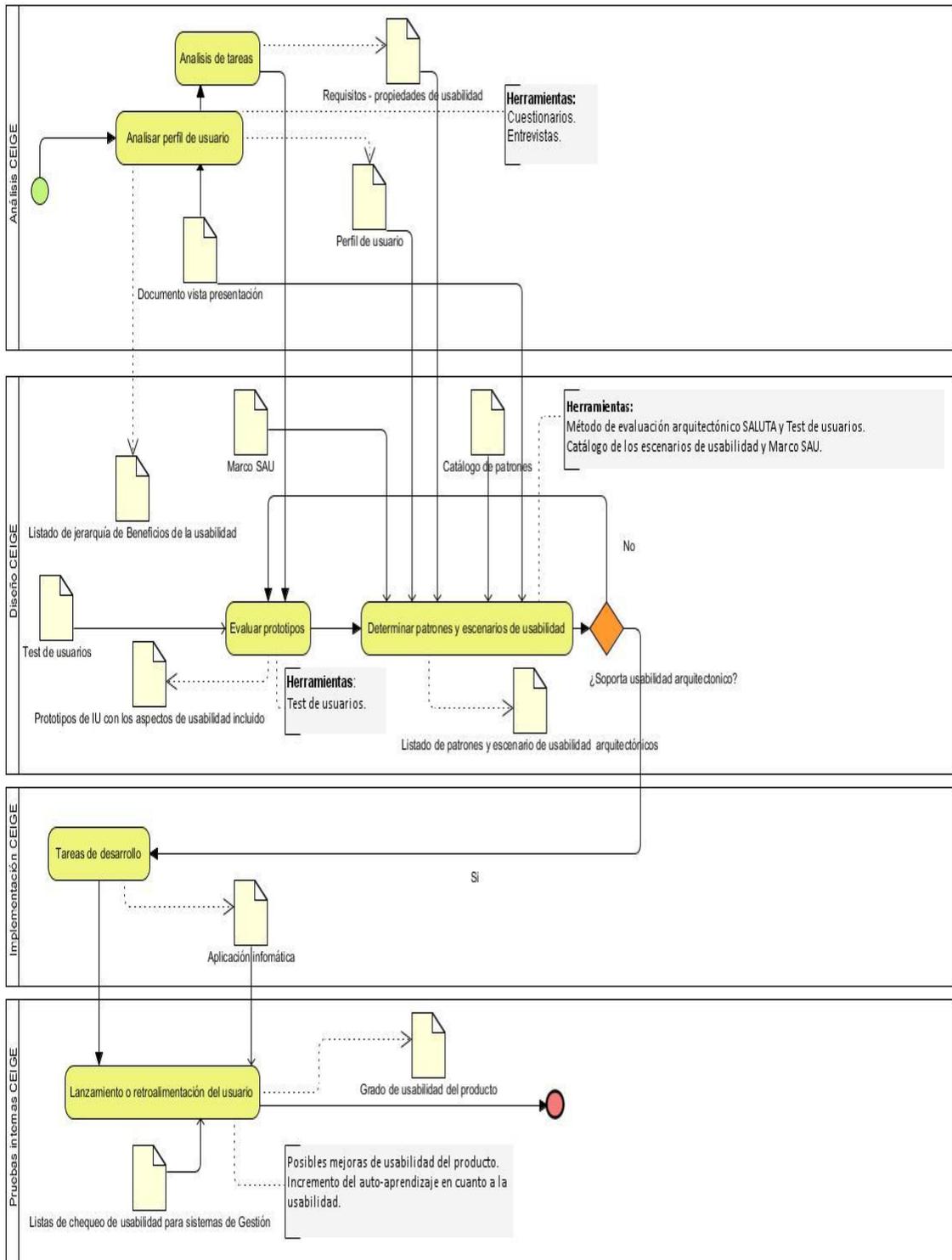


Figura 6. Actividades del procedimiento integradas a fases del modelo de desarrollo del CEIGE.

### 2.3 Descripción de las fases del procedimiento

#### 2.3.1 Fase de análisis

Se determina la audiencia y las plataformas destinos, las metas de los usuarios, las necesidades de los usuarios, los atributos de usabilidad, las propiedades o requisitos de usabilidad. Se determinan, enuncian y clasifican todas las características, capacidades, restricciones que debe cumplir el sistema, y se determinan los beneficios de usabilidad que debe brindar el sistema.

Para definir el contexto de uso del sistema hay que determinar los siguientes aspectos:

- Características de los usuarios que van a usar el sistema
- Las tareas que van a realizar estos usuarios en el sistema
- El entorno donde los usuarios realizarán las tareas
- Los atributos de usabilidad
- Caracterización de producto desde la perspectiva del usuario

Para obtener la anterior información se tiene que definir primero un documento (Ver anexo 3) que será uno de los artefactos generados por el procedimiento: la vista de presentación del expediente de arquitectura de la universidad, la misma fue creada a partir de la vista de presentación establecida en la Guía para la construcción de arquitecturas de software de la universidad en el 2010. (Mejora, 2010)

En cuanto a las especificaciones de los requisitos de la usabilidad, en el análisis los requerimientos no funcionales, tales como usabilidad o el rendimiento entre otros son débilmente especificados. Tradicionalmente los requisitos de usabilidad se han especificado de forma que estos pueden estar verificados por un sistema implementado. La especificación debe ser expresada en términos del dominio de solución para ser entrada en el diseño de la arquitectura.

#### Responsable(s):

- **Analistas:** Es el responsable de realizar el análisis del perfil de usuario y el análisis de tareas.
- **Arquitecto:** Es el responsable de la priorización de los beneficios de usabilidad según el impacto arquitectónico.

#### 2.3.1.1 Actividades de la fase de análisis

##### 2.3.1.1.1 Actividad: Análisis del perfil del usuario

Se obtiene el perfil de los usuarios potenciales. Una vez obtenidos los datos se realiza un análisis con el objetivo de describir los factores más relevantes de impacto sobre la usabilidad

del producto o servicio (por ejemplo, el tipo de uso, la cantidad de horas dedicadas al uso de sistemas informáticos y el nivel de experiencia previa). Este proceso, por tanto, aporta un conjunto de datos clave al análisis de tareas.

La evaluación del perfil del usuario se realiza a partir conocimiento del usuario. Este conocimiento puede dividirse en dos categorías.

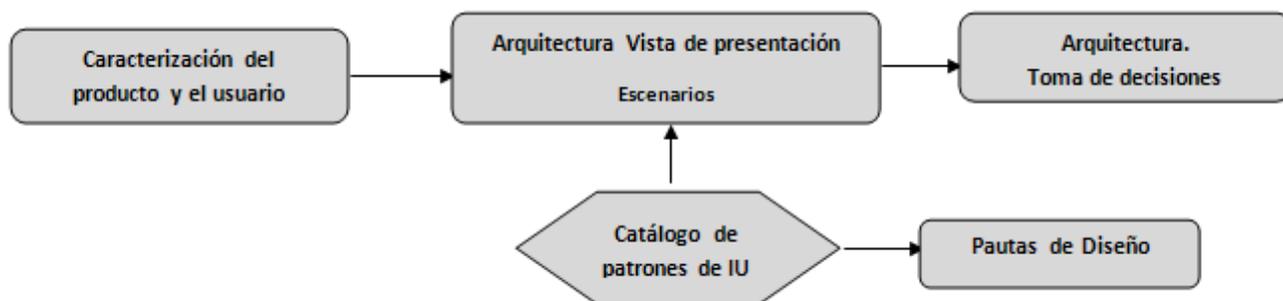
- **Conocimiento del dominio:** consiste en saber cuánto sabe el usuario sobre el dominio.
- **Conocimiento del sistema:** conocimiento, experiencia o estudios de alguna herramienta, cómo opera y cómo usarlo.
- **Evaluación de la Interacción,** consiste en obtener los patrones de uso del sistema por parte de los usuarios, frecuencia con la que el usuario emplea el sistema.
- **Complejidad:** si las tareas a realizar son muy complejas.
- **Foco de control:** quién dirige la interacción usuario-proceso.
- **Información:** determina dónde se origina la información y cómo fluye entre el sistema y el usuario o viceversa. Se va a estudiar el origen de la información, la dirección de la información.
- **Características especiales del usuario:** Nivel educacional, habilidades o discapacidades, edad, región o área geográfica e idioma.
- **Atributos de usabilidad:** además se deben tener en cuenta los criterios de usabilidad que son más relevantes para el sistema(tiempo de aprendizaje, eficiencia en el uso, confiabilidad y satisfacción)
- **Se caracteriza el producto teniendo en cuenta diferentes patrones de soluciones,** la descripción de los escenarios formará parte de la vista de presentación del expediente de arquitectura definido en la universidad: (Ver anexo7)

- **Escenario Aspecto genérico UCI:** Los elementos básicos que caracterizan a los sistemas producidos por la red de centros de la universidad.
- **Escenario Aspecto de editor:** Aplicación que se utiliza para editar archivos informáticos.
- **Escenario Aspecto de WEB:** conjunto de páginas Web relacionadas a través de una estructura de navegación, con el objetivo de hacer pública y accesible una cierta cantidad de información.

- **Escenario Aspecto de gestor de procesos:** Sus acciones fundamentales no se aplican a ficheros sino a registros y tablas de una base de datos. Aplicación que gestionan la información, automatiza procesos. Su propósito es facilitar el flujo de información entre actividades o procesos.
- **Escenario Presentación de reportes:** Es una salida de información sobre datos persistidos en alguna estructura de datos que se construye dinámicamente.

Uno de los escenarios de la vista de presentación es el **Aspecto genérico UCI**, este se elaboró en conjunto con la Dirección de Comunicación Visual de la Universidad, el mismo tiene como objetivo estandarizar la imagen gráfica de la universidad (iconografía, temas visuales, logos), y está muy relacionado con los manuales de entidad de los productos, al establecer que cada sistema debe tener un identificador visual del producto relacionada con los tipos de marcas y logos que se están creando actualmente en la universidad por parte de la Dirección de Comunicación Visual, además la vista de presentación se integra con un catálogo de patrones de IU que establecen las pautas de diseño, con estos elementos la vista de presentación permitirá mejorar la imagen gráfica de la universidad y posibilitará la toma de decisiones arquitectónicas.

En la figura 7 se muestra los elementos que forman parte de la vista de presentación.



**Figura 7. Integración de la vista de presentación con patrones de IU.**

Dentro del análisis del perfil del usuario también es necesario **determinar la jerarquía de beneficios de usabilidad** a partir de los atributos de usabilidad que definidos por el usuario son más relevantes para el sistema, se determinan los beneficios de usabilidad, a través de la tabla 3 que muestra la relación entre los atributos de usabilidad identificados en el cuestionario de la vista de presentación y los beneficios de usabilidad.

**Tabla 3: Relación entre los beneficios de usabilidad y sus atributos**

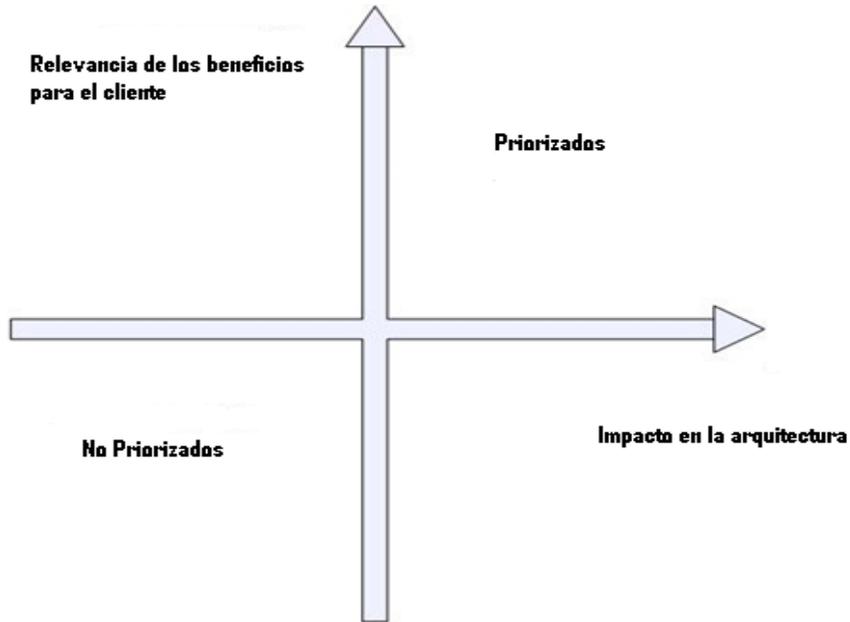
Atributo de	Beneficio de Usabilidad	Descripción
-------------	-------------------------	-------------

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

usabilidad		
<b>Eficiencia</b> <b>Confiabilidad</b>	Aumento del rendimiento frecuente.	En el sistema se realiza una misma tarea de forma frecuente, los pasos para ejecutar la misma deben ser fáciles de recordar.
	Maximizar el rendimiento sin errores para realizar tareas habituales.	Los sistemas son diseñados para maximizar el tiempo de rendimiento sin errores, reduciendo así el tiempo para realizar tareas habituales y aumentar la eficacia individual.
	Reducción del impacto de los errores de los usuarios en tareas habituales.	El sistema es diseñado para que los pasos que el usuario tenga que seguir para la ejecución de tareas tengan la menor dificultad posible y guíen al usuario a través de métodos de recuperación antes fallos.
<b>Facilidad de aprendizaje</b>	Mejor rendimiento en tareas no muy frecuentes	El sistema es diseñado para asistir al usuario cuando se esté realizando tareas que no realiza frecuentemente o que son nuevas para él.
	Soporte a la resolución de equivocaciones del usuario.	El sistema es diseñado para que en la realización de tareas nuevas o no muy frecuentes, ante equivocaciones por la inexperiencia del usuario, este pueda regresar a estados anteriores.
	Facilita el aprendizaje	El sistema debe ser diseñado para que de una forma rápida y fácil el usuario pueda empezar a hacer tareas en un sistema que sea nuevo para ellos.
	Reducción del impacto de las equivocaciones del usuario por falta de conocimiento en el sistema.	El sistema debe ser diseñado para reducir el número de equivocaciones que pueda realizar un usuario en la realización de una tarea.

A cada beneficio de usabilidad identificado se le asigna un par (x, y) representando, "x" que pertenece [1...10], "y" que pertenece [1...10] dónde "x" representa la relevancia del beneficio para el cliente y la "y" representa el impacto para la arquitectura que tiene el beneficio de usabilidad.

Luego se ubican los beneficios en la siguiente gráfica (la figura 8 mostrada más adelante), según los pares ordenados asignados a cada beneficio. El centro de la gráfica es el par (5,5). Se debe priorizar los beneficios a partir de su ubicación en la gráfica. Los beneficios en el primer cuadrante son los más priorizados, los ubicados en los cuadrantes 2 y 4 tienen un segundo nivel de prioridad mientras que los beneficios ubicados en el tercer cuadrante no deben tener prioridad para la arquitectura. Los beneficios priorizados de los cuadrantes 1,2 o 4 deben ser considerados para la selección de los escenarios arquitectónicamente usables.



**Figura 8: Priorización de los beneficios de usabilidad**

Esta experiencia puede ser muy beneficiosa para el diseño inicial y rediseño de una aplicación debido a que ya se conocen los gustos y necesidades del usuario; se reduce el coste de mantenimiento debido que los usuarios manejan el sistema, lo aceptan y lo usan con más facilidad.

### **Herramientas:**

- ✓ **Cuestionarios:** El cuestionario es menos flexible que la entrevista, pero puede llegar a un grupo más numeroso y se puede analizar con más rigor. Como también se han apuntado en el apartado de las entrevistas, suelen complementarse muy bien.
- ✓ **Listado de jerarquía de Beneficios de la usabilidad:** se presenta en el marco de relación de la tabla 3.

### **2.3.1.1.2 Actividad: Análisis de tareas**

En este proceso se describen las tareas realizadas actualmente por los usuarios, sus patrones definidos de flujo de trabajo, los cuales se originan de sus esquemas mentales y las necesidades de información para realizar su trabajo. Es decir, se procura identificar “qué el usuario hace”, “de qué manera lo hace”, y “qué necesita para hacerlo”. De esa manera, se logra el entendimiento conceptual de las tareas que deberán formar parte del sistema en desarrollo. Esta actividad está contemplada en el Estándar ISO 13407. También se determinan las propiedades de usabilidad que debe brindar el sistema.

Para cada uno de los roles obtenidos en el Análisis de Usuario se van a determinar las tareas que realizan así como el contexto donde trabajan. Las tareas y el contexto de trabajo de cada rol se van a obtener de las respuestas obtenidas en los formularios del Análisis de Usuarios y se llenan en la tabla 4.

**Tabla 4: Listado de las tareas por usuarios**

Usuario	Tarea	Contexto de uso

En el análisis de tareas se determinan las propiedades de usabilidad del sistema a partir de los atributos de usabilidad que son más relevantes para el sistema identificados, se determinan las propiedades de usabilidad que son la entradas para el diseño arquitectónico.

**Herramientas:** Formularios y marco de relación atributos-beneficios y atributos-propiedades

**Salidas de la fase de análisis:**

Perfil del usuario, análisis contextual de tareas, lista de tareas, requisitos de usabilidad, atributos de usabilidad, propiedades de usabilidad, listado de jerarquía de beneficios de usabilidad y vista de presentación del expediente de arquitectura.

**2.3.2 Fase de diseño**

Debe alcanzarse una idea clara de cómo será la interfaz de usuario y las relaciones con esta para desarrollar las especificaciones funcionales que sirvan de guía al diseño posterior. La interfaz determinará en gran medida la percepción que el usuario tendrá de la aplicación. Es recomendable tener en cuenta en esta fase, la usabilidad en el momento del diseño de la arquitectura de software. La norma ISO 9126 establece que para aumentar la calidad de software en el diseño de un sistema se debe tomar en cuenta, primordialmente, la presentación de la información, la funcionalidad de la aplicación y la arquitectura del software.

**Entradas**

Son las salidas de la fase anterior (Listado de jerarquía de beneficios de usabilidad del sistema, atributos de usabilidad presentes en el sistema, propiedades de usabilidad y requisitos de usabilidad)

### **Responsables:**

- **Evaluador:** Es el responsable de realizar la evaluación de los prototipos no funcionales mediante los test de usuarios.
- **Arquitecto:** Es el responsable de la toma de decisiones en la arquitectura del software en cuanto a los escenarios y patrones de usabilidad a tener en cuenta en el diseño arquitectónico.

### **2.3.2.2 Actividades de la fase de diseño**

#### **2.3.2.2.1 Actividad: Evaluar prototipos**

Son documentos, diseños o sistemas que simulan o tienen implementadas partes del sistema final los cuales pueden incluir sólo el diseño de la interfaz o también partes de código. Los prototipos son herramientas muy útiles para hacer participar al usuario en el diseño y poder evaluarlo en las primeras fases del desarrollo. Proporcionan las primeras versiones de los componentes del sistema, cuya versión o prototipo final se define a partir de varias interacciones prototipado-evaluación. La función del prototipo es poder evaluar si el sistema cumple los objetivos funcionales y de usabilidad que se definieron.

### **Herramientas:**

**Test de usuarios:** En este proceso no sólo se verifican y validan los prototipos, sino también se evalúa su usabilidad, usando como técnicas de inspección la evaluación heurística que consiste en analizar la conformidad de la interfaz con unos principios reconocidos de usabilidad (la "heurística") mediante la inspección de varios evaluadores y se examinan todos los aspectos del prototipo en relación a los requerimientos predeterminados.

Para la elaboración del test de usuario se han seguido los principios determinados por Jacob Nielsen (Nielsen, 1994). Para obtener la lista final de los principios de usabilidad a seguir en la evaluación se partió de una lista inicial donde se incluían las características básicas que se deben estudiar según los autores antes mencionados. Dado que el objeto de estudio incide en los sistemas de gestión, la lista inicial se fue depurando en función de las características del sistema. El formulario *checklist* (lista de chequeo) se realizó en paralelo a la determinación de los principios, debido que, si para un principio era costoso encontrar preguntas para evaluarlo, se determinaba eliminarlo o solaparlo con otro.

Para llegar a la lista de principios finales se ha procedido de la siguiente manera:

1. Se considera como parte de la prevención de errores la utilización de valores por defecto que el usuario pueda visualizar, ya que gracias a estos se evitan errores al introducir datos.
2. La característica "Limitar acciones según el usuario" se ha decidido que sea eliminada debido a que estas limitaciones están dadas por los permisos que posee el usuario y no por la herramienta en sí. El administrador será el encargado de dar estos permisos al dar de alta al usuario en la base de datos por lo que queda fuera del dominio del sistema.
4. Se han solapado las características relacionadas con la información que muestra el sistema para evitar elementos invisibles de navegación, sobrecarga de funcionalidades así como información superflua.
5. Se considera que si un usuario puede volver a un estado anterior ante un error es que tiene control en el sistema por lo que la tercera y cuarta características se han unido en un principio llamado "Control y autonomía".

A continuación se muestran los principios finales explicados de manera más extensa.

**1. Retroalimentación:** Los usuarios siempre tienen que estar informados del estado del sistema, las tareas que están realizando, la estructura del sistema y dónde se encuentran en cada momento.

**2. Visualizar el lenguaje del usuario:** Es recomendable que el sistema utilice el lenguaje de los usuarios y no uno propio relacionado con la tecnología.

**3. Control y autonomía:** El usuario tiene que tener la sensación de que posee el control sobre el sistema. Es importante que si el usuario se equivoca al realizar una acción tenga una opción para volver al estado anterior y salir de la situación indeseada. Esto puede no ser posible en algunos casos pero deben de existir salidas de emergencia en el máximo de situaciones posibles. Botones como "Cancelar" una operación en curso, "Deshacer" un cambio realizado, "Salir" para abandonar el sistema en cualquier momento o "Guardar" para que el usuario pueda volver a ese estado en el futuro ayudan a conseguir este principio. Además, es recomendable que el usuario pueda ponerse en contacto con una persona responsable de la del sistema si esta deja de funcionar correctamente o tiene algún problema grave que no pueda solucionar.

**4. Consistencia:** El usuario no tiene que encontrarse delante de palabras distintas para acciones y situaciones que significan lo mismo. La consistencia tiene que aplicarse a: sintaxis, lenguajes,

aparición visual, iconos (mismos iconos para las mismas acciones) y efectos (acciones que tienen el mismo efecto en situaciones equivalentes).

**5. Prevención de errores:** Es mejor realizar un diseño que prevenga los errores que un mensaje de error. Por otro lado, evitar que al introducir datos el usuario cometa errores es muy importante, esto se puede conseguir de diferentes maneras: con una ayuda completa, indicando al lado de donde se puede introducir información el formato exigido, sólo permitiendo escribir con el formato válido en una casilla, dar un ejemplo o mostrar unos valores por defecto.

**6. Minimizar la carga de memoria del usuario:** El usuario no tiene que estar recordando información, acciones u opciones que ha visto con anterioridad. Si es estrictamente necesario se ha de intentar que el aprendizaje sea mínimo y rápido y que se favorezca el reconocimiento antes que el recuerdo.

**7. Flexibilidad y eficiencia de uso:** Conocer las acciones más utilizadas por los usuarios e introducir aceleradores que permitan realizarlas más rápido. Va encaminado a usuarios avanzados que son expertos en el sistema. Algunas estrategias que pueden seguirse son las siguientes: doble clic o combinación de teclas en vez seleccionar de un menú, abreviaturas o guardar configuración anterior. Para hacer más eficiente el uso del sistema es necesario mantener al usuario ocupado todo el tiempo, es decir, que en todo momento pueda realizar acciones y si el sistema está ocupado procesando algún tipo de información dejar que el usuario realice otras tareas mientras tanto. Por otro lado, poner los botones con las acciones más importantes de manera destacada, más grandes o en posiciones estratégicas (los bordes y esquinas son más fácilmente alcanzables) ayuda a que el usuario visualice el botón más rápidamente y por tanto se reduzca la latencia.

**8. Interfaz visible y práctica:** Sólo debe aparecer información relevante y necesaria, evitar la sobrecarga inútil de información. La información debe aparecer en un orden natural que espera el usuario. Toda la información debe estar visible evitando tener submenús ocultos donde el usuario tiene que recordar dónde había visto una determinada información. El sistema debe ser lo más sencillo posible evitando poner operaciones diferentes para situaciones parecidas, sobrecarga de funcionalidades o pasos repetitivos.

**9. Ayuda para reconocer, diagnosticar y recuperar al usuario de errores:** Los mensajes de error tienen que ser expresados en lenguaje llano, indicando el problema y sugiriendo una solución de forma constructiva. Mensajes que ayuden a entender al usuario qué es lo que está

pasando y no tratarlo de manera hostil. Además, no deben de haber situaciones de error no controladas ni situaciones en las que la herramienta no haga ninguna acción.

**10. Ayuda completa y accesible:** El sistema puede en sí mismo prevenir errores poniendo una ayuda útil o documentación que clarifique los aspectos más complicados de la herramienta. Debe ser lo más precisa posible sin poner información irrelevante y utilizar un lenguaje claro para el usuario sin tecnicismos. Si se intenta explicar la manera de realizar una tarea larga se recomienda hacer una ayuda por pasos. La ayuda debe estar accesible desde cualquier punto del sistema, especialmente en aquellos donde el usuario pueda tener más problemas de comprensión.

**11. Protección del trabajo del usuario:** El usuario debe poder guardar en todo momento el trabajo que lleva realizado hasta ese momento para que en caso de error no se pierda información. Es muy aconsejable un botón tipo “Guardar” que permita ir almacenando las distintas versiones.

**12. Legibilidad:** El sistema debe poder verse de manera clara sin tener partes en las que sea dificultoso ver el contenido. El texto tiene que tener contraste con respecto al fondo de la pantalla y un tamaño adecuado para ser leído.

**13. Compatibilidad con navegadores:** El sistema debe poder utilizarse en los navegadores más comunes y versiones tanto actuales como antiguas.

El artefacto para la evaluación de los prototipos se basará en los principios antes mencionados, se encuentra en el anexo 4, los evaluadores podrán indicar para cada pregunta la frecuencia con que aparece el problema, el impacto del mismo así como una observación del problema. Ver tabla 5

**Tabla 5: Formulario de evaluación de los prototipos de IU.**

Directriz	Evaluación	Impacto	Frecuencia	Comentarios
El sistema está libre de información irrelevante e innecesaria	1			

El criterio utilizado para estimar el la evaluación de cada item del *checklist* es el siguiente:

-1: el sistema no satisface la directriz

+1: el sistema sí satisface la directriz

0: no aplica

Si una directriz no es relevante, puede dejar el espacio en blanco. Puede agregar algún comentario a la derecha si lo desea.

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

El criterio utilizado para estimar el impacto de cada una de las directrices es el siguiente:

1. Bajo: Aunque es recomendable que se cumpla la afirmación su incumplimiento no implica confusión ni error en el usuario. No daría problemas de usabilidad importantes.
2. Medio: El incumplimiento puede provocar problemas no muy graves de usabilidad aunque conviene resolverlos ya que se facilitaría el funcionamiento del sistema.
3. Alto: Produce problemas altos de comprensión y funcionalidad en el sistema por lo que es primordial que el problema sea solventado. Puede provocar problemas graves de usabilidad

La columna frecuencia puede tener los siguientes valores: siempre (s) si siempre se cumple la afirmación en el sistema, no (n) si nunca se cumple y a veces (a) si se cumple alguna vez. La respuesta afirmativa de la pregunta implica la satisfacción del criterio a evaluar, mientras que una negativa implica que el criterio no se cumple en el sistema.

El resultado de la evaluación heurística será de forma cuantitativa arrojando el por ciento de usabilidad del sistema, los resultados se presentarán como se muestra en la figura 9.

Principios de Usabilidad	Calificación Neta	# Preguntas	# Respuesta	Calificación
Pantalla de Bienvenida	11	11	11	100%
Minimizar carga de usuario	27	27	27	100%
Interfaz visible y práctica	33	33	33	100%
Prevención de errores	20	20	20	100%
Control y autonomía	9	9	9	100%
Legibilidad y eficiencia de uso	21	21	21	100%
Consistencia y Diseño Gráfico	20	20	20	100%
Búsquedas	15	15	15	100%
Ayuda, Retroalimentación y Recuperación de Errores	30	30	30	100%
Calificación Final		186	186	100%

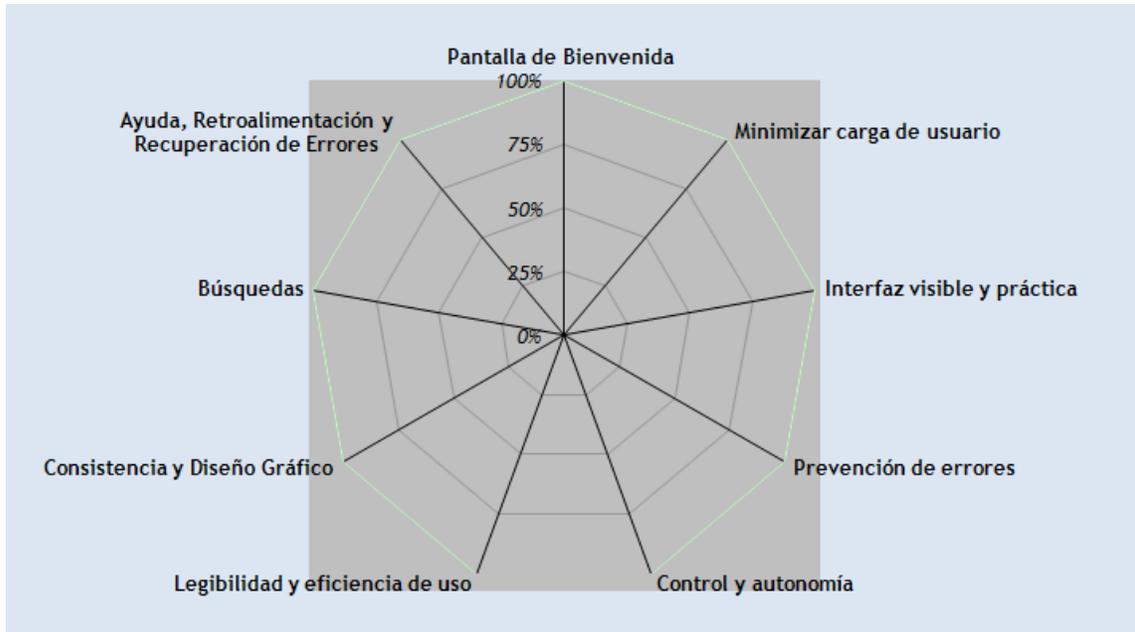


Figura 9. Resultados de la evaluación heurística.

**2.3.2.2.2 Actividad: Determinar escenarios y patrones de usabilidad a partir de los atributos de usabilidad.**

Se determina los patrones de usabilidad arquitectónicamente sensibles a partir de los atributos y propiedades de usabilidad identificado en la fase de análisis a través del marco SAU que muestran la relación entre las propiedades de usabilidad y los patrones de usabilidad.

**Herramientas:** Catálogo de escenarios de usabilidad y Marco SAU

Esta relación entre propiedades y patrones de usabilidad, es propuesta por Former (Folmer, 2003), mediante el marco SAU, este se compone de un conjunto integrado de soluciones de diseño que han sido identificados en varios casos en la industria del software para mejorar la usabilidad, el marco enlaza patrones de usabilidad arquitectónicamente sensibles a las propiedades y atributos de usabilidad. A continuación se presenta esta relación mediante la tabla 6:

**Tabla 6: Relación presente en el marco SAU**

Atributos de usabilidad	Propiedades de usabilidad en el sistema	Patrones de usabilidad
Eficiencia	Minimiza la carga cognitiva	Asistente Perfil de usuario
	Adaptabilidad	Perfil de usuario Historial de registro
	Gestión de errores	Historial de registro Forma de validación de campo

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

		Cancelar Deshacer Agregación de objeto Agregación de comandos Vista previa Reutilización de la información
	Retroalimentación	Indicación del estado
<b>Fiabilidad</b>	Minimiza la carga cognitiva	Asistente Perfil de usuario Apoyo de búsqueda completa
	Orientación	Asistente Ayuda contextual
	Gestión de errores	Historial de registro Forma de validación de campo Cancelar Deshacer Agregación de objeto Vista previa
	Consistencia	Múltiple vistas
<b>Satisfacción</b>	Adaptabilidad	Perfil de usuario Historial de registro
	Explícito control del usuario Accesibilidad	Cancelar Deshacer Agregación de objeto
<b>Facilidad de aprendizaje</b>	Orientación	Asistente Ayuda contextual Diferentes lenguajes
	Retroalimentación	Indicación del estado
	Consistencia	Múltiple vistas
	Accesibilidad	Multi –tarea

Los escenarios conforman cada patrón arquitectónico identificado en el marco SAU, los cuales toman lugar dentro de la jerarquía de ingeniería de software. Estos patrones pueden ser útiles para el diseño de la arquitectura, incluso si no son usados.

Para cada escenario, se proporciona un patrón de arquitectura que implementa el escenario y presenta los siguientes los elementos: El catálogo de escenarios se encuentra en el anexo 5 y brindará los siguientes elementos:

**Nombre del escenario:** coincide con el nombre del patrón de usabilidad que implementa el escenario

**Descripción:** Se describe en qué consiste el escenario de usabilidad.

**Aplicable a mi solución:** Si o No

**Solución del escenario:** Describe los elementos que conforman la arquitectura, sus relaciones, sus responsabilidades, etc. La solución para un patrón específico se hará explícita a partir de una:

- **Representación gráfica:** Una figura que representa los componentes de la arquitectura y sus interacciones. Las flechas numeradas entre los distintos componentes representarán las interacciones. Las flechas con las líneas ininterrumpidas indican el flujo de datos, mientras que las líneas interrumpidas representan el flujo de control entre los componentes.
- **Participantes:** Una descripción de los componentes que participan en la solución propuesta y las relaciones (representadas por flechas) para determinar cómo deben asumir sus responsabilidades.

**Restricciones del patrón solución:** Impacto del patrón en otros atributos de calidad, como la flexibilidad, la portabilidad, etc.

**Beneficios de usabilidad:** Una argumentación razonada acerca del impacto que tiene la aplicación del patrón en la usabilidad.

### 2.3.2.2.3 Actividad: Rediseño

Más que un proceso, el rediseño se caracteriza por ser un indicador de decisión basado en los resultados de los análisis de los test. De esa manera, si se identifica que el prototipo, producto o servicio no cumplen con los requerimientos y estándares establecidos, se desvía el flujo del ciclo de desarrollo a la definición de los objetivos de usabilidad, con el propósito de verificar su validez.

En el caso de la Arquitectura del software, si la usabilidad necesita mejorarse, es decir no cumple con los escenarios de usabilidad propuestos, se entraría en un ciclo de mejora en la arquitectura software y nuevas verificaciones, hasta que el nivel de usabilidad sea el adecuado a través del método de evaluación SALUTA mediante escenarios de uso. Este proceso no implica que no se evalúe la usabilidad del sistema una vez terminado, simplemente se pretende adelantar el ciclo de evaluación/mejora de la usabilidad con el fin de ahorrar esfuerzos en el proceso de desarrollo.

**Herramienta:** Método de evaluación arquitectónico SALUTA y Test de usuarios.

### Salidas de las fase de diseño

- Prototipos de IU con los aspectos de usabilidad incluido.
- Listado de patrones y escenario de usabilidad a tener en cuenta en el diseño arquitectónico

### 2.3.3. Fase de Lanzamiento o retroalimentación del usuario

La fase de lanzamiento de todo proyecto, suele ser una de las más críticas de todo el proceso. Es el momento en que se ven concretadas en mayor o menor grado las expectativas puestas en el producto. De todas formas cabe indicar que la percepción que tenga el usuario final del producto tiene un peso específico enorme a la hora de indicar si el producto será aceptado o no. Resumiendo, se puede indicar que el éxito total del producto dependerá de dos factores muy importantes:

Por un lado que el usuario se sienta cómodo con el sistema. Definiendo sentirse cómodo como que no le dé errores, que no le resulte complicado usarlo, que recuerde fácilmente donde están las diferentes opciones, sus funcionalidades y por otro lado que obtengan los resultados esperados.

#### Entradas

Listas de chequeo de usabilidad para sistemas de gestión y sistema implementado con aspecto de usabilidad incluido.

#### Responsables:

- Evaluadores: Son los encargados junto con los usuarios evaluar el sistema.
- Administrador de calidad: Es el encargado una vez realizada la evaluación del sistema con la lista de chequeo, aplicar las métricas para determinar el grado de usabilidad del mismo.

### 2.3.3.2 Actividades de la fase lanzamiento o retroalimentación del usuario

#### 2.3.3.2.1 Actividad: Determinar el grado de usabilidad del sistema.

#### Herramienta:

Una vez el producto ha sido instalado y puesto en explotación durante un cierto periodo denominado habitualmente como fase de pruebas, se recoge lo que se llama el *feedback*(retroalimentación) del usuario, o sea las impresiones, mejoras, defectos, etc. Para evaluar el producto una vez terminado, se realizará mediante la lista de chequeo de usabilidad para sistemas de gestión aplicada por los evaluadores.

Como resultado final se obtienen el grado de usabilidad mediante una métrica, para calcular la misma se debe determinar la evaluación parcial ( $E_p$ ), que consiste en promediar los indicadores de la lista de chequeo evaluados de "0", para ello se utiliza la siguiente ecuación:  $E_p = \frac{I}{N}$ , donde I: Cantidad de indicadores evaluados de "0" que es el caso que el elemento revisado no

presente errores y N: Total indicadores de la lista de chequeo. Luego se calcula el grado de usabilidad (U), calculando el promedio de la Ep de cada evaluador, mediante la siguiente ecuación:

$$U = \frac{\sum_{i=0}^k Ep}{K}$$

**Figura 11: Fórmula para el cálculo de la usabilidad**

Dónde:

U: Evaluación total del sistema.

K: Total de evaluadores que participa en la evaluación.

Ep: Evaluación parcial dada por cada evaluador.

Una vez obtenida la U se convierte a escala de % multiplicando el resultado obtenido por 100 y se mide en la escala de criterios para la aceptabilidad del nivel de usabilidad definido en el estándar ISO 14598, en tres regiones de un rango de 0 a 100%:

- Satisfactoria: cuando U es mayor que 60%.
- Aceptable: cuando U está entre 40% y 60%.
- Insatisfactoria: cuando U es menor que 40%.

La lista de chequeo de usabilidad para sistemas de gestión se encuentra en el **anexo 6**

### **Salidas de la fase Lanzamiento o retroalimentación del usuario**

- Grado de usabilidad del producto.
- Posibles mejoras de usabilidad del producto.
- Incremento del auto-aprendizaje en cuanto a la usabilidad (toda nueva experiencia supone un incremento en cuanto a conocimientos ya sean nuevos o mejoras de los ya adquiridos).

### **2.4 Beneficios identificados de la aplicación de las actividades de aseguramiento y evaluación de la usabilidad en el CEIGE**

La incorporación de las técnicas de usabilidad durante todo el proceso de desarrollo de software trae consigo varios beneficios clasificado de la siguiente manera:

**Reducción de Costos de Producción:** Al evitar el sobre-diseño y disminuir la probabilidad de cambios en el producto final.

**Reducción de Costos de Mantenimiento y Soporte:** Al brindar la posibilidad de que el desarrollo de la aplicación sea fácil de utilizar y de aprender.

**Mejoramiento de la Imagen Corporativa:** Productos con un buen grado de usabilidad permite incrementar las ventas y la calidad del producto mejorando la competitividad en el mercado, junto con la preocupación notoria por mejorar la calidad de vida de los usuarios al reducir los costos de aprendizaje de la aplicación.

Aunque los beneficios de la aplicación de la usabilidad en el ciclo de desarrollo de software son latentes y significantes, existen algunos inconvenientes que se deben tener en cuenta:

**Inexperiencia:** La mala utilización de las técnicas puede afectar negativamente en la duración y costos del proyecto debido a la inexperiencia de los arquitectos en el tema de la usabilidad.

### 2.6 Conclusiones parciales

A partir de los elementos teóricos estudiados se elaboró un procedimiento para el aseguramiento y la evaluación de la usabilidad en los productos del CEIGE.

Concluyendo que:

- El procedimiento propuesto permite lograr un mejor uso del sistema a través de las decisiones de diseño incorporadas en la arquitectura de software con soporte a la usabilidad.
- Las fases organizativas que requiere el procedimiento propuesto se ajusta sin mayores dificultades a las fases existente en el proceso de desarrollo del CEIGE.
- El procedimiento propuesto incidirá en la reducción de costos de producción y de mantenimiento, mejorará la imagen corporativa del centro, será aplicable para los sistemas de gestión del CEIGE.

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

#### 3.1 Introducción

En este capítulo se describen los resultados de la ejecución del pre-experimento realizado para comprobar el impacto de la aplicación de la propuesta en la usabilidad de los productos del CEIGE. Los resultados obtenidos se brindarán en función de un conjunto de indicadores demostrativos del soporte de la usabilidad en la arquitectura de los productos de software. Se mostrarán los resultados de la ejecución del pre-experimento puesto en práctica, haciendo uso de métodos estadísticos para el procesamiento de los resultados obtenidos.

Finalmente se hace una valoración del impacto costo-beneficio del procedimiento.

#### 3.2 Indicadores de medición

Para el análisis de los resultados de la puesta en práctica de la presente investigación, se toma como variable el soporte de la usabilidad en la arquitectura del software. Esta variable se medirá en función de una serie de indicadores, que son generados por las actividades que propone el procedimiento. La fase de análisis tiene como salida los requisitos, atributos y beneficios de usabilidad que son entradas para la fase posterior. La fase de diseño tiene como salida una lista de patrones de usabilidad, todos estos elementos son entradas para que el diseño arquitectónico soporte la usabilidad. Estos indicadores se muestran a continuación:

- **Número de atributos de usabilidad identificados y priorizados:** la usabilidad es un atributo de calidad, que se descompone en atributos específicos como son: la eficiencia, la satisfacción, la facilidad de aprendizaje y la fiabilidad con la que un producto permite alcanzar objetivos específicos.
- **Número de propiedades de usabilidad identificados y priorizados:** Las propiedades de usabilidad se derivan de la heurística del diseño y los principios del diseño, tales como proporcionar retroalimentación en cualquier momento o proporcionar consistencia. Existen en el dominio del problema, y forman parte de los requisitos que impulsan el proceso de diseño arquitectónico.
- **Números de patrones de usabilidad empleados:** los patrones de usabilidad se refieren a una técnica o mecanismo que debe aplicarse al diseño de la arquitectura de un sistema de software con el fin de hacer frente a una necesidad identificada por una propiedad de la usabilidad en la fase de requisitos.

**3.3 Características de la muestra**

La población total está constituida por todos los proyectos del sistema CEDRUX, para un total de 10. La muestra se seleccionó de forma intencionada teniendo en cuenta que todos los elementos de la población tienen características muy similares, además se seleccionaron los proyectos en los que por su estado dentro del proceso de desarrollo, permitiera una mejor evaluación de la variable de la presente investigación. La muestra quedó constituida por 7 proyectos, lo que representa un 70 por ciento de la población total. Todos estos proyectos presentan características similares, presentan negocios similares y todos comparten la misma arquitectura de sistema base, conceptualmente. Los proyectos seleccionados son:

**Tabla 7: Proyectos del sistema CEDRUX, pertenecientes a la muestra seleccionada.**

<b>NOMBRE DEL PROYECTO</b>	<b>SIGLAS</b>	<b>FASE EN LA QUE SE ENCUENTRA</b>
Banco	Banc.	Pruebas de aceptación
Costos y Procesos	Cost.	Pruebas de aceptación
Capital Humano	RRHH.	Pruebas de aceptación
Estructura y Composición	ESTRUCT.	Pruebas de aceptación
Inventario	INV.	Pruebas de aceptación
Caja	Caj.	Pruebas de aceptación
Configuración	Conf.	Pruebas de aceptación

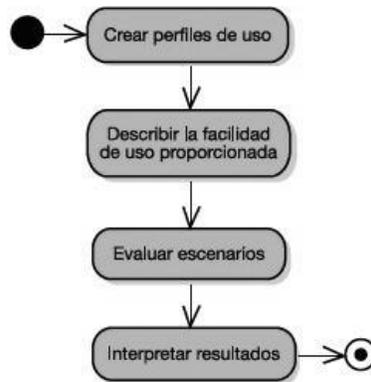
Para obtener los datos se diseñó un instrumento de captación de datos y se aplicó a los arquitectos y analistas de los proyectos antes de comenzar el pre-experimento y después de haberlo realizado, el instrumento aplicado se encuentra en el Anexo 2.

**3.4 Evaluación**

**3.4.1 Análisis del soporte de la usabilidad en la arquitectura del software**

**3.4.1.1 Análisis de los atributos de usabilidad, números de patrones y propiedades de usabilidad empleados**

El análisis de estos indicadores se va realizar a través del método de evaluación de SALUTA, este método propone que para cada escenario de uso, se debe priorizar los atributos de usabilidad. Los atributos que propone este método se presentan a continuación: eficiencia, satisfacción, confiabilidad y facilidad de aprendizaje descritos anteriormente en epígrafes anteriores. En la figura 12 se muestra el flujo de pasos del método SALUTA.



**Figura 12.**Flujo de pasos del método SALUTA.

Los productos de la muestra seleccionada no identificaron, ni priorizaron atributos de usabilidad, por lo que no tienen una visión clara de que patrones de usabilidad deben de implementar para darle respuesta a cada atributo. El marco SAU de Folmer, y en el que se basa el método SALUTA, expresa las relaciones entre la arquitectura de software y la usabilidad. El marco SAU se compone de un conjunto integrado de soluciones de diseño que han sido identificados en varios casos en la industria del software. EL marco relaciona patrones de usabilidad arquitectónicamente sensibles con las propiedades y atributos de usabilidad. De ahí la importancia de identificar los atributos, para estos traducirlo a propiedades que son utilizado como requisitos en el diseño.

#### **Aplicación del método SALUTA para determinar el soporte a la usabilidad en la arquitectura**

Para la aplicación del método SALUTA el primer paso es la creación de escenarios de usos. De acuerdo con la definición de la ISO 9241-11, la usabilidad depende de: los usuarios (para quienes va dirigido el sistema), las tareas (¿Qué es lo que los usuarios van a realizar en el sistema?), y el contexto de uso (¿Dónde y cómo es el producto utilizado?). Para cada escenario se priorizan los atributos de usabilidad en orden de prioridad. El analista en la fase de análisis interpreta para cada escenario los valores de prioridad con un peso del uno al cuatro, donde el atributo evaluado de cuatro tendrá mayor prioridad. A continuación se muestra un perfil de uso y los elementos que lo integran aplicado al proyecto Inventario, que forma parte de los proyectos de la muestra seleccionada, el cual se muestra en la tabla 8. La inserción de un objeto en el sistema es propensa a errores, por lo tanto, la confiabilidad es importante para este escenario, en el caso específico de la gestión de un producto en el subsistema Inventario (adicionar, eliminar, modificar) es lento y poco fiable cuando una misma acción se le aplica a

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

varios productos a la misma vez, en consecuencia, los valores altos son asignados a la fiabilidad y la eficiencia, mientras que los valores bajos a los otros atributos.

Tabla 8: Escenario de uso de inventario

Producto	Usuarios	Tareas	Satisfacción	Facilidad aprendizaje	Eficiencia	Fiabilidad
Inventario	Contador del almacén	Insertar producto	Usuario siente que tiene el control del sistema	Facilidad con el usuario realiza la tarea en el sistema	Tiempo de respuesta del sistema en la tarea	No debe existir error en la realización de la tarea.
	Perfil de usuario		1	2	3	4
	Requisitos de usabilidad					

Para la realización de los demás escenarios, las tareas similares se combinaron y ciertas tareas con frecuencias de ejecución baja no se tuvieron en cuenta. Se trató de minimizar el número de escenarios sin perder la representatividad de la usabilidad que requiere el sistema. Sin la identificación y especificación de los requisitos de usabilidad, elemento que no se tenía en cuenta antes de la aplicación del procedimiento, se hacía difícil la priorización de los atributos, como resultado de varios talleres con los analistas y arquitectos se crearon los escenarios de usos para los proyectos de la muestra seleccionada, de los cuales una muestra representativa es la que se presentan en la siguiente tabla, por problemas de espacio no se pudieron poner todos los escenario. Para lograr una solución óptima en la priorización de los atributos de usabilidad que propone SALUTA, al ser este proceso guiado por el conocimiento indocumentado de los arquitectos de software con experiencia, se recomienda involucrar a los analistas y a los usuarios del sistema.

En la priorización de los atributos de usabilidad se tuvo en cuenta para que para las tareas complejas en el sistema en cada escenario, la **facilidad de aprendizaje** es un atributo prioritario, para las búsquedas y cálculos internos el atributo **fiabilidad** es importante y para las tareas de procesamiento complejo la **eficiencia** debe ser priorizado.

Tabla 9: Escenarios de usos de la muestra seleccionada con la priorización de los atributos de usabilidad

Escenario	Proyecto	Usuario	Tarea	Eficiencia	Facilidad aprendizaje	fiabilidad	Satisfacción

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

1	Costo y Procesos	Financista	Ejecutar secuencia de traspaso	4	2	3	1
2	Costo y Procesos	Financista	Adicionar asociación para áreas de responsabilidad	3	2	4	1
3	Costo y Procesos	Financista	Modificar asociación para áreas de responsabilidad	2	4	1	3
4	Costo y Procesos	Financista	Eliminar asociación con áreas de responsabilidad	2	3	1	4
5	Costo y Procesos	Financista	Imprimir asociaciones de un área de responsabilidad	4	2	3	1
6	Inventario	Jefe de almacén	Aprobar hoja de inventario inicial	4	2	3	1
7	Inventario	Contador del almacén	Cancelar estado a la hoja de inventario inicial	4	2	3	1
8	Inventario	Contador del almacén	Confirmar hoja de inventario inicial	4	2	3	1
9	Inventario	Contador del almacén	Buscar producto de la hoja de inventario inicial	3	2	4	1
10	Estructura y Composición	Contador del almacén	Listar estructuras externas	2	3	1	4
11	Estructura y Composición	Contador del almacén	Listar estructuras internas	2	3	1	4
12	Banco	Financista	Calcular conciliación bancaria	3	2	4	1
13	Banco	Financista	Procesar estado de cuenta	3	2	4	1
14	Banco	Financista	Contabilizar estado de cuenta	4	2	3	1

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

15	Banco	Financista	Corregir diferencia con banco	3	2	4	1
16	Capital Humano	Especialista Recursos Humanos	Emitir movimiento de nómina	3	2	4	1
17	Capital Humano	Especialista Recursos Humanos	Consultar movimiento de nómina	4	2	3	1
18	Capital Humano	Especialista Recursos Humanos	Actualizar movimiento de nómina	2	4	1	3
19	Capital Humano	Especialista Recursos Humanos	Revertir movimiento de nómina.	4	2	3	1
20	Configuración	Financista	Adicionar operaciones	3	2	4	1
21	Configuración	Financista	Asociar cliente o proveedor a la entidad	3	2	4	1
22	Caja	Financista	Adicionar caja	3	2	4	1
23	Caja	Financista	Adicionar solicitud de reembolso	3	2	4	1

El siguiente paso es el análisis del diseño arquitectónico que se realizó en los proyectos antes de la aplicación del procedimiento, el mismo se obtuvo a través de entrevistas con los arquitectos y como resultado salió un listado de elementos de usabilidad que de cierta forma se habían aplicado.

De manera general se puede decir que los proyectos de la muestra seleccionada presentan elementos de usabilidad, los que por sus características, la autora de la investigación en colaboración con los arquitectos de estos sistemas los agrupó según los patrones de usabilidad, ya que estos elementos no fueron edificados ni concebidos como patrones de usabilidad durante el diseño arquitectónico, por lo que medirlos y modificarlos resultaba difícil debido a que no estaban estandarizados. A continuación se muestra la forma en que se aplicaron algunos patrones de usabilidad en el diseño de la arquitectura de la muestra seleccionada.

**Validación de formularios:** Las librerías que se usan en la capa de presentación proveen un mecanismo conocido como *MaskRe*, en el cual a través de una expresión regular se indican los datos compatibles con el campo. Estas librerías muestran errores cuando los datos introducidos son incorrectos.

**Retroalimentación:** Todas las acciones realizadas en el sistema devuelven como respuesta un código de mensaje, este código establece estados para indicar si se ha realizado exitosamente la acción o si ha ocurrido un error. Los componentes *widgets* de la capa de presentación proveen una propiedad llamada *loadMask* (máscara de carga), esta máscara le indica al usuario que el componente está cargando y tiene que esperar.

**Agregación de objetos:** Algunos sistemas como el de Inventario, provee este mecanismo en el cual se almacenan todos los objetos en un arreglo con la acción a ejecutarse, el cual es enviado al servidor y allí el sistema realiza varias veces la acción para cada uno de los objetos.

**Multi-Tarea:** El sistema usa las tecnologías AJAX, esta tecnología se caracteriza por ser no bloqueante, o sea que una vez que se realiza una acción que es enviada al servidor, el sistema no se queda esperando por la respuesta, por lo que no bloquea al resto de los componentes.

**Historial de registro:** La arquitectura provee componentes de traza el cual registra todas las acciones realizadas por los usuarios del sistema.

**Vista previa:** En algunos sistemas se imprimen documentos a partir de reportes, estos sistemas implementan un componente de previsualización de los datos a imprimir en una plantilla HTML.

### **Evaluación de los escenarios**

El siguiente paso consiste evaluar el soporte de la usabilidad en la arquitectura para cada uno de los escenarios. Utilizando el marco SAU se identifican las propiedades y patrones de usabilidad que deben de implementarse para dar respuesta al atributo priorizado en cada escenario. Este análisis es importante para la correcta toma de decisiones durante el diseño arquitectónico. Siguiendo el ejemplo para el proyecto Inventario, para el escenario Adicionar productos, los atributos de usabilidad priorizados en este caso fueron la fiabilidad y la eficiencia, a continuación siguiendo el marco SAU se muestra las propiedades y patrones de usabilidad que tenían que haberse implementado para que el sistema final cumpla con esos atributos. Ver figura 13 y 14

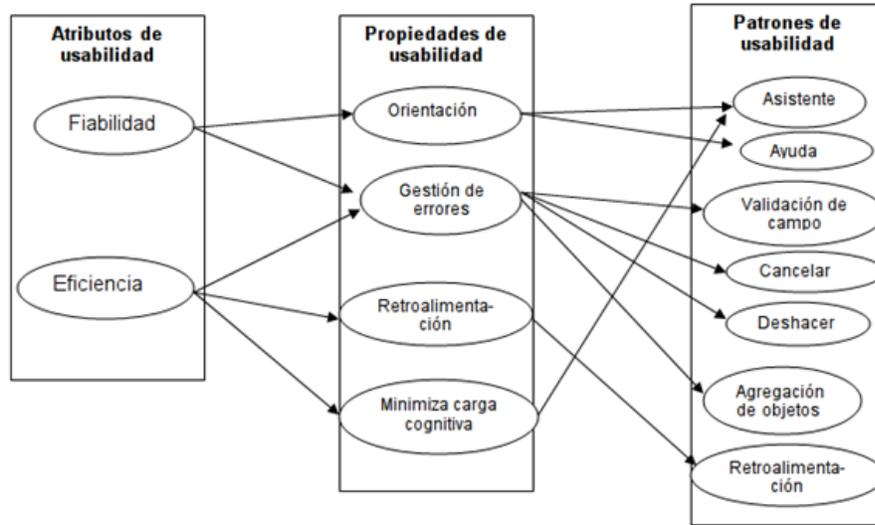


Figura 13: Marco SAU para los atributos eficiencia y fiabilidad

Producto	Tareas	Satisfacción	Facilidad aprendizaje	Eficiencia	Fiabilidad
Inventario	Adicionar producto	1	2	3	4

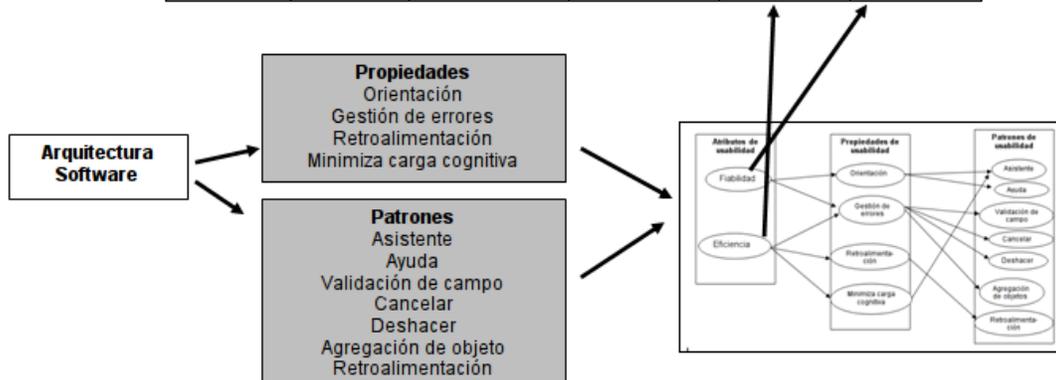


Figura 14: Aplicación del marco SAU

Este análisis se realizó para cada escenario, teniendo en cuenta las propiedades y patrones de usabilidad que deben de implementarse, este cuestionamiento solo tiene repuesta a nivel de requerimiento de usuario y del proceso de toma de decisiones de los arquitectos. Es importante analizar si el patrón de usabilidad es asimilado por la lógica (el concepto en base al cual funciona). Por tanto, la interacción debe diseñarse junto con la lógica de negocio, para asegurar de que la lógica del sistema sea usable. No es posible diseñar la lógica de negocio independientemente de la interacción y luego unirlas. Esta relación entre diseño de la interacción y el diseño software tradicional obliga a modificar este último para acomodar al primero.

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

Como resultado de un análisis similar de cada patrón descrito en el catálogo (Anexo 5), la siguiente matriz muestra que patrones y propiedades deben de tenerse en cuenta para el diseño arquitectónico de los escenarios descritos en la tabla 9.

**Tabla 10: Matriz Escenarios /patrones y propiedades de usabilidad**

Escenario	Patrones de usabilidad									Propiedades de usabilidad								
	Retroalimentación	Agregación de objeto	Cancelar	Validación de datos	Historial de usuario	Vista previa	Deshacer	Asistente	Ayuda contextual	Retroalimentación	Gestión de errores	Consistencia	Adaptabilidad	Orientación	Explicito control de usuario	Accesibilidad	Minimiza carga cognitiva	Explorabilidad
1	x	x	x	x	x	x		x		x	x		x				x	
2	x	x	x	x				x	x	x	x			x			x	
3	x							x	x	x		x		x		x		
4	x	x	x		x		x						x		x	x		
5	x			x	x	x		x					x		x	x		
6	x	x		x	x			x		x	x			x			x	
7	x	x	x	x	x			x		x	x			x			x	
8	x	x	x	x	x			x		x	x			x			x	
9	x			x				x	x	x	x			x			x	
10	x				x										x			
11	x				x										x			
12	x		x	x				x	x	x	x			x			x	
13	x	x		x				x	x	x	x			x			x	
14	x		x	x	x					x	x		x				x	
15	x			x				x	x	x	x			x			x	
16	x		x	x				x	x	x	x			x			x	
17	x	x		x				x		x	x						x	
18	x							x	x	x		x		x		x		
19	x	x	x	x	x		x	x		x	x		x				x	
20	x	x	x	x				x	x	x	x			x			x	
21	x	x	x	x				x	x	x	x			x			x	
22	x		x	x				x	x	x	x			x			x	
23	x	x	x	x				x	x	x	x			x			x	

**Interpretación de los resultados**

A continuación se presenta en la tabla 11, el resumen con la cantidad de patrones y propiedades de usabilidad por cada escenario y el soporte de cada escenario en la arquitectura de los proyectos, momentos antes y después de la aplicación del procedimiento. El soporte será expresado en una escala de tres niveles: + (Soporte alto), +/- (Soporte medio) y - (Soporte bajo).

**Tabla 11: Número de propiedades y patrones de usabilidad por Escenario**

Etapas	Proyecto	Escenarios	Número de patrones	Número de Propiedades	Soporte de la usabilidad en la arquitectura
Sin procedimiento	Costo Procesos	1	3/7	2/4	-
		2	3/7	2/4	-
		3	1/5	1/4	-
		4	3/5	2/3	+/-
		5	4/8	2/3	+/-
	Inventario	6	3/8	2/4	-
		7	3/8	2/4	-
		8	2/8	2/4	-
		9	2/7	2/3	-
	Estructura y composición	10	1/5	2/3	-
		11	1/5	2/3	-
	Banco	12	2/5	2/4	-
		13	2/5	2/4	-
		14	3/4	2/4	+/-
		15	2/4	2/4	+/-
	Capital Humano	16	2/5	2/4	+/-
		17	2/4	1/3	-
		18	1/3	1/4	-
		19	4/7	2/4	+/-
	Configuración	20	2/5	2/4	+/-
		21	2/6	2/4	-
	Caja	22	2/5	2/4	+/-
		23	3/6	2/4	+/-

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

<b>Con procedimiento</b>	Costo y Procesos	1	5/7	4/4	+/-
		2	6/7	4/4	+
		3	3/5	3/4	+/-
		4	5/5	3/3	+
		5	6/8	3/3	+
	Inventario	6	5/8	4/4	+/-
		7	5/8	4/4	+/-
		8	5/8	4/4	+/-
		9	3/7	3/3	-
	Estructura y composición	10	2/5	3/3	-
		11	2/5	3/3	-
	Banco	12	5/5	4/4	+
		13	5/5	4/4	+
		14	4/4	4/4	+
		15	4/4	4/4	+
	Capital Humano	16	5/5	4/4	+
		17	4/4	3/3	+
		18	3/3	4/4	+
		19	7/7	4/4	+
	Configuración	20	5/5	4/4	+
		21	6/6	4/4	+
	Caja	22	5/5	4/4	+
		23	6/6	4/4	+

En la Figura 15 y 16 se expone gráficamente la comparación de los indicadores número de patrones y propiedades de usabilidad respectivamente para los proyectos de la muestra.

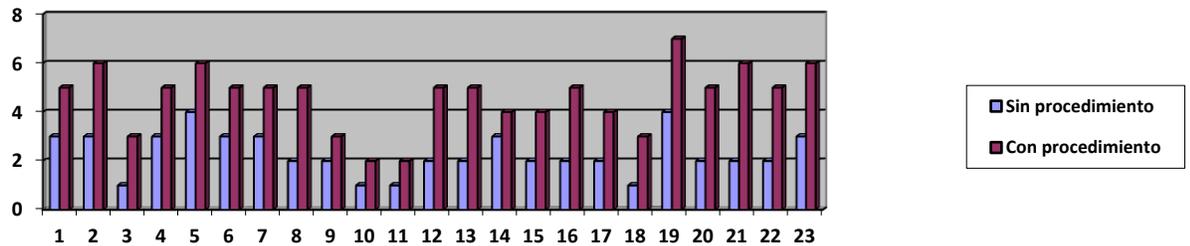


Figura 15: Número de patrones de usabilidad por cada escenario

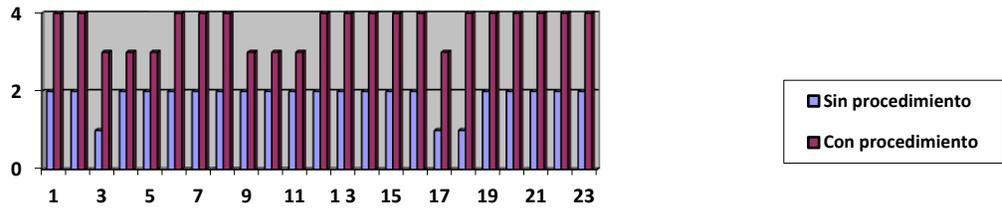


Figura 16: Número de propiedades de usabilidad por cada escenario

En la tabla 12 se presentan los resultados del total de evaluaciones de los escenarios, se muestra el número de escenarios analizados, y el resultado del soporte de la usabilidad en función de cuantos escenarios tienen soporte bajo, medio y alto antes de aplicar el procedimiento y después de aplicarlo.

Tabla 12: Número de propiedades y patrones de usabilidad por Escenario

Fases	Nro. de escenarios	Soporte bajo (-)	Soporte medio (+/-)	Soporte alto (+)
Sin el procedimiento	23	14	9	0
Con el procedimiento	23	3	5	15

En la Figura 17 se expone gráficamente la comparación del comportamiento antes mencionado.

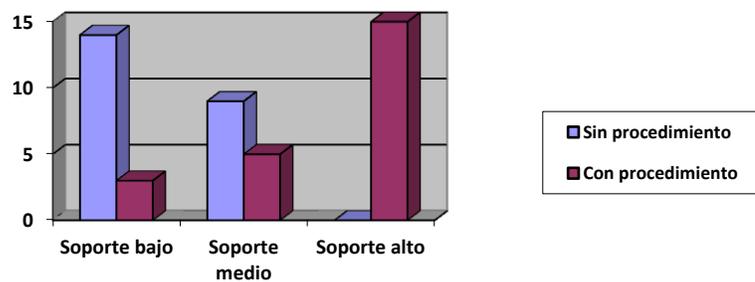


Figura 17: Soporte de la usabilidad

Se concluye que con la aplicación de procedimiento hubo un aumento en el número de atributos, propiedades y patrones de usabilidad que antes no se tenían en cuenta y por

consiguiendo un aumento del soporte de la usabilidad en la arquitectura de los productos de la muestra seleccionada.

**3.4.2 Análisis del indicador calidad del procedimiento**

**Capacidad de ser generalizado (Si):** El procedimiento fue implantado con éxito en los proyectos de la muestra seleccionada y fue adaptado a las necesidades de los mismos. El mismo puede ser generalizado a todos los proyectos del CEIGE.

Como uno de los resultados de la investigación se obtiene la vista de presentación que forma parte de la línea base de la arquitectura de la Universidad y se aplicará en todos los centros de desarrollo de la misma.

La vista de presentación en particular ha sido introducida en la Red de centros de desarrollo de la UCI a través del consejo técnico, elemento que se evidencia en un conjunto de acuerdos y actividades lo que avalan la introducción y generalización del mismo.

**Consejos técnicos presentados:**

Acuerdo	fecha	Cumplimiento
Acuerdo 01.Arq.08.12.11: La Dirección General Producción (DGP) debe circular Propuesta de Vista de Presentación y su relación con la Dirección de Comunicación Visual. Los centros deben enviar las inconformidades con la propuesta enviada y proponer su idea al respecto. Resp: DGP, red de centros.	8 /12/ 2011	Los centros no enviaron inconformidades con la propuesta.
Punto 1. Avances en la implantación de la vista de presentación. Resp: DGP. Despliegue del escenario genérico. Estandarización de la imagen gráfica de la UCI (para productos). Definición de etiquetado de sistemas, Manuales, Glosario de términos, Componentes, etc. Zonificación de las pantallas. Ej: Autenticación/Home	1 /3/ 2012	Los centros estuvieron de acuerdo con las definiciones presentadas.
Acuerdo 01.Arq.08.03.12: Circular información de la vista de presentación (patrones asociados), los centros deben enviar sus impresiones. Responsable DGP, red de centros	8/3/ 2012	Los centros no enviaron inconformidades con la propuesta.

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

Punto 5.Temas generales. (Avances próximo despliegue de Vista de Presentación.)	26/04/2012	La vista se montó en el GESPRO  <a href="http://gespro.prod.uci.cu/temario_poll/index_temario?father_id=10&amp;project_id=135">http://gespro.prod.uci.cu/temario_poll/index_temario?father_id=10&amp;project_id=135</a>
---	------------	---

El acta de aceptación se encuentra en el **anexo 8**.

**Homologación con modelos internacionales y nacionales (Si):** El procedimiento propuesto basa sus definiciones en modelos y estándares internacionales y nacionales. Ellos son: Ingeniería de la Usabilidad, el proyecto STATUS y USE, el método SALUTA, la norma ISO 13407 y el procedimiento establecido por CALISOFT. Mediante la observación científica de los modelos estudiados durante la investigación se obtuvieron un conjunto de variables que caracterizan la esencia de los modelos existentes, estas variables fueron utilizadas por la investigadora para realizar una comparación del procedimiento propuesto, defendiendo la idea de que si el mismo muestra valores superiores o iguales en las variables definidas, se puede llegar a la conclusión de que será factible como los modelos usados en la comparación, ver tabla 13.

**Tabla 13: Comparación del procedimiento propuesto con otros modelos**

Aspectos	STATUS	USE	Ingeniería de la Usabilidad	ISO 13407	Procedimiento CALISOFT	Procedimiento propuesto
Integración las fases de desarrollo	+	+	++	+	++	+++
Procesos definidos			+++		++	+++
Métodos de evaluación definidos	+	+	+++		++	+++
Cubrimiento al proceso de desarrollo	+	+	++	+	++	++
Roles definidos			++		++	+++

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

Artefactos definidos	+	+	++	+	++	+++
Técnicas y herramientas definidas	+	+	+++		+++	+++
Define Usabilidad como elemento visual			+++	++	+++	+++
Define usabilidad como elemento de la arquitectura	+++	+++				+++

#### 3.4.3 Ejecución del pre-experimento

El pre-experimento quedó diseñado de la siguiente manera:

De un total de 10 proyectos del sistema CEDRUX se seleccionó una muestra de 7 proyectos que representan el 70 % de la población, los mismos se encuentran en la tabla 14:

**Tabla 14: Proyectos de la muestra seleccionada**

Proyectos
Costo Proceso
Estructura y Composición
Inventario
Capital Humano
Banco
Configuración
Caja

Para las distintas comparaciones se hará uso de los indicadores: número de propiedades y de patrones de usabilidad empleados.

Los valores de cada uno de los proyectos presentes en el pre-experimento en función de los indicadores antes mencionados se presentan, ver tabla 15:

- El total de patrones de usabilidad por cada proyecto será la sumatoria de la cantidad empleado en cada escenario.

- El total de propiedades de usabilidad por cada proyecto será la sumatoria de la cantidad empleado en cada escenario.

**Tabla.15: Valores de los indicadores definidos por cada proyecto de la muestra antes y después del pre-experimento.**

Proyectos	Número de propiedades de usabilidad antes	Número de propiedades de usabilidad después	números de patrones de usabilidad antes	números de patrones de usabilidad después
Costos y Procesos	9	17	14	25
Estructura y Composición	4	6	2	4
Inventario	8	15	10	18
Capital Humano	6	15	9	19
Banco	8	16	9	18
Configuración	4	8	4	11
Caja	4	8	5	11

El método estadístico que se utilizará es el test de Wilcoxon para muestras apareadas, esta prueba permitirá comparar las diferencias entre dos muestras de datos tomados antes y después del tratamiento, se espera que con la aplicación del mismo existan diferencias significativas entre las observaciones.

**Prueba estadística 1:** Haciendo uso de la herramienta SPSS.

Aplicando el test de *Wilcoxon* para dos muestras apareadas a la variable *propiedades de usabilidad*, se muestra los resultados en la figura 18.

**Wilcoxon Signed Ranks Test**

**Ranks**

		N	Mean Rank	Sum of Ranks
Número de propiedades de usabilidad después -	Negative Ranks	0 <sup>a</sup>	,00	,00
Número de propiedades de usabilidad antes	Positive Ranks	6 <sup>b</sup>	3,50	21,00
	Ties	1 <sup>c</sup>		
	Total	7		

- a. Número de propiedades de usabilidad después < Número de propiedades de usabilidad antes
- b. Número de propiedades de usabilidad después > Número de propiedades de usabilidad antes
- c. Número de propiedades de usabilidad después = Número de propiedades de usabilidad antes

**Test Statistics<sup>b</sup>**

	Número de propiedades de usabilidad después - Número de propiedades de usabilidad antes
Z	-2,214 <sup>a</sup>
Asymp. Sig. (2-tailed)	,027

- a. Based on negative ranks.
- b. Wilcoxon Signed Ranks Test

**Figura 18. Test de Wilcoxon para muestras apareadas de la variable propiedades de usabilidad**

Como se puede apreciar en el resultado señalado, existe diferencia significativa entre los grupos de la muestra antes y después del pre-experimento en cuanto la variable **propiedades de usabilidad**, se puede concluir que hubo avance en los proyectos desde el inicio de la prueba hasta el final.

**Prueba estadística 2:** Haciendo uso de la herramienta SPSS.

Aplicando el test de *Wilcoxon* para dos muestras apareadas a la variable *patrones de usabilidad*, se muestra los resultados en la figura 19.

**Wilcoxon Signed Ranks Test**

		Ranks		
		N	Mean Rank	Sum of Ranks
números de patrones de usabilidad después	Negative Ranks	0 <sup>a</sup>	,00	,00
- números de patrones de usabilidad antes	Positive Ranks	6 <sup>b</sup>	3,50	21,00
	Ties	1 <sup>c</sup>		
	Total	7		

a. números de patrones de usabilidad después < números de patrones de usabilidad antes  
 b. números de patrones de usabilidad después > números de patrones de usabilidad antes  
 c. números de patrones de usabilidad después = números de patrones de usabilidad antes

Test Statistics <sup>b</sup>	
	números de patrones de usabilidad después - números de patrones de usabilidad antes
Z	-2,201 <sup>a</sup>
Asymp. Sig. (2-tailed)	,028

**Figura 19. Test de Wilcoxon para muestras apareadas de la variable patrones de usabilidad**

Como se puede apreciar en el resultado señalado, existen diferencia significativa entre los grupos de la muestra antes y después del pre-experimento en cuanto la variable **patrones de usabilidad**.

Después de haber aplicado las pruebas anteriores para las variables propiedades y patrones de usabilidad se evidencia que la inclusión de la aplicación de la propuesta introdujo cambios significativos en los proyectos de la muestra seleccionada determinado por un aumento del soporte de la usabilidad en la arquitectura del software.

**3.5 Análisis del impacto económico de la propuesta**

**3.5.1 Análisis costo- beneficio**

Un análisis de costo-beneficio podría ser un primer paso para la introducción de la usabilidad en una organización o un proyecto en particular. El análisis del costo beneficio de la usabilidad, tiene como objetivo demostrar que el costo de implementar varias actividades durante el proceso de desarrollo que garanticen la usabilidad, es menor que el costo de implementar estas actividades de forma tardía, es decir garantizar la usabilidad después que el sistema está desarrollado.

**3.5.1.1 Explicación del modelo utilizado para el cálculo del costo-beneficio**

El cálculo del análisis costo-beneficio de la propuesta se basará en el modelo que propone Nigel Bevan, Director de Desarrollo Profesional de la Asociación de Profesionales de Usabilidad (Saab, 2010). Bevan estima los beneficios potenciales de una mejor usabilidad para la

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

organización durante el desarrollo. Entre los beneficios potenciales identificados en el modelo se encuentra la reducción del costo de desarrollo por:

- Reducir el costo de rediseñar o el cambio radical de la arquitectura para versiones futuras del producto más usable. (Bevan, 2005)

El modelo de Bevan se basa en la norma ISO TR 18529 una versión mejorada de la ISO 13407. La primera describe siete procesos, cada uno de ellos contiene una conjunto de métodos con el fin de representar e incluir a los usuarios de un sistema durante el ciclo de vida del producto.

El procedimiento propuesto contiene varias actividades que coinciden con algunos de las métodos de la norma, más otras actividades relacionada con la inclusión de la usabilidad en la arquitectura del software, estas se muestran en la tabla 16:

**Tabla 16 : Equivalencia con los métodos de la norma ISO TR 18529**

Actividades del procedimiento	Equivalencia con los métodos de la norma ISO TR 18529	Descripción
Análisis del perfil del usuario	Análisis de las partes interesadas Contexto de usos	Se realizó talleres con los analistas de cada proyecto para identificar los objetivos de usabilidad, y cómo estos se relacionan con el negocio y criterios de éxito para el sistema, permitió priorizar los atributos de usabilidad, propiedades y beneficios de usabilidad.
Análisis de tareas	Escenario de usos Requerimientos de usabilidad	Se realizó talleres con los analistas para identificar las principales tareas en contextos específicos, para proporcionar una entrada para el diseño y una base para posteriores pruebas de usabilidad.
Determinar patrones y escenarios de usabilidad	No tiene equivalencia	Se realizó un taller para a partir de los atributos y propiedades de

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

		usabilidad identificados por los analistas, los arquitectos de cada proyectos identificaron que patrones y escenarios de usabilidad deben de tenerse en cuenta en el diseño arquitectónico.
Evaluación de los elementos de usabilidad a tener en cuenta en el diseño arquitectónico.	No tiene equivalencia	Se realizó entrevistas y talleres con los arquitectos de cada proyecto y se determinó los elementos de usabilidad que de cierta forma se habían implementado en el sistema y que patrón podía ser asimilado por cada escenario. Para proceder aplicar el método de evaluación de SALUTA.

La mayoría de los métodos de diseño centrado en el usuario son relativamente fáciles de aplicar. Los costos más importantes son el tiempo de las personas que aplican los métodos. (Bevan, 2005)

A continuación se muestra la tabla 17, propuesta por el modelo de Bevan que muestra el rango típico de esfuerzo y de personas requeridas para cada uno de los métodos, proporcionando una total que oscila entre el 26 y 80 persona- días. (Bevan, 2005)

**Tabla 17: Número de personas y días-hombre necesarias**(Bevan, 2005)

**CAPITULO 3: ANALISIS DE LOS RESULTADOS**

	Plan/ report (person days)		Execute (person days)		Total (person days)		Usability experts (number)		Managers/ developers (number)		Users (person days)	
	min	max	min	max	min	max	min	max	min	max	min	max
Maturity assessment	2	10	1	15	3	25	1	2	2	10		
1. Stakeholder meeting	1.5	3	.5	2	2	5	1	2	2	10	.5	2
2. Context of use	2	3	.5	2	2.5	5	1 <sup>1</sup>	2	2	6	.5	2
3. Scenarios of use	1	2	.5	1	1.5	3	1	2	1	4	.5	2
4. Baseline existing system	2	4	.5	4	2.5	8	2	2	0	4	.5	2
5. Usability requirements	1	2	.5	1	1.5	3	1	2	1	4	.5	2
6. Paper prototyping	3	6	1	6	4	12	2	2	0	6	.5	3
7. Style guide	1	10 <sup>2</sup>	.5	3	1.5	13	1	1				
8. Evaluate machine prototype	2	6	.5	6	2.5	12	1 <sup>1</sup>	2	0	6	.5	3
9. Test against requirements	2	7	2	6	4	13	2	2	0	6	.5	3
10. Feedback from use	3	5	1	5	4	6	1	1			* <sup>3</sup>	* <sup>3</sup>
Total	18.5	48	7.5	36	26	80						

Para algunos proyectos, incluso en el extremo inferior de esta escala puede ser demasiado ambicioso o sobrepasar el presupuesto disponible. Es necesario ajustar estos métodos teniendo en cuenta que el desarrollo del sistema CEDRUX se caracteriza por cronogramas de entrega ajustados, y escasa disponibilidad de los usuarios en el proceso de desarrollo.

**3.5.1.2 Modelo aplicado al costo-beneficio de la propuesta**

Para la estimación de costo de la propuesta es preciso conocer que en la universidad según la resolución 99 sobre organización salarial, los profesores de los centros dedican a la producción 33.3 horas semanales con una jornada diaria de 8 horas y una tarifa diaria de \$ 26.56. A continuación se muestra la tabla 18 con los costos de cada actividad después de haber aplicado el procedimiento.

**Tabla 18 : Costo por salario y número de personas-días de la propuesta**

### CAPITULO 3: ANALISIS DE LOS RESULTADOS

Actividades del procedimiento	Duración (horas)	Número de personas que participan	Tiempo de preparación y documentación (horas)	Total de número de personas- días	Tarifa diaria	Costo
Análisis de las partes interesadas	4	2 analistas	2	12/8 = 1,5	\$26.56	\$39.84
Contexto de usos	2	2 analistas	1	6/8=0.75	\$26.56	\$19.92
Escenario de usos	24	1 analista 1 arquitecto	8	64/8=8	\$26.56	\$212.48
Requerimientos de usabilidad	2	1 analista	1	3/8=0.37	\$26.56	\$9.82
Evaluar prototipo	16	2 analistas 3 funcionales	8	120/8=15	\$26.56	\$398.4
Determinar patrones de usabilidad	8	1 arquitecto	4	12/8=1.5	\$26.56	\$39.84
Determinar escenarios de usabilidad	8	1 arquitecto	4	12/8=1.5	\$26.56	\$39.84
Evaluación de los elementos de usabilidad a tener en cuenta e el diseño arquitectónico.	12	1 arquitecto 1 analista 1 especialista de calidad	4	48/8=6	\$26.56	\$159.36
<b>total</b>				<b>34.25 días-hombre</b>		<b>\$919.5</b>

➤ **Determinar cantidad de personas- días de cada actividad**

Para cada actividad, el total de personas-horas para llevarla a cabo esta dada por la siguiente fórmula:

**Cant personas-horas por actividades = (Tiempo de preparación x personas) + (Tiempo de aplicación x personas)**

Para el caso de la actividad Análisis de las partes interesadas para poder llevarlo a cabo se necesita 12 personas-horas, para convertirlo a personas- días, se divide entre 8h que es la jornada laboral. Aplicando la fórmula anterior quedaría:

Cant personas- horas por actividad = (Tiempo de preparación x personas) + (Tiempo de aplicación x persona) = (2 x 2) + (4 x 2) =4 + 8= 12 personas-horas

$$\frac{12 \text{ personas-horas}}{8h} = 1,5 \text{ personas-días}$$

El mismo procedimiento de calcular la cantidad personas-días se realizará para cada actividad y los resultados se muestran en la tabla 18.

➤ **Determinar costo por salario de la propuesta**

Para cada actividad el costo de desarrollarla estará dado por multiplicar la cantidad de persona-días por la tarifa diaria del salario de los involucrados.

Para el caso de la actividad Análisis de las partes interesadas, el costo que implica desarrollarla es de \$ 39.48, aplicando la fórmula siguiente:

**Costo por actividad= (Cant personas- días x tarifa diaria) = 1,5 personas-días x 26.56 = \$ 39.48**

El mismo procedimiento de calcular el Costo por actividad se realizará para cada una y los resultados se muestran en la tabla 18.

El costo por salario de la propuesta está dado por la sumatoria del costo de cada actividad.

**Costo por salario =  $\sum$  Costo por actividad=\$919.5**

**Otros costos:**

Para la aplicación de la propuesta se incurrieron en otros gastos como son los de materiales directos e indirectos referentes a la tecnología empleada y a los medios de oficinas.

Los materiales se evaluarán a su costo de adquisición:

**Tabla 19. Costo de adquisición de los recursos materiales**

Materiales Directos e indirectos	UM	Costo CUC	Costo CUP	Importe CUC	Importe CUP
Tóner P HP 2400	1	150	37.5	150	37.5
Papel 8 ½ x 11GR	2	0.516792	0.080528	1.033584	0.16116
P4 3.0 G ASUS P5 LD2 DDR 2 512	4	0.0	8 61.79	0.0	3447,16
Memoria externa USB 2GB	2	18.17	2.75	36,34	5.5
Electricidad			47.7396718		47.7396718
<b>Total</b>					3538.06

De lo antes señalado se puede calcular el costo aproximado de la aplicación de la propuesta, será de 4457.5 CUP como se muestra en la tabla 20.

**Tabla 20. Costo total de la propuesta**

Costos	CUP
Salario	919.5
Materiales directos e indirectos	3538.06

<b>Total</b>	<b>4457.5</b>
--------------	---------------

➤ **Determinar beneficios financieros económicos de la propuesta**

Algunos de los beneficios identificados en la norma ISO TR 18529, y que la propuesta trae consigo es la disminución de los costos de rediseñar o el de los cambios radicales de la arquitectura para versiones futuras del producto más usable y la reducción de costos por la detección y solución de problemas de usabilidad detectados en etapas tempranas en el proceso de desarrollo.

La disminución de los costos de rediseñar o el cambio radical de la arquitectura se ven expresados en los costos producidos durante la fase de pruebas, ante la petición del cliente de incorporar beneficios de usabilidad con impacto en la arquitectura. Esto es ocasionado porque durante la fase de toma de decisiones del diseño arquitectónico no se tuvieron en cuenta los escenarios de usabilidad.

Uno de estos escenarios es que para todos los módulos del sistema CEDRUX se puede contar con una funcionalidad que permita notificar a los clientes a través de alertas que emita el sistema sobre el estado de la información que se está registrando, debido a que es un sistema distribuido estas alertas deben ser configurables, al mismo tiempo el sistema no puede dejar de atender la petición para el que fue diseñado o sea, si un usuario realiza una acción sobre el sistema y esta acción desencadena un aviso, este aviso debe enviarse de forma concurrente a la ejecución de la acción.

Luego de un estudio exhaustivo por los arquitectos sobre la forma de garantizar multiprocesos en PHP se llegó a la conclusión de que implementar esta solución en el marco de trabajo requería del uso de nuevas tecnologías, en este caso se necesitó de un servidor de colas de mensajes, y la modificación de clases en el marco de trabajo para que pueda ejecutarse en modo de línea de interfaz de consola, lo que puede aumentar los riesgos durante el soporte del sistema. Para el desarrollo de estos elementos se requirió de 4 meses de trabajo con un equipo de 8 personas trabajando jornadas de 8 horas, con un aumento de los costo de \$12876,288 CUP por encima de lo planificado.

Otro ejemplo de aumento de los costos por la detención tardía de problemas de usabilidad es el caso de la no conformidad encontrada por la empresa CONABANA.SA durante la fase de pruebas de aceptación al sistema CEDRUX. En este sistema, específicamente en el subsistema

de Inventario, los desarrolladores aplicando semi-usabilidad<sup>1</sup> habían permitido que los usuarios registraran muchos productos antes de poderlos confirmar, esto en la usabilidad se conoce como el escenario agregación de objetos, que no es más que la capacidad que provee el sistema de agrupar varios datos(Productos) y aplicarles una misma acción(confirmar). El problema surgía cuando el sistema no le notificaba al usuario el límite máximo de productos a confirmar. Cuando los arquitectos analizaron el problema se percataron de que no se tuvo en cuenta una validación en que el sistema solo permitiera seleccionar hasta 20 productos, al llegar a éste límite, el sistema debía notificar a los usuarios de que al llegaron al margen máximo para procesar lotes y debía invalidar la opción de seleccionar productos para esta funcionalidad. Esto en usabilidad se conoce como “Validación”, este ejemplo muestra como la aplicación de patrones de usabilidad pueden colaborar para prevenir errores que pueden dejar al sistema sin respuesta y comprometer a otros atributos como el rendimiento. Esta funcionalidad requirió de 5 días con un desarrollador, lo que implicó un aumento de los costo en la fase de pruebas de \$123,8 CUP.

Por citar solo los dos ejemplos anteriores, el total de los beneficios financieros económico de los mismo está dado por :

$$\text{Beneficio financiero total} = \sum \text{beneficios financieros de cada ahorro identificado}$$

$$\text{Beneficio financiero total} = \$12876,288 + \$123,8 = 13009,088 \text{ CUP}$$

➤ **Calcular la relación costo-beneficio**

El costo-beneficio de la propuesta estará dado por :

$$\text{Costo – beneficio} = \frac{\text{Beneficio financiero total}}{\text{costo total}}$$

$$\text{Costo – beneficio} = \frac{\$13009,088}{\$4457.5} = \$2.9184 \text{ CUP}$$

Se concluye que como los beneficios financieros económico son mayores que el esfuerzo invertido, solo un costo de \$ 4457.5 CUP, la propuesta es rentable.

**3.5.2 Disminución de los costes de desarrollo**

El costo del desarrollo del sistema Cedrux (incluida su implementación) es equivalente a 920 970 CUP. Para calcular el costo de desarrollo de CEDRUX sin la aplicación del procedimiento

---

<sup>1</sup> Usabilidad alcanzada sin ninguna técnica de aseguramiento de la misma. Se logra con el sentido común y experiencia del equipo de desarrollo.

estará dado por la suma del costo de desarrollo más los beneficios financieros calculados con anterioridad:

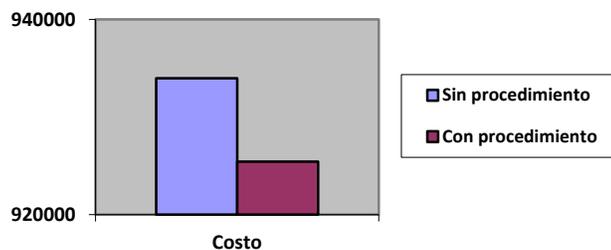
**Costo sin Procedimiento** = Costo de desarrollo CEDRUX + Beneficio financiero total = 920 970 CUP + 13000,088 CUP= **933 970,088 CUP**

Como se puede apreciar este aumentó en 13000,088 CUP por encima de lo planificado.

Sin embargo si se calcula el costo de desarrollo con la aplicación del procedimiento estará dado por :

**Costo con Procedimiento** = Costo de desarrollo CEDRUX + Costo de aplicación de la propuesta = 920 970 CUP+ 4457.5 CUP = **925 427.5 CUP**

Como se muestra en la figura 20 los costos con la aplicación del procedimiento disminuyen, por lo que la propuesta va a contribuir a la disminución de los costes de desarrollo del CEIGE.



**Figura 20. Comparación de los costos**

**3.6 Conclusiones parciales**

El análisis del soporte de la usabilidad, el pre-experimento realizado y el impacto descrito permiten arribar a:

- El uso del presente procedimiento, permite mediante el empleo de atributos, propiedades y patrones de usabilidad aumentar el soporte de la usabilidad en la arquitectura de los productos de software del CEIGE.
- El análisis del número de patrones y propiedades de usabilidad mediante el método de evaluación SALUTA en los proyectos de la muestra, demostraron que con la aplicación del procedimiento hubo solo tres escenarios con soporte bajo y aumentó el número de escenarios con soporte alto.
- El análisis costo -beneficio de la propuesta demostró ser rentable al ser mayor los beneficios financieros económico que el esfuerzo invertido.

### Conclusiones

- El análisis sobre los principales modelos y técnicas de evaluación de la usabilidad demuestran ser ineficientes en el aseguramiento de la usabilidad de un nuevo sistema desde sus inicios, se evidencia la necesidad del desarrollo de un procedimiento para el aseguramiento y evaluación de la usabilidad en todas las fases, incluido el diseño arquitectónico.
- Con el procedimiento para el aseguramiento y evaluación de la usabilidad de los sistemas de gestión del CEIGE se logra fortalecer la calidad de sus procesos de desarrollo y garantizar la usabilidad de sus productos. La misma consta de 3 fases, propuestas en la ingeniería de la usabilidad, que coinciden con las del modelo de desarrollo clásico y las establecidas en el centro: Análisis, Diseño, y lanzamiento.
- Las pruebas estadísticas utilizadas permitieron constatar del impacto positivo experimentado en las variaciones ocurridas en los proyectos pertenecientes a la población experimental.
- Después de realizar la valoración del impacto económica de la propuesta se llega a la conclusión de que la misma es rentable y contribuye a la disminución de los costes de desarrollo.

### Recomendaciones

Con el objetivo de dar continuidad a la presente investigación se recomienda:

- Profundizar la investigación con elementos de la arquitectura de la información con el propósito de abarcar más el campo de la usabilidad en todas sus variantes.
- Desplegar la vista de presentación por los centros de la universidad.
- Incorporar a la investigación nuevas métricas para la evaluación de la usabilidad.
- Desarrollar actividades en el procedimiento para incorporar patrones de usabilidad en arquitecturas ya diseñadas.
- Desarrollar actividades en el procedimiento para evaluar el impacto de la aplicación de los patrones de usabilidad en arquitecturas ya diseñadas.
- Conformar cursos de capacitación sobre temas relacionado con la usabilidad y la arquitectura para los arquitectos del centro.

## Bibliografía

1. Arciniegas, José L; Fernández, Verónica; Hormiga, Amparo; Tulande, Aleyda; Urbano. 2009. Proceso de requerimiento y análisis para la definición de la arquitectura desde la perspectiva de usabilidad para el desarrollo de aplicaciones en la Web. [En línea] 2 de septiembre de 2009. [Citado el: 12 de diciembre de 2011.] <http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=133113598023>. ISSN (Versión impresa): 1657-7663.
2. **Andrés, A. (2002)**. Identification of Usability Descomposition. Literature Survey and Industrial Experience). Ver. 1.0. From Proyecto STATUS [En línea] 22 de agosto de 2002 [Citado el: 13 de febrero de 2011.] <http://jcflowers1.iweb.bsu.edu/rlo/usability.htm>
3. **Christopher Alexander, Sara Ishikawa and Murray Silverstein. 1979**. *A Pattern Language Towns, Buildings, Construction (Cess Center for Environmental*. University of Houston : Ships from and sold by Amazon.com. Gift-wrap available., 1979. *by Oxford University Press, USA*
4. **Beatriz E. Florián, Oswaldo Solarte, Javier M. Reyes. 2010**. PROPUESTA PARA INCORPORAR EVALUACIÓN Y PRUEBAS DE USABILIDAD DENTRO DE UN PROCESO DE DESARROLLO DE SOFTWARE. Colombia : Revista EIA, 2010. ISSN 1794-1237 Número 13, p. 123-141. Julio 2010.
5. **Bevan, Nigel. 2005**. *Cost benefits evidence and case studies*. [TRUMP project was supported by the European Union.] España : s.n., 2005. D.J. (2005) Cost-Justifying Usability: An Update for the Internet Age.
6. **Bengtsson and Bosch, 1999**, P. O. Bengtsson and J. Bosch (1999), Architecture Level Prediction of Software Maintenance, Proceedings of the EuroMicro Conference on Software Engineering, IEEE, Amsterdam, Netherlands, page 139-147.
7. **Bengtsson, 2002**, P.O.Bengtsson (2002), Architecture-Level Modifiability Analysis, Department of Software Engineering and Computer Science, Blekinge Institute of Technology. Tekniska.Suecia
8. Bosch, J. (2000). Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach. Karlskrona/Ronneby: Addison-Wesley, 2000.
9. **Bevan, N.** "UsabilityNet Methods for User Centred Design. [En línea] julio de 2007. [Citado el: 3 de enero de 2011, de <http://www.usabilitynet.org/tools/13407stds.htm>.
10. **Beatriz E. Florián, Oswaldo Solarte, Javier M. Reyes. 2010**. PROPUESTA PARA INCORPORAR EVALUACIÓN Y PRUEBAS DE USABILIDAD DENTRO DE UN PROCESO DE DESARROLLO DE SOFTWARE. Colombia : Revista EIA, 2010. ISSN 1794-1237 Número 13, p. 123-141. Julio 2010.

11. **Bevan, Nigel. 2005.** *Cost benefits evidence and case studies.* [TRUMP project was supported by the European Union.] España : s.n., 2005. D.J. (2005) Cost-Justifying Usability: An Update for the Internet Age. Hand Court. London.
12. **Bauler, Tom. 2011.** An analytical framework to discuss the usability of (environmental) indicators for policy. 2011. [En línea] 12 de junio de 2011. [Citado el: 12 de enero de 2012.] <http://www.sciencedirect.com/science/article/pii/S1470160X11001440>.
13. **Bettina Biel, Thomas Grill, Volker Gruhna. 2010.** Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application. Austria : The Journal of Systems and Software, 2010, Vols. ICT& S Center, University of Salzburg, Sigmund-Haffner G. 18, 5020 Salzburg, Austria. journal homepage: [www.elsevier.com/locate/jss](http://www.elsevier.com/locate/jss).
14. **Casanovas, J.** Usabilidad y arquitectura del software. [En línea] 31 de julio de 2004 [Citado el: 23 de enero de 2011.] [http://www.alzado.org/articulo.php?id\\_art=355](http://www.alzado.org/articulo.php?id_art=355),
15. **D., M. Eric Johnson and Nicholas. 2011.** Usability Failures and Healthcare Data Hemorrhages. [En línea] abril de 2011. [Citado el: 23 de septiembre de 2011.] 1540-7993/11/\$26.00 © 2011 IEEE.
16. **Folmer et al, 2005, E. Folmer, M. Welie, and J. Bosch (2005),** "Bridging Patterns- an approach to bridge gaps between SE and HCI", Accepted for the Journal of Information & Software Technology. EEUU.The Netherlands
17. **Grau, Xavier Ferré. 2009.** Principios Básicos de Usabilidad para Ingenieros Software. España : 28660 – Boadilla del Monte (Madrid), 2009.
18. **Héctor F. Alarcón, Adrián M. Hurtado, César A. Collazos. 2007.** *Integración de Técnicas de Usabilidad y Accesibilidad en el Proceso de Desarrollo de Software de las MiPyMEs.* Colombia : Grupo IDIS, Universidad del Cauca, Colombia, Grupo ALARCOS, Universidad Castilla La Mancha, España, 2007. Revista Avances en Sistemas e Informática, Vol.4 No. 3, Medellín, 156 Diciembre de 2007, ISSN 16577663.
19. **Henrik Pestano Pino.** Propuesta de modelo de desarrollo para líneas de productos de software en centros de producción. La Habana, Cuba : Universidad de las Ciencias Informáticas, Laboratorio de Gestión de Proyectos.2011
20. **IEEE, 1998, IEEE, IEEE Architecture Working Group.** Recommended practice for architectural description. Draft IEEE Standard P1471/D4.1, IEEE, 1998: Massachusetts, EEUU.

21. IEEE. 1990. Standard Glossary of Software Engineering Terminology. [En línea] 28 de septiembre de 1990. [Citado el: 13 de enero de 2011.] [:http://www.idi.ntnu.no/about/groups.php](http://www.idi.ntnu.no/about/groups.php): Massachusetts, EEUU.
22. **Javier M. Reyes Vera, Freddy Alejandro Libreros Giraldo. 2011.**MÉTODO PARA LA EVALUACIÓN INTEGRAL DE LA USABILIDAD EN SISTEMAS E-LEARNING.. Diciembre de 2011 N°. 12 pp. 69-79, Colombia: Publicada en línea por la Asociación Colombiana de Facultades de Ingeniería -ACOFI- [www.acofi.edu.co](http://www.acofi.edu.co), 2011, Vols. ISSN 1900-8260.: Cartagena, Colombia.
23. Yusef Hassan Montero, F. J. (2004). Propuesta de adaptación de la metodología de diseño centrado en el usuario para el desarrollo de sitios web accesibles. Revista española de Documentación Científica, Vol 27, No 3 (2004) : Madrid, España.
24. **Jakob Nielsen. 1994.** Ten Usability Heuristics. [En línea] 3 de abril de 1994. [Citado el: 23 de septiembre de 2011.] [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html).
25. **Turkka K. Keinonen, 1. V. (2008).** Three-in-One User Study for Focused Collaboration. international Journal of Design, Home > Vol 2, No 1.
26. **León, Berenice Ruiz y Luis Vinicio. 2010.** Pruebas de usabilidad en sitios web. [En línea] 22 de agosto de 2010. [Citado el: 13 de febrero de 2011.] <http://www.sg.com.mx/content/view/1088>.
27. **Mario Carvajal y Juan Saab. 2010.** *Fundamentos conceptuales de las Directrices de Usabilidad de Gobierno en línea*. Colombia : CINTEL, 2010. Versión: 1.0.4: Cartagena, Colombia.
28. **Macleod, 1994, M. Macleod (1994),** Usability: Practical Methods for testing and Improvement, Proceedings of the Proceedings of the Norwegian Computer Society Software '94 Conference, Sandvika, Norway.
29. **MPIu+a.** "Modelo de Proceso de la Ingeniería de la Usabilidad "[En línea] julio de 2009. [Citado el: 2 de marzo de 2011], de <http://griho.udl.es/mpiu/index.htm>.
30. **Marín, J. M. 2003.** Ingeniería de la Usabilidad aplicada al desarrollo de un portal web administrado dinámicamente. [Online] septiembre 5, 2003. [Cited: marzo 21, 2011.]
31. **Natalia Juristo a, Ana M. Moreno a,\* , Maria-Isabel Sanchez-Segura. 2007.** *Analysing the impact of usability on software design*. España : School of Computing, Universidad Politécnica de Madrid, Campus de Montegancedo, 20 de junio de 2007. 30, 28911 Leganes, Madrid, Spain.

32. **Osmar, Leyet.** *Documento de descripción de la arquitectura de software.* La Habana, Cuba : Universidad de las Ciencias Informáticas, 2010
33. **Obeso, María Elena Alva.** Tesis doctoral :Metodología de Medición y Evaluación de la Usabilidad en Sitios Web Educativos. [En línea] 5 de marzo de 2005. [Citado el: 28 de febrero de 2011.], [www.aipo.es/articulos/2/30.pdf](http://www.aipo.es/articulos/2/30.pdf): Madrid, España
34. **Rossi, Alejandra Garrido and Gustavo.** 2011. *Refactoring for Usability in Web Applications.* Universidad Nacional de La Plata : Damiano Distanto, Unitelma Sapienza University, 2011. MAY/JUNE 2011 | IEEE SO.
35. **Roger S. Pressman, Ph.D.** 1992. *Software Engineering A PRACTITIONER'S APPROACH.* New York, NY : McGraw-Hill Higher Education, 1992. ISBN:0-070-50783-X.
36. **Sánchez-Segura, A. M.** (2009). Mejora de la Usabilidad del Software desde el momento de Arquitectónico. Universidad Politécnica de Madrid, España.
37. **Svahnberg et al,** 2000, M. Svahnberg, C. Wohlin, L. Lundberg, and M. Mattson (2000), A method for understanding quality attributes in software architecture structures, Proceedings of the Proceedings of the 14th international conference on Software engineering and knowledge engineering, SESSION: Workshop on software engineering decision support., Ischia, Italy, pages 819-826.
38. **Shackel,** 1991, B. Shackel (1991), Usability - context, framework, design and evaluation, in: B. Shackel and Richardson, S. (eds.), Human Factors for Informatics Usability, Cambridge University Press-. Cambridge, Inglaterra.
39. **Sánchez., C.** *Sirius. Nuevo sistema para la evaluación de la usabilidad web.* [En línea] 06 de septiembre de 2011. [Citado el: 23 de enero de 2012], de <http://olgacarreras.blogspot.com/2011/07/sirius-nueva-sistema-para-la-evaluacion.html>
40. **Landauer, Thomas K.** 1995. *The Trouble With Computers: Usefulness, Usability, and Productivity".* Cambridge : MIT Press, 1995.: Cambridge, Inglaterra.
41. **Travis, D.** *Heuristic Evaluation with Morae.* [En línea] 5 de junio de 2009 [Citado el: 21 de enero de 2012], de <http://www.userfocus.co.uk/articles/morae-he.html>
42. **Torrente, María del Carmen Suárez.** 2011. *TESIS DOCTORAL.SIRIUS: Sistema de Evaluación de la Usabilidad Web Orientado al Usuario y basado en la Determinación de.* España : Oviedo, Febrero de 2011.

43. **UsabilityNet.** Cost justifying Usability. [En línea] 3 de abril de 2003[Citado el: 22 de septiembre de 2011.] [http://www.usabilitynet.org/management/c\\_cost.htm](http://www.usabilitynet.org/management/c_cost.htm)
44. **Vivas, N.** (9 de enero de 2009). La importancia de las pruebas de usabilidad. Recuperado el 23 de marzo de 2011, de <http://www.usarte.org/profiles/blogs/1956624:BlogPost:131>
45. Wixon, D. and Jones, S. 1995. Usability for fun and profit: A case study of the re-design of the. [En línea] 1 de febrero de 1995. [Citado el: 8 de octubre de 2011.]: an Francisco, CA, EEUU.
46. Arquitectura de Software.Vista Presentación.Universidad de Ciencias Informática 2010 Programa Mejora. [calisoft.uci.cu/index.php/proceso-de-mejora/46](http://calisoft.uci.cu/index.php/proceso-de-mejora/46)

## Anexos

Algunos anexos por la extensión de los mismos se entregarán como artefactos entregables de la propuesta.

### Anexo 1: Diagnóstico inicial de la usabilidad aplicado a varios centros de la universidad

**Nota aclaratoria:** Altamente confidencial, solo se presentarán resultados globales, nunca específicos de una entidad o persona sin su previa consulta y autorización. El manejo de este material será restringido al grupo de investigación.

Centro/empresa:      Fecha:      Entrevistado:      Cargo/Rol:

1 - ¿Qué importancia le confiere a la usabilidad en los productos de software? \_\_\_ Alta \_\_\_ Baja \_\_\_ Media \_\_\_ no tiene importancia

2. La usabilidad tiene impacto en: \_\_\_ Interfaz de usuario \_\_\_ Funcionalidad \_\_\_ Satisfacción del cliente \_\_\_ Arquitectura de Software \_\_\_ Calidad del producto \_\_\_ Éxito del proyecto

Ordene los anteriores planteamientos dándoles un peso (1-6) a los que consideres que tiene un mayor impacto.

3. ¿En la identificación de los requisitos No funcionales del sistema, tienen en cuenta la usabilidad? ¿En qué términos describen la usabilidad?

4. ¿Se identifican en el producto requisitos o escenarios que se relacionen directamente con la usabilidad, como: deshacer, cancelar, información del estado del sistema.etc.? ¿Están documentados, en qué artefactos?

5. ¿En qué etapa del ciclo de vida de producto se evalúa la usabilidad? \_\_\_ Levantamiento de Requisitos \_\_\_ Análisis arquitectónico \_\_\_ Diseño arquitectónico \_\_\_ Pruebas de concepto de la arquitectura \_\_\_ Pruebas de software (Unitarias, integración, sistema, Funcionales, Carga y estrés) \_\_\_ Aceptación del producto.

6. Han detectado problemas de usabilidad que hayan afectado la satisfacción del usuario final.

Argumente. Además consideras que los problemas de usabilidad pueden traer como consecuencias: \_\_\_ Pérdida de la calidad del producto (Competitividad, Satisfacción del cliente, Mejora continua) \_\_\_ Experiencia no documentada \_\_\_ Falta a compromisos \_\_\_ Baja usabilidad \_\_\_ Pérdida de oportunidad de negocio \_\_\_ Pruebas insatisfactorias \_\_\_ Bajo nivel de reutilización en: \_\_\_ Clases, \_\_\_ Código, \_\_\_ Documentación

\_\_\_ Incremento de los costos por: \_\_\_ Dificultad de integración, \_\_\_ Dificultad en el despliegue, \_\_\_ Hito de pago afectado, \_\_\_ Costosas licencias para el cliente, \_\_\_ Costosas licencias de desarrollo \_\_\_ Costos en capacitación y certificación, \_\_\_ Desarrollo de componentes no previstos, \_\_\_ Duplicidad de esfuerzo

\_\_\_ Incremento del tiempo de desarrollo por: \_\_\_ Retraso, \_\_\_ Hito incumplido

**Anexo 2: Instrumento para la captación de datos de la usabilidad**

Nombre de la persona: \_\_\_\_\_ Proyecto: \_\_\_\_\_ Rol: \_\_\_\_\_ Periodo: \_\_\_\_\_ (Inicial o Final)

**1. Se realiza una caracterización del producto desde la perspectiva del usuario en la fase de análisis y antes del diseño arquitectónico en cuanto a :**

\_\_\_\_ Conocimiento del dominio \_\_\_\_ Conocimiento del sistema \_\_\_\_ Frecuencia con la que el usuario emplea el sistema \_\_\_\_ Complejidad de las tareas a realizar por el usuario \_\_\_\_ Quién dirige la interacción usuario proceso \_\_\_\_ Determinación de dónde se origina la información y cómo fluye entre el sistema y el usuario o viceversa

**2. Indique con valores entre 1-10 la prioridad de los siguientes atributos de calidad del producto en el diseño de la arquitectura del software o en el análisis.**

Usabilidad \_\_, Integridad \_\_, Confiabilidad \_\_, Rendimiento \_\_, Flexibilidad \_\_, Eficiencia \_\_, Mantenibilidad \_\_, Robustez \_\_, Funcionalidad \_\_

**3. Se identifican en el sistema requisitos de usabilidad, marque con una x los que están presentes en el sistema.**

\_\_\_\_\_ El sistema debe ser capaz de conocer el estado de los comandos en ejecución, en caso que el sistema falle, ya que los usuarios deben ser informados si la acción no funciona a causa de circunstancias externas.

\_\_\_\_\_ El sistema debe ser capaz de comprobar el estado de los recursos internos del sistema y alertar a los usuarios con antelación para salvar a su trabajo.

\_\_\_\_\_ Capacidad del software para interactuar con los usuarios en diferentes idiomas

\_\_\_\_\_ El sistema debe permitir en caso de un error, debe informar al usuario para tomar medidas. Las alertas deben ser emitidos por medio de mensajes visible, que el usuario pueda ver.

\_\_\_\_\_ Si el usuario comete un error al entrar en cualquiera de los datos o quiere volver a los valores predeterminados, el sistema debe permitir al usuario volver a los valores introducidos anteriormente

. \_\_\_\_\_ El sistema les permite a los usuarios mover o copiar datos de forma manual o automática de una parte del sistema, lo que permite su reutilización.

\_\_\_\_\_ El sistema debe permitir que los parámetros de la consulta deban ser validados.

\_\_\_\_\_ Cuando se solicita un informe, que debería ser posible solicitar una vista previa del resultado antes de dar la orden de impresión.

\_\_\_\_\_ El sistema permite a los usuarios crear y guardar una secuencia de comandos o acciones que pueden ejecutarse en diferentes objetos.

\_\_\_\_\_ El sistema registra las acciones realizadas por los usuarios, lo que les permite ver el registro de lo que han hecho.

\_\_\_\_\_ Durante el proceso de búsqueda de datos, el sistema debe informar sobre su estado para que el usuario no intente ejecutar una consulta en la misma ventana. Los usuarios deben estar informados de la situación actual del sistema.

\_\_\_\_\_ El usuario debe ser capaz de cancelar acciones, mientras se está procesando.

\_\_\_\_\_ El usuario debe ser capaz de manejar varias tareas al mismo tiempo, y cambiar de una tarea a otra.

\_\_\_\_\_ El sistema permite a los usuarios seleccionar varios objetos en el que se ejecuta una y la misma acción al mismo tiempo.

\_\_\_\_\_ El sistema crea y guarda perfiles de usuario para poder recuperar los atributos específicos del sistema asociado con un usuario cada vez que este usuario utiliza el sistema.

\_\_\_\_\_ El sistema debe proporcionar a los usuarios información específica que muestran cómo realizar efectivamente una tarea dada.

\_\_\_\_\_ El sistema debe ofrecer a los usuarios información y ayuda en cualquiera de las tareas que deben realizarse utilizando el sistema.

\_\_\_\_\_ Presenta al usuario una secuencia estructurada de pasos para llevar a cabo una tarea, proporcionando al usuario una guía para cada paso. El usuario puede volver atrás y cambiar cualquiera de las etapas anteriores en el proceso

**4. Se realizan la especificación de los requisitos de usabilidad \_\_\_\_\_ Si \_\_\_\_\_ No**

**5. Si se identifican atributos de usabilidad marque con una x los atributos que se identificaron en el sistema en la fase de análisis o en el diseño arquitectónico**

\_\_\_\_\_ Facilidad de aprendizaje: Qué tan fácil esta tarea es entender

\_\_\_\_\_ Eficiencia: El tiempo que tarda en realizar esta tarea.

\_\_\_\_\_ Satisfacción: El usuario debe sentir que tiene el control

\_\_\_\_\_ Confiabilidad: No debe ocurrir errores en realizar tarea

\_\_\_\_\_ No se identificó ningún atributo de usabilidad en la fase de análisis

**6. Se realiza la priorización de los atributos de usabilidad en conjunto con el usuario. \_\_\_\_\_ Si \_\_\_\_\_ No**

**7. Se confecciona los prototipos de IU \_\_\_\_\_ Si \_\_\_\_\_ No**

**8. Se validan los prototipos de IU mediante un test de usabilidad con el cliente. \_\_\_\_\_ Si \_\_\_\_\_ No**

**9. Que elementos se tienen en cuenta para construir y evaluar una arquitectura**

\_\_\_ Requisitos funcionales. \_\_\_ Requisitos no funcionales. \_\_\_ Atributos de usabilidad \_\_\_ Restricciones de diseño. \_\_\_ Propiedades de usabilidades \_\_\_ Elementos o componentes ya desarrollados. \_\_\_ Requisitos de usabilidad \_\_\_ Su entorno de desarrollo. \_\_\_ Propiedades de rendimiento \_\_\_ Los aspectos legales. \_\_\_ Las competencias del equipo. \_\_\_ Las vistas arquitectónicas. \_\_\_ Beneficios de usabilidad

10. Se identifican patrones de usabilidad en el diseño de la Arquitectura del software \_\_\_\_\_ Si  
\_\_\_\_\_ No

11. Marque con una x los patrones de usabilidad que se tuvieron en cuenta para el diseño de la arquitectura

\_\_\_ **Diferentes lenguajes:** Capacidad del software para interactuar con los usuarios en diferentes idiomas

\_\_\_ **Indicación del estado:** Los usuarios estén informados de la situación actual del sistema.

\_\_\_ **Forma de validación de campo:** Cuando un usuario rellena en varios campos, el sistema puede validar estos campos por completo cuando se envían los datos (validación de formularios) o de forma individual cada vez que se llena un campo (validación de campo).

\_\_\_ **Deshacer:** La capacidad del sistema para deshacer una acción y volver al estado anterior.

\_\_\_ **Ayuda Contextual:** Sensibles al contexto supervisa; ayudan a lo que el usuario está haciendo actualmente y proporciona información relevante para la finalización de la tarea en cuestión.

\_\_\_ **Asistente:** Presenta al usuario una secuencia estructurada de pasos para llevar a cabo una tarea, proporcionando al usuario una guía para cada paso. El usuario puede volver atrás y cambiar cualquiera de las etapas anteriores en el proceso

\_\_\_ **Historial de registro:** El sistema registra las acciones realizadas por los usuarios, lo que les permite ver el registro de lo que han hecho. El sistema registra las acciones realizadas por los usuarios, lo que les permite ver el registro de lo que han hecho.

\_\_\_ **Vista previa:** El sistema permite a los usuarios ver los resultados de una acción antes de ejecutar esta acción.

\_\_\_ **Perfil de usuario:** El sistema crea y guarda perfiles de usuario para poder recuperar los atributos específicos del sistema asociado con un usuario cada vez que este usuario utiliza el sistema.

\_\_\_ **Cancelación:** Los usuarios pueden cancelar lo que están haciendo, si se dan cuenta que ellos han hecho algo mal antes de un estado de error se alcanza

\_\_\_ **Multi –tarea:** La situación en la que el sistema (y el usuario) puede manejar varias tareas al mismo tiempo, se cambia de una tarea a otra.

\_\_\_ **Acción sobre varios objetos:** El sistema permite a los usuarios seleccionar varios objetos en el que se ejecuta una y la misma acción al mismo tiempo.

\_\_\_ **Agregación de comandos:** El sistema permite a los usuarios crear y guardar una secuencia de comandos o acciones que pueden ejecutarse en diferentes objetos.

\_\_\_ **Reutilización de la información:** Este patrón permite a los usuarios mover o copiar datos de forma manual o automática de una parte del sistema, lo que permite su reutilización

\_\_\_ **Apoyo de búsqueda completa:** proveerá la capacidad de permitir a los usuarios buscar datos de una manera integral y coherente por parte de los criterios pertinentes en el sistema.

**12. Valora el impacto en la arquitectura de implementar estos patrones en etapa tardía del desarrollo en cuanto costo de desarrollo** \_\_\_ Alto \_\_\_ Medio \_\_\_ Bajo

**13. Marque con una "X" si ha tenido afectaciones causadas por decisiones arquitectónicas en alguno de los siguientes aspectos**

\_\_\_ La satisfacción del cliente, y la evolución y mejora continua del producto se ha visto afectada \_\_\_  
Los elementos relacionados con los costos, los ingresos o la dependencia tecnológica se han visto afectados

\_\_\_ Los niveles de reutilización de las partes del producto y de componentes ya existentes no han permitido mejorar la productividad

\_\_\_ El cronograma pactado con el cliente (interno o externo) se ha visto afectado por rediseño exigido por cambios del cliente

**Anexo 3: Vista de presentación del expediente de arquitectura.**

**(Como artefacto entregable )**

Se encuentra en la siguiente dirección:

[http://gespro.prod.uci.cu/polls/polls\\_box?project\\_id=135&temario\\_id=103](http://gespro.prod.uci.cu/polls/polls_box?project_id=135&temario_id=103)

**Anexo 4: Formulario para la evaluación de la usabilidad en los prototipos de IU. (Como artefacto entregable )**

**Anexo 5: Catálogo de patrones de usabilidad (Como artefacto entregable )**

**Anexo 6: lista de chequeo de usabilidad para determinar el grado de usabilidad una vez implementado el sistema. (Como artefacto entregable )**

**Anexo 7: Catálogo de patrones de IU. (Como artefacto entregable )**

**Anexo 8: Acta de aceptación de la vista de presentación.**

21 de Mayo de 2012  
 Año 54 de la Revolución

#### Aval de introducción y generalización

Producto: **Vista de presentación del Modelo Arquitectónico de Referencia**

#### Resumen

Como resultado de la investigación se obtiene la vista de presentación que forma parte de la línea base de la arquitectura de la Universidad de las Ciencias Informáticas (UCI) y se aplica en todos los centros de desarrollo de la misma.

Además se obtiene un procedimiento para aseguramiento y evaluación de la usabilidad de sistemas informático, cuenta de tres fases que son las propuestas por la Ingeniería de la Usabilidad.

La propuesta realizada tiene un importante aporte práctico, su aplicación es muy efectiva para ayudar a detectar problemas de usabilidad temprano en el ciclo de desarrollo, cuando son más fáciles y menos costosos de solucionar.

Esta aproximación difiere de la idea tradicional de medir y mejorar la usabilidad una vez finalizado el sistema.

#### Introducción y generalización

Este producto de trabajo ha sido introducido en la Red de centros de desarrollo de la UCI a través del consejo técnico asesor de la Red evidenciadas en un conjunto de acuerdos y actividades lo que avala la introducción y generalización del mismo.

Consejo técnico del 8 de diciembre de 2011

- Punto 4. Propuesta de Vista de Presentación y su relación con la Dirección de Comunicación Visual.
- Acuerdo 01.Arq.08.12.11: La DGP debe circular Propuesta de Vista de Presentación y su relación con la Dirección de Comunicación Visual. Los centros deben enviar las inconformidades con la propuesta enviada y proponer su idea al respecto. Resp: DGP, red de centros, enero 2012.
- Acuerdo 03.Arq.08.12.11: Proponer a la Dirección de Comunicación Visual que garanticen que su propuesta de diseño funcionará en cualquier escenario (mobilla, explorer, etc). Resp: DGP, enero 2012.

Consejo técnico del 1 de marzo de 2012

- Punto 1. Avances en la implantación de la vista de presentación. (Resp: DGP (Iliannis)). Despliegue del escenario genérico. Estandarización de la imagen gráfica de la UCI (para productos). Definición



**DIRECCIÓN GENERAL DE PRODUCCIÓN**  
 Universidad de las Ciencias Informáticas  
 Carretera a San Antonio Km 2 1/2, Tomera,  
 Boyeros, Ciudad de La Habana, Cuba  
 Teléfono: + 53 (7) 637 3680  
 E-mail: software.gestion@uci.cu

de etiquetado de sistemas, Manuales, Glosario de términos, Componentes, etc. Zonificación de las pantallas. Ej: Autenticación/Home.

Consejo técnico del 8 de marzo de 2012

- Punto 3. Revisar avances en la Vista de Presentación.
- Acuerdo 01.Arq.08.03.12: Circular información de la vista de presentación (patrones asociados), los centros deben enviar sus impresiones. Responsable DGP, red de centros. Febrero 2012.

Consejo técnico del 15 de marzo de 2012

- Punto 2. Revisar avances en la Vista de Presentación.
- Acuerdo 01.Arq.15.03.12: Los centros deben enviar sus criterios sobre el documento de la vista de presentación circulado la semana pasada. Responsable: red de centros. 22 marzo 2012.

Consejo técnico del 29 de marzo de 2012

- Punto 4. Revisar Vista de Presentación.

Consejo técnico del 26 de abril de 2012

- Punto 5. Temas generales. (Avances próximo despliegue de Vista de Presentación.)



**Msc. Michael González Jorin**  
 Coordinador del Grupo de asesoría técnica



**Ing. Liset Feria González**  
 Coordinadora de los consejos  
 técnico