



FACULTAD 8

PROPUESTA Y ANÁLISIS DE LA ARQUITECTURA DE INFORMACIÓN EN EL PROYECTO CICPC

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Autora: Mailin Carballosa Infante

Tutores: Lic. Deymis Tamayo Rueda

Ing. Sasha Valdés Jiménez

Ciudad de La Habana, 2008

“Año 50 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora del presente Trabajo de Diploma y reconozco a la Facultad 8 de la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2008.

Firma de la Autora
Mailin Carballosa Infante

Firma de la Tutora
Deymis Tamayo Rueda

Firma del Tutor
Sasha Valdés Jiménez

DATOS DE CONTACTO

Deymis Tamayo Rueda. Licenciada en Bibliotecología y Ciencias de la Información, graduada en la Facultad de Comunicación de la Universidad de La Habana. 2 años de experiencia. Especialista en Arquitectura de Información. Brinda servicios al Departamento de Ingeniería de Software y Práctica Profesional y pertenece al Departamento de Humanidades de la Facultad 8 de la referida institución. Residencia UCI: Edificio 41, apto 203. Teléfono: 835-8836. Correo electrónico: drtamayo@uci.cu

Sasha Valdés Jiménez. Ingeniero en Ciencias Informáticas, graduado de la Universidad de las Ciencias Informáticas. 2 años de experiencia. Especialista en Análisis de Sistemas. Pertenece al Departamento de Ingeniería de Software y Práctica Profesional de la Facultad 8 de la referida institución. Residencia UCI: Edificio 26, apto 108. Teléfono: 835-8785. Correo electrónico: sasha@uci.cu

*"Hay tanto que decir, que ha de decirse en el menor número de palabras posible:
eso sí, que cada palabra lleve ala y color".*

José Martí

AGRADECIMIENTOS

...A La Revolución Cubana y al Comandante Fidel Castro por hacer realidad el sueño de convertirme en una profesional...A La Universidad de las Ciencias Informáticas (UCI) por los momentos vividos...A mis padres, por darme siempre lo mejor de sí, apoyarme en todo momento y por ser mi guía en la vida...A mi novio, por su comprensión y amor que me hace seguir adelante todos los días de mi vida...A mis abuelos, Cachita y Oscar, por ser únicos...A mi tutora, Deymis, por ser inigualable, por sus incontables consejos, tiempo brindado, sabiduría y ayuda a lo largo del desarrollo del trabajo...A mi tutor, Sasha, por los consejos y sugerencias en todo momento, por nunca estar conforme con nada y mostrarme que puede ser mejor... A toda mi familia en general, por su cariño, fe en que puedo hacer todo lo que me proponga...A mi hermano, primos y generaciones futuras, primera universitaria sí, pero no quiero ser la única...A Yadenis, porque esta tesis también es suya ...A todos mis amigos y compañeros, gracias por estar siempre...A Lazarita, por ser mi madre en la UCI y brindarme su cariño y a Frank también, por sus consejos...A los profesores del proyecto y de la facultad por sus recomendaciones, sugerencias en todo momento...A todos los que de una manera u otra, me han extendido su mano y han puesto en mí un granito de tesón, dándome fuerzas para seguir luchando por este sueño que hoy se cumple.

A todos, les doy mis más sentidos agradecimientos.

A mis Padres, por ser los mejores del mundo

RESUMEN

El inminente avance de las ciencias informáticas y la necesidad de desarrollo industrial y económico, han provocado el crecimiento de la industria cubana del software. La Universidad de las Ciencias Informáticas (UCI) promueve el desarrollo de la informática tanto para el ámbito nacional como internacional, mediante la realización de proyectos productivos que vinculan estudiantes, profesores y profesionales afines a la producción de software. Uno de estos proyectos es precisamente el plan de modernización del Sistema Integrado de Información Policial (SIIPOL), desarrollado para el Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC), de la República Bolivariana de Venezuela, con el objetivo de solucionar problemas generados por un control deficiente de la información en dicha organización.

La inserción de la disciplina Arquitectura de Información (AI) dentro del proceso de desarrollo de software es de total premura para lograr un producto de alta calidad que cumpla con principios de usabilidad y accesibilidad para el usuario final de la aplicación.

El presente trabajo constituye una propuesta de Arquitectura de Información aplicada al Proyecto CICPC, encaminada a la obtención de un software funcional de alta calidad que genere en el usuario un gran impacto. Además se establecen aportes teóricos que insertan la Arquitectura de Información dentro de la Metodología de Proceso Unificado de Software (RUP).

Palabras Clave

Arquitectura de Información, Calidad de Software, Usabilidad, Prototipado de Interfaz de Usuario, Proceso Unificado de Software

TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA	5
1.1 Introducción.....	5
1.2 Arquitectura de Información.....	5
1.2.1 Antecedentes	5
1.2.2 Algunas Definiciones	5
1.2.3 Importancia de la Arquitectura de Información	7
1.2.4 Arquitectura de Información como proceso gerencial.....	8
1.2.5 Arquitectura de Información y su relación con RUP	8
1.3 El Prototipo de Interfaz.....	13
1.3.1 Tipo de Prototipos	14
1.3.2 Niveles de Prototipado.....	15
1.3.3 Fases en el Proceso de Prototipado	16
1.3.4 Objetivos del Prototipo.....	17
1.3.5 Técnicas para la Aplicación de Prototipos	18
1.3.6 Estrategias de Diseño de Prototipos de Interfaces de Usuario	20
1.3.7 Papel del Prototipo.....	21
1.3.8 Usabilidad	22
1.4 Aplicaciones y Herramientas para Diseñar Prototipos	22
1.4.1 Normas que Respaldan el uso de Prototipos.....	23
1.4.2 Herramientas para realizar Prototipos.....	23
1.5 Calidad de Software	24
1.5.1 Listas de Chequeo como instrumento de Validación	25

1.6	Técnicas de Recopilación de Información.....	25
1.7	Rol del Arquitecto de Información.....	26
1.8	Conclusiones.....	26
CAPÍTULO 2. PROPUESTA DE ARQUITECTURA DE INFORMACIÓN		28
2.1	Introducción.....	28
2.2	Arquitectura de Información y RUP	28
2.3	Levantamiento de Información.....	30
2.3.1	Actividades en el Levantamiento de Información.....	32
2.3.2	Artefactos generados en el Levantamiento de Información	37
2.4	Desarrollo de Pautas de Arquitectura de Información	38
2.4.1	Definición de Pautas de Arquitectura de Información.....	39
2.4.2	Definición de Componentes visuales	40
2.4.3	Artefactos Generados en el Desarrollo de Pautas de Arquitectura de Información	41
2.5	Diseño de Prototipos de Interfaces de Usuario	45
2.5.1	Artefactos Generados	47
2.6	Evaluación de la Arquitectura de Información.....	48
2.7	Conclusiones.....	51
CAPÍTULO 3. ANÁLISIS DE RESULTADOS Y VALIDACIÓN DE LA PROPUESTA.....		53
3.1	Introducción.....	53
3.2	Aplicación de encuestas al Equipo de Desarrollo	53
3.2.1	Análisis de Resultados de las encuestas realizadas antes de la aplicación de la propuesta de Arquitectura de Información	54
3.2.2	Análisis de Resultados de las encuestas realizadas después de la aplicación de la propuesta de Arquitectura de Información	54
3.2.3	Tiempo de Desarrollo.....	55
3.3	Guía para la Validación por Expertos	56

3.4 Conclusiones.....	62
CONCLUSIONES	63
RECOMENDACIONES.....	64
REFERENCIAS BIBLIOGRÁFICAS	65
BIBLIOGRAFÍA	67
GLOSARIO DE TÉRMINOS	68
ANEXOS	70
Anexo I: Documento de Pautas de Arquitectura de Información del Proyecto CICPC	70
Anexo II: Librería de Componentes para el Proyecto CICPC.....	94
Anexo III: Resultados de la encuesta inicial.....	95
Anexo IV: Resultados de la encuesta después de aplicada la propuesta.....	96
Anexo V: Modelo para la recogida de información referente a la calificación de los criterios para la evaluación por expertos. Primera vuelta	97
Anexo VI: Modelo para la recogida de información referente a la calificación de los criterios para la evaluación por expertos. Segunda vuelta	98

ÍNDICE DE FIGURAS

Figura 1. Proceso Unificado	9
Figura 2. Actividades de la Arquitectura de Información ubicadas en RUP.....	29
Figura 3. Diagrama de Procesos del Flujo de Trabajo Modelamiento del Negocio incluyendo la Arquitectura de Información.....	31
Figura 4. Detalles del Proceso Levantamiento de Información.	32
Figura 5. Diagrama de Procesos del Flujo de Trabajo Requerimientos.....	38
Figura 6. Detalles del Subproceso Desarrollo de Pautas de Arquitectura de Información.	39
Figura 7. Diagrama de Procesos del Flujo de Trabajo Análisis y Diseño.	45
Figura 8. Detalles de Proceso Analizar el Comportamiento vinculado a la Arquitectura de Información	46
Figura 9. Diagrama de Proceso del Flujo de Trabajo de Prueba	49
Figura 10 . Detalles del Subproceso Probar y Evaluar.	50

ÍNDICE DE TABLAS

Tabla 1. Planificación de actividades de la Arquitectura de Información en el proyecto.	56
Tabla 2. Valor otorgado por los expertos a los criterios.....	58
Tabla 3. Resultado de las pruebas de Kendall y X^2 de Friedman.	59
Tabla 4. Valor otorgado por los expertos a los criterios en la segunda vuelta.	60
Tabla 5. Resultado de las pruebas de Kendall y X^2 de Friedman en la segunda vuelta.	61
Tabla 6. Cálculo del Peso por la Estadística Descriptiva del SPSS.....	61

INTRODUCCIÓN

La demanda de los productos de software y los servicios de información tecnológica tienen una de las tasas de crecimiento mundiales más alta en la actualidad. Las Tecnologías de la Información y Comunicación (TICs) juegan un papel relevante en la economía mundial. Su capacidad de atracción de inversión y generación de valor, se ve reflejada en la detonación de nuevas capacidades productivas, así como en la generación de empleos bien remunerados en diferentes economías del mundo. La industria del software y servicios informáticos (SSI) ha sido una de las más dinámicas a escala global en los últimos años. Esto no es sorprendente, considerando que el software juega un papel clave dentro del conjunto de avances tecnológicos.

Los países desarrollados son, sin lugar a dudas, los mayores productores y consumidores de software en el mundo, destacándose en un 50% los EEUU. Sin embargo, existen algunos países en desarrollo (PED) que han alcanzado una inclusión significativa en los mercados internacionales, siendo los casos más notorios el de la India, Irlanda e Israel, países de “ingreso tardío” que han alcanzado un gran éxito en esta industria.

Por la Industria del Software aportar significativos avances en diferentes facetas, como en la Economía, la Socialización del Conocimiento, en las Herramientas para manejar la información, por citar algunos ejemplos, merece que los productos que ella genera salgan con la calidad requerida.

Hoy en día, las compañías de todo el mundo industrializado, reconocen que la calidad del producto se traduce en ahorro de costos y en una mejora general. La Industria de Desarrollo de Software no es la excepción, por lo que en los últimos años se han realizado intensos trabajos para aplicar los conceptos de Calidad en el ámbito del software. La Calidad del Software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. Es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, portabilidad, usabilidad, seguridad e integridad. Para lograr que un software cumpla con dichas propiedades se hace necesaria la utilización de metodologías o procedimientos estándares para todas las fases de desarrollo del mismo.

La información y el acceso a ella, como primera necesidad en un mundo conectado a través de las TICs y los distintos niveles de información y de comprensión de la misma por el hombre, son problemas que surgen cuando se habla de calidad en el mundo de la Industria de Software. Este es, sin lugar a dudas, el nicho de la Arquitectura de Información (AI), disciplina encargada de la fundamentación, análisis, planificación y estudio de la disposición de los datos contenidos en los sistemas de información interactivos. Esta posibilita organizar conjuntos de información, permitiendo

que cualquier persona los entienda y los integre a su propio conocimiento, de manera simple. Su surgimiento está marcado por la necesidad de estandarizar la forma de presentarle la información al usuario para diferentes entornos y que esta información sea a su vez asequible, y fácil en su uso y entendimiento.

En sus inicios, la Arquitectura de Información se empleó solamente en el entorno Web, para desarrollar sitios y portales, sin embargo, ¿por qué no aplicarse a cualquier software? Esta interrogante fue la base para que se empezara a incorporar la Arquitectura de Información en la elaboración de cualquier producto informático.

El Gobierno Cubano se dio a la tarea de desarrollar la Industria del Software, con la finalidad de desarrollar Sistemas para la informatización de la sociedad, y para lograr escalar en el Mercado del Software a nivel mundial, debido a las ventajas económicas que esto ofrece. La Universidad de las Ciencias Informáticas (UCI), surgida al calor de la Batalla de Ideas y precisamente para impulsar el desarrollo de la informática en nuestro país, forma a sus estudiantes bajo la premisa de combinar la docencia con la producción, preparándolos como futuros informáticos. Su producción está enfocada tanto al ámbito nacional como al internacional y tiene como Perfiles de Producción, creación de Portales y Sitios Web, productos Multimedia y en menor proporción Software de Gestión. Por ser la aplicación de la Arquitectura de Información a un software de gestión algo novedoso, no se poseía un documento oficial que plasmara una guía de realización de los procesos vinculados a esta rama del conocimiento; además no existía uniformidad de criterios entre los especialistas encargados de establecer las tareas de la Arquitectura de Información en los proyectos productivos, ya que estos trabajaban de forma aislada guiándose por criterios propios.

En el marco de la colaboración de los países de la Alternativa Bolivariana para las Américas (ALBA), se involucran Cuba y Venezuela en un proyecto de Modernización del Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC) que abarca principalmente la creación de un software que gestione de manera más oportuna, ágil y eficaz sus procesos, para una mejor respuesta a la investigación y lucha contra el delito, en beneficio de la sociedad venezolana. Para dar cumplimiento a esta importante tarea asignada a la UCI fue creado el proyecto CICPC, con estudiantes pertenecientes a la Facultad 8. En los inicios del proyecto, el montaje de prototipo de interfaces de usuario no reflejaba los conocimientos básicos de Arquitectura de Información que era necesario para esta tarea. Existía una diferencia de criterios en cuanto a la forma de organizar y visualizar la información en las interfaces de usuario entre Analistas y Aseguradores de Calidad Interna del proyecto. Además, se desconocía cómo vincular la Arquitectura de Información al proceso de desarrollo del software. Todas

estas razones provocaban que el producto final de dichas interfaces no contara con la calidad necesaria.

A partir de lo antes expuesto, el **problema científico** es la ausencia de una estrategia para gestionar la Arquitectura de Información en el Software de Gestión del CICPC.

Por lo tanto el **objeto de estudio** está centrado en la Arquitectura de Información para Software de Gestión.

El **campo de acción** del presente trabajo está conformado por los procesos donde interviene la Arquitectura de Información en el Software de Gestión para el CICPC.

El presente trabajo surge por la necesidad de dar solución al problema planteado anteriormente, por tal razón se ha trazado como **objetivo general** crear una Guía para la Arquitectura de Información en el Software de Gestión para el CICPC.

Dentro de los **objetivos específicos** planteados en el trabajo se encuentran:

- Establecer los elementos teóricos de la Arquitectura de Información para Sistemas de Gestión en ambiente Web.
- Diseñar las Pautas de Arquitectura de Información para el Proyecto CICPC.
- Analizar el comportamiento de la aplicación de las Pautas de Arquitectura de Información.

Para cumplir con los objetivos específicos, se trazaron las siguientes **tareas**:

- Profundización en el estudio acerca de la Arquitectura de Información y Calidad de Software.
- Estudio y análisis de los requerimientos del Sistema de Investigación e Información Policial (SIIPOL).
- Estudio del plan de captura de requerimientos definido para el proyecto.
- Realización de un estudio de las herramientas para la aplicación de la Arquitectura de Información en prototipos de interfaces de usuario.
- Elaboración de una Biblioteca de Componentes.
- Elaboración de las Pautas de Arquitectura de Información.
- Definición de las actividades y artefactos a desarrollar en el producto y que es responsabilidad del arquitecto de información.
- Aplicación de las Pautas de Arquitectura de Información al proyecto CICPC.
- Evaluación del impacto de la solución en el proceso de Desarrollo.
- Validación, por parte de expertos, de la propuesta de las Pautas de Arquitectura de Información para el CICPC.
- Realizar encuestas a integrantes del CICPC para evaluar la aceptación de la propuesta.

Como **idea a defender** se expone que al diseñar y aplicar las Pautas de Arquitectura de Información al Proyecto CICPC, se logrará un software de alta calidad en cuanto a Organización de Información, Usabilidad y Accesibilidad para todos sus usuarios.

Los métodos empleados para nuestra investigación son los históricos-lógicos: analítico-sintético para la recopilación de información sobre el tema que se trata y sus antecedentes, también se utiliza el método empírico, y más específicamente la observación. Para validar la propuesta que se presenta en este trabajo también se emplea la entrevista a expertos con una población de 7 expertos en Arquitectura de Información. Para la aceptación del resultado final se realizarán encuestas a los integrantes de CICPC con una muestra de 20 personas.

Las técnicas empleadas son las de Análisis documental, Encuesta y Entrevistas.

Estructura Capitular

Capítulo 1. Marco teórico–conceptual sobre Arquitectura de Información, el arquitecto de información como especialista que desempeña la arquitectura, y sobre principios de la Metodología de Proceso Unificado de Software.

Capítulo 2. Vinculación de la Arquitectura de Información como proceso que se inserta dentro de la Metodología de Proceso Unificado de Software, y desarrollo de las fases de la Arquitectura de Información durante todo el periodo de creación de software, con sus respectivos artefactos generados de estas etapas.

Capítulo 3. Análisis de resultados mediante aplicación de encuestas a usuarios internos del Proyecto CICPC y validación de la propuesta para medir su calidad y efectividad mediante entrevista y encuesta a expertos en Arquitectura de Información.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

La Sociedad de la Información y las Tecnologías de la Información y las Comunicaciones se han materializado y han alcanzado su desarrollo a partir de la especificación del protocolo de transferencia de hipertexto (HTTP), el cual revolucionó la búsqueda y recuperación de información en Bases de Datos en los años 80 y principios de los 90 hasta la actualidad. A partir de este momento surge la WWW "...elemento que ha popularizado finalmente a Internet, por la variedad y versatilidad de las páginas, aunque sólo se trate de uno de sus servicios" (Gómez Reyes, 2002)

1.2 Arquitectura de Información

1.2.1 Antecedentes

El término Arquitectura de Información fue usado por primera vez por el arquitecto de profesión Richard Wurman en la conferencia del Instituto Americano de Arquitectos que se celebró en 1976. En el año 1996, publica su libro "Information Architects" en el que aportaba definiciones de su concepto de Arquitectura de Información.

La Arquitectura de Información (AI) se ocupa de transformar la información almacenada en información útil. Muchas veces las dificultades radican en que no siempre se puede recuperar toda la información existente sobre un tema determinado porque no se realiza un adecuado proceso de clasificación e indexación y en ocasiones también se dificulta la recuperación de la información contenida en una página Web por una inadecuada organización de los contenidos.

Se puede aseverar que los problemas que han de estudiarse en un sistema de comunicación tienen que ver con la cantidad de información y la capacidad del canal de comunicación para la difusión y la recuperación de la misma. (Gómez Reyes, 2002)

1.2.2 Algunas Definiciones

Existen varias definiciones sobre Arquitectura de Información como la de Rosenfeld y Peter Morville en su libro "Information Architecture for the World Wide Web", o como mejor se conoce "Libro del Oso Polar", los cuales plantean que la Arquitectura de Información se refiere al diseño, organización, etiquetado, navegación y sistemas de búsqueda que ayudan a los usuarios a encontrar y gestionar la información de manera efectiva. Plantean además que la Arquitectura de Información determina la

funcionalidad que el sitio va a tener, especifica cómo los usuarios van a encontrar la información al definir su organización, navegación, etiquetado y sistemas de búsqueda y representa cómo el sitio se va a acomodar al cambio y crecimiento en el tiempo, así como que organiza, rotula, sistematiza, titula y nombra los botones que realizan acciones en un sistema.

Otros autores como Steve Toub, de Arhus Associates, refiere que la Arquitectura de Información es el arte y la ciencia de estructurar y organizar el entorno informativo para ayudar a los usuarios eficientemente a satisfacer sus necesidades informativas.

James Garret, Jesse en “Elements of user Experience” la define como el Diseño estructural del espacio informacional para facilitar el acceso intuitivo a los contenidos”. (James Garret, 2002)

La Consultoría de Arquitectura de Información, en su Portal de Arquitectura de Información¹ resume los conceptos anteriores, pues establece que la Arquitectura de Información es la disciplina que organiza la información, permitiendo que cualquier persona los entienda y los integre a su propio conocimiento, de manera simple. Se utiliza fundamentalmente en espacios virtuales donde se requiere que el propio usuario obtenga la Información. Es el estudio de la organización de la información con el objetivo de permitir al usuario encontrar su vía de navegación hacia el conocimiento y la comprensión de la información.

Para la autora del presente trabajo la Arquitectura de Información es la disciplina que se encarga del estudio, análisis, fundamentación y disposición de los contenidos en un sistema de información.

La Arquitectura de Información como disciplina no busca definir una metodología de diseño universal sino *articular un conjunto de técnicas para ayudar al desarrollo y producción de los productos electrónicos*, no trata sobre el establecimiento de un conjunto de pasos o guías predefinidas, sino del *diseño inteligente que subyace detrás de una interfaz o espacio de información. Trata de maximizar las asociaciones que se producen entre interactividad, navegación y contenido con el objetivo de alcanzar una integración sistémica con el cerebro del usuario logrando un fenómeno de persuasión, conocimiento o información que se traspasa de un sistema de información a otro.*

Para lograr una mejor asimilación de los contenidos por parte de los usuarios de forma eficiente y efectiva, la Arquitectura de Información se encarga en los estudios de audiencia y la definición del público objetivo de:

- El diseño de la interacción.
- El diseño de la navegación.
- La planificación, desarrollo y gestión del contenido.

¹ Dirección electrónica: www.aichile.org/glosario

- La facilidad de búsqueda.
- La usabilidad y la accesibilidad.
- El rediseño de la interfaz.

Teniendo en cuenta los conceptos dados por los autores mencionados, se llega a la conclusión de la importancia misma de la arquitectura, aspecto que se detalla en el epígrafe siguiente.

1.2.3 Importancia de la Arquitectura de Información

Está sustentada su importancia porque reduce o evita durante la elaboración de un producto de software

- Los costos de mantenimientos (gestión de contenidos y rediseños), es necesario que en el primer momento del Ciclo de Vida del Software se gestione de manera eficiente los contenidos que contendrá el mismo, esto evitaría tener que rediseñar tantas veces lo mismo; así como los costos de entrenamiento (retención del personal), es muy importante que el equipo de trabajo esté capacitado para la labor y que a su vez se sienta motivado por lo que está haciendo, cuestión que sería positiva para el software.
- Los costos de encontrar (tiempo y frustración), no es saludable elaborar un producto que se demore mucho tiempo, esto provocaría pérdida de dinero y frustración por parte del cliente y de los responsables de su realización; como la arquitectura de información estudia al cliente, conoce sus necesidades y sabe qué información y de qué manera es óptimo mostrarle la información, se emplea menos tiempo en el desarrollo del producto; esto no sucedería así, si no se tienen las nociones básicas de qué es y cómo quiere el usuario la información que solicita.
- Los costos de no encontrar (malas decisiones y otros canales), el objetivo de reducir o eliminar este costo es evitar que una mala decisión arruine el desarrollo del software.
- Los costos de construcción (personal, tecnología y errores), es muy importante que el personal conozca la tecnología con la cual trabaja, esto ayudaría a minimizar la cantidad de errores que se pudieran presentar y garantizaría la calidad del producto.

De manera conveniente si se reducen esos costos se garantiza que la información esté organizada y contextualizada, que permita aportar conocimientos, demostrando el *Valor cognitivo*; donde exista la relación Productos-Proyectos-Personas, que es el *Valor educativo*; y por último se garantice el *Valor de Identidad* al lograr que el producto final tenga la identidad del cliente, reputación y confianza.

1.2.4 Arquitectura de Información como proceso gerencial

Es conveniente decir que la Arquitectura de Información divide sus tareas de la misma manera que las cuatro etapas de la gerencia. Estas etapas son: Planificación, Organización, Ejecución y Control. (Ronda León, 2005)

Planificación: Es la etapa donde se concibe el proyecto del producto. Es donde se realiza la definición de los objetivos (misión, objetivos...), el estudio de mercado y usuarios, la investigación temática y la selección de la información a utilizar y la definición de los procesos de la producción.

Organización: Es donde se diseña el producto. Se toma información suministrada en los pasos anteriores. Se establecen los procesos, además se organizan y representan los contenidos.

Ejecución: Es la etapa donde se realiza la programación y almacenamiento del producto (grabación, publicación, etc.)

Control: Es la etapa final donde se prueba el producto concluido. En esta fase se realizan las pruebas, test, controles de calidad.

Estas etapas se asemejan a algunos procesos y actividades del Ciclo de Vida de un Software planteado por RUP, por consiguiente la Arquitectura de Información también está presente en cada una de las fases, ocurriendo en determinadas disciplinas, y formando parte del desarrollo iterativo e incremental (desarrollo que se explicará en el transcurso de la investigación).

1.2.5 Arquitectura de Información y su relación con RUP

El Proceso Unificado de Software (RUP) es un proceso de software genérico que puede ser utilizado para una gran cantidad de Sistemas de Software, para diferentes áreas de aplicación, diversos tipos de organizaciones, niveles de competencia y tamaños de proyectos. (JACOBSON, y otros, 2000)

Esta metodología provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Al igual que la Arquitectura de Información, su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible.

El Proceso Unificado tiene dos dimensiones (Figura 1) (UABC, 2007):

- Un eje horizontal que representa el tiempo y muestra los aspectos del Ciclo de Vida del proceso a lo largo de su desenvolvimiento.
- Un eje vertical que representa las disciplinas, las cuales agrupan actividades de una manera lógica de acuerdo con su naturaleza.

La primera dimensión representa el aspecto dinámico del proceso conforme se va desarrollando, se expresa en términos de fases, iteraciones e hitos.

La segunda dimensión representa el aspecto estático del proceso: cómo es descrito en términos de componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles.

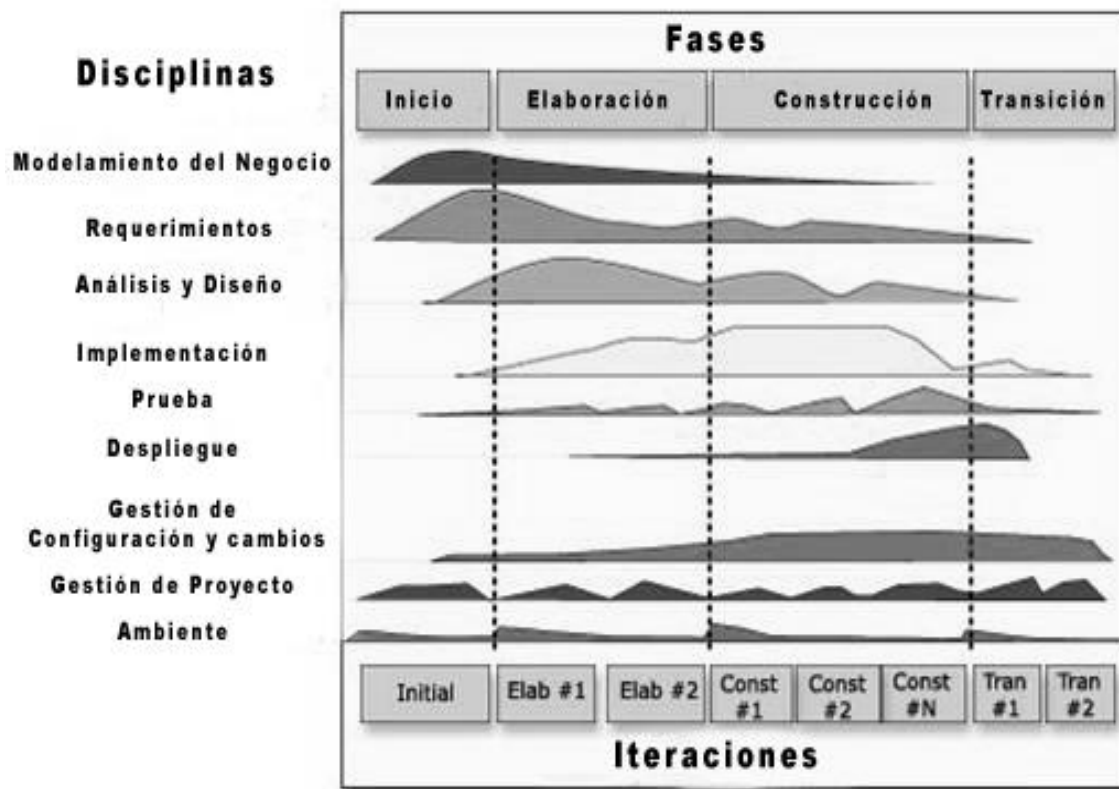


Figura 1. Proceso Unificado

En su modelación se define como sus principales elementos:

- *Trabajadores* (¿Quién?): Definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- *Actividades* (¿Cómo?): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.
- *Artefactos* (¿Qué?): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- *Flujo de actividades* (¿Cuándo?): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

Los aspectos distintivos del Proceso Unificado están capturados en tres conceptos clave:

➤ **Dirigido por Casos de Uso**

Un sistema de software se crea para servir a sus usuarios. Por lo tanto, para construir un sistema exitoso se debe conocer qué es lo que quieren y necesitan los usuarios potenciales.

El término usuario se refiere no solamente a los humanos, sino a otros sistemas. En este contexto, el término usuario representa algo o alguien que interactúa con el sistema a desarrollar.

Un *Caso de Uso* es una pieza en la funcionalidad del sistema que le da al usuario un resultado de valor. Los Casos de Uso capturan los requerimientos funcionales. Todos los casos de uso juntos constituyen el *Modelo de Casos de Uso*, el cual describe la funcionalidad completa del sistema. Este modelo reemplaza la tradicional especificación funcional del sistema. Una especificación funcional tradicional se concentra en responder la pregunta: ¿Qué se supone que el sistema debe hacer? La estrategia de casos de uso puede ser definida agregando tres palabras al final de la pregunta: ¿...por cada usuario? Estas tres palabras tienen una implicación importante, se fuerza a pensar en términos del valor a los usuarios y no solamente en términos de las funciones que sería bueno que tuviera. Sin embargo, los casos de uso no son solamente una herramienta para especificar los requerimientos del sistema, también dirigen su diseño, implementación y pruebas, esto es, dirigen el proceso de desarrollo.

Aún cuando los casos de uso dirigen el proceso, no son elegidos de manera aislada. Son desarrollados a la par con la arquitectura del sistema, esto es, los casos de uso dirigen la arquitectura del sistema y la arquitectura del sistema influencia la elección de los casos de uso. Por lo tanto, la arquitectura del sistema y los casos de uso maduran conforme avanza el Ciclo de Vida.

➤ **Centrado en la Arquitectura**

El papel del Arquitecto de Sistemas es similar en naturaleza al papel que el arquitecto desempeña en la construcción de edificios. El edificio se mira desde diferentes puntos de vista: estructura, servicios, plomería, electricidad, etc. Esto le permite al constructor ver una radiografía completa antes de empezar a construir. Similarmente, la arquitectura en un sistema de software es descrita como diferentes vistas del sistema que está siendo construido.

El concepto de Arquitectura de Software involucra los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, tal y como las interpretan los usuarios y otros involucrados, y tal y como están reflejadas en los casos de uso. Sin embargo, también está influenciada por muchos otros factores, como la plataforma de software en la que se ejecutará, la disponibilidad de componentes reutilizables, consideraciones de instalación, sistemas legados, requerimientos no funcionales, entre otros. La arquitectura es la vista del diseño completo con las características más importantes hechas más visibles y dejando los detalles de lado. El valor de la arquitectura depende del personal asignado a esta tarea. Sin embargo el proceso, ayuda al arquitecto a enfocarse en las metas correctas, tales como claridad y flexibilidad en los cambios futuros.

¿Cómo se relacionan los Casos de Uso con la Arquitectura? Cada producto tiene función y forma. Uno sólo no es suficiente. Estas dos fuerzas deben estar balanceadas para obtener un producto exitoso. En este caso, función corresponde a los casos de uso y forma a la arquitectura. Existe la necesidad de intercalar entre casos de uso y arquitectura. Por una parte, los casos de uso deben, cuando son realizados, acomodarse a la arquitectura. Por otra parte, la arquitectura debe proveer espacio para la realización de todos los casos de uso, hoy y en el futuro. En la realidad, arquitectura y casos de uso deben evolucionar en paralelo.

➤ **Iterativo e Incremental**

Desarrollar un producto de software comercial es una tarea enorme que puede continuar por varios meses o años. Es práctico dividir el trabajo en pequeños pedazos o mini-proyectos. Cada mini-proyecto es una iteración que finaliza en un incremento. Las iteraciones se refieren a pasos en el Flujo de Trabajo, los incrementos se refieren a crecimiento en el producto. Para ser más efectivo, las iteraciones deben estar controladas, esto es, deben ser seleccionadas y llevadas a cabo de una manera planeada.

Los desarrolladores basan su selección de lo que van a implementar en una iteración en dos factores. Primero, la iteración trata con un grupo de casos de uso que en conjunto extienden la usabilidad del producto. Segundo, la iteración trata con los riesgos más importantes. Las iteraciones sucesivas construyen los artefactos del desarrollo a partir del estado en el que fueron dejados en la iteración anterior.

En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean el diseño usando la arquitectura como guía, implementan el diseño en componentes y verifican que los componentes satisfagan los casos de uso. Si una iteración cumple sus metas

(y usualmente lo hace) el desarrollo continúa con la siguiente iteración. Cuando la iteración no cumple con sus metas, los desarrolladores deben revisar sus decisiones previas y probar un nuevo enfoque.

RUP también establece los siguientes flujos de trabajo:

Modelamiento del Negocio: describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

Requerimientos: define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

Análisis y Diseño: describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

Implementación: define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

Prueba: busca los defectos a lo largo del Ciclo de Vida.

Despliegue: produce liberaciones del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los Usuarios finales.

Administración del Proyecto: involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

Administración de Configuración y Cambios: describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a la utilización/actualización concurrente de elementos, control de versiones, etc.

Ambiente: contiene actividades que describen los procesos y herramientas que soportarán el Equipo de Trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Al analizar la Arquitectura de Información como proceso iterativo, se ubica dentro de RUP fundamentalmente en la *Fase de Inicio*, en el *Flujo de Trabajo Modelamiento del Negocio* mediante las siguientes actividades:

- Identificar el objeto, propósito y fines del software.
- Definir el público objetivo y los estudios de la audiencia.
- Gestionar la configuración enfocado a las políticas de organización de la información dentro del proyecto.

Generándose artefactos como:

- Documento de Necesidades de Usuario.

- Documento de Características del Software.

La Arquitectura de Información se sitúa además en el *Flujo de Trabajo Requerimientos* a través de las acciones:

- Prototipar Pautas de Diseño.
- Prototipar Interfaces de Usuario.

Obteniéndose como artefactos:

- Documento de Arquitectura de Información.

Unido a las actividades anteriormente dichas, la Arquitectura de Información también se encarga de definir las tareas: Diseñar la Navegación, Definir el Etiquetado (Ver Glosario de Términos) o rotulado de los contenidos para acceder a la información y además, de la Planificación, Desarrollo y Gestión de Contenidos.

1.3 El Prototipo de Interfaz

En el proceso de Arquitectura de Información aplicada a la Gestión de Contenidos en las Interfaces de Usuario de un software, es necesaria la Representación de la Información, para tener una idea de cómo quedará el producto final. Los Prototipos se utilizan en este momento para ilustrar cuál será la organización y estructuración visual de los diferentes elementos del futuro software, y con ello validar la propuesta con los Clientes y Usuarios finales.

Ahora bien, ¿Qué son los Prototipos?

Un Prototipo es una representación limitada del diseño de un producto, que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso. (La Calle, 2005)

Maner define que un Prototipo es un modelo (representación, demostración o simulación) fácilmente ampliable y modificable de un sistema planificado, probablemente incluyendo su interfaz y su funcionalidad de entradas y salidas (Maner, 1997). Es un modelo a escala o facsímil de lo real, pero no tan funcional para que equivalga a un producto final: es una implementación parcial, pero concreta de un sistema o una parte del mismo. Principalmente se crean para probar cuestiones de Diseño Gráfico y Visualización sobre todo lo que vaya a mostrar el sistema durante su desarrollo.

Los prototipos constituyen además una herramienta muy útil para hacer participar al usuario en el desarrollo y poder evaluar el producto desde las primeras fases de este.

Existen varios tipos de Prototipos de Interfaces en dependencia de la utilización que se le vaya a dar a los mismos, como se explica en el sub-epígrafe siguiente.

1.3.1 Tipo de Prototipos

Existen fundamentalmente dos tipos de prototipos:

1. *Prototipo Rápido (concept prototipe)*: es un mecanismo para lograr la validación pre-compromiso. Se utiliza para validar requerimientos en una etapa previa al diseño específico. En este sentido, el prototipo puede ser visto como una aceptación interna de que los requerimientos no son totalmente conocidos o entendidos antes del diseño y la implementación. El prototipo rápido puede ser usado como un medio para explorar nuevos requerimientos y así ayudar a "controlar" su constante evolución.
2. *Prototipo Evolutivo*: Desde una perspectiva diferente, todo el Ciclo de Vida de un producto puede ser visto como una serie incremental de detallados prototipos acumulativos. Tradicionalmente, el Ciclo de Vida está dividido en dos fases distintas: Desarrollo y Mantenimiento. La experiencia ha demostrado que esta distinción es arbitraria y va en contra de la realidad, ya que la mayor parte del costo del software ocurre después de que el producto se ha entregado. El punto de vista evolutivo del Ciclo de Vida del software considera a la primera entrega como un prototipo inicial en el campo. Modificaciones y mejoras subsecuentes resultan ser nuevas entregas de prototipos más maduros. Este proceso continúa hasta que se haya desarrollado el producto final. La adopción de esta óptica elimina la distinción arbitraria entre desarrollo y mantenimiento, resultando en un importante cambio de mentalidad que afecta las estrategias para la estimación de costos, enfoques de desarrollo y adquisición de productos.

Sin embargo existen otras clasificaciones de prototipos como por ejemplo:

- Baja Fidelidad: conjunto de dibujos (por ejemplo, una presentación de escenarios) que constituye una maqueta estática, no computarizada de una interfaz de usuario para un sistema en planificación.
- Alta Fidelidad: conjunto de pantallas que proporcionan un modelo dinámico, computarizado y operativo de un sistema en planificación.
- Exploratorio: prototipo no reutilizable usado para clarificar las metas del proyecto, identificar requerimientos, examinar alternativas de diseño o investigar un sistema extenso y complejo.
- Experimental: prototipo utilizado para la validación de especificaciones del sistema.
- Operacional: prototipo iterativo que es progresivamente refinado hasta que se convierte en el sistema final.
- Horizontal: prototipo que modela muchas características de un sistema pero con poco detalle. Dicho detalle alcanzará una profundidad determinada, va a resultar especialmente útil en las etapas tempranas de diseño y tiene como objetivo el test del modo de interacción global, al

contemplar funciones comunes que el usuario va a utilizar frecuentemente.

- Vertical: prototipo que modela pocas características de un sistema pero con mucho detalle. Va a resultar especialmente útil en etapas más avanzadas del diseño y tiene como objetivo el test de detalles del diseño
- Diagonal: prototipo horizontal hasta un cierto nivel, a partir del cual se puede considerar vertical.
- Global: prototipo del sistema completo. Prototipo horizontal expandido que modela una gran cantidad de características y cubre un amplio rango de funcionalidades. Va a resultar muy útil a lo largo de todo el proceso de diseño.
- Local: prototipo de un único componente o característica del sistema de usabilidad crítica. Va a resultar de utilidad en algunas etapas específicas del proceso de diseño.

1.3.2 Niveles de Prototipado

Se puede hacer una clasificación de los principales tipos de prototipos, variando su grado de complejidad, de acuerdo con las características que consideren su operabilidad para realizar simulaciones. Estos niveles son:

- *Prototipos Estáticos*: son aquellos que no permiten la alteración de sus componentes, pero sirven para identificar y resolver problemas de diseño. En esta categoría se incluyen las presentaciones sobre reproductores, papel u otro medio de visualización.
- *Prototipos Dinámicos*: permiten la evaluación de un modelo del sistema sobre una estación de trabajo o una PC. Estos prototipos involucran aspectos de diseño más detallados que los prototipos estáticos, incluyendo la validación del diseño del sistema en términos de requerimientos no funcionales, por ejemplo de rendimiento (performance). El inconveniente es que para hacer prototipos dinámicos se necesita más tiempo y personal para realizar la tarea.
- *Prototipos Robustos*: deben ser relativamente completos en la simulación de las características dinámicas de la interfaz (presentación de mensajes de error, entrada y edición de datos, etc.). Esta categoría puede ser utilizada para validar los objetivos de diseño.

El nivel de sofisticación del prototipo debería incrementarse a lo largo del proceso de Diseño de Interfaz de Usuario. La información recopilada durante las tareas de Análisis del Sistema y la Especificación de los Requisitos del Usuario constituyen los datos claves para el proceso de prototipación que se explica detalladamente a continuación.

1.3.3 Fases en el Proceso de Prototipado

Se definen cuatro pasos en el proceso de prototipado de cualquier interfaz de usuario. Esos cuatro pasos se corresponden con la estructura básica de cualquier propuesta de Ingeniería de Requisitos. El primer paso es determinar las necesidades del sistema, y no es un estado diferente al de Levantamiento de Requisitos en el que por lo general, el Analista y el Arquitecto de Información se pregunta quién y para qué se utilizará el sistema. En esta fase, el modelado de interfaces se mueve desde la definición de requisitos al análisis, punto en el que se decide evolucionar todo o parte de un prototipo esencial a un prototipo de interfaz tradicional. Esto implica convertir los borradores, elaborados a mano, en algo más sustancial, que se traduce en Prototipos de Interfaz sustentadas en herramientas informáticas.

Mientras se identifican y determinan las necesidades de los distintos involucrados en el desarrollo se pueden transformar los Prototipos de Interfaz de Usuarios esenciales, si fueron creados, y empezar a desarrollar *sketches* (bocetos). Una transformación entre prototipo esencial y tradicional no tiene por qué poseer una asociación directa, en muchos casos unos y otros prototipos pueden ser completamente diferentes.

Una vez que se entienden las necesidades de interfaz de los involucrados en el uso y desarrollo del sistema, el siguiente paso es construir un prototipo. Usando una herramienta de prototipado o un lenguaje de alto nivel, se pueden desarrollar pantallas, páginas e informes que los usuarios necesitan. Con la plataforma de interfaz seleccionada, se puede empezar el proceso de convertir aspectos individuales recogidos en los prototipos esenciales a los prototipos tradicionales. En este punto se pasa por la realización de *sketches* o directamente a la confección de interfaces. La realización de *sketches* involucra más a los usuarios.

No se necesita crear un prototipo para el sistema completo. Es muy habitual limitarse a prototipar una pequeña porción de la interfaz de usuario, quizás una simple pantalla o página HTML antes de proceder con la implementación. Debe tenerse también en cuenta que los desarrolladores trabajan siguiendo una filosofía evolutiva. Amén de que en algunos casos, y puntualmente, se tenga que prototipar una porción importante del sistema, se tratará de ejercicios de visionado o para ayudar a definir el alcance del proyecto y justificar los recursos económicos que se solicitan.

Una vez que el prototipo ha sido construido, debe ser evaluado o validado por los involucrados en el proceso de desarrollo con el fin de verificar que se contemplan todas las necesidades. Algunas veces esto es tan fácil como pedirle a alguien que gaste algo de tiempo mirando lo que otro ha construido y otras veces es tan complicado como fijar una reunión para discutir y presentar el desarrollo al grupo de

personas involucrada en el mismo. Cuando se evalúa un prototipo a nivel personal es útil hacerse preguntas como las siguientes:

- ¿Qué destacarías del prototipo elaborado?
- ¿Qué identificas como malo o controvertido?
- ¿Qué decisiones de diseño has considerado y qué se ha perdido en el camino?

Después de evaluar el prototipo, se puede necesitar partir, modificar o incluso enriquecer. También se puede dar por finalizado el proceso de evaluación de prototipo cuando no surjan nuevas ideas que considerar. En cualquier otro caso habrá que volver a capturar requisitos, volviendo a la primera etapa. En el libro “The Object Primer” (El Primer Objeto) de ScottW Ambler (Ambler, 2004) sugiere los siguientes trucos y técnicas que le han sido útiles en su trabajo realizando prototipos:

- Trabaja con usuarios reales.
- Haz que los involucrados en el proceso de desarrollo trabajen con los prototipos.
- Entiende el proceso real de negocio al que el sistema dará soporte.
- Prototipa utilizando características que estés capacitado para desarrollar.
- Consigue que alguien experto en interfaces te ayude a diseñar.
- Explica qué es un prototipo y su utilidad.
- Deja las decisiones de implementación para tan tarde como sea posible.

1.3.4 Objetivos del Prototipo

Siempre se debe establecer cuál es su objetivo pues el prototipo puede ser útil en diferentes fases del proyecto.

En la Fase de Requisitos los prototipos son usados para dar una visión general de cómo va a funcionar el sistema, de acuerdo con lo especificado en los Casos de Uso del sistema. Además de permitir que los usuarios aprueben dicha propuesta sin tener que implementar el software completo para enseñársela, esta retroalimentación facilita que se puedan cambiar o mantener funcionalidades de la aplicación de forma temprana.

En la Fase de Análisis de un proyecto se utiliza para obtener los requerimientos del usuario, su principal propósito es obtener y validar los requerimientos esenciales, manteniendo abiertas, las opciones de implementación. Esto implica que se deben tomar los comentarios de los usuarios, pero se debe regresar a sus objetivos para no perder la atención.

Por otra parte, en la Fase de Diseño se usa para ayudar a evaluar muchos aspectos de la implementación seleccionada, o sea verificar la factibilidad del diseño del sistema. Su propósito, basándose en los requerimientos previamente obtenidos, es mostrar las ventanas, su navegación,

interacción, controles y botones al usuario y obtener una retroalimentación que permita mejorar el Diseño de Interfaz.

1.3.5 Técnicas para la Aplicación de Prototipos

Si se desarrolla una Interfaz de Usuario confusa para un sistema, entonces no importará lo bueno o malo que sea el resto del sistema: los usuarios terminarán odiándolo, pues son precisamente ellos los que interactúan directamente con la Interfaz. Los desarrolladores consecuentes entienden al menos, y luego aplican, los fundamentos del desarrollo de Interfaces de Usuario. Esto implica la aplicación de algunas de las técnicas para ayudar a explorar la Interfaz junto con todos los involucrados en el proceso de Desarrollo, así analizarán sus necesidades y entonces diseñarán interfaces que sean idóneas con ellas.

Existen cuatro técnicas fundamentales para elaborar Prototipos de Interfaces de Usuario:

- **Prototipo de Interfaz de Usuario Esencial:** es un prototipo independiente de la tecnología, creado utilizando papel y puede ser usado para identificar requisitos de interfaz de usuario.

Un prototipo de interfaz esencial es el punto inicial de los prototipos de interfaz para desarrollar un sistema. Son modelos de baja-fidelidad de la interfaz de usuario que ofrecerá un sistema (Constantine, 1999). En estos prototipos se recogen las ideas que hay detrás de la interfaz de usuario, pero no los detalles exactos o concretos. Estos prototipos de interfaz de usuario sirven para recopilar e identificar los requisitos de la misma, independiente de la tecnología. En este sentido, son a los requisitos de interfaz lo que los casos de uso esenciales son a los requisitos de comportamiento.

- **Pantalla/informe sketch:** es un borrador que muestra el diseño de los elementos principales de la interfaz, es decir, se utilizan pantallas, páginas HTML, o informes.
- **Prototipo de Interfaz de Usuario Tradicional:** una interfaz de usuario que “funciona” y que presenta el sistema o parte del sistema a los usuarios; con frecuencia se crea para diseñar una interfaz de usuario en detalle.

Existen dos diferencias básicas entre el prototipado esencial y el tradicional. La primera es que el modelado esencial se centra en los usuarios y en el uso del sistema y no en las características del sistema. La segunda es que las herramientas de prototipado en el prototipado esencial son más simples, incluyéndose pizarras, papel y notas adhesivas. El uso de herramientas de prototipado electrónicas termina introduciendo condicionantes y prototipos que no son independientes de la tecnología como sucede con el prototipado esencial.

Cuando un equipo de trabajo crea un Prototipo Esencial de Interfaces de Usuario, éste itera a través de las siguientes tareas:

1. El equipo explora el uso del sistema y lo hace a través de diferentes medios. Primero, trabaja utilizando una pizarra sobre la cual discutir las ideas, parten de algún boceto o borrador inicial y se favorecen de la naturaleza dinámica que ofrece la pizarra para hacer entender y presentar la porción del sistema que se esté discutiendo en cada momento.
 2. Modelado utilizando elementos de interfaz de alto nivel. Los elementos de interfaz de alto nivel, tales como pantallas o informes potenciales, pueden ser modelados utilizando papel. Se emplea el atributo potencial al referirse a las pantallas o informes para recalcar el hecho de que finalmente se trata de una pantalla o un informe real que dependerá de una decisión de diseño. A cada página de papel se le dará un nombre y contendrá los elementos de interfaz de usuario de menor rango. El uso del papel tiene varias ventajas: pueden colgarse en una pared, son buenos para trabajar en grupo y que todos sean partícipes de su contenido e interactúen con ellos, son suficientemente grandes como para poner notas adhesivas en ellos, se puede escribir y se pueden guardar de una sesión de trabajo a la siguiente.
 3. Modelar los elementos de interfaz de bajo nivel. Los elementos de interfaz de menor nivel, tal como campos de entrada y listas, se modelan utilizando notas adhesivas. En “Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design” (Software para el uso: Una Guía Práctica para los Modelos y Métodos de Uso-Diseño Centrado) (Constantine, 1999), el autor recomienda utilizar diferentes colores para los diferentes tipos de componentes, por ejemplo, colores brillantes/vivos (amarillo o rojo) para los elementos de interfaz activos como son los campos de entrada y colores más suaves o apagados (blanco o canela) para los elementos pasivos o de menor nivel.
 4. Explorar la Usabilidad de la interfaz. Los sistemas altamente usables son fáciles de aprender, de usar, de recordar, se comenten pocos errores y hay una subjetiva sensación de satisfacción. (Nielsen, 1993)
- **Diagrama de Flujo de Interfaz de Usuario:** un diagrama que muestra los elementos principales de la interfaz y cómo los usuarios pasan de unos a otros; se usa para explorar la usabilidad (nivel alto) del sistema y para documentar y validar la interfaz.

El prototipado de interfaces de usuario es una técnica de análisis iterativa en la que los usuarios están involucrados de forma activa en el desarrollo de bocetos de interfaz para un sistema. Los prototipos de interfaces de usuario tienen varios propósitos:

- Como artefacto de análisis, permiten explorar el espacio del problema con el resto de involucrados en el desarrollo.
- Como artefacto de diseño, permiten explorar el espacio de la solución que ofrece el sistema.
- Como vehículo de comunicación, facilitan el intercambio de opiniones y propuestas entre los involucrados en el desarrollo.
- Como cimiento o base potencial permite sentar las bases sobre las que debe continuar el desarrollo del sistema.

Los prototipos de interfaz son un medio excelente para explorar la interfaz, pero desafortunadamente no es fácil hacerse una idea de “todo el cuadro” y muchas veces se quedan relegados a cuestiones de diseño. Los Diagramas de Flujo de Interfaz de Usuario también denominados Storyboards, Diagramas de Navegación entre Ventanas (Page-Jones, 2000) o Mapas de Navegación Contextual (Constantine, 1999), permiten modelar y reflejar las relaciones entre los elementos de interfaz de usuario.

Los diagramas de flujo de interfaz de usuario son útiles para dos propósitos. Primero, se usan para modelar las interacciones que los usuarios tienen con su software, como se define en los casos de uso. Segundo, permiten tomar conciencia del comportamiento de la interfaz de usuario a alto nivel. Esta visión es la vista conjunta de todos los comportamientos reflejados en los distintos casos de uso identificados y es lo que se denomina visión arquitectónica de la interfaz de usuario. (Constantine, 1999) Debido a que los diagramas de flujo de interfaz de usuario ofrecen una visión de alto nivel de la interfaz de usuario del sistema, con ellos se logrará entender cómo se espera que el sistema funcione. Además, el flujo de interfaz de usuario también sirve para determinar si la interfaz de usuario será usable o si el sistema será complejo de aprender y entender.

1.3.6 Estrategias de Diseño de Prototipos de Interfaces de Usuario

Las estrategias para el desarrollo de prototipos son:

- Prototipos para pantallas. El elemento clave es el intercambio de información con el usuario.
- Prototipos para procedimientos de procesamiento. Incluye solo procesos sin considerar errores.
- Prototipos para funciones básicas. Sólo se desarrolla el núcleo de la aplicación, es decir, los procesos básicos. (Donadello, 2004)

Cuando se desarrollan interfaces de usuario se debe tener presente y considerar técnicas y principios básicos de diseño de interfaces de usuario. La experiencia demuestra que al utilizar dichas recomendaciones los resultados son mucho más factibles. Otra recomendación es que dentro del proyecto de desarrollo del sistema haya alguien que ejerza el papel de experto en diseños de interfaz

de usuario, que podría hacerlo la persona que se encarga de la Arquitectura de Información, por explicaciones que se dan en el epígrafe Rol del Arquitecto de Información. De otra forma esa labor deben realizarla desarrolladores habituales que no siempre tienen conocimientos suficientes sobre el tema.

Constantine y Lockwood (Constantine, 1999) describen una colección de principios básicos para mejorar la calidad del diseño de las interfaces de usuario. Estos principios son:

- **El Principio de Estructura.** Los diseños deberían organizarse adecuadamente, modelos claros, consistentes que sean aparentes y fáciles de reconocer por los usuarios, donde se ubiquen las cosas relacionadas juntas, y separadas aquellas otras que no tengan nada que ver.
- **El Principio de Simplicidad.** El diseño debería ser simple, las tareas habituales deben ser fáciles de hacer, presentarse de forma clara y simple, utilizando el lenguaje del usuario, y proporcionando una asistencia significativa.
- **El Principio de Visibilidad.** Los diseños realizados deberían facilitar todas las opciones y material necesario para que una tarea pueda ser realizada sin distracciones o información redundante o innecesaria. Los buenos diseños no saturan o confunden al usuario con muchas alternativas o información superflua.
- **El Principio de Retroalimentación.** El diseño elaborado debería informar al usuario de las acciones, cambios de estado o condición, de los errores que surjan o de las excepciones relevantes y de interés para el usuario a través de mensajes elaborados utilizando lenguaje claro, conciso, correcto y familiar para los usuarios.
- **El Principio de Tolerancia.** Los diseños deberían ser flexibles y tolerantes al error, reduciendo los errores y permitiendo operaciones de hacer-y-deshacer.
- **El Principio de Reutilización.** Los diseños deberían reutilizar componentes internos y externos y sus comportamientos. La consistencia debe primar en los diseños, esto facilita aprender y recordar los sistemas.

1.3.7 Papel del Prototipo

Son útiles para comunicar, discutir y definir las ideas entre los diseñadores y las partes responsables. Los prototipos responden a preguntas y apoyan el trabajo de los diseñadores probando ideas, clarificando requisitos o definiendo alternativas. (La Calle, 2005) Es muy útil hacer uso del prototipo en la Fase de Análisis para prevenir algunos problemas que pudieran presentarse después en la Fase de Desarrollo. Con el prototipo pueden aparecer ciertas interrogantes que respondidas a tiempo evitarían un atraso en el proceso de desarrollo del software. El prototipo ayuda a comprender el funcionamiento

del sistema, a explorar sus funcionalidades y a que el usuario tenga una idea general de lo que va a hacer el producto para que de esta forma se pregunte si es eso realmente lo que quiere y no ocurra que al final el cliente quede insatisfecho y la aplicación no sea lo que realmente necesita. Brinda aportes de comunicación entre desarrolladores y clientes, logra un incremento de la participación constructiva del usuario, contribuye con el Aseguramiento de la Calidad, se reúne la mayor cantidad de requerimientos válidos, se hace una mejor Gestión de las peticiones de Cambios. (Maner, 1997).

1.3.8 Usabilidad

Un aspecto importante que debe considerarse cuando se desarrolla un prototipo de interfaz de usuario esencial o tradicional para un sistema, es la Usabilidad. Un sistema informático es usable cuando puede utilizarse para realizar una determinada tarea de forma eficiente. Centrándose en las aplicaciones para Internet, se podría decir que una web es usable cuando:

- Aporta información relevante y frecuentemente actualizada.
- Resulta fácil de aprender y recordar.
- Permite realizar las tareas para las que fue diseñado de forma rápida y sencilla.
- Genera pocos errores.
- Proporciona una experiencia subjetivamente agradable.

La importancia de la usabilidad radica en: primero, al centrarse inicialmente en el uso y en la usabilidad en lugar de en las características o la funcionalidad del usuario, y en sus usuarios y en sus necesidades más que en las interfaces, los sistemas se pueden convertir en entes menos complejos, simples y más baratos. Segundo, los mejores sistemas son aquellos placenteros, satisfactorios y la gente se siente bien al utilizarlos, en definitiva, son usables. Tercero, los sistemas que son difíciles de utilizar suelen ser difíciles de aprender y más proclives al cambio y a la necesidad de mantenimiento. Cuarto, conforme los usuarios se acostumbran a utilizar productos software se hacen más entendidos y exigentes y menos tolerantes a productos diseñados y elaborados de forma pobre.

1.4 Aplicaciones y Herramientas para Diseñar Prototipos

El prototipo actúa en varias fases del Proceso de Desarrollo del Software y en todas es importante por los obstáculos que soluciona en cada una de ellas. Es importante resaltar la necesidad de empezar a implementarlo aunque sea con un nivel menos complejo, desde etapas tempranas, es decir, desde la Captura de Requisitos, para comenzar a explorar las posibles dificultades que pudieran presentarse y llegar a la Fase de Desarrollo con ideas lo más claras posibles respecto al sistema a implementar.

1.4.1 Normas que Respaldan el uso de Prototipos

La norma IEEE 830, en su edición de 1998 de la Guía para el Desarrollo de Especificaciones de Requerimientos de Sistemas define al prototipo como un modelo experimental del sistema o parte del sistema, ya sea funcional o no funcional, usado para obtener retroalimentación de los usuarios, para ser capaces de mejorar y especificar una interfaz de usuario compleja, para realizar estudios de factibilidad, o para identificar requerimientos.

El estándar ISO 13407 "Human centred design for interactive systems" (Diseño Centrado en el Humano para Sistemas Interactivos), provee de un marco teórico para el desarrollo de actividades centradas en el usuario. Desarrolla prototipos y éstos son probados, cambiados o sustituidos con la retroalimentación de los usuarios reales.

1.4.2 Herramientas para realizar Prototipos

DENIM

Es un sistema que ayuda a los diseñadores de sitios web durante las primeras fases de diseño a través de bocetos a diferentes niveles de refinamiento: el mapa del sitio, *storyboard* y páginas individuales. También unifica los niveles a través de vistas. En esta herramienta los dibujos están pensados para ser realizados sobre la pantalla usando un lápiz electrónico. Combina los beneficios de un sketch, como esos que se crean mediante una hoja de papel, con una herramienta prototipo de código abierto, resultando una funcional, sencilla e interactiva interfaz gráfica.

SILK

Permite dibujar los bocetos tanto con pantalla gráfica como con el mouse. El usuario puede usar cuatro figuras predefinidas: rectángulo, línea libre (para el texto), línea recta y elipse. Combinando estas figuras se forman los prototipos de interfaz, que serán transformados mediante esta herramienta en ventanas reales en Visual Basic 5.0 o Common Lisp. El diseñador puede elegir el estilo de los componentes de la interfaz una vez la herramienta reconozca a cada uno de esos componentes. Para ilustrar la navegación entre las interfaces se utilizan *storyboards*.

Microsoft Office Visio 2003

Es una herramienta visual de modelado fácilmente personalizable. La versión Visio 2003 aporta un diseño de Interfaces Gráficas de Usuario (GUI) completo y fácil de emplear que facilita la creación de formularios, la exportación de páginas Web y la creación de biblioteca de componentes personalizados. (Visio, 2007).

Entre las principales ventajas que esta herramienta brinda están:

- Documentar y visualizar los sistemas y los procesos.
- Aumentar la productividad integrando diagramas con información de varias fuentes.
- Reducir la entrada manual de datos habilitando Office Visio para mantener los diagramas actualizados.
- Analizar información y formarse una idea de los datos mostrándolos en diagramas.
- Seguimiento más eficaz de la información del proyecto.
- Comunicar información compleja con nuevas plantillas y formas.
- Causar impacto en la audiencia con diagramas de aspecto profesional.
- Crear de bibliotecas personales de componentes.
- Simular funcionalidades del futuro sistema a través de la navegación entre pantallas, utilizando hipervínculos.

Existe la versión Visio 2007 pero como no se ha aprobado aun su uso por la gama de productos que se producen en Office y no se tiene certeza de su funcionamiento integrado, además, la dirección de la Infraestructura Productiva (IP) reguló que los documentos de los proyectos de la UCI deben estar en la versión de Office 2003. Por esta razón el proyecto CICPC pautó que se trabajaría con la facilidad de extensión aportada por Visio 2003.

1.5 Calidad de Software

Pressman (Pressman, 2002) define la Calidad de Software como la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente.

Por otra parte la Organización Internacional de Normalización (ISO en sus siglas en inglés), la define como el conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas (ISO 8402 UNE 66-001-92, 1992).

A modo de resumen la Calidad del Software es el grado en que el software satisface una serie de requerimientos funcionales preestablecidos, los estándares tecnológicos para el desarrollo, especificados con anterioridad y las características inherentes a todo producto de software desarrollado de manera profesional.

1.5.1 Listas de Chequeo como instrumento de Validación

Se entiende por Lista de Chequeo (Cheks-List) a un listado de indicadores, en forma de cuestionario o enumerados, que sirve para verificar el grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado.

La ckeck-list enumera una serie de ítems (muchos o pocos, dependerá de la exhaustividad que se pretenda) que deberían verificarse uno a uno para asegurarse de lograr el producto final con un nivel de calidad previamente aceptado.

1.6 Técnicas de Recopilación de Información

En el Congreso Internacional de Información INFO' 2006, Rodrigo Ronda (León, 2006) plantea que existen varias Técnicas para Recopilar Información que satisfaga al cliente como la:

- Técnica de Interacción con el Usuario. Con el empleo de esta técnica se obtiene información de los usuarios y el producto final. Esta información es la base para lograr un Diseño Centrado en el Usuario, además recopila información para las demás etapas de la producción. Esta técnica utiliza las reuniones y entrevistas con el cliente, así como la aplicación de encuestas.
- Técnica de Interacción con el Contexto. Se encarga del Análisis de Homólogos y del Análisis de la Competencia. El objetivo principal de esta técnica es conocer las potencialidades que tienen los productos similares o de la competencia para poderlos mejorar y superar. También analiza las dificultades que presentan los productos revisados con el fin de no repetirlos.
- Técnicas Matemáticas (coocurrencia). Esta técnica aplica análisis de coocurrencia para poder cuantificar resultados y hacer más precisa la toma de decisiones. Con el uso de esta técnica se logra definir los grupos y crear secuencias lo más cercano posible a los modelos mentales de los usuarios.
- Técnica de Representación de la Información. Con el empleo de esta técnica se pueden concretar las propuestas de diseño que establecen los productores de manera abstracta. Crea modelos y prototipos de lo que debe ser el producto final y permite una retroalimentación de los criterios y necesidades de los usuarios en cuanto a las soluciones de diseño del producto. Esta técnica emplea la Diagramación, la Representación de etiquetas, y el Prototipado (maqueta).

En el proyecto CICPC se utilizaron las Técnicas de Interacción con el Usuario, de Interacción con el Contexto y la de Coocurrencia, para lograr que el producto final tenga mayor calidad, cumpliendo con los principios de Usabilidad, Accesibilidad y satisfacción del cliente.

1.7 Rol del Arquitecto de Información

Como todos los procesos que se llevan a cabo en la Sociedad de la Información, la Arquitectura de Información depende del factor humano, por lo que es imprescindible delimitar el Rol del Arquitecto de Información como la persona idónea para ejecutar las tareas propias de la Arquitectura de Información. En un Proceso de Desarrollo, la persona encargada de llevar a cabo la Arquitectura de Información es el Arquitecto de Información, quién además verifica el proceso de diseño del software y trabaja estrechamente con los diseñadores gráficos. Está integrado en un equipo y sus tareas abarcan desde la fundamentación del proyecto hasta el rediseño, verificación y prueba del producto durante todas las fases de desarrollo hasta la obtención del resultado final.

Wurman en su libro su libro “Information Architects” (Arquitectos de Información) (Wurman, 1996) lo define como una persona que organiza los patrones inherentes en los datos, que crea el mapa o la estructura de información y que permite a otros encontrar su camino personal al conocimiento.

El Arquitecto de Información de un Equipo de Desarrollo debe tener habilidades como: capacidad para organizar la información, ponerse en el lugar del usuario, tener una visión global del software, entender la misión del software, así como conocer en profundidad el contenido y equilibrar las necesidades del productor y las de los usuarios. Debe tener además conocimientos de usabilidad, así como trabajar con herramientas que permitan el montaje de las pantallas de prototipo de interfaces. Un Arquitecto de Información es como un puente entre la tecnología, las personas y los procesos.

1.8 Conclusiones

La Arquitectura de Información es una poderosa herramienta para crear productos electrónicos de alta calidad, y cuyo objetivo fundamental es satisfacer al usuario que vaya a interactuar con la aplicación. Términos como usabilidad y accesibilidad son claves para la creación de un buen software.

Mientras que la información puede ser infinita, las formas de estructurarlas no, y una vez clasificada la primera, se vuelve mucho más útil. La Arquitectura de Información reduce el tiempo de construcción de un producto, permite minimizar malas decisiones que arruinen el desarrollo de este, además, disminuye considerablemente la Gestión de los Contenidos y rediseños pues una vez que se conoce al usuario, se realiza todo en base a éste.

En el proceso de Arquitectura de Información es necesaria la representación de la información, para tener una idea de cómo quedará el producto final y no esperar a que este esté listo para mostrarlo a los clientes, ya que puede pasar que los mismos no estén de acuerdo con algo y entonces se tenga que retornar al proceso de construcción. En este sentido se utilizan prototipos para ilustrar cuál será la

organización y estructuración visual de los diferentes elementos del futuro software, y con ello validar la propuesta con los Clientes y Usuarios finales.

Por las características del proyecto CICPC se elaborarán prototipos rápidos para el diseño de las pantallas de los Casos de Uso. También los que se clasifica en prototipo de alta fidelidad, experimental, vertical y local por cada caso de uso. El nivel de prototipado será estático, pues no permitirá la interacción de las pantallas, sin embargo se creará un mapa de navegación que permita saber los diferentes flujos en un CU. Para la creación de estos prototipos se utilizará el Microsoft Visio 2003 que es una herramienta visual de modelado fácilmente personalizable, y permite además la creación de una librería de componentes.

La evaluación es un proceso fundamental y en este sentido, la Arquitectura de Información debe también comprobarse para conocer si se cumplen los objetivos de esta. Las Listas de Chequeo son un instrumento de medición de la calidad, que en el caso del proyecto se creó una para la Arquitectura para comprobar la correcta elaboración de prototipos.

El objetivo de este trabajo es vincular la Arquitectura de Información a un Proceso de Desarrollo de Software siguiendo la metodología RUP. Es por ello que también se hizo un estudio de esta para buscar los puntos en contacto con la Arquitectura de Información.

CAPÍTULO 2. PROPUESTA DE ARQUITECTURA DE INFORMACIÓN

2.1 Introducción

El siguiente capítulo integra, con un enfoque práctico, la Arquitectura de Información al Proceso de Desarrollo de Software, definiendo procesos, actividades, artefactos y roles responsables de cada flujo de trabajo analizado. La metodología RUP no define la Arquitectura de Información dentro de su proceso de desarrollo pero si incluye procesos cuyos objetivos están encaminados a la misma función de la Arquitectura de Información. Es por esa razón que en algunos casos se han incluido nuevos procesos y en otros se han insertado actividades dentro de procesos ya existentes en la metodología.

2.2 Arquitectura de Información y RUP

La Arquitectura de Información consta de procesos que tienen lugar en diferentes actividades o procesos dentro del desarrollo de un producto software. Ellas son:

- Levantamiento de Información.
- Desarrollo de las pautas de Arquitectura de Información.
- Diseño de Prototipos de Interfaces de Usuario.
- Evaluación de la Arquitectura de Información en el Software.

Si se ubican dentro del proceso de desarrollo de software quedaría distribuida su carga de trabajo durante las distintas fases de RUP (Figura 2).

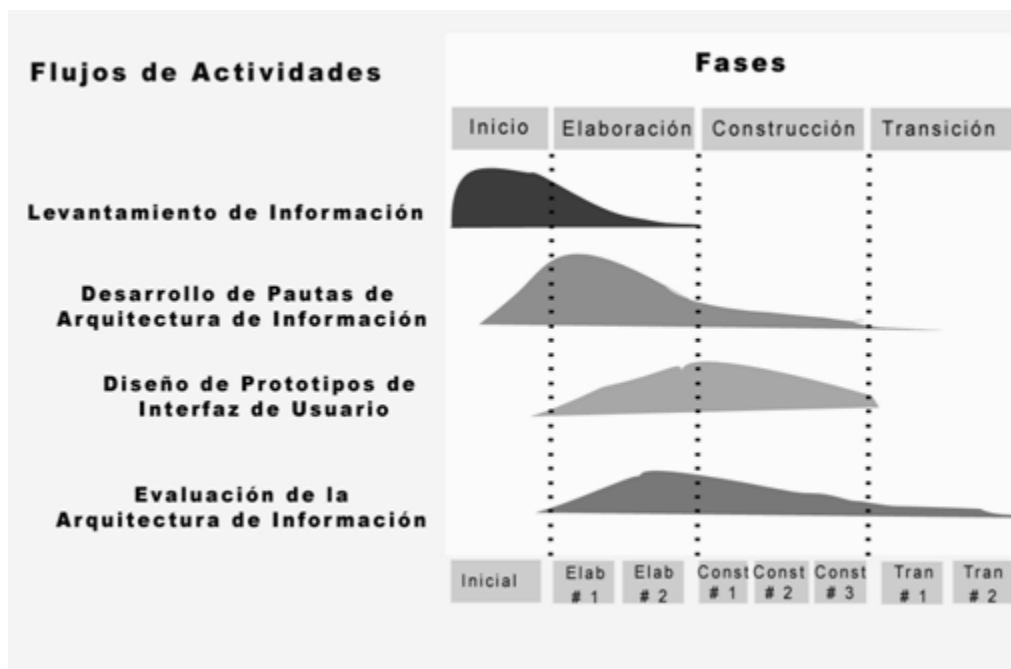


Figura 2. Actividades de la Arquitectura de Información ubicadas en RUP.

El Levantamiento de Información ocurre fundamentalmente en el Flujo de Trabajo Modelamiento del Negocio ya que este último, tiene como objetivo fundamental comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema, identificar las mejoras potenciales y asegurar un entendimiento común entre clientes, usuarios finales y desarrolladores, objetivo que en gran medida persigue también el Levantamiento de Información. Este se realiza en las fases iniciales del proyecto, específicamente durante las Fases de Inicio y Elaboración.

El desarrollo de Pautas de Arquitectura de Información tendrá lugar en el Flujo de Trabajo Requerimientos, el cual tiene como objetivos definir el ámbito del sistema; definir una interfaz de usuarios, enfocada a las necesidades y metas de la audiencia; establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el software debería hacer; y proveer a los desarrolladores un mejor entendimiento de los requerimientos. Ya que el desarrollo de Pautas de Arquitectura de Información persigue establecer una guía para el desarrollo de un software funcional y altamente usable, se inserta dentro de este flujo de trabajo, principalmente en las Fases de Inicio y Elaboración y en menor medida en las Fases de Construcción y Transición, si hay algo que necesita ser nuevamente pautado.

El Flujo de Trabajo de Análisis y Diseño tiene como objetivo fundamental traducir los requisitos a una especificación que describe cómo implementar el sistema y tiene en cuenta el cumplimiento de los requisitos funcionales y no funcionales. El diseño de Prototipos de Interfaz de Usuario pretende

representar y validar los requisitos de la aplicación antes de la construcción de las interfaces gráficas, es por ello que se inserta dentro del Flujo de Trabajo Análisis y Diseño, y presenta un mayor esfuerzo en las fases de Elaboración y Construcción.

El Flujo de Pruebas le presta servicios a los demás flujos. Su principal objetivo es evaluar o valorar la calidad del producto a través de la búsqueda y documentación de errores, validar el cumplimiento de los requerimientos, validar el desempeño y dar una indicación de su eficacia. Por su parte los procesos de la Arquitectura de Información deben ser evaluados para comprobar que se cumplan sus objetivos. Es por ello que el proceso de evaluación de la Arquitectura de Información ocurrirá en el Flujo de Trabajo de Prueba, en las fases de Elaboración, Construcción y Transición.

2.3 Levantamiento de Información

Durante el Modelamiento del Negocio, RUP define procesos encaminados a cumplir con los objetivos de este flujo de trabajo. Pero ninguno de ellos abarca la Arquitectura de Información y todo el trabajo que se debe desarrollar en esta fase. Esta propuesta incluye por tanto, un nuevo proceso Levantamiento de Información (Figura 3).

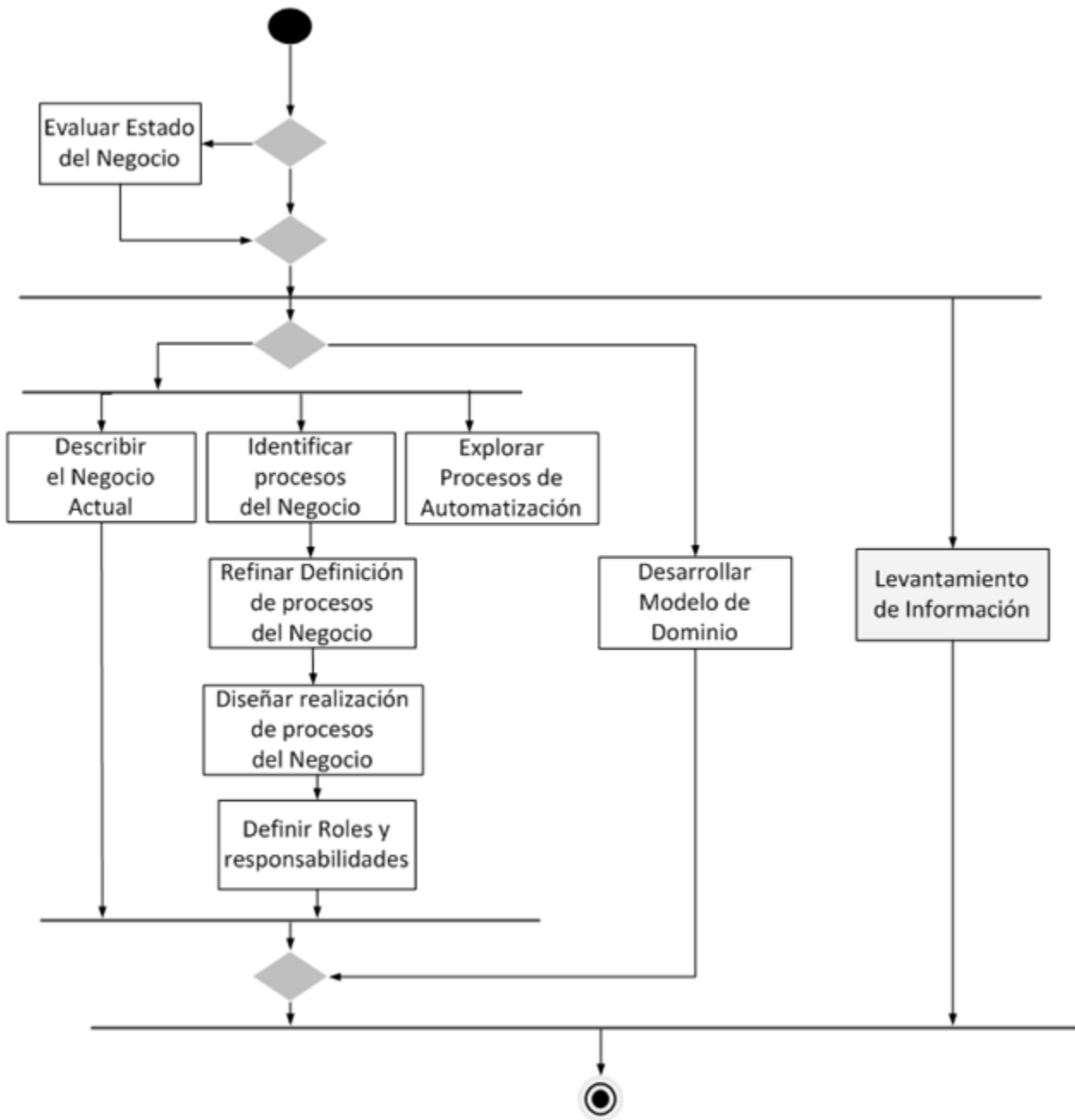


Figura 3. Diagrama de Procesos del Flujo de Trabajo Modelamiento del Negocio incluyendo la Arquitectura de Información.

Lograr una comunicación efectiva entre los Usuarios y el Equipo de Proyecto, así como entre los miembros de este último, con el objetivo de llegar a un entendimiento de lo que se debe hacer, es la clave del éxito en la producción de un software. En muchos casos, innumerables aplicaciones han fallado por las incongruencias entre lo que el Usuario quería, lo que realmente necesitaba, lo que interpretaba cada miembro del Equipo de Proyecto y lo que realmente se obtenía. En esto, radica la

importancia que en los últimos años se ha reconocido del Levantamiento de Información. En el Levantamiento de Información se debe hacer una auditoría exhaustiva de todos los procesos que ocurren en la organización teniendo en cuenta a los Usuarios finales del producto a construir y al Cliente; y se realizan actividades encaminadas a cumplimentar los objetivos de esta, las cuales se reflejan en artefactos de salida (Figura 4).

Paralelamente a este proceso, se realiza un Modelo de Negocio o de dominio en dependencia de las características de la organización en la cual el sistema será explotado.

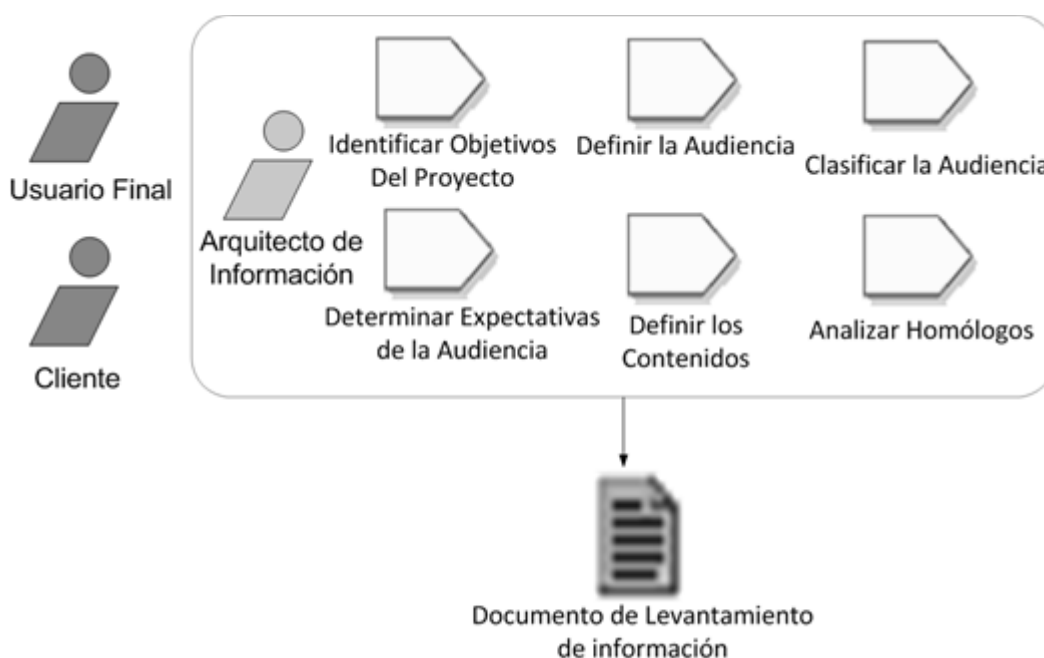


Figura 4. Detalles del Proceso Levantamiento de Infomación.

2.3.1 Actividades en el Levantamiento de Información

2.3.1.1 Objetivos del Proyecto

Para realizar un buen Levantamiento de Información es imprescindible determinar los objetivos de la aplicación a confeccionar y de esta forma establecer la forma de cumplirlos. En esta etapa, se analizan aspectos como:

- La misión y visión de la organización.
- La posibilidad que el software tendrá de aportar al cumplimiento de esa misión.
- El estudio de la factibilidad de reconsiderar la misión de la organización por el avenimiento del nuevo medio del ciberespacio.
- Planeación del desempeño de la aplicación en el futuro.

En el caso de nuestro proyecto el objetivo general que rige el desarrollo de la aplicación informática es desarrollar e implantar un sistema de información policial utilizando las nuevas plataformas tecnológicas, que soporte las decisiones estratégicas del Ministerio Interior de Justicia y la Dirección general del CICPC, que ayude a controlar y organizar el trabajo en las dependencias del CICPC, y mejore el nivel de respuesta a las necesidades de seguridad del ciudadano venezolano. Dicho objetivo está sustentado en la misión y visión del CICPC.

Una vez definidos y esclarecidos los objetivos del software en concordancia con la misión de la organización y las necesidades de los usuarios se tendrá un mayor conocimiento de lo que se persigue al desarrollar el producto y con ello se podrá idear con una visión óptima.

2.3.1.2 Definición de la Audiencia

La audiencia es el grupo de personas (Usuarios reales o potenciales, Públicos) a los que está dirigido el producto, estos pueden estar divididos por categorías o características similares, y se realiza una breve caracterización de las necesidades de esta audiencia que se pretenden satisfacer con el producto.

Una vez que se ha terminado adecuadamente la etapa anterior con la generación de los objetivos del sistema, se debe dar un siguiente paso, que es el de determinar el principal público hacia el cual se orientará el producto software.

Se clasifica a la Audiencia por los siguientes aspectos:

- Por Capacidad física. Se deberá estudiar si el público del software incluirá personas con discapacidades físicas (visual, auditiva, etc.), y si sucede así se debe prever que el sistema permita el acceso a ellos, a través del cumplimiento de las normas de Accesibilidad que se han recomendado como estándares internacionales.

Se realizaron para esto, cuestionarios a los públicos objetivos de la organización con el objetivo de clarificar el porcentaje de personas con ciertas discapacidades físicas, y así prever que nuestro producto cumpla con principios de accesibilidad y usabilidad para todos los usuarios.

- Por Capacidad técnica. No todos los usuarios que utilizarán el software a construir tienen la misma experiencia en el uso de aplicaciones y la computación en general. Es por ello que se debe estudiar este aspecto y de esta forma se dividirá la audiencia de acuerdo con la experiencia técnica que tenga cada grupo.

En este sentido, actualmente en no todas las dependencias del CICPC existen medios informáticos que los trabajadores de la organización utilicen, es por ello la desigualdad de conocimientos que existe entre la audiencia del producto futuro. Se prevé que nuestro software

funcione en cada una de las dependencias de la organización en todo el país, llevando a cabo una gran transferencia tecnológica, y con ello una capacitación general en la utilización del producto. Por esta razón, se establecieron niveles de conocimiento entre los usuarios.

- Por Conocimiento de la Institución. Los usuarios se dividen entre quienes conocen la institución y quiénes no. Por lo anterior, los primeros siempre sabrán dónde buscar lo que necesitan usando la terminología, siglas y nombres de departamentos internos; los segundos, en tanto, no entenderán nada de la nomenclatura interna y les será muy difícil acceder a la información que se les ofrezca de esa manera.
- Por Ubicación Geográfica. La investigación policial se realiza en Venezuela en disímiles dependencias distribuidas por todo el país. Nuestro software debe acortar los canales de comunicación actuales entre las distintas dependencias que no se encuentran físicamente cerca y necesitan compartir información para efectuar su trabajo. (Disminución de correspondencia de Memorándums y Oficios entre las dependencias). De ahí la importancia de estudiar este elemento para clasificar la audiencia.
- Por Necesidades de Información. El sistema deberá dar respuesta a las necesidades de Gestión de los procesos, Control de la Información y apoyo a la Toma de Decisiones del Cuerpo de Investigaciones Científicas, Penales y Criminalística. Debe dar acceso a una información organizada y variada, con un nivel de actualidad bastante confiable, teniendo en cuenta los niveles de confidencialidad requeridos para cada tipo de información. Además se debe tener en cuenta las clasificaciones del público interno en sus diferentes niveles: ejecutivo, gerencial, táctico y operativo.

2.3.1.3 Necesidades de la Audiencia

Al clasificar la audiencia por los criterios antes mencionados, se debe también indagar en las necesidades de información que tengan cada una, para lograr que nuestro software cumpla con las expectativas de cada uno de sus usuarios potenciales. Para ello se tiene que hacer un inventario de las actividades que realizan los diferentes usuarios, qué nivel de acceso tienen a la información y cómo se realiza el proceso de gestión de ésta.

2.3.1.4 Expectativas de la Audiencia

Una vez que se define la audiencia y las necesidades de ésta, se logra en gran medida saber lo que espera el cliente con la construcción del software. Primeramente debe ser capaz de cumplir todas las acciones que actualmente se realizan en la institución además de brindar un valor agregado al

producto, para que los usuarios acepten el software por lo novedoso, usable, por las grandes ventajas que ofrece el sistema actual con respecto al que tenían en principio.

2.3.1.5 Definición de los Contenidos

Una vez que se tuvo una visión general de los procesos que existen en el Negocio, se permitió comprender a qué se dedica el mismo, así como establecer una comunicación entre Cliente y Equipo de Desarrollo para definir qué es lo que debe hacer el software, siendo necesario realizar la Captura de los Requerimientos que el sistema debe cumplir, labor de la que se encargan normalmente los analistas.

En la Captura de Requerimientos se realiza la definición de los contenidos que se traducen, en la Ingeniería de Software, en los Casos de Uso del Sistema. Estos son los que recogen las funcionalidades que el sistema debe tener.

Muchas veces los requerimientos, existen en documentos generados anteriormente por la institución, como es el caso del CICPC, que por ser una institución de más de 30 años de trabajo, ha generado documentos administrativos que muestran su funcionamiento. Por esta razón, en esta etapa el Arquitecto de Información realiza un inventario de los contenidos teniendo en cuenta la categoría de estos (Fuentes, Servicios y Sistemas), nombre, el formato en que se encuentra, la actualización y disponibilidad, así como el responsable de facilitarlo en un momento determinado. Todo esto es para tener una noción global de las fuentes que nutrirán a los analistas para la Especificación de los Casos de Uso y que ofrecerá el sistema al Usuario final.

2.3.1.6 Técnicas utilizadas en el Levantamiento de Información

Para obtener la información en esta etapa se aplicaron técnicas de interacción con el usuario que facilitaron el trabajo de manera simple y auténtica. En la aplicación de estas técnicas, participó el Arquitecto de Información.

1. **Técnicas de Interacción con el Usuario:** son las técnicas donde se obtienen las informaciones relacionadas con los usuarios del producto final. Estas informaciones constituyen la base para lograr un diseño centrado en el usuario, es aquí donde se obtiene la información para las posteriores etapas de producción.
 - **Reunión:** encuentros realizados durante la captura de requerimientos, que tenían como finalidad conocer cómo se desarrollaba el flujo de trabajo e información entre los trabajadores del CICPC. También permitió a los analistas determinar requerimientos que les interesan a los futuros clientes.

- **Entrevista y encuesta:** contactos personalizados con usuarios, de manera oral o escrita. Con cada una se obtienen informaciones que se puede analizar de manera cualificada o cuantificada. Estas técnicas fueron utilizadas para clarificar la manera en que los analistas pensaban cómo se desarrollaría el sistema.
 - **Diseño de Escenarios:** es la aplicación de encuestas a usuarios donde se les solicita que definan el orden de las acciones que realizan para lograr algún objetivo específico, con la finalidad de poderlo representar metafóricamente en el producto electrónico. La razón de esta técnica es obtener las secuencias lógicas definidas directamente por los usuarios. Sirve, igualmente, para crear las metáforas (ver Glosario de Términos) funcionales y visuales que se incluirán en el producto.
 - **Análisis Documental:** en muchos casos se revisaron documentos de la institución para la búsqueda de determinada información.
2. **Técnicas de Interacción con el Contexto:** son las técnicas que buscan información de productos similares o productos que le hacen competencia al que estamos realizando. El objetivo principal de esta técnica es conocer qué potencialidades tienen los productos similares o de la competencia, para poderlos mejorar y superar, o qué dificultades tienen estos productos revisados, para no repetirlos en el nuestro y para darle soluciones y convertirlas en ventajas.
- **Evaluación de productos similares:** Se hace necesaria la revisión de productos similares al que se va a realizar, ya sea esa similitud, en aspectos de contenido, diseño o programación. Para realizar esta técnica es importante tener bien claros los objetivos del producto, proceso éste que se realiza desde las primeras etapas. Con los objetivos claros, se localizan productos que cumplan objetivos similares o que tengan semejanzas al nuestro. Luego se definen indicadores para evaluar a partir de los mismos. Ejemplos de indicadores pueden ser: los niveles de navegación, mantenimiento de la imagen de diseño en todo el producto, buena funcionalidad, etc. Finalmente se evalúan estos indicadores en cada producto escogido y se tabulan los resultados para una mejor comprensión de los mismos.
- Dentro de esta técnica se utilizó el Análisis de Homólogos. Se evaluaron las funcionalidades de la aplicación informática conocida como SIIPOL, la cual es la utilizada actualmente para gestionar la información en la institución.
- También se analizó la aplicación *Sistema Automatizado Jurídico Operativo (SAJO)* perteneciente al Ministerio del Interior de la República de Cuba.

- **Análisis de la Competencia:** se diferencia de la técnica anterior por el enfoque, o sea, por el punto de vista con que se mire la técnica, así como la amplitud del análisis. El análisis de la competencia no comprende solamente a los productos sino también a las instituciones. Por ejemplo: entre la competencia pueden existir instituciones que no tengan productos como el que se va a realizar, y es importante conocer la razón de esto, porque la competencia puede haber fracasado en un proyecto similar y es conveniente conocer las razones para que no se cometan los mismos errores. En este sentido las instituciones de esta índole no publican información referente a las funcionalidades de sus aplicaciones informáticas. No obstante se realizó un inventario de las organizaciones que presentan software para el control de la investigación.

Al concluir el proceso de Levantamiento de Información, se tiene un enfoque claro de los procesos que ocurren en la institución, el conocimiento de sus principios, misión, visión y proyección de la organización. Se conoce a los usuarios a quienes va dirigido el software con la finalidad de brindarles un mejor servicio a la medida de sus expectativas. Se hace un estudio de la competencia con el fin de tomar los aspectos positivos de estos. Toda esta información es utilizada en las siguientes fases de desarrollo del producto.

2.3.2 Artefactos generados en el Levantamiento de Información

Toda la información recopilada en las actividades antes explicadas, se estudia minuciosamente y se recoge en un documento formal denominado Documento de Levantamiento de Información (se encuentra en el Expediente del Proyecto).

Este documento tiene como objetivo compendiar todo el cúmulo de información obtenida, de tal forma que el Equipo de Desarrollo centre la producción del software en lograr las metas que se propone la institución, y a la altura de los usuarios que van a interactuar con el producto.

2.4 Desarrollo de Pautas de Arquitectura de Información

En RUP, el Flujo de Trabajo Requerimientos se desarrolla de la siguiente forma:

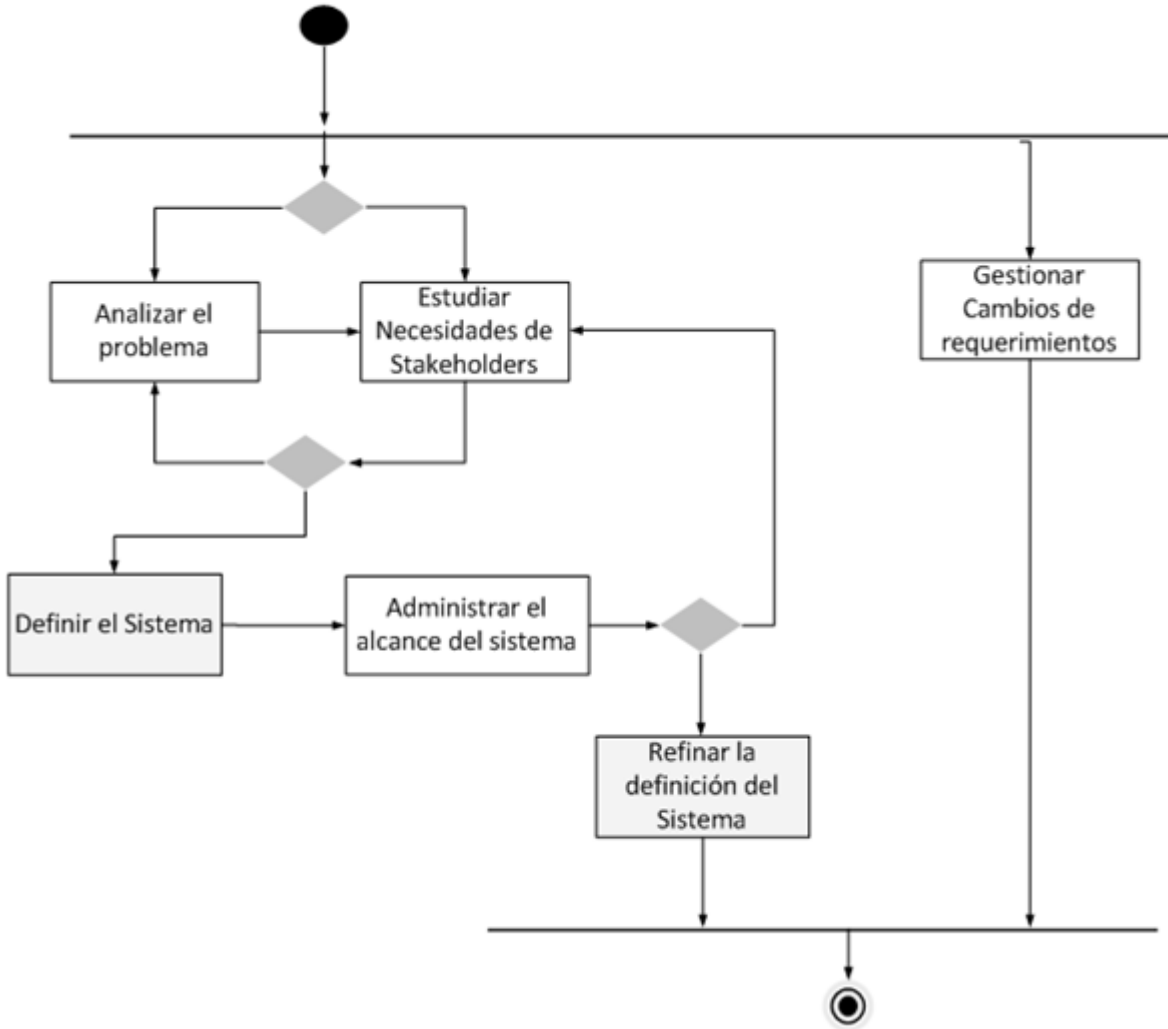


Figura 5. Diagrama de Procesos del Flujo de Trabajo Requerimientos.

Dentro los procesos que define RUP para este flujo de trabajo, se encuentran Definir el Sistema y Refinar la Definición del Sistema. Según las características de dichos procesos y las actividades que se realizan en ellos, se pueden ubicar nuevas actividades encaminadas a la Arquitectura de Información, sin necesidad de crear un nuevo proceso general para este flujo de trabajo. Precisamente, el Desarrollo de Pautas de Arquitectura de Información es la actividad fundamental que se incorpora a la metodología en estos dos procesos durante el Flujo de Trabajo de Requisitos.

La siguiente figura representa los detalles del Subproceso de Desarrollo de Pautas de Arquitectura de Información (Figura 6).

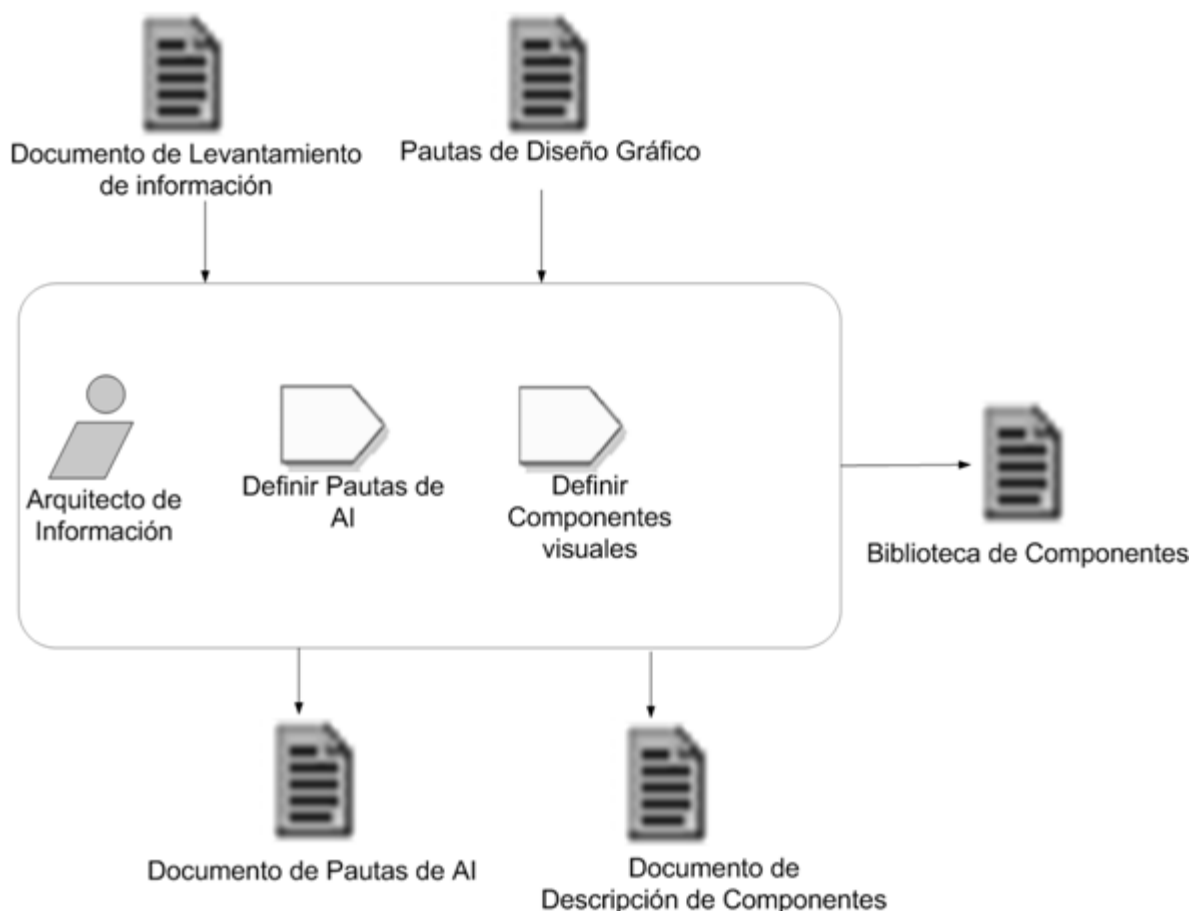


Figura 6. Detalles del Subproceso Desarrollo de Pautas de Arquitectura de Información.

Para cumplimentar la actividad Desarrollo de Pautas de Arquitectura de Información se deben realizar tareas que tributen a ella, como las que se explican a continuación.

2.4.1 Definición de Pautas de Arquitectura de Información

El Levantamiento de Información es un proceso imprescindible para el Desarrollo de Pautas de Arquitectura de Información, es sin duda una entrada importantísima para definir los lineamientos de Arquitectura de Información del proyecto.

Todo el proceso de definición de Pautas de Arquitectura se trasluce en un artefacto desarrollado en este flujo de trabajo, denominado Documento de Pautas de Arquitectura de Información. En él se especifican todas las características que desde el punto de vista de la Arquitectura de Información

tendrá la aplicación, con esto se logra que el software presente uniformidad a la hora de visualizar la información y que sea totalmente funcional para el usuario (Ver epígrafe 2.4.3.1).

2.4.2 Definición de Componentes visuales

En esta actividad tiene fundamental importancia las Pautas de Diseño Gráfico y la selección de la herramienta que se usa para diseñar las interfaces visuales. Existen disímiles herramientas para esta labor, su selección debe estar basada en la robustez de esta, en la usabilidad de los componentes que brinde y las facilidades de composición y/o elaboración de nuevos componentes.

En el proyecto, en específico la Arquitectura de Software, estableció el uso de JavaServer Faces (JSF), framework altamente robusto para aplicaciones Java basadas en la web. Ofrece una clara separación entre el comportamiento y la presentación, lo que permite a cada miembro del Equipo de Desarrollo de una aplicación Web enfocarse en su parte del Proceso de Desarrollo, y proporciona un sencillo modelo de programación para enlazar todas las piezas. Sin embargo, esta herramienta no simplifica el trabajo de diseño de interfaces, pues no presenta una Biblioteca de Componentes gráficos que permita a alguien sin conocimientos de JFS montar una interfaz de usuario rápidamente. Por esta razón se hizo un inventario de los componentes necesarios a utilizar en el software para seleccionar este framework, teniendo en cuenta, desde el punto de vista de Usabilidad cuales se podrían utilizar y se analizó la composición y la elaboración de nuevos componentes que ofrezcan una mayor funcionalidad y riqueza al sistema.

Para realizar un verdadero análisis de la Arquitectura de Información en el proyecto se debe trabajar en concordancia con el Diseño Gráfico y la tecnología a usar en el Desarrollo del Sistema, pues Arquitectura de Información y Diseño Gráfico al unísono, podrán proveer al software la calidad requerida en cuanto a la forma de visualizar la información. Por ello, de los componentes que brinda JSF, se escogieron para su uso, aquellos que cumplen con los principios de Diseño Gráfico pautados para el proyecto CICPC, además, se crearon otros nuevos, todos según los criterios de Arquitectura de Información.

2.4.3 Artefactos Generados en el Desarrollo de Pautas de Arquitectura de Información

2.4.3.1 Documento de Pautas de Arquitectura de Información

La UCI definió un formato para la Arquitectura de Información orientada a Portales y Sitios Web. Al ser necesaria su extensión para la aplicabilidad a Software de Gestión, se elaboró por parte de la autora, una nueva propuesta que cumpliera con las necesidades del proyecto CICPC.

Este documento (ver Anexo I), establece las pautas de organización, diagramación, jerarquización de la información que se muestran en las interfaces del Software de Gestión Policial SIIPOL.

El objetivo de esta plantilla es unificar los criterios que existen y garantizar que todos los productos salgan con la misma estructura y calidad.

El documento muestra una representación simple de la estructura de la aplicación en cuanto a etiquetas y jerarquía de los contenidos. Describe textualmente cada uno de los elementos de la estructura, características y comportamientos, entre otros. Evidencia una representación en forma de árbol de secciones, niveles y contenidos relacionados para definir dicha estructura.

Contiene además, una descripción detallada de los componentes que se utilizan en la herramienta Visio para elaborar los Prototipos de Interfaces de Usuario.

El artefacto Pautas de Arquitectura de Información se divide en diferentes secciones.

- **Elementos de la Estructura o Taxonomía (Esbozo y Descripción)**

Es una representación simple de la estructura de la aplicación en cuanto a etiquetas y jerarquía de los contenidos. En esta sección también se incluye una descripción textual de cada uno de los elementos de la estructura, características y comportamiento de los mismos.

- **Definición de la Estructura**

Se establecen las principales etiquetas que están presentes en las Interfaces de Usuario, se reglamenta la forma de visualizar la información en cuanto a orden, organización de los contenidos. Se especifica la tipografía, puntaje de letra, si los términos aparecen capitalizados o no.

- **Sistema de Navegación**

Un mapa de navegación es la Representación Gráfica de la Organización de la información de una estructura web. Expresa todas las relaciones de jerarquía y secuencia y permite elaborar escenarios de comportamiento de los usuarios.

En este acápite se realiza un mapa de navegación general para dar una idea global del funcionamiento del software.

Por la complejidad y la extensión de nuestro software se decidió realizar un Mapa de Navegación para cada Caso de Uso, de modo que represente la interacción entre las diferentes

pantallas o escenarios de este y en el acápite se establecen las pautas para la realización de mapas de navegación a nivel de Casos de Uso.

- **Diseño de la Estructura de las Páginas Tipo**

En esta sección se muestran las plantillas básicas para el modelado y diseño de las interfaces. Además se brinda la dirección de donde se pueden ver las Plantillas de guía definidas por Diseño Gráfico para comprender mejor las Pautas de Diseño.

- **Diagrama de Interacción**

Un diagrama de interacción se entiende como la representación de secuencias de intercambios y mensajes entre los roles que participan y se relacionan con un sistema, sea éste un sitio Web, programa, aplicación o software (Ferrer, 2005).

Debido a la complejidad del software se realiza un diagrama de interacción que abarque solo los escenarios principales de la aplicación con el objetivo de representar el Flujo de Información Usuario-Software. Este diagrama se ha realizado en el Visio 2003.

- **Descripción de los Elementos que componen las Pantallas**

- Botones e Hipervínculos.

Un botón es una metáfora común utilizada en interfaces gráficas con objetivo similar al de un botón corriente. Los botones suelen ser representados como rectángulos con una leyenda o ícono dentro, generalmente con efecto de relieve.

Los botones se emplean para las acciones a realizar en las interfaces de usuario y para las transacciones en el sistema, por ser estos los que denotan más relevancia.

La importancia de definir esta sección dentro del Documento de Pautas radica en que si no se unifican los criterios a la hora de tipificar los botones, definir cómo y dónde se van a ubicar, no se logrará uniformidad dentro de la aplicación.

Se plasma en este acápite además, la ubicación dentro de una página de estos elementos. Se define, en caso de tener más de un botón, la prioridad de estos dentro de la interfaz de usuario y el orden. Además se indica que debe predeterminarse un botón, de manera que el usuario sepa que esa acción va a ser la fundamental dentro de esa interfaz y a fin de que el usuario pueda hacer uso del teclado y no dependa necesariamente del mouse.

En el software, de un Caso de Uso Base se puede navegar a otro para acceder a las funcionalidades disponibles en este último. En ese caso, el vínculo para acceder a esta información es mediante Hipervínculos, el cual se define normalmente como una conexión entre una información y otra dentro de un documento HTML (nuestro software es para ambiente

web), que infieren la realización de acciones menos importantes, opcionales o prescindibles para la aplicación.

En este sentido también se pauta donde se ubican los hipervínculos dentro de la interfaz, de qué color se mostrarán antes y después de ser accedidos. Además de algunos ejemplos de nombres para establecer uniformidad.

➤ Tablas

Se definen y explican las tablas que se van a utilizar y la razón para cada una de ellas.

Con el objetivo de que al usuario final le sea más fácil usar la aplicación y recordar la forma en que funciona, se definieron dos tipo de tablas con marcadas diferencias entre sí; una que va a mostrar los datos predeterminados con la información contenida en el sistema, y la otra muestra los datos cargados dinámicamente a la hora que el usuario realice una consulta en la aplicación.

Para no saturar demasiado las interfaces, y teniendo en cuenta la magnitud de información existente en CICPC, se definen la cantidad de resultados que serán mostrados por página de resultados y cómo será la navegación de una a otra.

➤ Mensajes de Información

Los mensajes de información sirven para notificar al usuario cualquier evento que ha ocurrido en el sistema. Estos no solo ocurren cuando algo ha fallado en el sistema, sino también para informar alguna advertencia, atención o error que se pueda presentar al ejecutar alguna acción.

Para que el usuario note la diferencia de cada uno de estos mensajes, se definen colores para cada tipo de ellos.

2.4.3.2 Librería de Componentes

Tiene como objetivo estandarizar y agilizar el trabajo del modelado de prototipo de interfaces de usuario en las etapas futuras de Desarrollo del Software. Para su elaboración, se utilizó el Visio 2003, programa que brinda la posibilidad de utilizar las librerías que contiene por defecto, personalizarlas y crear nuevas, además por ser la herramienta para modelado de prototipo seleccionada en el proyecto. Esta Librería contiene todos los componentes que desde el punto de vista de Diseño y Arquitectura de Información fueron seleccionados para usarse en el software, logrando personalizarse con las Pautas del Diseño establecidas en el proyecto hacia una representación exacta de los componentes que se usan en la herramienta para diseñar las interfaces de usuario en JSF. Esto permite que el usuario tenga un mayor acercamiento a las interfaces reales sin necesidad de programarse un prototipo dinámico. (Ver imagen de la misma en Anexo II).

2.4.3.3 Documento de Descripción de Componentes

Paralelamente al desarrollo de la Librería de Componentes se realiza un documento que especifique, para cada uno de los componentes de la librería, su uso en la aplicación (se encuentra en el Expediente del Proyecto). Este documento sirve de guía para diseñar los prototipos de interfaces de usuario, pues expresa de cada componente visual, la función que realiza dentro de la pantalla, así como ejemplos correctos e incorrectos para su mayor entendimiento. El formato de justificación de uso de cada componente se divide en:

Componente. Se especifica el nombre del componente en cuestión (en idioma español e inglés, porque la mayoría de los términos de las herramientas informáticas está en inglés).

Descripción (uso). Se describe de forma abreviada el uso que se le va dar a cada componente dentro de la aplicación.

Ejemplo Correcto: Una o más imágenes del comportamiento del componente en la aplicación de forma correcta. Tiene como objetivo que quien consulte este documento sepa de forma práctica cómo usarlo.

Ejemplo Incorrecto: Es una o más imágenes del comportamiento del componente en la aplicación de forma incorrecta. Permite advertir errores comunes posibles de evitar al incorporar estos ejemplos.

2.5 Diseño de Prototipos de Interfaces de Usuario

El Flujo de Trabajo Análisis y Diseño se desarrolla de la siguiente forma (Figura 7):

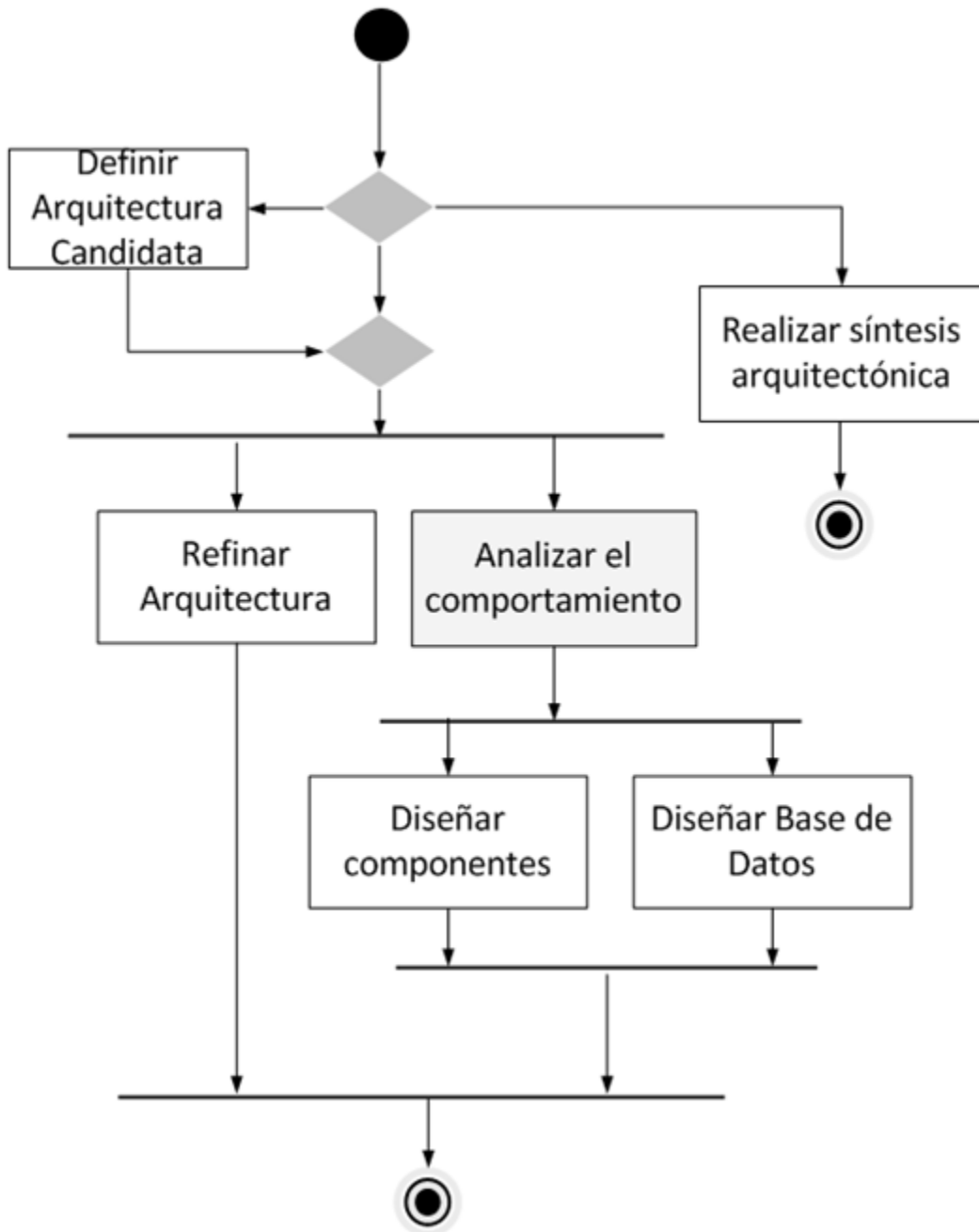


Figura 7. Diagrama de Procesos del Flujo de Trabajo Análisis y Diseño.

Dentro los procesos que define RUP para este Flujo de Trabajo, se encuentra Analizar el Comportamiento. Según las características de este proceso y las actividades que se realizan, se

pueden ubicar nuevas actividades encaminadas a la Arquitectura de Información, sin necesidad de crear un nuevo proceso general para este flujo de trabajo. RUP puntualiza en este proceso a un trabajador denominado Diseñador de Interfaz de Usuario que es responsable de las actividades Diseñar Prototipos de Interfaz de Usuario y Diseñar Interfaz de Usuario. Como nuestra propuesta define el rol del Arquitecto de Información, la actividad Diseñar Prototipos de Interfaz de Usuario es parte de las responsabilidades de éste, y no del Diseñador de Interfaz de Usuario que se encargaría solamente de diseñar las interfaces, guiándose por los prototipos previamente elaborados. No significa esto que una misma persona no pueda tener ambos roles y desarrollar por lo tanto ambas actividades en el proyecto (Ver Figura 8).

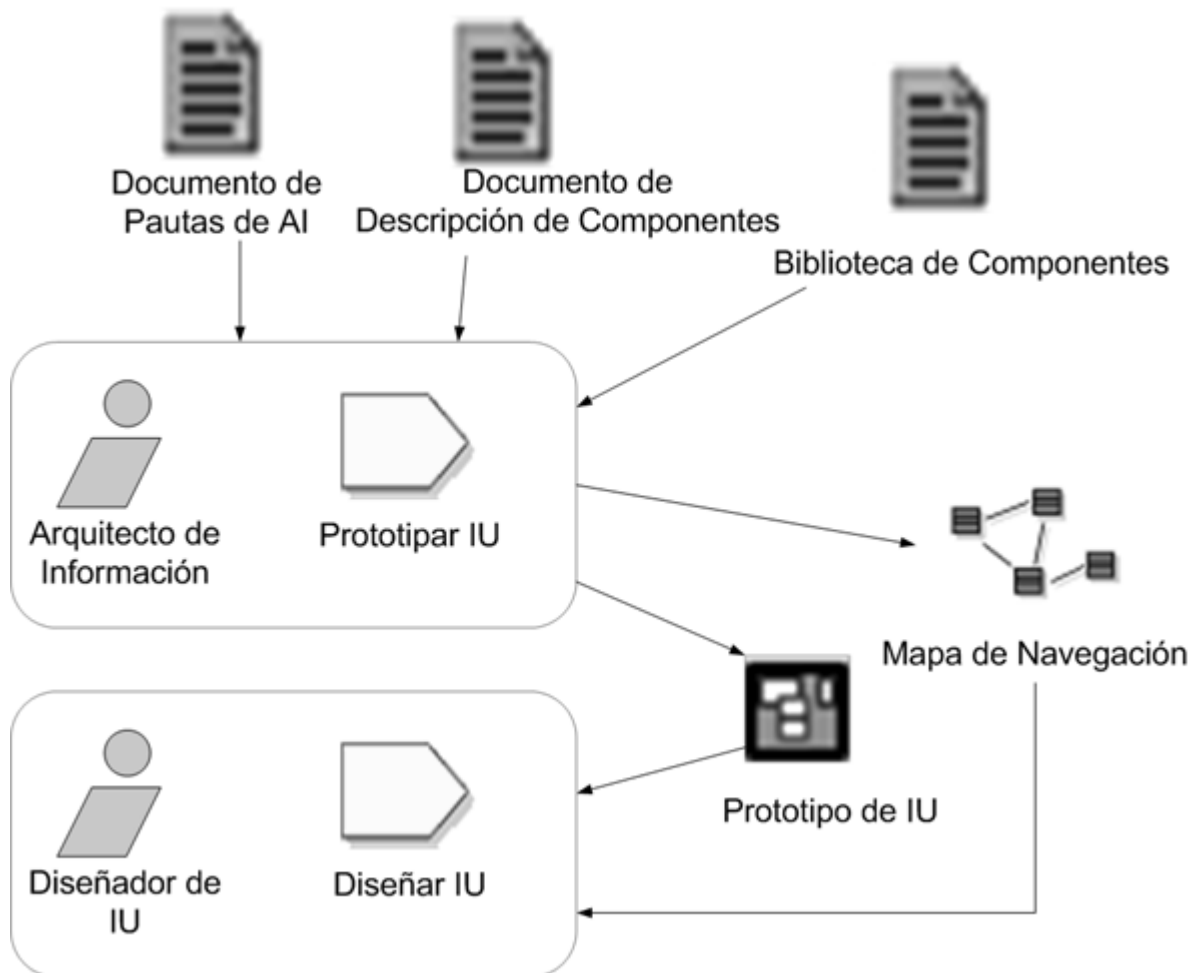


Figura 8. Detalles de Proceso Analizar el Comportamiento vinculado a la Arquitectura de Información

2.5.1 Artefactos Generados

En la Fase de Requerimientos los Analistas del sistema generan las especificaciones de los Casos de Uso del Sistema.

Un Caso de Uso es una descripción narrativa textual de la secuencia de eventos y acciones que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Dicha descripción es la entrada más importante para esta etapa ya que indica la interacción Usuario - Sistema y con esta información se trabajará para Diseñar los Prototipos.

Una vez que se describe el Flujo de Eventos de un Caso de Uso, este puede pasar a la fase de Diseño del Prototipo de Interfaz de Usuario. Este diseño se realiza en el programa de la suite ofimática Visio 2003 ya que permite crear prototipos de alta fidelidad y aporta un Diseño de Interfaces Gráficas de Usuario completo y fácil de emplear. Los prototipos creados en el Visio son dibujos que representan la interfaz visual de la aplicación, facilitan el trabajo de los Montadores de Interfaz de Usuario y de los Programadores ya que no se pasa a esta fase de implementación hasta que los Prototipos de Interfaz de un Caso de Uso no sean aceptados por el cliente. Una vez aprobados, los Diseñadores de Interfaz de Usuario solo tendrán que hacer una copia fiel de los prototipos en la herramienta que se usa para crear las interfaces.

La creación de prototipos incluye organización y visualización de la información y diseño de presentación.

El proceso de Diseño Gráfico se realiza en la UCI independiente de la Arquitectura de Información, pero siempre teniendo en cuenta los criterios que esta última brinda sobre Usabilidad y Accesibilidad. Una vez que se elaboran las Pautas de Diseño, las cuales tienen que estar en sintonía con la herramienta para crear las interfaces (JFS), se prosigue a confeccionar los prototipos de interfaz de usuario siguiendo esas pautas.

Para implementar el diseño de los Prototipos de Interfaz de Usuario se debe leer la descripción textual del CU, teniendo en cuenta las Pautas de Arquitectura de Información que se realizaron en la etapa anterior, y las de Diseño.

Se deben realizar los prototipos en el Visio guiándose por el Flujo de Eventos de la descripción textual del CU, tanto el flujo básico como el alterno. Para cada CU se debe crear un Documento de Visio cuyo nombre va a ser el del CU en cuestión y en él se agregará una página cada vez que haya un cambio en lo que se debe mostrar en la interfaz externa. Es necesario nombrar las interfaces para saber cuáles son las acciones que reflejan.

Lo que en la aplicación de cara al cliente es una pantalla, en el diseño de prototipos pueden ser varias, pues se trata que el cliente se lleve la mejor idea de cómo realmente va a funcionar el sistema; como

este prototipo no es funcional se deben crear muchas pantallas para reflejar acciones sucesivas y dependientes entre sí que realizará el sistema.

Una vez realizado el prototipo de interfaz de usuario, cada una de las pantallas se incluye en el documento de la descripción del Caso de Uso. Se analiza si el prototipo se corresponde con lo que debe hacer el caso de uso y el analista elabora una Tabla de Eventos asociada a esa pantalla. Esta tabla contendrá los campos siguientes:

Control. En este, se especifica el nombre de la acción a realizar en botones, hipervínculos, áreas de mensajes. Por ejemplo: Botón Cancelar, Botón Imprimir/ Exportar PDF, Hipervínculo Orden de Allanamiento.

Valor. Los Analistas en cada página de los prototipos marcan cada uno de los controles con un número, en la tabla entonces se incluye este número asociado a cada elemento.

Evento. Se corresponde con la forma en que va a accederse a este control. Puede ser mediante un clic, al pasarle el mouse por encima del elemento, al desplegar una lista, etc.

Acción. Brinda una explicación de lo que debe hacer el sistema al ejecutar dicho control.

Por ejemplo, al hacer clic al Botón Consultar se muestra la información que coincide con los criterios de búsqueda especificados.

Un artefacto muy útil es sin dudas el Mapa de Navegación, que expresa las relaciones de jerarquía y secuencia que se establecen entre las Interfaces de Usuario, tanto de ese mismo Caso de Uso como de los otros con los que se relaciona. Para un CU se elabora un mapa de navegación que incluya la interacción entre las diferentes pantallas prototipadas y las relaciones entre ellas y otros CU.

2.6 Evaluación de la Arquitectura de Información

La calidad es un concepto muy importante para cualquier producto, en la Industria del Software es un eslabón que determina el prestigio de una empresa.

RUP define un Flujo de Trabajo para evaluar la calidad en un proceso de Desarrollo de Software. Este flujo de trabajo se representa en la siguiente figura (Figura 9).

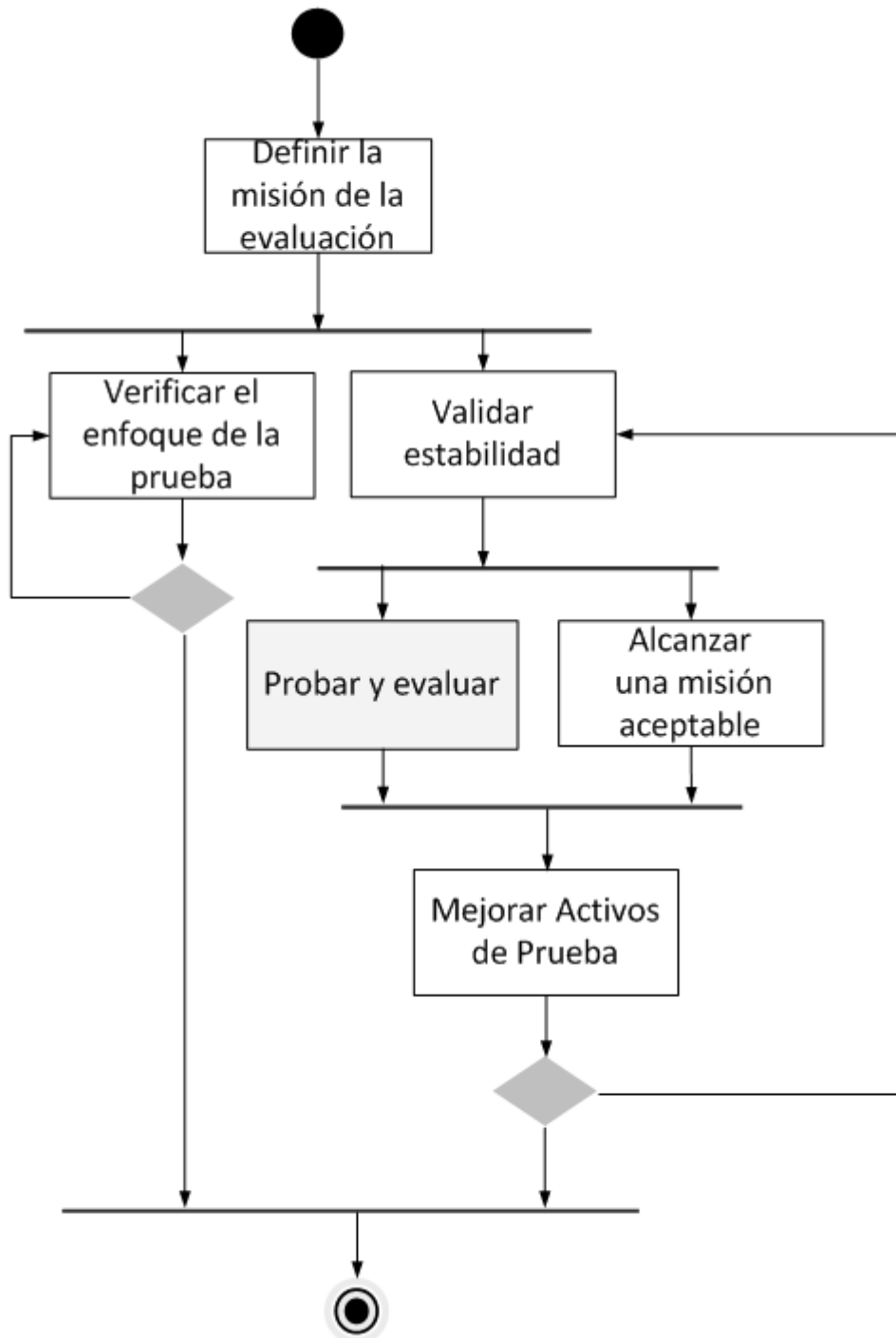


Figura 9. Diagrama de Proceso del Flujo de Trabajo de Prueba

Para la Arquitectura de Información tiene gran importancia el proceso definido por RUP en este flujo: Probar y Evaluar, ya que los artefactos que se generan al insertar la Arquitectura de Información deben

medirse y comprobar el cumplimiento de su fin. La Figura 10 refleja la vinculación de la Arquitectura de Información al Flujo de Trabajo de Prueba.

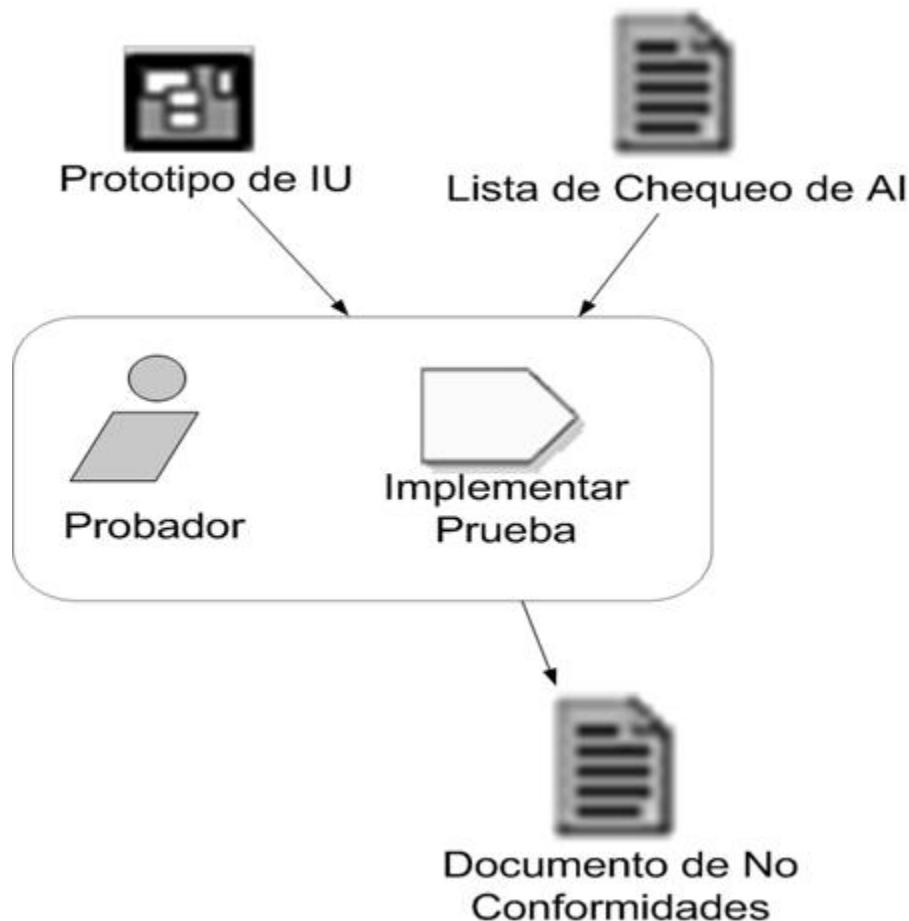


Figura 10 . Detalles del Subproceso Probar y Evaluar.

En el proyecto, la calidad es la correspondencia de los Productos desarrollados con las Especificaciones de Casos de Uso y estándares establecidos, además es el cumplimiento de Planes y Cronogramas definidos. Para garantizar su cumplimiento, un Equipo Interno de Calidad revisa todos los entregables del proyecto antes que pasen al Laboratorio de Certificación de Calidad UCI donde se vuelve a revisar el producto, manteniendo el criterio de que mientras más personas revisen el producto, existe una mayor posibilidad de que salga con mayor calidad.

La Arquitectura de Información y la Calidad deben estar relacionadas, ya que los productos que genera la primera tienen que ser revisados para comprobar su correcta funcionalidad.

Para evaluar la Arquitectura de Información dentro del software se confeccionó una Lista de Chequeo que permita valorar los Prototipos de Interfaz de Usuario elaborados y también las interfaces reales una vez construido el software.

La Lista de Chequeo es un mecanismo de revisión y comprobación de la calidad, mediante la cual el responsable de revisar los Prototipos de Interfaces de Usuario, puede identificar los errores que tengan estos artefactos. Se redactó de tal forma que el revisor entienda en cada momento, los aspectos a examinar, brindando además representaciones gráficas para dar una idea de los elementos correctos e incorrectos. Esto permite que no se deje dudas de cómo es correcto al mostrarlo visualmente en cada uno de los aspectos a evaluar.

La Lista de Chequeo contiene tópicos para que el revisor pueda medir la realización de los objetivos de forma secuencial. Refiere dos secciones importantes: una dedicada a la Interfaz de Usuario y la otra dedicada a evaluar el Mapa de Navegación.

La sección de las Interfaces de Usuario pretende validar estos artefactos de forma tal que no presenten errores por parte de la Arquitectura de Información y el Diseño Gráfico. En este sentido se comprueba que se hayan creado tantas pantallas como el CU requiera, dado por la cantidad de flujos que tenga. Además se presentan los componentes que se deben utilizar, mostrando ejemplos correctos e incorrectos.

El Mapa de Navegación por su parte, permite comprobar el Flujo de Información entre las diferentes pantallas. Para la revisión de este, se especifican pautas que permiten comprobar si está elaborado correctamente. Se dictamina que por cada Caso de Uso sólo se debe realizar un mapa de navegación, legible y con todas las pantallas modeladas en el CU. La creación del mapa debe ser en el Visio (se encuentra en el Expediente del Proyecto).

2.7 Conclusiones

En este capítulo se detalló el proceso de inserción práctica de la Arquitectura de Información dentro del proceso de Desarrollo de Software siguiendo la Metodología RUP, y su aplicación al Proyecto CICPC. Se definen 4 procesos de la Arquitectura de Información que interactúan con los flujos de trabajo y fases de RUP:

➤ **Levantamiento de Información**

Ocurre fundamentalmente en el Flujo de Trabajo Modelamiento del Negocio. Se estudia la organización en la que se implementará el sistema, las audiencias potenciales y sus necesidades de información, para lograr un entendimiento entre Clientes, Usuarios finales y Desarrolladores. Se

realiza durante las Fases Inicio y Elaboración propuestas por RUP. Se genera un documento de Levantamiento de Información que recoja todos los contenidos referentes a este proceso.

➤ Desarrollo de Pautas de Arquitectura de Información

Persigue establecer una guía para el desarrollo de un software funcional y usable. Tiene lugar en el Flujo de Trabajo Requerimientos, principalmente en las Fases de Inicio y Elaboración y en menor medida en las Fases de Construcción y Transición. En esta fase se elabora un Documento de Pautas de Arquitectura de Información que especifica la organización, diagramación, jerarquización de la información que se muestran en las Interfaces del software; una Librería de Componentes con el objetivo de estandarizar y agilizar el trabajo en las etapas futuras de desarrollo del software, ya que contiene todos los componentes que desde el punto de vista de Diseño y Arquitectura de Información fueron seleccionados para usarse en el software; y además un documento de especificación de los componentes, que describe el uso de cada uno de los elementos incluidos en la Librería.

➤ Diseño de Prototipos de Interfaces de Usuario

Pretende representar y validar los requisitos de la aplicación antes de la construcción de las interfaces gráficas, es por ello que se inserta dentro del flujo de trabajo Análisis y Diseño, y presenta un mayor esfuerzo en las fases de Elaboración y Construcción. En esta fase se obtiene, para cada CU del sistema, un prototipo que incluye una serie de pantallas que detallan el flujo de eventos de dicho CU, y un mapa de navegación que especifica la interacción entre estas páginas.

➤ Evaluación de la Arquitectura de Información

Los procesos de la Arquitectura de Información deben ser evaluados para comprobar que se cumplan sus objetivos, validar los requerimientos y dar una indicación de su calidad. Es por ello que el proceso de Evaluación de la Arquitectura de Información ocurrirá en el Flujo de Trabajo de Prueba, en las Fases de Elaboración, Construcción y Transición. En esta evaluación, se revisan los Prototipos de Interfaces de Usuario y los Mapas de Navegación pertenecientes a cada CU. Esta revisión se realiza basándose en una Lista de Chequeo, elaborada previamente por el Arquitecto de Información, que pauta los aspectos a evaluar en dichos artefactos.

CAPÍTULO 3. ANÁLISIS DE RESULTADOS Y VALIDACIÓN DE LA PROPUESTA

3.1 Introducción

Una vez desarrollada la Guía propuesta para la Arquitectura de Información en el Proyecto, es necesario validar su funcionamiento y comprobar el éxito de ella. Para cumplir este objetivo es necesario aplicar la propuesta al proyecto CICPC y luego realizar un Análisis de los Resultados obtenidos. En este capítulo se compendia la experiencia y beneficios alcanzados por el proyecto una vez que se aplicó la Propuesta de Solución.

Para validar dicho trabajo se analizaron los resultados obtenidos en las encuestas realizadas a los trabajadores que forman parte del proyecto. La muestra fue dirigida a roles en específico relacionados en algún momento con los procesos de Arquitectura de Información y por esta razón dominan elementos relacionados con la misma. Además se hizo uso del Método de Validación por Expertos para evaluar la propuesta y cuantificar el valor práctico de esta, según el criterio de los especialistas.

3.2 Aplicación de encuestas al Equipo de Desarrollo

Durante diferentes momentos en el proceso de construcción del software se aplicaron encuestas con el objetivo de determinar aspectos esenciales, tanto para la aplicación de la propuesta como para valorar en qué medida se cumplían los propósitos que se pretendían. Hubo cierta diferenciación al escoger la población de las encuestas, ya que se decidió que los roles que interactúan de forma directa con los procesos de la Arquitectura de Información, son los que deben aplicar estos conocimientos para obtener un producto del que su audiencia se sienta altamente satisfecho.

El cuestionario en ambos casos pretende medir el impacto que la Arquitectura de Información causó en el producto, así como valorar el conocimiento que tenían los encuestados sobre esta disciplina, el cumplimiento de su aplicación e incorporación al proceso de desarrollo. Se realizaron 15 preguntas en cada cuestionario, intercalando preguntas abiertas para conocer las opiniones de los encuestados sobre determinados temas, preguntas cerradas para cuantificar cuestiones específicas, además se realizaron preguntas de control, haciendo dos veces la misma pregunta de forma indirecta o distinta para comprobar la seguridad de sus respuestas.

3.2.1 Análisis de Resultados de las encuestas realizadas antes de la aplicación de la propuesta de Arquitectura de Información

Al comenzar el proyecto CICPC con la producción, se realizó una encuesta a 20 integrantes, seleccionados fundamentalmente por los roles que desempeñaban. Se escogieron Analistas, Arquitectos de Información y representantes del Equipo de Calidad Interna pues eran, los que de una forma u otra, deben interactuar con los procesos de Arquitectura de Información, y aplicaban a su trabajo, conocimientos de Ingeniería y Calidad de Software (Ver Anexo III).

El 80% de los encuestados refirieron que los Prototipos de Interfaz de Usuario presentaban poca calidad en su confección.

Un 85% dictó que no conocían el desempeño del rol de Arquitecto de Información dentro de un proyecto y el resto les dio suma importancia a estas personas dentro del proceso de construcción del software aunque hasta el momento no se desempeñaban como un rol imprescindible.

Sólo un 15% tenía conocimiento de las pautas de Arquitectura de Información que hasta el momento se habían definido, y las utilizaban; por otra parte un 55% manifestó que las pautas estaban difíciles de comprender y no las ponían en práctica y el resto afirmó que no tenían dominio de pauta alguna.

El 75% de los encuestados afirmó que no tenía conocimiento de si el Equipo de Calidad u otra persona revisaba la Arquitectura de Información (en específico los Prototipos de Interfaces de Usuario) y el resto manifestó que sí se realizaban revisiones, pero no sabía por qué principios se hacían estas.

Sobre la interrogante de la inserción de la Arquitectura de Información en el Proceso de Desarrollo de Software el 65% asintió que no tenía ni idea de este proceso, el 20% un conocimiento superficial y el resto solamente manifestó que tenía conocimiento.

El 65% de los encuestados determinó que la comunicación entre las personas que trabajan con los Prototipos o Interfaces de Usuario era mala y en algunos casos regular, mientras que un por ciento muy bajo asintió que era buena.

3.2.2 Análisis de Resultados de las encuestas realizadas después de la aplicación de la propuesta de Arquitectura de Información

El reporte de los resultados se realizó aplicando encuestas a los integrantes del proyecto CICPC. Al aplicar la propuesta de solución fueron encuestados 20 integrantes del proyecto (Ver Anexo IV).

En esta oportunidad, el 98% manifestó que el nivel de calidad que presentaban los prototipos de interfaces de usuario del proyecto había aumentado considerablemente.

El 86% opinó que los Arquitectos de Información desarrollaban un buen trabajo dentro del Equipo de Desarrollo, de ellos el 85% afirmó que el trabajo estaba bien definido entre los diferentes roles y que

esto posibilitaba que el Flujo de Información se redujera y se conocía en cualquier momento los responsables de las actividades.

Algunos encuestados que juegan el rol de Analistas del proyecto, refirieron que para ellos era más factible desarrollar los prototipos ellos mismos, pero eso no incumple con la propuesta de solución, pues en este caso, ellos tuvieran dos roles en ese momento, de Analista y Arquitecto de Información.

El 98% afirmó tener conocimiento de las pautas de Arquitectura de Información, admitiendo la importancia de conocerlas y aplicarlas día a día.

El 98% admitió que la propuesta de solución aplicada al proyecto sirve de gran ayuda para mejorar la calidad de los Prototipos de Interfaces de Usuario, agilizar y optimizar el tiempo de desarrollo del proyecto CICPC. Esto sin lugar a dudas mejoraba la comunicación entre los integrantes del proyecto involucrados en la Arquitectura de Información.

3.2.3 Tiempo de Desarrollo

La comunicación con la Líder del proyecto y con el Analista Principal permitió conocer que la inserción del Rol del Arquitecto de Información dentro del Equipo de Desarrollo del Sistema no fue una decisión desde el principio, tomada después debido a la importancia de incorporar esta disciplina al proceso de desarrollo del producto que se iba a implementar.

Además que las personas que empezaron a desarrollar este rol dentro del proyecto no tenían mucho conocimiento sobre la disciplina, pues sólo recibieron un curso de iniciación al respecto, y se desconocía cuál era el tiempo que demoraba insertar los procesos de Arquitectura de Información al proyecto. No estaban definidas las actividades que debían realizarse y para la elaboración de interfaces de usuario no se seguían pautas de ningún tipo para su diseño y desarrollo, lo que afectaba la calidad de los artefactos que se elaboraban y se exigía tiempo adicional para corregir y perfeccionarlos.

Según la planificación inicial definida en el proyecto, dependiendo de la complejidad, se le asignó un tiempo a cada CU para que fuese descrito por el Analista, luego otro para que el Arquitecto de Información crease los prototipos de interfaces de usuario para dicho CU, después, el analista comprobaba que se cumpliesen los requerimientos funcionales en dichos prototipos. Luego que estuviese conforme con ellos, los integraba a la descripción del Caso de Uso, el cual era revisado posteriormente por el Equipo de Calidad Interna.

Todo este proceso era planificado para hacerse en 9 secciones de trabajo², pero la planificación se atrasaba porque no se desarrollaban las actividades en el tiempo establecido (Ver Tabla 1).

Tabla 1. Planificación de actividades de la Arquitectura de Información en el proyecto.

Actividades	Sesiones Estimadas (h)			Sesiones Reales (h)		
	Complejidad del CU			Complejidad del CU		
	Básico	Medio	Complejo	Básico	Medio	Complejo
Especificar CU	2	2.5	3	2	2.5	3
Elaborar Prototipos de Interfaces de Usuario	1	1.5	2	2	2.5	3
Integrar CU	1	1	2	1	1	1
Revisar CU	1	1	2	1	1	1

Al aplicar esta propuesta de Arquitectura de Información al proyecto se redujo considerablemente el tiempo dedicado a las tareas de Arquitectura de Información, no produciendo afectaciones y retrasos en la planificación general del proyecto.

3.3 Guía para la Validación por Expertos

Para realizar la valoración técnica de la propuesta de Arquitectura de Información, se utilizó un método basado en la evaluación cuantitativa de criterios previamente definidos, que permite realizar un Estudio de Expertos para determinar si se acepta o no el proyecto analizado.

Para llevar a cabo la evaluación se efectuaron los pasos sugeridos para el Método de Delphi o de Expertos (Estévez, y otros, 2005).

El Método Delphi se basa en el principio de la inteligencia colectiva y trata de lograr un consenso de opiniones expresadas individualmente por un grupo de personas seleccionadas cuidadosamente como Expertos Calificados en torno al tema, por medio de la iteración sucesiva de un cuestionario retroalimentado de los resultados promedio de la ronda anterior, aplicando cálculos estadísticos.

Las principales características del método están dadas por el anonimato de los participantes (excepto el Investigador), iteración (manejar tantas rondas como sean necesarias), retroalimentación controlada,

² Una sección de trabajo en el Proyecto CICPC es de 4 horas.

sin presiones para la conformidad, respuesta de grupo en forma estadística (el grado de consenso se procesa por medio de técnicas estadísticas) y justificación de respuestas (discrepancias/consenso).

Se distinguen tres fases para la aplicación del Método:

1. Fase preliminar. Se delimita el contexto, los objetivos, el diseño, los elementos básicos del trabajo y la selección de los expertos.
2. Fase exploratoria. Elaboración y aplicación de los cuestionarios según sucesivas vueltas, de tal forma que con las respuestas más comunes de la primera se confecciona la siguiente.
3. Fase final. Análisis estadísticos y presentación de la información.

En la presente investigación se determinaron los siguientes pasos:

1. Se definen los criterios que serán utilizados en la evaluación, atendiendo a las categorías de mayor interés pero evaluando de manera independiente cada elemento en un rango de 1-10, pues cada uno expresa un elemento determinista.

Categoría: Mérito Científico

1. Valor novedoso de la propuesta.
2. Calidad de la propuesta.
3. Novedad científica.

Categoría: Implantación

4. Satisfacción de las necesidades.
5. Necesidad del empleo de la propuesta.

Categoría: Flexibilidad

6. Adaptabilidad a otros proyectos productivos con características similares.
7. Capacidad de lograr un proyecto de calidad.

Categoría: Impacto

8. Repercusión en el proyecto productivo (de acuerdo con las interfaces de usuario antes de la propuesta y después).
9. Mayor calidad en los proyectos productivos que se adapten a la propuesta.
10. Posibilidades de aplicación.
11. Impacto en el área para la cual está destinada.

2. Se realizaron dos rondas de cuestionarios para conseguir un consenso, pero con la máxima autonomía por parte de los participantes. Se utilizaron preguntas abiertas para conocer consideraciones comunes y esenciales en las valoraciones, cerradas para cuantificar los criterios a medir y preguntas de control, para comprobar la seguridad e integridad de las respuestas.

El análisis cualitativo es fundamentalmente para la primera ronda de preguntas, se leen

detalladamente cada una de las respuestas y se resumen los elementos más comunes y esenciales para elaborar la ronda siguiente.

En la segunda ronda se le suministra a cada experto las opiniones de sus colegas, y se plantean preguntas cerradas a evaluar cuantitativamente, para obtener un consenso en los resultados y una generación de conocimiento sobre el tema. Cada experto argumentará los pro y los contra de las opiniones de los demás y de la suya propia. En la segunda ronda se hace un análisis cuantitativo de las puntuaciones que los expertos dan a cada criterio.

En cuestión, para la primera vuelta, se definieron los cuestionarios a aplicar en la primera ronda, formulando una pregunta cuantitativa por criterio, permitiendo las preguntas abiertas a partir de observaciones y puntos de vista. La encuesta aplicada se muestra en el Anexo V.

3. Se realiza una selección de 7 expertos en la cual se tiene en cuenta su especialidad, grado científico, currículum, tiempo de trabajo desde la docencia y la producción aplicando el tema en cuestión para Proyectos productivos dentro de la Universidad de las Ciencias Informáticas. Se les entregaron los cuestionarios, un resumen de la estrategia y la guía propuesta.
4. Durante el análisis de los resultados de la primera encuesta se obtuvieron los resultados que se muestra en la Tabla 2.

Tabla 2. Valor otorgado por los expertos a los criterios.

C/E	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇
C ₁	10	9	10	9	10	10	10
C ₂	10	10	10	9	9	9	7
C ₃	9	10	8	10	10	9	9
C ₄	10	9	10	10	8	9	9
C ₅	9	10	10	10	10	10	9
C ₆	7	8	6	5	6	7	8
C ₇	8	7	9	10	7	6	7
C ₈	9	9	8	10	10	9	9
C ₉	9	8	8	9	9	9	8
C ₁₀	8	9	10	7	8	9	9
C ₁₁	9	10	8	9	7	8	9

Se verificó la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y la prueba de Friedman con el estadígrafo Chi cuadrado (X^2) (Legendre, 2005).

Este estadígrafo debe resultar entre 0 y 1, donde un resultado cercano a 1 refuta la hipótesis nula de Kendall acerca de la independencia de la prueba de la hipótesis de Friedman.

Los valores del Coeficiente de Concordancia (W) deben oscilar entre 0 y 1 ($0 < W < 1$), mientras mayor sea el valor de W, es decir, cuanto más se acerque a uno, mayor será la dependencia entre las respuestas.

Para la prueba de la hipótesis de Friedman se plantean la nula y la alternativa de la siguiente forma:

H0: No existe concordancia entre los criterios de los expertos.

H1: Existe una opinión común de preferencia entre los expertos.

Para esto se utilizó el paquete de análisis estadísticos SPSS (Mehta, y otros, 1996) que arrojó el resultado que se muestra en la Tabla 3.

Tabla 3. Resultado de las pruebas de Kendall y X^2 de Friedman.

N	7
Kendall's W ³	,490
Chi-Square ⁴	34,266
df	10
Asymp. Sig.	,000

El resultado muestra un $W = 0,490$ menor de la media entre 0 y 1. Se puede inferir que no existe dependencia entre los criterios de los expertos. Para probar la hipótesis se compara el resultado de X^2 de Friedman con el tabulado en la tabla de estándares para $\alpha=0,01$ y $c - 1 = 10$, si cumple que $X^2_{calculada} > X^2_{(\alpha, c-1)}$ entonces se rechaza H0 y se infiere que existe concordancia de criterios preferenciales entre los expertos, al considerar válida la hipótesis alternativa H1.

La comparación en correspondencia con las variables anteriores resulta $34,266 > 23,2093$, luego el resultado calculado es mayor que el tabulado, se acepta entonces H1 y se puede decir que existe concordancia en el trabajo de los expertos. (Estévez, y otros, 2005)

- No obstante los resultados y dado un análisis de varianza de las respuestas se identificó la necesidad de orientar algunos criterios sobre su verdadero significado y objetivo, dado que algunos

³ Coeficiente de Concordancia de Kendall

⁴ X^2 de Friedman

argumentos manifestaban desconocimiento del tema en cuestión.

Con motivo a esto se realizó una segunda vuelta con la encuesta, donde atendiendo a los criterios de los expertos expresados de manera abierta, se refinaron las preguntas y se presentaron solamente de manera cerrada en una escala cualitativa de 1-10, según el criterio de importancia marcado (Ver Anexo VI).

Esta vez se les entregó a los expertos el resultado del análisis inicial y de las justificaciones generales planteadas, aquellas que de una manera positiva o negativa, argumentaban sobre el objetivo del criterio.

En el análisis de esta ronda, se realizaron los resultados dados por los expertos de la misma manera. Se obtuvo el resultado que muestra la Tabla 4.

Tabla 4. Valor otorgado por los expertos a los criterios en la segunda vuelta.

C/E	E ₁	E ₂	E ₃	E ₄	E ₅	E ₆	E ₇
C ₁	10	9	10	9	10	10	10
C ₂	10	10	10	9	10	10	9
C ₃	9	10	9	10	10	9	9
C ₄	10	9	10	9	10	9	10
C ₅	9	10	10	10	10	10	9
C ₆	7	8	7	7	8	7	8
C ₇	8	8	9	8	7	7	7
C ₈	9	9	9	10	10	9	9
C ₉	9	8	8	9	9	9	8
C ₁₀	8	9	9	9	8	9	9
C ₁₁	9	10	9	10	10	9	9

Calculado por el SPSS, el W de Kendall y el X^2 de Friedman se obtiene el resultado de la Tabla 5.

Tabla 5. Resultado de las pruebas de Kendall y X^2 de Friedman en la segunda vuelta.

N	7
Kendall's W	,682
Chi-Square	47,741
df	10
Asymp. Sig.	,000

El resultado muestra un $W = 0,682$ mucho más cercano a 1 que la primera vez, se puede inferir entonces una mayor convergencia hacia un criterio común a la concordancia de los expertos sobre la propuesta.

Analizando la hipótesis de Friedman se obtiene un resultado similar a la vez anterior cuando, bajo los mismos criterios, se logra $47,741 > 23,2093$ (Estévez, y otros, 2005).

6. Hasta aquí solamente se ha logrado probar la efectividad del método de selección al conocer que los expertos están de acuerdo. Se debe entonces probar la efectividad de la propuesta a partir de los resultados definitivos obtenidos en la segunda vuelta.

Se propone calcular el Peso relativo de cada criterio $P = \sum C / E_T$, donde C es el criterio individual para el total de expertos E_T y a partir de este, el Índice de Aceptación de la propuesta con $IA = \sum P / E_T$. (Estévez, y otros, 2005).

Se predefine como rangos de éxito:

$IA > 80 \%$ Existe alta probabilidad de éxito.

$80 \% > IA > 50 \%$ Existe probabilidad media de éxito.

$50 \% > IA > 30 \%$ Probabilidad de éxito baja.

$30 \% > IA$ Fracaso seguro.

Los resultados del cálculo se muestran en la Tabla 6 (Mehta, y otros, 1996).

Tabla 6. Cálculo del Peso por la Estadística Descriptiva del SPSS.

	N	Minimum	Maximum	Mean	Std. Deviation
C1	7	9	10	9,71	,488
C2	7	9	10	9,71	,488
C3	7	9	10	9,43	,535
C4	7	9	10	9,57	,535

C5		7	9	10	9,71	,488
C6		7	7	8	7,43	,535
C7		7	7	9	7,71	,756
CO		7	9	10	9,29	,488
C9		7	8	9	8,57	,535
C10		7	8	9	8,71	,488
C11		7	9	10	9,43	,535
Valid (listwise)	N	7				

7. El IA sería entonces 9,026, que dividiendo entre el máximo del rango que es 10 y multiplicando por 100 para llevarlo a por ciento, quedaría en un 90,26 %.

Podemos concluir que la propuesta fue vista con un índice de aceptación exitoso.

3.4 Conclusiones

En este capítulo se evaluó la propuesta de Arquitectura de Información mediante encuestas realizadas a integrantes del Proyecto CICPC que arrojaron que la solución trae ventajas para el desarrollo del software con calidad.

También se evaluó mediante el método Delphi o de expertos, el cual permitió analizar los criterios de 7 expertos en Arquitectura de Información y con ellos se logró determinar el Índice de Aceptación exitoso que tiene la propuesta del presente trabajo, obteniéndose concordancia en el trabajo realizado y una alta probabilidad de éxito de ser implementado en otros proyectos con características similares al SIIPOL.

CONCLUSIONES

La realización de este trabajo permitió cumplir los objetivos para los cuales se desarrolló la investigación. Se adquirieron conocimientos teóricos sobre la Arquitectura de Información en general, que fueron aplicados a la realidad del proyecto CICPC con el objetivo de obtener un producto altamente usable y funcional para los usuarios.

Se definió y describió los pasos necesarios para aplicar la Arquitectura de Información en un Software de Gestión de Información, así como actividades a realizar y responsabilidades para lograr la construcción de un producto de alta calidad.

Como resultado se obtuvo una propuesta de Arquitectura de Información para el Software de Gestión SIIPOL del Proyecto CICPC, que amplió su radio de acción con la creación de la Librería de Componentes y las Plantillas para las Interfaces de Usuario, tareas atribuidas por lo general, al Diseño Gráfico y en las que la Arquitectura puede desempeñarse sin entorpecer el trabajo del diseñador.

La propuesta de solución introdujo la inserción de la Arquitectura de Información al Proceso de Desarrollo de Software siguiendo la metodología RUP.

Mediante el Método de Validación de Expertos se certificó la propuesta, arrojando como resultado que posee alta novedad científica y altas posibilidades de adaptación a otros productos con características similares al que se le aplicó en el proyecto.

RECOMENDACIONES

Gracias a las ventajas que ofrece la incorporación de la Arquitectura de Información dentro del Proceso de Desarrollo del Software para aumentar la calidad de este y realizar un producto centrado en el usuario, se sugiere el seguimiento de la solución propuesta. Por tal razón se hacen las siguientes recomendaciones:

- Aplicar la Propuesta de Arquitectura de Información a productos software con características similares a las del proyecto donde se aplicó.
- Introducir los elementos teórico-prácticos de la Arquitectura de Información como proceso dentro de RUP para todos los proyectos productivos en la UCI.
- Realizar talleres entre los arquitectos de información de sistemas complejos a fin de transmitir sus experiencias y adaptaciones en cuanto a Arquitectura de Información.
- Proponer una plantilla guía de Arquitectura de Información específicamente para Sistemas de Gestión.
- Documentar dentro del proceso de Arquitectura de Información otros artefactos tales como reportes y documentos oficiales que el sistema genera.

REFERENCIAS BIBLIOGRÁFICAS

Ambler, S. 2004. *The Object Primer, Third Edition.* Douglas Baldwin and Greg W. Scragg. s.l. : Cambridge University Press, 2004.

Constantine, L. L., and Lockwood, L. A. D. 1999. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design.* New York : ACM Press, 1999.

Donadello, Lic. Domingo F. 2004. METODOLOGIA INTEGRAL DE DESARROLLO DE SISTEMA. [Online] 2004. [Cited: Diciembre 15, 2007.] http://www.ub.edu.ar/catedras/ingenieria/ing_software/ubftecwwwdfd/mids_web/prototyp/estrdes.htm.

Estévez, María de Lourdes Bravo and Gallastegui, José Joaquín Arrieta. 2005. *EL MÉTODO DELPHI. SU IMPLEMENTACIÓN EN UNA ESTRATEGIA DIDÁCTICA PARA LA ENSEÑANZA DE LAS DEMOSTRACIONES GEOMÉTRICAS.* s.l. : Revista Iberoamericana de Educación, 2005. ISSN: 1681-5653.

Ferrer, Malisa Gutiérrez. 2005. Pienso, Luego existo. [Online] 2005. [Cited: febrero 25, 2008.] <http://www.malisa.cl/que-representamos-cuando-hacemos-diagramas-de-interaccion>.

Gómez Reyes, M. 2002. *Arquitectura de Información.* Ciudad de La Habana : s.n., 2002.

ISO 8402 UNE 66-001-92. 1992. ISO 8402 UNE 66-001-92. 1992.

JACOBSON, Ivar, BOOCH, Grady and RUMBAUGH, James. 2000. *El Proceso Unificado de Desarrollo de Software.* s.l. : Addison-Wesley, 2000.

James Garret, Jesse. 2002. *The Elements of User Experience.* New York : New Riders Publishing, 2002.

La Calle, Alberto. 2005. [Online] 2005. [Cited: Diciembre 15, 2007.] http://www.albertolacalle.com/hci_prototipos.htm.

Legendre, Pierre. 2005. *Species Associations: The Kendall Coefficient.* Quebec : American Statistical Association and the International Biometric Society, 2005. pp. 226–245. Vol. 10. 10.1198/108571105X46642.

León, Rodrigo Ronda. 2006. *Algunas técnicas para los arquitectos de información.* La Habana : s.n., 2006.

- Maner, Walter. 1997.** [Online] 1997. [Cited: Diciembre 10, 2007.] <http://www.sidar.org/recur/desdi/traduc/es/visitable/maner/Prototipado.htm..>
- Mehta, Cyrus R. and Patel, Nitin R. 1996.** *SPSS Exact Tests™ 7.0 for Windows*. [Documento] Chicago : s.n., 1996. 1-56827-108-5.
- Nielsen, J. 1993.** *Usability engineering*. Boston : AP Professiona, 1993.
- Page-Jones, M. 2000.** *Fundamentals of Object-Oriented Design in UML*. New Yor : Dorset-House, 2000.
- Pressman, Roger S. 2002.** *Ingeniería del Software. Un enfoque práctico 5ed*. Connecticut : s.n., 2002.
- Ronda León, Rodrigo. 2005.** No Solo Usabilidad Journal. [Online] Abril 25, 2005. [Cited: Diciembre 8, 2007.] http://www.nosolousabilidad.com/articulos/ai_cc_informacion.htm.
- UABC. 2007.** UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA. *SERVIDOR YAQUI*. [Online] 2007. [Cited: Diciembre 5, 2007.] <http://yaqui.mxl.uabc.mx/>.
- Visio. 2007.** Sitio Web oficial del producto Microsoft Visio. [Online] 2007. [Cited: enero 20, 2008.] <http://office.microsoft.com/es-es/visio/FX100487863082.aspx>.
- Wurman, Richard Saul. 1996.** *Information Architects* . 1996.

BIBLIOGRAFÍA

Consultora Especializada en Arquitectura de Información y Usabilidad. *Arquitectura de Información*. [En línea] luleDesign. <http://www.arquitecturadeinformacion.cl>.

León, Rodrigo Ronda. Web de Rodrigo Ronda León sobre diseño digital de la información. [En línea] [Citado el: 16 de Enero de 2008.] <http://elrodriweb.blog.com/>.

del V.Sosa, Mabel, Velázquez, Isabel y Digión, Leda. INTERACCIÓN HOMBRE-COMPUTADORA.HERRAMIENTAS CONCEPTUALES, METODOLÓGICAS Y TÉCNICAS. [Digital Disponible en: <http://www.ing.unp.edu.ar/wicc2007/trabajos/ISBD/083.pdf>] Santiago del Estero : s.n.

Ferré, Xavier y Moreno, Ana M. Integración de la IPO en el Proceso de Desarrollo de la Ingeniería del Software: Propuestas Existentes y Temas a Resolver. [Digital Disponible en: <http://is.ls.fi.upm.es/xavier/papers/interaccion2004.pdf>] Madrid : s.n.

Interacción Humano-Computador y diseño de interfaces. Dr. Girardi, Rosario y Ing. Deagostini, Adriana. 2005. Uruguay : s.n., 2005.

Camus, Juan Carlos. usando info. [En línea] <http://www.usando.info>.

Olguín, Miguel Argollo Jr.Carlos José Maria. Considerando el Usuario en el Proceso de Desarrollo de Aplicaciones con Interfaces Gráficas. [Digital Disponible en: <http://www.congreso-info.cu/UserFiles/File/Info/Info97/Ponencias/005.pdf>] Brasil : s.n.

Yusef Hassan Montero. No solo Usabilidad journal. [En línea] <http://www.nosolousabilidad.com>.

Ibáñez, Juan Carlos Pantoja. 2004. Arquitectura de la Información: Introducción al proceso de desarrollo en el Diseño de Interfaces de Usuario. Chile : s.n., 2004.

Instituto para la Arquitectura de Información. [Online] [Cited: Enero 24, 2008.] <http://iainstitute.org/en>.

IWS, Colectivo Dpto. 2006. Conferencia 1 Ingeniería de Software I. UCI. [Electrónico] La Habana : s.n., 2006.

Sanchez, María A. Mendoza. 2004. Metodologías De Desarrollo De Software. [Online] 2004. http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html.

GLOSARIO DE TÉRMINOS

Arquitectura de Información: Es la disciplina y arte encargada del estudio, análisis, organización, disposición y estructuración de la información en espacios de información, y de la selección y presentación de los datos en los sistemas de información interactivos y no interactivos.

Calidad: Conjunto de propiedades inherentes a una cosa que permiten apreciarla como igual, mejor o peor que las restantes de su especie.

Calidad de software: Grado con que el que un sistema, componente o proceso cumple con los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

Cliente: Son las personas para las cuales se elabora un producto determinado.

Etiqueta: Las etiquetas constituyen una forma de representación. Tal como se utilizan las palabras para representar pensamientos, las etiquetas permiten representar un conjunto de información en un sitio web, permiten la comunicación con los usuarios del web para mostrarles cómo está organizado el sitio y las posibilidades de navegación que presenta. Las etiquetas describen o designan los elementos que integran el sistema de navegación.

Fase: Período de tiempo entre dos hitos principales de un proceso de desarrollo.

Framework: Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

Herramienta: Software que se utiliza para automatizar las actividades definidas en el proceso.

Metáfora: consiste en el uso de una expresión con un significado o en un contexto diferente del habitual. En la Arquitectura de Información se usa principalmente este término en la utilización de imágenes o figuras que infieran el significado de las acciones, en vez de utilizar palabras.

Modelo de negocio: Describe los procesos de la empresa, identificando quiénes participan y las actividades que requieren automatización. Comprende la estructura y la dinámica de la organización en la cual se va a implantar el sistema, sus problemas actuales y evalúa el estado actual de la organización en la cual el sistema será explotado.

Proyecto: Es un esfuerzo temporal emprendido para crear un producto o servicio único.

Proyecto de software: Es un elemento organizativo a través del cual se gestiona el desarrollo del software. Obtiene un versión de un producto que contiene modelos, código fuente, documentación y un ejecutable.

Producto de software: Son los artefactos que se crean durante la vida del proyecto, como los modelos, código fuente, ejecutables, y documentación.

Proceso: Es un conjunto de actividades o eventos que se realizan o suceden con un determinado fin.

Proceso de desarrollo de software (PDS): Es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto para transformar los requisitos de usuario en un producto.

Requisitos del software: Son las funciones, servicios y restricciones operativas del sistema.

Rol: Es el papel que desempeña una persona o grupo de ellas en cualquier actividad. Una persona puede tener varios roles, así como un rol puede ser desempeñado por varias personas.

RUP: El Rational Unified Process (RUP) es un marco de proceso de IBM Rational para el uso de los casos, impulsado, centrada en la arquitectura, e iterativo e incremental de desarrollo de software. El RUP abraza la base del componente de desarrollo y hace uso del Lenguaje Unificado de Modelado. La parte más valiosa de las RUP consiste en la identificación de un conjunto de mejores prácticas fundamentales que son bien aceptados e indiscutible en la industria del software: desarrollar iterativamente, la gestión de requisitos, la utilización de componentes de arquitecturas, modelo visualmente, verificar la calidad y el control de cambios.

Sistema: Colección de unidades conectadas que se organiza para lograr un propósito. El sistema es el “modelo completo”.

Software: Se refiere a los programas y datos almacenados en un ordenador.

Software de Gestión: Programas informáticos para facilitar los procesos de gestión e intercambio de información.

Tecnología: Conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico.

Usuario: Persona que utiliza o trabaja con algún objeto o que es destinataria de algún servicio público o privado, empresarial o profesional.

Storyboards: Son ilustraciones mostradas en secuencia con el objetivo de servir de guía para entender una historia, previsualizar una animación o seguir la estructura de una película antes de realizarse o filmarse.

ANEXOS

Anexo I: Documento de Pautas de Arquitectura de Información del Proyecto CICPC

A continuación se muestra solamente las secciones interesadas al contenido relevante, obviando el control de la documentación, declaración de confidencialidad y demás formalidades del documento. Se aclara que las imágenes no poseen los colores originales por razones de impresión.

Esbozo de la estructura o taxonomía

1.0 Inicio Sesión

1.1 Mensaje de Bienvenida

1.2 Agenda de Trabajo

1.2.1 Notificaciones

1.2.2 Asignaciones

1.2.3 Aprobaciones

1.2.4 Remisiones

1.2.5 Borradores

1.2.6 Archivo

1.3 Menú Principal.

1.3.1 Gestión de Despacho

1.3.1.1 Registros

1.3.1.1.1 Correspondencia Entrada

1.3.1.1.2 Correspondencia Salida

1.3.1.1.3 Novedad

1.3.1.2 Consultas

1.3.1.2.1 Libro de Correspondencia

1.3.1.2.2 Libro de Novedades

1.3.2 Atención Telefónica

1.3.2.1 Registros

1.3.2.1.1 Notificación sobre Vehículos

1.3.2.2 Controles

1.3.2.2.1 Notificaciones prescritas

1.3.2.3 Consultas

1.3.2.3.1 Notificaciones sobre Vehículos

1.3.3 Investigación Penal

1.3.3.1 Registros

1.3.3.1.1 Denuncia

1.3.3.1.2 Notificación sobre Amenazas

1.3.3.1.3 Notificación sobre Objetos

1.3.3.2 Controles

1.3.3.2.1 Asignar Acta Procesal

1.3.3.2.2 Revisar/Aprobar Diligencia

1.3.3.3 Consultas

1.3.3.3.1 Notificaciones

1.3.3.3.2 Libro de Control de Investigación

1.3.4 Aprehensión

1.3.4.1 Registros

1.3.4.1.1 Detenido

1.3.4.2 Consultas

1.3.4.2.1 Carpeta de Detenido

1.3.4.2.2 Órdenes de Aprehensión

1.3.5 Investigación Criminalística

1.3.5.1 Solicitudes

- 1.3.5.1.1 Experticia Criminalística
- 1.3.5.1.2 Inspección de Robo a Banco
- 1.3.5.1.3 Identificación de Individuos

1.3.5.2 Registros de Informes

- 1.3.5.2.1 Avalúo
- 1.3.5.2.2 Balística
- 1.3.5.2.3 Contable
- 1.3.5.2.4 Inspección de Robo a Banco
- 1.3.5.2.5 Inspección de Inmueble
- 1.3.5.2.6 Inspección Técnica
- 1.3.5.2.7 Identificación de Individuos
- 1.3.5.2.8 Kit de muestras
- 1.3.5.2.9 Levantamiento Planimétrico
- 1.3.5.2.10 Microscopía Electrónica
- 1.3.5.2.11 Pericial Criminalístico
- 1.3.5.2.12 Reconstrucción de Hechos
- 1.3.5.2.13 Retrato Hablado
- 1.3.5.2.14 Siniestro
- 1.3.5.2.15 Trayectoria Balística
- 1.3.5.2.16 Trayectoria Intraorgánica
- 1.3.5.2.17 Vehículo

1.3.5.3 Consultas

- 1.3.5.3.1 Informes de Robo a Banco
- 1.3.5.3.2 Libro de Control de Vehículos

1.3.6 Investigación en Ciencias Forenses

1.3.6.1 Solicitudes

1.3.6.1.1 Experticia Forense

1.3.7 Análisis de Información

1.3.7.1 Registros

1.3.7.1.1 Funcionario

1.3.7.2 Consultas

1.3.7.2.1 Persona

1.3.7.2.2 Persona en SAIME

1.3.7.2.3 Objeto

1.3.7.2.4 Vehículo

1.3.7.2.5 Vehículo en INTTT

1.3.7.2.6 Propietario de Vehículo en INTTT

1.3.7.2.7 Lugar de Vehículo

1.3.7.2.8 Funcionario

1.3.7.3 Validaciones

1.3.7.3.1 Serial de Carrocería

1.3.8 Estadísticas

1.3.8.1 Delitos

1.3.8.2 Actas Procesales

1.3.9 Dotación de Equipos Policiales

1.3.9.1 Solicitudes

1.3.9.1.1 Dotación de Equipos Policiales

1.3.9.2 Registros

1.3.9.2.1 Asignación de Dotación

1.3.9.2.2 Devolución de Dotación

1.3.9.2.3 Equipos Policiales

1.3.9.3 Consultas

1.3.9.3.1 Dotación de Equipos Policiales

1.3.9.3.2 Equipos Policiales

1.3.10 Evidencia

1.3.10.1 Registros

1.3.10.1.1 Cambio de Cadena de Custodia

1.3.10.2 Consultas

1.3.10.2.1 Evidencia

1.3.11 Auditoría

1.3.11.1 Consultas

1.3.11.1.1 Trazas de Acta Procesal

1.3.11.1.2 Trazas de Entidades

1.3.11.1.3 Trazas de Sistema

1.3.11.1.4 Trazas de Usuario

1.3.12 Administración

1.3.12.1 Registros

1.3.12.1.1 Perfil

1.3.12.1.2 Usuario

1.3.12.2 Consultas

1.3.12.2.1 Perfiles

1.3.12.2.2 Usuarios

1.3.12.3 Configuraciones

1.3.12.3.1 Despacho

1.3.12.3.2 Direcciones IP y MAC

1.3.12.3.3 Sistema

1.3.12.4 Configuraciones Auxiliares

1.3.12.4.1 Calendario Anual

1.3.12.4.2 Preguntas de Entrevista

1.3.12.4.3 Ubicación Geográfica

Descripción de los elementos de la estructura

1.0 Iniciar Sesión

El usuario debe introducir su usuario y contraseña para acceder al Sistema SIIPOL.

1.1 Mensaje de Bienvenida

Se incluyen especificaciones de utilidad para la usabilidad del sistema desde el punto de vista usuario.

1.2 Agenda de Trabajo

Muestra de manera centralizada y fácil de acceder:

*1.2.1 Las **notificaciones** del sistema,*

*1.2.2 Las **asignaciones** de tareas a realizar,*

*1.2.3 Las comunicaciones en espera de ser **revisadas***

*1.2.4 Las comunicaciones en espera de ser **aprobadas**,*

*1.2.5 Las **comunicaciones listas para ser entregadas** a un funcionario en rol de mensajero,*

*1.2.6 Los **informes temporales** en espera de ser finalizados*

*1.2.7 Las **comunicaciones** atendidas que han sido **archivadas** para el usuario.*

1.3 Menú Principal

Se pueden acceder a los módulos, submódulos y casos de uso que desee el usuario o que le competan.

Definición de la estructura

Criterios de organización y etiquetado en el Software SIIPOL
<p>Las interfaces deben estar modeladas en Microsoft Visio y con interfaces elementales de la Forma UML. Esto es para las interfaces de las especificaciones de los CU.</p> <p>En el caso de la aplicación se elaborará en JSF.</p>
<p>Debe mostrar los campos con ejemplos de posibles valores, siempre y cuando estos ejemplos sean totalmente verídicos para el cliente o generales. Por ejemplo: Dependencia Carabobo o Dependencia 1.</p>
<p>Los términos deben presentarse capitalizados (primera letra en mayúsculas)</p>
<p>Las etiquetas deben aparecer con la tipografía Arial, en puntaje 9, en el caso de las IU-Visio.</p> <p>En el caso de las IU de la aplicación debe aparecer en Arial con el siguiente puntaje:</p> <p>Menú 12 px , con módulos en negrita</p> <p>Labels: 12 px</p> <p>Encabezados: 12 px bold</p> <p>Texto Pantalla Inicio: 12 px interlineado 18</p>
<p>Las etiquetas se deben mostrar encima de los controles para todos las IU Consultar.</p> <p>En caso de que la etiqueta vaya al lado de un componente, que corresponda a información aislada dentro de la IU, entonces se colocará a la izquierda del componente.</p>
<p>No se debe mostrar en ningún campo algún resultado cuando la página o sección se muestre nueva o se hayan limpiado los datos.</p>
<p>Los campos que se comporten de manera errónea, se deben mostrar en rosado los controles de los mismos.</p>
<p>El etiquetado debe seguir los estándares del usuario en cuanto a su argot profesional o el de su organización, esto es, que debe utilizar palabras sencillas.</p>
<p>Los ejemplos de entrada de datos se deben usar cuando sea estrictamente necesario, estos se deben ubicar debajo de los campos, en cursiva a un puntaje 8.</p>

<p>Las pestañas rojas, que señalan el uso de más de una entidad en un mismo CU, deben tener el nombre de la entidad que se muestra en el momento y no el nombre de la acción, siempre y cuando la acción no sea imprescindible para comprender bien lo que se muestra. Ejemplo correcto: Persona. Ejemplo Incorrecto: Ver persona</p>
<p>Solo existirán abreviaturas reconocidas internacionalmente, como en el caso de la palabra Número que se mostrará siempre como No. En el caso de los cargos de los funcionarios también se mostrará en abreviaturas, ejemplo: Como a la hora de mostrar el nombre y apellido del funcionario, se mostrara primeramente el cargo o rango. Por ejemplo: Com. Juan Pérez</p>
<p>No se debe mostrar la hora de entrada de uno o varios elementos en las IU Incluir, sólo es posible mostrar la fecha. Esto porque puede que no se incluya exactamente en la hora estática que se muestra.</p>
<p>En el caso de los datos que son nombres compuesto y el valor mostrado o a mostrar refleje el tipo de dato, no se pondrá el nombre del mismo, sino a lo que hace referencia.</p> <p>Por ejemplo: No se debe colocar Fecha de Creación al lado de un componente de Fecha, es sabido que es una fecha, luego será admitido como correcto colocar solamente Creación y al lado el componente o el dato.</p>
<p>En las UI para ingresar el Nombre y Apellidos de una Persona debe utilizarse siempre el siguiente formato para la entrada.</p> <ul style="list-style-type: none">• Primer Nombre• Segundo Nombre• Primer Apellido• Segundo Apellido <p>(A excepción de que dicho nombre y apellidos no vaya a parar a la entidad Persona, o sea, que sea un campo que se recoge como dato adicional a otra entidad).</p> <p>El formato para la salida debe ser siempre todo junto, o sea el Nombre y Apellidos en una misma línea de información.</p>
<p>El campo Funcionario (donde quiera que este se utilice ya sea como Entidad Funcionario o como parte de una Entidad Superior XXX, como por ejemplo: Funcionario Asignado, Funcionario Emisor, etc.) debe llevar como datos: Nombre y Apellidos, Cargo y Rango.</p>

En las IU para ingresar la Cédula de una Persona debe utilizarse siempre el siguiente formato para la entrada, la cual se ingresará bajo la etiqueta Cédula.

- Un select con la letra V o E para la letra de la cédula
- Un edit para el No. de Cédula

Ejemplo:

Cédula

El formato para la salida debe ser siempre todo junto, o sea el Letra y Número de Cédula en una misma línea de información. Ej: V-67896543, que se mostrará con la etiqueta “No. de Identificación”

Ejemplo:

No. de Identificación V-

Las tablas que muestren información no estén nunca vacías, o sea, si no hay registros, se debe mostrar una fila combinada con una información que diga: “No existe información en el sistema”. (Esto no aplica para los CU de Consultar, que solo muestra la tabla cuando existieron coincidencias)

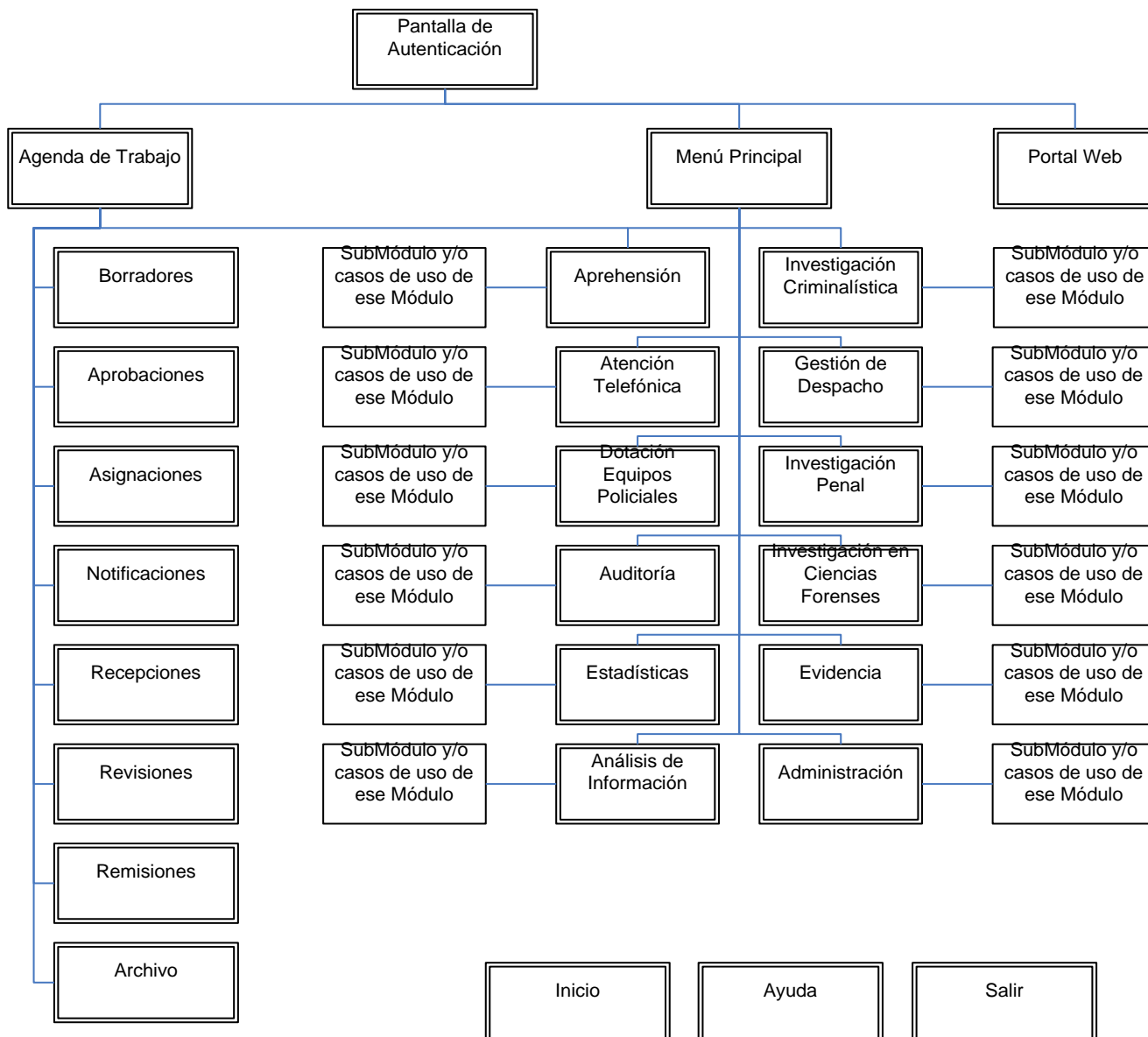
The screenshot shows a web application interface with two tabs: 'Listado' and 'Prueba'. Below the tabs is a section titled 'Listado de Personas'. It contains a table with the following columns: Cédula, Nombres y Apellidos, Sexo, Pasaporte, Natural de, Estado, Validado SAIME, Ver, Modificar, and Disociar. The table is currently empty and displays the message "No existe información en el sistema". At the bottom right of the interface is a button labeled 'Asociar'.

Para los casos de uso especificados como Gestionar *Entidad* por el patrón CRUD, las funcionalidades correspondientes se modelarán en una IU con los nombres y botones de aceptación siguientes:

- Sección Incluir: Incluir *Entidad*, botón de aceptación *Incluir* y de cancelación *Cancelar*.
- Sección Ver: *Entidad*, botón de finalización *Cerrar*.
- Sección Modificar: Modificar *Entidad*, botón de aceptación *Actualizar* y de cancelación *Cancelar*.
- Sección Eliminar: Eliminar *Entidad*, mensaje de confirmación de eliminación obligatorio, botón de aceptación *Aceptar* y de cancelación *Cancelar*.

Para una IU de consultar deberán aparecer siempre tres opciones, Consultar, Nueva Búsqueda y Cerrar.

Mapa de Navegación General

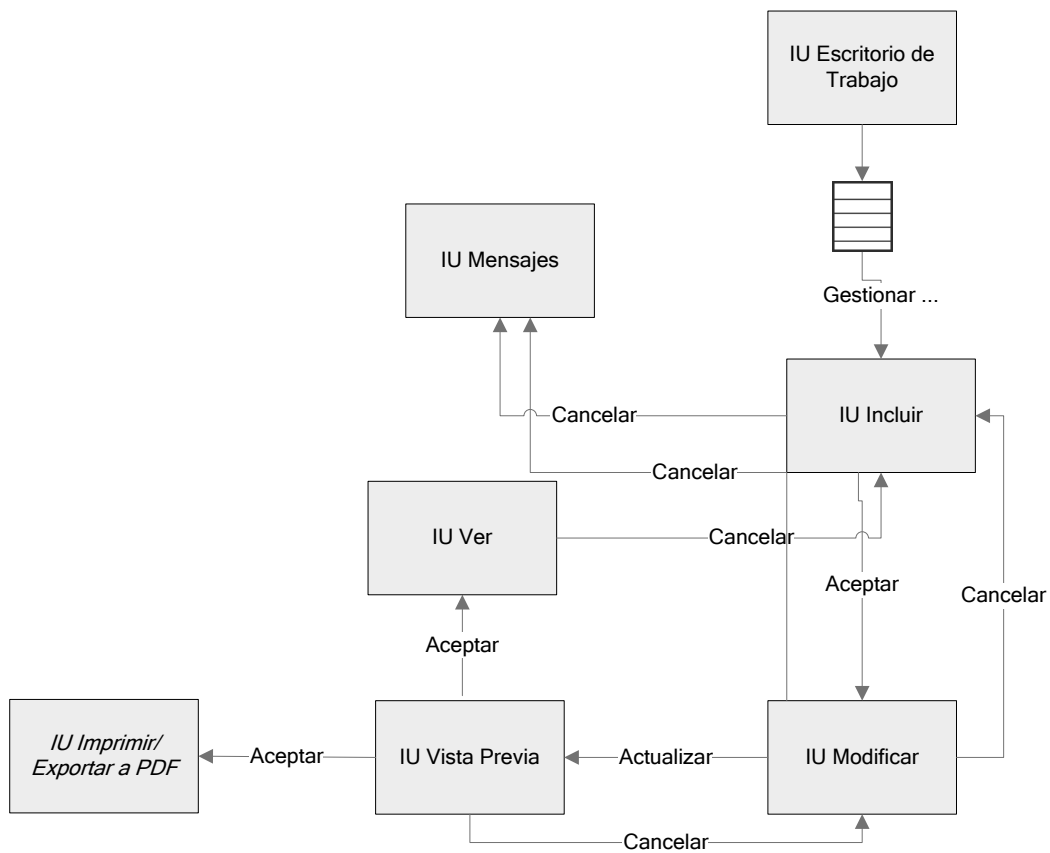


Elementos y Descripción del Sistema de Navegación

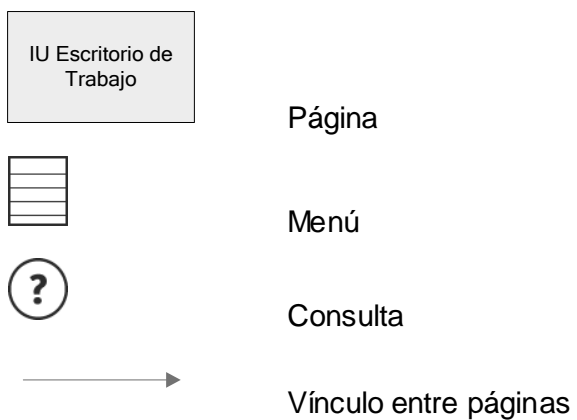
Después que el usuario se autentique en el sistema, puede acceder directamente al Escritorio de Trabajo, donde se le brinda la posibilidad de Ver su Agenda de Trabajo donde puede revisar los casos que tiene a su cargo en los diferentes estados de la investigación, distribuidos en las secciones Borradores, Aprobaciones, Asignaciones, Notificaciones, Recepciones, Revisiones, Remisiones y Archivo. Además puede acceder a estos elementos mediante el Menú Principal, donde se encuentran todas las funcionalidades de la aplicación distribuidas en los diferentes Módulos y SubMódulos. Es importante destacar que en Escritorio de Trabajo de un usuario solo se mostrarán las funcionalidades a las que pueda acceder este según sus permisos y/cargos en la institución. También se puede acceder al Portal de Web de la Institución que es otro servicio de valor agregado que brinda el sistema y en cualquier momento se le da la posibilidad de ir nuevamente a la Pantalla de Inicio, Consultar la Ayuda de sistema y Cerrar la aplicación.

Diseño de Mapas de Navegación para Especificaciones de Casos de Usos

- La lógica de navegación del mapa es la del flujo básico y el alterno del CU en cuestión.
- Todos los CU cuya especificación tenga en la categoría Nivel el valor Usuario, salen del menú principal, esto se representa con el icono de menú.
- Cuando se llame a otro CU, el nombre del mismo se escribirá en la pantalla en Cursiva.
- Solo se hará un mapa de navegación, pero debe ser legible.
- Se hará una sola IU de mensajes, por lo que todos los mensajes, sean de información o de error, se contemplarán dentro de la misma.




Legenda de un Mapa de Navegación


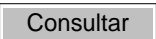




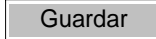


Diseño de la Estructura de las Páginas Tipo

En esta sección están definidas unas pantallas básicas que son la guía para el modelado de interfaces. Se encuentran en el Expediente del Proyecto de CICPC.

Descripción de componentes

<i>Barra de progreso</i>		
Deben mostrarse el progreso del evento usando colores visibles, y con un 50% de avance.		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos
	<i>Indica la progresión de una acción que tendrá respuesta del sistema, se usa por lo general, a la hora de seleccionar una imagen.</i>	
Ubicación: Repositorio/Modelo de Sistema/Especificación/ Estándar/Pautas de Interfaces/Librería de Componentes Rojos.		


Botones (Buttons)		
Deben nombrarse de manera que resulte fácil de entender las acciones para lo que están diseñados y con una descripción clara de sus funciones.		
Los botones se emplearán para las acciones a realizar en las IU y para las transacciones en el sistema.		
Todos los botones se agruparán en la región derecha de la pantalla.		
Se predeterminará un botón por IU, de acuerdo con la usabilidad que tengan en la misma. Ej: En un IU <i>Incluir...</i> puede ser   		
Los botones se ordenarán de acuerdo con la prioridad de las acciones a realizar en la IU		
Los hipervínculos siempre irán a la izquierda de los botones.		
En caso de algún CU que especifique la acción, dirá la acción y la síntesis de lo que plantea el CU. Esto se hará si es necesario verdaderamente. Ej: 		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos
	Concluir una acción en el sistema	 

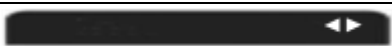
Actualizar	Modificar datos en el sistema	Modificar	Si
Cancelar	Invaldar una acción en el sistema	Limpiar	No
Consultar	Acción para consultar el sistema	Buscar	Búsqueda
		Consulta	Comenzar Búsqueda
Eliminar	Descartar uno o varios elementos de una entidad del sistema	Borrar	Descartar
		Eliminar del Listado	
Imprimir/Exportar	Imprimir y/o Exportar a PDF el listado o formato mostrado por el sistema que se desea imprimir	Imprimir	Exportar
Incluir	Agregar datos a una lista o entidad similar en el sistema	Agregar	Añadir
Guardar	Guardar temporalmente en el sistema	Aceptar	Añadir
Vista Previa	Visualizar los datos que se incluyeron o se desean imprimir	Ver	Mostrar
Limpiar	Borrar elementos insertados en los componentes de los formularios	Cancelar	Borrar
		Cancelar Listado	
Visualizar	Ver uno o todos los elementos nuevos o modificados en el sistema	Ver	
Modificar	Modificar los elementos incluidos en el sistema, se usa por lo general en la IU Incluir	Actualizar	Cambiar
Asociar	Adjuntar un elemento al CU	Adjuntar	

<input type="button" value="Examinar"/>	Buscar una imagen para incluir en el sistema	<input type="button" value="Buscar"/>
<input type="button" value="Ampliar"/>	Mostrar una imagen en su tamaño original	<input type="button" value="Expandir"/>
<input type="button" value="Validar"/>	Aprobar una acción o elemento	<input type="button" value="Aceptar"/> <input type="button" value="Si"/>
<input type="button" value="Disociar"/>	Eliminar la relación de asociación	<input type="button" value="Eliminar"/>
<input type="button" value="Registrar"/>	Reconocer un/unos elementos en el sistema	<input type="button" value="Reconocer"/>
<input type="button" value="Responder"/>	Enviar una respuesta a una solicitud en el sistema	<input type="button" value="Contestar"/>
<input type="button" value="Crear"/>	Crear un elemento	<input type="button" value="Añadir"/>
<input type="button" value="Cerrar"/>	Para cancelar la acción del consultar y cuando se desea terminar en la IU en la que se esté.	<input type="button" value="Cancelar"/>
<input type="button" value="Salir"/>	Retirarse definitivamente del sistema	<input type="button" value="Cancelar"/>

Otros Botones que se pueden utilizar:

<input type="button" value="Ver Informe"/>	<input type="button" value="Asociar Kit de Muestra"/>	<input type="button" value="Crear Informe"/>	<input type="button" value="Cambiar Custodia"/>
<input type="button" value="Validar Identidad"/>	<input type="button" value="Asignar Direcciones"/>	<input type="button" value="Registrar Propietario"/>	<input type="button" value="Exportar a XML"/>
<input type="button" value="Validar Funcionario"/>	<input type="button" value="Validar Funcionario"/>	<input type="button" value="Validar Funcionario"/>	<input type="button" value="Validar Funcionario"/>
<input type="button" value="Ver Relaciones"/>	<input type="button" value="Registrar Recuperación"/>	<input type="button" value="Ratificar Orden"/>	<input type="button" value="Comparar Reporte"/>
<input type="button" value="Generar Gráfico"/>	<input type="button" value="Crear Informe"/>	<input type="button" value="Solicitar Experticia"/>	<input type="button" value="Buscar Coincidencias"/>
			<input type="button" value="Efectuar Entrega"/>

Botones de número, horizontal, vertical		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos
	Para ordenar verticalmente dentro de una misma columna	

	en una tabla.	
	Para navegar horizontalmente dentro de una lista de resultados	

Botones de Opción (Radio Buttons)		
No se deben colocar textos extensos		
Se usa cuando la información es excluyente.		
Si el conjunto de opciones tiene un número superior a cuatro, se debe alinear verticalmente. En caso contrario se deberá hacerse horizontalmente .		
El orden de las opciones debe ser por la prioridad de la información que se muestra y/o necesite.		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos
<input checked="" type="radio"/> Informe <input type="radio"/> Solicitud	Conjunto de opciones en donde se escoge solo una.	<div style="border: 1px solid black; padding: 5px; display: inline-block;"> <input checked="" type="radio"/> Informe <input type="radio"/> Solicitud </div> <div style="border: 1px solid black; padding: 5px; display: inline-block;"> <input checked="" type="radio"/> Informe <input type="radio"/> Solicitud </div> <input checked="" type="radio"/> Informe <input type="radio"/> Solicitud

Casillas de Verificación (Check box)		
No se deben colocar textos extensos		
Si el conjunto de opciones tiene un número superior a cuatro, se debe alinear verticalmente. En caso contrario se deberá hacer horizontalmente.		
El orden de las opciones debe ser por la prioridad de la información que se muestra y/o necesite.		
Por lo general se puede utilizar en una tabla de resultado cuando se desea realizar alguna acción sobre el dato que se muestra. Casi siempre se puede usar cuando se puede eliminar varios elementos.		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos
<input checked="" type="checkbox"/> Falta la pieza <input checked="" type="checkbox"/> Falta el complemento <input type="checkbox"/> Sobran componentes	Conjunto de opciones en	<input checked="" type="checkbox"/> Falta la pieza <input checked="" type="checkbox"/> Falta el complemento <input type="checkbox"/> Sobran componentes

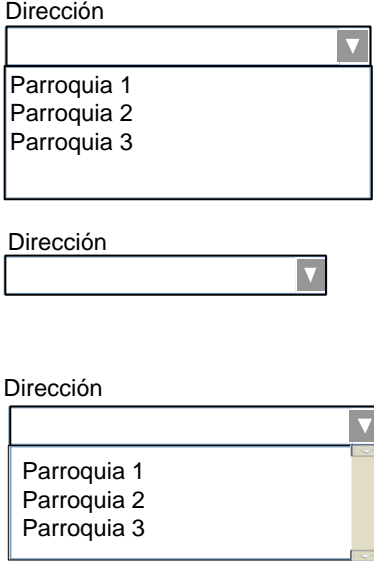
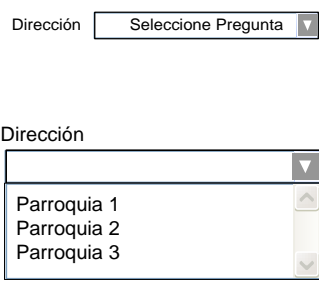
	<p>donde se escogen varias.</p>	<div data-bbox="1114 286 1302 360" style="border: 1px solid black; padding: 2px;"> <input checked="" type="checkbox"/> Falta la pieza <input checked="" type="checkbox"/> Falta el complemento <input type="checkbox"/> Sobran componentes </div> <div data-bbox="1114 389 1302 412" style="border: 1px solid black; padding: 2px; margin-top: 5px;"> <input type="checkbox"/> Sobran componentes <input type="checkbox"/> Falta la pieza <input type="checkbox"/> Falta el complemento </div>
--	---------------------------------	--

Cuadro Combinado

Los nombres de los campos se colocarán de acuerdo con las pautas establecidas en las pautas generales.

Mientras el texto sea visible en toda su extensión no se debe mostrar las barras de desplazamiento vertical.

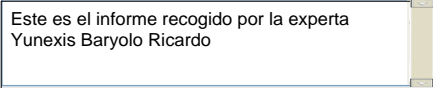
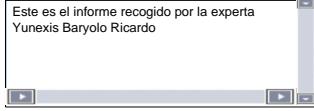
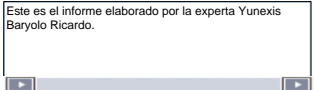
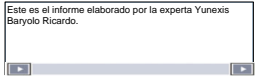
Si la interfaz se encuentra muy cargada no deben estar desplegados los cuadros combinados pues esto implica mucho espacio y no es objetivo recargar mucho la interfaz pues esto puede entorpecer el entendimiento del cliente.

Ejemplos Correctos	Descripción	Ejemplos Incorrectos
	<p>Conjunto de opciones en donde se escoge una.</p>	

Cuadro de Lista (List Box)

Mientras el texto sea visible en toda su extensión no se debe mostrar las barras de desplazamiento vertical.

A este componente se le dará tanto espacio horizontal como lo permita la pantalla, aprovechando este para no explotar, si no es necesario, el espacio vertical.

No se deben usar barras de desplazamiento horizontal solo las de desplazamiento vertical.		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos
<p>Información</p> 	<p>Control para incluir extensos volúmenes de texto.</p>	<p>Información</p>  <p>Información</p>  <p>Información</p> 

Cuadro de Lista (List Box)		
No se colocarán textos extensos.		
Si el conjunto de opciones tiene un número superior a cuatro, se debe alinear verticalmente. En caso contrario se deberá hacer horizontalmente.		
El orden de las opciones debe ser por la jerarquía de los conjuntos de elementos descrita en el caso de uso.		
Cuando sean varios elementos se deben crear columnas		
Los campos que sea necesario indicar que son obligatorios se deben señalar con un asterisco negro al lado derecho del cuadro de lista.		
Se debe escribir en negrita la pestaña activa		
No se deben dejar pestañas sin etiquetar		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos

<p>Datos Persona</p> <p>Primer Nombre <input type="text"/> *</p> <p>Segundo Nombre <input type="text"/></p> <p>Primer Apellido <input type="text"/> *</p> <p>Segundo Apellido <input type="text"/></p>	<p>Control para incluir extensos volúmenes de texto.</p>	<p>Datos Persona</p> <p>Primer Nombre <input type="text"/> *</p> <p>Segundo Nombre <input type="text"/></p> <p>Primer Apellido <input type="text"/> *</p> <p>Segundo Apellido <input type="text"/></p> <hr/> <p>Datos Persona</p> <p>Primer Nombre <input type="text"/> * Segundo Nombre <input type="text"/></p> <p>Primer Apellido <input type="text"/> * Segundo Apellido <input type="text"/></p> <hr/> <p>Datos Persona</p> <p>Primer Nombre <input type="text"/> * Segundo Nombre <input type="text"/></p> <p>Primer Apellido <input type="text"/> * Segundo Apellido <input type="text"/></p>
---	--	---

<p>Hipervínculo (Link)</p>		
<p>Los hipervínculos se utilizarán para enlazar un elemento a otra IU o cuando se llame dentro de un listado a otra acción o pantalla</p>		
<p>Los hipervínculos se conforman con la acción y la síntesis de lo que plantea el CU</p>		
<p>En caso de que estén presentes en una Lista de Resultados se alinearán centrados</p>		
<p>“Eliminar” no será un hipervínculo, si se pueden eliminar varios elementos al mismo tiempo.</p>		
<p>Para hacer un vínculo se deben usar solo textos con el enlace hacia donde vaya dirigido.</p>		
<p>Ejemplos Correctos</p>	<p>Descripción</p>	<p>Ejemplos Incorrectos</p>
<p>Búsqueda Avanzada</p>	<p>Aumentar los criterios de búsqueda</p>	<p>Búsqueda avanzada</p>
<p>Eliminar</p>	<p>Descartar solo un elemento del listado</p>	
<p>Modificar</p>	<p>Visualizar una única entidad para cambiar datos. Se muestra en cada uno de los elementos del listado</p>	
<p>Ampliar Información</p>	<p>Mostrar información completa o más extensa</p>	<p>Ampliar Información</p>
<p>Configurar Encabezado</p>	<p>Editar el encabezado de los</p>	

	informes	
Búsqueda Estándar	Restringir la búsqueda a criterios elementales	Búsqueda Simple
Ampliar	Ampliar un elemento de un listado	
Incluir Reseña		
Disociar	Eliminar la asociación hecha a un elemento de un listado.	
Asociar	Asociar un elemento de un listado	

Listado de resultados

No se utilizarán las abreviaturas, en ninguna celda de la tabla de resultados, salvo aquellas que estén reconocidas por el sistema (No., CICPC, INTT, SIIPOL., Com., Inv.,)

Dentro de los listados, se pueden ejecutar acciones, las cuales se harán, mediante el uso de hipervínculos o Chex box

Los campos denominadores se escribirán en Negrita (Ctrl.+N) y se centrarán dentro de las celdas

La información se mostrará alineada hacia la izquierda, solo los hipervínculos que denoten acción (modificar, ver, asociar...) aparecerán centrados. Los hipervínculos que sean información resultante de la búsqueda, se alinearán igual que los demás, esto es, a la izquierda.

Ejemplos Correctos

No.	Fecha	Persona	Emitido por ▼

Recursos


Tipo	Matricula	Desabilitación
<input type="checkbox"/> Nombre Recurso 1	123456789	12:30
<input type="checkbox"/> Nombre Recurso 2	123456789	19:30
<input type="checkbox"/> Nombre Recurso 3	123456789	19:30
<input type="checkbox"/> Nombre Recurso 4	123456789	19:30
<input type="checkbox"/> Nombre Recurso 5	123456789	19:30
<input type="checkbox"/> Nombre Recurso 6	123456789	19:30
<input type="checkbox"/> Nombre Recurso 7	123456789	19:30
<input type="checkbox"/> Nombre Recurso 8	123456789	19:30
<input type="checkbox"/> Nombre Recurso 9	123456789	19:30
<input type="checkbox"/> Nombre Recurso 10	123456789	19:30
<input type="checkbox"/> Nombre Recurso 11	123456789	19:30

Total: 32 | < < 1 2 3 ... > > |

	Fecha	Emitido por	Modificar
<input checked="" type="checkbox"/>	Viernes, 12/02/08	Com. Jose Urbina	Modificar
<input type="checkbox"/>	Viernes, 12/02/08	Com. Jose Urbina	Modificar
<input type="checkbox"/>	Viernes, 12/02/08	Com. Jose Urbina	Modificar
<input checked="" type="checkbox"/>	Viernes, 12/02/08	Com. Jose Urbina	Modificar

Tabla de Datos Predeterminados		
Las etiquetas que contengan información predeterminada se mostrarán en una tabla.		
Las etiquetas se mostrarán en la sección izquierda, y las celdas en esta sección, serán celdas de encabezado. Se ordenarán hacia la derecha de la tabla.		
Nunca los campos se escribirán en Negrita (Ctrl.+N)		
La información correspondiente a las etiquetas, se mostrará en la sección derecha, las celdas en esta sección, serán celdas sin encabezado. Se ordenarán hacia la izquierda de la tabla.		
El tamaño de las tablas dependerá de la cantidad de información que deban mostrar, siempre debe tenerse en cuenta optimizar los espacios en las IU.		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos

<p>Datos del Informe</p> <table border="1"> <tr><td>Identificador</td><td>I-07-059-0001-0001</td></tr> <tr><td>No Solicitud</td><td>S07-057-0001</td></tr> <tr><td>No Acta procesal</td><td>K-057-050-0001</td></tr> <tr><td>Importancia</td><td>Alta</td></tr> <tr><td>Despacho Solicitante</td><td>Sub. El Llanito</td></tr> <tr><td>Destinatarios</td><td>Sub. El Llanito</td></tr> </table> <p>Informe</p> <table border="1"> <tr><td>Funcionario Emisor</td><td>Com. Juan José Urbina</td></tr> <tr><td>Fecha de Envío</td><td>Viernes, 12/10/07</td></tr> <tr><td>Serial</td><td>V133Y2493C45</td></tr> </table>	Identificador	I-07-059-0001-0001	No Solicitud	S07-057-0001	No Acta procesal	K-057-050-0001	Importancia	Alta	Despacho Solicitante	Sub. El Llanito	Destinatarios	Sub. El Llanito	Funcionario Emisor	Com. Juan José Urbina	Fecha de Envío	Viernes, 12/10/07	Serial	V133Y2493C45	<p>Muestra la información predeterminada que se necesita para realizar alguna acción en el sistema.</p>	<p>Informe</p> <table border="1"> <tr><td>Funcionario Emisor</td><td>Com. Juan José Urbina</td></tr> <tr><td>Fecha de Envío</td><td>Viernes, 12/10/07</td></tr> <tr><td>Serial</td><td>V133Y2493C45</td></tr> </table>	Funcionario Emisor	Com. Juan José Urbina	Fecha de Envío	Viernes, 12/10/07	Serial	V133Y2493C45
Identificador	I-07-059-0001-0001																									
No Solicitud	S07-057-0001																									
No Acta procesal	K-057-050-0001																									
Importancia	Alta																									
Despacho Solicitante	Sub. El Llanito																									
Destinatarios	Sub. El Llanito																									
Funcionario Emisor	Com. Juan José Urbina																									
Fecha de Envío	Viernes, 12/10/07																									
Serial	V133Y2493C45																									
Funcionario Emisor	Com. Juan José Urbina																									
Fecha de Envío	Viernes, 12/10/07																									
Serial	V133Y2493C45																									

<p>Pantalla base de Interfaz</p>	
<p>Ejemplos Correctos</p> 	<p>Ejemplos Incorrectos</p> 

Menú Lateral Izquierdo
El menú lateral izquierdo se ordenará por orden jerárquico, primero se ponen los módulos ordenados de acuerdo con la prioridad del sistema, excepto en el módulo en el que se esté trabajando, que el mismo va en la parte superior de los módulos. Esto es específicamente para las interfaces del visio. Para la aplicación se mantiene dinámicamente.
Al módulo Investigación Forense, se le llamará Investigación en Ciencias Forenses .
En el caso de que los nombres de los módulos no quepan en las líneas, los mismos se pondrán hasta donde quepan las palabras completas y luego se pondrán tres puntos suspensivos, tal es el caso de Investigación en Ciencias Forenses, se pondría así: Investigación en Ciencias...
Se eliminarán los nombres de los submódulos, como está determinado por los analistas, en cambio de establecen nombres genéricos, dados por el tipo documental.
Se eliminarán los nombres tan complejos y largos de los casos de uso.
Los casos de uso se dividirán de acuerdo con el tipo de documento al que respondan (actas, solicitudes, estadísticas, listados, experticias, etc.) y se pondrán solo el sujeto del caso de uso.. Por ejemplo: Consultas, y debajo Armas, Vehículos, Personas.
Los módulos se ponen en Negritas (Ctrl+N)

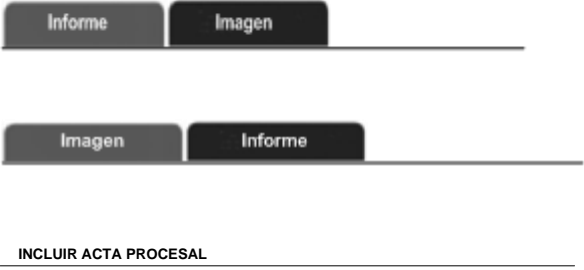

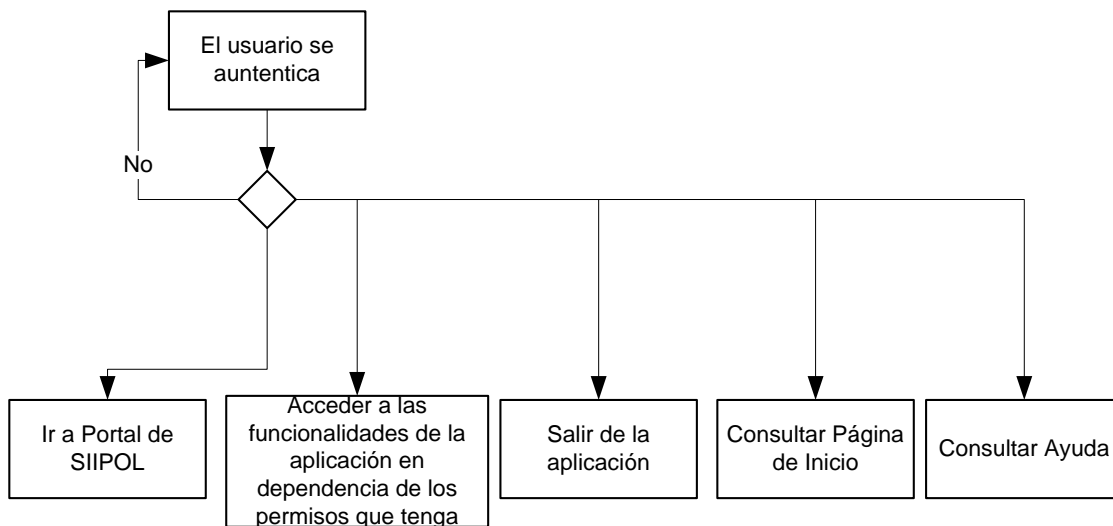
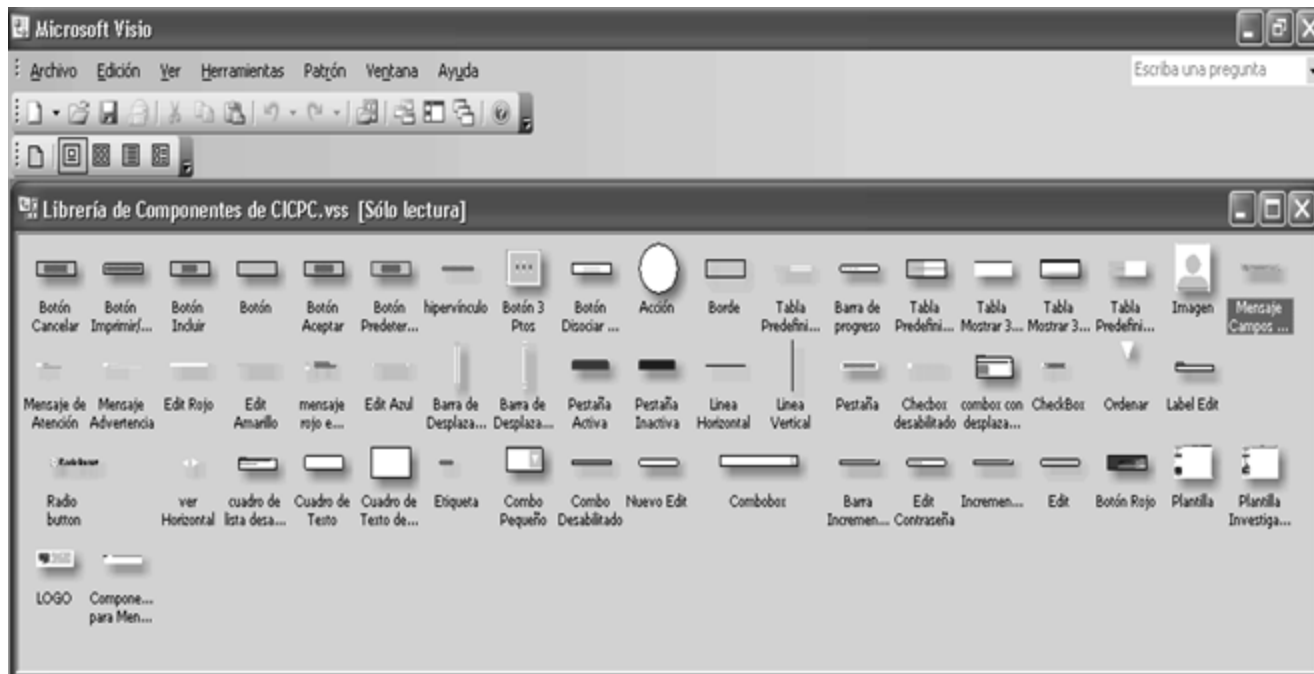
Pestaña de Entidades		
Se usa una raya en la que se pone encima la acción y el nombre completo del caso de uso y en negrita y mayúscula todo, ejemplo: INCLUIR INFORME DE EXPERTICIA TOXICOLÓGICA .		
Ejemplos Correctos	Descripción	Ejemplos Incorrectos
		

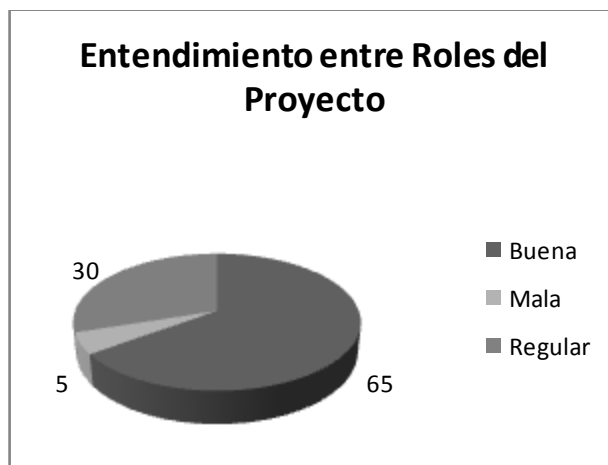
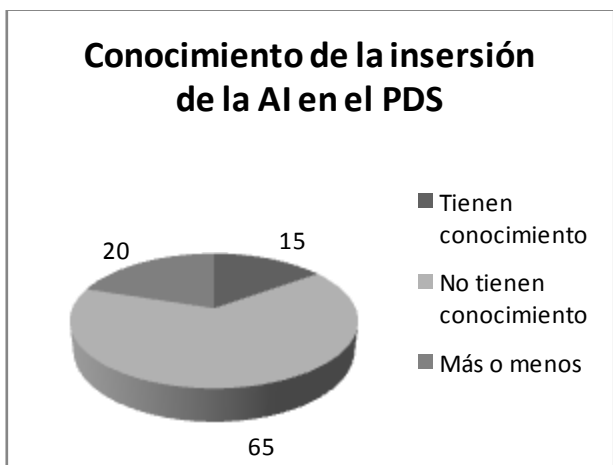
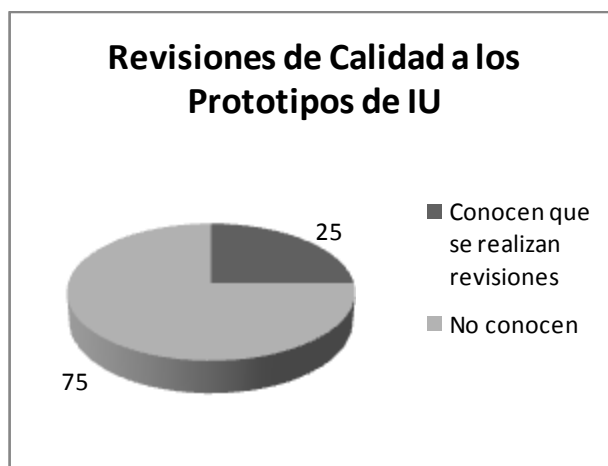
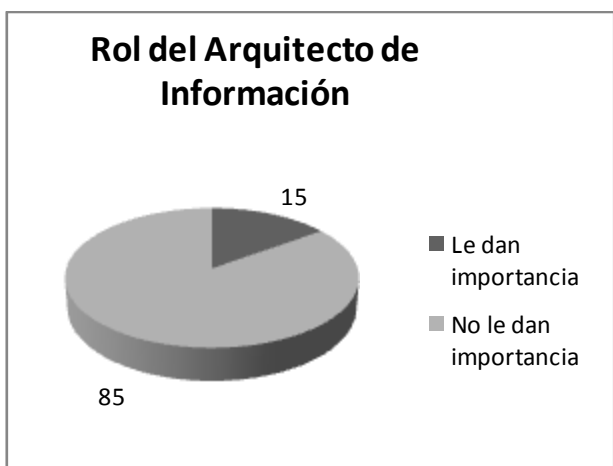
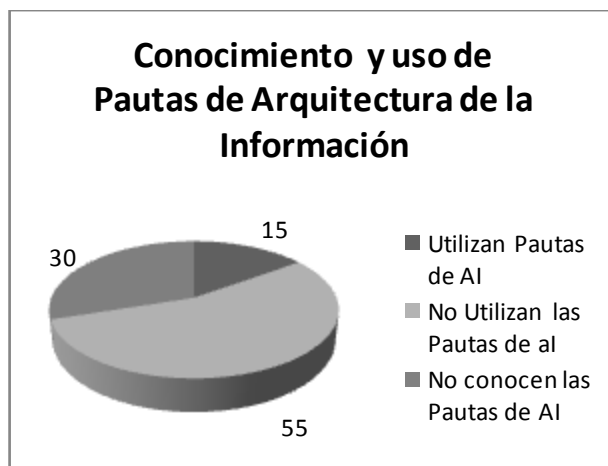
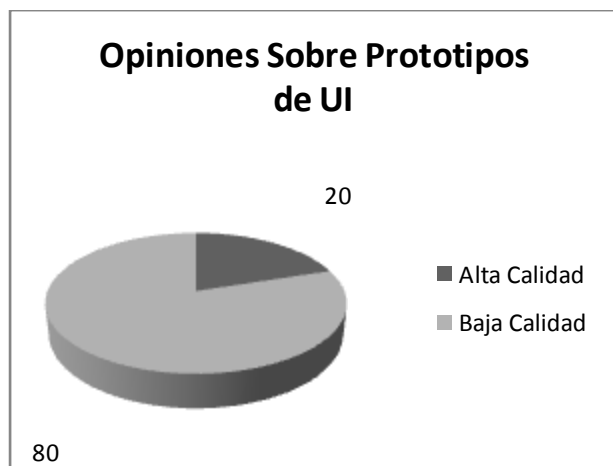
Diagrama de Interacción a Nivel Usuario



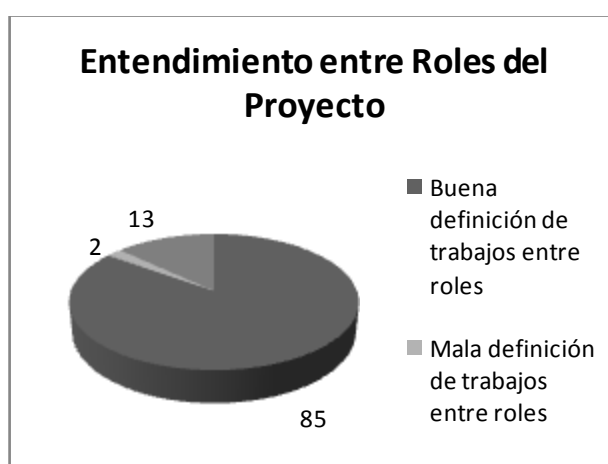
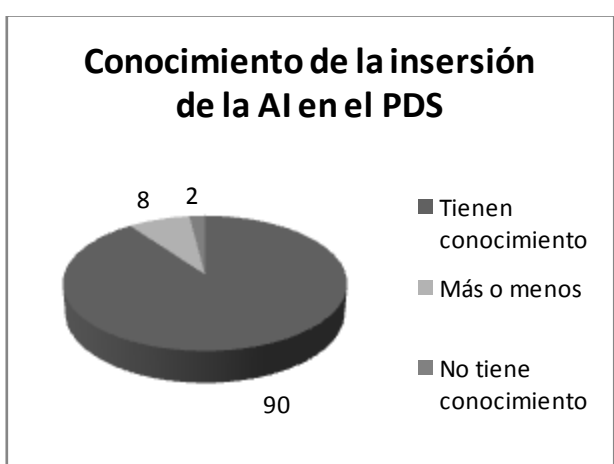
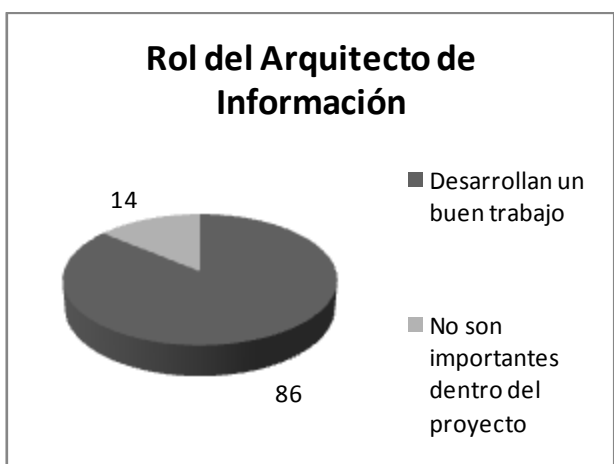
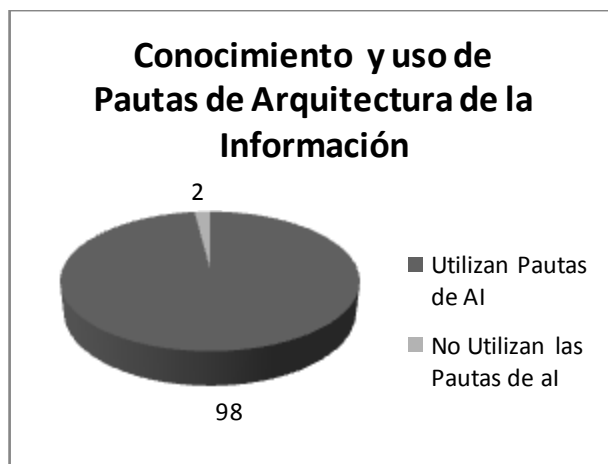
Anexo II: Librería de Componentes para el Proyecto CICPC



Anexo III: Resultados de la encuesta inicial



Anexo IV: Resultados de la encuesta después de aplicada la propuesta



Anexo V: Modelo para la recogida de información referente a la calificación de los criterios para la evaluación por expertos. Primera vuelta

Validación de Propuesta de Arquitectura de Información para el Proyecto CICPC

Nombre y Apellidos del Experto _____

Fecha de recepción _____ Fecha de entrega _____

Le otorgará un valor a cada criterio entre 1 y 10 donde 10 es la máxima puntuación, de acuerdo con su opinión.

Sobre el mérito científico

1. Valor novedoso de la propuesta _____
2. Calidad de la propuesta _____
3. Novedad científica _____

Sobre las ventajas de su Implantación

4. Satisfacción de las necesidades _____
5. Necesidad del empleo de la propuesta _____

Sobre la Flexibilidad

6. Adaptabilidad a otros proyectos productivos con características similares _____
7. Capacidad de lograr un proyecto de calidad _____

Sobre el Impacto

8. Repercusión en el proyecto productivo (de acuerdo con las interfaces de usuario antes de la propuesta y después) _____
9. Mayor calidad en los proyectos productivos que se adapten a la propuesta _____
10. Posibilidades de aplicación _____
11. Impacto en el área para la cual está destinada _____

Sugerencias del experto para mejorar la calidad del proyecto

Elementos críticos señalados que deben mejorarse.

Anexo VI: Modelo para la recogida de información referente a la calificación de los criterios para la evaluación por expertos. Segunda vuelta

Validación de Propuesta de Arquitectura de Información para el Proyecto CICPC

Nombre y Apellidos del Experto _____

Fecha de recepción _____ Fecha de entrega _____

Le otorgará un valor a cada criterio entre 1 y 10 donde 10 es la máxima puntuación, de acuerdo con su opinión.

1. ¿Considera usted que la propuesta ofrece un contenido nuevo dentro de la rama de la informática aplicada y el desarrollo de software en Cuba? _____
2. ¿Cuenta la propuesta con una alta Calidad para su entendimiento, aceptación y aplicación? _____
3. ¿Desde el punto de vista científico, se considera una novedad en la informática aplicada la combinación de las teorías planteadas en la investigación? _____
4. ¿Satisface la propuesta las necesidades actuales de la producción de software en la rama concerniente? _____
5. ¿Existe necesidad de soluciones a los problemas planteados por la investigación? _____
6. ¿Puede ser adaptada la propuesta a un proyecto con las características similares a las propuestas? _____
7. ¿Ofrecerá para estos un producto con alta calidad? _____
8. ¿La aplicación de la estrategia a un proyecto productivo será vista de manera favorable de acuerdo con las interfaces de usuario después de su aplicación? _____
9. ¿Mejorará la calidad de los proyectos productivos que se adapten a la propuesta ante aquellos que no tengan ninguna parecida? _____
10. ¿Cree posible aplicarlo en varios proyectos de la UCI? _____
11. ¿De qué rango se puede clasificar el impacto esperado en el área para la cual está destinada? _____

De acuerdo con todos los valores otorgados en los criterios anteriores, establezca valores para una categoría final.

- Categoría final del proyecto

___ Excelente si posee alta novedad científica, con aplicabilidad y resultados relevantes.

___ Bueno si posee novedad científica, resultados destacados.

___ Aceptable si posee suficientemente bueno con reservas.

___ Cuestionable si cree que no tiene relevancia científica y los resultados son malos.

___ Malo si cree que no aplicable.

- Valoración final

Sugerencias del experto para mejorar la calidad del proyecto

Elementos críticos que deben mejorarse.
