

Universidad de las Ciencias Informáticas
Facultad 8



METODOLOGÍA PARA EL DISEÑO Y DESARROLLO DE INTERFACES DE USUARIO DEL SIIPOL



TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS

Autor(es): Yudenia Velazco Socarras
Greilan Garcia Balmaseda

Tutor(es): Ing. Lesky Alfonso De León
D.I. Héctor M. Nápoles Pompa

LA HABANA, JUNIO DE 2008

Declaración de Autoría

Por este medio declaramos que somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

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

Yudenia Velazco Socarras

Autor

Greilan Garcia Balmaseda

Autor

Ing. Lesky Alfonso De León

Tutor

D.I. Héctor M. Nápoles Pompa

Tutor

Opinión del Usuario del Trabajo de Diploma

El Trabajo de Diploma, titulado “Metodología para el diseño y desarrollo de interfaces de usuario del SIIPOL”, fue realizado en la Universidad de Ciencias Informáticas. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un 95 %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

1. Organización del flujo de trabajo para llevar a cabo un diseño y desarrollo de interfaces de usuario con calidad y eficiencia.
2. Desarrollo de interfaces acordes a los procesos complejos que se informatizan, definiendo componentes visuales reutilizables que permiten el manejo sencillo de las funcionalidades definidas para informatizar los procesos identificados.
3. Aumento en la calidad del diseño y desarrollo de las interfaces de usuario en el proceso de implementación del software, debido al aprovechamiento del tiempo establecido en cada fase definida en la metodología.
4. Entrega de interfaces finales amigables, con gran usabilidad y adecuadas al diseño deseado, permitiendo romper las barreras de la comunicación entre el usuario y la máquina.

Como resultado de la implantación de este trabajo se reportará un efecto económico que asciende a 694.4 USD.

Y para que así conste, se firma la presente a los ____ días del mes de _____ del año _____.

Representante de la entidad

Cargo

Firma

Cuño

Opinión de los Tutores del Trabajo de Diploma

Los Tutores del presente Trabajo de Diploma consideran que durante su ejecución las estudiantes mostraron las cualidades que a continuación se detallan.

Muy alta independencia así como originalidad y creatividad en la confección de su trabajo de diploma el cual asumieron con un alto nivel de responsabilidad cumpliendo objetivos y tareas necesarias en tiempo y con calidad. La laboriosidad que demostraron dio como resultado la posibilidad de adquirir y aplicar conocimientos y buenas prácticas dentro del proceso de desarrollo del proyecto.

El presente trabajo contribuye a mejorar y agilizar el proceso de diseño e implementación de las interfaces de usuario para la aplicación que se construye en el proyecto CICPC. El valor científico-técnico viene reflejado en el avance y calidad del proceso de diseño y desarrollo de interfaces del SIIPOL, haciendo de estas interfaces un producto agradable, robusto y funcional.

Por todo lo anteriormente expresado considero que las estudiantes están aptas para ejercer como Ingenieras Informáticas; y proponemos que se le otorgue al Trabajo de Diploma la calificación de 5 puntos. Además considero que el resultado del trabajo posee valor para ser publicado y extendido su uso.

Ing. Lesky Alfonso De León

Firma

D.I. Héctor M. Nápoles Pompa

Firma

Fecha

Datos de contacto

Nombre y Apellidos: Lesky Alfonso De León.

Fecha de nacimiento: 24 de mayo de 1980.

Institución: Universidad de las Ciencias Informáticas.

País: Cuba.

Ciudadanía: Cubana.

Carnet de Identidad: 80052410108

Pasaporte B001540

Correo: leskyalf@uci.cu

Situación laboral: Profesor Instructor, Departamento de la Especialidad

Institución: Universidad de las Ciencias Informáticas

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba,

Código postal 19370.

Graduado de Ingeniería Informática en el Instituto Superior Politécnico José Antonio Echeverría, año 2005. Graduado de Profesor en Ciencias Informáticas en el Instituto Superior Politécnico José Antonio Echeverría, año 2005.

Al graduarse se integra el claustro de profesores de la Universidad de las Ciencias Informáticas, en la Disciplina de Programación. Obtiene la categoría de Profesor Instructor en octubre del 2006. Ha impartido asignaturas como Introducción a la Programación, Programación I, Técnicas de modelación y diseño avanzado. Ha ejercido como tutor de varias tesis de grado, así como ponente y miembro de tribunales. Ha trabajado en proyectos productivos de exportación como el Proyecto Identidad y el Proyecto Prisiones, desempeñando roles como: Desarrollador, Especialista de Capacitación, Despliegue y Soporte Técnico. Actualmente desempeña el rol de Jefe de Sistema en el proyecto CICPC.

Nombre y Apellidos: Héctor Manuel Nápoles Pompa.

Fecha de nacimiento: 30 de Agosto de 1981.

Institución: Universidad de las Ciencias Informáticas.

País: Cuba.

Ciudadanía: Cubana.

Carnet de Identidad: 81083019849

Pasaporte: B244614

Correo: hectornp@uci.cu

Situación laboral: Especialista General, Dirección de Comunicación Visual.

Institución: Universidad de las Ciencias Informáticas.

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba,

Código postal 19370.

Graduado como Diseñador Informacional (Especialista en Comunicación Visual), en el Instituto Superior de Diseño Industrial, ubicado en Centro Habana, año 2005.

Al graduarse se incorpora al colectivo de diseñadores de la Dirección de Comunicación Visual, de la Universidad de Ciencias Informáticas. Ha participado desde sus inicios en tareas productivas y en el desarrollo de productos de alcance nacional e internacional desarrollados en la Universidad. Ha impartido cursos optativos relacionados con su especialidad y trabajado en proyectos productivos nacionales y de exportación como el Proyecto Identidad, Registro y Notaría, CTAISC, y el Proyecto Prisiones, desarrollando diversos roles, a cargo del Diseño de las interfaces gráficas, desarrollando identidades visuales o como asesor comunicacional, entre otros. Actualmente desempeña el rol de Líder del Suproyecto Comunicación Institucional perteneciente el proyecto CICPC.



“Sólo es posible avanzar cuando se mira lejos. Solo cabe progresar cuando se piensa en grande.”

Agradecimientos

A la Revolución por darnos la oportunidad de conocer la materialización de un sueño del compañero Fidel Castro.

A la Universidad de Ciencias Informáticas, por hacernos parte de su inmensa tropa de futuro.

A nuestro tutor Lesky Alfonso De León, por brindarnos sus conocimientos, ayudarnos y guiarnos durante toda la investigación.

A Héctor Manuel Nápoles Pompa, por su ayuda incondicional, por sus ideas siempre oportunas.

Al colectivo de trabajo del proyecto CICPC, por ser la base para el desarrollo de la investigación.

A todos y cada uno de los compañeros que nos acompañaron durante cinco años, por todos los momentos inolvidables.

A nuestros padres y familiares por su apoyo, amor, confianza, dedicación, por guiarnos y apoyarnos siempre.

A todos los que de una manera u otra nos han brindado su apoyo y han sido parte de este logro.

A todos, Gracias.

Dedicatorias

Dedico este trabajo y todos los méritos alcanzados durante mis estudios a mi tío
Rogelio.

A toda mi familia, a mis padres y mi abuelo por quererme y apoyarme tanto.

A mi novio Yoanky, por amarme y estar siempre a mi lado.

A mi hermana y mis dos bellas sobrinas.

A todos mis amigos.

YUDENIA VELAZCO SOCARRAS

A mis padres, por ser los mejores del mundo.

GREILAN GARCIA BALMASEDA

Resumen

El inminente avance de las ciencias informáticas ha provocado la evolución de la industria cubana del software. La Universidad de las Ciencias Informáticas (UCI) promueve el desarrollo de la informática tanto en 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 de Investigación e 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 dicho país.

La creación de las interfaces de usuario (UI) es una etapa importante en el proceso de desarrollo del software, influyendo en gran medida en su efectividad comunicacional, facilidad de uso e impacto generado en el usuario. El trabajo que se presenta a continuación, propone el desarrollo de una metodología encaminada a mejorar y agilizar el proceso de diseño y desarrollo de UI para el SIIPOL.

La efectividad de este trabajo se refleja en el avance y calidad del proceso de montaje de UI de la aplicación que se construye en el proyecto CICPC.

Palabras Clave

Cuerpo de Investigaciones Científicas Penales y Criminalísticas (CICPC)

Sistema de Investigación e Información Policial (SIIPOL)

Metodología, Interfaz de Usuario (UI), Diseño, Desarrollo, Framework, Usabilidad, Accesibilidad, Componentes

INTRODUCCIÓN	1
FUNDAMENTOS DEL TEMA	5
1.1 INTRODUCCIÓN	5
1.2 INTERFACES PARA SISTEMAS INFORMÁTICOS	5
1.2.1 Conceptos fundamentales	6
1.3 HISTORIA Y EVOLUCIÓN DE LAS INTERFACES	7
1.3.1 Interfaces por Comandos	8
1.3.2 Interfaces Gráficas	9
1.3.3 Interfaces Web	10
1.4 TECNOLOGÍAS PARA EL DESARROLLO DE INTERFACES WEB	11
1.4.1 HTML	12
1.4.2 CSS	13
1.4.3 Javascript	13
1.4.4 Ajax	14
1.4.5 JSF	16
1.4.6 Flash	17
1.5 HERRAMIENTAS ÚTILES PARA LA ELABORACIÓN DE UI	19
1.5.1 Dreamweaver	19
1.5.2 Eclipse	20
1.5.3 Adobe Flash	20
1.6 ANÁLISIS DE OTRAS SOLUCIONES EXISTENTES	21
1.6.1 Antecedentes en la UCI	29
1.7 CONCLUSIONES	30
METODOLOGÍA PARA EL DISEÑO Y DESARROLLO DE UI DEL SIIPOL	31
2.1 INTRODUCCIÓN	31
2.2 PRESENTACIÓN DE DDWUI	31
2.3 DEFINICIONES PRELIMINARES	32
2.4 ASPECTOS ESENCIALES	33
2.4.1 Crear equipo de montadores UI	34
2.4.2 Capacitar equipo de montadores UI	34
2.4.3 Realizar casos de estudio	35
2.5 FASES DE LA METODOLOGÍA	35
2.5.1 Estudio y modelado del Sistema	37

2.5.1.1	Objetivos	37
2.5.1.2	Entrada	37
2.5.1.3	Actividades.....	37
2.5.1.4	Salida	38
2.5.1.5	Personas responsables	38
2.5.1.6	Herramientas.....	39
2.5.2	Arquitectura de la Información.....	39
2.5.2.1	Objetivos	39
2.5.2.2	Entrada	40
2.5.2.3	Actividades.....	40
2.5.2.4	Salida	43
2.5.2.5	Personas responsables	43
2.5.2.6	Herramientas.....	43
2.5.3	Diseño Gráfico	44
2.5.3.1	Objetivos	44
2.5.3.2	Entrada	44
2.5.3.3	Actividades.....	44
2.5.3.4	Salida	47
2.5.3.5	Personas responsables	47
2.5.3.6	Herramientas.....	47
2.5.4	Elaboración de los recursos de montaje de interfaces	48
2.5.4.1	Objetivos	48
2.5.4.2	Entrada	48
2.5.4.3	Actividades.....	48
2.5.4.4	Salida	53
2.5.4.5	Personas responsables	54
2.5.4.6	Herramientas.....	54
2.5.5	Desarrollo de las UI	54
2.5.5.1	Objetivos	54
2.5.5.2	Entrada	54
2.5.5.3	Actividades.....	55
2.5.5.4	Salida	55
2.5.5.5	Personas responsables	55
2.5.5.6	Herramientas.....	56
2.6	CONCLUSIONES.....	56

ANÁLISIS DE LOS RESULTADOS.....	58
3.1 INTRODUCCIÓN.....	58
3.2 VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN.....	58
3.2.1 Reportes de los resultados antes de aplicar DDWUI	58
3.2.2 Reportes de los resultados después de aplicar DDWUI.....	59
3.2.3 Determinación del valor práctico de la solución según expertos	60
3.3 ENTREGABLES OBTENIDOS.....	61
3.4 CONCLUSIONES.....	66
CONCLUSIONES.....	67
RECOMENDACIONES	68
REFERENCIAS BIBLIOGRÁFICAS.....	69
BIBLIOGRAFÍA.....	72
GLOSARIO DE TÉRMINOS.....	74
ANEXOS.....	77
Anexo I. Encuesta realizada a integrantes del CICPC antes de aplicar DDWUI	77
Anexo II. Encuesta realizada a integrantes del CICPC después de aplicar DDWUI.....	78
Anexo III. Interfaz por Comandos.....	79
Anexo IV. Interfaz Gráfica Windows 3.11.....	79
Anexo V. Interfaz Gráfica Windows 95.....	80
Anexo VI. Interfaz Gráfica Windows XP	80
Anexo VII. Interfaz Web 1	81
Anexo VIII. Interfaz Web 2	81
Anexo IX. Seccionamiento de imágenes mediante la herramienta Slice 1	82
Anexo X. Seccionamiento de imágenes mediante la herramienta Slice 2	82
Anexo XI. Resultados de la encuesta inicial.....	83
Anexo XII. Resultados de la encuesta después de aplicar DDWUI	84
Anexo XIII Cuestionario para la validación de la metodología DDWUI	85
Anexo XIV Evaluación de Expertos.....	85
Anexo XV Especificación del Caso de Uso.....	86
Anexo XVI Documento de Pautas de Arquitectura de la Información	86
Anexo XVII Prototipo de Interfaz de Usuario del Caso de Uso	86
Anexo XVIII Documento de Pautas de Diseño Gráfico	86
Anexo XIX Representación Gráfica de las pantallas.....	86
Anexo XX Muestras de pantallas	86
Anexo XXI	86

Anexo XXII Hojas de Estilo.....	86
Anexo XXIII Interfaces.....	86

Introducción

Con el avance tecnológico, el desarrollo de la informática y el continuo progreso de la industria del software, se incrementan las necesidades de digitalización de la información y el uso de las tecnologías, como medios para lograr el cumplimiento de los objetivos y metas de los usuarios. Para la inmensa mayoría de las empresas es de vital importancia garantizar un soporte informático que satisfaga las necesidades de gestión de la información. En Cuba, el desarrollo de software progresa y será futuro sendero y sostén de la economía.

El Ministerio de la Informática y las Comunicaciones de Cuba (MIC) y la UCI, han aprobado contratos de negocios con varias empresas venezolanas, relacionados con las Tecnologías de la Informática y las Comunicaciones (TIC), para la creación de proyectos software. Los proyectos se distribuyen y realizan según los perfiles de producción establecidos en la universidad. Entre los proyectos que pertenecen a la línea de producción de software de gestión se encuentra en desarrollo en la UCI el proyecto productivo “CICPC”, que se encarga de la modernización y mantenimiento del SIIPOL para sustituir la manera en que se maneja el extenso flujo de información de dicha institución.

En la actualidad, el empleo y desarrollo de las UI basadas en la tecnología web como soporte de información se incrementa a cada instante. Se hace imprescindible que la interacción hombre-computadora sea cada vez mejor a través de una adecuada UI, que brinde comodidad y eficiencia. La UI representa el punto de encuentro entre el usuario y la computadora, definiéndose por un conjunto de componentes que el usuario utiliza para comunicarse con ella.

La UI es la parte de una aplicación que el usuario ve y con la cual interactúa. En esta interacción, el usuario juzga la utilidad de la interfaz; el hardware y el software se convierten en simples herramientas sobre los cuales fue construida. Al pensar en la UI la mayoría de los desarrolladores de software piensan en ventanas, menús, gráficos, colores y se olvidan del componente principal: el usuario, por esta razón los programadores no son las personas más indicadas para llevar a cabo el diseño y desarrollo de las mismas; se hace necesario que dentro del equipo de desarrollo de software existan personas especializadas en la creación de UI para de esta forma facilitar su empleo y lograr una mayor identificación del usuario con el producto.

En la mayoría de los proyectos que se llevan a cabo en la UCI no existen personas que se dediquen exclusivamente al trabajo con las UI y se especialicen en ello, son los mismos programadores los que las desarrollan según sus necesidades. Actualmente no se ha definido un procedimiento o metodología encaminada a mejorar y agilizar el montaje de UI de los proyectos productivos que en la UCI se desarrollan.

En el proyecto CICPC se comenzó su etapa de implementación, para ello se utilizó el lenguaje de programación orientado a objetos multiplataforma Java. Inicialmente existe poco conocimiento sobre cómo unir la lógica del negocio con la forma en que se presentará ésta a los usuarios finales del software. Hasta el momento, las UI se desarrollan en lenguaje HyperText Markup Language (HTML) utilizando Dreamweaver que a pesar de ser una potente herramienta para el diseño de páginas web, realizar el montaje de las UI se hace un poco difícil a la hora de ajustarse a las pautas de diseño establecidas por el diseñador; se utiliza como plantilla un archivo .dwt, que se aplica a todas las páginas, lo que trae como consecuencia que exista un cantidad innecesaria de código repetido en cada una de ellas. No se utiliza un framework de Java para llevar a cabo un correcto diseño y desarrollo de las UI del proyecto, esto provoca que los programadores deban modificar y adaptar los componentes utilizados hasta lograr el prototipo funcional que necesitan para luego proceder con su implementación.

Esta situación produce un atraso en la realización del proyecto puesto que se hace complejo el diseño de las UI, aumenta el tiempo de implementación y no se logra la calidad requerida en el resultado final.

Teniendo en cuenta las consideraciones anteriormente expuestas y la necesidad de encontrar una solución para estas dificultades que frenan el desarrollo del proyecto CICPC en general, se plantea el siguiente **problema científico**: Ausencia de una metodología que guíe el proceso de diseño y desarrollo de UI en la aplicación SIIPOL.

Para dar solución a la problemática presentada se define como **objeto de estudio**: Proceso de diseño y desarrollo de UI para sistemas empresariales de gestión sobre web.

Del objeto de estudio analizado se define como **campo de acción**: Diseño y desarrollo de las UI para la aplicación SIIPOL.

Objetivo general:

Desarrollar una metodología orientada a guiar el proceso de diseño y desarrollo de UI de la aplicación SIIPOL.

Para cumplir con el objetivo planteado y resolver los problemas mencionados se proponen las siguientes **Tareas investigativas**:

1. Identificar guías y tendencias actuales en el desarrollo de UI.
2. Estudiar las principales características, principios y pautas de diseño de las UI.
3. Hacer un estudio comparativo de aplicaciones similares desarrolladas con anterioridad.
4. Analizar resultados de las UI de diferentes proyectos realizados en la UCI.
5. Estudiar las distintas tecnologías y herramientas orientadas al diseño de UI y seleccionar las adecuadas según las características de la aplicación.
6. Diseñar y describir la metodología para el desarrollo de las UI de la aplicación SIIPOL.
7. Aplicar la metodología para desarrollar las UI del SIIPOL. Valorar ventajas y limitaciones.

Idea a Defender:

Si se define y aplica una metodología que oriente el proceso de diseño y desarrollo de UI de la aplicación SIIPOL, se obtendrán interfaces mejor construidas, de mayor aceptación por el usuario, en menor tiempo.

Para constatar el problema que da origen a la investigación y cumplir el objetivo propuesto se utiliza como estrategia de investigación, la investigación explicativa o experimental. Con ayuda de varias técnicas como entrevistas y encuestas, que se realizan para evaluar el estado y funcionamiento del proyecto y las dificultades que presenta el mismo relacionadas con las UI, las mismas demuestran que existe inexperiencia respecto a las actividades relacionadas con el montaje de UI y desconocimiento de herramientas tecnológicas para desarrollarlas, provocando afectaciones durante el proceso de desarrollo del proyecto.

Entrevistas: Se entrevistaron profesores y estudiantes que pertenecen al proyecto, que desempeñan diferentes roles dentro del mismo, entre ellos, los especializados en el trabajo con las UI.

Encuestas: Se realizaron encuestas a montadores de UI y programadores del proyecto. (Ver Anexo I)

Para dar cumplimiento a los objetivos planteados en este trabajo se emplea la siguiente metodología de investigación: muestra y población, las mismas fueron seleccionadas intencionalmente e integradas por docentes y estudiantes que forman parte del proyecto CICPC.

Población: Todos los integrantes del proyecto CICPC.

Unidad de estudio: Estudiantes que integran el equipo de montadores de UI y algunos programadores del proyecto CICPC.

Debido a que el equipo de montadores UI del proyecto está conformado por un reducido grupo de personas, no es necesario hacer uso de las técnicas de muestreo y la muestra será exactamente ese mismo grupo de personas.

De los métodos científicos de la investigación, se hace uso de los métodos teóricos: histórico-lógico, hipotético-deductivo, de los métodos empíricos: la observación y la entrevista.

Esta investigación se realiza en busca del avance y mejora en la actividad de desarrollo de UI, con el objetivo de facilitar el trabajo a los montadores de UI y programadores, logrando con ello el progreso del proyecto. Propone una metodología que brinde directrices de trabajo, y adelante gran parte de éste, normalice y estandarice el desarrollo de las UI en el SIIPOL. Los aportes que se esperan de esta investigación son principalmente: contribuir a la mejora y agilización del proceso de montaje de UI para el SIIPOL, facilitar el trabajo de los programadores y servir de referencia a proyectos similares.

Para realizar una descripción detallada, este documento muestra el resultado de la investigación distribuido de la siguiente manera:

Capítulo 1: Fundamentos del Tema. Describe conceptos fundamentales que ayudan a la comprensión del tema tratado en la investigación.

Capítulo 2: Metodología para el diseño y desarrollo de UI de SIIPOL. Describe la investigación científica realizada y propone la metodología orientada al diseño y desarrollo de UI de SIIPOL.

Capítulo 3: Análisis de los Resultados. Se valoran los resultados, qué aportó al proyecto "CICPC" la aplicación de la metodología.

El documento tiene además las conclusiones del trabajo, recomendaciones, bibliografía y un conjunto de anexos que permiten una mejor ilustración del estudio realizado.



Fundamentos del Tema

1.1 Introducción

En este capítulo se definen un grupo de conceptos importantes relacionados con las interfaces de sistemas informáticos, su historia y evolución hasta llegar a las interfaces web, en las que se centrará esta investigación, y para las que se estudian además las tendencias en cuanto a tecnologías y herramientas. Se enfocará la atención principalmente en el análisis realizado sobre algunas soluciones para el proceso de diseño y desarrollo de interfaces ya existentes, aportando una base teórica que posibilitará argumentar la necesidad de conformar una nueva propuesta que guíe dicho proceso para la aplicación que construye el proyecto CICPC.

1.2 Interfaces para sistemas informáticos

Cuando un usuario usa una herramienta, o accede e interactúa con un sistema informático, suele existir "algo" entre el usuario y el objeto de interacción. Ese algo, que es a la vez un límite y un espacio común entre ambas partes, es la interfaz. Hoy en día es posible la interacción con el mundo informático que nos rodea a través de cientos de ellas. La interfaz de un programa es un conjunto de elementos hardware y software de una computadora que muestran información al usuario y le permiten interactuar con dicha información y con la máquina. Los programas son usados por usuarios con distintos niveles de conocimientos, desde principiantes hasta expertos, por lo que no existe una interfaz válida para todos los usuarios y todas las tareas por eso, la elaboración de las interfaces, exige una gran dedicación pues generalmente son grandes, complejas y difíciles de implementar. (1)

Diseñar y crear las interfaces es una de las partes más importantes de cualquier software. Un programa muy poderoso con una interfaz pobremente elaborada tiene poco valor para un usuario. Hoy en día las interfaces son prácticamente universales, las que utilizan ventanas, íconos y menús se han convertido en estándares de diseño para los sistemas informáticos. Un buen diseño de las interfaces puede tener sin dudas, un impacto significativo en el éxito del software construido.

1.2.1 Conceptos fundamentales

El objetivo de la presente investigación es proponer una metodología orientada a guiar el diseño y desarrollo de las UI de la aplicación SIIPOL, para hacer referencia al término metodología de UI es imprescindible definir algunos conceptos como interfaz e interfaz de usuario, abordar su historia y evolución y detallar, dentro del proceso de construcción de estas, las tendencias de tecnologías y herramientas actuales implicadas en dicho proceso.

Interfaz

En términos generales, una interfaz es el punto, el área, o la superficie a lo largo de la cual dos objetos de naturaleza distinta convergen. (2)

En electrónica, telecomunicaciones y hardware, una interfaz es el puerto (circuito físico) a través del cual se envían o reciben señales desde un sistema o subsistemas hacia otros. No existe una interfaz universal, sino que existen diferentes estándares (Interfaz USB, interfaz SCSI, etc. (3)

El Diccionario de la Real Academia de la Lengua Española define interfaz como una palabra derivada del término inglés “interface” (superficie de contacto) y la define de la siguiente manera: *Conexión física y funcional entre dos aparatos o sistemas independientes.* (4)

A partir de las definiciones anteriores los autores de esta investigación proponen la siguiente definición de interfaz: dados dos sistemas cualesquiera que se deben comunicar entre ellos la interfaz será el mecanismo, entorno o herramienta que hace posible dicha comunicación. Esta definición es amplia en sí misma, utilizándose para describir multitud de entornos de comunicación entre sistemas físicos, electrónicos y lógicos, utilizándose para referirse a cualquier dispositivo que permite establecer una comunicación entre dos unidades de diferente naturaleza o a determinados componentes de software que habilitan el entendimiento correcto entre dos aplicaciones u objetos lógicos.

Cuando uno de los sistemas que se comunican es un ser humano, se pasa entonces al concepto de interfaz de usuario. Por un lado tenemos un sistema físico o informático y por otro a una persona que desea interactuar con él, siendo la interfaz de usuario la herramienta que entiende a ambos y es capaz de traducir los mensajes que se intercambian.

Interfaz de Usuario

Es la parte de una aplicación que se encarga de interactuar con el usuario. (5)

Una conexión e interacción entre hardware, software y usuario, es decir como la plataforma o medio de comunicación entre usuario o programa. (5)

La interfaz de usuario es la parte de cualquier equipo, sistema que determina el control y la forma en que las personas operan el sistema. Cuando la interfaz está bien diseñada, es comprensible, previsible y controlable; usuarios se sientan competentes, satisfechos y responsables de sus actos. (6)

Teniendo en cuenta las definiciones de interfaz de usuario encontradas para fomentar el tema de la investigación se puede afirmar que una interfaz de usuario es un sistema de comunicación de un programa con su usuario, comprende las pantallas y los elementos que informan al usuario sobre lo que puede o no hacer, o sobre lo que está ocurriendo. Cuando en este trabajo se habla de interfaz, se hace referencia a la cara visible de los programas tal y como se presenta a los usuarios para que interactúen con el sistema. La interfaz de usuario implica la presencia de una serie de pantallas constituidas por menús, iconos entre otros elementos que representan las opciones que el usuario puede tomar dentro del sistema, son utilizadas en multitud de sistemas y hacen la interacción usuario-computadora más cómoda y segura.

1.3 Historia y evolución de las interfaces

Inicialmente las interfaces de usuario eran desarrolladas pensando solamente en su correcto funcionamiento, luego se comprobó la necesidad de que fueran fácilmente usables y después se hizo patente la importancia de la estética en las mismas. Un sistema debe funcionar correctamente y además ser agradable para el usuario, acorde con las inquietudes estéticas del mundo que nos rodea. Colores, formas, agrupaciones y comunicación forman en la actualidad parte indispensable de una UI. Con la aparición de las computadoras y su posterior popularización se hizo necesario la creación de mecanismos que hicieran posible a un usuario comunicarse de forma correcta con los sistemas operativos y las aplicaciones que en ellos se estaban ejecutando, estos no son más que UI.

Las primeras interfaces se crearon en un principio en modo comandos, pero su utilidad práctica era limitada, ya que no eran aptas para usuarios que no poseían conocimientos avanzados de informática. Fue necesario entonces crear interfaces basadas en iconos y menús, accesibles por medio del mouse o el teclado, surgiendo los entornos de ventanas, como Windows o MAC, que empezaron a utilizar interfaces gráficas de usuario (GUI), verdaderas protagonistas del mundo informático actual.

En nuestros días los usuarios de aplicaciones informáticas están tan acostumbrados a las GUI que se puede afirmar que si desaparecieran la gran mayoría de ellos quedarían anulados. Tanto es así que sistemas operativos y programas que antes funcionaban casi exclusivamente en modo de comandos han tenido que actualizarse y ofrecer a los usuarios interfaces gráficas equivalentes para no quedarse olvidados por el público general.

Así estaban las cosas cuando se produjo un fenómeno sorprendente, llamado a revolucionar la comunicación entre seres humanos: Internet. Con la aparición de la web se hizo posible que cualquier persona pudiera ofrecer información particularizada a los demás y encontrar documentos interactivos sobre cualquier tema, relacionados unos con otros mediante enlaces que permitían acceder de una página a otra página alrededor del mundo.

Las páginas web supusieron la aparición de las interfaces web, interfaces gráficas de usuario con unos elementos comunes de presentación y navegación que pronto se convirtieron en estándares de facto. Este tipo de interfaces deben servir de intermediarias entre unos usuarios genéricos, no acostumbrados generalmente al uso de aplicaciones informáticas, y unos sistemas de información y procesos transaccionales que corren por debajo, debiendo posibilitar la localización de la información deseada, el entendimiento claro de las funcionalidades ofrecidas, la realización práctica de tareas específicas por parte de los usuarios y la navegación intuitiva por las diferentes páginas que forman el sitio web.

Buscando una homogeneidad entre los millones de páginas web que existen actualmente en Internet, el diseño de las mismas ha evolucionado con el tiempo hacia un esquema general, ofreciendo unas interfaces bien definidas, con un conjunto de componentes gráficos y funcionales similares que hacen posible que sea cual sea el usuario que accede a un sitio web cualquiera, la comunicación entre ellos sea posible y efectiva. De esta forma se han definido elementos y agrupaciones de estos que han demostrado su utilidad y su comprensión por los usuarios, entre los que podemos destacar los sistemas de navegación, los banners, los pies de página, los formularios de entrada de datos, etc., que normalmente encontraremos en todas las páginas web y cuyo diseño y funcionalidad son similares en todas ellas. (7)

1.3.1 Interfaces por Comandos

Interfaz por comandos también conocida como Intérprete de órdenes, Intérprete de mandatos, Intérprete de línea de mandatos, Terminal, Consola, Shell ó su acrónimo en inglés CLI por Command Line Interface, es un programa que actúa como UI para comunicar al usuario con el

sistema operativo mediante una ventana que espera órdenes escritas por el usuario, los interpreta y los entrega al sistema operativo para su ejecución. La respuesta del sistema operativo es mostrada al usuario en la misma ventana y a continuación el programa queda esperando más instrucciones. Esa forma de trabajo es interactiva, es decir usuario y máquina se comunican en forma sucesiva, se interactúa con la información de la manera más sencilla posible, sin gráficos, sólo texto. (Ver Anexo III)

Su contraparte es la Interfaz gráfica de usuario que ofrece una estética mejorada a costa de mayor consumo de recursos computacionales, una mayor vulnerabilidad por complejidad y, en general, una reducción en la funcionalidad ofrecida. Comparada con un ambiente gráfico la CLI consume menos recursos del CPU, es más rápida y más fácil de utilizar a través de una red de computadoras.

Como herramienta primaria de trabajo, las CLI son usadas ampliamente por programadores y administradores de sistemas, especialmente en sistemas operativos basados en Unix; en entornos científicos y de ingeniería; y por un subconjunto más pequeño de usuarios hogareños avanzados. Las CLI son parte integral de varias aplicaciones como interfaz alternativa y/o paralela a las GUI. En la actualidad permanecen como un importante (y a veces único) medio para interactuar con programas. (7)

1.3.2 Interfaces Gráficas

La GUI es un tipo de interfaz de usuario que utiliza un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles del sistema. Habitualmente las acciones se realizan mediante manipulación directa para facilitar la interacción del usuario con la computadora. (Ver Anexos IV, V y VI)

Las interfaces gráficas surgen de la necesidad de hacer las computadoras más accesibles para el uso de los usuarios comunes. La mayoría de computadoras ofrecían una interfaz de línea de comandos, lo que requería conocimientos por encima de la media si se deseaba hacer algo más que usarlo como consola de videojuegos. Esta limitación fue salvada gracias al desarrollo de los entornos gráficos, que permitieron que las personas pudieran acceder a una computadora sin tener que aprender a manejar un entorno bajo línea de comandos. Con la idea de simplificar el uso de las computadoras para usuarios de todo tipo y no sólo para los expertos, se ha convertido en una práctica habitual utilizar metáforas visuales por medio de la interfaz gráfica de usuario para que el usuario interactúe y establezca un contacto más fácil e

intuitivo con la computadora. Un simple clic del mouse sobre algún gráfico (imagen) que aparece en la pantalla, sustituye a la tediosa tarea de escribir código fuente para que la computadora interprete que debe realizar alguna acción.

En 1981 aparecieron las primeras computadoras personales, llamadas PC, pero hasta 1993 no se generalizaron las GUI. Las computadoras Macintosh de la compañía Apple fueron las primeras que introdujeron las GUI quedando desplazadas por las del sistema operativo Windows de Microsoft con su escritorio y su sistema de ventanas sobre la pantalla, que se ha estandarizado y universalizado por su alto nivel de accesibilidad y funcionalidad. El diseño del espacio de trabajo y el refinamiento de esta nueva interfaz hacen que su popularidad alcance valores insospechados. La aparición del escritorio como espacio de trabajo, la barra de tareas y el famoso menú Inicio reformaron y mejoraron el aspecto y la comodidad a la hora de usar un PC.

La GUI proporciona al usuario el conjunto de posibilidades que brinda el software detallando lo que observa y escucha en cada momento, y las acciones que puede realizar, así como las respuestas que puede obtener del sistema. Una buena interfaz requiere poco esfuerzo por parte del usuario, simplicidad y funcionalidad. La historia reciente de la informática está indisolublemente unida a las interfaces gráficas, puesto que los sistemas operativos gráficos han producido grandes resultados en la industria del software y del hardware. (8)

1.3.3 Interfaces Web

Antes de hacer referencia a las Interfaces Web, es preciso esclarecer los conceptos servidor web y modelo cliente/servidor.

Servidor Web

Un servidor web es aquel que permite a los clientes compartir datos, documentos y multimedia en formato web.

Modelo Cliente/Servidor

Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o "clientes", resultan en un trabajo realizado por otros computadores llamados servidores".

Una aplicación web es un sistema que los usuarios utilizan accediendo a un servidor web a través de Internet o de una intranet. Las aplicaciones web son populares debido a la practicidad del navegador web como cliente ligero y la facilidad para actualizar y mantener las aplicaciones web sin distribuir e instalar software en miles de potenciales clientes.

Una Interfaz Web es medio de interacción entre los usuarios y el conjunto de páginas de un sitio web y las aplicaciones que corren por debajo de forma transparente al usuario. Al principio de la tecnología cliente-servidor, cada aplicación tenía su propio programa cliente y su interfaz de usuario, estos tenían que ser instalados separadamente en cada estación de trabajo. La web era simplemente una colección de páginas estáticas, documentos, etc., que podían consultarse o descargarse. El siguiente paso en su evolución fue la inclusión de un método para confeccionar páginas dinámicas. Dicho método fue llamado interfaz de entrada común conocido como CGI por sus siglas en inglés de Common Gateway Interface que definía un mecanismo mediante el cual se podía pasar información entre el servidor y programas externos. Los CGI siguen siendo muy utilizados, puesto que la mayoría de los servidores web los soportan debido a su sencillez. (Ver Anexos VII y VIII)

Actualmente, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores web comunes. Se utilizan lenguajes interpretados del lado del cliente, para añadir elementos dinámicos a la UI. Generalmente cada página web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva. Las interfaces web tienen ciertas limitaciones en la funcionalidad del cliente, métodos comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar y soltar no están soportados aún por las tecnologías web estándar. (9)

Debido a las características particulares del proyecto CICPC la aplicación que se moderniza, SIIPOL, es una aplicación Web por esta razón se debe centrar el estudio de esta investigación en las interfaces web. Para ello es preciso analizar las tecnologías y herramientas que se emplean en la actualidad para el desarrollo de este tipo de interfaces.

1.4 Tecnologías para el desarrollo de interfaces web

A partir de la revolución de las TIC y la inminente masificación de internet, con la incorporación de la computadora a los medios electrónicos y otras novedades de comunicación e información que se desarrollan y tienen aplicación social, la web se ha convertido en un gran medio de comunicación utilizado para el intercambio de información. Actualmente la inmensa mayoría de

las aplicaciones están en red. Los proyectos, aplicaciones y servicios que se desarrollan hoy, están basados en la web.

En la selección de las tecnologías a utilizar para el diseño y desarrollo de UI del SIIPOL, se realizó un estudio del estado actual de las mismas que en este momento se usan para el desarrollo de las UI; así como el análisis de los requerimientos impuestos por el cliente, llegando a la selección de las más apropiadas para darle cumplimiento a dichos requisitos y lograr la mayor satisfacción posible de los usuarios finales del producto.

1.4.1 HTML

HTML es el acrónimo inglés de HyperText Markup Language, que se traduce al español como Lenguaje de Etiquetas de Hipertexto. Es un lenguaje de marcado o lenguaje de marcas que no es más que una forma de codificar un documento que, junto al texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación, diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores como Internet Explorer, Firefox entre otros, el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para web.

El lenguaje HTML puede ser creado y editado con cualquier editor de textos básico que admita texto sin formato independientemente del Sistema Operativo. HTML no es una herramienta para procesar texto, una solución de edición electrónica, ni tampoco un lenguaje de programación. Su propósito fundamental es definir la estructura y apariencia básica de documentos y conjuntos de documentos y hacerlos más accesibles de tal manera que puedan ser manejados de forma rápida y fácil por un usuario en red.

Ofrece diferentes maneras de definir la apariencia de sus documentos: especificaciones de tipografía, saltos de línea y texto con formato previo. La apariencia es importante, porque puede lograr efectos tanto nocivos como benéficos sobre la forma en que los usuarios pueden tener acceso y utilizar la información de sus documentos de HTML.

Actualmente se encuentra en la versión 4.01 y empieza a proporcionar funcionalidades más avanzadas para crear páginas más ricas en contenido. Además se ha definido una especificación compatible con HTML, el XHTML (Extensible HyperText Markup Language) que se suele definir como una versión XML validable de HTML. (10)

1.4.2 CSS

CSS (Cascading Style Sheets) en español Hojas de Estilo en Cascada es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es muy útil para la creación de páginas web complejas. La separación de los contenidos y su presentación presenta numerosas ventajas, porque obliga a crear documentos bien definidos y con significado completo. Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

Si el lenguaje HTML se utiliza para marcar los contenidos, es decir, para designar lo que es un párrafo, lo que es un titular o lo que es una lista de elementos, el lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, la tabulación con la que se muestran los elementos de una lista, etc. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento. CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML. (11)

1.4.3 Javascript

Javascript es un lenguaje de programación interpretado (un lenguaje de tipo *script*). A pesar de que existen intérpretes no dependientes de ningún navegador, es un lenguaje de script que suele encontrarse vinculado a páginas web. Javascript y Java son dos lenguajes de programación distintos con filosofías muy diferentes. El único punto en común es la sintaxis, porque cuando Netscape diseñó Javascript, se inspiró en la sintaxis de Java. Dentro de sus características fundamentales se pueden destacar:

- Javascript es un lenguaje interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias Javascript contenidas en una página HTML y las ejecuta adecuadamente.

- Javascript es un lenguaje orientado a eventos. Cuando un usuario pulsa sobre un enlace o mueve el puntero sobre una imagen se produce un evento. Mediante Javascript se pueden desarrollar guiones que ejecuten acciones en respuesta a estos eventos.
- Javascript es un lenguaje orientado a objetos. Aunque el modelo de objetos de Javascript está reducido y simplificado, incluye los elementos necesarios para que los guiones puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador.

Script: cada uno de los programas, aplicaciones o trozos de código creados con el lenguaje de programación Javascript. Unas pocas líneas de código forman un script y un archivo de miles de líneas de Javascript también es un script.

Sintaxis: La sintaxis de un lenguaje de programación se define como el conjunto de reglas que deben seguirse al escribir el código fuente de los programas para considerarse como correctos para ese lenguaje de programación. La sintaxis de Javascript es muy similar a la de otros lenguajes como Java y C.

Todos los navegadores interpretan el código Javascript integrado dentro de las páginas web. Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del servidor. Javascript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML. Javascript siempre ha sido utilizado de forma masiva por casi todos los sitios de Internet. La aparición de Flash disminuyó su popularidad, porque Flash permitía realizar algunas acciones imposibles de llevar a cabo mediante un script. Sin embargo, la aparición del objeto XMLHttpRequest y la consiguiente creación de AJAX han permitido a Javascript volver a gozar de una popularidad sin igual dentro de los lenguajes de programación web. La inigualable popularidad de Javascript como lenguaje de programación de aplicaciones web se ha extendido a otras aplicaciones y otros entornos no relacionados con la web. Herramientas como Adobe Acrobat permiten incluir código Javascript en archivos PDF. Otras herramientas de Adobe como Flash y Flex utilizan ActionScript, un dialecto del mismo estándar de Javascript. Photoshop permite realizar pequeños scripts mediante Javascript y la versión 6 de Java incluye un nuevo paquete (denominado javax.script) que permite integrar ambos lenguajes. (12)

1.4.4 Ajax

El término Ajax es un acrónimo de Asynchronous Javascript + XML, que se puede traducir como “Javascript asíncrono + XML”.

“Ajax” no es una tecnología, en realidad, se trata de la unión de varias tecnologías que se desarrollan de forma independiente y que se unen de manera sorprendente. Las tecnologías que forman Ajax son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM (Documento Object Model), para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- Javascript, para unir todas las demás tecnologías.

Desarrollar aplicaciones Ajax requiere un conocimiento de todas las tecnologías anteriores.

En las aplicaciones web tradicionales, las acciones del usuario en la página (hacer clic en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario. Esta técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, la aplicación web se convierte en algo más molesto que útil. Ajax permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

Las aplicaciones construidas con Ajax eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de Ajax mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor. Desde su primera definición, se han creado cientos de aplicaciones basadas en Ajax que en la mayoría de casos pueden sustituir completamente a otras técnicas como Flash y en el caso de las aplicaciones más avanzadas, pueden sustituir a complejas aplicaciones de escritorio.

Los proyectos implementados con Ajax demuestran que Ajax no es solo técnicamente importante, sino también prácticos para aplicaciones en el mundo real. Esta no es otra

tecnología para trabajar en un laboratorio, las aplicaciones Ajax pueden ser de cualquier tamaño, pueden ir de las más simples, a las muy complejas y sofisticadas. (13)

1.4.5 JSF

JavaServer Faces (JSF) es un framework para aplicaciones Java basadas en la web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF usa Java Server Pages (JSP) como la tecnología que permite hacer el despliegue de las páginas.

JSF pretende normalizar y estandarizar el desarrollo de aplicaciones 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. La tecnología JSF surge como una solución a la separación entre la presentación y el comportamiento en una aplicación Web, de forma tal que las actividades de los autores de las páginas puedan separarse de las actividades de los desarrolladores de la lógica. Permite programar más rápidamente, ahorra trabajo, da consistencia al código, permite separar presentación de negocio y mejora la seguridad de la aplicación JSF es posterior a la herramienta de soporte para el desarrollo de aplicaciones Web Apache Struts, por lo que se nutre de su experiencia y mejora algunas sus deficiencias.

Framework: es una estructura de soporte definida en la cual otro proyecto puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio. Los frameworks son diseñados con el intento de facilitar el trabajo a los desarrolladores de software.

Java EE: Java Platform, Enterprise Edition (anteriormente conocido como Java 2 Platform, Enterprise Edition o J2EE hasta la versión 1.4), es una plataforma de programación, parte de la Plataforma Java para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de n niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

JSP: es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Permite la utilización de código Java mediante scripts. Además es posible utilizar algunas acciones JSP predefinidas mediante etiquetas que pueden

ser enriquecidas mediante la utilización de Librerías de Etiquetas externas e incluso personalizadas. (14)

1.4.6 Flash

Flash es un software originalmente diseñado para crear animaciones que se pueden usar en páginas web. Usa vectores gráficos que son ideales para la web, porque son ligeros y no consumen muchos recursos.

En las nuevas versiones de Flash, es más controlable, usando nuevas herramientas de programación, lo que lo tiene posicionado ya como un competidor para el HTML, en la creación de sitios web interactivos.

En pocas palabras se puede decir, que Flash es un programa que bien utilizado, puede:

- Producir cuñas, animaciones o películas completas destinadas a la Web, es decir, con la máxima compresión.
- Incluir fácilmente en las películas archivos de sonido con la tecnología más moderna (MP3, etc.) y comprimirlos conjuntamente con los fotogramas.
- Incluir la máxima interactividad en las páginas.
- Producir proyectores ejecutables standalone para que las películas se puedan ver sin necesidad de un navegador, o sea, desde un CD-ROM.
- Producir cambios en vivo (*in real time*) dentro de los elementos de la película.
- Conectar con las bases de datos de los servidores, para la actualización en tiempo real de los datos incluidos en la película (cotizaciones de Bolsa, etc.).
- Conectar con otras tecnologías para incluir en las películas datos que les sean propios, como las páginas ASP, JSP, etc.
- Conectar con las aplicaciones que se ejecutan en los servidores de la Red de redes. (15)

De todas las tecnologías anteriormente expuestas se utilizan en el proyecto CICPC:

- CSS, porque se utiliza para definir el estilo y apariencia de todos los contenidos de las páginas, o sea permite definir, mediante una sintaxis especial, la forma de presentación que se le aplicará a un sitio completo, de modo que se puede definir la apariencia de toda la aplicación de una sola vez; a una página, a una porción de esta o a una etiqueta en concreto, además consigue que la aplicación se vea igual en todos los navegadores. CSS es muy útil para separar el contenido del diseño, siendo esto muy ventajoso cuando se

quiere cambiar un aspecto del diseño de un sitio web, porque sin hojas de estilo se tendría que cambiar página a página dicho aspecto; sin embargo, cuando se han definido estas, se puede cambiar dicho aspecto modificándola a ella únicamente, con lo que se logra, además de un ahorro de tiempo, una mayor uniformidad en el diseño. El uso de CSS lleva a que el documento que se construye sea más respetuoso con los estándares de desarrollo web, lo que implica que sea más funcional en los diferentes navegadores, el código será más sencillo y se tendrá una documentación más precisa para trabajar dicho documento. En resumen, accesibilidad, rapidez de carga, reducción de tráfico de datos, etiquetado semántico y control centralizado del aspecto de toda la aplicación web, son algunas de las ventajas que aporta el uso de CSS, las que han validado su elección como una de las tecnologías a utilizar.

- Javascript para la validación en la entrada de datos, pues es seguro y fiable porque está en claro y hay que interpretarlo, por lo que puede ser filtrado, y la seguridad es casi total. Además como el código Javascript se ejecuta en el cliente, el servidor no es solicitado más de lo debido; un script ejecutado en el servidor, sin embargo, sometería a éste a dura prueba y los servidores de capacidades más limitadas podrían resentir de una continua solicitud por un mayor número de usuarios. En general se escoge Java Script porque no requiere un tiempo de compilación, los scripts se pueden desarrollar en un período de tiempo relativamente corto, las características de interfaz como por ejemplo, cuadro de diálogo, formularios y otros elementos de la UI son gestionados por el navegador y por el código HTML, por lo que los programadores no se deben preocupar en crear o controlar dichos elementos, además los programas escritos en Javascript son independientes de la plataforma hardware o sistema operativo, tienden a ser pequeños y compactos, no requieren mucha memoria ni tiempo adicional de transmisión. Además, al incluirse dentro de las mismas páginas HTML se reduce el número de accesos independientes a la red.
- Ajax para tratar las peticiones al servidor pues es una de las más novedosas tecnologías, o combinación de ellas, para crear aplicaciones web interactivas. Con el uso de esta se reducen los tiempos de respuesta entre la petición del usuario y la respuesta del servidor, pues al hacer un pedido al servidor, no se envía toda la página, lo que trae como consecuencia que el tráfico al servidor se reduzca, además la experiencia de usuario en la navegación es mucho más rica, pues ya no se refresca la página constantemente al interactuar con ella, también se reduce el tamaño de la información intercambiada.

- JSF para la implementación de los componentes en la capa de presentación. Se ha seleccionado este framework, porque es el estándar orientado a componentes, el código es mucho más independiente y permite hacer la presentación de una página con solo arrastrar los componentes ignorando el código HTML que está por debajo, haciendo más fácil y rápido el trabajo de los programadores. JSF trata la vista (la interfaz de usuario) de una forma algo diferente a lo que se está acostumbrado en aplicaciones web. Sería más similar al estilo de Swing, Visual Basic o Delphi, donde la programación de la interfaz se hace a través de componentes y basada en eventos. JSF es muy flexible y sencillo. En general algunas ventajas de JSF serían: el soporte de JSF en Entornos de Desarrollo Integrados (IDEs) como Eclipse, Netbeans, etc. es mucho mejor; constantemente se crean nuevos componentes JSF; JSF es parte de Java EE; todos los servidores de aplicaciones, por tanto, incluyen JSF.

1.5 Herramientas útiles para la elaboración de UI

En el desarrollo de este trabajo el primer paso consistió en analizar, obviamente, lo que se quiere lograr y a partir de ahí se comienza a investigar mediante Internet, libros y personas con cierta experiencia en desarrollo de software de gestión. Por esa razón se hace necesario buscar herramientas con determinadas características que hagan posible que el desarrollo de las UI esté a la altura del software, que sean optimas y lo más funcional posible.

1.5.1 Dreamweaver

Adobe Dreamweaver es una aplicación en forma de estudio (Basada en la forma de estudio de Adobe Flash) destinado para la edición de páginas web, creado inicialmente por Macromedia, actualmente es propiedad de Adobe Systems. Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades y su integración con otras herramientas. Hasta la versión MX, fue duramente criticado por su escaso soporte de los estándares de la web, ya que el código que generaba era con frecuencia sólo válido para Internet Explorer, y no validaba como HTML estándar. Esto se ha ido corrigiendo en las versiones recientes.

Dreamweaver ha tenido un gran éxito desde finales de los 90 y actualmente mantiene el 90% del mercado de editores HTML. Esta aplicación está disponible tanto para la plataforma MAC como Windows. Permite al usuario utilizar la mayoría de los navegadores Web instalados en su computadora para previsualizar las páginas web. También dispone de herramientas de

administración de sitios dirigidas a principiantes como, por ejemplo, la habilidad de encontrar y reemplazar líneas de texto y código por cualquier tipo de parámetro especificado, hasta el sitio web completo. El panel de comportamientos también permite crear JavaScript básico sin conocimientos de código.

Un aspecto de alta consideración de Dreamweaver es su arquitectura extensible. Es decir, permite el uso de "Extensiones". Las extensiones, tal y como se conocen, son pequeños programas, que cualquier desarrollador web puede escribir (normalmente en HTML y Javascript) y que cualquiera puede descargar e instalar, ofreciendo así funcionalidades añadidas a la aplicación. (16)

1.5.2 Eclipse

Eclipse es un IDE que en un principio fue diseñado y desarrollado por IBM y que luego fue lanzada a la comunidad de software libre. Es una plataforma de herramientas universal y portable que proporciona un marco de trabajo o framework para desarrollar aplicaciones y herramientas. El entorno de trabajo presenta un entorno de desarrollo integrado en perspectivas personalizables. Las perspectivas son combinaciones formadas por vistas y editores que muestran los diversos aspectos de los recursos del proyecto y están organizados por el rol o la tarea del desarrollador.

Eclipse es una plataforma de software cuya característica clave es la extensibilidad. Eclipse es una gran estructura formada por un núcleo y muchos plug-ins que van conformando la funcionalidad final. La forma en que los plug-ins interactúan es mediante interfaces o puntos de extensión; así, las nuevas aportaciones se integran sin dificultad ni conflictos. El Proyecto Eclipse es un proyecto de desarrollo de software de código abierto dedicado a proporcionar una plataforma industrial robusta, con amplias características y con calidad comercial para el desarrollo de herramientas altamente integradas. Además permite integrar diferentes aplicaciones para construir un entorno integrado de desarrollo. (17)

Plug-in: Es la mínima unidad de la plataforma que puede ser desarrollado por separado y que le aporta una nueva funcionalidad.

1.5.3 Adobe Flash

Adobe Flash Es una aplicación en forma de estudio que trabaja sobre un "escenario" y sobre "fotogramas" destinado a la producción de animación. Es actualmente escrito y distribuido por

Adobe Systems, y utiliza gráficos vectoriales e imágenes rasters, sonido, código de programa, flujo de vídeo y audio bidireccional (el flujo de subida sólo está disponible si se usa conjuntamente con Macromedia Flash Communication Server). En sentido estricto, Flash es el entorno y Flash Player es el programa de máquina virtual utilizado para ejecutar los archivos generados con Flash.

Los archivos de Flash, que tienen generalmente la extensión de archivo SWF, pueden aparecer en una página web para ser vista en un navegador, o pueden ser reproducidos independientemente por un reproductor Flash. Los archivos de Flash aparecen muy a menudo como animaciones en páginas Web y sitios Web multimedia, y más recientemente Aplicaciones de Internet Ricas. Son también ampliamente utilizados en anuncios de la web. En versiones anteriores, Macromedia amplió a Flash más allá de las animaciones simples, convirtiéndolo en una herramienta de desarrollo completa, para crear principalmente elementos multimedia e interactivos para Internet.

Fue hasta 2005 perteneciente a la empresa Macromedia conocido hasta entonces como Macromedia Flash y adquirido por Adobe Systems (desde entonces conocido como Adobe Flash) ampliando con ello su portafolio de productos dentro del mercado. (15)

Evidentemente con la herramienta que se utiliza en el proyecto CICPC, Eclipse, se puede lograr todo lo mencionado anteriormente por las potencialidades que brinda para la programación en JAVA.

1.6 Análisis de otras soluciones existentes

Con el tiempo la tecnología fue evolucionando y el mercado con ella; se fueron incorporando innovaciones una de ellas fueron las UI, que supusieron una mejora abismal para los usuarios al interactuar con los software. Este hecho supuso que cualquier persona podía usar una computadora sin necesidad de tener grandes conocimientos. En la actualidad las UI son más ricas visualmente, mucho más expresivas y más sencillas de utilizar. En cualquier aplicación, un mal diseño de la UI puede echar por tierra todo el trabajo que se haya realizado. Es la parte más cercana al usuario y, realmente, es la aplicación para el usuario. Cualquier mejora o error, será detectada de forma inmediata por el usuario por lo que su correcto diseño es vital para que el usuario pueda evaluar de forma positiva cualquier aplicación.

Entre las soluciones para el diseño y desarrollo de UI más destacadas hasta el momento se pueden nombrar:

- Interfaces de Usuario inteligentes
- Metodología y Técnicas en Proyectos software para la Web
- Diseño Centrado en el Usuario
- Propuesta de adaptación de la Metodología de Diseño Centrado en el Usuario para el desarrollo de Sitios Web Accesibles
- Diseño de Interfaces y Usabilidad: cómo hacer productos más útiles, eficientes y seductores

A continuación, se da una breve descripción de dichas metodologías. La idea es poder observar las similitudes entre las mismas y aquellas bases que unifican los criterios con los que fueron creadas.

Interfaces de Usuario inteligentes

El concepto clave en el funcionamiento de este tipo de interfaces es el de “modelo de usuario”. A través del mismo se pretende sintetizar las características y habilidades de un grupo de personas con el fin de facilitar y mejorar la interacción con el sistema. La adaptabilidad del mismo se logra interpretando las acciones del usuario, según sus opciones y el historial del diálogo con el sistema, y generando respuestas tanto a nivel lógico como físico. El proceso no es sencillo: no existe el “usuario promedio”, el conocimiento del usuario no es estático y somos incapaces de crear modelos precisos. Pero podemos desarrollar un modelo razonablemente valioso que nos permita mejorar la adaptabilidad de este tipo de interfaces.

Según expertos en el tema, el diseño de este tipo de interfaz debería seguir una metodología que le asegure el logro de los objetivos y beneficios esperados. La misma se compone de los siguientes elementos:

- Análisis de requerimientos
- Análisis de Viabilidad
- Diseño
- Construcción
- Evaluación posterior

La metodología propuesta es sólo un ejemplo de otras tantas posibles. Su utilidad y eficacia podría ponerse en duda, pero la necesidad de disponer de nuevos métodos que permitan mejorar el diseño y desarrollo de este tipo de herramientas es una realidad indiscutible.

Todas estas metodologías centran su atención casi únicamente en la adaptabilidad de los sistemas a los usuarios; es evidente la importancia que tiene esto, sobre todo en cuanto a la parte de facilitar la interacción del usuario con el sistema. Se pueden considerar algunos aspectos, pero no es exactamente la metodología que se necesita, pues no es el objetivo fundamental de este trabajo construir un sistema inteligente que sea capaz de adaptarse a cada tipo de usuario, pues el SIIPOL es un sistema que será utilizado por grupos de usuarios bien específicos. Este tipo de metodologías son aplicables a Sistemas de Inteligencia Artificial.
(18)

Metodología y Técnicas en Proyectos software para la Web

Metodología realizada por la Dra. María José Escalona Cuaresma y el Dr. José Mariano González Romano como Programa de Doctorado Tecnología e Ingeniería del Software en la Universidad de Sevilla el Curso 2006/07. Tiene como objetivo fundamental desarrollar o mejorar la seguridad, utilidad, efectividad, eficiencia y usabilidad de sistemas de software, para lograrlo los doctores plantean que para hacer sistemas usables es preciso:

Comprender los factores (psicológicos, ergonómicos, organizativos y sociales) que determinan cómo las personas trabajan y hacen uso de las computadoras. Desarrollar herramientas y técnicas para ayudar a los diseñadores de sistemas interactivos que son aquellos sistema que dependen de las acciones de un usuario para realizar una tarea, es decir, todo sistema en el que interactúan persona y máquina. Conseguir una interacción eficiente, efectiva y segura. Los usuarios no han de cambiar radicalmente su manera de ser, sino que los sistemas han de ser diseñados para satisfacer los requisitos del usuario.

Los sistemas interactivos se caracterizan por la importancia del diálogo con el usuario. Se implica al mismo tanto como sea posible, incluso incluyéndolo en el equipo de diseño y desarrollo. La interfaz es una parte fundamental en el proceso de desarrollo y debe tenerse en cuenta desde el principio. Se realizan prototipos, escenarios o maquetas para que el usuario pueda ir evaluando el diseño durante el ciclo de vida del proyecto.

A manera de conclusión esta metodología explica que no se debe realizar la especificación, diseñar las funciones y estructuras de datos y escribir el código y una vez casi terminado el

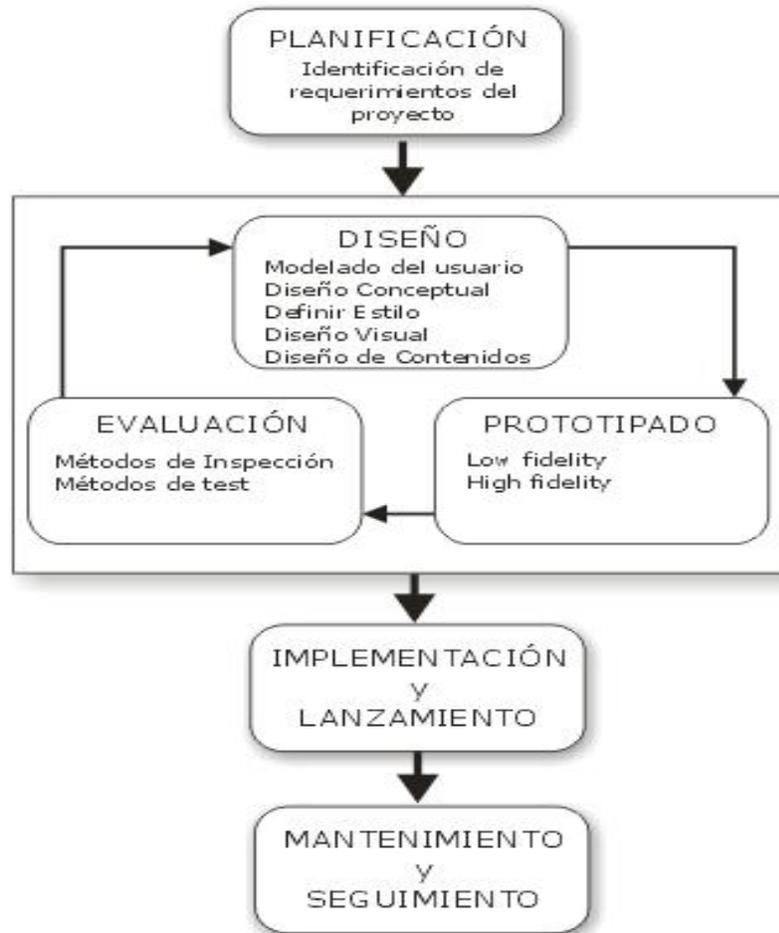
proceso de desarrollo de la aplicación plantearse el diseño de la interfaz de usuario. De esta forma se obtienen interfaces muy dependientes del diseño de los datos y las funciones, sin tener en cuenta al usuario que es quien va a utilizarlos e interactuar con ellos. (19)

Diseño Centrado en el Usuario

La Metodología Diseño Centrado en el Usuario (DCU) es la metodología de Diseño que se acepta y aplica en el mundo para la investigación, enseñanza y desarrollo del Diseño para Sitios Web. Considera el estudio y aplicación de las disciplinas de la Arquitectura de la Información, la Usabilidad y el Diseño de Interfaz de Usuario, en una dinámica de comprensión y aplicación de la investigación del comportamiento y uso de las interfaces desde la perspectiva del Usuario.

Los objetivos generales que persigue esta metodología son: La comprensión y aplicación de las metodologías para hacer Diseño de Interfaces en la Web, desde la perspectiva del usuario final. La comprensión y aplicación de todo el proceso productivo para el Diseño de un sitio web, incluida la manutención de los servidores que alojan las piezas de diseño elaboradas.

El Diseño Web Centrado en el Usuario se caracteriza por asumir que todo el proceso de diseño y desarrollo del sitio web debe estar conducido por el usuario, sus necesidades, características y objetivos. Centrar el diseño en sus usuarios (en oposición a centrarlo en las posibilidades tecnológicas o en los diseñadores) implica involucrar desde el comienzo a los usuarios en el proceso de desarrollo del sitio; conocer cómo son, qué necesitan, para qué usan el sitio; testar el sitio con los propios usuarios; investigar cómo reaccionan ante el diseño, cómo es su experiencia de uso; e innovar siempre con el objetivo claro de mejorar la experiencia del usuario. El proceso de Diseño Web Centrado en el Usuario se divide en varias fases o etapas, algunas de las cuales tienen carácter iterativo. Sirva como aproximación el siguiente esquema:



Como indica el esquema, las fases de "diseño", "prototipado" y "evaluación" son cíclicas e iterativas. Esto quiere decir que todo lo que se diseñe debe ser constantemente evaluado a través de su prototipado, para así poder corregir errores de usabilidad desde los primeros momentos del desarrollo. Evaluar el sitio web únicamente una vez finalizado su desarrollo haría mucho más costosa la reparación de errores de usabilidad, ya que siempre es más económico reconducir un diseño que rediseñar completamente el sitio.

El Diseño Web Centrado en el Usuario es un marco metodológico y una filosofía de diseño que describe, a grandes rasgos, cómo diseñar sitios web usables a través de la aplicación de técnicas, recomendaciones de diseño, métodos y procedimientos de Diseño Centrado en el Usuario. (20)

Propuesta de adaptación de la Metodología de Diseño Centrado en el Usuario para el desarrollo de Sitios Web Accesibles

Como su nombre lo indica es una adaptación metodológica para el desarrollo de sitios web accesibles. Basándose en la extensión de métodos y técnicas del marco metodológico de Diseño Centrado en el Usuario con la intención de satisfacer las necesidades de un mayor rango de usuarios que aquellos representados por el “usuario medio”, involucrando para ello a usuarios discapacitados o en contextos de uso desfavorables en el proceso de desarrollo.

La principal diferencia entre la presente propuesta de adaptación metodológica respecto a otros modelos, marcos metodológicos o adaptaciones basadas en Diseño Inclusivo, se fundamenta básicamente en una característica, su aplicabilidad. Por un lado se trata de un modelo muy sencillo, basado en la adaptación de técnicas y métodos de Diseño Centrado en el Usuario para la inclusión de más tipos de usuarios en más contextos de uso, por lo que se establece una familiaridad a priori entre desarrolladores y modelo, ya que estos métodos son ampliamente conocidos por el entorno de profesionales del diseño web. Por otro, en esta propuesta de adaptación metodológica no se excluyen soluciones basadas en un único interfaz común para todos los usuarios, ya que se entiende que no siempre resulta viable económicamente el desarrollo de múltiples interfaces o interfaces adaptables.

Como parte de las estrategias y metodologías de diseño que se plantean en esta propuesta se encuentra un concepto controvertido como estrategia para mejorar la accesibilidad de los sitios web con el fin de que sean usables por el máximo número de personas posibles, sin necesidad de adaptación o diseño especializado. De igual manera se pone en duda que en sitios web un diseño común para todos los usuarios sea la mejor decisión para acabar con las barreras de accesibilidad, ya que resultaría más eficaz la adaptación dinámica de la interfaz al usuario según sus propias necesidades y características.

Más allá de estrategias generales de diseño, los desarrolladores web necesitan de metodologías específicas mediante las cuales diseñar productos web usables y accesibles. El Diseño Centrado en el Usuario es un conjunto metodológico en el que se asume que todo el proceso de diseño debe estar conducido por el usuario sus necesidades, objetivos y características. Esta adaptación no se trata de un modelo metodológico que en sí mismo permita guiar al diseñador en el desarrollo de interfaces para sitios web usables y accesibles.

(21)

Diseño de Interfaces y Usabilidad: cómo hacer productos más útiles, eficientes y seductores

El diseño de interfaces es una disciplina que estudia y trata de poner en práctica procesos orientados a construir la interfaz más usable posible, dadas ciertas condiciones de entorno.

El entorno dentro del cual se inscribe el diseño de una interfaz y la medida de su usabilidad, está dado por tres factores:



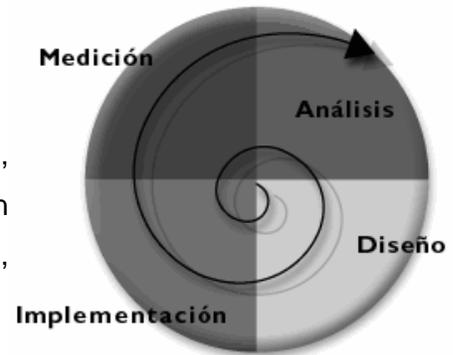
1. Una **persona**.
2. Una **tarea**.
3. Un **contexto**.

Diseño iterativo.

El diseño iterativo de interfaces es un proceso independiente de las técnicas utilizadas para llevarlo a cabo. Actualmente, el proceso del desarrollo de una interfaz se concibe como un ciclo que consta de 4 etapas, en varios niveles:

1. **Diseño**
2. **Implementación**
3. **Medición**
4. **Evaluación**

El resultado de cada etapa es la alimentación de la que sigue, incluso el de la última. Los resultados de la etapa de evaluación se toman para re-diseñar la interfaz, implementarla nuevamente, medir, y así sucesivamente.



Debido a esa repetición o auto-alimentación se lo llama diseño iterativo.

Las etapas y tareas del ciclo.

Diseño	<ul style="list-style-type: none"> • Análisis de requerimientos del producto. • Análisis de las tareas. • Conocimiento del usuario. • Generación de posibles metáforas y análisis de tipo de diálogo. • Revisión de posibilidades para la implementación.
Implementación	<ul style="list-style-type: none"> • Generación de prototipos (profundos o amplios, para investigación general o de ajustes).

	<ul style="list-style-type: none"> • Desarrollo de la aplicación, sitio o sistema.
<p>Medición (Test de usabilidad)</p>	<ul style="list-style-type: none"> • Planificación (desarrollo del plan, definición de las medidas, selección de participantes, formación de observadores, preparación de los materiales). • Test (prueba piloto, tests con usuarios).
<p>Evaluación</p>	<ul style="list-style-type: none"> • Conclusión (análisis de los datos, elaboración del informe, resultados y recomendaciones). • Comparación contra estándares (internos y/o externos), versiones anteriores del mismo producto y productos competidores. • Verificación de las diferencias. • Generación de nuevas metas.

Además de la recursividad, otra característica del enfoque actual del diseño de interfaces es que involucra no sólo a los especialistas en usabilidad o diseño, sino a todo el equipo de desarrollo que está constituido por todos aquellos que participan de alguna manera en desarrollo o comercialización del sistema o herramienta: gente de marketing, comunicación, documentación, sistemas e informática, diseño y usabilidad, etc. Cada uno tiene conocimiento acerca de un área específica, y su participación a lo largo del desarrollo aumenta las probabilidades de éxito.

Una de las claves más importantes para articular un buen proceso de diseño de interfaz y así aumentar la usabilidad del producto resultante, es comenzar con el ciclo de diseño iterativo lo más temprano posible. Cuanto antes se comience, hay una menor probabilidad de que se llegue a la versión pública con errores importantes y más tiempo para mejorar aquellas características que puedan ser mejorables. Además, es mucho más rápido y barato modificar prototipos que hacer un cambio en un producto avanzado o ya desarrollado. Respecto de la implementación, ésta debe estar tan “despegada” de la interfaz como sea posible para permitir cambios a medida que los resultados de los tests dictan las modificaciones. (22)

Existen otras propuestas que han aunado aspectos importantes de las UI pero sin cubrir todo el proceso de desarrollo de las mismas. En nuestro país la creación de UI aún no cuenta con un patrón descriptivo del proceso. Para realizar el montaje de UI durante el desarrollo de un software las empresas dedicadas a esta labor han formulado sus propias inventivas centradas en los sistemas específicos que se desean construir o abogando por una generalidad muy ambigua en el intento de generalizar la aplicación de sus propuestas a otros sistemas.

1.6.1 Antecedentes en la UCI

En la UCI no existe definida metodología alguna que estandarice, organice o explique la manera en que debe realizarse el diseño y desarrollo de las UI para los proyectos productivos. Según las características de cada proyecto el montaje de UI se hace de manera diferente en cada uno de ellos. Tomando como base el proyecto Identidad desarrollado por la Facultad 1 y el proyecto Prisiones desarrollado en la Facultad 4 se puede afirmar que no existe dominio de métodos certeros para llevar a cabo la realización de las UI de los proyectos.

A pesar de que en el proyecto Identidad el software Sistema Automatizado de Identificación, Migración y Extranjería (SAIME) no presenta mucha similitud con el software SIIPOL del proyecto CICPC por ser una aplicación de escritorio, cabe destacar que en el mismo no existen personas especializadas o al menos capacitadas para diseñar y desarrollar UI, en el mejor de los casos las UI que se obtienen son dibujadas por los analistas en una hoja de papel que le es entregada a los programadores para que procedan a su implementación, generalmente son estos últimos quienes diseñan las UI, puede que exista algún programador con conocimientos básicos sobre UI pero difícilmente dominan las características esenciales y principios fundamentales que lleva ligado un buen diseño de UI, de este depende el éxito del producto final.

En el caso particular del proyecto Prisiones que es uno de los proyectos con características similares al proyecto CICPC, para la construcción del Sistema de Gestión Penitenciaria (SIGEP) no existe un equipo capacitado para el desarrollo de las UI, pero al menos una persona se dedica a esa tarea en específico. Se realiza todo el montaje de las UI utilizando HTML, haciendo algún uso de tecnología Ajax. Se hace muy difícil ajustarse a las pautas de diseño que en ocasiones no pueden cumplirse como está establecido por el diseñador del proyecto. El prototipo no funcional que se les entrega a los programadores no tiene la calidad requerida, por lo que es de esperar que al culminar la etapa de implementación las UI funcionales que componen el software tampoco la tendrán.

Se hace imprescindible destacar que si se quiere obtener productos de alta calidad y aceptación por los usuarios, todas las actividades que se llevan a cabo durante del proceso de desarrollo del software deben hacerse de forma correcta y para cada una de ellas deben existir metodologías, procedimientos o guías que indiquen cómo deben realizarse. Los esfuerzos de este trabajo estarán centrados en describir una metodología para el diseño y desarrollo de UI del SIIPOL, estudiar su aplicabilidad en el desarrollo del software y especificar la información

necesaria para utilizarla en la práctica, observar sus fuerzas y debilidades y analizar limitaciones y alcances de la misma.

1.7 Conclusiones

Se ha logrado en este capítulo establecer la base teórica necesaria para facilitar la comprensión de la propuesta que se mostrará en el siguiente capítulo.

Se concluye que las soluciones existentes para el proceso de diseño y desarrollo de UI no son totalmente aplicables al SIIPOL, aunque se deben tomar en cuenta algunos aspectos incluidos en estas. Finalmente se escogieron las herramientas y tecnologías que se utilizarán en el proceso de diseño y desarrollo de las UI del proyecto considerando las características del mismo.



Metodología para el diseño y desarrollo de UI del SIIPOL

2.1 Introducción

En el modelo de interfaz de usuario existen tres puntos de vista: usuario, programador y diseñador. Cada uno tiene su propio esquema mental de la interfaz, generado a través de su experiencia. El modelo de usuario tiene su propia forma de ver las cosas, y espera de ellas un comportamiento predecible. El modelo del programador, normalmente no tienen en cuenta la forma de pensar del usuario y sí su perspectiva de la computadora, los objetos que maneja son distintos de los del usuario y deben ser ocultados al mismo. El modelo del diseñador mezcla las necesidades, ideas, deseos del usuario y los materiales de los que dispone el programador para diseñar un producto software. La idea de la solución que se propone es unificar dichos puntos de vista, y lograr que en el proceso de diseño y desarrollo de UI no se vean como entes separados, sino que funcionen de forma integrada.

En el presente capítulo se describe la propuesta de solución. Primeramente se definen términos preliminares para facilitar el entendimiento de la propuesta, se especifican y describen aspectos fundamentales de la metodología, así como las fases o etapas que presenta la misma.

2.2 Presentación de DDWUI

El propósito de esta tesis es construir una metodología que reúna buenas prácticas dentro del proceso de diseño y desarrollo de UI adaptándolas a la plataforma de desarrollo usada en el proyecto CICPC.

Se puede argumentar que la interfaz de usuario es un elemento de gran importancia en un producto electrónico. Si el software es difícil de utilizar, si se necesita de conocimiento especializado sobre el tema para manejarlo, si conduce a cometer errores, o causa frustración para conseguir los objetivos, no será de agrado, independientemente de la potencia informática que demuestre o de la funcionalidad que ofrezca. Dado que la interfaz condiciona la percepción del software por parte del usuario, tiene que estar bien diseñada, estandarizada y construida, por esta razón es objetivo de esta metodología ofrecer una guía para el proceso de montaje de las UI que parte desde las bases para lograr un buen diseño, hasta llegar al desarrollo eficiente

y funcional de las interfaces. El nombre de la metodología es *DDWUI* por sus siglas en inglés Design and Development of Web User's Interface (Diseño y Desarrollo de Interfaces Web de Usuario).

Objetivos Específicos de la Metodología DDWUI

- Establecer pautas de Diseño Gráfico (DG) y Arquitectura de Información (AI) para la aplicación.
- Crear el soporte que cumpla con las pautas de DG y AI orientado a la plataforma de desarrollo usada en el proyecto.
- Brindar al programador interfaces funcionales que cumplan con las pautas.

La Metodología DDWUI consta de tres partes fundamentales. Primero, “Definiciones preliminares”, en esta parte se explican algunos términos que tienen un significado específico dentro de la metodología y son utilizados a lo largo de la misma. Segundo, “Aspectos esenciales”, son indicaciones importantes que se proponen para complementar la metodología y que no deben obviarse si se desea alcanzar el éxito. Tercero, “Fases de la Metodología”, esta parte constituye el fundamento de la metodología y es aquí donde se realiza una descripción detallada de las diferentes etapas por las que transita la propuesta.

2.3 Definiciones preliminares

Como primer paso importante se hace imprescindible precisar algunos términos que se utilizarán a lo largo de la metodología para facilitar una mejor comprensión:

Montadores: Personas encargadas de realizar el montaje (diseño y desarrollo) de las UI.

Caso de estudio: Es un proyecto sencillo que se realiza con el fin de ejercitar los conocimientos aprendidos durante la capacitación y de probar lo que se va a hacer en el proyecto real.

Artefactos: Entregables que se obtienen al terminar cada una de las fases, estos pueden ser documentos, imágenes, hojas de estilo, páginas web.

Actividades: Conjunto de tareas o acciones que se llevan a cabo dentro de cada una de las fases y que guían el proceso de diseño y desarrollo de las interfaces.

Personas responsables: Son los integrantes del equipo de desarrollo encargados de realizar las actividades y crear o modificar los artefactos en cada una de las fases.

Sistema: Es el producto electrónico que se construye para satisfacer las necesidades de un cliente expresadas como requisitos.

Proceso de Negocio: conjunto de acciones o actividades relacionadas lógicamente, llevadas a cabo para transformar información. Cada proceso de negocio tiene sus artefactos de entradas y salidas, personas responsables y mecanismos de control.

Requisitos: Condición o capacidad que tiene que ser alcanzada o poseída por un sistema y que necesita un usuario para resolver un problema o lograr un objetivo.

Caso de Uso (CU): Fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante, define el comportamiento del sistema desde el punto de vista del usuario y establece un acuerdo entre clientes y desarrolladores sobre los requisitos que debe cumplir el sistema.

Prototipo: Un prototipo es una representación visual, es decir una imagen de una UI del sistema.

Arquitectura de la Información (AI): Es la disciplina encargada de la fundamentación, análisis, planificación y estudio de la disposición de los datos contenidos en los sistemas informáticos interactivos.

Recursos de montaje: Artefactos elaborados para ser utilizados posteriormente en el montaje de las interfaces, que facilitan este trabajo:

Escritorio de trabajo: Es un archivo en el que se establece la estructura y organización que tendrán las interfaces de la aplicación.

Hoja de estilo (CSS): Es un archivo en el que se especifica una colección de reglas de formato que definen la apariencia que tendrán los contenidos de las interfaces de la aplicación.

2.4 Aspectos esenciales

El secreto de aplicaciones web de calidad está en la especialización de los equipos que trabajan en ello y en la metodología empleada. Para el usuario, cada vez más la interfaz es el sistema, es por esta razón se hace imprescindible definir la estructura y perfiles que debe tener un equipo dedicado a las UI. Un equipo de diseño de UI debe estar compuesto por personal

especialista que debe tener en cuenta tanto las necesidades y requisitos del usuario final, como conocer y aplicar los estándares y pautas de diseño y desarrollo web definidos, y además adapten su trabajo al desarrollo del producto aportando soluciones y dando soporte continuado durante las fases subsiguientes en el desarrollo del software. A continuación se hace una descripción detallada de los pasos que se deben llevar a cabo para estructurar y mantener el equipo, que se denomina en lo adelante, equipo de montadores UI.

2.4.1 Crear equipo de montadores UI

Es importante que dentro del equipo de desarrollo del software se disponga de un grupo de personas que se especialicen solamente en el trabajo con las UI, que serán los montadores UI, nuevo rol dentro de equipo de desarrollo del proyecto. Se debe hacer una selección de personas que tengan conocimientos, aptitudes y se inclinen hacia el diseño, o sea, que prefieran el trabajo con gráficos, colores e interfaces. A partir de este momento queda conformado el equipo de montadores UI del proyecto, responsable de la mayor parte de las tareas, acciones o actividades que involucren las UI, logrando, con esta especialización, que otros miembros del equipo de desarrollo, por ejemplo los desarrolladores, no se sobrecarguen de trabajo. Se habla de un pequeño grupo de desarrollo, de no más de 10 personas las cuales deben compartir un ambiente de trabajo común, con mínima separación física. Cabe mencionar que en caso de contar con proyectos de gran magnitud y una considerable cantidad de recursos se debería particionar en subgrupos de menor tamaño, a razón de 2 montadores por cada subsistema de desarrollo, trabajando en forma paralela.

2.4.2 Capacitar equipo de montadores UI

Una vez conformado el equipo se procede a su capacitación. Se deben impartir una serie de cursos, clases, conferencias, talleres, prácticas de laboratorio con el objetivo de proporcionar los conocimientos y habilidades necesarios para el posterior desempeño en las labores dentro del proyecto. Es conveniente ilustrar el uso de diferentes herramientas, tecnologías y lenguajes que serán usados para el diseño y desarrollo de las UI. Entre los cursos básicos que debe incluir esta capacitación se encuentran: Photoshop, Corel Draw, HTML, CSS, Javascript, Ajax, además se pueden incluir otros que se consideren útiles. Es responsabilidad de la dirección del proyecto buscar personas capacitadas que impartan dichos cursos con calidad, enfocados a las necesidades del proyecto.

2.4.3 Realizar casos de estudio

Se hace necesario familiarizar al equipo con las particularidades de la aplicación. Es por esta razón que antes de comenzar a trabajar en la realización del proyecto, es recomendable realizar uno o varios casos de estudio, que no son más que pequeños proyectos de entrenamiento, que servirán para poner en práctica todos los conocimientos y habilidades adquiridos en la etapa de capacitación y ensayar el funcionamiento del equipo con los medios y procedimientos que se utilizarán en el proyecto real. Se deben incluir todos los integrantes del equipo de montadores haciendo énfasis en el buen diseño y desarrollo de las UI.

2.5 Fases de la Metodología

La creación de cualquier producto involucra la ejecución de distintas etapas o fases. Al considerarse las UI de una aplicación una parte significativa dentro de la construcción de un producto, concreto y tangible, su creación también debe atravesar por distintas etapas. Es indispensable, además, organizar el trabajo, y haciendo un seguimiento lógico a las actividades, lograr que el diseño y desarrollo de las UI sea exitoso.

Normalmente en cualquier proceso de desarrollo de software, aplicando cualquier metodología, para llegar a la concepción del sistema, se llevan a cabo varias actividades fundamentales. Debido a que el objetivo de la presente investigación es definir una metodología para diseñar y desarrollar UI, solamente se hace una breve descripción de las actividades que se consideran necesarias para llegar a las UI.

La metodología que se presenta está comprendida en cinco fases. La primera, “Estudio y modelado del Sistema”, es la fase en la que se estudia el ambiente o entorno donde está enmarcado el negocio, se realiza el levantamiento de requisitos y se modela el sistema, es aquí donde se obtiene una primera representación de lo que serán las UI. Es en la segunda, “Arquitectura de la Información”, donde se hace un estudio de la audiencia para establecer la organización y la forma en que se muestran los contenidos dentro de las interfaces. En la tercera, “Diseño Gráfico”, se hace un boceto de las interfaces basado en principios y estándares de diseño web. La cuarta, “Elaboración de los recursos de montaje de interfaces”, es donde se construyen elementos que serán utilizados en el desarrollo de las UI. La quinta, “Desarrollo de las UI”, es en la que se pone a punto las interfaces, es decir se desarrolla y se dejan listas para entregar a los programadores, terminado con esta última fase el proceso de diseño y desarrollo de las UI. Dentro de cada una de las fases se especifican: objetivos, pues estos sirven para guiar el trabajo y determinar en qué medida estas se llevan a cabo

exitosamente; entradas y salidas, que son los artefactos que se obtienen al concluir una fase y se utilizan para comenzar otra; actividades, son las acciones que se realizan para dar cumplimiento a los objetivos; personas responsables, son las encargadas de realizar las actividades y construir los artefactos y herramientas que son utilizadas en la construcción de los artefactos. Dentro de las herramientas se incluyen además las alternativas existentes en Software Libre, por todas las ventajas que este ofrece y con el objetivo de hacer de DDWUI una metodología más aplicable.

Software Libre: es la denominación del software que brinda libertad a los usuarios sobre su producto adquirido y por tanto, una vez obtenido, puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

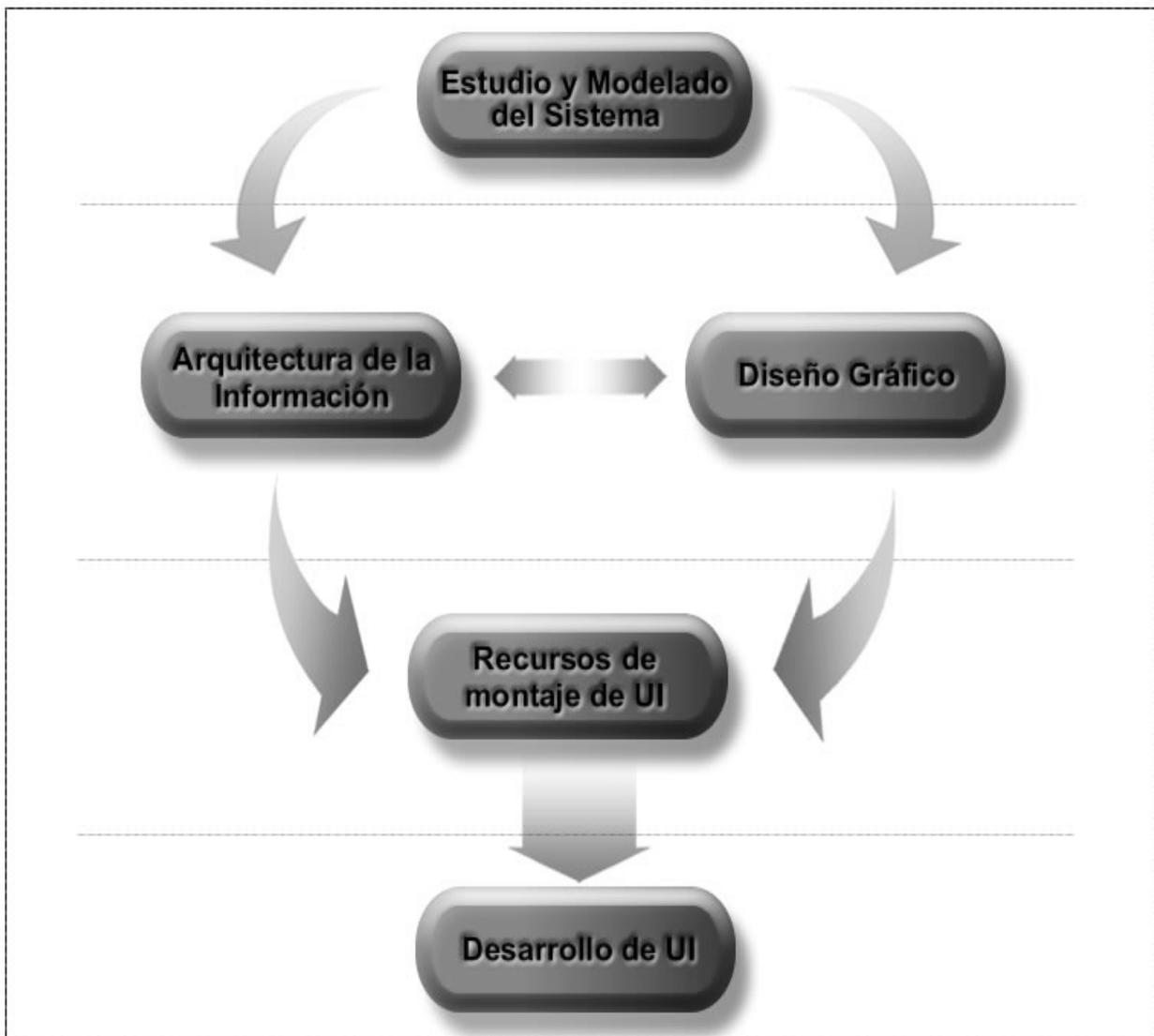


Fig. 01 Fases que componen DDWUI.

Como se muestra en la figura 01 en un primer momento se realiza el estudio y modelado del sistema, con los elementos y datos que se obtienen de esta fase se comienzan a desarrollar de forma paralela las fases de arquitectura de la información y diseño gráfico, aunque la de arquitectura puede iniciarse un poco antes que la de diseño, siempre deben ir a la par porque se complementan mutuamente. Es de dichas fases que se consiguen los elementos necesarios para comenzar la fase donde se construyen los recursos de montaje para las UI. Una vez concluida esta última fase, se inicia la de desarrollo de interfaces.

2.5.1 Estudio y modelado del Sistema

2.5.1.1 Objetivos

- Obtener las características del sistema a partir de la identificación de los procesos de negocio y los requisitos del cliente.
- Obtener especificaciones de casos de uso con prototipos de interfaces.

2.5.1.2 Entrada

- Ambiente o entorno donde está enmarcado el negocio.

2.5.1.3 Actividades

- Modelamiento del negocio.

El propósito del Modelamiento del negocio es comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema, comprender los problemas actuales de la organización e identificar las mejoras potenciales, derivar los requerimientos del sistema que va a soportar la organización y lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer. Dentro del Modelamiento del negocio se debe describir el negocio, o sea entender los procesos existentes u observados y la estructura de la organización con el propósito de comprenderlos, para ello se identifican los procesos de negocio, se especifican las actividades y personas involucradas mediante una secuencia de pasos lógicos y explorar la automatización que significa investigar qué partes del negocio pueden y deben ser automatizadas.

- Levantamiento de requisitos.

En este punto, se hace necesario comenzar a definir qué es lo que debe hacer el sistema, siendo necesario ir a la captura de los requisitos que este debe cumplir. Los objetivos del

levantamiento de requisitos son definir el ámbito del sistema, definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario, establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debe hacer, proveer a los desarrolladores de un mejor entendimiento de los requisitos del sistema, de una base para estimar recursos y tiempo de desarrollo del sistema. Todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema deben ser analizadas como candidatas a requisitos. Los requisitos deben ser especificados por escrito y posibles de probar o verificar.

Los requisitos se pueden clasificar en funcionales y no funcionales. Los funcionales son capacidades o condiciones que el sistema debe cumplir. Los no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto.

- **Concepción del sistema.**

El objetivo de la concepción de sistema es guiar a los desarrolladores en la construcción de sistemas que se ajusten a las necesidades de los clientes y traducir los requisitos a una especificación que describe cómo implementar el sistema. En esta parte se refinan y estructuran los requisitos obtenidos con anterioridad, construyendo el conjunto de artefactos que describen las funcionalidades que debe tener el sistema, lo que permite una mayor comprensión del problema para modelar la solución. En general consiste en obtener una visión del sistema que se preocupa de ver QUÉ hace y CÓMO cumple sus objetivos, debe ser suficiente para que pueda ser implementado sin ambigüedades.

2.5.1.4 Salida

- **Especificaciones de casos de uso.**

Es un documento que contiene la descripción del CU, la descripción puede ser elaborada de forma breve o extendida e incluye tantos caminos que conducen a resultados de éxito o no. Esta especificación contiene un prototipo inicial de UI, elaborado por los analistas.

2.5.1.5 Personas responsables

- **Analista.**

Es el responsable de definir el producto, al menos inicialmente. Su objetivo es obtener los requisitos de los usuarios finales y definir cómo quedarán en las UI. Lo anterior debe ser

plasmado en la documentación textual que incluye prototipo de UI, es decir el CU integrado.

2.5.1.6 Herramientas

- Microsoft Office Word.

Es el procesador de textos de la suite ofimática Microsoft Office creado por Microsoft. Su formato propietario DOC es considerado un estándar de facto, aunque en su más reciente versión, Word 2007 utiliza un nuevo formato basado en XML llamado .DOCX, pero también tiene la capacidad de guardar y abrir documentos en el formato DOC. Word ha llegado a ser el procesador de texto más popular.

- Open Office Writer.

Es un procesador de textos que forma parte del conjunto de aplicaciones libres de oficina Open Office. Soporta el formato propietario .doc de Microsoft Word casi en su totalidad, además de otros formatos clásicos de documentos. Su formato nativo de fichero esta en formato XML. Puede exportar a ficheros PDF nativamente sin usar programas intermedios. Es multiplataforma como la suite ofimática Open Office, que lo compone.

Esta primera etapa puede considerarse como la más importante puesto que los resultados que se obtengan serán la guía y el enfoque que se reflejará en toda la aplicación. Se obtiene una visión de qué es necesario hacer para dar respuesta a las solicitudes del usuario, lo cual se logra definiendo los procesos, roles y responsabilidades de la organización. Se establece una estrecha comunicación entre el cliente y el equipo de desarrollo orientada a las aspiraciones y objetivos del usuario para obtener una concepción del sistema.

2.5.2 Arquitectura de la Información

La AI se ocupa del diseño estructural de los sistemas de información. Su actividad central es la organización, recuperación y presentación de información, mediante el diseño de ambientes intuitivos. (23)

2.5.2.1 Objetivos

- Definir los contenidos que se incluirán en un producto.
- Establecer cómo se organizarán estos contenidos.
- Estandarizar las etiquetas para representar los contenidos.
- Organizar la información.

- Ayudar a crear un software que le sea fácil y agradable al usuario final.

2.5.2.2 Entrada

- Resultados de entrevistas o encuestas a los usuarios finales y a los usuarios internos, que son los que construyen el software.

A partir de estos resultados 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. Las entrevistas y encuestas son contactos personalizados con todos los usuarios, ya sea de manera oral o escrita. Con cada una se obtienen informaciones que se puede analizar de manera cualificada o cuantificada.

- Especificaciones de casos de uso.

2.5.2.3 Actividades

- Estudio de la audiencia y sus necesidades.

Como el éxito o el fracaso de una aplicación web depende de sus usuarios finales, se considera que para el lanzamiento de cualquier producto electrónico, debe realizarse en primer lugar un estudio de las necesidades de información y características de la audiencia o usuarios potenciales del sitio. Es importante recolectar información que sirva como referencia a las etapas subsiguientes en el proceso de desarrollo del software. Esta información se consulta y valida en las etapas que le suceden. Esta investigación preliminar genera una lista de posibles audiencias que deben agruparse según sus características comunes. Una vez agrupadas, debe establecerse un orden de prioridad o jerarquía de los grupos de audiencias a criterio y conveniencia del equipo de desarrollo. Deben enumerarse las necesidades más importantes de cada grupo. Estos datos serán de suma utilidad para la creación de alternativas de arquitecturas de información para el sitio en desarrollo.

- Organización de la Información.

La organización de la información es el proceso donde se dispone y ordena la secuencia de los elementos que integran el contenido de la aplicación. En este proceso, se consideran las características de los sistemas de clasificación y ordenamiento como son la ambigüedad, la heterogeneidad y la homogeneidad. También, se seleccionan los

esquemas de organización de la información y las estructuras de organización de la información que se utilizarán en la aplicación.

- Estructuración del sistema de navegación.

La razón para diseñar correctamente un Sistema de Navegación (SN) radica en prevenir que los usuarios puedan hallarse perdidos frente a un producto electrónico y experimenten sensaciones de confusión, frustración e ira entre otras. El buen diseño de un SN evitará la desorientación de los usuarios, es necesario para brindar un sentido del entorno y permitir flexibilidad en la navegación dentro del producto. Los SN pueden diseñarse de manera tal que apoyen el aprendizaje asociativo a través de la presentación de recursos relacionados con el contenido que se está mostrando. Debe prestarse especial atención al balance que debe existir entre la flexibilidad en la navegación y la inclusión de excesivas opciones que podrían confundir y desorientar al usuario. Los SN pueden conformarse por varios elementos tales como: gráficos, barras de menús horizontales y verticales y menús emergentes. Otros como las tablas de contenido y el mapa del sitio permiten el acceso remoto al contenido. Cada uno de estos elementos debe de estar implementado en cada página y la suma de todos ellos conforman el SN. (24)

- Elaboración del sistema de etiquetado.

Las etiquetas constituyen una forma de representación. Tal como se usan las palabras para representar pensamientos, las etiquetas permiten representar un conjunto de información en un sitio Web, nos permiten comunicarnos con los usuarios para mostrarles cómo está organizado el sitio y las posibilidades de navegación que tiene. Las etiquetas no son más que el nombre que se le va a asignar a cada área dentro de la UI, describen o designan los elementos que integran el sistema de navegación (Contáctenos, Descargar, Ayuda, etc.), por esto se debe considerar en sí como un sistema que interactúa con las demás entidades sistémicas que conforman la AI. (24)

- Diseño conceptual.

Los objetivos fundamentales de la etapa de diseño conceptual son específicamente:

- Ordenar todo el cúmulo de información que se ha recogido sobre el producto.
- Decidir cómo será organizada y etiquetada la información, así como qué tipos de sistemas de navegación se utilizarán y cómo se implementarán las búsquedas.
- Realizar bocetos y los mapas conceptuales del producto.

Los mapas conceptuales son un tipo de diagramas que suministran un lenguaje visual similar en sus características al texto del lenguaje natural en el que los textos, bloques de texto o nodos de cualquier morfología pueden someterse a relaciones sintácticas y semánticas, y la capacidad representativa puede establecerse tanto de una manera informal o formal. Constituyen una herramienta imprescindible para facilitar la navegación.

- Usabilidad.

La usabilidad debe considerarse un factor esencial que se debe tener muy en cuenta si se desea obtener resultados óptimos ante la necesidad de convertir una aplicación web en una herramienta flexible, amigable, eficaz y fácil de usar. Es importante que la facilidad de uso de una aplicación web está íntimamente ligada a la interacción con el usuario, siendo elementos fundamentales de valoración en este proceso aspectos como: eficacia, efectividad, utilidad, facilidad de uso y de aprendizaje, rendimiento y satisfacción generada. La usabilidad se refiere a la capacidad de un software o sistema interactivo de ser comprendido, aprendido, usado fácilmente, y ser atractivo para el usuario en condiciones específicas de uso. La aplicación de la usabilidad en UI se manifiesta realmente cuando nada funciona mal, no cuando todo funciona bien. El usuario simplemente debe poder realizar sus tareas, búsquedas y obtención de información, aprendizaje de la manera más espontánea e intuitiva posible.

- Accesibilidad.

La accesibilidad es la capacidad que tiene una aplicación de facilitar a cualquier usuario el acceso a la información, mediante el diseño de interfaces que consigan que dicho usuario reduzca el tiempo de asimilación y de aprendizaje de uso. La accesibilidad persigue también el objetivo de que ningún usuario este discriminado ante el acceso y asimilación de los contenidos. El uso de los estándares web, que son un conjunto de recomendaciones dadas por el W3C y otras organizaciones internacionales acerca de cómo crear e interpretar documentos basados en la web y el uso de CSS en la construcción de aplicaciones web ayuda mucho a lograr un alto nivel de accesibilidad. A través del cumplimiento de las líneas de accesibilidad se procura buscar confianza por parte de los usuarios, aumentando así la cantidad de personas que hacen uso de la aplicación.

- Diseño de prototipo de UI.

La etapa de diseño de prototipo se basa en la elaboración de modelos o prototipos de las UI de la aplicación, siguiendo pautas de AI a partir de la especificación del CU. Pueden

servir para evaluar la usabilidad del sitio y verificar si cumplen con los requisitos del cliente, una vez que así suceda, sirven de guía a los montadores UI para proceder con el diseño y desarrollo de las UI reales de la aplicación. La utilidad real del prototipo se fundamenta en que no tendría sentido comenzar a desarrollar las UI sin antes estar seguros de que el diseño es usable y está aprobado por el usuario final. Una vez que el arquitecto de información elabora el prototipo UI, lo entrega al analista para que lo incluya dentro de la especificación del CU correspondiente, de esta forma se obtiene la integración del CU.

2.5.2.4 Salida

- Documento de Pautas de Arquitectura e la Información.
Documento que rige y recoge las etapas de la AI, en el cual se plasma el etiquetado, la organización de la información, el mapa de navegación del software y el flujo de comunicación cliente-software. Todo ello para que los usuarios internos visualicen y plasmen en el producto lo que al usuario le interesa y de la manera más cómoda para él.
- Prototipo de UI.
El prototipo de las UI de la aplicación que se construye en este punto, por el arquitecto de información, da una visión de las interfaces diseñadas para cada CU teniendo en cuenta su especificación y las pautas de AI establecidas.
- Integración de casos de uso
Cuando el documento Especificación de casos de uso contiene sus respectivos prototipos de UI, se está en presencia de la integración de un CU.

2.5.2.5 Personas responsables

- Arquitecto de Información.
Diseñador de la presentación de la información para facilitar el entendimiento. Clarifica la misión y visión del sitio, equilibrando las necesidades del patrocinador y las necesidades de la audiencia. Determina el contenido y funcionalidad que el sitio va a tener. Especifica como los usuarios van a encontrar la información al definir su organización, navegación y sistemas de etiquetado. (23)

2.5.2.6 Herramientas

- Microsoft Office Word.

- Open Office Writer.

- Microsoft Office Visio 2003.

Microsoft Visio es un software de dibujo vectorial para Microsoft Windows. Las herramientas que lo componen permiten realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML, y más.

- KOffice Kivio.

Es una aplicación informática para generar diagramas y organigramas, incluida dentro de la suite ofimática para KDE (Entorno de Escritorio K) KOffice. Tiene un interfaz de usuario similar a Microsoft Visio y está completamente integrada a KOffice.

En esta fase es donde se establece la organización eficaz y eficiente de los contenidos y servicios disponibles en la aplicación, se puede definir la misión y la visión de la misma, centrándose en las necesidades de sus audiencias, generar estructuras que soporten el cambio y el crecimiento en el tiempo, determinar qué contenidos y funcionalidades deberá contener la aplicación y poner nombres entendibles a las áreas de información, crear sistemas de navegación intuitivos, con el fin de que la asimilación de contenidos por parte del usuario sea positiva.

2.5.3 Diseño Gráfico

2.5.3.1 Objetivos

- Elaborar las pautas de diseño gráfico para el desarrollo de las interfaces de la aplicación web.

2.5.3.2 Entrada

- Documento de arquitectura de la información (incluye):
 - ✓ Estructura de contenidos.
 - ✓ Mapa de Navegación (puede entregarse hecho o elaborarse arquitecto-diseñador).
- Comportamiento de la aplicación, especificidades en materia de funcionamiento, plataforma o soporte de implementación y sus limitantes técnicas para la implementación del diseño.

2.5.3.3 Actividades

- Evaluar la información obtenida de la fase de AI.

Una vez obtenida toda la información referente a la Arquitectura de Información y al comportamiento y características de la aplicación, se analiza y basándose en ella se definen los alcances del diseño.

- Evaluar homólogos (nacionales e internacionales).

Se realiza un estudio de aplicaciones similares, construidas con anterioridad, con el objetivo de tener una idea de las tendencias del diseño para ese tipo de aplicaciones.

- Definir atributos genéricos y específicos y su traducción a códigos visuales (elaborar metáfora).

- Diseño de las interfaces incluye:

- ✓ Analizar mapa de navegación.

Se evalúa la estructura de contenidos propuesta por los arquitectos de información y su correspondencia con el mapa de navegación. En caso de no existir el mapa se realiza una propuesta y se somete a consideración de la dirección del proyecto.

- ✓ Diagramación general y específica de las pantallas.

Se definen las áreas principales de la aplicación, lo que incluye cabezal, zona de menú, área de trabajo y en una segunda instancia, se incluye una definición más específica del área de trabajo: columnas donde se inscribirá el texto, fotos, y gráficos. Podrán combinarse el uso de las columnas de forma dinámica en una misma pantalla, aunque recomendamos mantener predominantemente una diagramación sobre otra para establecer un orden visual.

- ✓ Composición de los contenidos en pantalla.

Se estructuran los contenidos, estableciéndose niveles jerárquicos de contenidos y posición determinados por su importancia comunicativa y códigos culturales preestablecidos.

- ✓ Definir etiquetado para vínculos, hipervínculos, etiquetas de contenido.

Se definen las fuentes digitales, sus puntajes e interlineados, colores, logrando una integración dinámica entre los estándares web y las características de la aplicación.

- ✓ Comportamiento tipográfico para cuerpo de texto.

Se definen las tipografías a usar de acuerdo a sus connotaciones y funcionalidades. Se debe tener en cuenta si es para cuerpo de texto, debe ser ante todo, legible, en el caso de los titulares la función más que legible es estética.

✓ Comportamiento de menús.

Se define el comportamiento de los posibles menús a usar en la aplicación en dependencia del propio diseño. Se tiene en cuenta el tipo de aplicación y el volumen de contenidos mostrados.

✓ Comportamiento de Botones.

Se define el tratamiento general de los botones en función del concepto que representan, se precisan estados y especificidades.

✓ Pauta cromática general.

Se definen colores a usar determinando sus valores exactos en RGB. RGB es el acrónimo inglés Red, Green, Blue (Rojo, Verde, Azul). Es un modelo de color en el cual es posible representar un color mediante la mezcla por adición de tres colores primarios: rojo, verde y azul.

✓ Estilos para íconos.

Elaborar un conjunto de íconos que se integren sistémicamente en coherencia con el resto de la aplicación. Responderán a un grupo de parámetros que garanticen su efectividad comunicativa y validez icónica: semántico, sintáctico, pragmático.

✓ Comportamiento de gráficos.

Definir estilos visuales en correspondencia con el carácter de la aplicación y el tipo de información mostrada. Se definen colores, líneas, para los tipos de gráficos.

✓ Imágenes y sus tamaños.

Se definen tipos de imágenes: fotográficas, ilustraciones. Los tamaños se norman en función de las necesidades comunicacionales de la aplicación.

- Elaborar documento general de pautas para los montadores UI.
- Preparar el archivo .psd usado en el diseño para su entrega en función de la estructura propuesta en la arquitectura.

El diseñador grafico estructura el archivo .psd de modo que contenga, de forma detallada, una representación visual de las interfaces de la aplicación. Este archivo puede organizarse en función de la aplicación en una serie de carpetas y subcarpetas que contendrán las capas finales con el diseño propuesto, listo para su montaje. Se puede extraer de él información como la diagramación general, dimensiones, colores usados, tipografías, puntajes de títulos y contenidos, estilos de representación de eventos (los cambios que se observan cuando se activa o desactiva un botón o menú) que presentarán las interfaces finales del sitio, etc.

2.5.3.4 Salida

- Documento explicativo de las pautas para su montaje.
Incluye: muestras de pantallas y sus correspondientes dimensiones, diagramación general de la aplicación, comportamiento de etiquetas, botones, pauta cromática, etc.
- Archivo .psd.
PSD: Junto al Formato Extendido de Documentos (PSB) es el formato nativo del Photoshop. Admite capas y aunque su tamaño es grande admite altos niveles de compactación y modificaciones posteriores. Sus diferentes versiones son perfectamente compatibles entre sí. (25)
- Muestra de la totalidad de las pantallas contenidas en el psd. en formato jpg.
Incluye la misma información del PSD, pero en un formato de rápida visualización en un visor estándar. Su objetivo principal es brindar una idea general de todas las interfaces pautadas.

2.5.3.5 Personas responsables

- Diseñador gráfico.
Es un especialista en comunicación visual. En el diseño web pauta el comportamiento visual y funcional de la aplicación, desde un nivel macro hasta el nivel mas específico posible, en sintonía con la estructura de contenidos propuesta por el arquitecto de información. Establece códigos visuales coherentes capaces de ser decodificados por el usuario, y ordenando visualmente los contenidos para su óptimo uso.

2.5.3.6 Herramientas

- Adobe Photoshop CS2.

Es el software profesional por excelencia de edición y creación de imágenes rasters.

Una imagen raster es un mapa de bits, el cual sólo representa puntos (en color o blanco y negro) en determinados lugares de la pantalla de una computadora, formando lo que se entiende como una imagen digital.

Aunque en sus inicios fue concebido como un programa para edición de imágenes, las potentes herramientas con que cuenta actualmente lo han cambiado sustancialmente permitiendo además de editar, crear ilustraciones e imágenes de de alta calidad. Permite casi todas las extensiones disponibles de imágenes, así como intercambiar con múltiples aplicaciones. (25)

- Gimp (GNU Image Manipulation Program).

Es un programa de edición de imágenes, tanto dibujos como fotografías. Es un programa libre y gratuito, englobado en el proyecto GNU. Existen versiones totalmente funcionales para Windows, y para otros sistemas operativos, haciéndolo el programa de manipulación de gráficos disponible en más sistemas operativos. Se le puede considerar como la alternativa libre más firme para Photoshop.

2.5.4 Elaboración de los recursos de montaje de interfaces

2.5.4.1 Objetivos

- Definir las clases de estilos que soporten las pautas de diseño establecidas.
- Elaborar un prototipo funcional básico de la aplicación con el diseño propuesto.

2.5.4.2 Entrada

- Documento explicativo de las pautas para su montaje.
- Archivo .psd.
- Muestras de las pantallas en formato .jpg.

2.5.4.3 Actividades

- Proceso de optimización y salva de imágenes para la Web desde un archivo psd.
 - ✓ Definir áreas o imágenes necesarias para montar las UI.

El archivo .psd contiene el diseño del sitio completo. Es una representación gráfica que sirve de guía a los montadores a la hora de elaborar el archivo que funcionará como plantilla de la aplicación (que se nombrará escritorio de trabajo) y el resto de las páginas porque muestra con exactitud cómo deben quedar las páginas finales de la aplicación. Ahora, para confeccionar este archivo es necesario hacer una página que represente

exactamente lo que muestra el archivo de Photoshop, por lo que se definen las áreas que se van a recortar en este archivo para obtener algunos fragmentos de la imagen que son difíciles de elaborar a código y que luego se utilizarán para armar la estructura de esta página que funcionará como plantilla de la aplicación.

- ✓ Seccionamiento de las partes mediante la herramienta Slice.

El proceso de recortado se realiza haciendo uso de la herramienta **Cuchillo (Slice)**  de Photoshop que permite dividir el diseño para obtener pequeñas imágenes que se insertarán en el código HTML de la página. Al trazar el primer corte automáticamente se generan otros cortes. Por esto es importante tomar en cuenta que entre menos colores y cortes se empleen, menor será el peso de la página, o sea que tardará menos en cargarse. (Ver Anexos IX y X)

- ✓ Optimización y salva de imágenes a tamaño real.

Una vez realizados los cortes necesarios, se utiliza la opción “Salvar para Web” que crea un archivo HTML con su respectiva carpeta de imágenes donde aparecen los recortes realizados con el “Slice” y con esto se obtiene la página web cortada y lista para trabajarla en el editor HTML que se utilice.

La elección de los formatos de imágenes digitales se hace en función de sus propiedades. Existen dos tipos fundamentales de imágenes en dependencia de la forma de representarlas: los llamados mapas de bits y las imágenes vectoriales.

- Mapas de bits (bitmaps).

Se representan a través de una matriz de puntos, donde cada uno de ellos se corresponde con un pixel en la pantalla. A cada punto corresponde uno o más bits donde se almacena el color y la intensidad de cada punto. La resolución de la pantalla afecta la apariencia de la imagen en la misma. Son difíciles de editar y redimensionar sin distorsión. La velocidad de visualización depende del tamaño de la imagen y no de su complejidad. Permiten representar con mucha fidelidad la Realidad. Suelen generar archivos grandes.

- ✓ Formatos gráficos de los bitmaps.

Existen muchos tipos de formatos gráficos para los mapas de bits los que suelen ser reconocidos por su extensión. Entre los más comunes se pueden citar:

- Windows Bitmap (BMP). Formato no comprimido, optimizado para su rápido despliegue en pantalla. Su rapidez de carga lo hace indicado para aplicaciones multimedia. Permite hasta 24 bits de color.
- Graphical Interchange Format (GIF). Formato indexado (hasta 256 colores) y comprimido sin pérdida de información. Muy utilizado en gráficos que se muestran en las páginas Web de Internet. Se utiliza también para producir animaciones simples.
- Joint Photographic Experts Group es el nombre del comité que diseñó un estándar para la compresión de imágenes (JPG): Formato comprimido con pérdida de calidad (controlable por el usuario). Admite modos de 8 bits (indexado, de 256 colores) y 24 bits (hasta 16 millones de colores). Sus algoritmos de compresión permiten obtener archivos pequeños, por lo que es uno de los formatos más utilizados en el Web.
- Otros formatos de mapas de bits son: Portable Network Graphics (PNG), PC Paintbrush (PCX) y Tag Image File Format (TIF). En realidad no se aprecian grandes diferencias en cuanto a calidad de unos y otros, sino que simplemente son el resultado del trabajo de diferentes empresas de software.
- Imágenes vectoriales (Draw type).

Son el resultado de un método que utiliza descripciones matemáticas para determinar la posición, la longitud y la dirección de las líneas que deben dibujarse. Se almacenan como un conjunto de instrucciones que describen el tamaño, posición y forma de cada elemento que compone la imagen. Se visualizan correctamente independientemente de la resolución de pantalla. Se pueden editar partes individuales y hacerles transformaciones sin distorsión. Mientras más complejos, más tiempo toman en formar la imagen en la pantalla. No dan detalles fotorealistas. Requieren poco espacio de almacenamiento.

- ✓ Formatos gráficos de las imágenes vectoriales.

Entre los formatos existentes para las imágenes vectoriales se pueden mencionar como comunes los siguientes:

- Postscript Encapsulated. (EPS). Facilita el intercambio entre programas vectoriales y programas de armado de páginas o editores de bitmaps.

- Windows Metafile Format (WMF). Formato de prestaciones similares al EPS, aunque es reconocido sólo en ambiente Windows.

- Otros formatos de imágenes vectoriales son:

Formato de archivos producidos por CorelDraw (CDR), los utilizados por los programas CAD – Computer Aided Design, Diseño Asistido por Computadoras (DXF), Hewlett-Packard Graphics Language (HPGL) y Macintosh para ficheros gráficos (PICT).

- Diseño y maquetado de escritorio de trabajo.

En lo adelante lo que se busca es hacer que la imagen, la cual tiene un considerable peso en bytes, sea más ligera y así ayudar a la rápida carga de la misma en el navegador. Con los cortes lo que se intenta es separar las zonas de un solo color de las que están compuestas por un conjunto de estos, como las imágenes con profundidad. Las imágenes de un sólo color pueden ser sustituidas por color de fondo en el editor HTML o haciendo uso de CSS, lo que agiliza la carga de la página. El siguiente paso consiste en armar el escritorio de trabajo (archivo .jsp), definiendo el diseño estructural de la página (explicado más adelante) en dependencia del diseño de la aplicación, definido en el archivo .psd entregado por el diseñador gráfico. Esta estructura queda conformada colocando las imágenes recortadas en las posiciones que le corresponden teniendo en cuenta las dimensiones exactas que muestra el archivo .psd y rellenando el resto de los espacios con repetición de estas imágenes o color, haciendo uso de código HTML y CSS y en algunos casos Javascript, por ejemplo para menús desplegados. Así queda conformado el escritorio de trabajo que no es más que un archivo que contiene el diseño estructural de la aplicación en el que se define la zona en que se incluirán, de forma dinámica, los archivos de contenidos, esta operación se realiza en tiempo de ejecución mediante el menú que es el que le indica a dicha zona cual es el archivo que se debe a incluir.

La plantilla (archivo .dwt), que generalmente se elabora en Dreamweaver, queda sustituida por el escritorio de trabajo (archivo .jsp) que aparentemente es igual, pero que funciona de forma diferente, la plantilla (archivo .dwt) es aplicada a todas las páginas haciendo que cada una de estas contenga todo su código, mientras que con el escritorio de trabajo se tiene un solo archivo cuyo código no es necesario incluir en las demás páginas lo que posibilita no tener código repetido innecesariamente.

Diseño estructural:

Una buena práctica para el desarrollo de UI, es hacer un diseño estructural que consiste en realizar un esquema previo de cómo se van a visualizar cada una de las pantallas determinando componentes comunes y singulares en cada de ellas.

Una de las ventajas de realizar este diseño es la identificación de elementos comunes que pueden ser utilizados en todas las pantallas, lo que mejora el tiempo de desarrollo. Otra ventaja es que se le da uniformidad al sistema haciendo que este sea más agradable estéticamente. Entre los elementos que se deben tener en cuenta en el diseño estructural se encuentran:

✓ Zona de Encabezado.

El encabezado se ubica en la parte superior de la pagina, por lo general contiene un logo o una imagen que identifique la aplicación.

✓ Zona de Menú.

El menú es necesario para una navegabilidad rápida, se puede ubicar en varios lugares y debe ser accesible desde cualquier página. Es una buena práctica de programación web el utilizar menús, para no tener que regresar a páginas y causar mayor demora.

✓ Zona de Contenido.

La zona de contenido cambia constantemente, dependiendo de la operación requerida por el usuario, en esta zona se podrán visualizar el resto de las interfaces de la aplicación y a las que se pueda dirigir según el tipo de rol de usuario que esté en interacción.

✓ Componentes.

Un componente es un elemento que posee unas características definidas para cumplir un objetivo específico. Algunos componentes son únicos pero la gran mayoría son comunes para todas las páginas. Al observar cuales se necesitan, también se debe tener en cuenta cómo van a configurarse, es decir cómo se van a manejar las validaciones y cuales se van a hacer. Al definir los componentes desde el comienzo se tiene la ventaja de que se tendrá estipulado lo que se necesita para el desarrollo de las interfaces, así no se incluirán componentes que no aporten, porque todos estarán cumpliendo con un objetivo.

✓ Zona de Mensajes.

Esta zona se encarga de mostrar diferentes mensajes los cuales pueden ser de tipo informativo, de error, de advertencia o mixto. La implementación de esta zona se describe a continuación:

- **Estilo de campos:** Consisten en resaltar el campo donde hubo errores con el fin de que el usuario lo identifique para su posterior corrección, adicionalmente se cambia el color del nombre de este campo por colores llamativos que especifican la razón del error.
 - **Mensajes en la parte superior:** Consiste en ubicar el mensaje en la parte superior de la página, donde se especifique qué lo generó, ya sea la razón de un error o la confirmación exitosa de una operación.
- **Elaborar la hoja de estilos.**

Para elaborar la hoja de estilos se debe tener en cuenta las pautas de diseño gráfico. En este documento se definen las propiedades de cada una de las zonas delimitadas en el escritorio de trabajo como dimensiones, ubicación, márgenes, colores o imágenes de fondo, tipografías, bordes entre otras; además se definen las clases que caracterizan los estilos de los diferentes elementos que se visualizarán en las pantallas como pueden ser componentes, separadores, distancia entre ellos, mensajes de error, etc.

Se recomienda que:

- ✓ Solamente una persona se encargue del control de cambios en el archivo .css para evitar que el resto de las personas que lo utilizan lo transformen libremente según sus necesidades y así mantener la uniformidad en los estilos definidos.
- ✓ Se debe elaborar un documento explicativo que contenga nombre y descripción de las clases de estilos definidas con el fin de que las personas que van a trabajar con ellas tengan claro cual clase utilizar en cada momento.

2.5.4.4 Salida

- **Escritorio de trabajo (archivo .jsp).**
Es un archivo único que contiene el diseño estructural de la aplicación, en el que se definen las regiones que no cambiarán en ninguna interfaz como por ejemplo las zonas de banner y menú que siempre deben ser accesibles, y la zona de contenidos, que es la que cambia en cada una de las interfaces, en la que se cargan dinámicamente los archivos que contienen los componentes.
- **Hoja de estilo (archivo .css).**

La hoja de estilo (archivo .css) es en un archivo aparte que se enlaza con las demás páginas de la aplicación, solo en casos sencillos se puede escribir dentro del código HTML de la página web. Se utiliza para dar estilo a documentos HTML, separando el contenido de la presentación, permite controlar la apariencia de las páginas de la aplicación.

2.5.4.5 Personas responsables

- Equipo de montadores UI.

Como se explica en “Aspectos esenciales”, el equipo de montadores UI, queda creado, capacitado y preparado para enfrentarse al diseño y desarrollo de las UI, en esta fase es responsabilidad de los montadores, conformar el escritorio de trabajo y la hoja de estilo que serán usadas en el desarrollo posterior de las UI.

2.5.4.6 Herramientas

- Adobe Photoshop CS2.
- Gimp.
- Eclipse 3.2.

Es un completo entorno de desarrollo para trabajar en Java, permite programar, desarrollar y compilar aplicaciones en Java, sitios web, programas en C++, etc. Se utiliza Exadel Studio como plug-in de Eclipse que proporciona una interfaz gráfica atractiva para la creación de JSF, incluyendo librerías, es un programa confiable y muy utilizado. (26)

Los recursos de montaje de las UI, son aquellos que hacen referencia a la presentación estética (distribución, colores, fuentes, etc.) de cada una de las interfaces. En esta fase, destinada a la elaboración de dichos recursos, que se utilizan en el posterior desarrollo de las UI, es donde se obtiene el ensamblaje de lo que serán las interfaces del software y es de gran importancia para enfocar a los montadores, encargados de la construcción de las UI, en el resultado que se desea alcanzar, estandarizando el procedimiento de montaje de UI.

2.5.5 Desarrollo de las UI

2.5.5.1 Objetivos

- Proporcionar al programador interfaces funcionales, listas para proceder a su implementación que cumplan con las pautas de DG y AI establecidas.

2.5.5.2 Entrada

- Hoja de estilo (archivo .css).
- Integración de CU.

2.5.5.3 Actividades

- Montar las UI haciendo uso de la hoja de estilo ajustándose a las pautas predefinidas.
Haciendo uso de Eclipse se crea un nuevo proyecto JSF. Luego se procede a la creación de las nuevas páginas (archivos .jsp), dentro de las que se incluyen y organizan todos los elementos de la zona de contenidos de la página, a código JSF (que es el framework de Java utilizado para la programación de la capa de presentación del proyecto) y para darles la apariencia correspondiente, se hace uso de las clases de estilos definidas en el archivo .css elaborado en la fase anterior, se utiliza como guía la integración de CU. Una vez construidas todas las pantallas del CU, se verifica que concuerden con el prototipo del CU y se ajusten a las pautas de DG y AI y así se obtienen los prototipos funcionales, es decir, las interfaces listas para proceder a su implementación.
- Revisión de las UI.
En el proceso de implementación de las interfaces, en algunas ocasiones, ocurren desajustes o cambios en las interfaces entregadas por los montadores. Esto se debe fundamentalmente a la falta de práctica de los programadores en el manejo de los estilos definidos para la apariencia de las interfaces, además de su poca familiarización con las pautas de DG y AI. Para rectificar estos errores se debe llevar a cabo una revisión de las UI, por parte de los montadores, que consiste en reevaluar las interfaces ya implementadas y hacer los cambios necesarios para llevarlas a su estructura original.

2.5.5.4 Salida

- Interfaces (Páginas web archivos .jsp).
Java Server Pages (JSPs) son páginas web con etiquetas especiales para programar scripts de servidor en sintaxis Java, mezclado con código HTML. JSF es una librería de etiquetas (tags) que se integra con las páginas JSP a través de las etiquetas personalizadas (custom tags) de JSP.

2.5.5.5 Personas responsables

- Equipo de montadores UI.

A partir de este momento los montadores son responsables de crear y mantener un conjunto de UI que sean atractivas, comunicativas, usables y se ajusten a las pautas de DG y AI.

2.5.5.6 Herramientas

- Eclipse 3.2.

En esta etapa culminante se desarrollan las UI finales del software. Al concluir esta fase se provee a los programadores de interfaces terminadas que cumplen, en primer lugar, con los requisitos del usuario y con las pautas de diseño y arquitectura establecidas en las fases anteriores. La importancia de esta etapa consiste además en que las interfaces entregadas ya contienen todos los componentes listos para ser implementados, lo que contribuye a agilizar el trabajo de los programadores y con esto el tiempo de desarrollo del proyecto.

2.6 Conclusiones

En este capítulo se desarrolló la propuesta de solución, obteniéndose una metodología para el diseño y desarrollo de UI de la aplicación SIIPOL (DDWUI). Para su creación se definieron los objetivos de la misma, los cuales se enfocaron en la obtención de interfaces de usuario con buena calidad y en facilitar el trabajo de los programadores y con esto optimizar tiempo en la realización del proyecto. Partiendo de una disposición de definiciones preliminares y aspectos esenciales, que complementan la solución y son importantes para facilitar una mejor comprensión y lograr los resultados esperados. Se definieron las fases que componen esta metodología y que además son su fundamento, se identificaron los roles y responsabilidades involucrados directamente con las UI los cuales se pueden adaptar a las características propias de los proyectos que se realizan en la universidad. Se determinaron las actividades generales y específicas que se deben tener en cuenta para un correcto diseño y desarrollo de UI, se especificaron las principales herramientas con las que se realizará el montaje de las mismas y los artefactos que se generarán para la entrada y salida de información en cada una de las fases que componen DDWUI. Al aplicar la propuesta desarrollada se espera obtener las siguientes ventajas:

- Fácil comprensión, por ser un mecanismo de diseño y desarrollo de UI para web sencillo y fácil de seguir.

- Aumento de la calidad en las UI, porque obliga a cumplir con pautas de DG y AI, se evalúan las UI después de ser implementadas, lo que permite detectar errores a tiempo permitiendo una corrección temprana.
- Facilita el trabajo a los programadores, porque se les entrega UI que cumplen con pautas de DG y AI y contienen todos los componentes necesarios, listos para ser implementados.

3

Análisis de los Resultados

3.1 Introducción

Para dar cumplimiento a la validez y precisión de la solución propuesta, fue necesario aplicar la metodología desde su planificación hasta su ejecución y, luego, realizar el análisis de los resultados. La metodología definida se aplicó de manera correcta en el proyecto productivo CICPC. En este capítulo se resume la experiencia, beneficios obtenidos y una demostración de las ventajas que puede ofrecer aplicar DDWUI; se hace un análisis de los resultados alcanzados luego de aplicar la metodología, según los criterios expresados en las encuestas realizadas a un grupo representativo de personas, que forman parte del proyecto; además se hace uso del método de validación por expertos para evaluar el valor práctico de la metodología según el criterio de los especialistas.

3.2 Validación de la propuesta de solución

3.2.1 Reportes de los resultados antes de aplicar DDWUI

Se aplicaron las encuestas a los 25 integrantes del proyecto CICPC, previstos inicialmente, los 10 que conforman el equipo de montadores UI y 15 desarrolladores escogidos de los diferentes módulos del proyecto.

Con esto se llegó a la conclusión de que el 80% de los encuestados opinaron que las UI que se realizaban en el proyecto tenían poca calidad.

El 76% manifestó que el rol de los montadores UI no desempeñaba un buen papel dentro del proyecto y solo un 24% afirmó que los montadores trabajaban bien y que eran necesarios en el equipo de desarrollo del proyecto.

Sólo un 12% tenía conocimiento de las pautas de DG y AI, tan necesarias para construir UI adecuadas, el resto afirmó que no tenían dominio de pauta alguna.

El 45% de los encuestados determinó que la comunicación entre las personas que trabajan con las UI era mala y en algunos casos regular, un por ciento muy bajo asintió que era buena.

Existía un 70% que desconocía totalmente las herramientas a utilizar en el proyecto y con las cuales se desarrollaban las UI. (Ver Anexo XI)

Las comunicaciones con el líder del proyecto permitieron conocer que el rol de montadores UI no estaba debidamente establecido, ni definidas las actividades que ellos debían realizar. Las UI se construían prácticamente haciendo estimaciones, no se seguían pautas de ningún tipo para diseñarlas y desarrollarlas lo que dificultaba el trabajo de los programadores influyendo en el tiempo empleado en la implementación de un CU. Según la planificación definida en el proyecto, dependiendo de la complejidad de un CU, a un programador de la capa de presentación se le asigna: 6 sesiones de trabajo para la implementación de un CU básico, 12 sesiones para un CU medio y 18 sesiones para un CU complejo. Antes de aplicar DDWUI los programadores debían utilizar aproximadamente de 3 a 5 de sus sesiones planificadas (según la complejidad del CU), para ajustar los componentes dentro de las interfaces, hasta lograr el prototipo funcional que necesitaban para comenzar la implementación, por tanto los tiempos planificados aumentaban notablemente, para un CU básico utilizaban 9 sesiones de trabajo, para uno medio 16 y para uno complejo 22, aproximadamente, alcanzando un promedio de 4 sesiones de atraso por cada CU, esto provocaba un demora en el tiempo de realización del proyecto.

Niveles de complejidad de un CU: básico, medio y complejo.

Sesión de trabajo: jornada de trabajo de 4 horas o más.

3.2.2 Reportes de los resultados después de aplicar DDWUI

El reporte de los resultados se realizó aplicando encuestas a los integrantes del proyecto CICPC. Al comenzar a probar la metodología al proyecto fueron encuestados los 15 desarrolladores y los 10 montadores UI nuevamente. (Ver Anexo II)

El 85% manifestó que el nivel de calidad que presentaban las UI del proyecto había aumentado considerablemente.

El 95% de los encuestados opinó que la nueva forma de trabajo que seguían los montadores UI era buena, de ellos el 67% afirmó que ayudaba a mejorar la comunicación entre los integrantes de los equipos que trabajan directamente con las UI, notando la diferencia producida en el proyecto después de aplicar la metodología DDWUI.

El 90% afirmó tener conocimiento de pautas de DG Y AI, reconociendo además la importancia de conocerlas, mientras que el 100% demostró poseer completo dominio de la nueva herramienta utilizada en el proyecto para diseñar y desarrollar UI y el 96% coincidió en que DDWUI sirve de gran ayuda para mejorar la calidad de las UI, agilizar y optimizar el tiempo de desarrollo del proyecto CICPC. (Ver Anexo XII). Actualmente un programador de la capa de presentación no tiene que emplear ninguna de las sesiones de trabajo asignadas para la

implementación en modificar las interfaces; luego del proceso de montaje, obtiene UI listas para programar, de esta manera puede cumplir satisfactoriamente con su planificación.

3.2.3 Determinación del valor práctico de la solución según expertos

Al culminar la Metodología para el diseño y desarrollo de UI de la aplicación SIIPOL (DDWUI) se circuló entre especialistas en el tema un resumen de las propuestas principales dentro de la metodología.

La Metodología fue evaluada por ocho Ingenieros Informáticos con dos años de experiencia como promedio, uno de ellos, Máster en Ciencias. La selección de los jueces expertos, se realizó teniendo en consideración su experiencia profesional e investigativa en el ejercicio de las ciencias informáticas, para de este modo tener la posibilidad de obtener criterios de diversos especialistas y realizar un análisis integral de la solución propuesta. Se incluyeron profesionales considerados expertos, todos Ingenieros en Ciencias Informáticas que ejercen su actividad profesional vinculados a la producción en la UCI quienes respondieron el cuestionario de validación definitivo. (Ver Anexo XIII)

La caracterización de los jueces expertos es como sigue: tres analistas principales de diferentes módulos del proyecto CICPC con 2 años de experiencia, el analista principal del proyecto CICPC con 2 años de experiencia, la líder del proyecto CICPC con 3 años de experiencia, el jefe del módulo Registro y Control con 2 años de experiencia, un Máster en Ciencias Informáticas jefe de desarrollo de software del proyecto CICPC y jefe del módulo Investigación Criminalística con 3 años de experiencia, y un Especialista Superior en Informática de DESOFT jefe del módulo Investigación Forense.

Para evaluar el valor práctico de la metodología se fijaron un grupo de indicadores. Los criterios de los especialistas reflejados en por ciento aparecen totalizados en la siguiente tabla.

	Bueno	Regular	Malo	No Aplicable
Objetivos que persigue	100%	0%	0%	0%
Contenido que incluye	87.5%	12.5%	0%	0%
Actividades propuestas	87.5%	12.5%	0%	0%
Herramientas propuestas	100%	0%	0%	0%
Aplicabilidad al proyecto CICPC	62.5%	37.5%	0%	0%
Posibilidad de utilización en proyectos similares	62.5%	37.5%	0%	0%

Tabla 01: Juicio de Expertos

La validación mediante juicio de expertos reveló que la Metodología DDWUI se puede integrar dentro del proceso de desarrollo del software del proyecto CICPC. En general los objetivos que persigue, el contenido que incluye, las actividades y herramientas propuestas, la aplicabilidad al proyecto CICPC y otros proyectos similares, son acertados, aunque los expertos coinciden en expresar que estos dos últimos aspectos pueden ser perfeccionados. (Ver Anexo XIV)

3.3 Entregables obtenidos

Los resultados obtenidos al aplicar DDWUI se exponen realizando una iteración completa de la Metodología, tomando como ejemplo para este análisis un caso de uso con el que se transita por todas las fases mostrando los resultados de cada una de ellas, hasta llegar a las interfaces finales de dicho CU. Es importante destacar que dentro de cada una de las fases las actividades se realizan de forma general, para todo el proyecto. En esta parte se centrará la atención específicamente en los artefactos referentes al CU que se estudia.

El proyecto CICPC se encuentra dividido en subsistemas que a su vez están divididos en módulos. El CU que se utiliza es Gestionar Resultado de Experticia Balística, perteneciente al módulo Experticias dentro del subsistema Investigación Criminalística.

En la primera fase Estudio y Modelado del Sistema, a partir del análisis del entorno donde está enmarcado el negocio, los analistas del proyecto llevaron a cabo el Modelamiento del negocio y el levantamiento de requisitos para lograr la concepción del sistema. Se obtuvieron las especificaciones de los CU. Especificación del CU Gestionar Resultado de Experticia Balística. (Ver Anexo XV).

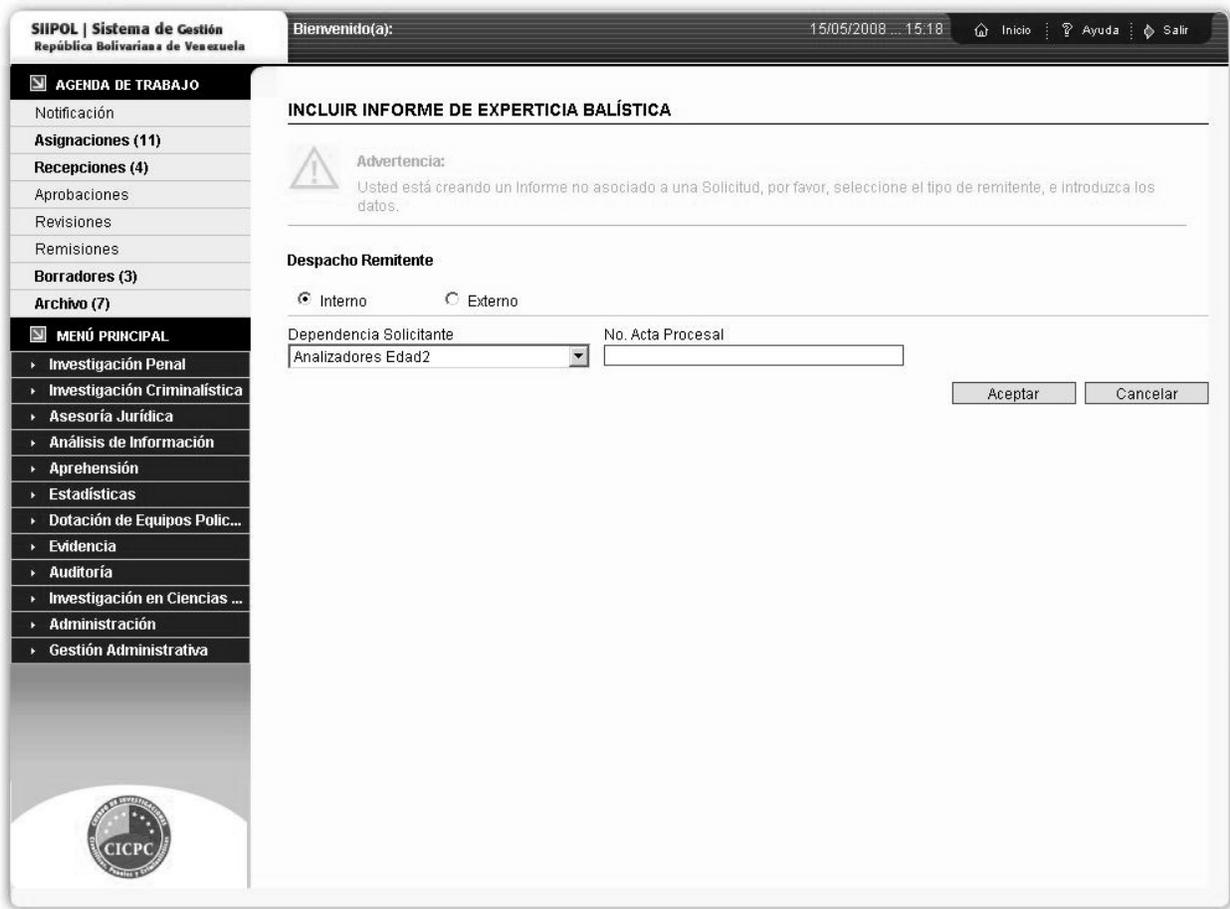
En la segunda fase, Arquitectura de la Información se realizó, por parte de los arquitectos de información, el estudio de la audiencia y sus necesidades. A partir de este estudio se definieron los contenidos a incluir en el producto y su organización. Se estructuró el sistema de navegación, el sistema de etiquetado y el diseño conceptual, considerando los factores usabilidad y accesibilidad. Además se elaboró el documento de Pautas de Arquitectura de Información (Ver Anexo XVI) y se construyeron los prototipos de UI para cada CU. Prototipo de UI del CU Gestionar resultado de Experticia Balística. (Ver Anexo XVII).

En la tercera fase, Diseño Gráfico, a partir de estudiar el comportamiento de la aplicación y el documento de Pautas de Arquitectura de Información, el diseñador gráfico diseñó las interfaces de la aplicación, elaboró el documento general de pautas para los montadores UI

Manual de Pautas SIIPOL (Ver Anexo XVIII). Se obtuvo el archivo .psd con el diseño de las interfaces (Ver Anexo XIX) y las muestras de las pantallas en formato .jpg (Ver Anexo XX).

En la cuarta fase, Elaboración de los recursos de montaje de interfaces, teniendo en cuenta el documento explicativo de las pautas para el montaje de UI, el archivo .psd y las muestras de las pantallas en formato .jpg, los montadores UI construyeron el escritorio de trabajo (Ver Anexo XXI) para la aplicación y la hoja de estilo (Ver Anexo XXII).

En la quinta fase, Desarrollo de UI utilizando la hoja de estilo y la especificación de los CU y el prototipo de UI, los montadores se encargan de construir las interfaces, y revisarlas luego de su implementación. Interfaces del CU Gestionar Resultado de Experticia Balística. (Ver Anexo XXIII). A continuación se observan muestras de estas interfaces tomadas de la aplicación.



SIIPOL | Sistema de Gestión
 República Bolivariana de Venezuela

Bienvenido(a): 15/05/2008 ... 12:42
Inicio
Ayuda
Salir

AGENDA DE TRABAJO

- Notificación (1)
- Asignaciones (6)
- Recepciones (1)
- Aprobaciones (2)
- Revisiones
- Remisiones
- Borradores (3)
- Archivo (2)

MENÚ PRINCIPAL

- Investigación Penal
- Investigación Criminalística
- Solicitudes
- Registros de Informes
 - Avalúo
 - Balística
 - Contable
 - Inspección de Inmueble
 - Inspección Técnica
 - Levantamiento Planimét...
 - Microscopía Electrónica
 - Pericial Criminalístico
 - Reconstrucción de Hec...
 - Siniestro
 - Trayectoria Balística
 - Trayectoria Intraorgánica
 - Vehículo
- Consultas
- Prueba
- Asesoría Jurídica
- Análisis de Información
- Aprehensión
- Estadísticas
- Dotación de Equipos Polic...
- Evidencia
- Auditoría
- Investigación en Ciencias ...
- Administración
- Gestión Administrativa



INCLUIR INFORME DE EXPERTICIA BALÍSTICA

Informe de Experticias Balística
Elementos
Imágenes

Número de Acta Procesal	J-01-002-0003
Despacho Solicitante	Fiscalía Penal
Representante del Despacho	Yudanis Gago Martínez

Resultados

Peritaje Conclusiones

Informe / Peritación

N X S | -- Tipo de fuente --

A | -- Tamaño del texto --

Base Legal

Ley Artículo Adicionar

Ley	Artículo	Eliminar
No existe registro		

Total: 0

Vista Previa Guardar Limpiar Cancelar

SIIPOL | Sistema de Gestión
 República Bolivariana de Venezuela

Bienvenido(a): 15/05/2008 ... 12:49
 Inicio
Ayuda
Salir

AGENDA DE TRABAJO

Notificación (1)

Asignaciones (6)

Recepciones (1)

Aprobaciones (2)

Revisiones

Remisiones

Borradores (3)

Archivo (2)

MENÚ PRINCIPAL

▶ Investigación Penal

▶ Investigación Criminalística

▶ Asesoría Jurídica

▶ Análisis de Información

▶ Aprehensión

▶ Estadísticas

▶ Dotación de Equipos Polic...

▶ Evidencia

▶ Auditoría

▶ Investigación en Ciencias ...

▶ Administración

▶ Gestión Administrativa



INCLUIR INFORME DE EXPERTICIA BALÍSTICA

Informe de Experticias Balística

Elementos

Imágenes

Evidencias Relacionadas

Armas

Objetos

No. Evidencia	Nombre Evidencia	Descripción	Disociar
90	Arma	descripción de la evidencia	Disociar
Total: 1			

[Incluir Arma](#)

SIIPOL | Sistema de Gestión
 República Bolivariana de Venezuela

Bienvenido(a): 15/05/2008 ... 12:52

 Inicio Ayuda Salir

AGENDA DE TRABAJO

- Notificación (1)
- Asignaciones (6)
- Recepciones (1)
- Aprobaciones (2)
- Revisiones
- Remisiones
- Borradores (3)
- Archivo (2)

MENÚ PRINCIPAL

- ▶ Investigación Penal
- ▶ Investigación Criminalística
- ▶ Asesoría Jurídica
- ▶ Análisis de Información
- ▶ Apreensión
- ▶ Estadísticas
- ▶ Dotación de Equipos Polic...
- ▶ Evidencia
- ▶ Auditoría
- ▶ Investigación en Ciencias ...
- ▶ Administración
- ▶ Gestión Administrativa



INCLUIR INFORME DE EXPERTICIA BALÍSTICA

Informe de Experticias Balística
Elementos
Imágenes

Imágenes Asociadas

Nombre Imagen	Descripción	Disociar
Arma	Arma de la experticia balística	Disociar
Total: 1		

Vista Previa



Ampliar

Asociar imagen



Ampliar

Nombre	<input style="width: 90%;" type="text"/>
Ubicación	<input style="width: 90%;" type="text"/> Examinar...
Descripción	<input style="width: 90%;" type="text"/>
Fecha de captura	<input style="width: 90%;" type="text"/> 🗑

Incluir Imagen

The screenshot displays the SIIPOL web application interface. At the top, the header includes the logo 'SIIPOL | Sistema de Gestión República Bolivariana de Venezuela', a welcome message 'Bienvenido(a):', the date '15/05/2008 ... 12:53', and navigation links for 'Inicio', 'Ayuda', and 'Salir'. The left sidebar contains a menu with sections: 'AGENDA DE TRABAJO' (with sub-items like 'Notificación (1)', 'Asignaciones (6)', etc.) and 'MENÚ PRINCIPAL' (with sub-items like 'Investigación Penal', 'Investigación Criminalística', etc.). The main content area is titled 'VISTA PREVIA INFORME DE EXPERTICIA BALÍSTICA' and contains the following information:

Informe Pericial de Experticia Balística

Número de Acta Procesal	J-01-002-0003
Despacho Solicitante	Fiscalía Penal
Representante del Despacho	Yudanis Gago Martínez

No.Evidencia:
90

Descripción:
descripción de la evidencia

Resultado:

Peritación:

Base Legal

De Conformidad con lo establecido en la Ley	Artículo
No existen registros	

Navigation arrows: << < > >>

A 'Cerrar' button is located at the bottom right of the main content area.

3.4 Conclusiones

En este capítulo se hizo un análisis comparativo de la situación existente respecto a las UI del proyecto antes y después de aplicar la solución; dicho análisis se realizó estudiando los resultados obtenidos en encuestas realizadas a usuarios internos del software. Se obtuvieron resultados favorables dentro del proceso de desarrollo del software al incluir la metodología dentro del mismo.

Además se evaluó la Metodología DDWUI mediante el método de Evaluación por Expertos; el cual permitió analizar los criterios de cada uno de los expertos y determinar el índice de aceptación que tiene la propuesta del presente trabajo, obteniéndose concordancia en el trabajo realizado y una alta probabilidad de éxito de ser aplicado al proceso de desarrollo de software de proyecto CICPC, con expectativas de utilizarse en proyectos similares que se desarrollan en la Universidad.

Conclusiones

La realización de este trabajo ha posibilitado cumplimentar los objetivos para los cuales se desarrolló la investigación. Se adquirieron conocimientos importantes sobre las interfaces de usuario que pueden ser aplicados a cualquier proyecto con características similares en busca de organizar el trabajo y elevar la calidad en la producción de software.

Se obtuvo una metodología para guiar el proceso de diseño y desarrollo de interfaces de usuario de la aplicación SIIPOL, la cual especifica los objetivos y actividades a realizar, los roles y responsabilidades de los involucrados en cada una de ellas, los artefactos de entrada y salida que deben ser elaborados en cada una de las fases y las herramientas que se deben utilizar para la construcción de los mismos. Además explica en detalle los pasos que se deben seguir para que la ejecución de la metodología sea exitosa.

De la Metodología DDWUI se puede destacar que:

- Contribuye al mejoramiento de la calidad de las interfaces del software detallando un proceso de diseño y desarrollo de UI bien definido.
- Por ser una metodología flexible, se le puede incorporar cualquier actividad adicional y se puede hacer uso de ella independientemente de la tecnología que se utilice; su principal ventaja está en la adaptabilidad que puede tener a cualquier proyecto de gestión empresarial sobre web.

Finalmente se realizó la evaluación de la aplicabilidad de la propuesta mediante el método de validación de expertos, en el cual se obtuvo gran aceptación al insertar la metodología dentro del proceso de desarrollo del software del proyecto CICPC y una considerable probabilidad de éxito de ser aplicada en proyectos productivos que desarrollen aplicaciones de gestión empresarial sobre web. Experimentar con la metodología diseñada en el proyecto CICPC fue más que una necesidad un reto. Se logró probar en su totalidad brindando todos los resultados esperados, sirvió de gran utilidad para la organización interna del trabajo y lograr calidad en las interfaces finales del software, lo que implica desde el punto de vista teórico, el cumplimiento de la idea a defender planteada.

Recomendaciones

Por los aportes que tiene incluir dentro del proceso de desarrollo del software una metodología que guíe el proceso de diseño y desarrollo de UI se sugiere el seguimiento de la solución propuesta para obtener un producto de gran aceptación. Por tanto se proponen las siguientes recomendaciones:

- Definir con exactitud las características que deben tener otros proyectos para considerarlos posibles candidatos a utilizar la metodología DDWUI.
- Extender y aplicar la metodología DDWUI a los proyectos productivos de la universidad que se consideren similares.
- Generalizar la metodología DDWUI de modo que sea totalmente aplicable a cualquier tipo de proyecto.

Durante el período de prueba de la metodología DDWUI se arrojó como resultado la necesidad de incluir dentro de la misma una fase que inicialmente no se encontraba dentro del alcance de la investigación. La Gestión de Configuración, esta nueva fase se encargaría principalmente de la gestión de cambios, por ser este un proceso complejo que requiere un grupo de actividades que se deben definir para las especificidades del proyecto, se propone:

- Incluir como extensión de la metodología DDWUI una fase que contemple la gestión de configuración.

Referencias Bibliográficas

1. Gaiasur. [En línea] [Citado el: 2 de Febrero de 2008.]
<http://www.gaiasur.com.ar/infoteca/siggraph99/disenio-de-interfaces-y-usabilidad.html>.
2. El mundo.es. [En línea] [Citado el: 4 de Febrero de 2008.]
http://diccionarios.elmundo.es/diccionarios/cgi/lee_diccionario.html?busca=interfaz.
3. WordReference.com. [En línea] [Citado el: 25 de Enero de 2008.]
<http://www.reference.com/definicion/interfaz>.
4. Española, Real Academia de la Lengua. Diccionario de la Real Academia de la Lengua Española 22 ed. [<http://www.rae.es>] España: s.n., 2001.
5. Glosarium.com. Diccionario Informático. [En línea] [Citado el: 20 de Enero de 2008.]
<http://www.glosarium.com/list/14/5,l,,xhtml>.
6. Shneiderman, Ben. El diseño de la interfaz de usuario: estrategias para una eficaz Interacción Hombre-Máquina, 4ed. Estados Unidos: s.n., 2005. 978-84-205-4803-6.
7. DesarrolloWeb. [En línea] [Citado el: 25 de Enero de 2008.]
http://www.desarrolloweb.com/manuales/64/Compometes_de_un_interfaz_web_Interfaces.html.
8. María Jesús Lamarca Lapuente. Hipertexto: El nuevo concepto de documento en la cultura de la imagen. [En línea] [Citado el: 15 de Enero de 2008.]
http://www.hipertexto.info/documentos/interfaz_grafica.html.
9. Mateu, Ing. Carles. Desarrollo de aplicaciones web. Barcelona: s.n., 2004. 84-9788-118-4.
10. Musciano, Chuck y Kennedy, Bill. HTML. La Guía Completa 2ed. actualizada. México: s.n., 2001. 970-10-2141-X.
11. Eguíluz Pérez, Javier. Introducción a CSS. [En línea] [Citado el: 3 de Febrero de 2008.]
<http://www.librosweb.es/css/index.html>.
12. —. Introducción a Javascript. [En línea] [Citado el: 3 de Febrero de 2008.]
<http://www.librosweb.es/javascript/index.html>.

13. —. Introducción a AJAX. [En línea] [Citado el: 3 de Febrero de 2008.]
<http://www.librosweb.es/ajax/index.html>.
14. DesarrolloWeb. [En línea] [Citado el: 7 de Febrero de 2008.]
<http://www.desarrolloweb.com/articulos/2380.php>.
15. Adobe. [En línea] [Citado el: 7 de Febrero de 2008.]
<http://www.adobe.com/es/products/flash>.
16. Adobe. [En línea] [Citado el: 14 de Febrero de 2008.]
<http://www.adobe.com/es/products/dreamweaver>.
17. Eclipse. [En línea] [Citado el: 5 de Febrero de 2008.] <http://plataformaclipse.com>.
18. Interacción Humano-Computador y diseño de interfaces. Dr. Girardi, Rosario e Ing. Deagostini, Adriana. Uruguay: s.n., 2005.
19. Dra. Escalona Cuaresma, María José y Dr. González Romano, José Mariano. Programa de Doctorado Tecnología e Ingeniería del Software. España: s.n., 2006.
20. Hassan, Yusef y Martín Fernández, Francisco J. y Iazza, Ghzala. Hipertext.net. Diseño Web Centrado en el Usuario: Usabilidad y Arquitectura de la Información. [En línea] Mayo de 2004. [Citado el: 21 de Febrero de 2008.]
<http://www.hipertext.net/web/pag206.html#Introduccion>.
21. Propuesta de Adaptación De la Metodología de Diseño Centrado en el Usuario para el desarrollo de sitios web accesibles. Hassan Montero, Yudsef y Martín Fernández, Francisco J. 3, Granada: s.n., 2004, Vol. XXVII.
22. Mercovich, Eduardo. Gaiasur. [En línea] [Citado el: 16 de Marzo de 2008.]
<http://planeta.gaiasur.ar/infoteca/disenio-de-interfaces-y-usabilidad.html>.
23. Ibáñez, Juan Carlos Pantoja. Arquitectura de la Información: Introducción al proceso de desarrollo en el Diseño de Interfaces de Usuario. Chile: s.n., 2004.
24. Bustamante, Antonio Montes de Oca Sánchez de. Arquitectura de Información y Usabilidad para la World Wide Web. La Habana: s.n., 2005.

25. Adobe. [En línea] [Citado el: 14 de Febrero de 2008.]

<http://www.adobe.com/es/products/photoshop>.

26. Todo Programas. [En línea] [Citado el: 14 de Febrero de 2008.]

<http://www.todoprogramas.com/programa/eclipsesdk>.

27. Pressman, Roger S. Ingeniería del Software. Un enfoque práctico 5ed. Connecticut: s.n., 2002.

Bibliografía

- Ball, Jennifer; Carson, Debbie; Evans, Ian; Fordin, Scott; Haase, Kim; Jendrock, Eric. *The Java EE 5 Tutorial*. Sun Java System Application Server Platform Edition 9. June, 2006
- Bobadilla, Jesús Sancho; Alcocer, Alejandro Jarabo. Active server page 3: creación de aplicaciones web a través de ejemplos. 2005
Disponibile en: <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo> (09/02/08)
- C. Lewis; J. Rieman: *Task-Centered user Interface Design. A practical introduction*. Libro Shareware: [ftp.cs.colorado.edu](ftp://ftp.cs.colorado.edu), 2002.
- Carroll, John M. *Making use. Scenario-Based Design of Human-Computer Interactions*. MIT Press 2000. ISBN 0-262-03279-1
- Codina, Lluís. El libro digital y la www. Madrid, Tauro, 2000.
- Coutin, A. Arquitectura de información para sitios web. Anaya Multimedia. Madrid. Julio 2002
- D. Mayhew. Principles and Guidelines in Software User Interface Design. Prentice Hall PTR, 2002. ISBN: 0-13-721929-6
- D. R. Olsen. Developing User Interfaces. Morgan Kaufmann Publisher, 2002. ISBN: 1-55860-418-9
- Dürsteler, Juan Carlos. Visualización de información. Gestión 2000.com, 2002
- Eccher, Clint; Hunley, Eric; Simmons, Erik. Creación y diseño web profesional. Anaya Multimedia. 1ª ed. 2005.
- Horton, Sara; Lynch, Patrick J. Manual de estilo web: Principios de diseño básicos para la creación de sitios web. 2005
- Jackwind, Li Guojie. UI development with JavaServer Face. Nanyang Technological University. Singapore. 2001

- Jacobson, Ivar; Booch, Grady; Rumbaugh, James, El Proceso Unificado de Desarrollo de Software. Addison Wesley, 2000
- K. Cox, D. Walker. User Interface Design, 2ª ed. Prentice Hall, 2000.
- L. Bartfield. The User Interface: Concepts & Design. Addison Wesley. 2003. ISBN: 0-201-54441-5
- Mora, Marcos; Mora, Mari Carmen. Pautas para el diseño y la evaluación de interfaces de usuario. Información y documentación digital. Barcelona: IULA; Documenta Universitaria, 2004.
Disponibile en: http://www.mcmarcos.com/pdf/2004_pautas-iula.pdf (20/02/08)
- Powell, Thomas A. Manual de referencia: Diseño de sitios web. McGraw Hill, Osborne. 2000
- Pressman, Roger Ingeniería del Software. Un enfoque práctico. McGraw-Hill/Interamericana de España. 2002
- Shneiderman, Ben. El diseño de la interfaz de usuario: estrategias para una eficaz Interacción Hombre-Máquina, 4ª ed. Estados Unidos, 2005.
- Valdés Pardo, Víctor Giraldo. Nuevas tecnologías de la información y la comunicación. Editorial Feijoo, Villa Clara, 2003
- W. Newman; M. Lamming. Interactive System Design. Addison Wesley Publishers, 2005. ISBN: 0-201-63162-8
- Zeldman, Jefrerey. Diseño con estándares web. Anaya Multimedia. 2003

Glosario de Términos

- **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.
- **Componentes:** Conjunto de elementos que componen la interfaz gráfica de una aplicación, que son utilizados por el usuario final para realizar las operaciones de entrada y salida de información.
- **Estándar:** Normas de desempeño definidas para una actividad, un proceso, un producto o un servicio.
- **Estándar Web:** Un estándar define un conjunto de criterios o normas de que sirven para guiar el desarrollo y expansión de la Web.
- **Fase:** Periodo de tiempo entre dos hitos principales de un proceso de desarrollo.
- **Hardware:** Término que se utiliza para describir los artefactos físicos de una tecnología. Puede ser un equipo electrónico o informático. En la informática se denomina hardware al conjunto de materiales físicos que componen una computadora.
- **Herramientas:** Software que se utiliza para automatizar las actividades definidas en el proceso.
- **Implementación:** Es un flujo de trabajo de la ingeniería de software donde se define cómo organizar los objetos en componentes de software. En esta etapa se construye el software y se obtiene un producto listo para probar y utilizar.

- **Metodología:** Es un conjunto de métodos de investigación apropiados al quehacer de una ciencia determinada. Es la parte de un proceso de investigación que permite sistematizar los métodos y técnicas para realizar el proceso con eficiencia.
- **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 software:** 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.
- **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 una computadora.
- **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.
- **W3C:** World Wide Web Consortium. Organización que desarrolla estándares para guiar el desarrollo y expansión de la Web.

Anexos

Anexo I. Encuesta realizada a integrantes del CICPC antes de aplicar DDWUI

Se desea conocer sus opiniones sobre las Interfaces de Usuario (UI) de la aplicación SIIPOL perteneciente al proyecto CICPC. Su respuesta a cada pregunta se considera de gran importancia para el perfeccionamiento futuro de las UI del proyecto.

1. ¿Qué opina sobre las interfaces de usuario (UI) del proyecto CICPC?

2. Diga las deficiencias que le encuentra a las UI del proyecto CICPC.

3. Marque con una cruz (X) su respuesta.

Considera que el rol montadores UI desempeña un buen papel dentro del proyecto.

Si ___ No ___

En caso negativo diga por qué: _____

4. ¿Existe algún procedimiento que garantice la calidad de las UI elaboradas hasta el momento en el proyecto?

Si ___ No ___ No sé ___

5. ¿Existe alguien encargado de supervisar la calidad de las UI del proyecto?

Si ___ No ___ No sé ___

6. Conoce las pautas de Diseño Grafico y Arquitectura de la Información establecidas el proyecto. Si ___ No ___

7. Considera que las UI que se elaboran actualmente en el proyecto están listas para ser implementadas. Si ___ No ___

En caso negativo diga por qué: _____

8. Diga cómo es la comunicación entre los integrantes de los equipos involucrados en el trabajo con las UI.

Buena ___ Mala ___ Regular ___ No sé ___

Si tiene alguna observación o comentario: _____

9. Mencione la herramienta que se utiliza para el diseño y desarrollo de las UI del proyecto _____

10. ¿Cree que es necesario mejorar el proceso de diseño y desarrollo de las UI del proyecto CICPC? _____

Anexo II. Encuesta realizada a integrantes del CICPC después de aplicar DDWUI

1. ¿Qué opina sobre las interfaces de usuario (UI) del proyecto CICPC?

2. Diga las mejoras que le encuentra a las UI del proyecto CICPC.

3. ¿Cómo cree usted que labora el equipo de montadores UI del proyecto?

4. ¿Qué piensa de la nueva metodología para el diseño y desarrollo de las UI?

5. ¿Para qué le ha servido conocer las pautas de Diseño Grafico y Arquitectura de la Información establecidas el proyecto?

6. Considera que las UI realizadas en este momento en el proyecto:
 Son más fáciles de programar.
 Se ajustan a las pautas de diseño y arquitectura establecidas.
 Optimizan tiempo a los desarrolladores.
 Están buenas, pero aún les falta.
 No sirven para nada.
7. ¿Cree que sea necesario utilizar la nueva metodología para elaborar UI con calidad?
Si No No sé
8. ¿Cómo ha sido la comunicación entre los integrantes de los equipos involucrados en el trabajo con las UI, después de aplicar DDWUI? Buena Mala Regular
9. Considera que la nueva herramienta utilizada para el diseño y desarrollo de las UI del proyecto es: Buena Mala Regular
10. Diga que opina acerca del desenvolvimiento del proyecto después de aplicar la metodología DDWUI. _____

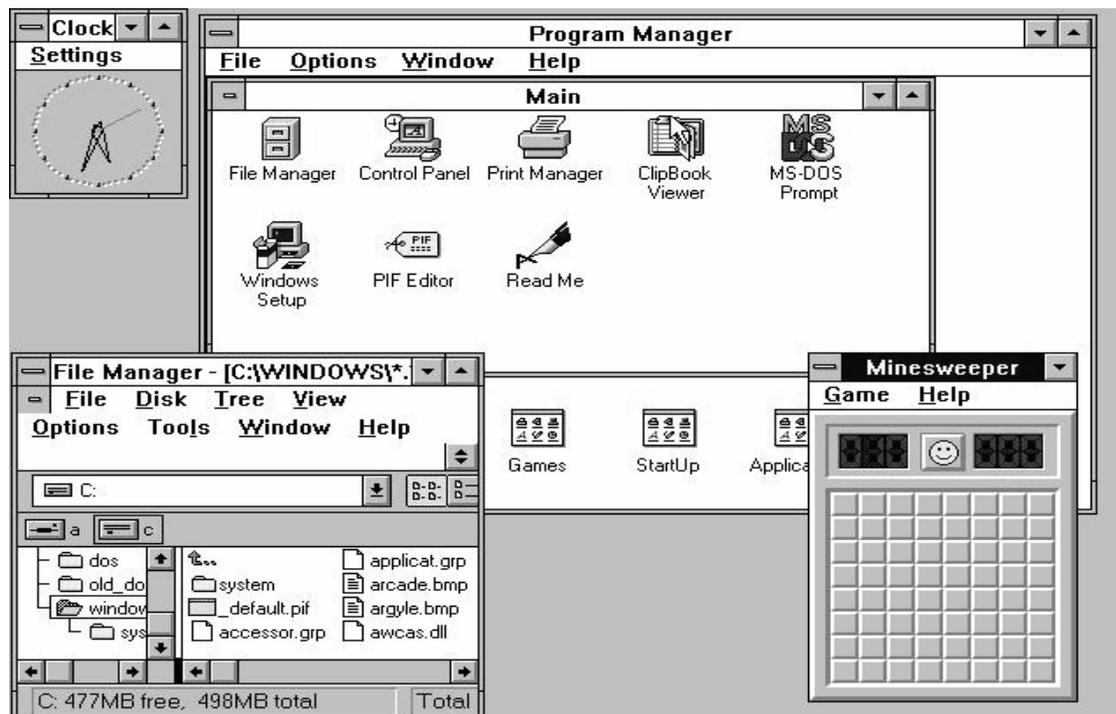
Anexo III. Interfaz por Comandos

```

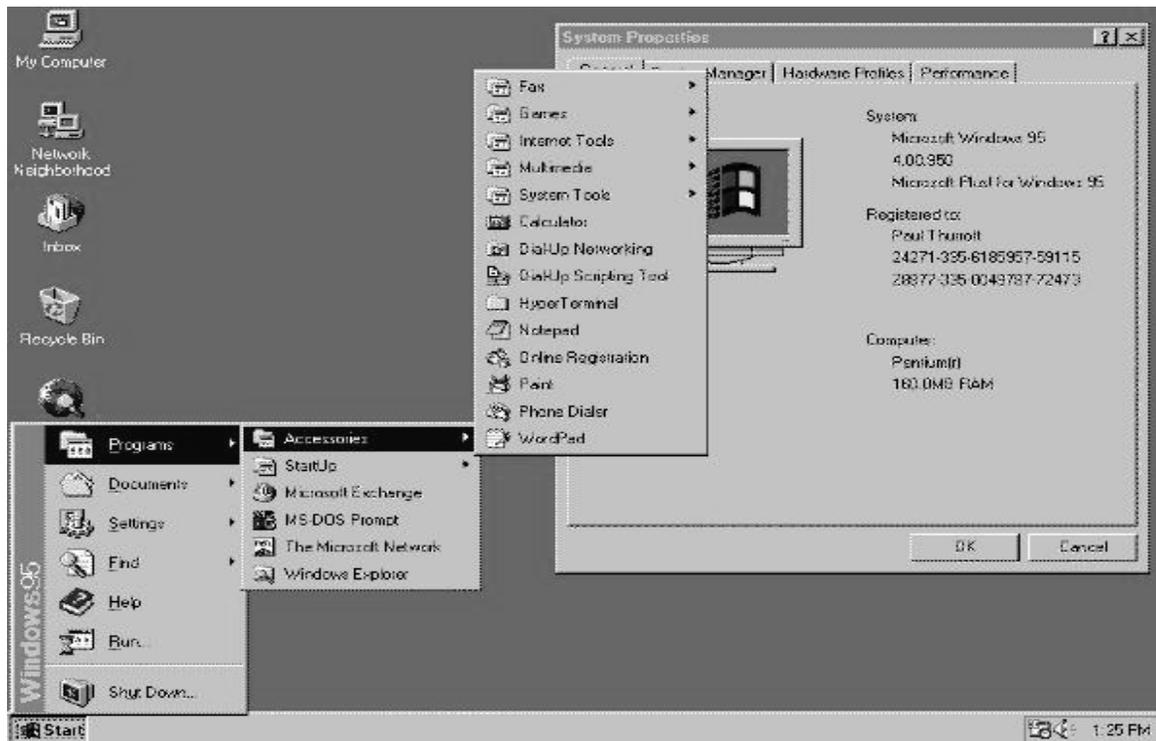
bash-2.05b$ pwd
/home/dxtone
bash-2.05b$ cd /usr/portage/app-shells/bash
bash-2.05b$ ls -al
total 68
drwxr-xr-x  3 root root  4896 May 14 12:05 .
drwxr-xr-x 26 root root  4896 May 17 02:36 ..
-rw-r--r--  1 root root 13710 May  3 22:35 ChangeLog
-rw-r--r--  1 root root  2924 May 14 12:05 Manifest
-rw-r--r--  1 root root  3720 May 14 12:05 bash-2.05b-r11.ebuild
-rw-r--r--  1 root root  3516 May  2 20:05 bash-2.05b-r9.ebuild
-rw-r--r--  1 root root  5003 May  3 22:35 bash-3.0-r11.ebuild
-rw-r--r--  1 root root  4830 May 14 12:05 bash-3.0-r7.ebuild
-rw-r--r--  1 root root  3931 May 14 12:05 bash-3.0-r8.ebuild
-rw-r--r--  1 root root  4267 Mar 29 21:11 bash-3.0-r9.ebuild
drwxr-xr-x  2 root root  4896 May  3 22:35 files
-rw-r--r--  1 root root   164 Dec 29  2003 metadata.xml
bash-2.05b$ cat metadata.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
  <herd>base-sys</herd>
</pkgmetadata>
bash-2.05b$ sudo /etc/init.d/bluetooth status
Password:
* status: stopped
bash-2.05b$ ping -q -c1 en.wikipedia.org
PING rr.chtpa.wikimedia.org (207.142.131.247) 56(84) bytes of data:
--- rr.chtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 112.076/112.076/112.076/0.000 ms
bash-2.05b$ grep -l /dev/sda /etc/fstab | cut --fields--3
/dev/sda1      /mnt/usbkey
/dev/sda2      /mnt/lp0d
bash-2.05b$ date
Wed May 25 11:36:56 PDT 2005
bash-2.05b$ lsmod
Module                Size  Used by
jmgdev                 8256  0
ipu2200              175112  0
ieee80211             44272  1 ipu2200
ieee80211_crypt       4072  2 ipu2200,ieee80211
e1000                 94464  0
bash-2.05b$ █

```

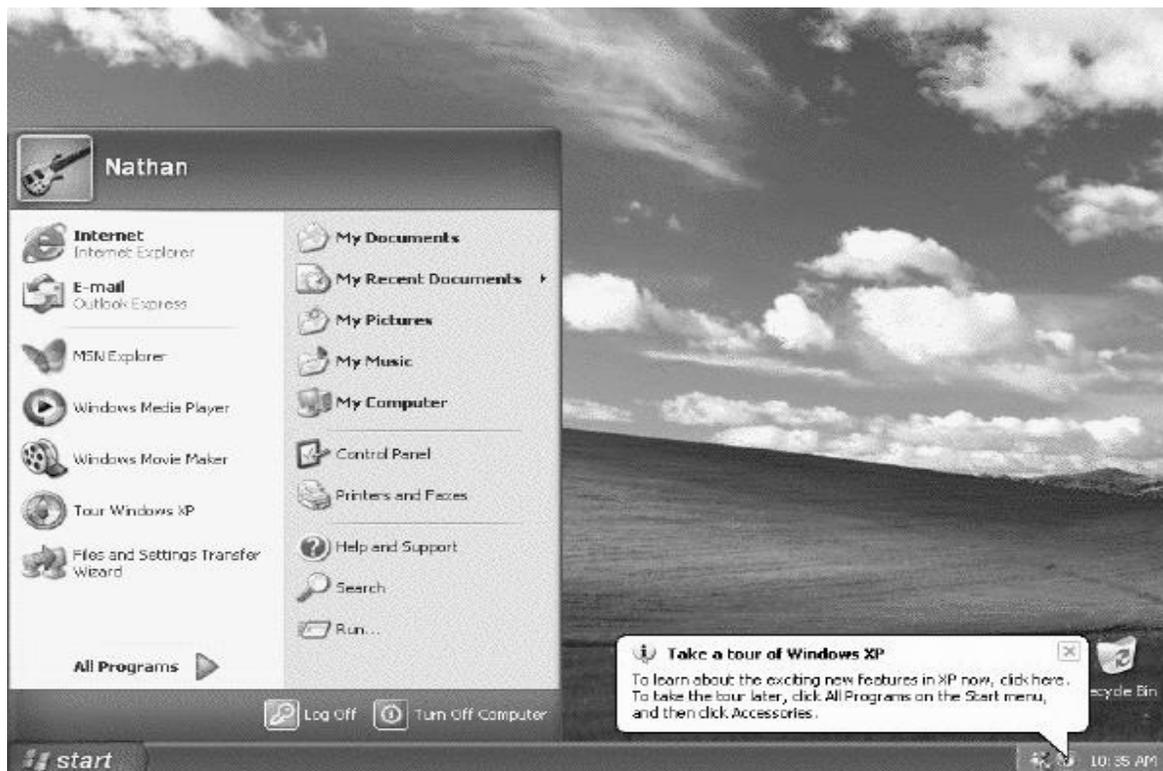
Anexo IV. Interfaz Gráfica Windows 3.11



Anexo V. Interfaz Gráfica Windows 95



Anexo VI. Interfaz Gráfica Windows XP



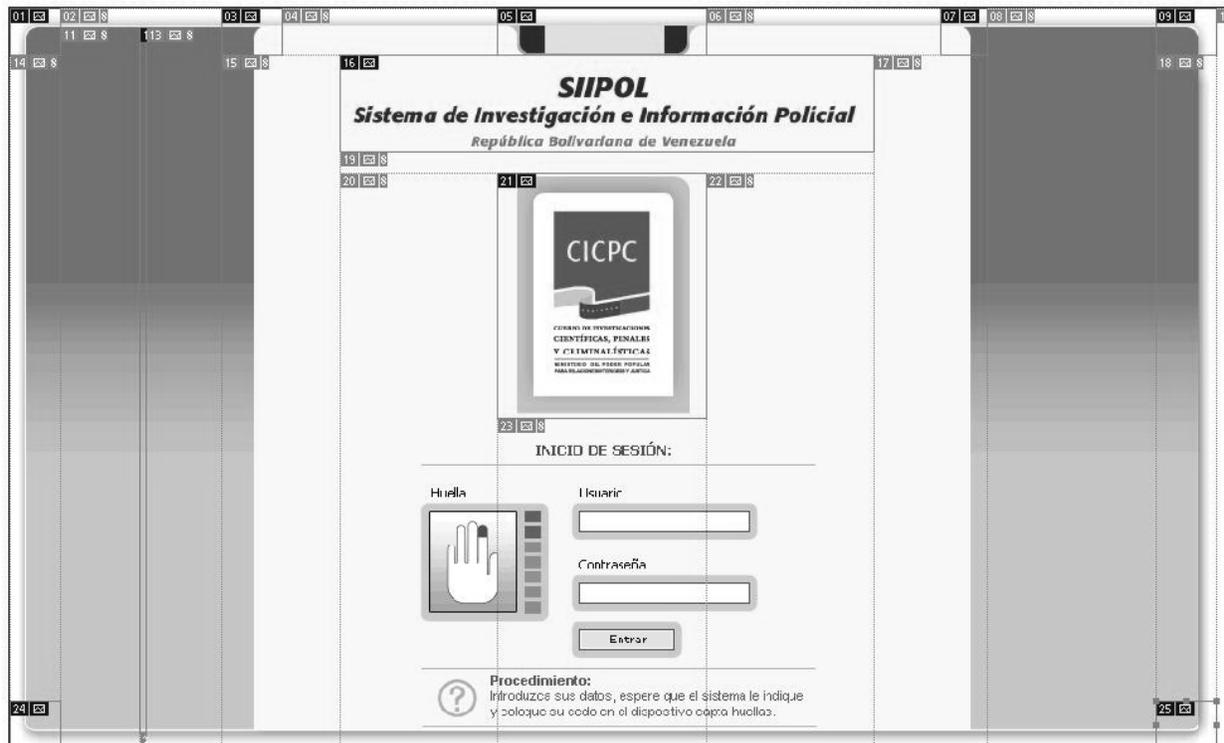
Anexo VII. Interfaz Web 1



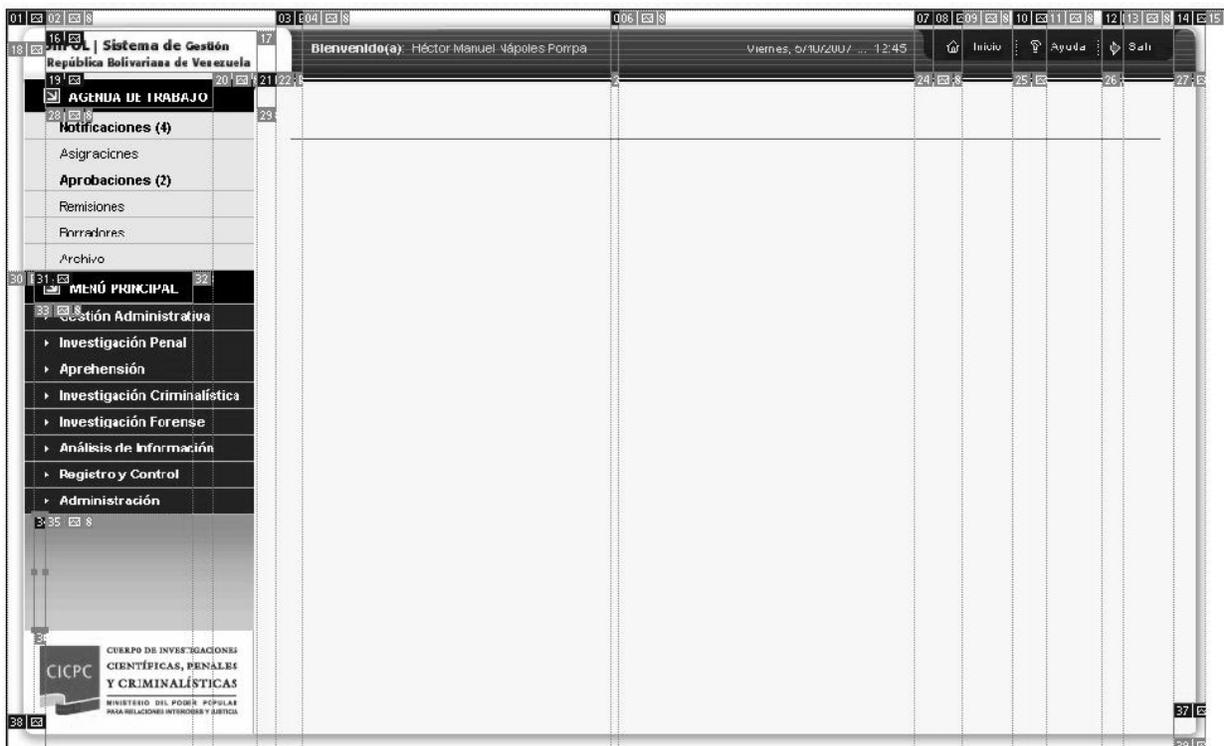
Anexo VIII. Interfaz Web 2



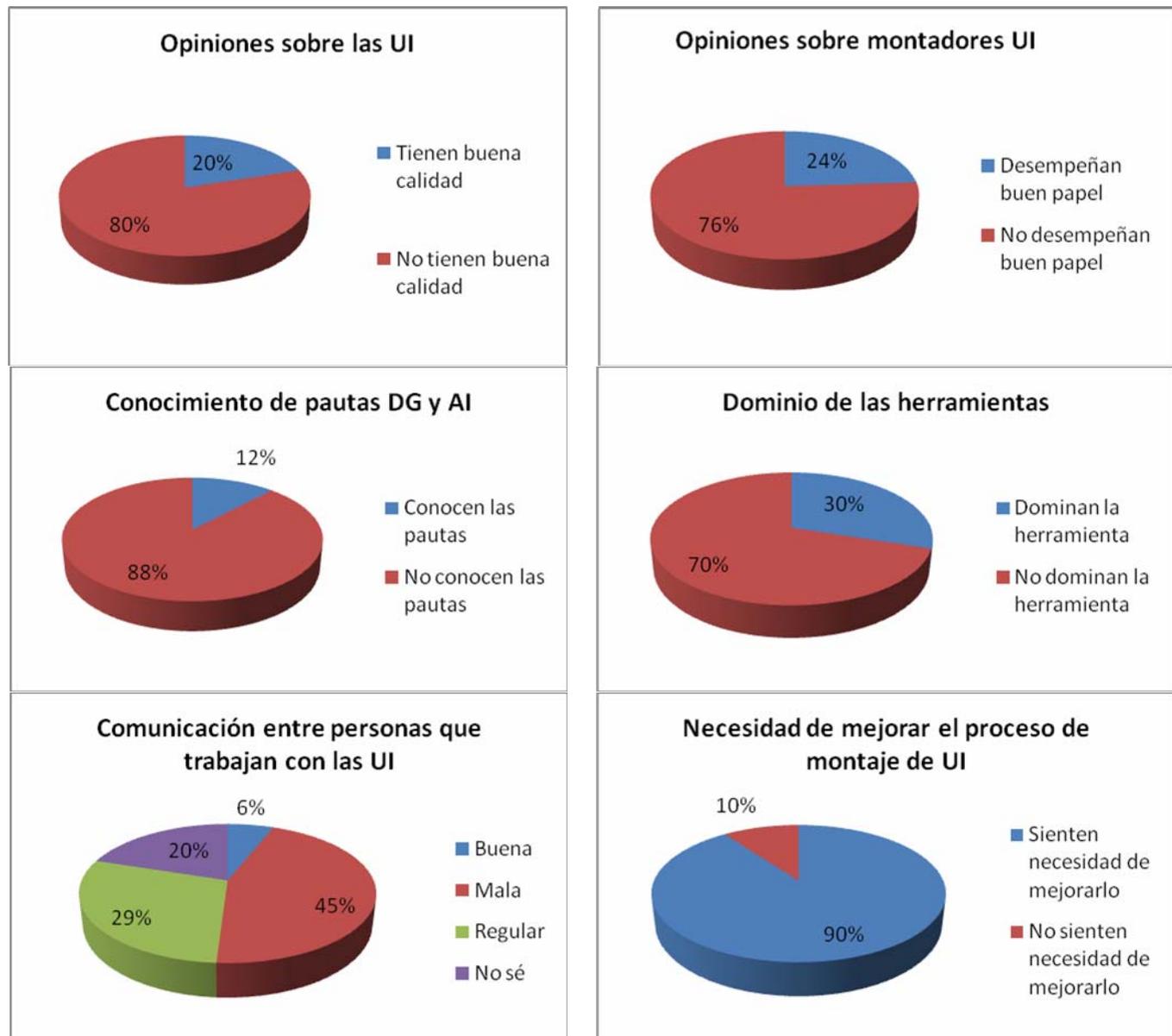
Anexo IX. Seccionamiento de imágenes mediante la herramienta Slice 1



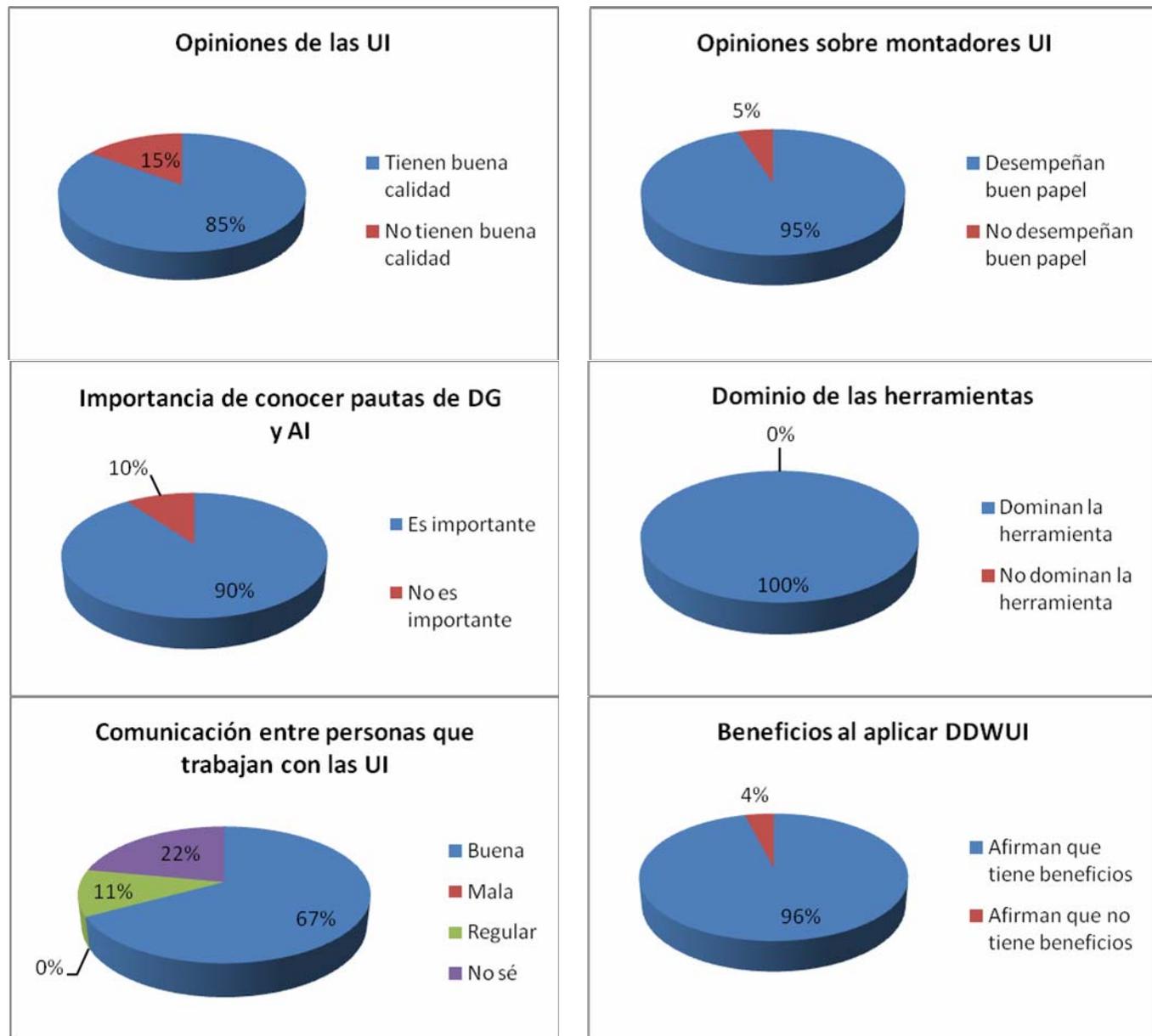
Anexo X. Seccionamiento de imágenes mediante la herramienta Slice 2



Anexo XI. Resultados de la encuesta inicial



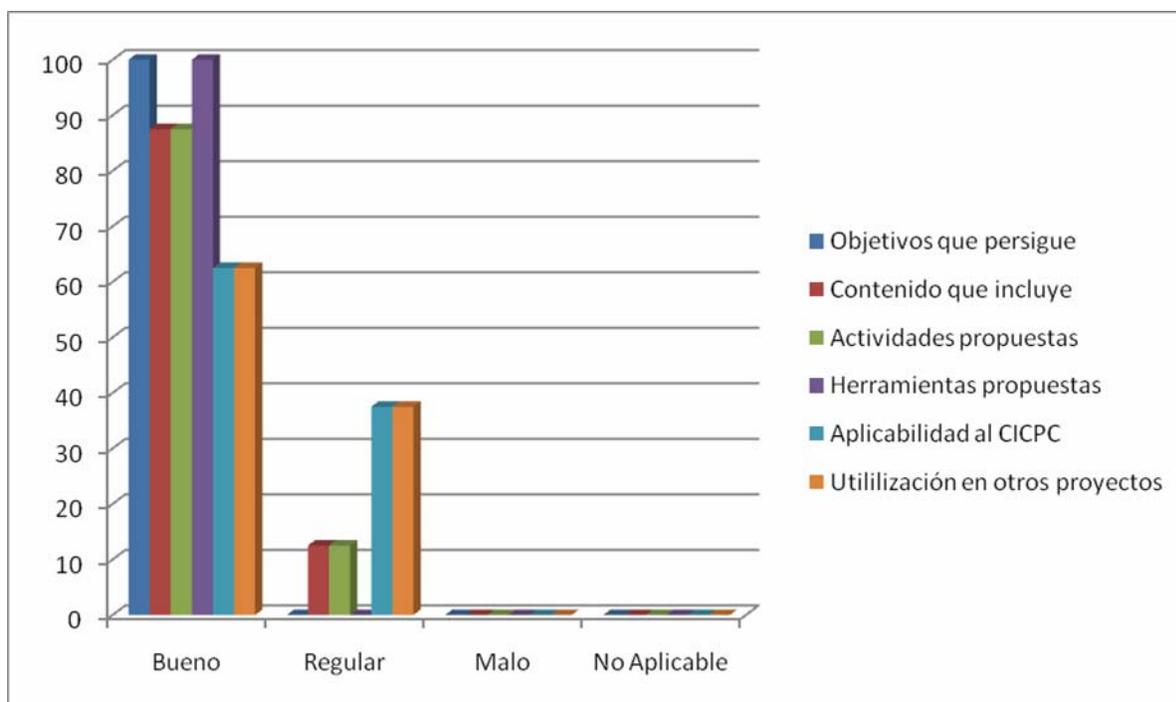
Anexo XII. Resultados de la encuesta después de aplicar DDWUI



Anexo XIII Cuestionario para la validación de la metodología DDWUI

1. Evalúe los objetivos que persigue la metodología DDWUI
Bueno ___ Regular ___ Malo___ No aplicable___
2. Evalúe los contenidos incluidos en la metodología DDWUI
Bueno ___ Regular ___ Malo___ No aplicable___
3. Evalúe las actividades propuestas dentro de la metodología DDWUI
Bueno ___ Regular ___ Malo___ No aplicable___
4. Evalúe las herramientas propuestas dentro de la metodología DDWUI
Bueno ___ Regular ___ Malo___ No aplicable___
5. Evalúe la aplicabilidad de la metodología DDWUI al proyecto CICPC de la facultad 8
Bueno ___ Regular ___ Malo___ No aplicable___
6. Evalúe la posibilidad de utilización de la metodología DDWUI en proyectos similares.
Bueno ___ Regular ___ Malo___ No aplicable___

Anexo XIV Evaluación de Expertos



Anexo XV Especificación del Caso de Uso

Especificación del Caso de Uso Gestionar Resultado de Experticia Balística.

Anexo XVI Documento de Pautas de Arquitectura de la Información

Documento de Pautas de Arquitectura de la Información.

Anexo XVII Prototipo de Interfaz de Usuario del Caso de Uso

Prototipos de UI del Caso de Uso Gestionar Resultado de Experticia Balística.

Anexo XVIII Documento de Pautas de Diseño Gráfico

Documento Manual de Pautas SIIPOL.

Anexo XIX Representación Gráfica de las pantallas

Archivo .psd

Anexo XX Muestras de pantallas

Muestra de las pantallas en formato jpg.

Anexo XXI

Escritorio de trabajo (archivo .jsp)

Anexo XXII Hojas de Estilo

Hojas de estilo (archivos .css)

Anexo XXIII Interfaces

Interfaces (Páginas web archivos .jsp)