

# Universidad de las Ciencias Informáticas

## Facultad 3



Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.

**Título:** Documentación técnica y ayuda formal del componente  
“Portal” del ERP.

**Autor:**

Aliexer Ramírez Rosabal

**Tutores:**

Ing. Dionisdel Ponce Santana.

Ing. Adrián Capdevila Cruz.

Ciudad de La Habana, Junio 2009

## AGRADECIMIENTOS

*A la Revolución Cubana y al Comandante Fidel Castro por hacer realidad el sueño de convertirme en ingeniero... A la Universidad de las Ciencias Informáticas por permitirme formar parte de esta grandísima familia, donde viví unos de los mejores años de mi vida como estudiante, y en la cual hice valiosas amistades... A mi madre que es y será la mano que me ayude a levantarme en tiempos difíciles, es además mi razón de ser... A mis queridas tías tanto la de Centro Habana como las de Playa las adoro con toda mi alma... A mis amigos del alma, hermanos de corazón Javier Limonta y Odín Vázquez.... A mi novia Yuseelis por comprenderme y apoyarme todo este corto tiempo de relación la cual la hace merecedora de mi amor y confianza... A mis tutores por apoyarme y ayudarme en esta ardua tarea y por tener tanta paciencia conmigo... A Jorgito que me a enseñado todas las cosas que se, le estoy agradecido... A Dagny por su ayuda desinteresada y en especial al grandioso equipo de proyecto en el cual compartí todo este tiempo integrado por el Noel, Aquiles, David, Avatar, Damaris, Yania Gretel, Adriana,*

*el borby, Yormis, Mile, Magdanis a los cuales no olvidaré nunca y todos aquellas magnificas personas... A mi querida amiga Yelena por apoyarme cuando ha sido necesario.*

*A todos aquellos que compartieron momentos de alegría junto a mí, los que brindaron su mano amiga y a todos en general mis más cálidos agradecimientos.*

## DEDICATORIA

*Este trabajo de tesis se lo dedico especialmente a la que ha sido capaz de formarme como persona y como un hombre con convicciones sólidas, dándome una buena educación e inculcando los valores más valiosos de los cuales hoy dispongo, me refiero a mi querida madre Gloria Ma. Rosabal Corona.*

*También a mis queridas tías Marta Roig Rosabal, Mirna Casas Corona y a mi tíaabuela Yiye, que siempre me han apoyado desde la distancia, todas mis decisiones como adulto y echo de mi una mejor persona.*

*Especialmente a mis dos hermanos, que ha sabido corregirme cuando me he equivocado, han reído junto a mí y han estado presente junto a mí en las buenas y en las malas situaciones, me refiero a mis hermanos negros Odín Vázquez Robaina y Javier Limonta Neyra. Además a flora que me ha dado ánimos cuando los he necesitado.*

<b>RESUMEN.....</b>	<b>VIII</b>
<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO I.....</b>	<b>7</b>
<b>FUNDAMENTACIÓN TEÓRICA.....</b>	<b>7</b>
INTRODUCCIÓN .....	7
1. USO Y REUTILIZACIÓN DE CSS .....	7
1.2 USO DE IMÁGENES.....	11
1.2.1 <i>Formatos de imágenes para WEB.</i> .....	11
1.2.2 <i>Características de formatos.</i> .....	12
1.2.3 <i>Comparación entre uso y rendimiento de diferentes formatos.</i> .....	14
1.3. ESTRUCTURA ARQUITECTÓNICA DE LOS OBJETOS DEL DOM.....	15
1.3.1 <i>Historia y surgimiento del DOM</i> .....	15
1.3.2 <i>Definición de DOM</i> .....	15
1.3.3 <i>Definición de BOM</i> .....	15
1.3.4 <i>Definición de AJAX</i> .....	16
1.4 SELECCIÓN DE LA TECNOLOGÍA Y MARCO DE TRABAJO .....	16
1.4.2 <i>Ventajas de la utilización del Marco de Trabajo</i> .....	23
1.5 HERRAMIENTAS .....	24
1.5.1 <i>Editores Web</i> .....	24
1.6 TRABAJO CON COOKIES. ....	28
1.6.1 <i>Definición de Cookie</i> .....	28
1.6.2 <i>Uso de las Cookies por el lado del cliente.</i> .....	29
1.7 USO DE CACHÉ EN EL NAVEGADOR.....	30
1.7.1 <i>Caché</i> .....	30
1.8 CONCLUSIONES.....	31
<b>CAPÍTULO II ARQUITECTURA DE PRESENTACIÓN.....</b>	<b>33</b>
INTRODUCCIÓN .....	33
2.1 LÍNEA BASE DE LA ARQUITECTURA DE PRESENTACIÓN.....	33
2.2 ESTÁNDAR DE USABILIDAD DE ESCENARIOS .....	44
2.3 NORMAS Y ESTÁNDARES PARA EL DESARROLLO.....	50
2.3.1 <i>Estándar de diseño de componentes de presentación</i> .....	50
2.4 CONCLUSIONES.....	72
<b>CAPÍTULO III RESULTADOS DE LA SOLUCIÓN.....</b>	<b>74</b>
INTRODUCCIÓN .....	74
3.1 REUTILIZACIÓN DE COMPONENTES PRESENTACIÓN .....	74
3.2 RENDIMIENTO .....	83
3.3 CONCLUSIONES.....	84

---

CONCLUSIONES GENERALES.....	85
RECOMENDACIONES.....	86
REFERENCIAS BIBLIOGRÁFICAS.....	87
BIBLIOGRAFÍA.....	91
GLOSARIO DE TÉRMINOS.....	93

<b>Figura 1.</b> Capa de Presentación .....	20
<b>Figura 3.</b> Sistema de etiquetado del subsistema de seguridad.....	47
<b>Figura 4.</b> Colores utilizados en la entidad del Sistema CEDRUX.....	47
<b>Figura 5.</b> Áreas de trabajos estructuradas .....	49
<b>Figura 6.</b> Componente GridPanel.....	53
<b>Figura 7.</b> Componente FormPanel (formulario).....	53
<b>Figura 8.</b> Validaciones utilizando Tool-Tips.....	54
<b>Figura 9.</b> Validaciones utilizando marcado del framework Ext.....	54
<b>Figura 10.</b> Zonificación de un formulario.....	58
<b>Figura 11.</b> División del portal en varias zonas.....	60
<b>Figura 12.</b> Pantalla de búsqueda.....	61
<b>Figura 13.</b> Pantalla de búsqueda.....	62
<b>Figura 14.</b> Pantalla típica de inserción de datos.....	62
<b>Figura 15.</b> Pantalla típica de modificación de datos.....	63
<b>Figura 16.</b> Pantalla de tipo máster-detalle.....	64
<b>Figura 17.</b> Pantalla de tipo máster-detalle-complejo.....	65
<b>Figura 18.</b> Pantalla de tipo máster-detalle-complejo-grid.....	65
<b>Figura 19.</b> Pantalla de tipo máster-detalle-grid.....	66
<b>Figura 20.</b> Pantalla de filtro avanzado.....	67
<b>Figura 21.</b> Componente combo-iframe-árbol.....	75
<b>Figura 22.</b> Componente UCID-win-iframe.....	79

## RESUMEN

En la Universidad de las Ciencias Informáticas el proceso de documentación en la mayoría de los proyectos productivos es escaso o nulo en ocasiones. Dentro de estos proyectos se encuentran unos de gran envergadura como es el ERP - CUBA, que innova en cuanto a su metodología y arquitectura con el fin de obtener un producto de excelente calidad y que traiga consigo un gran alcance tecnológico para el país. El desconocimiento del trabajo de un proyecto innovador como este, si no está explícitamente documentado cada fase, o parte de su desarrollo, o cómo se debe trabajar en unos de los componentes claves del sistema Cedrux, trae consigo el incremento de dudas a la hora de reutilizar sus aspectos técnicos. Además puede provocar otros problemas que pueden llevar al traste el desarrollo del mismo. El presente trabajo tiene como principal objetivo la elaboración de la documentación técnica y ayuda formal del componente Portal, del proyecto ERP-CUBA.

**Palabras Claves:** Componente Portal



## INTRODUCCIÓN

Internet desde su nacimiento ha sido un instrumento de comunicación y de distribución de información, tiene sus raíces en un proyecto de defensa militar en Estados Unidos de América (USA) para mantener las comunicaciones en caso de una posible guerra nuclear a finales de los 60. Para el intercambio de información se crearon distintos servicios como Telnet<sup>1</sup>, FTP (File Trasference Protocol), HTTP<sup>2</sup>, servicios de mensajería, la World Wide Web (www), etc. Este último llamado simplemente como Web, se transformó en el mecanismo para compartir información más utilizada de Internet mediante su cara visible, las páginas Web.

A mediados de los 90, el crecimiento demográfico de Internet fue exponencial en los EEUU y en Europa. Así, durante la segunda mitad de la década de los 90 comenzó a nacer un conjunto de empresas que ofrecían servicios de directorios de información. Estas buscaron ordenar el caos informativo que se estaba generando mediante una estructura de árbol de directorios que organizaba las páginas Web por contenidos [1]. Empresas como Yahoo nacieron en esta época, y fueron conocidos como los primeros Portales Web.

Portal es un nuevo término, que se utiliza para referirse a un Sitio Web que sirve de punto de partida para iniciar actividades de navegación en Internet, en la cual se visita con frecuencia y al que generalmente se designa como página de inicio en el navegador, trátase de Microsoft® Internet Explorer o Netscape® Communicator.

Existen portales genéricos y especializados, conocidos estos últimos como portales de nicho. Algunos de los portales más conocidos son el de Yahoo!, Google, Excite, Netscape, Lycos, CNET, Microsoft Network, y el de America Online. Ejemplos de portales especializados tales como Garden.com (para jardinería), Fool.com (para inversionistas), y SearchNT.com (para administración de Windows NT).

La definición de Portal se refiere a un sitio que es punto de partida para la navegación en Internet. Aún para los Portales especializados, existen requisitos que cumplir para recibir la denominación de Portal, lo cual no se limita simplemente a contar con un Sitio Web robusto.

Los portales surgieron por la necesidad de integrar y clasificar el exceso de información distribuido a través de Internet, nacieron como un punto de entrada donde buscar información. Un fenómeno similar al de Internet ocurrió con las Intranets en empresas, se creó un exceso de información, difícil de navegar y con poco valor para los empleados [2]. Esta primera generación de portales en el caso empresarial se desarrolla con el objetivo de clasificar y proveer de un único punto de acceso a la información que se encontraba distribuida a través de la empresa (tanto estructurada como no estructurada), típicamente noticias, información de empleados, documentos de planificación y otros contenidos estáticos. Los portales como Yahoo cuya experiencia se limitaba a páginas Web y servicios de Internet como información versátil y meteorológica, los portales empresariales buscaban ofrecer una perspectiva organizada y enfocada a ayudar en las tareas que comúnmente realizaban la mayoría de sus usuarios: documentos, informes desde bases de datos, aplicaciones empresariales y un sin fin de recursos corporativos [3].

En la segunda generación de portales, cada usuario puede personalizar su vista del portal. Si en la generación anterior se mostraban todas las categorías para buscar información, en esta generación solo se muestra las categorías que el usuario desea ver. En este caso un usuario necesita estar registrado en el portal e ingresar mediante un usuario y contraseña [4].

A medida que las organizaciones construyeron sus portales, integrando así los sistemas y contenidos existentes, descubrieron que era necesario mucho más que una tecnología de integración para garantizar el éxito de la implementación de sus portales [5]. Los portales se expandieron para incluir un conjunto más amplio de tecnologías, esta tercera generación comenzó a incluirlas como herramientas de colaboración. Estas herramientas permitieron realizar tareas esenciales en la organización sin que sus integrantes tuviesen la necesidad de utilizar otro software independiente. Estas incluyeron correo electrónico, calendarios, chat, etc.

En la cuarta generación los portales están basados en roles de usuarios que permiten manejar funciones corporativas específicas. Esto implica la integración de aplicaciones empresariales en el portal a fin de que los usuarios puedan leer, escribir y actualizar datos corporativos [4].

Hasta el 2003 las generaciones de portales evolucionaron junto con el crecimiento en el interés, sin embargo, surgió un problema cuando se hizo necesario hacer compatibles los diferentes

componentes desarrollados por cada empresa. Para dar solución a este problema fue lanzada en Octubre del 2003 la JSR 168<sup>3</sup>, la cual define como desarrollar los componentes que integran un portal. Esta nueva generación representa una mayor evolución en funcionalidad, permitiendo conectar diferentes portales a través de estándares de desarrollo. Aunque las primeras generaciones se centraron en la agregación de contenido, la personalización básica y la integración a nivel muy básico, esta generación fue la primera en dar una real funcionalidad, dando a los usuario que componen el portal (empleados, clientes, proveedores y socios comerciales) acceso a toda la información empresarial disponible (contenido, aplicaciones, servicios y procesos de negocio) desde cualquier dispositivo como browsers, teléfonos o PDA's<sup>4</sup> [5].

La actual generación de portales proporciona una arquitectura modular orientada a servicios para el desarrollo de portales [5]. Son los más utilizados hoy en día y ofrecen a las empresas muchas de las capacidades necesarias para introducir SOA<sup>5</sup> en una organización.

Actualmente Cuba no se encuentra ajena a este tipo de clasificación de aplicaciones Web, ya que en muchas esferas del archipiélago cubano se desarrollan numerosos portales, específicamente en la Universidad de las Ciencias Informáticas (UCI) donde estudiantes y profesores colaboran entre sí en la construcción de software especialmente en el desarrollo de aplicaciones Web, dentro de estas los portales; ya sean con objetivos económicos u organizativos. En este centro universitario se lleva a cabo hoy en día el desarrollo de una de las aplicaciones informáticas más importantes para Cuba, se refiere a un Sistema Integral de Gestión de Entidades (CEDRUX), basándose en la estructura de un ERP<sup>6</sup> (Enterprise Resource Planning), donde, definido por el equipo de arquitectura del proyecto ERP-Cuba se desarrolla sobre uno de los componentes que forman parte del núcleo de la aplicación, denominado "Portal". Con el desarrollo de este sistema se pretende lograr una solución altamente parametrizable y modular de alcance nacional para la gestión integral de los procesos contables y financieros de las entidades cubanas. Esta brindará una mejor seguridad, oportunidad y fiabilidad en la gestión de la información de las entidades nacionales y de los diferentes niveles de gobierno.

En la Universidad de las Ciencias Informáticas el proceso de documentación en la mayoría los proyectos productivos es escasa o nula en ocasiones, dentro de estos proyectos se encuentran unos de gran envergadura como es el ERP - CUBA, que innova en cuanto a su metodología y arquitectura con el fin de obtener un producto de excelente calidad y que traiga consigo un gran alcance tecnológico para el país. El ERP - CUBA se enfoca en un producto centralizado que abarca diversas entidades del país específicamente en el sector de la economía. Para el desarrollo del sistema integral de gestión de entidades (CEDRUX), se trabajó sobre el (componente "Portal")<sup>7</sup>, donde la manipulación de la información es más fluida y operable. El desconocimiento del trabajo de un proyecto innovador como este, si no está explícitamente documentado cada fase, o parte de su desarrollo, o cómo se debe trabajar en el componente "Portal", trae consigo el incremento de dudas a la hora de reutilizar sus aspectos técnicos, lo que conlleva un atraso en lo que se desea desarrollar debido a que el tiempo de estudio es más prolongado y se podría llegar a incumplir con la entrega o despliegue de futuros productos.

Dada la situación planteada el **problema a resolver** queda formulado de la siguiente manera:

Los desarrolladores no cuentan con la documentación técnica ni ayuda formal de manera estructurada del componente Portal que facilite el entendimiento del funcionamiento del mismo.

Para la solución del problema planteado el **objeto de estudio** lo constituye la arquitectura de software cuyo campo de acción queda enmarcado en la arquitectura de presentación.

Para llevar a cabo este trabajo se plantea como **objetivo general**: Elaboración de la documentación técnica y ayuda formal del componente Portal del ERP

Para darle cumplimiento a dicho objetivo se trazan los siguientes **objetivos específicos**:

1. Definir características arquitectónicas del componente de integración de la capa de presentación.
2. Identificar los contenidos para incluir en la definición de la documentación técnica.
3. Describir características arquitectónicas

4. Describir las pautas de los elementos de diseño del componente.
5. Describir la iconografía a utilizar.
6. Describir las pruebas de conceptos realizadas al componente.
7. Construir API (Application Programming Interface o Interfaz de Programación de Aplicaciones) del componente “Portal” del ERP.

### **Idea a defender:**

Si se cuenta con una herramienta fácil de utilizar que contenga la documentación técnica y ayuda formal del componente Portal garantizará un mejor entendimiento del funcionamiento del mismo.

Los resultados esperados de este trabajo de diploma son:

- API del componente “portal” para la integración de la capa de presentación de la arquitectura del ERP

El presente trabajo de diploma se estructura en 3 capítulos:

### **Capítulo 1: Fundamentación teórica.**

En este capítulo se realiza un estudio del estado del arte de temas de interés para la investigación, como son: el uso y reutilización de las hojas de estilos (CSS), el uso de imágenes, sus formatos para el diseño Web. Además se hace referencia a la historia y surgimiento de los objetos Java scripts y la definición de Ajax, la selección de la tecnología y marco de trabajo, las posibles herramientas a utilizar en la solución de dicha investigación entre otros.

### **Capítulo 2: Arquitectura de presentación.**

En esta sección se aborda sobre los aspectos significativos del trabajo de diploma en curso, enfatizando sobre la arquitectura de presentación aplicada al componente Portal. En este se

emprende temas relacionados como: la línea base de la arquitectura de presentación, el estándar de usabilidad de los escenarios y las normas y estándares para el desarrollo.

### **Capítulo 3: Resultado de la solución.**

Se plasma los resultados de la solución, como son: la reutilización de los componentes de presentación y una breve descripción sobre las pruebas de conceptos realizadas al componente Portal.

## CAPÍTULO I

### FUNDAMENTACIÓN TEÓRICA

#### INTRODUCCIÓN

En el presente capítulo se abordará sobre la fundamentación teórica de algunos conceptos que son necesarios para la realización del trabajo de diploma. Primeramente se describen algunas definiciones o conceptos sobre los aspectos esenciales de la arquitectura de presentación como el uso y reutilización de las hojas de estilos (CSS), y otros elementos de importancia sobre el mismo. Posteriormente se aborda sobre los distintos formatos de imágenes para el desarrollo Web, así como las características particulares de cada uno de ellos y otros tópicos a evaluar. También en este capítulo se abordarán los temas de Ajax, DOM y BOM, y cada unos de estos aspectos con sus elementos más importantes a considerar. Se mencionarán también algunos puntos importantes para este capítulo como son el uso de las cookies entre otras cuestiones.

#### 1. USO Y REUTILIZACIÓN DE CSS

##### 1.1 Surgimiento de las CSS (Cascading Style Sheets u Hojas de Estilo en Cascada).

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML (Standard Generalized Markup Language o Lenguaje de Marcación Generalizado), alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

La separación de los contenidos y su presentación presenta numerosas ventajas, ya que obliga a crear documentos HTML (Hypertext Markup Language o lenguaje de marcado de hipertexto) /XHTML (Extensible Hypertext Markup o lenguaje extensible de marcado de hipertexto) bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Si el lenguaje HTML/XHTML se utiliza para marcar los contenidos, es decir, para designar lo que es un párrafo, lo que es un encabezado o lo que es una lista de elementos, el lenguaje CSS se utiliza para definir el aspecto

de todos los contenidos, 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.

El gran impulso de los lenguajes de hojas de estilos se produjo con el advenimiento de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores. El W3C (World Wide Web Consortium) propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron 9 propuestas. Las 2 propuestas que se tuvieron en cuenta fueron la CHSS (*Cascading HTML Style Sheets*) y la SSP (*Stream-based Style Sheet Proposal*).

La propuesta CHSS fue realizada por Håkon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS.

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como

“CSS nivel 1”.

El 12 de Mayo de 1998, el grupo de trabajo de CSS del W3C publica su segunda recomendación oficial, conocida como “CSS nivel 2”. La siguiente recomendación, conocida como “CSS nivel 3”, continúa en desarrollo desde 1998 y hasta el momento sólo se han publicado borradores.

Varios navegadores soportan CSS 2, tales como: Firefox 3.5 (con su motor Gecko), Opera 9.5 (con su motor Presto), Internet Explorer 8.0 (con su motor Trident) entre otros.

Desde la publicación de la versión CSS 2, se han publicado pequeñas correcciones de errores y variaciones en el estándar, hasta llegar a la actual versión CSS 2.1 [6].



### 1.1.1 Definición de CSS

CSS es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. Es la mejor forma de separar los contenidos y su presentación y es imprescindible para la creación de páginas Web complejas. [6]

### 1.1.2 Eficiencia de uso de CSS en la WEB.

Al igual que XHTML hace que el código de las páginas Web sea más limpio y claro a los ojos de los robots de búsqueda.

Por otra parte, con CSS se logra aumentar la densidad de las palabras claves dentro de los contenidos, ya que las etiquetas ocupan mucho menos espacio. Esto también supone un menor peso para las páginas Web, lo cual agradecen tanto los robots de búsqueda como los usuarios finales. Adicionalmente se puede cambiar rápidamente los estilos de ciertas palabras, modificando la importancia que se les otorga ante los robots de búsqueda.

También, aunque dentro de la página, los elementos se muestran en un orden (Ej.: cabecera, menú superior, menú izquierdo, contenido principal), a través de CSS se logra que en el código (los que ven los buscadores) aparezcan los elementos en el orden que se desea (contenido principal, menú izquierdo, menú principal, cabecera). Otras de las características que tributan la eficiencia de uso de CSS es que conserva el ancho de banda del usuario, lo que acelera la carga de páginas, especialmente en conexiones telefónicas. También reduce la sobrecarga del servidor. Se añade a estas características la reducción del tiempo de diseño y programación [7].

### 1.1.3 Ventajas de uso

- **Separación del contenido y presentación.**

Las hojas de estilo generalmente se encuentran en archivos separados del código principal (HTML, por ejemplo). Esto permite que en un equipo de trabajo, tanto el programador como el diseñador

puedan realizar sus tareas de forma independiente aunque paralelas, sin correr el riesgo de que haya interferencias entre ambos, y para ello no alterará el resultado final.

- **Flexibilidad.**

Las hojas de estilo permiten cambiar en cualquier momento alguna parte o la totalidad del diseño de las páginas con sólo modificar la hoja de estilo, sin que modifique el contenido.

- **Optimización de los tiempos de carga y de tráfico en el servidor.**

Al dividir el contenido y apariencia se obtiene archivos más ligeros, y reportándose dos beneficios: por un lado, se reduce notablemente los tiempos de carga del sitio en el navegador. También se une la capacidad de éste para mantener la hoja de estilo en caché.

- **Precisión o elasticidad.**

Desde el momento en que se usa CSS, el tamaño y posicionamiento de los elementos que formen las páginas podrá ser exacto, indicándole al navegador en qué píxel debe colocar la imagen, o qué alto o ancho debe mostrar.

Pero al mismo tiempo, se emplea medidas variables o relativas que permiten expandir el contenido hasta ocupar la totalidad de la ventana de navegación, o contraerla a sólo una parte de la misma, con independencia de la resolución de pantalla del usuario.

- **Accesibilidad y estructuración.**

La combinación de CSS y marcadores descriptivos posibilita que la página se vea correctamente con o sin hoja de estilos, donde la información se mantiene estructurada y ordenada.

Lográndose la accesibilidad sin ningún tipo de problemas tanto por navegadores antiguos o sin soporte para CSS, como para personas con algún tipo discapacidad.

- **Limpieza del código fuente.**

Si se escribe en una hoja de estilo independiente, el código fuente de la Web va a resultar menos tedioso y se agiliza las tareas de localización de las líneas que se busquen.

- **Permite la diferenciación de estilos para imprimir / visualizar en pantalla.**

El uso de CSS permite maquetar separadamente el contenido de la Web para ser mostrado en pantalla o para ser impreso, dicho maquetado permite que siempre se tenga una apariencia limpia, ordenada y agradable visualmente [8].

## 1.2 USO DE IMÁGENES.

### 1.2.1 Formatos de imágenes para WEB.

Al igual que cualquier otro tipo de información, las imágenes en un ordenador se encuentran codificadas por un patrón o formato específico. Existen numerosos formatos gráficos para la Web, los cuales se dividen en dos tipos de gráficos estos son:

Gráficos de mapa de bits basados en una retícula de píxeles.

Gráficos vectoriales basados en información numérica sobre la geometría de los mismos.

Actualmente los navegadores por sí solos soportan gráficos de mapa de bits. Dentro de los gráficos de mapa de bits o como también se le conoce en inglés como Bitmap se encuentran:

#### **BMP (Bitmap o Mapa de Bits):**

Formato desarrollado para aplicaciones Windows y, por tanto, muy utilizado. Las imágenes de mapa de bits se denominan así porque el formato crea un mapa de los píxeles línea a línea. El formato BMP no sufre pérdida y, por tanto, resulta adecuado para guardar imágenes que se desean manipular o ampliar posteriormente.

**GIF (Graphic Interchange Format o Formato de Intercambio Gráfico)**

El formato GIF fue desarrollado por CompuServe y está pensado especialmente para la transmisión de imágenes a través de servicios online y es comprimido sin pérdida de información.

**JPEG (Joint Photographic Expert Group o Grupo Conjunto de Expertos en Fotografía).**

El formato JPEG creado por fotógrafos especialistas, resulta realmente idóneo su uso en documentos que contengan millones de colores, documentos escaneados, tal es el caso de fotografías de grandes degradaciones de matiz. Igualmente resulta efectivo en documentos en escala de grises. Es adecuado su empleo en imágenes de tono continuo y suaves transiciones de color. Por el contrario los resultados pueden ser insatisfactorios si los documentos son indexados.

**PNG (Portable Network Graphics)**

Al igual que JPEG, este novedoso formato creado por CompuServe y Unisys permite trabajar con documentos que contengan millones de colores de alta resolución al mismo tiempo que reduce la pérdida de información de la imagen.

**TIF o TIFF (Tagged Image File Format)**

Este formato puede generar archivos comparativamente grandes, pero con una calidad de imagen excelente. Los archivos TIFF pueden almacenar cualquier nivel de profundidad de color de 1 a 32 bits y es, sin duda, el formato ideal si se tiene previsto editar o imprimir una imagen [9].

**1.2.2 Características de formatos.****GIF**

Las imágenes GIF son, probablemente, el formato de imagen con mayor soporte en la Web. Este tipo de imágenes soportan lo que se conoce como transparencia. Un bit se reserva para la transparencia, lo que significa que puede configurarse un color como transparente. La transparencia permite ver el fondo sobre el que esté colocada la imagen, haciendo posible efectos múltiples muy complejos. Existen dos tipos básicos de formatos GIF: GIF 87 y GIF 89a. Ambos tipos de formatos GIF soportan color de 8 bits (256 colores), utilizan el esquema de compresión sin pérdida LZW (Lempel-Ziv-Welch) y, generalmente, tienen una extensión .gif. El formato GIF 89a soporta también la transparencia y la animación. El esquema de la codificación de compresión run-

length, utilizado por este, funciona bien con grandes superficies de color continuo, así que, dicho formato es una compresión muy eficiente con ilustraciones planas.

### **JPEG**

Es un formato de compresión de imagen con pérdidas diseñado para utilizar con imágenes de fotografías que pueden contener miles, o incluso millones, de colores o de tonalidades grises. Puesto que el formato JPEG es un formato de imagen con pérdidas, existe un cierto compromiso entre la calidad de la imagen y el tamaño del archivo. Sin embargo, el formato JPEG almacena imágenes de color de 24 bits en un espacio significativamente menor que con GIF, ahorrando así espacio el disco duro del ordenador y disminuyendo el tiempo de descarga. El formato JPEG puede guardarse tanto en el formato RGB como en el formato CMYK.

### **PNG**

Al igual que JPEG, este novedoso formato creado por CompuServe y Unisys permite trabajar con documentos que contengan millones de colores de alta resolución al mismo tiempo que reduce la pérdida de información de la imagen. Diseñado para imágenes y fotografías tiene la particularidad de conservar

Todos los valores referentes al color y canales alfa que posea el documento, permitiendo de este modo crear transparencias en los documentos al modo que lo realiza GIF89a. Por lo que resulta efectivo en documentos escaneados, en imágenes en escala de grises y en imágenes de tono continuo y suaves transiciones de color, aunque puede ser igualmente utilizado en documentos indexados.

El formato PNG está basado en la tecnología que tendrá el formato GIF24 llamado a ser el sucesor del formato GIF87a e incluso de GIF89a. Mejora las capacidades gráficas del GIF clásico pudiendo trabajar igualmente con imágenes de 24 bits (más de 16 millones de colores) sin pérdida de calidad de imagen debido al tipo de compresión LZ778 que utiliza. Además, resuelve el problema de las diferentes visualizaciones de la imagen según la plataforma en la que se trabaje, debido a las diferentes cromaticidades y valores gamma existentes. No obstante, tiene como contrapartida no dar soporte para la creación de animaciones a diferencia del formato GIF89a. [10]

1.2.3 Comparación entre uso y rendimiento de diferentes formatos.

JPG	GIF	PNG
<ul style="list-style-type: none"> <li>• Número de colores: 24 bits color o 8 bits B/N</li> <li>• Muy alto grado de compresión.</li> <li>• Formato de compresión con pérdida.</li> <li>• No permite transparencia</li> <li>• No permite animación.</li> </ul>	<ul style="list-style-type: none"> <li>• Permite hasta 256 colores</li> <li>• Formato de compresión sin pérdida.</li> <li>• Carga progresiva</li> <li>• Permite transparencia</li> <li>• Animación simple</li> </ul>	<ul style="list-style-type: none"> <li>• Color indexado hasta 256 colores y no indexado hasta 48 bits por pixel.</li> <li>• Mayor compresión que el formato GIF (+10%)</li> <li>• Compresión sin pérdida.</li> <li>• Transparencia variable.</li> <li>• No permite animación.</li> </ul>
<b>OTROS ASPECTOS</b>		
<ul style="list-style-type: none"> <li>• Ideal para grandes imágenes fotográficas, principalmente para impresión y para la Web.</li> <li>• La calidad disminuye cada vez que se guarda el archivo, por lo que no conviene emplearlo para archivar originales.</li> </ul>	<ul style="list-style-type: none"> <li>• Ideal para páginas Web con gráficos pequeños.</li> <li>• No es recomendable para colocar fotografías de alta calidad en la Web o para archivar originales.</li> </ul>	<ul style="list-style-type: none"> <li>• Produce archivos excesivamente grandes para la reproducción de fotografías o cuadro comparados con los equivalentes archivos JPG para la Web.</li> <li>• No permite crear pequeñas animaciones como el formato GIF [9].</li> </ul>

## 1.3. ESTRUCTURA ARQUITECTÓNICA DE LOS OBJETOS DEL DOM.

### 1.3.1 Historia y surgimiento del DOM

#### (Document Object Model o Modelo de Objetos de Documento)

Cuando JavaScript <sup>9</sup> fue introducido inicialmente en los navegadores, se requería algún tipo de interfaz para permitir el acceso a los elementos en la página vía scripting. La guerra entre navegadores que hubo entre el Netscape Navigator y el Internet Explorer de Microsoft creó graves problemas para los programadores de páginas Web, ya que, aunque ambos navegadores utilizaban Javascript como lenguaje de programación, los objetos no se comportaban de la misma forma, lo que obligaba con frecuencia a programar dos veces las páginas, una para el Netscape, y otra para el Internet Explorer; aún así, seguían teniendo problemas, ya que no todas las versiones de un mismo navegador se comportaban igual.

Finalmente, el W3C, encargado de definir los estándares de la Web, decidió crear un modelo de objetos único, el DOM, para que todos los fabricantes pudieran adoptarlo, facilitando la compatibilidad plena entre ellos. Afortunadamente, la mayor parte de los vendedores empezaron a adoptar el estándar del DOM fijado por el W3C. No obstante, Microsoft ha añadido su propia extensión al DOM, creando problemas de interoperabilidad para los navegadores Web [11].

### 1.3.2 Definición de DOM

El Modelo de Objetos de Documento o DOM, es la interfaz que permite acceder y manipular, mediante la programación, los contenidos de una página Web (documento HTML o XML.). Proporciona una representación estructurada, orientada a objetos, de los elementos individuales y el contenido de una página, con métodos para recuperar y fijar las propiedades de estos. Además, proporciona métodos para agregar y eliminar dichos objetos, permitiendo crear contenido dinámico [12].

### 1.3.3 Definición de BOM

Las versiones 3.0 de los navegadores Internet Explorer y Netscape Navigator introdujeron el concepto de Browser Object Model (BOM), empleado para acceder y modificar las propiedades de las ventanas del propio navegador. Mediante BOM, es posible redimensionar y mover la ventana

del navegador, modificar el texto que se muestra en la barra de estado y realizar muchas otras manipulaciones no relacionadas con el contenido de la página HTML. El mayor inconveniente del BOM es que su comportamiento no es estándar entre los navegadores [13].

#### 1.3.4 Definición de AJAX

AJAX, acrónimo de Asynchronous Java Script And XML (Java Script y XML asíncronos, donde XML es un acrónimo de Extensible Markup Lenguaje), es una técnica de desarrollo Web para crear aplicaciones interactivas. Éstas se ejecutan en el navegador del usuario y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla [14].

## 1.4 SELECCIÓN DE LA TECNOLOGÍA Y MARCO DE TRABAJO

### 1.4.1 Características del marco de trabajo.

Las características del marco de trabajo se conceptualizan de las propias definiciones de los componentes y clases de desarrollo, dándole así un perfil técnico al marco de trabajo del proyecto ERP –CUBA. También el componente Portal forma parte del núcleo del marco de trabajo, en el cual interactúa espontáneamente con otros componentes definidos en el marco de trabajo. A continuación se le especifican los componentes y las clases fundamentales definidos por el equipo de arquitectura de dicho proyecto:

- **Componente Zend Ext.:**

Este componente es una extensión del Zend framework el cual reutiliza el Modelo-Vista – Controlador<sup>10</sup> (MVC). [15]

- **IOC (Inversión de Control):**

Es un concepto junto a unas técnicas de programación en las que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la



interacción se expresa de forma imperativa haciendo llamadas a procedimientos (procedure calls) o funciones. Tradicionalmente el programador especifica la secuencia de decisiones y procedimientos que pueden darse durante el ciclo de vida de un programa mediante llamadas a funciones. En su lugar, en la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir. El flujo habitual se da cuando es el código del usuario quien invoca a un procedimiento de una librería. La inversión de control sucede cuando es la librería la que invoca el código del usuario. Típicamente sucede cuando la librería es la que implementa las estructuras de alto nivel y es el código del usuario el que implementa las tareas de bajo nivel. La utilización de interfaces y la aparición de los frameworks han popularizado este término. De hecho es el concepto central del Framework de Spring, ya que implementa un "Contenedor" que se encarga de gestionar las instancias (así como sus creaciones y destrucciones) de los objetos del usuario. Por tanto las aplicaciones que utilicen el framework de Spring (no Spring propiamente dicho) utilizarán Inversión de Control [15].

La inversión de control se utiliza para gestionar las dependencias entre subsistemas y el framework.

- **Global Concept:**

Permite la utilización de elementos comunes con cada definición [15].

- **Nomencladores genéricos**

Permiten unificar nomencladores comunes en una aplicación, además de la gestión de los mismos. Según la definición más difundida metadatos son «datos sobre datos». Debido a que muchas veces no se tiene en cuenta la diferencia entre datos e informaciones también hay muchas declaraciones como «informaciones sobre datos», «datos sobre informaciones» y «informaciones sobre informaciones» [15].

Los metadatos pueden describir colecciones de objetos y también los procesos en los que están involucrados, describiendo cada uno de los eventos, sus componentes y cada una de las

restricciones que se les aplican. Los metadatos definen las relaciones entre los objetos, como las tuplas en una base de datos o clases en orientación a objetos, generando estructuras [15].

- **Aspect**

Trabaja sobre funciones programadas orientadas a objetos, la cual unifica los aspectos generales de la aplicación que luego, pueden ser configuradas para la misma. Una de las características a destacar es la separación de responsabilidades [15].

**Manager Exception:**

Este componente realiza varias características funcionales como son; la unificación y el manejo de excepciones en la aplicación, en donde permite configurarlo de forma declarativa en un XML. [15]

- **Validación en el servidor (Validator)**

El componente de validación se utiliza para validar todas aquellas acciones y datos que lo requieran. Antes de ser procesados por el controlador. Este componente se utiliza declarativamente por lo que el desarrollador solo debe entrar los datos pertinentes en cada uno de los XML que se utilizan. En el XML se registran cada una de las precondiciones que se deben cumplir para que una acción sea ejecutada así como la clase y el método que ejecutará cada una de estas precondiciones. También se pueden validar parámetros, para esto se debe registrar el nombre de cada uno de estos que viajan por el POST<sup>11</sup> y el tipo de parámetro que es. El tipo de parámetro se consulta en el XML de las expresiones.

La seguridad de las aplicaciones Web es un tema muy importante ya que en esta se manejan informaciones de muchas instituciones que si alguien no autorizado tuviera acceso a ellas podría causar daños irreparables. Dada la importancia del sistema CEDRUX y toda la información que se maneja se decidió prestar el máximo de interés a las validaciones de cada una de las aplicaciones que conforman este software. En la mayoría de las aplicaciones solo se valida en la capa de presentación, esto trae como consecuencia que si el desarrollador hace una mala validación o el atacante logra violar esta capa entonces puede adquirir acceso total a la información almacenada. Como la seguridad se basa en la cantidad de barreras que un desarrollador implementa desde la vista que se le muestra del usuario hasta los datos, se decidió desarrollar un componente para validar en el servidor todas las acciones que requieran de ellas.

Este componente puede validar todos los datos y acciones que se activan ya sean por un usuario o por un sistema externo que serán ejecutadas en un controlador de una aplicación en específico. La acción será validada antes de llegar al controlador para no hacer peticiones innecesarias a este. [15]

- **App**

Sirve de inicializador de toda la configuración, los manejadores de excepciones y aspectos y de toda la gestión de la seguridad del sistema. [15]

- **Componente Exportador – Importador**

Permite exportar o importar a partir de un fichero XML los datos existentes de otra aplicación, Esto ocurre en un formato definido por la aplicación CEDRUX. Contiene una referencia al componente IOC y Global Concept. [15]

- **Clases:**

- **Controller:**

Permite la extensión entre la clase controladora definida por Zend y la del ERP – CUBA.

Contiene una referencia al componente IOC y Global Concept.

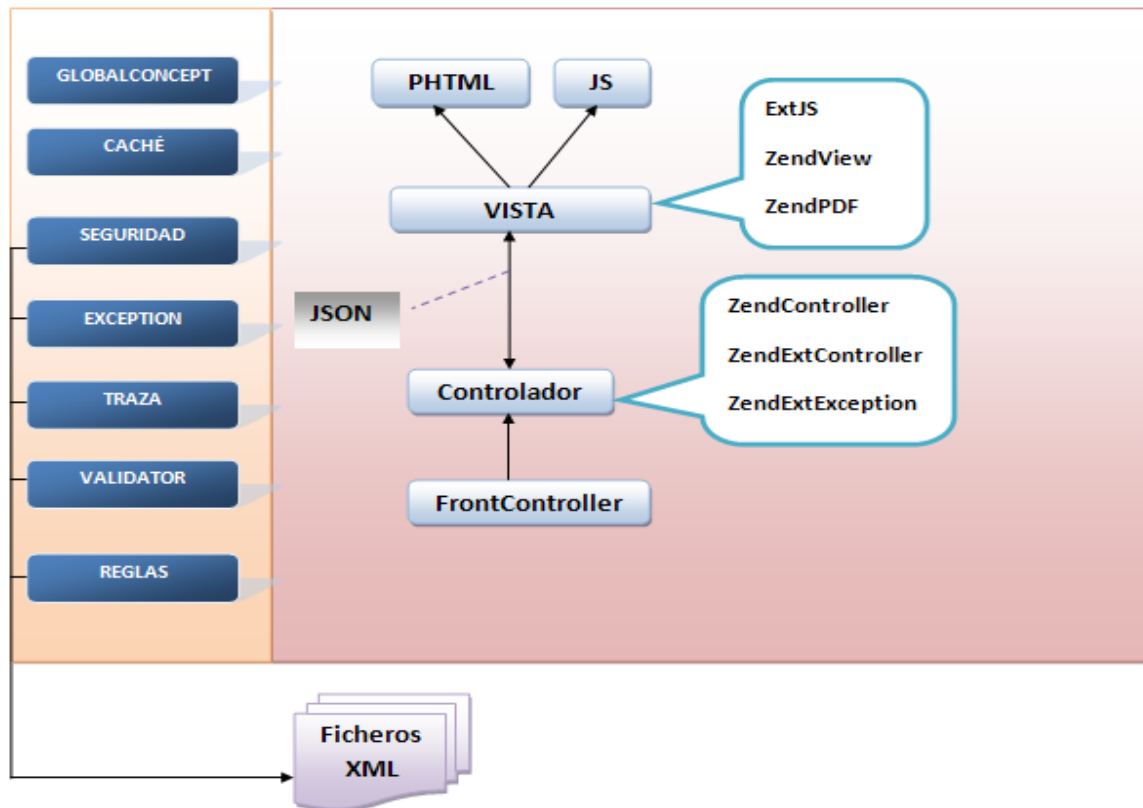
- **Trazas:**

La traza no es más que el registro de determinadas acciones tanto de los usuarios como del sistema, este componente se va a componer de dos elementos básicos, un lanzador de eventos que muestran una acción especificada al usuario y un manejador de trazas que va a encargarse de registrar dichos historiales ya sea en bases de datos o en algún fichero en específico.

El componente Portal está desarrollado sobre el marco de trabajo definido por el equipo de arquitectura del proyecto ERP –CUBA, donde el núcleo de éste está certificado legalmente con el uso de la tecnología en PHP 5.2.4 con la utilización de los frameworks Zend Framework, Doctrine, ExtJS y ezComponent agregándole una extensión al Zend Framework (Zend Ext.) para darle

solución a algunas necesidades propias del negocio a desarrollar, y un estilo híbrido basado en Capas y MVC.

De estos framework, se va a profundizar más en detalle los framework ExtJS y Zend Framework en los cuales se desarrollaron los elementos de la capa de presentación del componente Portal. [15]



**Figura 1.** Capa de Presentación

Desde el controlador se instancia la clase **ZendView** que es la encargada de edificar una interfaz grafica, utilizando ficheros **.js** y **.phtml**.

En este estilo o capa, para las interacciones del cliente con el servidor se define:

- I. Peticiones HTTP Utilizando la tecnología que brinda AJAX.
- II. Peticiones HTTP Utilizando el componente iframe de HTML.

III. Peticiones HTTP de inclusión de archivos de código .js y .css

Los formatos de intercambio definidos son:

**JSON:** para intercambio de datos entre componentes y demás elementos del servidor. Para archivos de Multi-lenguaje

**XML:** para los archivos de configuración que se almacenan en el servidor

Para el desarrollo de interfaces se trabaja en los siguientes aspectos:

- I. Nomenclatura para la definición de componentes.
- II. Definición mínima de uso para componentes del framework EXT.
- III. Validación de Componentes.
- IV. Multilenguaje.
- V. Multitema.

Otro de los aspectos desarrollados en esta capa es la integración de interfaces a partir de una lógica determinada utilizando los elementos de otra totalmente independiente que necesite del servicio. Es similar a utilizar un servicio Web, pero no a nivel de lógica sino a nivel de interfaces.

- **ZendFramework**

Es un framework para el desarrollo de aplicaciones y servicios Web con PHP, brinda soluciones para construir sitios Web modernos, robustos y seguros. Además es Open Source y trabaja con PHP 5.

Tiene la ventaja de ser desarrollado por Zend que es la empresa que respalda comercialmente a PHP. [16]

- **Características**
  - i. Trabaja con MVC (Model - View - Controller)
  - ii. Cuenta con módulos para manejar archivos PDF, canales RSS, Servicios Web (Amazon, Flickr, Yahoo), etc.
  - iii. El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
  - iv. Una solución para el acceso a base de datos que balancea el ORM con eficiencia y simplicidad. Se puede ver esta función en el futuro.
  - v. Completa documentación y pruebas de alta calidad.
  - vi. Robustas clases para autenticación y filtrado de entrada.
  - vii. Clientes para servicios Web, incluidos Google Data APIs y StrikIron.
  - viii. Muchas otras clases útiles para hacerlo tan productivo como sea posible.
- **ExtJS**
  - **Características**
    - i. Es un framework completamente Orientado a Objetos.
    - ii. Multiplataforma.
    - iii. Implementación de la tecnología AJAX.
    - iv. Múltiples posibilidades para el trabajo con las validaciones y manejo de errores en el cliente.
    - v. Personalización de temas de estilos.
    - vi. Amplia configuración e Intenso trabajo con las hojas de estilos CSS.

- vii. Catalogado por la comunidad de desarrolladores como uno de los frameworks más aceptados y completos.
- viii. Basa toda su funcionalidad en Java Script a través de librerías ya muy conocidas:
  - ✓ YUI
  - ✓ JQuery
  - ✓ Prototype/Script.aculo.us

O haciendo uso de la librería nativa, así en tiempo de ejecución carga y crea todos los objetos HTML a través del uso intenso de DOM.

#### 1.4.2 Ventajas de la utilización del Marco de Trabajo.

Las que se derivan de utilizar un estándar; entre otras:

- I. El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona un esqueleto que hay que "rellenar".
- II. Facilita la colaboración. Cualquiera que haya tenido que "pelearse" con el código fuente de otro programador (¡o incluso con el propio, pasado algún tiempo!) sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos.
- III. Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo.
- IV. Alta productividad, extensibilidad, facilidad del mantenimiento, reusabilidad.
- V. Los desarrolladores se concentran en la lógica solamente
- VI. El marco conecta las capas de forma automática y extensible, mediante el uso de los aspectos declarados en XML

Parametrización y extensibilidad en varios criterios arquitectónicos

- Excepciones
- Validación en el servidor
- Concurrencia
- Cargas y cacheo de variables y nomencladores globales
- Menú, multilinguaje, apariencia.

## 1.5 HERRAMIENTAS

### 1.5.1 Editores Web

Como en cualquier aspecto de la informática, no existe una única aplicación que sirva para todo, y el amplio mundo del desarrollo Web no es una excepción. El mismo, es quizás la rama informática que aglutina más lenguajes (HTML, CSS, Javascript, PHP, Java) distintos que deben interactuar entre sí, y por ello se necesita un editor Web que facilite el trabajo de cada usuario. [17]

Son muchos los editores Web existentes en el mercado (Dreamweaver, Front Page, CoffeCup HTML Editor, Homesite) cada uno con sus propias peculiaridades, aunque en realidad no son muy distintos unos de otros. [17]

Las características principales que se observan en todos ellos son la edición WYSIWYG (What You See Is What You Get o Lo que ves es lo que obtienes), cliente FTP, control de versiones, autocompletado de código, resaltado de sintaxis, soporte de lenguajes de servidor, depuración, documentación en línea. [17]

Y cada uno de los usuarios le dará más valor a unas u otras dependiendo del entorno en el que trabaje y el trabajo específico que realice. [17]

#### **Adobe Dreamweaver CS3:**



Dreamweaver aunque es de pago, es una de las herramientas más utilizadas para la creación de sitios Web. Desarrollada para los grupos de usuarios principiantes y avanzados.

La importancia que brinda su herramienta de extensiones, es que permite añadir funciones a Dreamweaver desarrolladas por Adobe, comunidad de usuarios u otros, las cuales le agrega funcionalidades a dicha aplicación. [18]

- **Características:**

Plataformas Windows y MAC, disponible en varios idiomas, completamiento muy bueno, coloreado en la s sintaxis del código, lenguajes: HTML, Coldfusion, PHP, ASP VBScript, ASP.NET, C#, Javascript, XML, XSLT, CSS, ActionScript, JSP, entre otros. Incluye Framework para AJAX, administrador CSS, compatibilidad con dispositivos móviles, integración con Adobe Photoshop y Adobe Fireworks. [18]

- **Ventajas:**

Genera código bastante limpio, permite la creación de páginas en diferentes lenguajes, creación de plantillas, definición de sitios. [18]

- **Desventajas:**

Requiere de conexión a Internet para registrar el producto, necesita como mínimo de 512 MB de RAM para poder instalarlo, costo de la licencia. [18]

## NVU

Es un editor compatible con las plataformas Windows, MAC y Linux. Es un proyecto opensource, con licencias MPL/GPL/LGPL. NVU está basado en la plataforma de Mozilla y en el diseño de Gecko. Desarrollado principalmente por Linspire. [18]

- **Características**

Administrador de sitios mediante FTP, Editor CSS, Incluye validador HTML y CSS. [18]

- **Ventajas**

Muy rápido, fácil de usar, permite mantenimiento según los estándares W3C, soporte para XML y CSS. [18]

- **Desventajas**

Sólo permite el trabajo mediante tablas, muy pocas opciones, soporte para pocos lenguajes. [18]

### Kompozer

Kompozer es un editor Web WYSIWYG basado en Nvu, que cuenta con licencia GPL. Creado con el fin de ser una alternativa libre a Adobe Dreamweaver, surge luego de Nvu, quien dejó de ser desarrollado. Fue bautizado con el nombre de Kompozer debido a que el mismo estaba registrado por Linspire. Compatible con las plataformas Windows, MAC, Linux, Debian, RedHat. [18]

- **Características**

Incluido administrador de ficheros vía FTP, editor CSS, barras de herramientas administrables, validador HTML. [18]

- **Ventajas**

Fácil de usar, software libre, multiplataforma, soporte para varios idiomas. [18]

- **Desventajas**

Pocas opciones para el desarrollo, puede dificultar el desarrollo de la aplicación, al contar con pocas herramientas que agilicen el trabajo. [18]

### Quanta Plus

Quanta es un Editor HTML basado en el código de Bluefish. Es un proyecto opensource, bajo la licencia GPL. Disponible para plataformas Linux, bajo el entorno KDE.

- **Características**

Administrador de proyectos, soporte HTML, XML, PHP, CSS, plantillas, soporte de plugins, integrado vista previa de los archivos. [18]

- **Ventajas**

Gratuito, fácil de utilizar, brinda color al código, rápido, completamiento de código. [18]

- **Desventajas**

Dificulta la rapidez de desarrollo. [18]

## 1.5.2 Herramientas de generación de ayudas técnicas.

### JSDoc

Herramienta que analiza el código de los archivos Javascript y genera la documentación en formato HTML, XML o XMI (UML).

- **Características**

JSDoc está basada en el conocido Javadoc que hace lo mismo pero para el lenguaje de programación Java. Está desarrollado en Perl lo que requiere que se tenga instalado el Perl en la computadora.

- **Ventajas**

Permite de manera sencilla a cualquier individuo con un mínimo de experiencia la personalización de cualquier atributo prefijado.

- **Desventajas**

La interfaz de la documentación a obtener como resultado posee una apariencia poco amigable

## EasyCHM

Herramienta de fácil acceso que permite crear archivos de ayuda en CHM.

- **Características**

Básicamente, consta de un completo editor que se maneja a base de tablas de contenidos, las cuales servirán de contenedor para el texto, enlaces o imágenes que se desee añadir en un trabajo a realizar. Luego estos se indexan a un título (el cual aparecerá en el listado de árbol) que sirve para acceder a los mismos.

- **Ventajas**

Admite varios formatos, entre ellos HTML, HTM, MHT, JPG, GIF, etc., los cuales se exportarán finalmente con extensión CHM. Genera automáticamente Tabla de Contenidos e Índice analizando el título HTML, nombre de archivo o texto de cierta línea. Multiplataforma.

Para la creación de la documentación técnica y ayuda formal del componente Portal soportado por el API-PORTAL se utilizarán las herramientas Easy CHM, para generación de la ayuda técnica en formato .chm y para la apariencia gráfica, el editor Web Dreamweaver CS3 atendiendo a los aspectos antes expuestos.

## 1.6 TRABAJO CON COOKIES.

### 1.6.1 Definición de Cookie

Las cookies representan mecanismos que permiten al servidor almacenar en el cliente información relativa a la transacción. Así, cuando el servidor responde a una petición del usuario, inserta una información que se conservará en este. Esta información es una descripción relativa a un conjunto de direcciones URL. En adelante, será insertada por el cliente en una petición que referencie estas direcciones URL y se denomina un cookie. [19]

Los usuarios pueden configurar los navegadores para que estos les avisen con una alerta de la existencia de “cookies” en los sitios Web visitados, o para impedir directamente que las “cookies” puedan instalarse en sus discos duros. [20]

Las “cookies” se dividen en dos tipos distintos, denominados “de sesión” y “permanentes”. Las “cookies” de sesión no se graban en el disco duro de los usuarios, pero permiten el almacenamiento en una base de datos de los datos recopilados durante la permanencia de estos en el sitio Web. Las permanentes se graban al disco duro de los usuarios, permaneciendo en éste durante un periodo determinado de tiempo especificado durante su programación. [20]

### **1.6.2 Uso de las Cookies por el lado del cliente.**

Las cookies se guardan del lado del cliente por lo que se pueden alterar, por tanto no confiar en su contenido y siempre verificar que los datos sean correctos antes de utilizarlos en lugares 'sensitivos' (que puedan abrir un hoyo de seguridad). [21]

También no deben de contener información privada tales como información del usuario ya que si el usuario utiliza un sistema en un lugar 'compartido' (un café internet, etc.) otro usuario podría ver el contenido de ellas, en muchos casos no se utiliza encriptación. [21]

En un caso extremo el usuario podría bloquearlas, por tanto el uso debe hacerse de manera opcional, si estas no pueden omitirse (por ejemplo más adelante con el uso de sesiones las cookies deben aceptarse) entonces advertir al usuario. [21]

Por medio de las cookies es posible realizar la autenticación, efectuar el rastreo, mantener información específica, preferencias y compras de los usuarios. Su principal ventaja es que no se requiere que el usuario registre ningún tipo de información. Sin embargo, el usuario debe configurar el navegador de Internet para permitir la recepción de las cookies, pero además estas pueden ser borradas en cualquier momento con la consecuencia de que se pierde el historial de información con el usuario, existen implicaciones sobre la privacidad y anonimato de los usuarios mientras navegan y aunque las cookies solo son enviadas entre el cliente y el servidor que las

originó, la utilización de cookies de terceros es posible cuando se carga una página que contiene anuncios o imágenes de otros dominios. [22]

Con la utilización de cookies la identificación de usuarios y sesiones puede resultar complicada debido a que estas no identifican a una persona sino a una combinación de la cuenta del usuario, el ordenador y el navegador Web, por lo que un usuario que utiliza múltiples ordenadores, cuentas de usuario o navegadores será identificado por una diversidad de cookies. En el caso de varios usuarios utilizando el mismo ordenador, con la misma cuenta de usuario y navegador Web, tendrán el inconveniente de que serán identificados como un único usuario. También existen algunos problemas de seguridad relacionados con la captura de cookies de otros usuarios con el fin de simular la identidad de otro usuario ante un sistema, y también con el cambio de los montos que un usuario tiene que pagar por una compra. Otro problema de seguridad consiste en la explotación de las vulnerabilidades de algunos navegadores que permiten que un atacante coloque una cookie en el espacio de otro dominio. [22]

Cuando se crea una cookie es importante la configuración que realiza el servidor de algunos parámetros como son nombre de la cookie, la fecha de expiración y el nombre del dominio. Si la fecha de la cookie no ha sido especificada, esta se remueve una vez que el usuario sale del navegador. Por lo que es mejor especificar la fecha de expiración, en cuyo caso se les llama cookies persistentes. [22]

## **1.7 USO DE CACHÉ EN EL NAVEGADOR.**

### **1.7.1 Caché**

Es la cache el encargado de almacenar el medio físico (HD) de los elementos más usados por el sistema (nomencladores, XML y otros) estos se almacenan en este medio con una estructura que, a pesar de no encontrarse en la memoria volátil del sistema, la recuperación de los mismos es casi instantánea (serialización).

El uso de la caché en los navegadores de internet, está tan extendido que incluso el protocolo HTTP ha incluido ciertas cabeceras para facilitar su control (como "Last-modified", "Expires", etc.)

así como el lenguaje de marcas HTML, que también incluye sus marcas especiales para indicar cuando una página debe almacenarse en el caché, hasta que fecha, etc. [23]

La caché del explorador que se esté utilizando, debe tener en cuenta estos aspectos (las cabeceras HTTP o las marcas especiales en las páginas), para saber si debe almacenar en el caché cierto dato, o si, a partir de cierta fecha, debe dar por caducado algún recurso almacenado en el caché. [23]

## 1.8 CONCLUSIONES

En el capítulo se detallan algunas cuestiones que son fundamentales en el desarrollo de la arquitectura de presentación como son el uso de CSS el cual permite controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. También se destaca el uso de los formatos de imágenes, así como una breve comparación entre ellos. La estructura arquitectónica de los objetos del DOM es otro de los aspectos que se consolidó en el precedente capítulo, en el cual se enmarca la definición de estos, siendo la interfaz que permite acceder y manipular, mediante la programación, los contenidos de una página Web. También otras de las características arquitectónicas que se destacan es la definición de Ajax, donde se enfatiza que es una técnica de desarrollo Web para crear aplicaciones interactivas. La selección de la tecnología y del marco de trabajo es otro de los aspectos que fueron tratados en este apartado, detallando las características de sus principales componentes, así como la decisión tecnológica para el desarrollo del componente Portal. Además se abordó el trabajo con cookies por el lado del cliente y el uso de la caché.





## CAPÍTULO II ARQUITECTURA DE PRESENTACIÓN

### INTRODUCCIÓN

El siguiente capítulo integra, los aspectos que conforman la solución propuesta. Se menciona de forma explícita la línea base de la arquitectura de presentación, en la que se exponen varias explicaciones de los documentos necesarios que la conforman. Posteriormente se describen algunos aspectos de arquitectura de información como el estándar de navegación y usabilidad de escenarios, así como otros puntos de interés. Para definir las pautas de desarrollo en la arquitectura de presentación se referencian en el epígrafe normas y estándares para el desarrollo.

### 2.1 LÍNEA BASE DE LA ARQUITECTURA DE PRESENTACIÓN

Para garantizar que un proceso se haga de manera ordenada y cumpla de manera exitosa todos los requerimientos y estándares establecidos, se definieron una serie de elementos que ayudarán durante todo el proceso a la comprensión, asimilación y cumplimiento de la línea base de la arquitectura de presentación.

La mayoría de los IDE<sup>12</sup> de desarrollo establecen su propia nomenclatura de definición para los componentes, lo cual se pone de manifiesto sobre todo a la hora de la generación automática de código, ayudando a organizar, estandarizar y comprender de una forma mucho más clara y concisa los elementos que se encuentran en el código a medida en que se desarrolla.

Para generar las interfaces de los subsistemas, módulos, y demás que contengan funcionalidades se empleará el Framework de ExtJS con la versión dictaminada y validada por el equipo de arquitectura.

La documentación para el desarrollo de la arquitectura de presentación aplicada a los componentes, se encuentra estructurada en el repositorio central del proyecto ERP-CUBA, donde se encuentra toda la información del desarrollo del sistema Cedrux. El área de Arquitectura de presentación es la encargada de implementar, disponer y actualizar los temas de presentación, así

como documentar todos los artefactos generados durante el desarrollo de dicha arquitectura. Se ubican específicamente en <http://10.12.179.3/svn/erp/ERP/Ingenieria/Arquitectura/Desarrollo/ArquitecturaPresentacion/Expediente de la AP>. A continuación se detallan dichos artefactos:

**Nomenclatura para definición componentes de ExtJS**

Para la definición en un fichero .js los componentes que hacen referencia a las clases de ExtJS cumplen un estándar de nomenclatura definido en el fichero “Nomenclatura para definición componentes de ExtJS.doc”, en breve se plasma elementos fundamentales de este.

- **Nomenclatura para definir componentes.**

Cuando se define un componente dentro de una interfaz específica, siempre se cumplirá el siguiente estándar de nomenclatura de nombres.

Componente	prefijo + nombre CU	
Panel	p	+CasoUso
TabPanel	tp	+CasoUso
Viewport	vp	+CasoUso
Window	win	+CasoUso
GridPanel	gp	+CasoUso
TreePanel	trp	+CasoUso
FormPanel	fp	+CasoUso
ComboBox	cmb	+IdCampo
DateField	df	+IdCampo

NumberField	nf	+IdCampo
Radio	rd	+IdCampo
Checkbox	chk	+IdCampo
TextArea	ta	+IdCampo
Store	st	+CasoUso
Reader	rd	+CasoUso

- **Nomenclatura para definir las etiquetas (labels)**

Al definirse una etiqueta para un componente dentro de una interfaz específica, siempre se cumplirá este estándar de nomenclatura de nombres.

<b>Tipo campo</b>	<b>prefijo</b>	<b>+nombre CU</b>
Campo BD	lb	+idCampoBD
Título ventana	lbTit	+winCasoUso
Botón	lbBtn	+CasoUso
Mensaje	lbMsg	+CasoUso
Título Msg	lbTitMsg	+CasoUso

Para la validación de los datos en la parte del cliente utilizando Java Script se realizará de la forma que se expone en el documento para validaciones: “**Validaciones Utilizando Ext-2.x.x.doc**”, el cual plantea los siguientes argumentos:

- **Validación de un formulario con Javascript**

La validación de un formulario es el proceso que permite comprobar la validez de los datos introducidos en función de unos requerimientos establecidos. De esta manera si se intenta

introducir en un campo un valor que no le corresponde la validación debe controlar esta contingencia y mostrar un mensaje u otro tipo de acción similar.

¿Cómo validar un formulario en una página Web, en el lado del cliente con Java Script?

Se comprueba que todos los campos cumplirán con las precondiciones y reglas establecidas para su cumplimiento como requerimiento. Si son correctos, se envía el formulario, en otro caso, se retiene el envío.

Existen dos tipos de validaciones dictaminadas por el arquitecto de presentación del proyecto ERP-CUBA a la hora de desarrollar a nivel de interfaz sobre un componente determinado. La primera es que a medida que el usuario va escribiendo, verifica si cumple o no con el carácter especificado, de cumplirse coloca el carácter, si no se cancela el evento y la otra consiste en la verificación de una cadena al ser introducida señalando el error utilizando dos de los campos de las opciones de configuración de algunos de los componentes que brinda el framework Ext, que más adelante se explican.

- **Tipos de validaciones en el cliente:**

1. Valor Entero

- Entero +
- Entero -
- Que se encuentre en un rango de valores determinado.

2. Valor Real

- Real +
- Real -
- Que se encuentre en un rango de valores determinado.

3. Que sea de tipo email.

4. Que el campo no esté vacío.

5. Número

- "0-9".

6. Alfabeto

- "a-z"

- "A-Z".
7. Alfa-Num
- "a-z"
  - "A-Z"
  - "0-9"
8. Fecha
- Que sólo escriba fechas actuales.
  - Fecha en el formato dd/mm/aaaa.
  - Fecha en un rango de valores válidos predeterminados.

- **Expresiones regulares disponibles.**

Estas expresiones regulares se pueden utilizar siempre que se ajusten a los requerimientos de validación, y constituyen una herramienta muy útil y eficiente para validar. Siempre que la validación que se necesite llevar a cabo se pueda implementar a nivel de expresiones regulares, es una muy buena práctica.

Las Expresiones regulares cuyo nombre comienza con "sólo [algo] " se utilizan como en el siguiente ejemplo que sólo deja escribir valores enteros utilizando el campo maskRe:

```
var carneId = new Ext.form.TextField ({  
  
fieldLabel:'Carne de identidad',  
  
maskRe: /^-?\d*$/  
  
});
```

Este tipo de expresiones regulares funcionan como máscara en la medida que el usuario va escribiendo, se realiza la verificación, si cumple se coloca el carácter, si no cumple se cancela el evento y no se coloca el carácter. Esto no siempre se puede lograr a nivel de expresiones

regulares pues a veces se hacen un poco más complejas en el tipo de verificación, y lo que se hace es que se combinan de varias formas.

Para solucionar el tema de la máscara no a nivel de expresiones regulares, generalmente se controla el evento al presionar una tecla, se realiza la verificación de alguna forma y se cancela o libera el evento, pero implementando una función propia. A este tipo de validación se le conoce como función formatear texto, pues a medida que se escribe se va formateando el texto con la función determinada y que cumpla con los requerimientos específicos de la validación.

Las Expresiones regulares cuyo nombre comienza con "es [algo] " validan que una cadena cumple el requisito especificado, señalando el error al invocar la validación, permitiendo la escritura. Para ello se utiliza el regex que es un objeto JavaScript que comprueba contra el valor del campo durante el proceso de validación. Si esta disponible el actual valor del campo lo admite, si en la comprobación falla el campo será marcado de no válido utilizando el campo `regexText`, que se encarga de mostrar un texto de error si se utiliza la expresión regular en cuestión.

```
var carnelId = new Ext.form.TextField ({  
  
    fieldLabel:'Carne de identidad',  
  
    regex: /^-\?d*$/,  
  
    regexText : 'este valor no es correcto'  
  
});
```

Se recomienda que mientras existan las dos posibilidades es bueno utilizar ambas.

- **Empleando validaciones con EXT-2.0:**

Como se comentó anteriormente el Framework Ext brinda ya un conjunto de configuraciones que permite abstraerse de tener que realizar validaciones, simplemente se configura el tipo de validación y este se encarga de determinar las acciones a partir de las precondiciones.

Estas configuraciones son de tipo general, y están presentes en algunos componentes que soportan unas configuraciones u otras, pero nunca se utilizará alguna que no implemente el componente.

### **Integración de componentes**

Para la integración de las interfaces que coincidan con funcionalidades comunes para dos o varias funcionalidades y se utilicen de la misma manera, se realiza utilizando los dos componentes relacionados en el documento: “**Integración de Componentes de Presentación.doc**”. Este documento pretende poner a disposición dos componentes que van a ser de mucha utilidad para la reutilización de componentes a nivel de interfaz, brindando una serie de configuraciones con las cuales se logra hacer que dos interfaces totalmente independientes de su lógica a partir de contratos definidos por ambas, sean capaces de comunicarse entre sí. Este documento se aborda de manera más profunda en el capítulo 3.

### **Definición de código en ficheros**

Para definir el código de una plantilla .phtml y de un fichero de código .js en todos los casos se hará de la forma en que se expresa en el documento: “**Definición de Código en Ficheros.doc**”. A continuación se le presenta una breve descripción técnica de dicho documento.

- **Creación de una plantilla.**

Para la creación de una de un fichero phtml se construyen los elementos dentro de esta los cuales le permitirá funcionar. Para favorecer la organización y su funcionalidad en el momento que se desea desarrollar un componente se debe colocar la plantilla phtml dentro de la carpeta del módulo en la dirección que se especifica:

**módulox/views/scripts/controlador/nombreVista.phtml**

En este fichero estaría todo el código HTML relacionado con la plantilla del caso de uso, se recomienda colocarle el mismo nombre que el controlador para garantizar una mejor ubicación y estructuración para un fructífero desarrollo. Para esto primeramente se incluyen los tag de HTML

que comúnmente se utilizan en este tipo de plantilla. Unos de los tag a destacar es el que se presenta en la línea de código siguiente: **<meta http-equiv="content-type" content="text/html; charset=utf-8" />** que indica el tipo de codificación usada en la página. También se incluye el tag **script** que responde a: **<script type="text/javascript src="../../../frameworks/UCID/js/imporcss.js"></script>**, que es usado para importar ficheros JavaScript (.js).

Una vez incluidos el script anterior, para garantizar que la vista de la página se cargue en correspondencia con el tema que tiene configurado el usuario, se incluye todas las hojas de estilos(css) que influyen en la vista de colores, de manera dinámica escribiendo en la página .phtml el siguiente código script:

```
<script language="javascript" type="text/javascript">  
  
importarCSS('../../../frameworks/ExtJS/temas/'+window.parent.perfil.tema+'/css/ext-all.css');  
  
</script>
```

Esto garantiza que el CSS se importe antes de que los componentes se visualicen en la página, lo cual podría traer problemas.

Antes de cerrar la etiqueta body (cuerpo) que es lo último que se hace, se coloca la inclusión de los tag script:

```
<script type="text/javascript" src="../../../frameworks/ExtJS/js/ext-base.js"></script>  
  
<script type="text/javascript" src="../../../frameworks/ExtJS/js/ext-all.js"></script>
```

El equipo de arquitectura definió que la inclusión de las etiquetas script fueran antes de cerrar el body, por una cuestión establecer un estilo más apropiado a la hora de desarrollar las interfaces. Ya que a partir de estos ficheros ya incluidos, le da libertad al desarrollador de seguir incluyendo cuantos ficheros necesita para lo que desee realizar.

Con estos nodos scripts se está preparado para utilizar los elementos de las librerías de ExtJS, y a continuación se incluyen en un orden lógico los ficheros scripts que sean propios del CU.



```
<script type="text/javascript" src="../../views/js/nombreVista.js"></script>
```

Por convenio una vista siempre se corresponderá con un fichero JS, y estos se llamarán iguales con diferentes extensiones. Si este utilizan varios ficheros, siempre incluirlo en orden lógico en dependencia de la ubicación específica.

### **Multilenguaje**

Para garantizar que el sistema en general sea configurable para varios lenguajes, los desarrolladores garantizarán los elementos relacionados en el documento: “Garantizando Multi-Lenguaje.doc”. Inicialmente se crea un fichero .json, funcionando como un formato ligero en el intercambio de datos. Esta plantilla se coloca en la carpeta...\**views\idiomales\controlador\plantilla.json**, correspondiendo una carpeta por cada controlador y un fichero de configuración por cada plantilla. Como se indica va a contener cada uno de los labels que son configurables para un idioma u otro, y se crea con el siguiente formato:

```
{ lbNombre:'Nombre'

  lbDisplayMsg:'Resultados del {0} - {1} de {2}',

  lbEmptyMsg:'0 Resultados para mostrar',

  lbTitWinGestCE:'Gestionar Estudiantes',

  lbBtnAdicionar:'Adicionar',

  lbMsgSelFila:'Debe seleccionar la fila que desea eliminar',

  lbWaitMsgRegEst:'Registrando estudiante ...',

  lbTitMsgConfirm:'Confirmaci&oacute;n',

  lbMsgConfRegOk:'El estudiante fue registrado corresectamente',

  lbTitMsgError:'Error',

  lbMsgError:'Ha ocurrido un error al insertar el estudiante' }
```

Para garantizar que el sistema en general sea configurable para varios temas, los desarrolladores responderán a los elementos relacionados en el documento: “Garantizando Multi-Tema.doc”, está ubicado en la vista de arquitectura dentro del expediente de la Arquitectura de Presentación en documentos complementarios.

Para avalar que desde el punto de vista de la usabilidad la aplicación funcione con un ambiente visual homogéneo y se ajuste a las necesidades trazadas, los desarrolladores de interfaz de usuario, asegurarán que la interfaz cumpla con la especificación del fichero: “Estándar para el diseño de interfaces.doc”, con el propósito de lograr un estándar de diseño que satisfaga las necesidades y exigencias de los clientes, sea compatible con las tecnologías utilizadas, cumpla con todos los objetivos perseguidos y que represente al proyecto ERP-CUBA a manera de sello y presentación de su trabajo. Este artefacto representa una propuesta que constituirá una pauta a cumplir por todas las aplicaciones Web de tipo gestión que se desarrollen en el centro. Este tema se describe adelante con mayor profundidad en el epígrafe de **Normas y Estándares del desarrollo del componente portal**.

### **Política de Utilización de CSS e imágenes**

Cuando un desarrollador por requerimientos de la interfaz necesite crear un estilo o utilizar una imagen que no esté de forma global y que sea particular para determinada interfaz, Siempre garantizará cumplir lo establecido en el documento: “**Política de Utilización de CSS e imágenes.doc**”. Dicho documento se describe a continuación:

Los ficheros de configuración de estilos tienen que cumplir determinadas especificidades para poder garantizar los requerimientos de multi-tema y multi-lenguaje. Estos son requerimientos en los que se tiene incidencia a la hora de definir la apariencia. Cabe recapacitar que hay imágenes que varían en colores, tamaños y formas de un estilo a otro. Además pueden poseer palabras grabadas que varían de un idioma a otro. Esta estructura se define a nivel de framework, que permite bajo determinadas configuraciones garantizar el cumplimiento de estos dos requerimientos.

En la carpeta:

/comun/frameworks/ExtJS/idioma/es/temas/default/css

Se encuentran los ficheros:

desktop.css

Ext-all.css

Icon-clases.css

Y en la carpeta:

/comun/frameworks/ExtJS/idioma/es/temas/default/images

Se encuentran las carpetas con las imágenes por componentes.

Estas carpetas se van a repetir por cada estilo y cada idioma que se haya definido en el framework para el portal que juega como componente fundamental de integración para todos los componentes desarrollados. El contenido de cada carpeta se va a repetir con cada configuración antes mencionada, que se incluyen unos u otros según corresponda con el tema y el idioma seleccionado.

En las direcciones antes expuestas se usa el ejemplo en que se tiene un tema de estilos predeterminado por defecto (default) y un paquete de configuración para el idioma español que se determina con la abreviatura “es”.

### **Ficheros de configuración de estilos.**

El framework de manera general cuenta con dos ficheros de configuración de estilos CSS los cuales son los ejes principales que dan la apariencia y estilo a los componentes y los elementos a desarrollar. Ambos a nivel estructural y con las variables de configuración correctamente empleadas para incluirlos en una plantilla, se cargan en dependencia de la configuración pedida por el usuario del sistema, donde se garantiza por cada estilo dos ficheros con configuraciones diferentes para las apariencias, en colores, imágenes y otros.

El primer fichero que se hace referencia es al que se denomina `ext-all.css`. Este fichero es el núcleo del framework EXTJS con todos los estilos de configuración, y es quien brinda la apariencia a todas las interfaces relacionadas con este.

El segundo fichero se denomina `icon-clases.css` y forma parte de los complementos para desarrollar las interfaces a varios niveles, en este se colocan las imágenes personalizadas que no forman parte del núcleo de EXTJS, principalmente serán íconos, y otros similares.

## 2.2 ESTÁNDAR DE USABILIDAD DE ESCENARIOS

La creciente complejidad de los contenidos informativos y de los servicios que se ofrecen a los usuarios hace que en los últimos años, las herramientas de navegación y orientación dentro de las sedes o portales, cobren cada vez mayor importancia. De esta forma, se están recuperando e incorporando a la Web presentaciones de los sistemas hipertextuales clásicos. Entre las características de los mismos se encontraban funciones que pretendían evitar el desbordamiento cognoscitivo de los usuarios, y orientar sus procesos de exploración y recuperación de la información deseada. [24]

- **Pantalla Estándar Árbol**

La navegación de esta pantalla está conformada por un menú estructurado en forma de acordeón, donde su navegación es completamente vertical; como elemento de dicha pantalla también se encuentra otro menú estructurado con una navegación horizontal.

- **Pantalla Escritorio de Módulos (por sus siglas en inglés PDM).**

Esta es una pantalla destinada a que los usuarios que prefieren ambiente Linux se sientan más familiarizados y les resulte más cómodo el trabajo, en ella se asiste a un módulo específico del sistema. La navegación se estructura horizontalmente.

- **Pantalla Escritorio de Acciones (por sus siglas en inglés PDA).**

Esta es una pantalla destinada a que los usuarios que prefieren ambiente Linux se sientan más familiarizados y les resulte más cómodo el trabajo, con la particularidad de poseer una estructura arbórea asistiendo a una funcionalidad específica del sistema. También está sometido a la navegación horizontal [A-3].

### **2.2.1 Estándar de usabilidad de los menús**

La Usabilidad es una disciplina que se ha consolidado como una rama de los estudios sobre Interacción Hombre- Máquina (Human Computer Interaction), se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso. Es por ello que no depende sólo del producto, sino también del usuario; el programador adopta estándares de usabilidad para crear de la aplicación, al mismo tiempo el usuario final, sobre todo con las nuevas tendencias de la Web, puede modificar la interfaz según sus propias particularidades.

Usabilidad es la eficiencia y satisfacción con la que un producto permite alcanzar objetivos específicos a usuarios específicos en un contexto de uso. Esta es una definición centrada en el concepto de calidad en el uso, es decir, se refiere a cómo el usuario realiza tareas específicas en escenarios particulares con cierta efectividad. [24]

#### **Tipo de menú**

El menú principal está colocado en la izquierda superior, de ahí en adelante todos los menús complementarios se van desplegando hacia la derecha consecutivamente.

#### **Apariencias del menú**

Los menús existentes en el componente tienen una apariencia de forma arbórea en el mayor de los casos unos de forma horizontal y otros verticalmente.

### 2.2.2 Taxonomías

Se define como taxonomía a la organización jerarquizada, que se emplea para estructurar contenidos, ella divide estos en clases y los agrupa según sus características. Por tanto, favorecen la navegabilidad y facilitan además acceder al capital intelectual de una determinada organización. [25]

### Etiquetado

El etiquetado es un sistema de representación que utiliza términos y/o expresiones de varios términos, para identificar, de la forma más inequívoca posible, contenido informativo. Tal como se utilizan las palabras para representar pensamientos, las etiquetas permiten representar un conjunto de información. Las etiquetas describen o designan los elementos que integran el sistema de navegación. [26]

A continuación se muestra un fragmento del sistema de etiquetado del componente Portal:

**Para el caso del subsistema de Seguridad.**

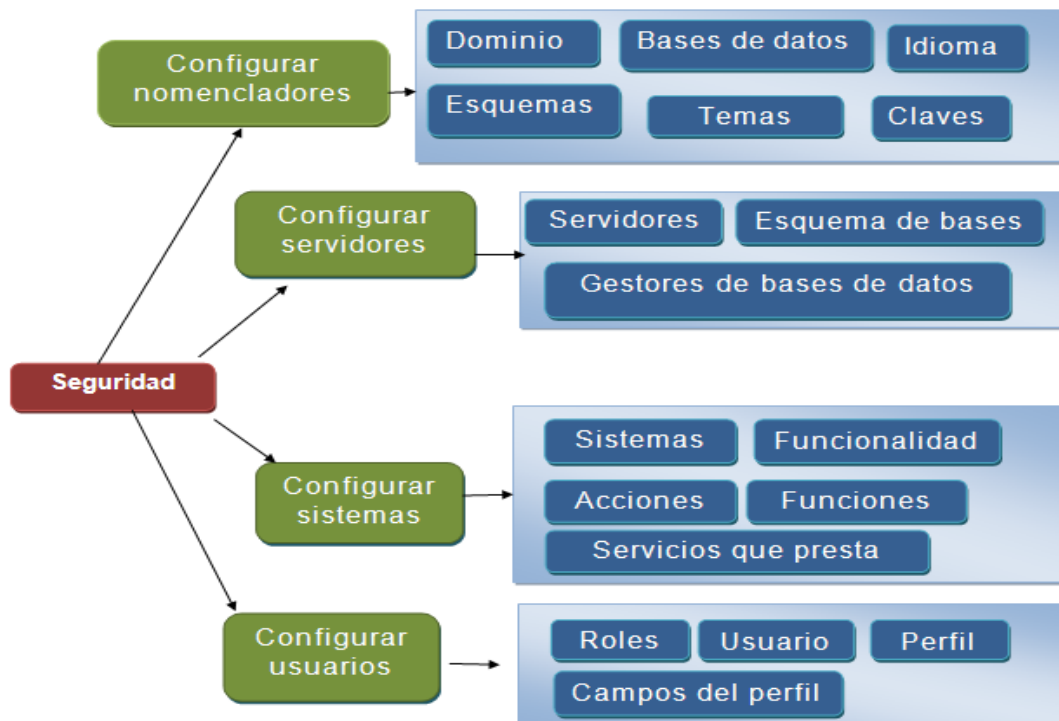


Figura 2. Sistema de etiquetado del subsistema de seguridad.

Tarjetas de colores de la identidad



Figura 3. Colores utilizados en la entidad del Sistema CEDRUX

En la figura anterior se destacan el uso de las tarjetas de colores, en la que se visualiza el modo de color sustractivo CMYK que es el acrónimo de los colores básicos (Cian, Magenta, Amarillo y Negro) que se utilizan en las impresiones a cuatro colores. También se muestra el modo adictivo RGB, que representa los colores primarios de la luz rojo, verde y azul.

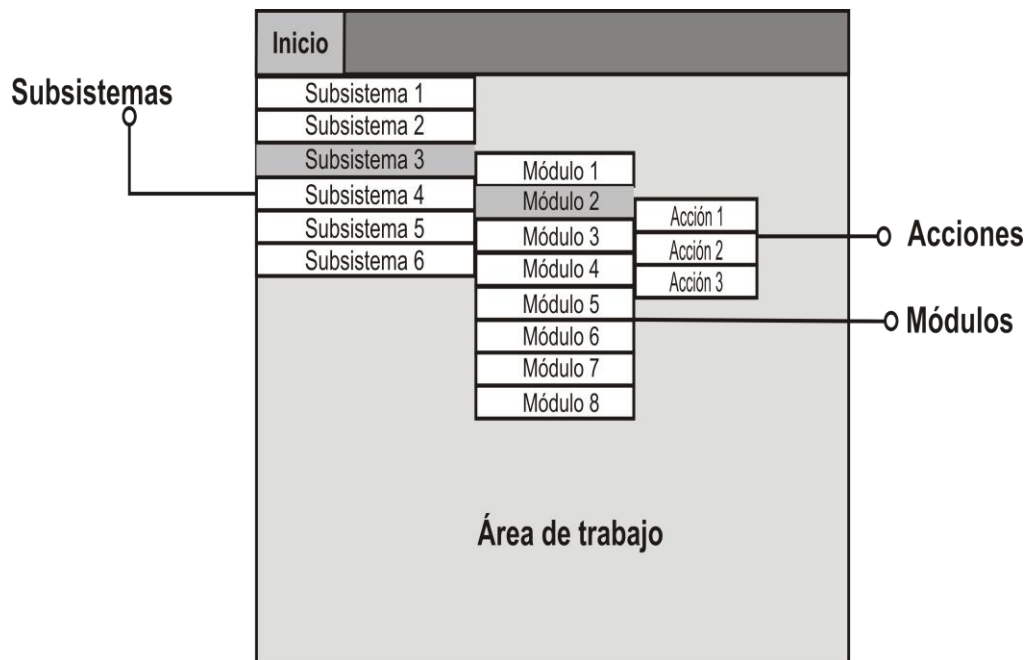
La gama cromática empleada responde a los atributos del producto que se pretende comunicar, a la vez que actúa como elemento diferenciador de sus homólogos cubanos y extranjeros. Donde solo se encuentra una coincidencia; en el caso del verde, con Open Bravo<sup>13</sup>. Además de ser verdes con diferentes niveles de saturación, son combinados con colores de apoyo diferentes como el Gris, un verde similar al de la empresa Open Bravo y finalmente el azul, obteniéndose como resultado los colores que imperan en la entidad de CEDRUX.

Al verde claro y saturado (TECNOLOGÍA), lo acompaña un azul oscuro y un poco menos saturado (NACIONALIDAD, aunque este es un atributo de menos peso que el resto). La intención en todo momento es trabajar con colores que se alejen de los que comuniquen tradición. Pues esta es una forma MODERNA de llevar finanzas, control, economías, gestiones... La combinación de ambos transmite la SEGURIDAD y CONFIANZA tan necesarios en este tipo de herramientas. Sobre todo, cuando el poco tiempo de existencia del software, puede crear cierta desconfianza en el consumidor. El verde, color identificativo de la marca, es positivo por excelencia, tranquilizante y a la vez tropical y esperanzador.

### **Estructura de las áreas de trabajo**

A continuación se muestra la estructura del área de trabajo del componente Portal, donde se especifican sus regiones, con su descripción:





**Figura 4.** Áreas de trabajos estructuradas

**Subsistemas:** Clasificaciones de un sistema que agrupan según sus funcionalidades módulos específicos.

**Módulos:** Agrupan acciones específicas dentro de un subsistema.

**Acciones:** Operaciones que se pueden realizar en cada módulo.

**Área de Trabajo:** Área donde tiene lugar el desarrollo de las tareas o funcionalidades inherentes al sistema.

En el Área exterior se muestra en la zona superior una lista desplegable, a través de la cual el usuario podrá seleccionar un subsistema y el módulo de su interés correspondiente a este. Una vez que se accede a un módulo desde la lista desplegable de acceso a las acciones se abre una ventana genérica donde el usuario podrá realizar las operaciones contenidas en la acción seleccionada. [27]

## 2.3 NORMAS Y ESTÁNDARES PARA EL DESARROLLO

Se define como normas a la especificación que reglamenta procesos y productos para garantizar la interoperabilidad. Conceptualizando además los estándares como la definición clara de modelos, criterios, reglas de medidas o de los requisitos mínimos aceptables para la operación de procesos específicos, con el fin asegurar la calidad en lo que se desea desarrollar. [28]

### 2.3.1 Estándar de diseño de componentes de presentación

El trabajo en equipo es uno de los requisitos fundamentales para que una interfaz responda a las características que se exigen, es importante que todas las áreas implicadas sigan una misma línea que se cumpla con las reglas y las responsabilidades del trabajo en cuestión. A continuación se presenta una política estándar a cumplir por todas las aplicaciones desarrolladas en el proyecto ERP – CUBA en conjunto con el centro UCID<sup>14</sup>. Para realizar algún tipo de acción que no se incluya en este documento, se consultará con el equipo de arquitectura para tramitar el mismo.

#### Reglas a cumplir

- i. La interfaz corresponderá con las características, metas y nivel de experiencia de los usuarios de la aplicación, siempre que no se atente contra las pautas para el diseño.
- ii. La información estará libre de errores gramaticales, ortográficos y tipográficos.
- iii. La taxonomía de los títulos encima de los controles estará según la estipulada, primera letra con mayúscula, el resto con minúscula.
- iv. En controles de tipo input (de entrada), select (selección), text-área (componente perteneciente al framework Ext.) cambiará el color de fondo indicando que el cursor está situado en él para modificar los datos. Una vez que el cursor deja de estar situado sobre ellos, regresarán a su estado normal.

- v. En controles de tipo button (Botones) cambiará el color de fondo indicando que el cursor está situado en él para presionar. Una vez que el cursor deja de estar situado sobre ellos, regresarán a su estado normal. El cursor mantendrá la forma de puntero.
- vi. Los íconos se corresponderán con las tareas a realizar.
- vii. Los botones en todas las barras de herramientas se corresponderán con los comandos de un menú.
- viii. En caso necesario un comando del menú tendrá una secuencia de “hot-keys” (teclas de acceso rápido), que lo invocarán cuando sean activadas por el usuario.
- ix. En cajas de diálogo tabuladas, no se emplearán abreviaturas en los nombres de las lengüetas o pestañas.
- x. En cajas de diálogo tabuladas, las lengüetas o pestañas se podrán identificar con combinaciones de “hot-keys” apropiadas para su tarea. (Opcional).
- xi. En las cajas dialogadas no existirán “hot-keys” duplicadas.
- xii. Las lengüetas serán colocadas horizontalmente.
- xiii. Se propiciará el uso apropiado de la tecla Escape. (opcional).
- xiv. Un botón de comando se habilitará cuando la acción se corresponda con su uso y no en otras ocasiones, donde se pone en peligro la información por un posible error de utilización.
- xv. Los nombres de los botones de comandos no serán abreviaturas y tendrán un significado claro para los usuarios del sistema.
- xvi. Los botones de comando serán de tamaño y forma similares.
- xvii. Los nombres de los botones de opción (botón de radio o radiobutton) y de las cajas de opción no serán abreviaturas.
- xviii. El usuario siempre tendrá un acceso directo a la ayuda de la interfaz con la que interactúe.
- xix. El sistema asistirá al usuario de forma efectiva para hacer las tareas en caso de que no haya una única manera de hacerlas.
- xx. Sólo se presentará al usuario la información que realmente necesita.
- xxi. Existirán diferentes niveles de ayuda.
- xxii. En un formulario los controles estarán a 10 píxeles de margen en la horizontal uno con otro, y a 10 píxeles de margen en la vertical. Los tamaños variarán según la especificación del

tipo de dato y la cantidad de caracteres requeridos, además, se establecerá una alineación uniforme.

- xxiii. La etiqueta siempre se colocará encima del control, excepto en pantallas de autenticación, en barras de herramientas y otros como el radiobutton, y el checkbox que se colocará seguido o delante del control a la parte derecha o izquierda.
- xxiv. Pantallas de autenticación: Alineado a la izquierda.
- xxv. Para modificar valores que se muestran en un grid, se empleará la barra de herramientas, donde al presionar algún botón se abrirá una ventana (empleando el componente que corresponda) y se realizarán los cambios pertinentes. Siempre que el usuario estime conveniente, se brindará la posibilidad de editar en el propio grid.
- xxvi. Para editar en un grid utilizará el doble clic, si el grid no es editable, al dar doble clic sobre una tupla, habilitará una ventana para modificar el registro siempre que este lo requiera.
- xxvii. En todos los casos el idioma definido por el sistema se empleará en todos los textos. Si es español no se muestran palabras en inglés u otro, y de igual forma para los demás.
- xxviii. La ventana que se emplea para adicionar un elemento siempre comenzará el título con la palabra “Adicionar seguida por el elemento específico” Ej>Adicionar un estudiante.
- xxix. La ventana que se emplea para modificar un elemento siempre comenzará el título con la palabra “Modificar seguida por el elemento específico” Ej>Modificar un estudiante.
- xxx. La fecha siempre se utilizará en formato de d/m/Y (01/10/2008).

#### **Reglas a cumplir en componentes de tipo “grid”:**

1. Siempre utilizará una máscara (loadMask). Indicando siempre que se encuentre cargando.
2. La etiqueta mostrará un mensaje que dirá “Cargando...”.
3. Siempre que se cargue un grid, automáticamente se seleccionará la primera fila del conjunto de resultados.

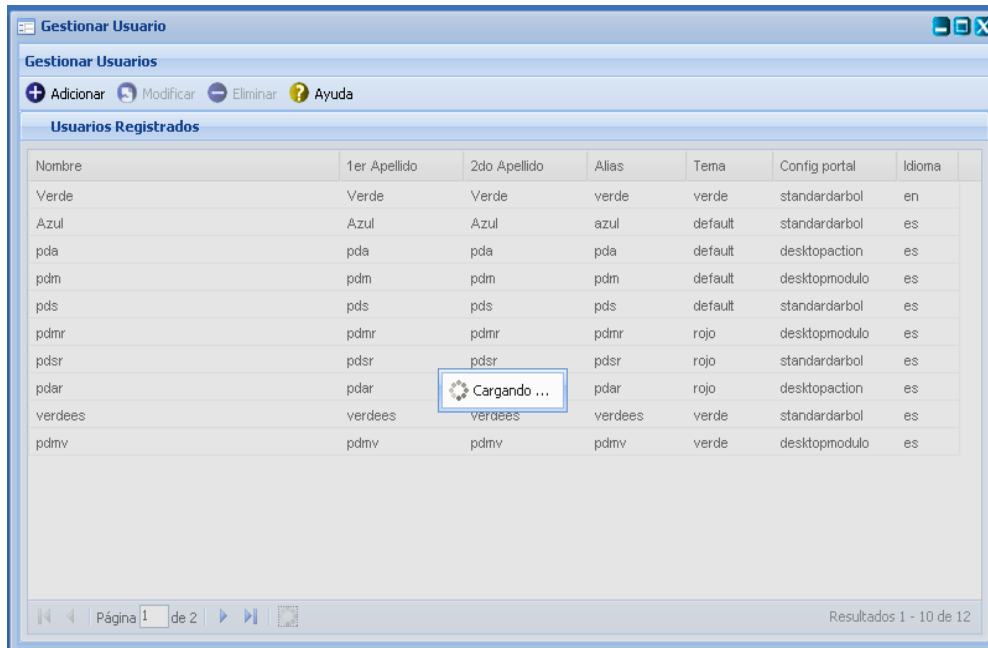


Figura 5. Componente GridPanel.

**Reglas a cumplir en componentes de tipo “formulario”:**

En un formulario siempre se utilizará una máscara (waitMsg) Indicando cuando se encuentre ocupado intercambiando con el servidor.

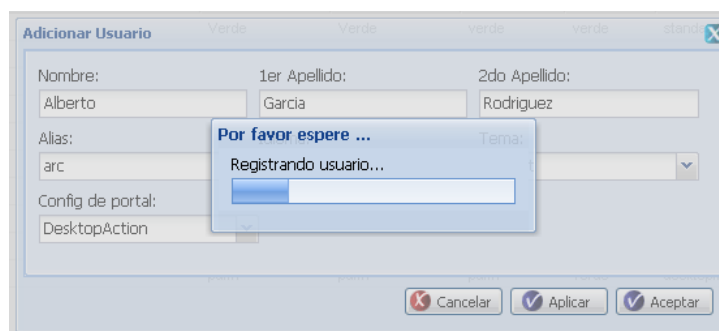


Figura 6. Componente FormPanel (formulario).

**Reglas a cumplir referente a “validaciones”:**

1. Siempre se habilitará validaciones a nivel de expresiones regulares.

2. El mensaje de validación de un campo incorrecto se colocará en un “tool-tips” correspondiente.

A screenshot of a web form with a light blue background. It contains four input fields arranged in a 2x2 grid. The top-left field is labeled 'Nombre:' and is empty. The top-right field is labeled '1er Apellido:' and contains a single vertical bar. The bottom-left field is labeled 'Alias:' and is empty. The bottom-right field is labeled 'Idioma:' and has a dropdown arrow. A red dashed border highlights the 'Alias:' field, and a tooltip with a red exclamation mark icon and the text 'Este campo es obligatorio' is positioned over it.

**Figura 7.** Validaciones utilizando Tool-Tips.

3. Si no se puede hacer una expresión regular, se validará con una función propia, pero se marcará el elemento en un estado válido o no válido empleando el marcado del Framework.

A screenshot of a web form with a light blue background. It contains two input fields side-by-side. The left field is labeled 'Municipio:' and contains the text 'khgsdakf'. The right field is labeled 'Provincia:' and contains the text 'jkhgkjasfg'. Both fields are highlighted with a red dashed border.

**Figura 8.** Validaciones utilizando marcado del framework Ext.

4. El tab index corresponderá con el orden de los campos de izquierda a derecha por filas completas y de arriba hacia abajo.
5. Si el usuario realiza un clic en aplicar o aceptar y en el formulario hay campos no válidos, automáticamente los que falten por validar se marcarán como no válidos y se mostrará un mensaje de error que dirá:

“Por favor verifique nuevamente que hay campo(s) con valor(es) incorrecto(s)”.

**Definición de mensajería para Ext-2.2 al español:**

Los mensajes se mostrarán por defecto a nivel de Framework si tiene incluido el singleton: **Ext.QuickTips.init()**.

Esto quiere decir que puede prescindir de colocarlos cada vez que los necesite siempre que lo utilice por defecto, y de esta forma evita la introducción de errores gramaticales y de riesgos que puedan ocurrir. Posteriormente se describen la sintaxis de dichos mensajes:

- **displayMsg**:"Mostrando {0} - {1} de {2}"
- **emptyMsg**:"0 Resultado a mostrar"
- **beforePageText**:"Página"
- **afterPageText**:"de {0}"
- **firstText**:"Primera Página"
- **prevText**:"Página anterior"
- **nextText**:"Página siguiente"
- **lastText**:"Última Página"
- **refreshText**:"Actualizar"
- **groupByText**:"Agrupar este campo"
- **showGroupsText**:" Mostrar en grupos"
- **waitTitle**:"Por favor espere ..."
- **invalidText**:"El valor en el campo no es correcto"
- **minLengthText**:"El tamaño mínimo requerido es {0}"
- **maxLengthText**:" El tamaño máximo requerido es {0}"
- **blankText**:"Este campo es obligatorio"
- **regexText**:"Este valor es incorrecto"
- **invalidText** : "Este valor es incorrecto"
- **dayNames**:["Domingo","Lunes","Martes","Miércoles","Jueves","Viernes","Sábado"],
- **monthNames**:["Enero","Febrero","Marzo","Abril","Mayo","Junio","Julio","Agosto","Septiembre","Octubre","Noviembre","Diciembre"]
- **todayText**:"Hoy"
- **todayTip**:"{0} (Barra de espacio)"
- **ok**:"Aceptar"
- **cancel**:"Cancelar"
- **cancelText**:"Cancelar"

- **yes:**"Si"
- **no:**"No"
- **minText:**"Esta fecha es menor que el rango mínimo establecido"
- **maxText:**"Esta fecha es mayor que el rango máximo establecido"
- **disabledDaysText:**"Deshabilitado"
- **disabledDatesText:**"Deshabilitado"
- **nextText:**"Próximo mes (Control+Derecha)"
- **prevText:**"Mes anterior (Control+Izquierda)"
- **monthYearText:**"Seleccione un mes (Control+Arriva /Abajo para desplazarse por los años)"
- **minText:**" El valor mínimo es {0}"
- **maxText:**"El valor máximo es {0}"
- **nanText:**"{0} no es un valor correcto"
- **maxText:**"La fecha tiene que ser menor o igual que {0}"
- **invalidText:**"{0} no es válido – tiene que ser en formato {1}"
- **monthYearText:**"Seleccione un mes (Control+Arriva/Abajo para desplazarse por los años)"
- **sortAscText:**"Orden ascendente"
- **sortDescText:**"Orden descendente"
- **columnsText:**"Columnas"

A continuación se da una definición de los ficheros que componen la capa de presentación, los que en su conjunto serán los encargados de gestionar la forma en que se muestra la información a los usuarios, y cómo manipularán los datos tratados a partir de la automatización de un proceso.

### Archivos de la capa de presentación

Los ficheros que conforman esta capa son:

- **Archivos de tipo "\*.js" (javascript)**

Se ejecutan en la parte del cliente, y pueden contener validaciones, la gestión de los componentes EXT que se empleen, entre otros.

- **Archivos de tipo "\*.CSS" (plantillas de estilo)**



Contienen las clases que se definen en las plantillas asociadas a un control, las que se encargan de atribuir los valores correspondientes a las propiedades de este o del componente referenciado por la clase.

- **Archivos de tipo ".phtml" (Plantillas con código PHP+HTML)**

Contiene la definición de los elementos, componentes u objetos nativos de HTML necesarios para gestionar la interfaz. Además puede contener código php con valores necesarios para su manipulación.

- **Archivos de tipo ".png" o ".gif" (imágenes).**

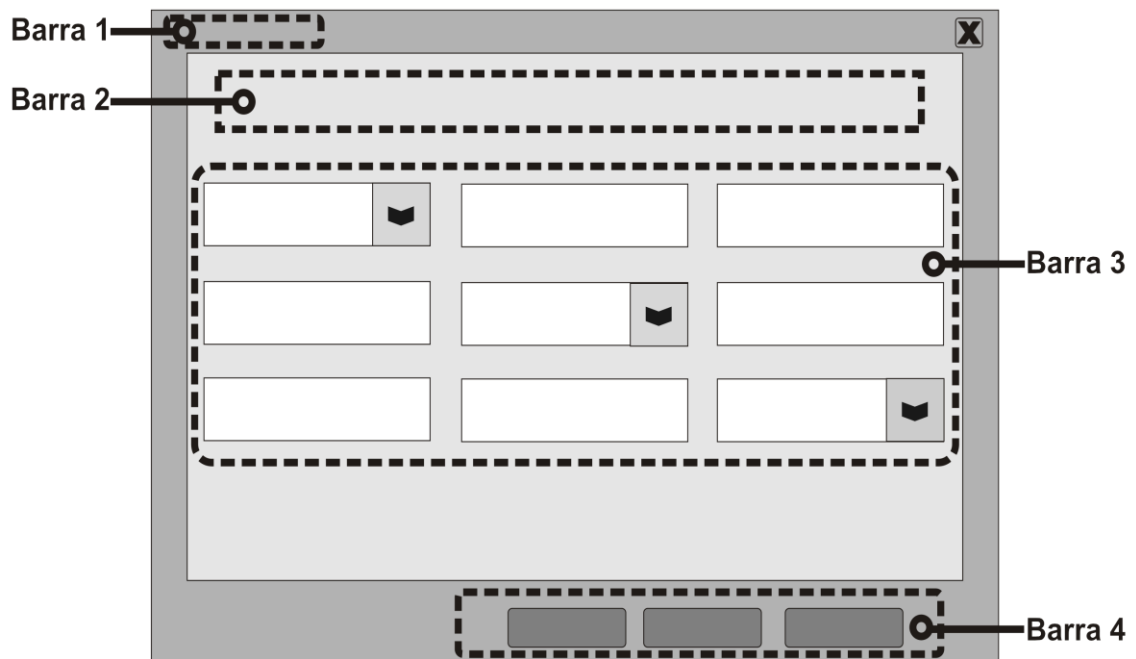
Son las imágenes que se utilizan en el sistema. Ejemplo: íconos, banner, entre otras.

### Zonificación

La zonificación se define como la división de un área en varias subáreas o zonas caracterizadas por una función determinada. [29]

### **Zonas de un Formulario**

La región de los formularios se divide en varios elementos. Estos se dividirán en forma de barras de contenido horizontales, teniendo en cuenta que el diseño en la Web está dado por la organización que se proporcione a los componentes, conformando una ventana y se estructurarán de la siguiente manera en dependencia de la necesidad y de la complejidad del mismo.



**Figura 9.** Zonificación de un formulario.

**Barra 1: Título del formulario**

El título debe estar en correspondencia con el objetivo del formulario que se define, juega con su contenido y lo identifica, estará alineado a la izquierda, excepto cuando sea un grid (que es opcional colocar el título).

**Barra 2: Barra de herramientas para manipulación del formulario**

Es opcional, y se colocan botones con texto o ícono (o ambos) que determinan acciones para la manipulación de los datos. Contiene 2 áreas (**A1**: Botones con funciones específicas relacionadas con cada formulario, **A2**: traza de ubicación para el usuario).

**Barra 3: Controles del formulario**

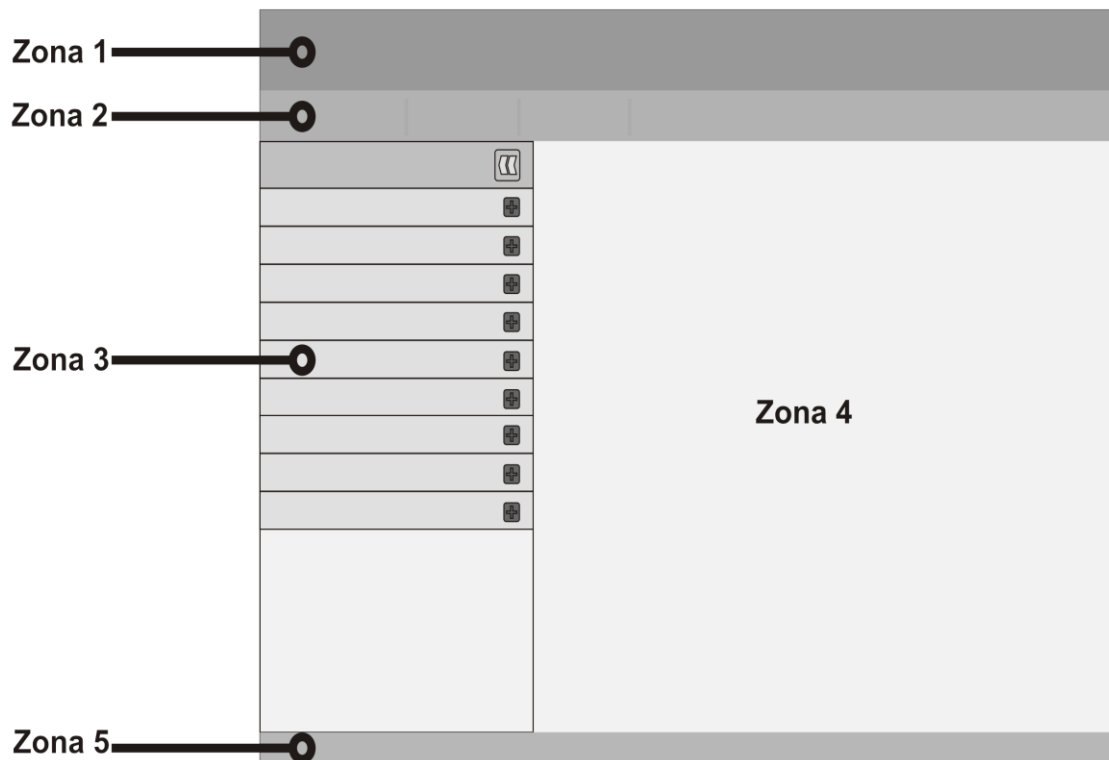
Pueden ser Inputs, ComboBox, TextBox, CheckBox, Labels, RadioButton u otros, (GridPanel, TreePanel, etc.) Una misma área horizontal se puede dividir según la necesidad de espacio siempre que no pierda la estructura principal de la pauta de diseño.

### **Barra 4: Botones**

Se ubicarán de izquierda a derecha los botones de cancelación y luego de aceptación alineados a la parte derecha.

### **Zonificación del portal**

Se definen 5 zonas permanentes, que en su conjunto aprovechan de manera óptima el espacio de trabajo, teniendo en cuenta que para los sistemas de gestión esta condición es de vital importancia.



**Figura 10.** División del portal en varias zonas.

**Desglose.**

- **Zona 1**

Soporta el Banner. Contiene subdivisiones donde se coloca información general y específica de cada Sistema.

- **Zona 2**

Barra de Menú, contiene 3 áreas.

(A1: Menú, A2: Nombre del usuario en cuestión, A3: Botón definido para cerrar la aplicación).

- **Zona 3**

Acordeón de navegación vertical.

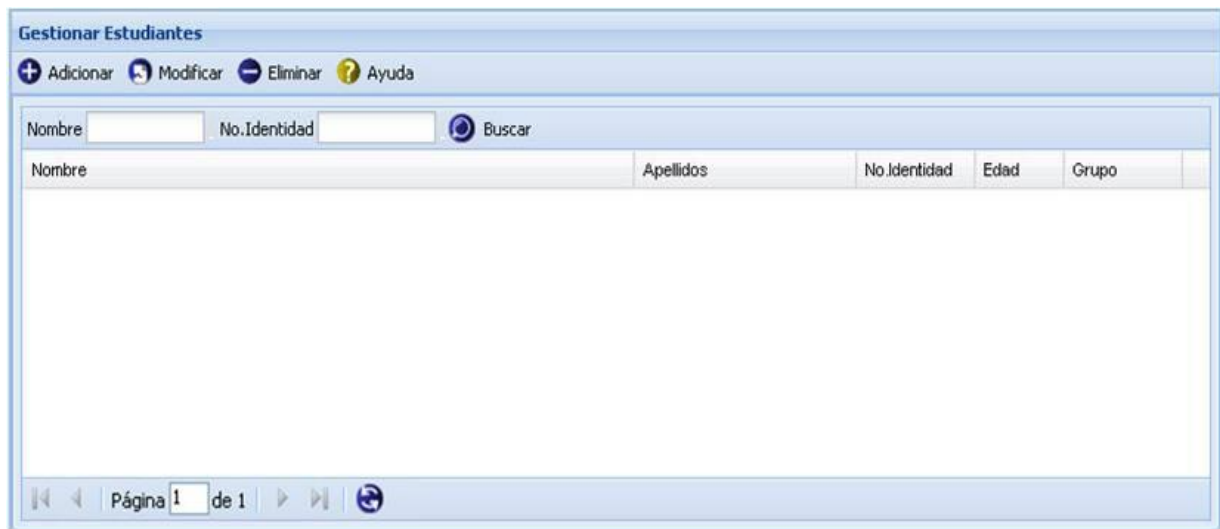
- **Zona 4**

Zona de trabajo editable. Se colocarán y mostrarán todos los controles asociados a los formularios que implementarán las funcionalidades del sistema.

- **Zona 5**

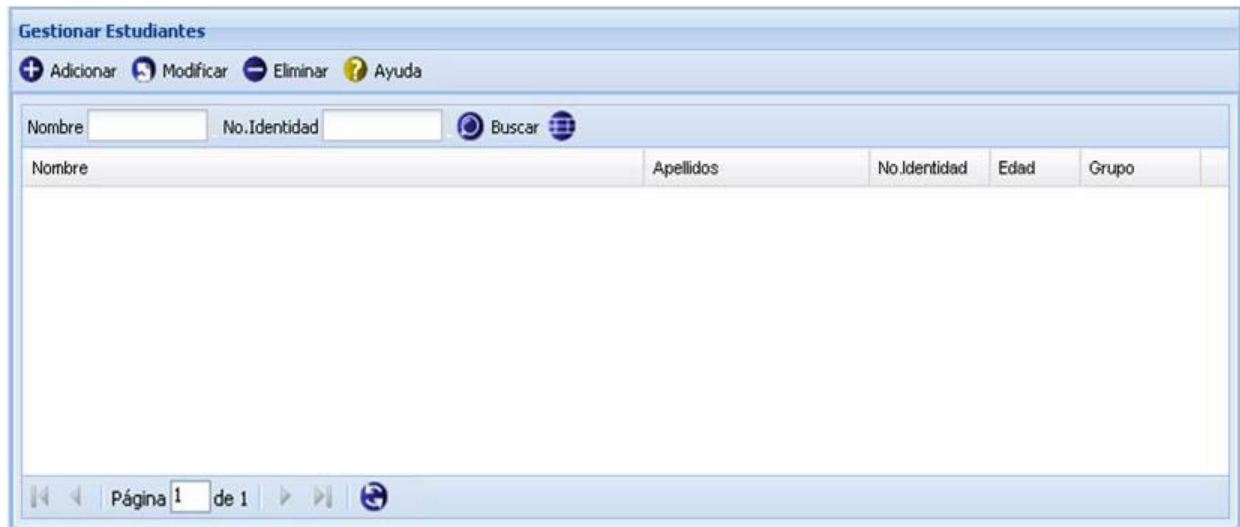
Barra de estado del navegador.

### Ejemplos de interfaces



**Figura 11.** Pantalla de búsqueda.

Esta es la combinación de una pantalla típica de gestión además, donde se puede apreciar que las acciones sobre el conjunto de resultados gestionados, se colocan encima del componente que lo gestiona. Generalmente va a ser adicionar, modificar, eliminar y ayuda.



**Figura 12.** Pantalla de búsqueda.

En el caso anterior sólo se muestra en la barra de búsqueda dos campos, en caso de que sean muchos y aun así se necesite filtrar y esto pone en riesgo la visibilidad, se agrupan los campos críticos y se coloca una búsqueda avanzada a la parte derecha del botón que inicia la acción de buscar como se muestra.

El ícono aun no está por lo que temporalmente se coloca el icono falta.png que es el destinado para los que aún no cuentan. El tool-tips se especifica como Búsqueda avanzada, y cuando se le de clic, abre una ventana con los restantes campos y los botones cancelar y aceptar.



**Figura 13.** Pantalla típica de inserción de datos.

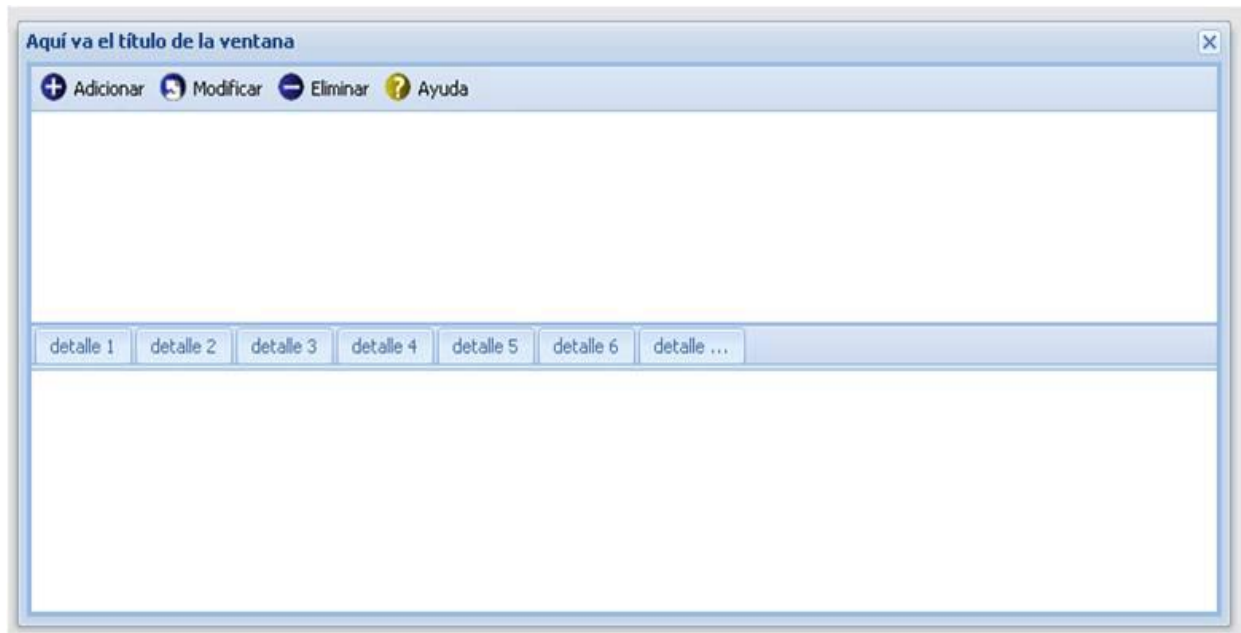
Aquí se pone de manifiesto como cada vez que se vaya a insertar datos en un formulario, se colocan los botones como se indica, cancelar, aplicar y aceptar, cada uno aplicando el principio que se define para su uso adecuado.



The image shows a Windows-style dialog box titled "Modificar Estudiante". It contains five text input fields arranged in two rows. The first row has "Nombre:", "Apellidos:", and "Grupo:". The second row has "No. Identidad:" and "Edad:". At the bottom right, there are two buttons: "Cancelar" with a red 'X' icon and "Aceptar" with a blue checkmark icon.

**Figura 14.** Pantalla típica de modificación de datos.

Aquí se pone de manifiesto como cada vez que se vaya a modificar datos en un formulario, se colocan los botones como se indica, cancelar y aceptar, cada uno aplicando el principio que se define para su uso adecuado, empleando el mismo formulario de inserción de datos, cambiando el título de la ventana y los botones.



**Figura 15.** Pantalla de tipo máster-detalle.

En esta pantalla se visualizará los detalles de operación que el usuario haya especificado, ofreciendo mas de un detalle que va estar compuesto por tabs. Destacar que dicha interfaz contiene además botones, los cuales van estar sometidos a una función determinada.



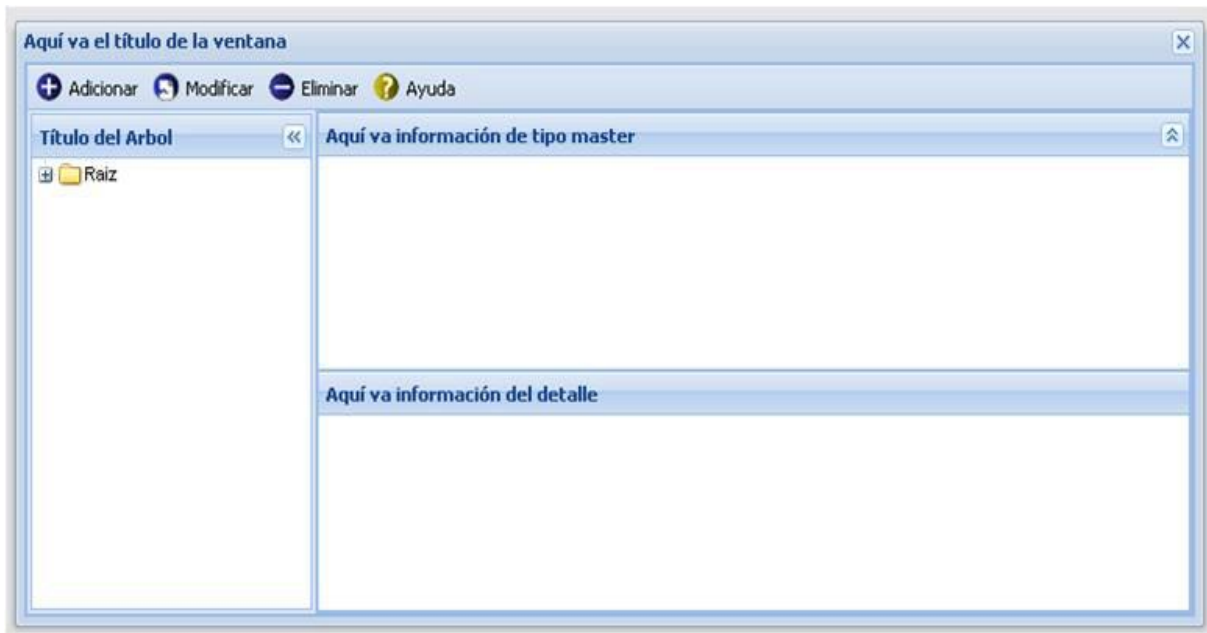
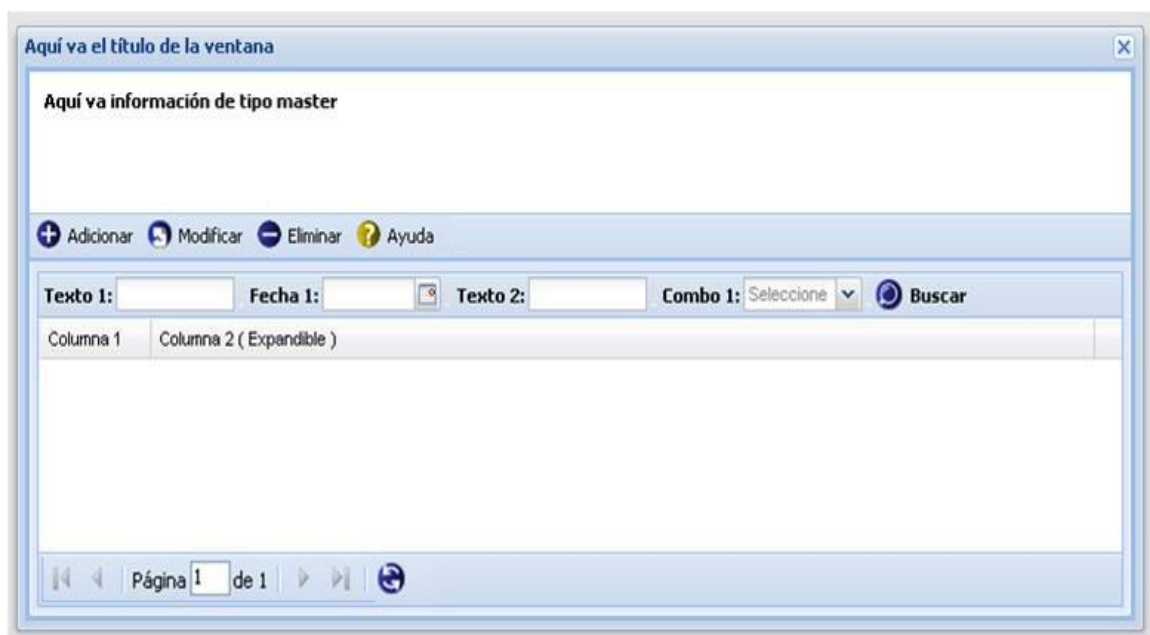


Figura 16. Pantalla de tipo máster-detalle-complejo.



Figura 17. Pantalla de tipo máster-detalle-complejo-grid.

En esta interfaz están representados dos de los componentes primarios como es el caso del TreePanel ubicado en la parte derecha de la interfaz, éste es el encargado de contener la información requerida para poder realizar las operaciones, también se visualizan el máster detalle en la parte superior de la pantalla y debajo está representado un grid que está compuesto por un TextField para la entrada de datos, un DateField para elegir la fecha y un ComboBox .Estos constituyen los campos para un criterio de búsqueda.



**Figura 18.** Pantalla de tipo máster-detalle-grid.

Esta pantalla está conformada principalmente por un grid que mostrará cierta cantidad de información, que a su vez está compuesto por varios componentes como son, TextField, DateField y ComboBox utilizando estos como criterio de búsqueda, mostrando los detalles de los resultados a seleccionar en el componente tipo máster-detalle.

**Figura 19.** Pantalla de filtro avanzado.

La interfaz que se muestra representa la funcionalidad de filtrar contenido donde son utilizados varios ComboBox.

### **Iconografía**


La iconografía propuesta fue dictaminada por el diseñador gráfico del proyecto ERP-CUBA, donde siguió una serie de pautas para la confección de los mismos. Los íconos mantienen un estándar muy fácil de comprender, como es el caso que para visualizar los módulos de una manera gráfica se utilizaron íconos con forma cuadrada y dentro de esta una pequeña forma que distinga al módulo en cuestión.


Ej.: **Seguridad** 


También es importante señalar que para representar visualmente las funcionalidades se definieron estos elementos con una forma redonda, dentro de ella una simbología que represente a dicha función. Tal es el caso de: **aceptar**


### Botones para colocar en topToolbar (tbar)

Se normó el uso de los botones en las barras de tareas y que sólo son estos los que se pueden usar que a continuación se detallan:

**Abrir:** Dependiente del contexto. Abre un objeto determinado a lo que se haga referencia en el formulario. Estos pueden ser: ficheros con información, documentos archivados, objetos de configuración. En cualquier caso se habilita la interfaz correspondiente para manejar el objeto a contener. 


**Actualizar:** Realiza la solicitud a la fuente de datos que determina lo que se muestra en la interfaz en un instante determinado. Se receptiona el pedido y se vuelve a recargar la interfaz con los datos actualizados. 


**Adicionar:** Dependiente del contexto. Comienza la acción de inserción de los objetos que se manejan en el formulario. Estos pueden ser: productos, equipos, personas. En cualquier caso se habilita la interfaz correspondiente para manejar los datos contenidos en la inserción. 


**Deshabilitar:** Dependiente del contexto. Cambia el estado del objeto de activo a inactivo, en otras palabras, es inhabilitado de alguna forma tal, que se diferencia de los demás. 


**Buscar:** Dependiente del contexto.


Si está ubicado en 'tbar' habilita inmediatamente una interfaz de búsqueda relacionada con los datos que se manejan en el formulario.


Si está ubicado en una interfaz típica de búsqueda, este indica el comienzo de envío de la petición al servidor. Es quien da paso a ejecutar la acción de buscar hacia este. 


**Copiar:** Dependiente del contexto. Copia la configuración de los datos de un objeto de los que se esté manejando en el formulario. 


**Crear reporte:** Invoca la creación de un reporte determinado por una selección y muestra una interfaz con los datos solicitados. 


**Duplicar:** Dependiente del contexto. Copia la configuración de los datos de un objeto de los que se este manejando en el formulario y además crea un nuevo objeto de iguales características con el mismo contenido copiado. 


**Eliminar:** Dependiente del contexto. Elimina la existencia física de un objeto determinado por el contenido del formulario. 


**Exportar:** Dependiente del contexto. Crea un fichero con una estructura de datos determinada, y se archiva fuera de la aplicación. 


**Guardar:** Dependiente del contexto. Guarda persistentemente en la BD la configuración de un objeto determinado. 


**Importar:** Dependiente del contexto. Toma un fichero con una estructura de datos determinada, lo abre, lo analiza, y muestra una interfaz con la configuración de lo que se está añadiendo a la aplicación. 

**Imprimir:** Dependiente del contexto. Envía a la impresora una configuración determinada por una pantalla de datos para plasmarla en soporte de papel. 


**Modificar:** Dependiente del contexto. Habilita una interfaz con la configuración y datos actuales de un objeto para modificarlo. 


**Nuevo:** Dependiente del contexto. Habilita las condiciones para comenzar desde el principio con la inserción de un determinado objeto. 


**Pegar:** Dependiente del contexto. Toma un objeto de configuración previamente almacenado y lo inserta en el formulario. 


**Ver:** Dependiente del contexto. Muestra en una interfaz, detalles de algún objeto que en ese momento no es visible para el usuario. 

### Barra inferior buttonToolBar (bbar)

**Aceptar:** Dependiente del contexto. Guarda los datos que se reflejan en un formulario y pasa a otra interfaz. Si se emplea en una ventana flotante, cierra la ventana. 


**Anterior:** Dependiente del contexto. En una interfaz determinada, regresa un paso atrás en una fase que continúa una secuencia lógica de tratamiento de datos. 


**Aplicar:** Dependiente del contexto. Guarda los datos que se reflejan en un formulario. Si se emplea en una ventana flotante, limpia los campos y abre paso para una nueva inserción. 


**Ayuda:** Invoca la ayuda específica o general sobre la interfaz o sobre el sistema. 

**Cancelar:** Dependiente del contexto. 

- En una ventana, cierra la ventana descartando los datos introducidos.
- Durante la ejecución de un proceso, detiene la acción y regresa a un estado anterior.

**Limpiar:** Pone en un estado cero la entrada de datos de un formulario en una interfaz. 

**Salir:** Dependiente del contexto. Sale de la interfaz donde se encuentre el usuario. 

**Siguiente:** Dependiente del contexto. En una interfaz determinada, da paso a la próxima fase que continúa una secuencia lógica de tratamiento de datos. 

**Teclas calientes (Hot keys). Definición.**

Las teclas calientes o como se le conoce en idioma inglés Hot keys, no son más que teclas de acceso rápido a una funcionalidad, que pueden ser personalizadas por el usuario.

**Tbar.**

<b>Comando</b>	<b>Combinación</b>
<b>Abrir.</b>	Alt+A
<b>Adicionar.</b>	Alt+I
<b>Buscar.</b>	Alt+B
<b>Copiar.</b>	Alt+C
<b>Crear reporte.</b>	Alt+R
<b>Duplicar.</b>	Alt+D
<b>Eliminar.</b>	DEL
<b>Exportar.</b>	Alt+E
<b>Guardar.</b>	Alt+G
<b>Imprimir.</b>	Alt+P
<b>Nuevo.</b>	Alt+N
<b>Pegar.</b>	Alt+V

**Bbar.**

<b>Aceptar.</b>	Enter
<b>Limpiar.</b>	Alt+L
<b>Salir.</b>	Alt+S

## 2.4 CONCLUSIONES

En este capítulo se destacaron tres aspectos fundamentales que contribuyeron al desarrollo del mismo, como fueron la línea base de la arquitectura de presentación, estándar de usabilidad de escenarios y normas y estándares en presentación.

Se definieron varios tópicos en la línea base de presentación; por ejemplo:

- Nomenclatura para definición componentes de ExtJS, donde se define la nomenclatura de los componentes a utilizar en un fichero **.js**.
- Validaciones Utilizando Ext-2.x.x. , en el cual se expone las validaciones de los datos en la parte del cliente utilizando Java Script.
- Definición de Código en Ficheros, en donde se especifica como crear una plantilla **.phtml** y un fichero **.js**, así como la definición de sus elementos.
- Multi-Tema , donde permite que el sistema en general sea configurable para varios temas
- Multi- lenguajes, permite que el sistema sea configurable para varios lenguajes.



Otros de los temas que se trató en el capítulo fue el estándar de usabilidad de escenario, donde se tocaron puntos cruciales como son:

- El estándar de usabilidad de los menús.
- El etiquetado que representa la utilización de términos y/o expresiones de varios términos para identificar el contenido informativo.
- Estructura de las áreas de trabajos del componente Portal.

Finalmente se mostró la definición de normas y estándares de desarrollo en la capa de presentación y se destacan las principales pautas a tener en cuenta que rigen el desarrollo a nivel de interfaces.

## CAPÍTULO III RESULTADOS DE LA SOLUCIÓN

### INTRODUCCIÓN

Una vez desarrollado el capítulo anterior es necesario exponer los resultados de esta, como son el rendimiento del componente Portal, en el cual se describen brevemente las pruebas de conceptos de carga y estrés que se realizó a dicho componente. También se muestra la reutilización de componentes en presentación.

### 3.1 REUTILIZACIÓN DE COMPONENTES PRESENTACIÓN

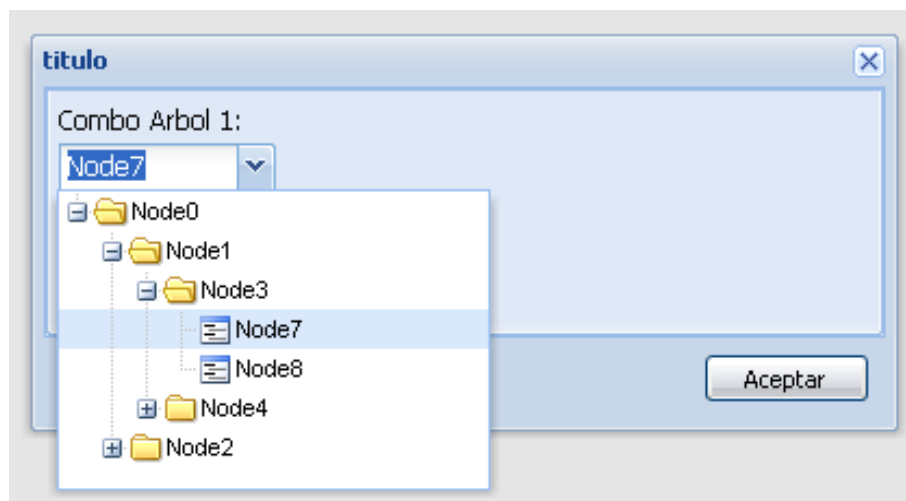
La reutilización de los componentes de presentación en el componente Portal se basa en la integración de componentes, es decir está implicado todo lo referente a nivel de interfaces, a nivel de lógica de negocio y mediante servicios. En este epígrafe se aborda de forma superficial los otros puntos que conforman la integración en general, centrando la principal atención en la integración a nivel de interfaces.

Se define como integración de interfaces al modo de utilizar los elementos y bondades que pueda implementarse en una interfaz que responde a una lógica determinada, desde otra totalmente independiente que necesite del servicio. Es similar a utilizar un servicio Web, pero no a nivel de lógica sino a nivel de interfaz.

En el capa de presentación específicamente en la realización de interfaces intervienen dos componentes para la realización de la integración a nivel de interfaz, estos son: **Componente combo-iframe-árbol** y **Componente ventana-iframe**. Donde el primer componente se utiliza cuando se necesita seleccionar y tomar de una interfaz determinada información y esta información tiene una estructura arbórea, de selección simple. El segundo componente se utiliza cuando de un conjunto de resultados determinados se necesita tomar una información específica, o toda la información del record (tupla), y teniendo una selección simple o múltiple, pero lo fundamental es

que permite utilizar una lógica totalmente independiente y que sea llamada dentro de cualquier final.

### 3.1.1 Especificación del componente combo-iframe-árbol.



**Figura 20.** Componente combo-iframe-árbol.

#### **Uso del servicio.**

El componente Ext.UCID.Combolframe es una extensión de la clase Ext.form.ComboBox que tiene dentro de sus características distintivas, que es un combo que despliega un iframe<sup>15</sup> y dentro, al igual que todos los componentes de Ext cumple con las especificaciones de la clase padre y está preparado para desplegar un árbol e integrarse para ser extendido con nuevas funcionalidades.

Para utilizar este componente sólo se tiene que asegurar que esté incluido en la plantilla luego de ext-base.js y ext-all.js el fichero ucid-combo-iframe.js que está en la carpeta:

`\repo\VDesarrollo\erp\comun\frameworks\UCID\js` y de esta forma se puede instanciar y utilizar el componente. En la plantilla (.phtml) utilizando el framework se tiene que incluir:

```
<script type="text/javascript" src="<?php echo $this->dir_ucid;?>js/ucid-combo-lframe.js"></script>
```

### **Especificaciones de uso.**

El objetivo de este tópico es especificar las descripciones de las propiedades extendidas. Se crea una instancia, se almacena una referencia y luego la utiliza como a continuación se menciona.

**Nota:** Cuando aparece opcional significa que es la misma propiedad que implementa la clase padre, la cual no tienen implicación alguna para la correcta integración.

Es importante subrayar que el combo es colocado dentro de un formulario, al realizar submit (enviar), se envían los valores, y si se quiere acceder al nodo seleccionado en el árbol, una vez que se selecciona, solo se tiene que acceder a la propiedad del combo que se denomina "nodoSelected" y cada vez que seleccione un nuevo nodo en el árbol, este cambiará su valor y se podrá acceder a través de `combolframe1.nodoSelected` y estará referenciando directamente el nodo en selección en el árbol. Para acceder a las propiedades y configuraciones es recomendable utilizar las bondades de `Ext.tree.TreeNode` para manipularlo.

Primeramente se define la variable `combolframe1` donde se almacena una referencia al objeto que se construye a partir de la configuración especificada.

```
var comboIframe1 = new Ext.UCID.ComboIframe({
```

También es importante destacar la definición del identificador (id) que se va a enviar para el servidor (Si se tiene el componente dentro de un formulario) con el valor del id del nodo seleccionado en el árbol (es necesario capturarlo). Posteriormente el id que se va a enviar para el servidor (Si se tiene el componente dentro de un formulario) con el valor del text del nodo seleccionado en el árbol, y el id a partir del cual se va a capturar el componente dentro del Iframe, por lo que es clave en el proceso. Sin embargo se crea el id a partir del cual se va a capturar el iframe<sup>14</sup> dentro del componente, por lo que es clave en el proceso (obligatorio en cualquier caso).

```
idIframe: 'comboIframe1',
```

Se define la acción a través de la que se va a recibir la plantilla a utilizar dentro del iframe. Es necesario colocar esta línea de código.

```
url: 'tpCasoUso.html',
```

El evento `onSelectNodo` al que si le coloca una función, esta será llamada automáticamente con el nodo seleccionado. En este componente el evento `onSelect` es anulado, porque no selecciona de una lista de ítems comunes, pero si se utiliza esta propiedad, simula el evento anterior de la misma manera.

Es importante destacar que se le puede añadir cualquier propiedad de las que implementa un combo ordinario de Ext siempre y cuando la funcionalidad no afecte el funcionamiento de su extensión.

### **Implementación del servicio.**

Dado los elementos antes expuestos se tiene un combo que despliega un iframe y dentro de este se reenderiza la interfaz que devuelve una acción, pero ¿Cómo lograr a nivel lógico que una interfaz se comunique con otra sin tener que implementarla, sólo especificando cierta configuración de conexión? Pues a continuación se expone detalladamente.

El componente antes mencionado es sólo de selección y por tanto se puede seleccionar sólo un elemento a la vez. Es por eso que se tiene que implementar de la siguiente manera:

Se define un árbol (TreePanel) con los elementos básicos de configuración y con sus siguientes especificaciones como es el caso de la línea de código:

```
dataUrl: '../xml/damenodoarbolcombo.php'
```

Se utiliza para darle obtener la información apropiada para el funcionamiento de éste.

Para poder visualizar dicho árbol se debe incluir el Viewport que es un componente contenedor especializado en la representación de la zona visible de la aplicación en el navegador.

Una vez definidos los componentes anteriores se garantiza la integración de la manera que se expone a continuación.

Se incluye la plantilla (.phtml), luego de la inclusión de los ficheros de extjs (ext-base.js y ext-all.js) el fichero ucid-integra-interfaz.js como sigue:

```
<script type="text/javascript" src="<?php echo $this->dir_ucid;?>js/ucid-integra-interfaz.js"></script>
```

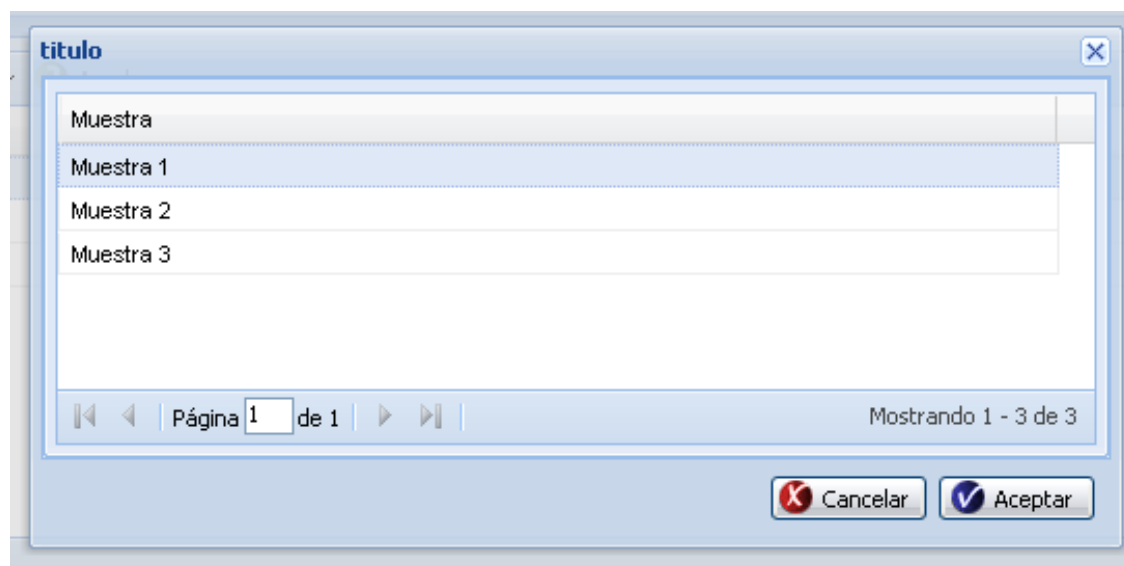
Con este fichero incluido el objeto **Ext.UCID** va a garantizar la comunicación entre las interfaces a través de la propiedad (integralInterfaz) con los parámetros necesarios.

Se verifica la existencia del componente que tiene el iframe, donde la variable global `window.idcomboiframe` va a contener el id del componente que tiene el iframe (combo) la cual se pasa automáticamente a crearse con la configuración definida. Después se pregunta, si tiene algún valor, si es así es porque la interfaz fue invocada correctamente por un combo definido, se captura en la variable `comboParent` y se consulta si existe la variable `comboParent` y si tiene algún valor se puede integrar la interfaz, y para ello se llama la función que está definida en el fichero ucid-integra-interfaz.js, referenciándola a través del objeto `Ext.UCID.integralInterfaz` como sigue:

```
if (comboParent) Ext.UCID.integralInterfaz(tpCasoUso,comboParent);
```

Consecutivamente se le pasan los objetos en el orden que preside, donde el `TreePanel` es denominado como `tpCasoUso` y el combo es nombrado como `comboParent`. Esta función garantiza de todo lo demás.

### 3.1.2 Especificación del componente ucid-win-iframe.



**Figura 21.** Componente UCID-win-iframe.

### Uso del servicio.

El componente Ext.UCID.Winiframe es una extensión del componente Ext.Window que tiene dentro de sus características distintivas, es una ventana que en su body (cuerpo) tiene un iframe y dentro, al igual que todos los componentes de Ext cumple con las especificaciones de la clase padre y está preparado para mostrar un conjunto de informaciones en un grid.

Para utilizar este componente sólo se tiene que asegurar que esté incluido en la plantilla seguidamente de ext-base.js y ext-all.js, el fichero ucid-win-iframe.js que está en la carpeta: `\repo\VDesarrollo\erp\comun\frameworks\UCID\js` y de esta forma se puede instanciar y utilizar el componente. En la plantilla (.phtml) utilizando el framework solo tiene que incluir:

```
<script type="text/javascript" src="<?php echo $this->dir_ucid;?>js/ucid-win-iframe.js"></script>
```

### Especificaciones de uso.

El objetivo de este tema es especificar las descripciones de las propiedades extendidas. Las configuraciones que se especifican de acuerdo a como se necesite brindar el servicio. Tiene que

quedar siempre una constancia del contrato de cómo se implementa, permitiendo utilizarla posteriormente y una vez ya creada puede incrementarse, de tal manera que no afecte el funcionamiento en lo que se quiere desarrollar, pues de esta forma se le brindará soporte de manera centralizada.

Para el desarrollo de dicho componente se construyen los elementos apropiados para darle funcionalidad al winframe. Para ello se crea una instancia, se almacena una referencia y luego la utiliza como a continuación se describe.

Inicialmente se define la variable winframe donde se almacena una referencia al objeto que se construye a partir de la configuración especificada. Seguidamente se define el id que va a tener el componente ventana. Después se realiza la definición del id partir del cual se va a capturar el iframe dentro del componente, por lo que es clave en el proceso.

```
idIframe: 'idWinIframe1',
```

Además se incluye la definición de la acción a través de la que va a recibir la plantilla a utilizar dentro del iframe.

```
url: 'gpCasoUso.html',
```

La propiedad **params** se utiliza para colocar variables con algún valor que necesite el servicio para procesar algo. Estas variables se van a poder capturar dentro del iframe una vez que sean referenciadas con el formato que se determine. También para éxito de dicho componente se mencionan varios eventos como id son: el evento **onLoad** que se dispara cuando termine de cargar la página que se incluye dentro del iframe. A esta función se le puede pasar cualquier elemento como parámetro, depende del acuerdo al que se llegue con la parte que brinda el servicio y el evento **doAction** que se dispara cuando se realiza las acciones que se defina implicando un procesamiento o ejecutar algo en la interfaz. A esta función se le puede pasar cualquier elemento como parámetro, depende del acuerdo al que se llegue con la parte que brinda el servicio. Por último se incluye la definición de los botones de la ventana.



Generalmente van a ser los botones Aceptar y Cancelar, donde al oprimir el aceptar, realiza determinada acción si es necesario y cierra la ventana, y al oprimir cancelar, siempre cierra la ventana sin realizar ninguna acción, pero se deja a la dependencia de la necesidad del desarrollador.

Es importante destacar que se puede añadir cualquier propiedad de las que implementa una ventana ordinaria de Ext siempre y cuando la funcionalidad no afecte el funcionamiento de su extensión.

### **Implementación del servicio.**

Si el componente winIframe está compuesto por otro componente denominado GridPanel, en este se puede seleccionar uno o varios elementos a la vez, implementándose primeramente un store (almacenar) que es el encargado de proporcionar los datos de entrada de los componentes tales como el GridPanel, ComboBox, o el DataView, consecutivamente se realiza la definición del Grid, compuesto por configuraciones básicas del propio componente. Finalmente para visualizar dicho componente se incluye la implementación del viewport.

Para garantizar la integración si fuera llamada la vista con este interés, sólo debe realizar las especificaciones antes mencionadas en el tema implementación del servicio del subepígrafe **Especificación del componente combo-iframe-árbol** agregando la configuración particular de este subepígrafe como la pregunta condicional `if (winParent)`, que responde a si la variable existe y posee algún valor se puede integrar la interfaz, para ello se invoca la función que está definida en el fichero `ucid-integra-interfaz.js` que se puede referenciar a través del objeto `Ext.UCID.integralInterfaz`. Sucesivamente la variable invoca la función `onLoad` para que la página se cargue en el `Iframe`.

Posteriormente se da una breve explicación de los dos últimos aspectos que responden a la integración de componentes en si.

### **Interacción a nivel de lógica.**

Después de creado el componente en el fichero JavaScript, se tiene que haber implementado una acción que coincida con la definida en la propiedad “url” del componente donde se escribe el código como a continuación se muestra con un mínimo de detalle:

#### Implementación del servicio interno:

```
$this->pIntegrator->'sub-modulo-que-brinda-el-servicio'->'servicio-que-brinda' ();
```

```
/*ejemplo: $this->pIntegrator->nomina->gettiponominas ();
```

#### Implementación del servicio externo:

```
$this->integrator->'modulo que brinda el servicio'->'servicio que brinda' ();
```

```
ejemplo: $this->pIntegrator->contabilidad->arbolcuentas ();
```

#### Implementación del servicio de la interfaz:

Después de implementada la interfaz se define el servicio en el fichero (ioc.xml) correspondiente:

```
<modulo src:'deireccion del modulo'>// o submodulo
```

```
<servicio      reference:'dirección-de-la-interfaz,-si-no-es-la-misma-del-modulo'>
```

```
<inyector clase:'clase-del-servicio' método:'método de la clase' />
```

Esta clase está ubicada dentro de la carpeta bussines (negocio) del módulo, y cumple con las especificaciones del componente IOC para la integración como establecen las definiciones del Marco de Trabajo del proyecto.

Con esta breve explicación se avala la integración de componentes, donde de una manera u otra se manifiesta la reutilización de componentes de presentación ya que como se resaltó en un

principio aquellas personas o usuarios que utilicen estas bondades podrán ser capaces de avanzar con éxito a la hora del diseño de un componente o en la confección de un sistema en específico.

## 3.2 RENDIMIENTO

### Pruebas de Estrés al componente Portal

Las pruebas de carga y estrés al componente Portal del Sistema de Gestión de Entidades se realizan con el objetivo de descubrir ausencias de respuestas del servidor y medir los tiempos de respuestas del servidor de la aplicación, para, encontrar resultados negativos, los aspectos que consumen mayor parte de rendimiento del sistema y con ello trazar estrategias de programación que ayuden a mejorar esas ineficiencias. [30]

#### Con 20 hilos (usuarios concurrentes):

El sistema muestra un soporte de concurrencia de 20 usuarios, se realizan 960 peticiones simultáneas, que incurren en una transferencia de datos de aproximadamente 124385.6 KB/sec. Mostrando un tiempo de recuperación de aproximadamente 7.7 segundos. [30]

#### Con 50 hilos (usuarios concurrentes):

El sistema muestra un soporte de concurrencia de 50 usuarios, se realizan 2400 peticiones simultáneas, que incurren en una transferencia de datos de aproximadamente 132676.3 KB/sec. Mostrando un tiempo de recuperación de aproximadamente 8.2 segundos. [30]

#### Con 100 hilos (usuarios concurrentes):

El sistema muestra un soporte de concurrencia de 100 usuarios, se realizan 4800 peticiones simultáneas, que incurren en una transferencia de datos de aproximadamente 176127.5 KB/sec. Mostrando un tiempo de recuperación de aproximadamente 10.8 segundos. [30]

La principal causa de los errores en condiciones inferiores y superiores fueron de un 8.33 % donde son fallidas las peticiones a las siguientes direcciones: [30]

- */erp2/comun/frameworks/ExtJS/idioma/es/temas/default/images/s.gif*
- */repo/VDesarrollo/erp2/comun/frameworks/ExtJS/idioma/es/temas/default/images/images/desktop/hared/icons/fam/user.gif*

El componente Portal se sometió a pruebas de estrés en condiciones inferiores y superiores de forma exploratoria a las reales en las que se debe desplegar, para la cantidad de usuarios que soporta en estos momentos con las prestaciones anteriormente mencionadas, se considera insatisfactorio el rendimiento que muestra la aplicación. En general las peticiones tienen un tiempo de respuesta de 7 a 10 segundos aproximadamente. [30]

### 3.3 CONCLUSIONES

En este capítulo se abordó sobre la reutilización de componentes, donde la integración de estos es el núcleo principal de dicha reutilización. También se llevó a cabo la descripción de las pruebas de conceptos, de carga y estrés que se le realizaron al componente Portal, exponiendo finalmente su resultado en cuanto a rendimiento.

## CONCLUSIONES GENERALES

La realización de este trabajo posibilitó cumplimentar los objetivos para los cuales se desarrolló dicho trabajo. Se adquirieron conocimientos teóricos sobre la arquitectura de presentación en general, la cual forma parte de la estructura de la documentación técnica del componente Portal. Se definieron y describieron las pautas de diseño del componente, lo que permitió seguir un estándar de diseño para los desarrolladores en el desarrollo de componentes en el sistema integral de gestión de entidades CEDRUX. Las pruebas de conceptos realizadas al componente portal arrojaron valores insatisfactorios ya que se esperaba que el tiempo de respuesta este fuera en un intervalo de tiempo mínimo. También se identificó los contenidos para incluir en la definición de la documentación técnica, que fue incorporado en el API<sup>16</sup>.

Como resultado se obtuvo la documentación técnica y ayuda formal del componente Portal, apoyándose en la creación de un API con este fin. La creación de la solución propuesta arrojó un resultado exitoso en la utilización de este ya que varias líneas del proyecto ERP- CUBA la han utilizado brindándole una mejor ayuda en los aspectos técnicos del componente Portal.

## RECOMENDACIONES

Gracias a las ventajas que ofrece la documentación técnica del componente Portal, ayuda a un mejor entendimiento y funcionamiento de dicho componente, se sugiere el seguimiento de la solución propuesta expuestas en el trabajo de diploma. Por tal razón se hacen las siguientes recomendaciones:

- ✓ Seguir actualizando el API, preferentemente confeccionar y agregar el mapa de navegación del componente Portal.
- ✓ Utilizar el API aquellas personas que necesiten implementar un componente similar al componente Portal.
- ✓ Utilizar el presente trabajo como guía para proyectos o equipos de desarrollo o alguna persona en específico que desee documentar técnicamente un componente a nivel de presentación.

## REFERENCIAS BIBLIOGRÁFICAS

1. **Mendoza, Jorge A.** <http://www.divisait.com>. [En línea] 1 de Diciembre de 2000. [Citado el: 8 de Noviembre de 2008.]  
<http://www.divisait.com/contenidos/content.asp?contentid=604&nodeid=275&page=2>
2. **Polgar, Jana, Bram, Robert Mark y Polgar, Anton.** *Building and Managing Enterprise-Wide Portals*. s.l. : Idea Group Publishing, 2005.
3. *Evolving Portal Solutions*. **BEA Systems**. 2006.
4. *Plumtree blossoms: New version fulfills enterprise portal requirements*. **Patricia Seybold Group**. 1999.
5. **Phifer, Gene, Valdés, Ray y Gootzit, David.** *Portals Are the “Swiss Army Knives” of Enterprise Software*. s.l. : Gartner, 2006.
6. **Pérez, J.E.**, Introducción a Css (Bibliografía básica en teleformación). 2007 [Citado; Disponible en: <http://www.librosWeb.es/css>
7. **Zeldman, J.** Diciembre 2006 Diseño con estándares Web. [Citado; Disponible en: <http://posicionamientobuscadores.developers4Web.com>.
8. **Carrero, R.G.** (2006.) 10 razones para usar CSS. Volume.
9. **Navarra, U.d.** [Citado; Disponible en: <http://www.unav.es/innovacioneducativa>.
10. **Escudero, S.**, Diseñar Web con Photoshop.
11. <http://www.w3.org/DOM/> . 2000
12. **Hall, M.** Introduction to the Document Object Model. 1999-2002 [Citado; Disponible en: <http://kursor.net/traducciones/brainjar.es/introdom1.es.html> ; <http://www.brainjar.com/dhtml/intro>
13. **Pérez, J.E.**, Introducción a AJAX. 2007.

14. <http://www.pergaminovirtual.com.ar>. 1998-2007
15. **Cuba, E.d.A.E.** Ayuda al usuario del caso de estudio en el nuevo marco de trabajo del ERP v 1.0 Volume, .febrero 2009
16. **Company, Z.P.** Zend studio for eclipse.2009 [Citado; Disponible en: <http://www.zend.com>.
17. **Web, M.d.** Conociendo los principales editores de páginas web del mercado. 2007 [Citado; Disponible en <http://www.maestrosdelweb.com/>.
18. **Web, M.d.** Editores web que facilitan tu trabajo. 2008 [Citado; Disponible en: [www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo/](http://www.maestrosdelweb.com/editorial/editores-web-que-facilitan-tu-trabajo/).
19. **ULPGC, U.d.L.P.d.G.C.** Los cookies 2006 [Citado; Disponible en: <http://www.ulpgc.es/otros/tutoriales/JavaScript/cap9.html>.
20. **G., J.O.** Políticas de Privacidad 2009 [Citado; Disponible en: <http://www.fifabolivia.com/content/view/311/197/>.
21. **Guerrero, D.** Curso de Tecnologías Web 2007 [Citado; Disponible en: <http://www.danquer.com/cursos/TecnologiasWeb/4-1-Sesiones>.
22. **Torres, A.G.** Minería web y personalización: Revisión bibliográfica y propuesta de un marco de referencia. 2007 [Citado; Disponible en: <http://zarza.usal.es/~fgarcia/doctorado/iweb/05-07/Trabajos/MineriaWeb%20y%20Personalizacion.pdf>.
23. **Manuel., J.** El caché en internet. 1999 [Citado; Disponible en: [http://www.wikilearning.com/tutorial/los\\_rincones\\_del\\_api\\_win32\\_el\\_cache\\_de\\_winet](http://www.wikilearning.com/tutorial/los_rincones_del_api_win32_el_cache_de_winet)el\_cache\_en\_internet/3846-3 ; <http://www.lawebdejm.com>
24. **Cruz, A.R.H.Y.R.** (2008) LA USABILIDAD. UN ACERCAMIENTO A SU UTILIZACIÓN Y ACTUALIDAD EN LA UCI. Volume,
25. **Hernández, L.A.F.** Organización de los contenidos en los sitios Web: las taxonomías. 2007 [Citado; Disponible en: <http://scielo.sld.cu/pdf/aci/v15n5/aci13507.pdf>.



26. **Aballe, Y.S.F.D.H.**, “Propuesta de un proceso para realizar la Arquitectura de Información en los proyectos productivos de la Universidad de las Ciencias Informáticas”. junio 2008.
27. **Información., G.d.A.d.** (2008) LÍNEA BASE DE ARQUITECTURA DE INFORMACIÓN PARA LOS PRODUCTOS DESARROLLADOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS. Volume,
28. Definición de Estándares. 2006 [Citado; Disponible en: <http://dgplades.salud.gob.mx/2006/htdocs/hg/Nuevas/hmc6.pdf>.
29. Definición de ZONIFICACIÓN. [Citado; Disponible en: <http://www.definiciones.com.mx/definicion/Z/zonificacion/>.
30. **Adriana Valdespino González, Y.M.C.**, Guía práctica para realizar pruebas de concepto a la arquitectura de software de sistemas de gestión. . 2009.



**BIBLIOGRAFÍA**

1. Aballe, Y.S.F.D.H., 2008 "Propuesta de un proceso para realizar la AI". 2008.
2. AntonioVallecillo, L.F.J.M.T. Volume,
3. Carrero, R.G. (2006.) 10 razones para usar CSS. Volume,
4. Company, Z.P. 2009 Zend studio for eclipse. [Citado; Disponible en: <http://www.zend.com>.
5. Cruz, A.R.H.Y.R. LA USABILIDAD. UN ACERCAMIENTO A SU UTILIZACIÓN Y ACTUALIDAD EN LA UCI. Volume,
6. Cuba, E.d.A.E. Ayuda al usuario del caso de estudio en el nuevo marco de trabajo del ERP v 1.0  
Volume,
7. Escudero, S., Diseñar Web con Photoshop.
8. Gootzit, G.P.R.V.D. Portals Are the "Swiss Army Knives" of Enterprise Software. . 2006 [citado.
9. Group, P.S., Plumtree blossoms: New version fulfills enterprise portal requirements. 1999.
10. Hall, M. Introduction to the Document Object Model. [Citado; Disponible en: <http://www.brainjar.com/dhtml/intro/> ; <http://kursor.net/traducciones/brainjar.es/introdom1.es.html>.
11. Mendoza, J.A. 2000 [Citado; Disponible en: <http://www.divisait.com>.
12. Navarra, U.d. [Citado; Disponible en: <http://www.unav.es/innovacioneducativa>.
13. Pérez, J.E., Introducción a Css (Bibliografía básica en teleformación).
14. Pérez, J.E., Introducción a AJAX.
15. Polgar, J.P.R.B.A., Building and Managing Enterprise-Wide Portals. 2005.
16. Systems, B., Evolving Portal Solutions. 2006.

17. Torres, A.G.[Citado; Disponible en: <http://zarza.usal.es/~fgarcia/doctorado/iWeb/05-07/Trabajos/MineriaWeb%20y%20Personalizacion.pdf>.
18. ULPGC, U.d.L.P.d.G.C. 2006 [Citado; Disponible en: <http://www.ulpgc.es>.
19. Wood, R.W.L. Compresión de Datos y Criptografía. 2009 [Citado; Disponible en: <http://www.w3.org/DOM/>.
20. Zeldman, J. Diseño con estandares Web. [Citado; Disponible en: <http://posicionamientobuscadores.developers4Web.com>.
21. Información., G.d.A.d. (2008) LÍNEA BASE DE ARQUITECTURA DE INFORMACIÓN PARA LOS PRODUCTOS DESARROLLADOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS. Volume,

## GLOSARIO DE TÉRMINOS

1. **TELNET:** Conexión a un Host en la que el ordenador cliente emula un terminal de manera que se configura como terminal virtual del ordenador servidor.
2. **HTTP:** De sus siglas en inglés HyperText Transfer Protocol (Protocolo de transferencia de hipertexto) es el método más común de intercambio de información en la world wide Web, el método mediante el cual se transfieren las páginas Web a un ordenador.
3. **JSR 168:** Estandariza cómo los componentes de un portal deben ser desarrollados y permite la interoperabilidad de los portlets entre diferentes portales. La JSR-168 fue desarrollada por la Java Community Process (JCP), y tiene el respaldo de los principales proveedores de servidores de portal, como por ejemplo Apache, BEA, IBM, Oracle, SAP, Sun Microsystems, y muchos otros.
4. **PDA's:** (Personal Digital Assistant o Ayudante personal digital), dispositivo de pequeño tamaño que combina un ordenador, teléfono/fax, Internet y conexiones de red.
5. **SOA:** Service Oriented Architecture o Arquitectura Orientada a Servicios.
6. **ERP:** Planificación de Recursos Empresariales.
7. **COMPONENTE PORTAL:** Conjunto de interfaces contractualmente especificadas y explícitas sólo con dependencias dentro de un contexto, el cual posee una infraestructura, es decir funciona como una Plataforma que integra múltiples aplicaciones, basándose generalmente en datos o aplicaciones agrupados en Silos de Información o Islas de Servicios Aplicativos estos son interpretados o accedidos por Servicios Web para ser mostrados en entidades casi independientes del Browser dentro del Browser.
8. **Compresión LZ77:** Algoritmo de compresión sin pérdida basado en diccionario publicado en 1977 por Abraham Lempel y Jacob Ziv.
9. **Javascript:** Lenguaje de creación de secuencias de comandos en el lado del cliente que se utiliza en navegadores Web. En la actualidad se usa para que los programadores puedan

interactuar con páginas Web y con la propia ventana del navegador. Javascript se basa someramente en Java, un lenguaje de programación orientado a objetos que se utiliza en la Web a través de subprogramas incrustados.

10. **Modelo-Vista Controlador:** Patrón de diseño que separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

Modelo: Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

Vista: Maneja la visualización de la información.

Controlador: Controla el flujo entre la vista y el modelo (los datos).

11. **POST:** Método por defecto que utiliza HTTP que realiza las peticiones al servidor.

12. **IDE :**

13. **OpenBravo:** Reconocido desarrollador de soluciones profesionales en software libre para empresas, que ofrece la primera alternativa real al software propietario para empresas. Su sistema en entorno Web de gestión integral de empresas (ERP) y de gestión de punto de venta (PoS), los más populares en su área, han sido descargados más de un millón de veces y se utilizan en alrededor de 50 países.

14. **UCID:** Unidad de Compatibilización, Integración y Desarrollo de software para la defensa

15. **Iframe:** Marco o sub-ventana incrustada en línea con el contenido. Es un cuadrado cuyas dimensiones debe especificar el desarrollador en la propia etiqueta iframe, que tiene asociada una página Web que se carga en dicho espacio. Esa página Web tendrá sus propios contenidos y estilos, independientes del contexto donde se está mostrando. Además será perfectamente funcional: si tiene enlaces se mostrarán en ese mismo espacio y si tiene scripts

o aplicaciones dentro se ejecutarán también de manera autónoma en el espacio reservado al `iframe`.

16. **API:** (Application Programming Interface o Interfaz de Programación de Aplicaciones) es un conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Herramientas de programación para rutinas, protocolo y software.