# ****8 reglas de oro****

**1.- Luchar por la consistencia o coherencia:** Este punto está enfocado a las diferentes terminologías y estructuras de pantallas para situaciones o escenarios semejantes, estos deben utilizarse en anuncios, menús y pantallas de ayuda, esto enfocado a los nombres de botones, utilizar los mismos nombres para la misma acción, distribución de las pantallas en cuanto a contenidos.

**2.- Utilizar frecuentemente accesos directos:** Esto consiste en presentar botones o link de acceso rápido para acceder a las diferentes prestaciones que tiene el sistema, permitir al usuario utilizar estos accesos se fundamenta en que en la medida que el tiempo en que usa el software aumenta, también lo hace la necesidad de este por reducir los pasos o acciones para realizar acceder a cierto servicio, disminuyendo el tiempo de respuesta y aumentando el ritmo de la interacción. En este punto se pueden utilizar acrónimos y abreviaturas, teclas de función, comandos ocultos, entre otros.

**3.- Ofrece comentarios informativos:** Por cada operador de acción, debe haber algún sistema de retroalimentación. Para acciones frecuentes y de menor uso, la respuesta puede ser modesta, mientras que para los poco frecuentes y las principales acciones, tales como errores graves del sistema, que puedan producir perdidas de información, la respuesta debería ser más sustancial.

**4.- Diseño de diálogo para producir la clausura:** Este punto está enfocado a la retroalimentación informativa, que se produzca en la conclusión de un grupo de acciones, esto da a los usuarios la satisfacción de logro, una sensación de alivio, esto en parte indica al operador que debe continuar con las acciones, o que se debe preparar para realizarlas.

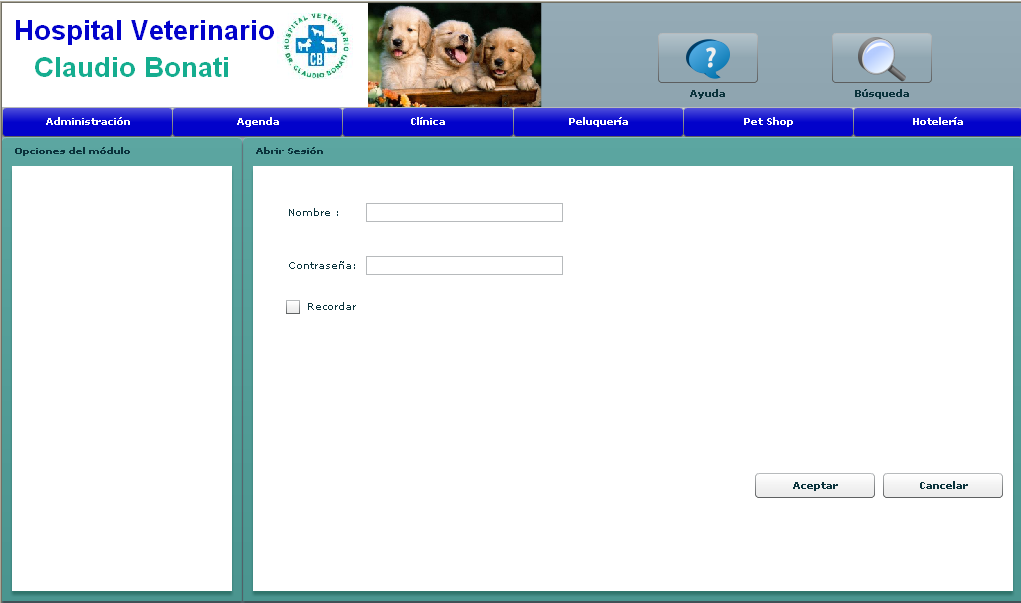
**5.- Ofrece una manipulación de errores simples:** En la medida de lo posible, diseñar el sistema para que el usuario no ocasione un grave error. Si aparece un error, el sistema debería ser capaz de detectar el error y ofrecer de manera sencilla y comprensible una manera para identificar el dicho error.

**6.- Permitir un fácil retroceso de las acciones:** Esta característica alivia la ansiedad, ya que el usuario sabe que los errores se pueden deshacer, esto lleva a que el usuario tenga menos miedo al explorar el software y las diferentes opciones desconocidas para el usuario. Las unidades de reversibilidad pueden ser una sola acción, una entrada de datos, o un grupo de acciones.

**7.- Apoyo interno a un enfoque de control total:** Los usuarios experimentados desean el sentido de que están a cargo del sistema y que el sistema responde a sus acciones. Para esto es necesario diseñar el sistema para que los usuarios inicien las acciones en lugar de que inicien las respuestas. Y que este pueda ingresar y salir de diversas opciones de manera fácil, sin pasar por procesos intermedios.

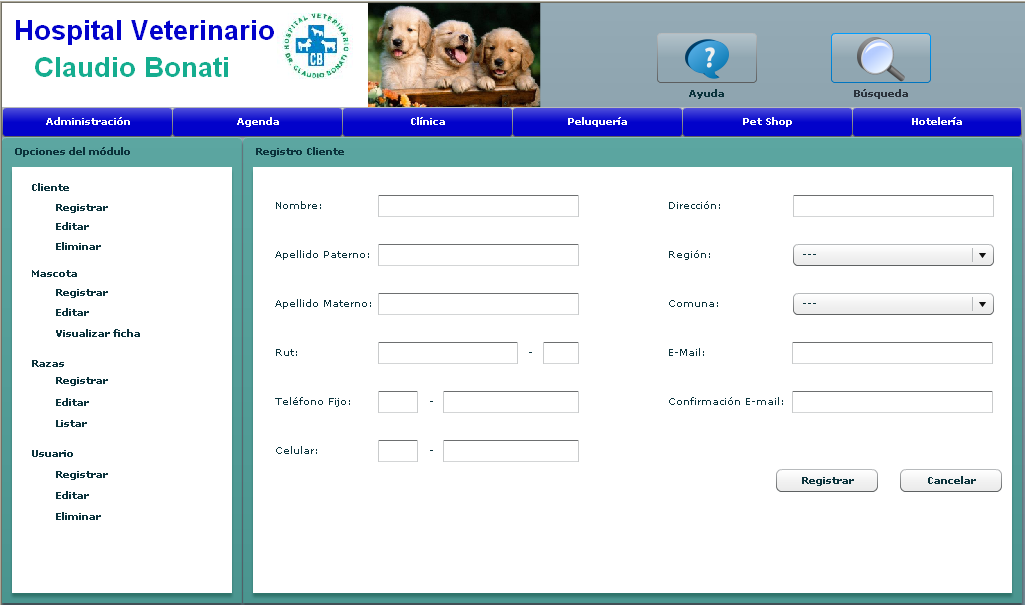
**8.- Reducir la carga de la memoria a corto plazo:** Este punto presenta vital importancia debido a las limitaciones de recursos humanos, en cuanto al procesamiento de la información en la memoria a corto plazo, esto exige que se muestren de manera sencilla, y que el software no ponga a prueba la memoria del usuario, si no que este le facilite al usuario el recordar la información, mediante un escenario de interacción.

# Estructura de la interfaz



**Imagen 1**

En la **imagen 1** se puede apreciar la interfaz que se desarrollo, la cual no necesariamente será la definitiva en cuanto a la forma de presentar los datos o las entradas de datos, más bien esto está enfocado a los espacios que ocupara la información que se presenta en este software, esto se puede apreciar claramente con los paneles, pudiendo visualizar claramente cómo estará dividido el programa o la interfaz que interactuara con el usuario.



**Imagen 2**

En la imagen 2 se puede observar cómo se debería estructurar el software, y la forma en que deberían presentarse los datos y los menús correspondientes a cada sub-modulo, esto al momento de seleccionar dicho modulo. Es por esto que se ha definido la mayor parte de la forma de estructurar la visualización de datos, el ingreso de datos, y en qué lugares deberán aparecer partes importantes de una interfaz como imágenes, botones, etc.

La forma de utilizar el software, explicado rápidamente, sería hacer login con la cuenta de usuario correspondiente. Luego el deberá seleccionar un modulo del sistema, que son los que se encuentran en la barra horizontal, después de esto aparecerá en el panel izquierdo Opciones del modulo, los diferentes servicios que este modulo presentara, se selecciona un servicio y en el panel de datos, aparecen los datos a ingresar o las diferentes visualizaciones. Y a continuación se seleccionara el botón correspondiente a la acción que se desea realizar del panel de Opciones de selección.

## Descripción de los componentes de la interfaz

En el software se definieron sectores que no deberían modificarse al momento de ingresar a cada sub-modulo, para lograr una consistencia en el software y evitar que al abrir cada sub-modulo aparezca una estructura diferente de los datos. Es por esto que se definió que solo deberán cambiar sectores internos de los paneles: Opciones del módulo, Datos, Opciones de selección.

## Opciones de módulo

****

**Imagen 3**

En este panel, se encontraran los servicios o procesos principales que presenta el modulo que fue seleccionado, los cuales si tienen sub-categorías, deberían aparecer indentados, como se muestra en la **imagen 3**, esto servirá de gran forma para retroceder rápidamente a secciones anteriores o servicios anteriores, ya que mientras se esté en el mismo modulo, estarán visibles las opciones anteriores y las de otros servicios relacionados con el modulo, por lo que si el usuario desea ingresar a otro servicio de este modulo, lo podrá hacer de manera rápida. Esto cubre en gran parte la regla número 2 de los anteriormente mencionados, el de presentar accesos directos. De igual forma se pueden definir teclas o combinaciones de teclas que permitan acceder rápidamente a cada servicio.

Para este panel el porte será definido por cada realizador, y a la vez deberá ajustar el panel de datos para que calcen en la pantalla y no queden espacios vacios.

Es importante destacar que la forma en que se mostraran los datos es la siguiente: El titulo del panel debe ser el del modulo al cual pertenece, y los servicios que se prestan se deben dividir, agrupándolos en sustantivos si se puede decir, por ejemplo:

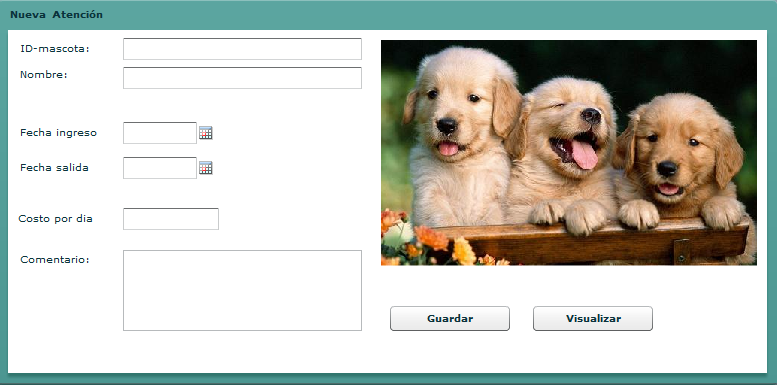


De esta forma las acciones deben ir indentados mas a la derecha de una “cosa” o sustantivo a la cual se le está aplicando la acción, de esta forma se entenderá fácilmente que se editara el cliente o que se registrara una mascota.

Ya que el profesor Narciso Cerpa menciono que debe ser coherente lo que va en este panel, en cuanto a que los paneles aparezcan en el mismo tipo de palabra, aquí en este ejemplo los paneles deberían aparecer al presionar los verbos que son editar o registrar, etc. Y a su vez los sub-servicios que aparecen indentados deberían ser todas acciones.

En este panel se cubre también el punto número 7 de las reglas de oro, ya que estas opciones aparezcan constantemente, permite al usuario ingresar y salir fácilmente de las opciones.

# Datos



**Imagen 4.1**

****

**Imagen 4.2**

En este panel aparecerán los diversos campos a visualizar o en los cuales se ingresaran datos, aquí deberían cambiar los datos que se entregan o ingresan en los diferentes módulos, pero se debe seguir un patrón especifico, en cuanto a la forma de presentar la forma de multi-selección, las imágenes, los botones, etc. El titulo de este panel debe ser el nombre de la acción o servicio que está prestando, en este caso se está realizando una Nueva Atención.

En el caso de presentar muchos datos a ingresar o mostrar, el panel se dividirá en dos partes iguales, en las presentando dos columnas de datos, para aprovechar de mejor forma el espacio.

En cuanto a los campos en los que se deba hacer una selección múltiple, esta deberá representarse con un combo box. A excepción de cuando se puedan seleccionar más de una opción, pues aquí se utilizaran otros componentes, que permitan esto, como CheckBox o RadioButton.

Si se presentan imágenes deberán aparecer en el lado derecho superior de éste panel, es decir deberá dividir el panel en dos, y utilizar la división derecha para visualizar la imagen.

Para los ingresos de datos comunes se utilizara el componente TexInput, que presenta Flex. Y para el ingreso de comentarios si es que se necesita, se utilizo TextArea. Mientras que para seleccionar una fecha se utilizo el DateField.

Como en este panel pueden existir diferentes botones, para visualizar o agregar datos, y que no son tan importantes para ir en el panel de selección, se creó un botón como componente, el cual será utilizado en el interior de este panel, para evitar que existan diferentes modelos y tamaños de botón, además se creó de un porte adecuado para una palabra bastante larga para evitar que después se pierda la palabra y aparezca “visuali…” por ejemplo. Y se definió que este botón no debe llevar imagen, solo llevaran imagen los del panel de opciones de selección. El botón creado se puede ver en la **imagen 4.2.** Y todas estas estructuras o patrones se pueden observar en la **imagen 4.** Al definir las diferentes estructuras que deben ir dentro de este panel, y la forma en que debe aparecer por pantalla los datos, se cubre claramente el punto 1 de las reglas de oro, el cual es el de luchar por la consistencia.

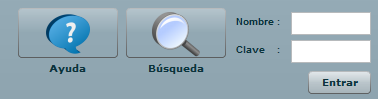
# Botones principales de acceso a los módulos



**Imagen 6**

Es donde se encuentra el acceso directo a los diferentes módulos, es desde aquí donde se accederá a los diversos módulos del sistema, además estos botones, o panel aparecen en todos los módulos en la misma posición, por lo que es una rápida forma de cambiar de un modulo a otro. Esto permite con una estructura más clara, permitiendo una consistencia en el sistema, al ingresar a cada modulo, punto 1 de las reglas de oro, y sirven a la vez de acceso directo, ya que al ingresar a un subsistema no deberá retroceder todo lo que ingreso, sino que bastara con presionar un botón, y podrá ingresar a otro subsistema, confirmando previamente la acción. Cubriendo el punto número 2 de las reglas de oro. También cumple el punto 7, permitiendo un apoyo interno al control total, ya que permite salir o ingresar a un sistema de forma fácil, sin procesos intermedios.

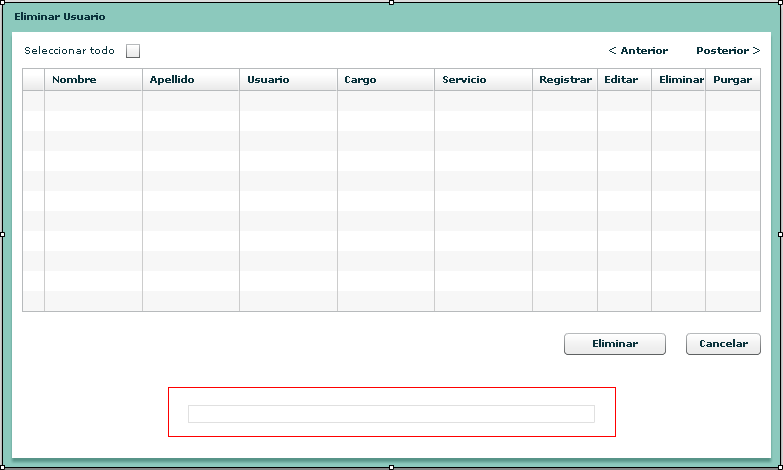
# Botones Rápidos



**Imagen 7**

En la **imagen 7** se muestran los diferentes botones que se encuentran en la parte superior derecha de las pantalla. Esto cubre los puntos 1, ya que siempre están en pantalla, permitiendo acceder fácilmente a ellos y siempre estarán en la misma posición aunque cambien de módulos. Se utiliza como acceso directo a algunos servicios, y permite salir fácilmente de otros, o acceder a esas opciones fácilmente cubriendo los puntos 2 y 7.

Son los que en cierta forma, presentan funcionalidades más generales del sistema, las cuales serán funciones de ayuda, búsqueda de datos importantes dentro del sistema, y el login en donde iniciara sesión cualquier miembro de la clínica veterinaria. Mensajes



**Imagen 8**

En la parte inferior de la componente de datos se alojaran los mensajes del sistema que serán los siguientes: los datos fueron guardado con éxito, modificado con éxito. Este sector del software cubre claramente las reglas numero 3 y 4 de las reglas de oro, ya que se presentan los resultados de operaciones al concluir, y además información relacionada con el proceso que se está llevando a cabo, como ayuda posiblemente.

Este mensaje tendrá el siguiente formato que se deberá seguir para que cada uno de los paneles posea una consistencia, los parámetros de este “Label” serán los siguientes:

* **Width: 407**
* **Height: 17**
* **X: 169**
* **Y: 372**

El mensaje deberá ir centrado dentro del “label” y los mensajes estándares dentro de este menú serán los siguientes:

**Ingreso El o La (nombre del registro) Con Éxito**

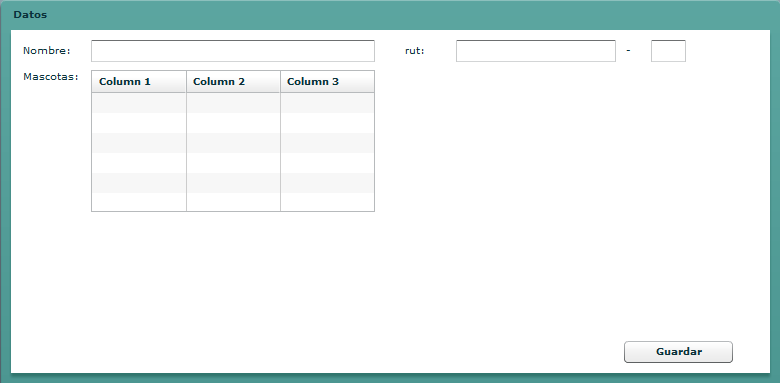
Por ejemplo:

Mensaje [0]: Ingreso El Cliente Con Éxito.

Mensaje [1]: Ingreso La Cita Con Éxito.

Mensaje [2]: Ingreso La Mascota Con Éxito.

Mensaje [3]: Ingreso El Usuario Con Éxito.



**Imagen 9**

En la imagen 9 se muestra como se representaran las relaciones 1 a 1, 1 a n y n a n. La forma que tendrán los botones y el lugar en donde preferentemente irán.

# Componentes

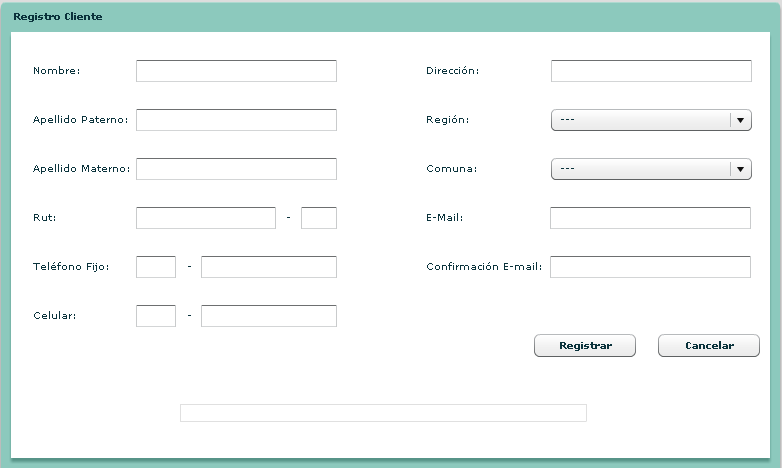
Cada una de las componentes tendrá los siguientes tamaños:

* Width: 779
* Height:383

El borde de cada una de las componentes deberá poseer los siguientes colores que son representativos para la clínica veterinaria:

Border color : #: 15ad8f

# Registro



Para la realización de los registros se deberán seguir el siguiente formato:

1. En la parte superior de la componente ira el nombre del registro expresado de la siguiente forma: “Registro (nombre del registro)” y el nombre deberá poseer tipo de letra “Verdana”, tamaño: “10” y el color a utilizar será: “# 0b333c”.
2. Todos los TextInput tendrán que tener un mismo tamaño el cual será de width: 201 e identados de igual forma y deberán tener una separación entre cada TextInput desde arriba hacia abajo de 49 pixeles.
3. En los TextInput cuando se requiera el ingreso solamente de un parámetro serán definidos a través de expresiones regulares que nos restringirá el acceso de valores no deseados, los que serán implementados de la siguiente forma:
   * Solamente String

<mx:TextInput restrict="A-Z a-z" x="125" y="28" width="201" id="nombre" maxChars="12"/>

**restrict="A-Z a-z":** Este será el código que deberá ser modificado para poder obtener cualquier expresión regular que se requiera por ingreso.

1. Cada uno de los TextInput deberán tener un máximo de carácter de ingreso, respetando el número de caracteres determinados en la base de datos, para tener una consistencia en los datos.

<mx:TextInput restrict="A-Z a-z" x="125" y="28" width="201" id="nombre" maxChars="12"/>

**maxChars=”12”:** Este será el código que se requerirá para poder darle un máximo de carácter a las entradas de teclado.

1. Cuando sea necesario obtener los apellidos de un usuario irán divididos de dos formas, primero será requerido el **apellido paterno** y luego el **apellido materno** del usuario.
2. Para el requerimiento del Rut se seguirá el siguiente formato, será dividido en dos TextInput donde en primer lugar se requerirá el Rut y en segundo lugar será pedido el código verificador del Rut, como serán mostrado en la siguiente imagen:



1. Para la petición de un teléfono fijo se deberá seguir el siguiente formato, será dividido en dos TextInput en donde en primer lugar deberá ir asignar el código de la ciudad y en segundo lugar el numero como será presentado en la siguiente imagen:



1. Para la petición de un celular se deberá seguir el siguiente formato, será dividido en dos TextInput en donde en primer lugar ira el cogido del celular y en segundo lugar ira el numero de celular, que será representado en la siguiente imagen:



1. Cada vez que se solicite el email de un usuario deberá existir una confirmación del email, que permitirá poder validar su ingreso, el email no podrá ser copiado del email anteriormente ingresado por el usuario.
2. Los botones a utilizar en esta componente serán los siguientes, que deberán ir alojados en la parte inferior de la componente:

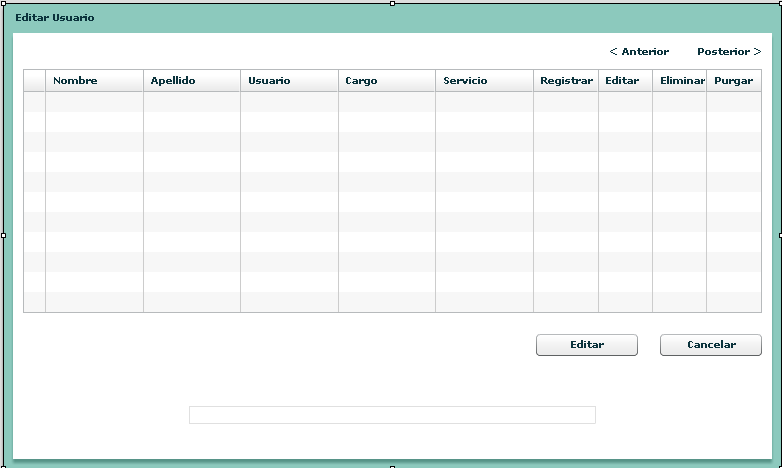
|  |  |  |
| --- | --- | --- |
| **Botones** | **Acciones** | **Formato** |
|  | Realizar el registro de los datos a la base de datos del sistema y luego volverá a la componente principal del servicio que se está utilizando. | Width: 102  Height:23  Corner Radius: 6  X: 523  Y: 302 |
|  | Vuelve a la componente principal del menú utilizado. | Width: 102  Height:23  Corner Radius: 6  X: 647  Y: 302 |

Al lado izquierdo ira alojado el “botón registrar” y el lado derecho ira el “botón cancelar”, los cuales llevaran una distancia de 22 pixeles.

## Editar

Poseerá dos pantallas para poder editar los datos del sistema, los cuales serán la siguiente:

1. En primer lugar se seleccionara el listado de datos que se desee editar, pero esta opción permitirá solamente seleccionar un solo listado no como el caso de la eliminación y purgación, a continuación será mostrada la componente a utilizar:



Para la realización del editar, los datos serán cargados automáticamente desde la base de datos una vez seleccionada la opción desde el menú del modulo, por lo cual se deberán seguir el siguiente formato:

1. En la parte superior de la componente ira el nombre del editar expresado de la siguiente forma: “Editar (nombre del editar)”. El nombre deberá poseer tipo de letra “Verdana”, tamaño: “10” y el color a utilizar será: “# 0b333c”.
2. Deberá llevar un “DataGrid” que le permitirá al usuario poder visualizar cada uno de los datos almacenados en la base de datos de un formulario seleccionado, en la primera columna de datagrid ira incorporado un checkbox que le permitirá al usuario poder seleccionar el dato que desea editar, como será mostrado a continuación:



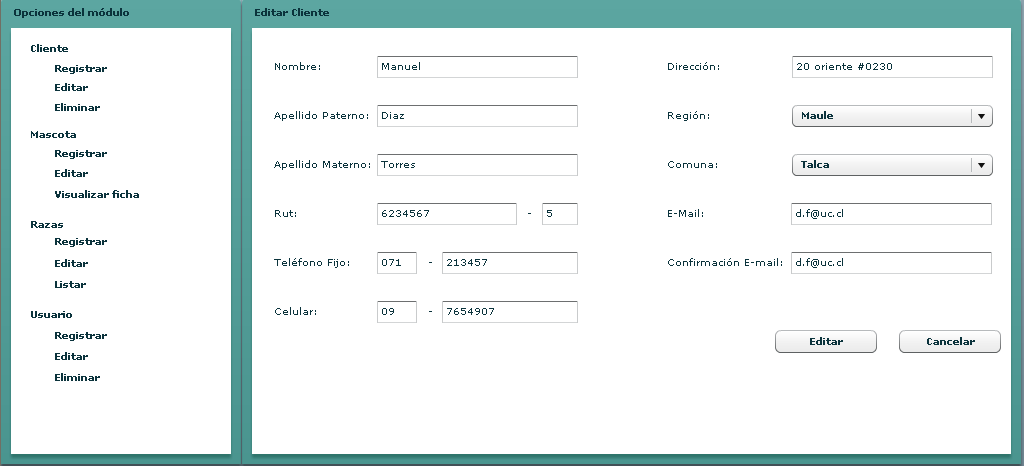
Al interior del CheckBox solamente serán mostrados 11 datos en una primera parte, porque esta componente tendrá opciones de avanzar o retroceder en búsqueda de un objetivo que se desee editar, cada una de esta acciones serán realizadas con los “LinkButton” llamados “Posterior” y “Anterior”.

1. Los botones a utilizar en esta componente serán los siguientes, que deberán ir alojados en la parte inferior de la componente:

|  |  |  |
| --- | --- | --- |
| **Botones** | **Acciones** | **Formato** |
|  | Al momento de presionar el botón el sistema arrojara un mensaje “pop up” que consultara por la acción de editar los datos y aparecerán las siguientes opciones: “Yes” cambiara de componente para editar los parámetros que se deseen y en caso contrario se volverá a la componente principal del menú. | Width: 102  Height:23  Corner Radius: 6  X: 523  Y: 302 |
|  | Vuelve a la componente principal del menú utilizado. | Width: 102  Height:23  Corner Radius: 6  X: 647  Y: 302 |

Al lado izquierdo ira alojado el “botón editar” y el lado derecho ira el “botón cancelar”, los cuales llevaran una distancia de 22 pixeles.

1. En segundo lugar se mostrara un formulario con el listado seleccionado en la etapa anterior en donde se podrán modificar cada uno de los parámetros que se deseen modificar y que puedan ser modificables por el usuario:



Para la realización del editar, los datos serán cargados automáticamente desde la base de datos en el formulario del modulo que se desea modificar, por lo cual se deberán seguir el siguiente formato:

1. En la parte superior de la componente ira el nombre del editar expresado de la siguiente forma: “Editar (nombre del editar)”. El nombre deberá poseer tipo de letra “Verdana”, tamaño: “10” y el color a utilizar será: “# 0b333c”.
2. Llevara el mismo formulario de la etapa de registro, que serán formados por TextInput que tendrán cargado cada uno de los datos obtenidos de la base de datos, en donde se podrán modificar algunos datos como por ejemplo:
   * + Teléfono Fijo (Cliente).
     + Fecha (Cita).
     + E-mail(Cliente).
   * No se pueden modificar.
     + Rut(Cliente).
     + Raza(Mascota).
3. Los botones a utilizar en esta componente serán los siguientes, que deberán ir alojados en la parte inferior de la componente:

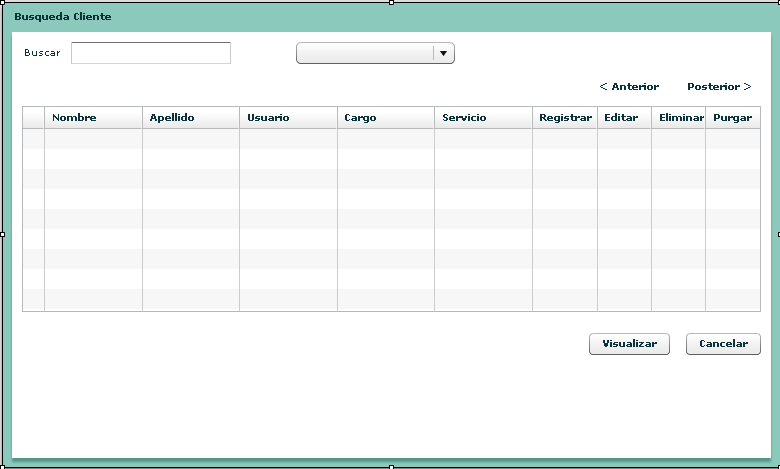
|  |  |  |
| --- | --- | --- |
| **Botones** | **Acciones** | **Formato** |
|  | Al momento de presionar el botón el sistema arrojara un mensaje “pop up” que consultara por la acción de editar los datos y aparecerán las siguientes opciones: “Yes” editara cada uno de los parámetros modificados pro el usuario al interior de un formulario y una vez efectuada esta operación volverá al menú principal del servicio, y en caso contrario se volverá a la componente principal del menú. | Width: 102  Height:23  Corner Radius: 6  X: 523  Y: 302 |
|  | Vuelve a la componente principal del menú utilizado. | Width: 102  Height:23  Corner Radius: 6  X: 647  Y: 302 |

Al lado izquierdo ira alojado el “botón editar” y el lado derecho ira el “botón cancelar”, los cuales llevaran una distancia de 22 pixeles.

## Búsqueda

Poseerá dos tipos de pantallas que le permitirá al usuario poder visualizar los datos buscados, que serán explicados a continuación:

1. En primer lugar nos encontramos con un panel que nos permitirá buscar un registro al interior de la base de datos como será mostrado a continuación:



Para la realización de una búsqueda, primero que nada el usuario deberá seleccionar el tipo de dato que desea buscar al interior del “ComboBox” y una vez seleccionada la opción, el usuario podrá ir escribiendo al interior de la TextInput donde serán cargados automáticamente desde la base de datos a medida que el usuario va escribiendo en el textInput, por lo cual se deberán seguir el siguiente formato:

1. En la parte superior de la componente ira el nombre de la búsqueda expresado de la siguiente forma: “Búsqueda (nombre de la Búsqueda)”. El nombre deberá poseer tipo de letra “Verdana”, tamaño: “10” y el color a utilizar será: “# 0b333c”.
2. En la parte superior izquierda ira un textInput en donde se introducirán los datos que el usuario desee buscar, este tendrá el siguiente formato:

* Width: 169 pixeles.
* Height: 23 pixeles.

1. Al lado del textInput ira situado un comboBox que tendrán una separación de 56 pixeles, en este comboBox irán cada uno de los parámetros que se desea buscar en el modulo seleccionado y tendrá las siguientes medidas:
   * Width: 169 pixeles.
   * Height: 23 pixeles.
2. Deberá llevar un “DataGrid” que le permitirá al usuario poder visualizar cada uno de los datos encontrados en la base de datos con los parámetros introducidos por el usuario, en la primera columna de dataGrid ira incorporado un checkbox que le permitirá al usuario poder seleccionar el dato que desea visualizar posteriormente, como será mostrado a continuación:



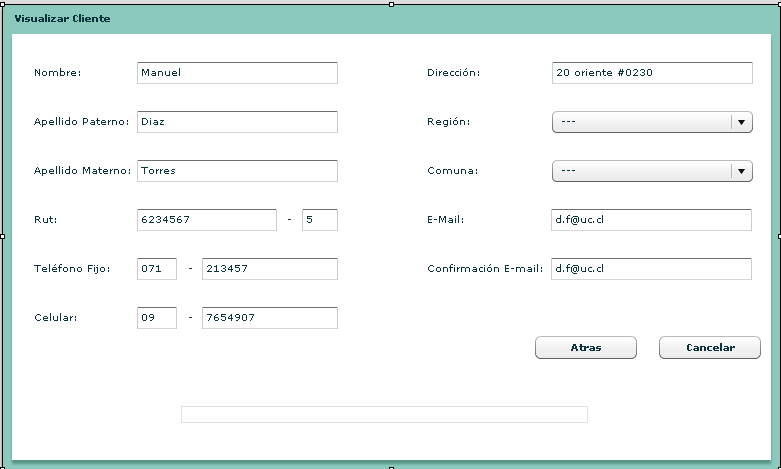
Al interior del CheckBox solamente serán mostrados 9 datos en una primera parte, porque esta componente tendrá opciones de avanzar o retroceder en búsqueda de un objetivo que se desee visualizar, cada una de esta acciones serán realizadas con los “LinkButton” llamados “Posterior” y “Anterior”.

1. Los botones a utilizar en esta componente serán los siguientes, que deberán ir alojados en la parte inferior de la componente:

|  |  |  |
| --- | --- | --- |
| **Botones** | **Acciones** | **Formato** |
|  | Al momento de presionar el botón el sistema cambiara automáticamente de pantalla para que el usuario pueda visualizar más detalladamente los datos buscados. | Width: 102  Height:23  Corner Radius: 6  X: 523  Y: 302 |
|  | Vuelve a la componente principal del menú utilizado. | Width: 102  Height:23  Corner Radius: 6  X: 647  Y: 302 |

Al lado izquierdo ira alojado el “botón visualizar” y el lado derecho ira el “botón cancelar”, los cuales llevaran una distancia de 22 pixeles.

1. Este componente nos permitirá poder visualizar todos los datos del listado anteriormente seleccionado, que será el siguiente:



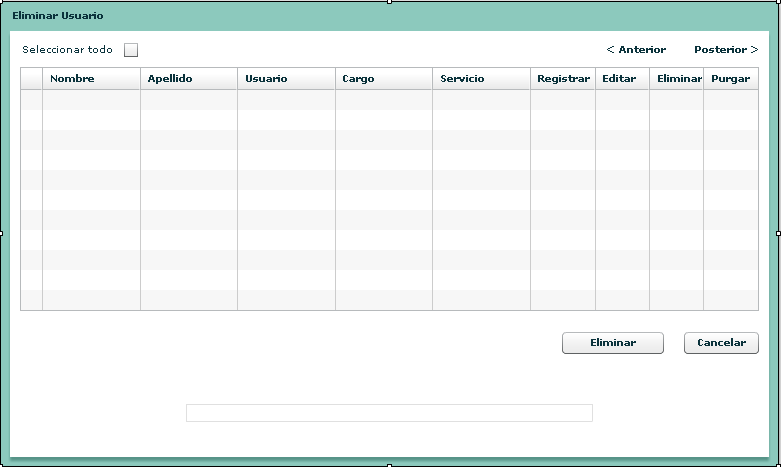
Aquí solamente nos permitirá poder visualizar los datos del listado seleccionado anteriormente sin poder realizar ningún tipo de modificación en los datos, el formato será descrito a continuación:

1. En la parte superior de la componente ira el nombre de la visualización expresado de la siguiente forma: “Visualizar (nombre del Visualizar)”. El nombre deberá poseer tipo de letra “Verdana”, tamaño: “10” y el color a utilizar será: “# 0b333c”.
2. Cada uno de los textInput serán cargado desde la base de datos, pero no podrán ser modificado por el usuario al momento de adquirir algún tipo de dato al interior de la componente.
3. Los botones a utilizar en esta componente serán los siguientes, que deberán ir alojados en la parte inferior de la componente:

|  |  |  |
| --- | --- | --- |
| **Botones** | **Acciones** | **Formato** |
|  | Al momento de presionar el botón el sistema cambiara automáticamente de a la pantalla anteriormente vista para tener la opción de visualizar un nuevo listado. | Width: 102  Height:23  Corner Radius: 6  X: 523  Y: 302 |
|  | Vuelve a la componente principal del menú utilizado. | Width: 102  Height:23  Corner Radius: 6  X: 647  Y: 302 |

Al lado izquierdo ira alojado el “botón atras” y el lado derecho ira el “botón cancelar”, los cuales llevaran una distancia de 22 pixeles.

## Eliminar o Pugar



Para la realización de la eliminación o purgación, los datos serán cargados automáticamente una vez seleccionada la opción desde el menú del modulo, por lo cual se deberán seguir el siguiente formato:

1. En la parte superior de la componente ira el nombre de la eliminación o purgación expresado de la siguiente forma: “Eliminar (nombre de eliminación)” y “Pugar (nombre de purgación)”. El nombre deberá poseer tipo de letra “Verdana”, tamaño: “10” y el color a utilizar será: “# 0b333c”.
2. En la parte superior izquierda de la componente llevara un “CheckBox”, en la cual deberá ir el modo de seleccionar todos los datos, que serán posibles de eliminar o purgar de la base de datos del sistema, se aplicara de la siguiente forma:



1. Deberá llevar un “DataGrid” que le permitirá al usuario poder visualizar cada uno de los datos almacenados en la base de datos de un formulario seleccionado, en la primera columna de datagrid ira incorporado un checkbox que le permitirá al usuario poder seleccionar el dato que desea eliminar, como será mostrado a continuación:



Al interior del CheckBox solamente serán mostrados 11 datos en una primera parte, porque esta componente tendrá opciones de avanzar o retroceder en búsqueda de un objetivo que se desee eliminar o purgar, cada una de esta acciones serán realizadas con los “LinkButton” llamados “Posterior” y “Anterior”.

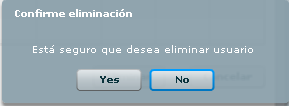
1. Los botones a utilizar en esta componente serán los siguientes, que deberán ir alojados en la parte inferior de la componente:

|  |  |  |
| --- | --- | --- |
| **Botones** | **Acciones** | **Formato** |
|  | Al momento de presionar el botón el sistema arrojara un mensaje “pop up” que consultara por la eliminación de los datos y aparecerán las siguientes opciones: “Yes” eliminara los datos seleccionados y volverá al menú del modulo utilizado y en caso contrario se volverá a la componente principal del menú. | Width: 102  Height:23  Corner Radius: 6  X: 523  Y: 302 |
|  | Al momento de presionar el botón el sistema arrojara un mensaje “pop up” que consultara por la purgación de los datos y aparecerán las siguientes opciones: “Yes” purgara los datos seleccionados y volverá al menú del modulo utilizado y en caso contrario se volverá a la componente principal del menú. | Width: 102  Height:23  Corner Radius: 6  X: 523  Y: 302 |
|  | Vuelve a la componente principal del menú utilizado. | Width: 102  Height:23  Corner Radius: 6  X: 647  Y: 302 |

Al lado izquierdo ira alojado el “botón eliminar” y el lado derecho ira el “botón cancelar”, los cuales llevaran una distancia de 22 pixeles.

## Advertencia Del Sistema

A continuación se mostrara un ejemplo de los mensajes de las advertencias del sistema producidos al interior del sistema:



Formato del mensaje de advertencia:

Esta seguro que desea eliminar (nombre registro)

O

Esta seguro que desea editar (nombre registro)

Ejemplos de tipo de advertencias:

Advertencia [0]: Esta seguro que desea eliminar la mascota.

Advertencia [1]: Esta seguro que desea editar el usuario.

Advertencia [2]: Esta seguro que desea purgar la ficha.

Advertencia [3]: Esta seguro que desea eliminar la cita

Advertencia [4]: Esta seguro que desea editar el usuario.

Advertencia [5]: Esta seguro que desea purgar el cliente.

Este tipo de pop up llevara dos tipos de opciones que deberá seleccionar el usuario, que serán las siguientes:

1. “Yes”: Eliminara, Pugara o editara los datos que fueron seleccionados por el usuario y volverá al menú principal utilizado por el modulo.
2. “NO”: volverá al menú principal utilizado por el modulo seleccionado.

## Validación textInput

Para la validación de un textInput, flex nos entrega múltiples herramientas que nos permitirá poder avisarle al usuario de posibles errores que ocurran al interior del sistema, sin la necesidad de codificar mucho al interior del programa flex, la visualización de esta herramienta es observada de la siguiente manera:





Para poder realizar esta aplicación al interior de flex se deberá incorporar el siguiente código:

## Validación String

<mx:StringValidator id="email2Validator" requiredFieldError="Ingrese La Confirmación De E-Mail" source="{email2}" property="text" />

Llevara los siguientes parámetros que serán importantes configurar para el buen funcionamiento de la funcionalidad:

**Id:** Será el identificador del StringValidator.

**requiredFieldError:** Aquí ira el mensaje que será mostrado al usuario de acuerdo a la información faltante al interior del TextInput.

**Source:** Llevara el id del TextInput que desea que se le incorpore el mensaje anteriormente incorporado a este.

Al momento de utilizar este tipo de validor se nos permite poder marcar el marco del TextInput con color rojo para orientar al usuario donde se encuentra el error dentro de la componente.

## Validación E-Mail

**<mx:EmailValidator id="emailValidator" requiredFieldError="Complete su email" source="{email}" property="text" invalidCharError="Caracteres inválidos" invalidDomainError= "El dominio no es válido"**

**missingAtSignError="Necesita una @ en su email" missingPeriodInDomainError= "Necesita un nombre de dominio" />**

Esta validación llevara los siguientes parámetros, para la realización de una configuración correcta del validador:

**Id:** Será el identificador del StringValidator.

**requiredFieldError:** Aquí ira el mensaje que será mostrado al usuario de acuerdo a la información faltante al interior del TextInput.

**Source:** Llevara el id del TextInput que desea que se le incorpore el mensaje anteriormente incorporado a este.

**missingAtSignError:** Nos permite corroborar que al interior del TextInput exista la “@”, para seguir el formato de un email, a continuación será mostrado como se ve reflejado este mensaje:



**missingPeriodInDomainError:** No permitirá poder corroborar que al interior del TextInput exista el dominio de un email como por ejemplo “.cl o com”, que serán mostrado a continuación:

