|  |
| --- |
| **Agenda Telefónica On line** |
| Proyecto web |
|  |
| Página web donde se gestionarán los números de teléfono del usuario |
|  |
| **Saray Carralero** |
| **17/03/2014** |
|  |

***Análisis funcional***

El proyecto web va a ser una página donde se podrá:

* Insertar
* Modificar
* Eliminar
* Ver todos los números de teléfono.

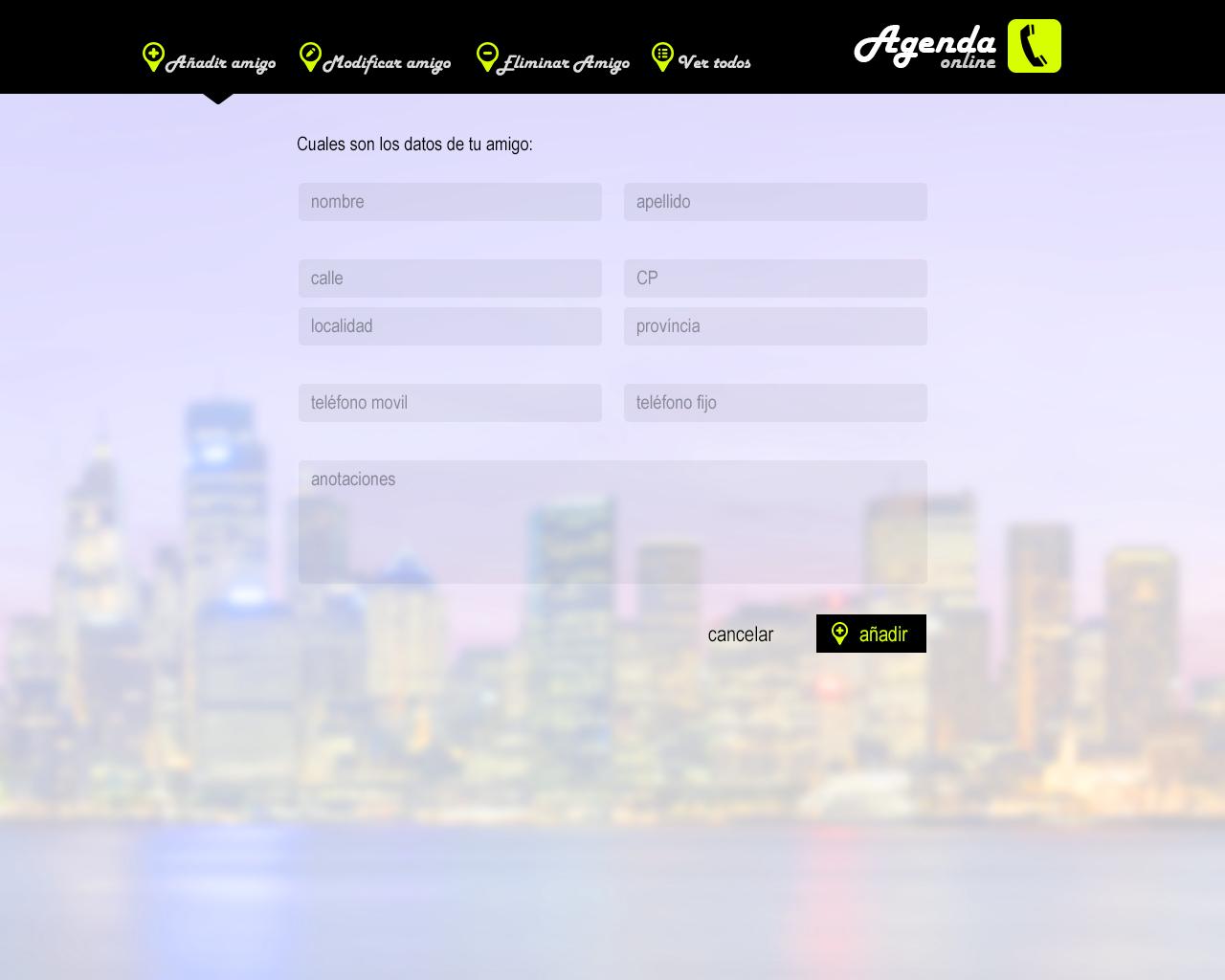
Todo desarrollado a base de código Java en entorno servidor. HTML5, CSS3 en cliente y BBDD, en este caso mySql

* La *PÁGINA PRINCIPAL* será así:



Un index intuitivo donde se reflejan 4 opciones generales: añadir amigo, modificar amigo, eliminar amigo y ver todos los amigos. El diseño está desarrollado con estilos CSS.

* Si pulsamos *AÑADIR AMIGO*:



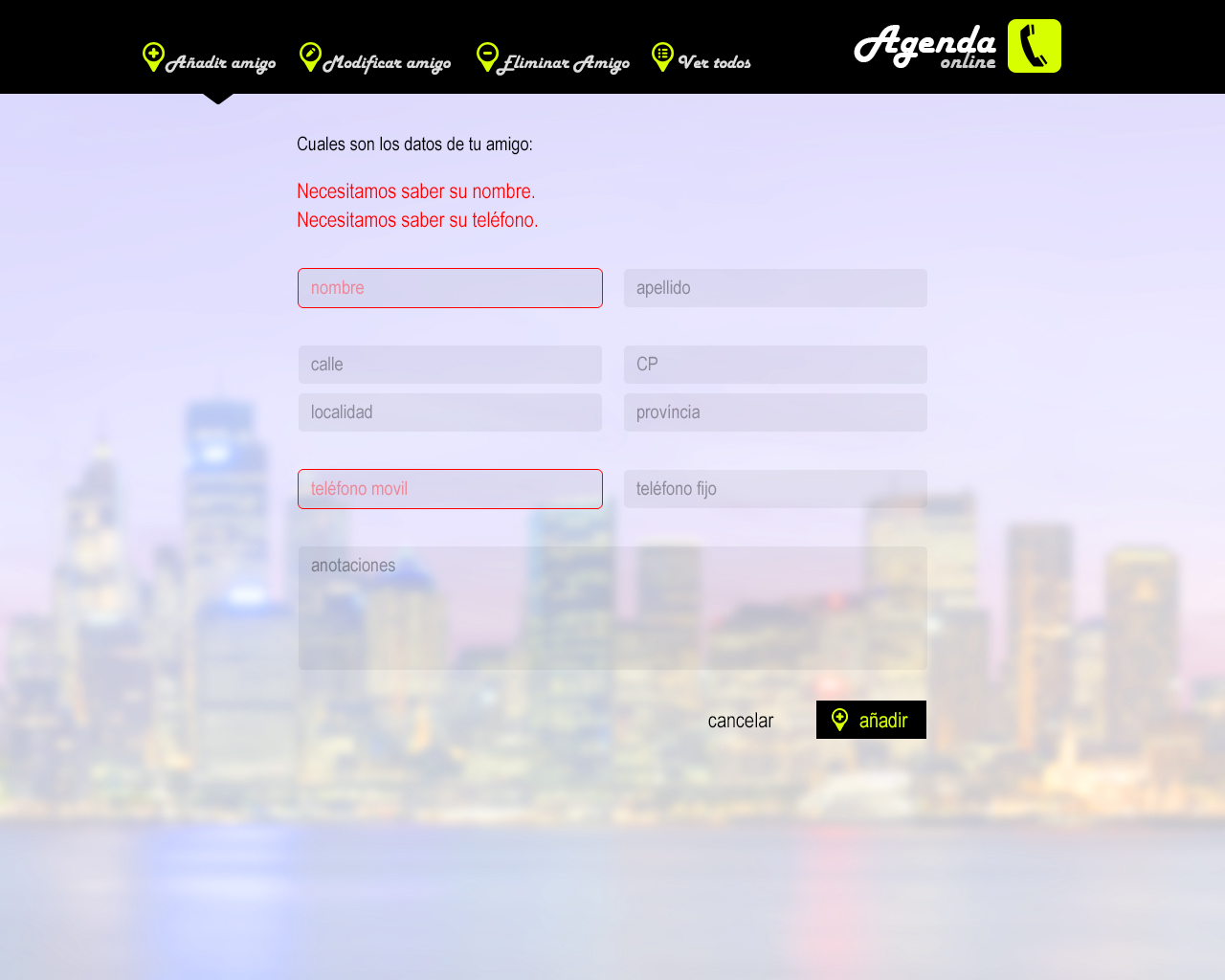
Se visualiza una barra de herramientas con las opciones generales y un formulario con unas cajas de texto creadas en HTML (input) para insertar los datos del nuevo amigo y un textarea con anotaciones de forma opcional.

Los datos del amigo:

* Nombre
* Apellido
* Calle
* Cp.
* Localidad
* Provincia
* Teléfono móvil
* Teléfono fijo
* Anotaciones

El formulario consta de dos botones Cancelar y Añadir. El primero cancela el formulario y te redirige a la página principal, el segundo en cambio recoge los datos insertados, se procesan mediante una clase DAO que conecta con la base de datos mediante una factoria

Una vez pulsado el botón añadir si falta algún campo, aparecerá un mensaje de error indicando dicho problema que se visualizará rodeando el campo que no se haya escrito en rojo.

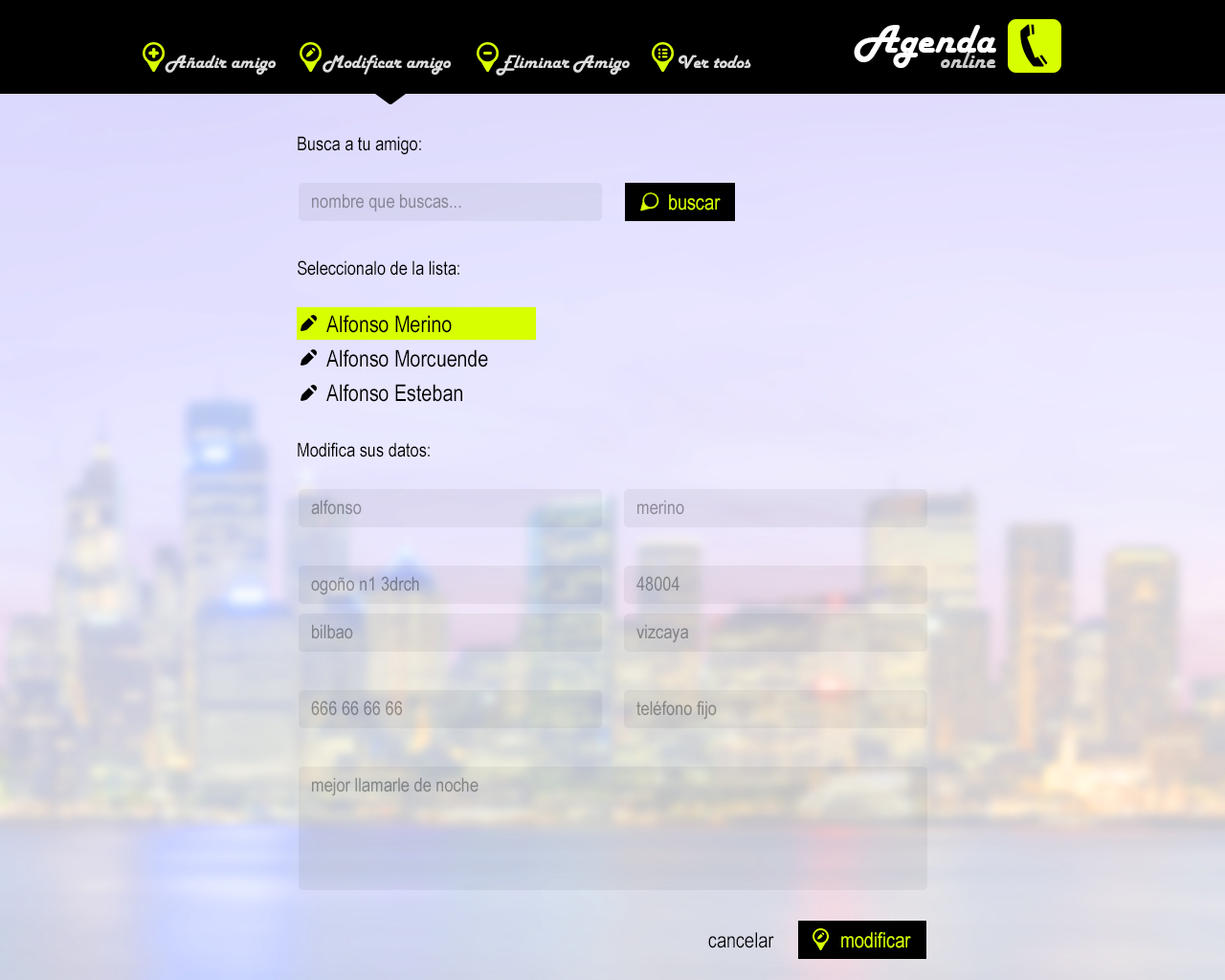


Por lo contrario, si los datos son correctos aparecerá una pantalla informando que el amigo ha sido insertado y con un formulario para pedirle al usuario que desea hacer:

* Añadir
* Modificar
* Eliminar



* Si pulsamos *MODIFICAR AMIGO*:

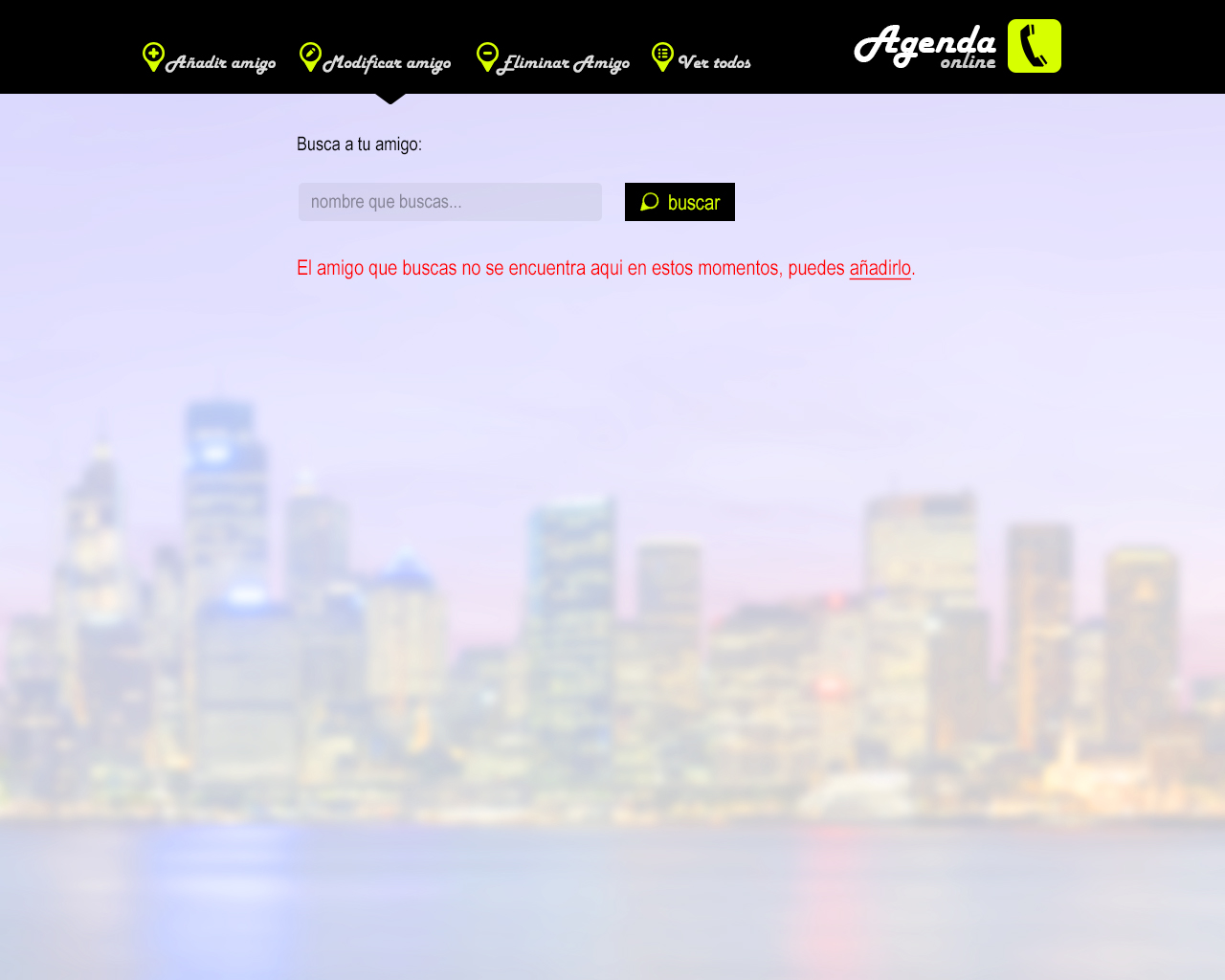


Se muestra una pantalla donde hay una barra de buscador dinámico con un botón. En la barra se escribe el nombre del amigo que deseas buscar y al pulsar el botón de Buscar se mostraran los nombres cuyas letras coincidan con las escritas en la barra. Al seleccionar el nombre deseado se rellenaran los input con el detalle del amigo. Una vez modificado el campo deseado se pulsa el botón *modificar,* donde mostrará una pantalla de confirmación informando que el modificado del amigo ha sido correcto y 3 opciones:

* Añadir
* Modificar
* Eliminar

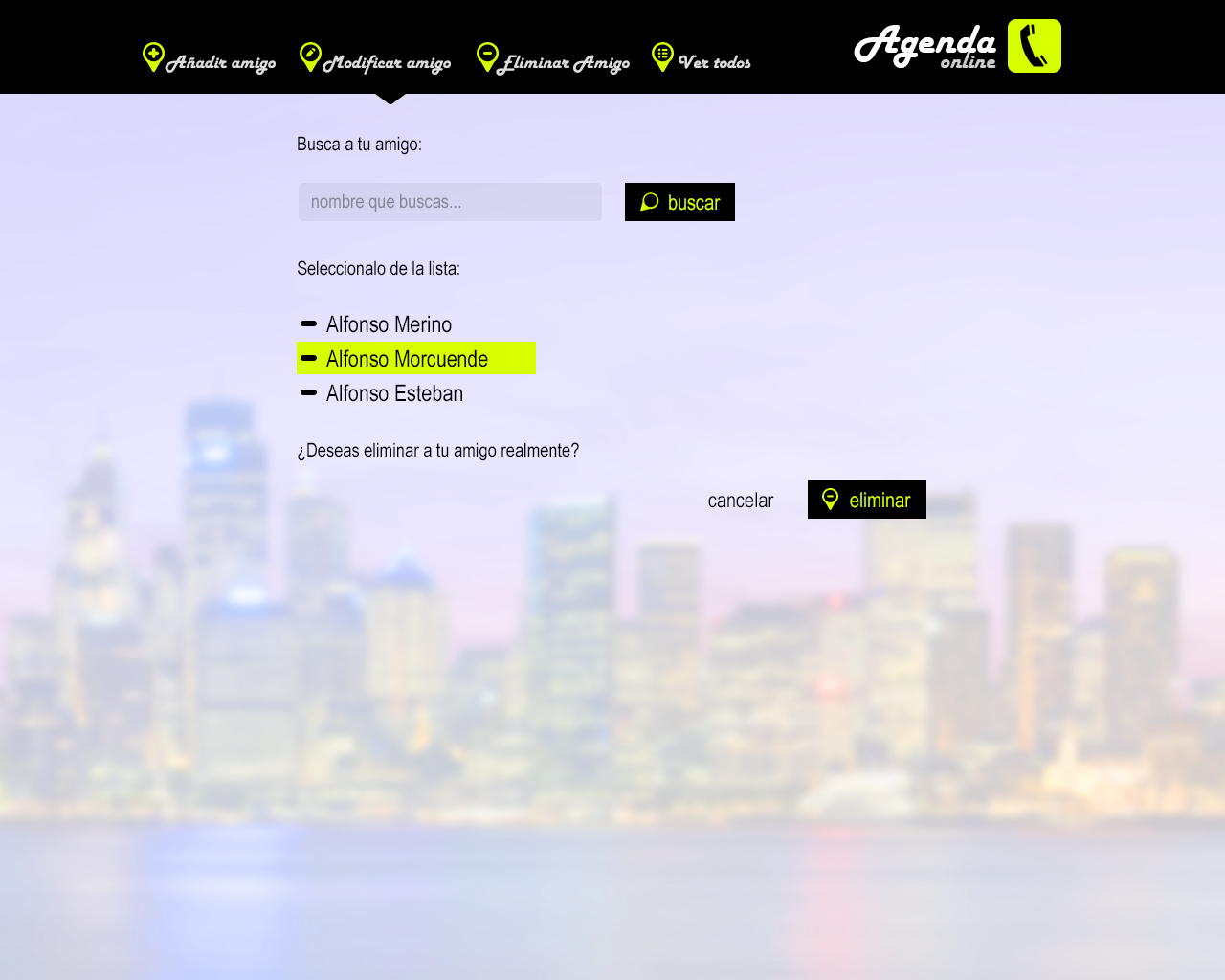


Si a la hora de buscar el amigo no se encuentra en la bbdd sacaremos al usuario un mensaje advirtiendo de lo que ocurre y que puede añadirlo si lo desea. La palabra añadir re direccionará a la pagina *Añadir* del proyecto.



* Si pulsamos *ELIMINAR AMIGO*:

Se busca al amigo de la misma forma que se hace cuando modificamos pero la diferencia es que si se pulsa eliminar se mostrará un mensaje de si realmente quiere eliminar al amigo.



Si se elimina saldrá un mensaje de confirmación informando del cambio y 3 opciones:

* Añadir
* Modificar
* Eliminar



Si no se encuentra al amigo que desea eliminar sacará un mensaje mostrando el error de búsqueda y dando opción a poder añadir en la bbdd a dicho amigo. El proceso es parecido al de modificar.



* Por último si se pulsa *VER TODOS*:

Se mostrará una tabla con **todos** los amigos, mostrando también el detalle de cada uno.

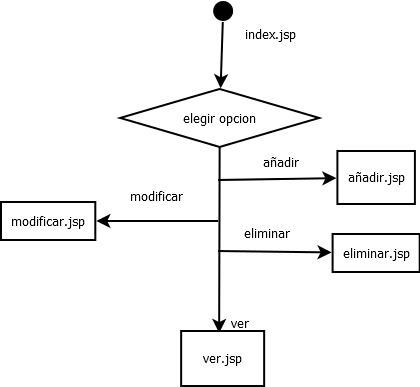


De esta forma es como desarrollaremos el proyecto, de forma dinámica, fácil e intuitiva con las opciones básicas de gestión y con diseño moderno.

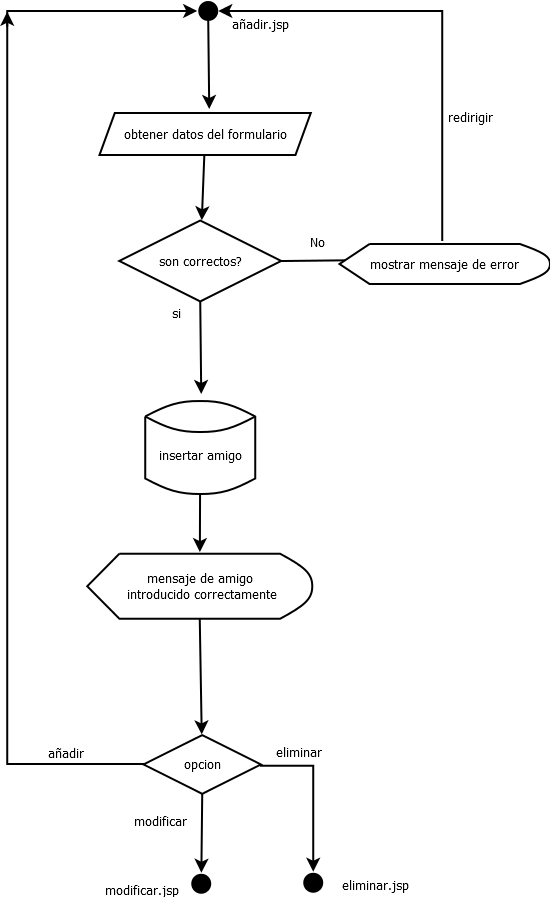
NOTA: El botón cancelar de todas las pantallas redirigen siempre a la home.

***Diagrama de flujo***

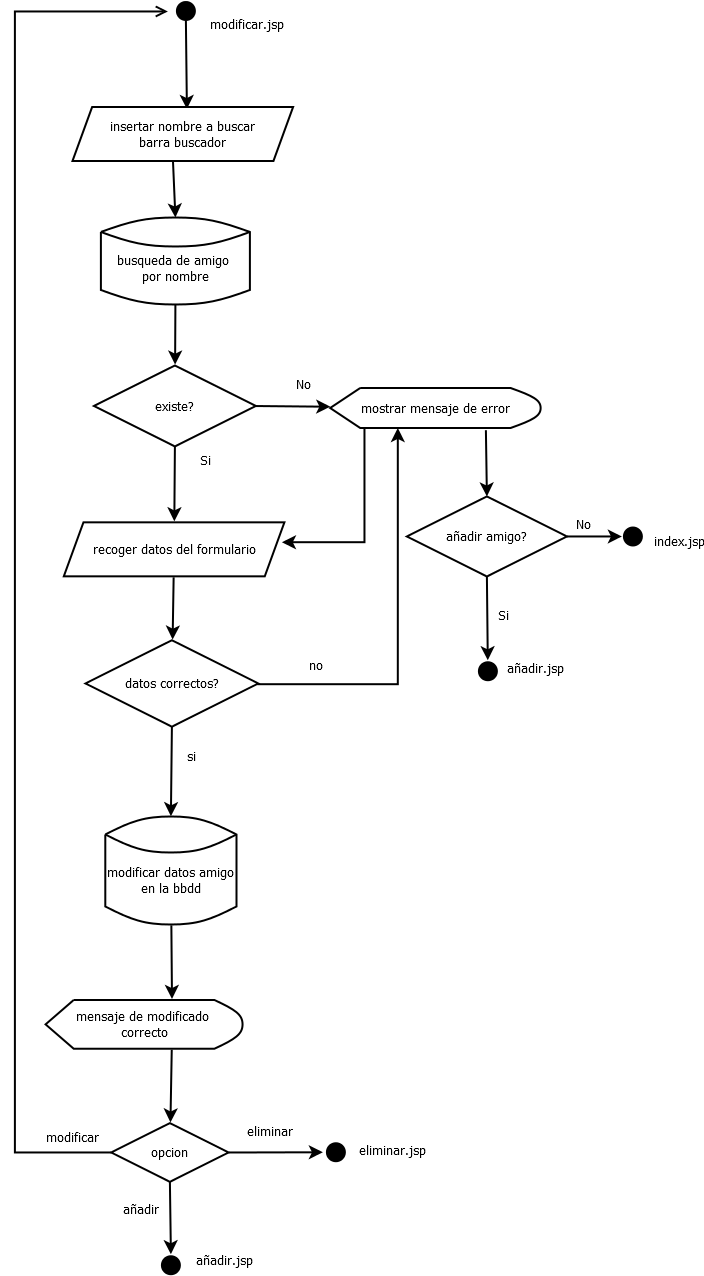
Index:



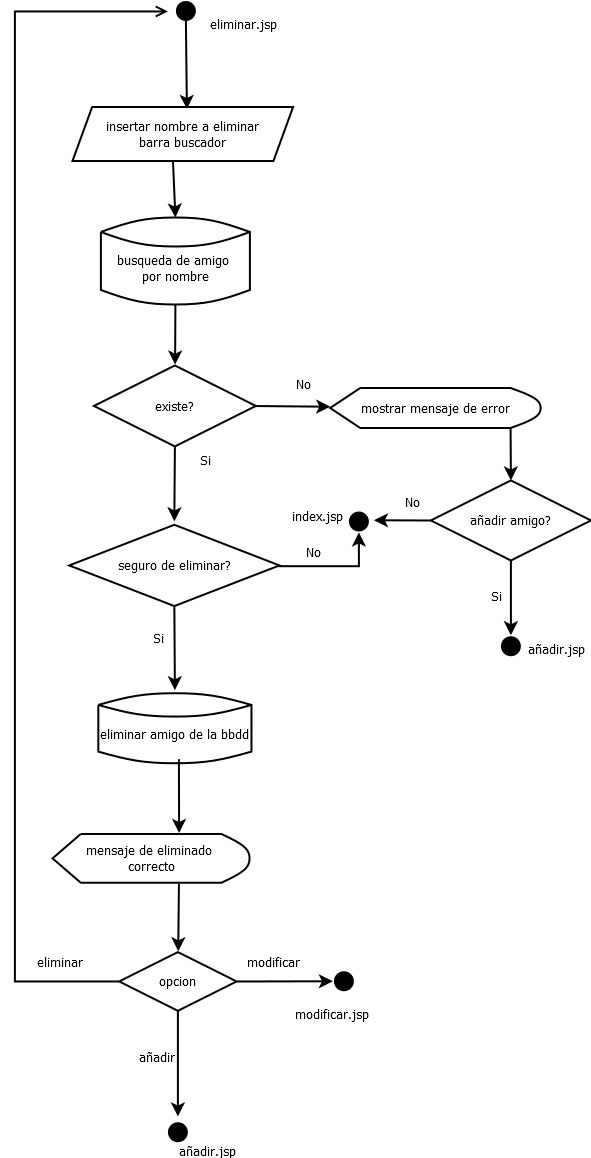
Añadir:



Modificar:



Eliminar:



Ver:



***Clases y Jsp`s necesarias***

Para empezar se necesitan 6 jsp:

* + Index.jsp
    - Pantalla inicial (HOME) de la aplicación.
  + Anadir.jsp
    - Pantalla para añadir los contactos.
  + Modificar.jsp
    - Pantalla para modificar los datos de un contacto existente.
  + Eliminar.jsp
    - Pantalla para eliminar un contacto de la agenda.
  + Ver.jsp
    - Pantalla que nos muestra la lista de contactos de la agenda.
  + todoOk.jsp
    - Pantalla que se muestra una vez hayamos añadido, modificado o eliminado un contacto.

Para cada funcionalidad que tenemos dentro de estas jsp, tendremos otra jsp:

* head.jsp
  + La cabecera de la página, con las configuraciones (estilos…), se incluye en el
* header.jsp
  + La cabecera de la página, junto con el logo de la página y el menú de navegación, se incluye en el index.jsp
* footer.jsp
  + Es el pie de página que incluimos en el index.jsp para cerrar el /body y el /html.
  + navBar.jsp
    - Es el menú de navegación de la página, se inserta en el index.jsp. Según en qué punto de la aplicación estemos será diferente. En el home se muestra con unos iconos grandes en el centro de la pantalla y en el resto de ventanas se muestra arriba con botones pequeños.
  + buscador.jsp
  + Es el diseño para hacer la búsqueda del nombre de un contacto para luego poder modificar o eliminarlo. Se incluye en eliminar.jsp y modificar.jsp.
* anadir.jsp
  + Es el diseño de las cajas de texto o campos que tenemos que rellenar para poder añadir un contacto, y también de los botones.
* eliminar.jsp
  + Es el diseño de las cajas de texto y botones
* modificar.jsp
* Es el diseño de las cajas de texto o campos que tenemos que rellenar para poder modificar un contacto, y también de los botones

Las clases que se necesitaran en esta aplicación son las siguientes:

* MainServlet: para poder navegar por la aplicación.
* ConnectionFactory: para conectarse a la base de datos.
* Amigo: Clase con los atributos necesarios del contacto.
* IAmigable: Interfaz para declarar los métodos que vamos a necesitar para trabajar con los contactos.
* ModeloAmigo: métodos que utilizará la clase DAOAmigo para las consultas.
* AgendaServlet: Para poder realizar operaciones del AppWeb.
* DAOAmigo: las operaciones a realizar con la base de datos.(CRUD).