AGENDA TELEFONICA



EDUARDO MONTERRUBIO ROBLEDO

16077019L

**INDICE**

[1.Análisis funcional](#h.ceyckx2guz3e)

[1.1 Insertar Amigo](#h.3u3b7fgnfqlh)

[1.2 Modificar Amigo](#h.x2pvd83ichpr)

[1.3 Borrar Amigo](#h.2u0wbjkj99r3)

[1.4 Listado Amigos](#h.y98x6j2ef7lm)

[1.5 Navegación](#h.2oashscihasc)

[2. MVC (Modelo Vista Controlador)](#h.bxn8cbk24xxj)

[2.1 Modelo](#h.kwdwp0bm7li4)

[2.2 Vista](#h.2p0d0s615s2)

[2.3 Controlador](#h.o2fbojmkntdw)

[3. Diagramas de Flujo](#h.ak64e5o8xabo)

[3.1 Insertar Amigo](#h.cgj7ezx5chb)

[3.2 Modificar Amigo](#h.qyf15x2lw2yx)

[3.3 Borrar Amigo](#h.e8ks6cfuqoxx)

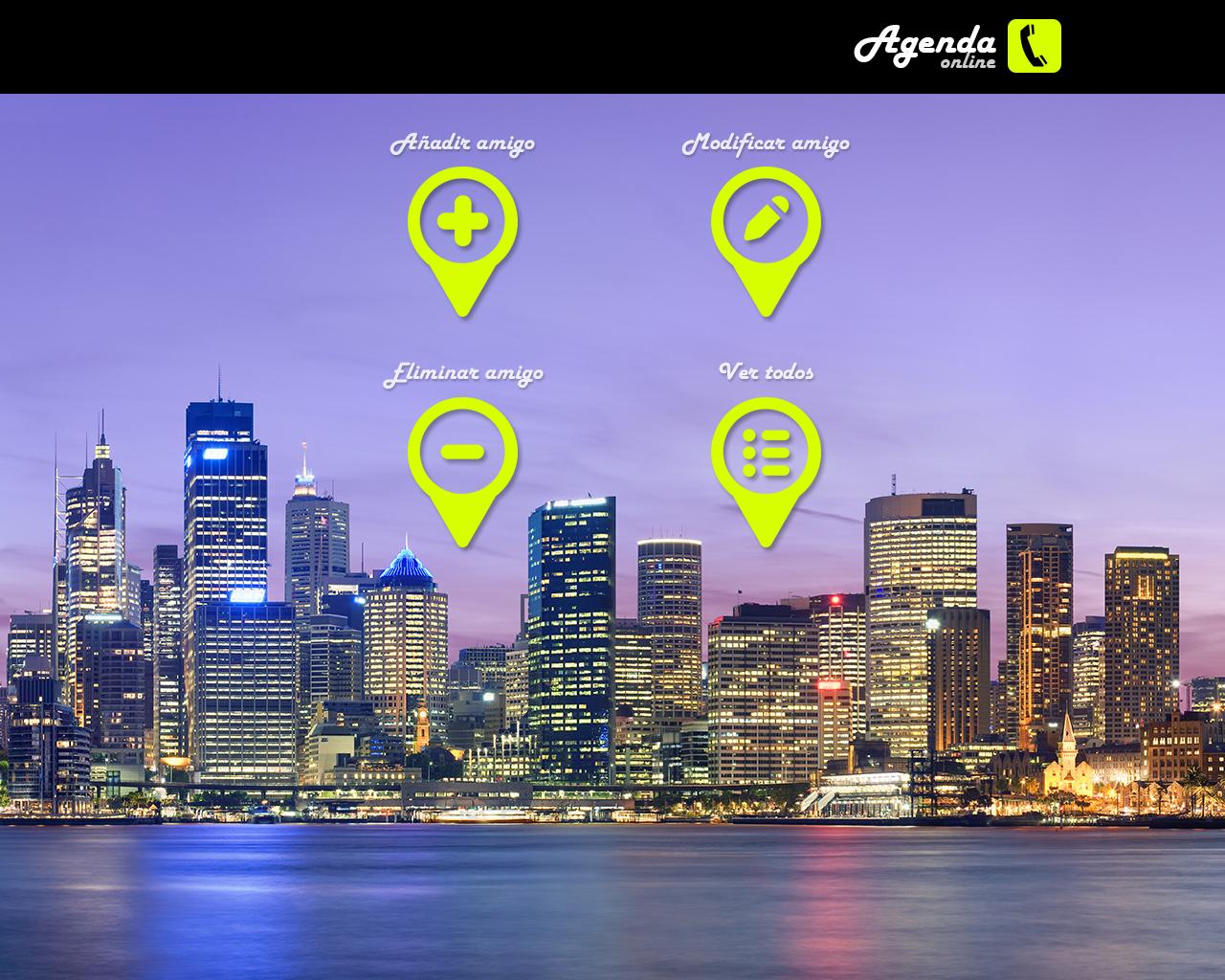
[3.4 Ver Amigos](#h.83wbfgkx5qu5)

## 1.Análisis funcional

Mediante la Aplicación Agenda Telefónica se pueden realizar las siguientes operaciones:

* Insertar Amigo
* Modificar Amigo
* Borrar Amigo
* Listado Amigos

Al arrancar la aplicación nos cargará la página principal de la aplicación que nos permitirá realizar una de las acciones anteriormente comentadas:

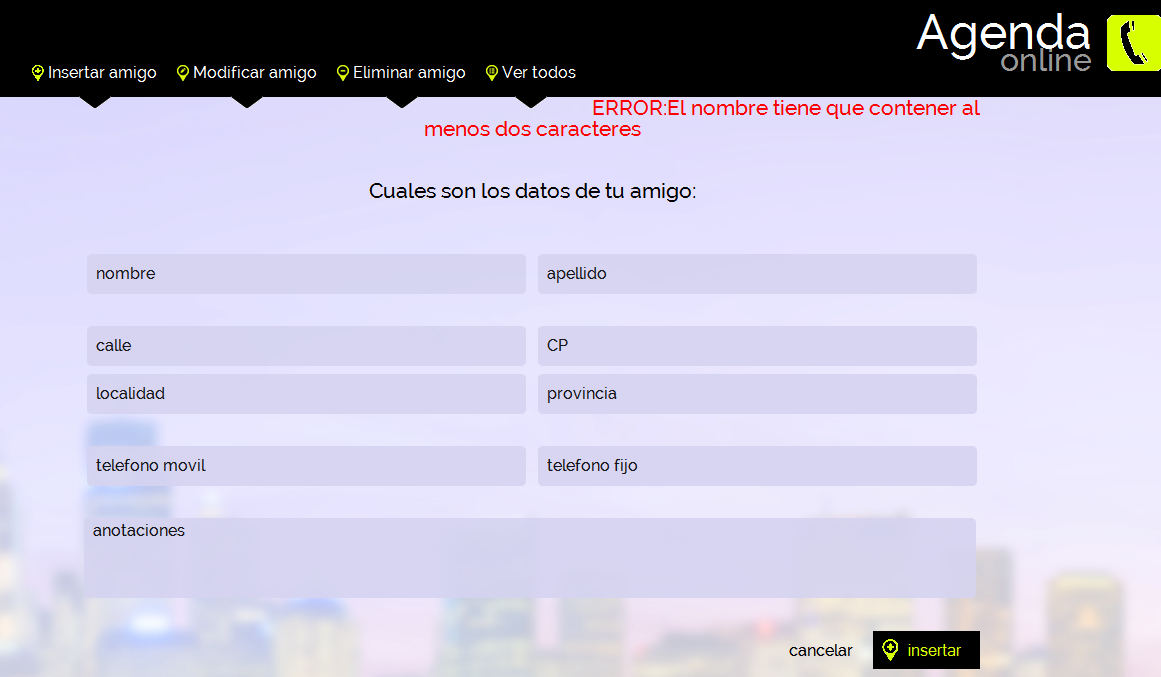


### 1.1 Insertar Amigo

Haciendo clic en añadir amigo aparecerá la siguiente ventana en la que introduciremos todos los datos del Amigo:



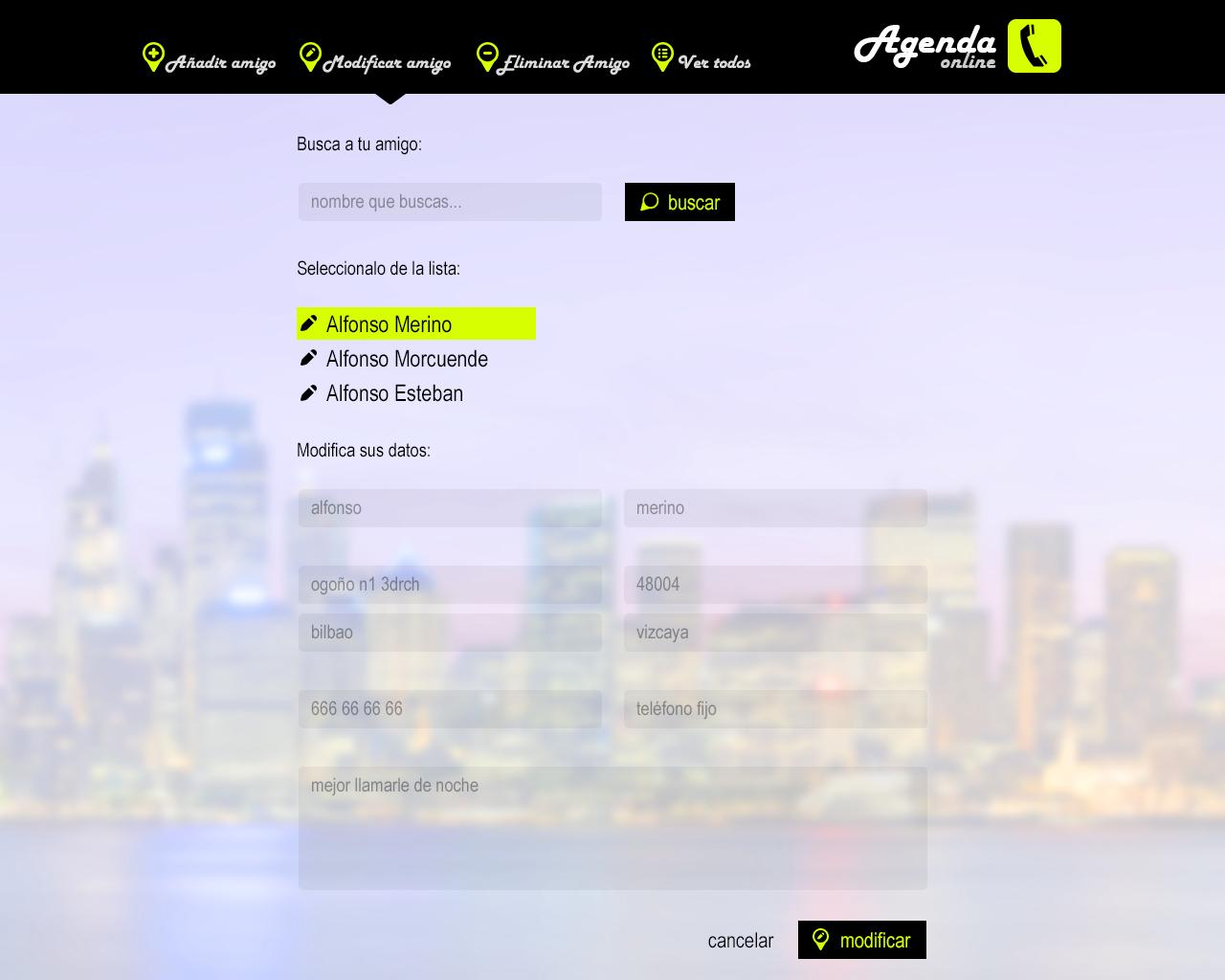
Si el usuario no introduce todos los campos o algun campo no cumple con unas caracteristicas saltará un mensaje de error:

Para almacenar al amigo en la BD hay que hacer clic en el botón añadir. Si todo está OK aparecerá la siguiente ventana con un mensaje de confirmación que nos permitirá añadir un nuevo amigo, modificar uno existente o eliminarlo y listar todos los amigos.



### 1.2 Modificar Amigo

Si se desea cambiar algún dato personal del amigo haremos clic en Modificar Amigo. Te permitirá buscar al amigo/amigos por nombre. Debajo del formulario de búsqueda aparecerá el amigo seleccionado con todos sus campos recogidos desde la BD. Aparecerá la siguiente ventana.



Si se ha introducido en el formulario de búsqueda un amigo que no se encuentra en la BD parecerá el siguiente error:

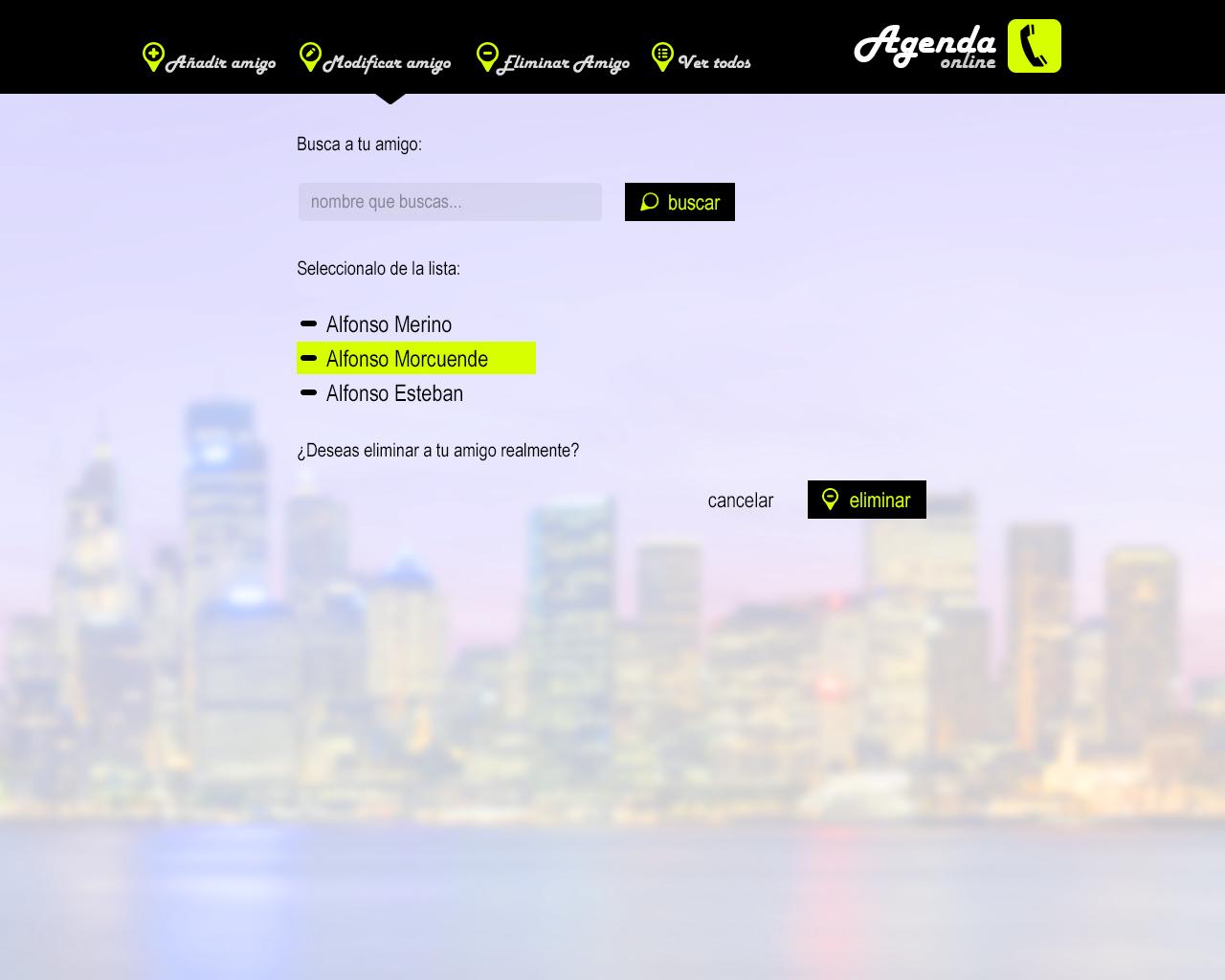


Para modificar al amigo en la BD hay que hacer clic en el botón modificar. Si todo está OK aparecerá la siguiente ventana con un mensaje de confirmación que nos permitirá añadir un nuevo amigo, modificar uno existente o eliminarlo y listar todos los amigos.



1.3 Borrar Amigo

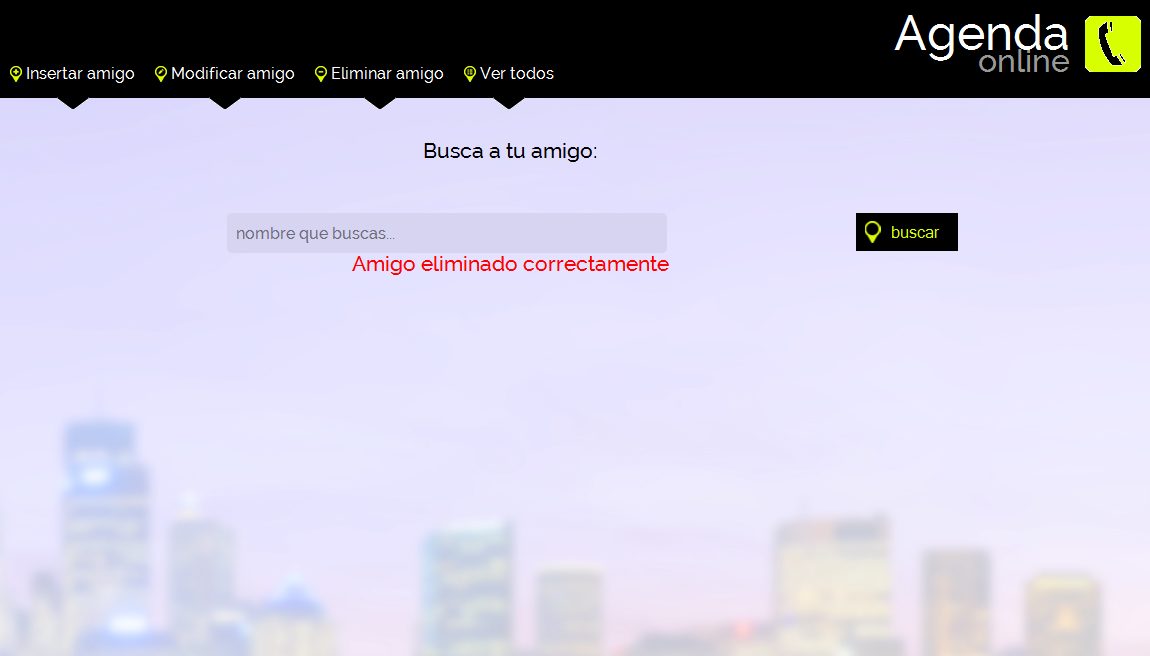
Si se desea eliminar a un amigo haremos clic en Eliminar Amigo. Te permitirá buscar al amigo/amigos alfabéticamente. Debajo del formulario de búsqueda aparecerá el amigo seleccionado a eliminar. Aparecerá la siguiente ventana.



Si se ha introducido en el formulario de búsqueda un amigo que no se encuentra en la BD parecerá el siguiente error:



Para borrar al amigo de la BD hay que aceptar el mensaje de confirmación. Si todo está OK aparecerá la siguiente ventana:



### 

### 

### 

### 

### 1.4 Listado Amigos

En la siguiente ventana, podremos ver los amigos existentes en la Base de Datos:



1.5 Navegación

Durante todo momento, salvo en la home, contaremos con una barra de navegación en la parte superior de la página que nos permitirá acceder a las diferentes secciones de la aplicación, es decir, la home, añadir, modificar, eliminar y ver.

## 2. MVC (Modelo Vista Controlador)

Para construir esta aplicación hemos utilizado el patrón MVC.

### 2.1 Modelo

Las clases mostradas en el siguiente listado son las necesarias para hacer funcionar la parte servidor de la aplicación Agenda Telefónica.

* **Amigo.java**: Será la clase encargada de gestionar los objetos amigo que almacenarán los datos de los contactos que esté tratando la aplicación. Contará con los mismos atributos que los datos a introducir en la aplicación más un campo id autoincremental para la base de datos:
  + id
  + nombre
  + apellido
  + calle
  + cp
  + localidad
  + provincia
  + móvil
  + fijo
  + anotaciones
* **AmigoException.java**: Será la clase encargada de gestionar las excepciones necesarios al crear el objeto Amigo, modificarlo o introducirlo en la Base de datos.
  + nombre: El campo tiene que tener al menos 2 caracteres y no ser nulo.
  + apellido: El campo tiene que tener al menos 2 caracteres y no ser nulo.
  + calle: El campo tiene que tener al menos 2 caracteres y no ser nulo.
  + cp: El campo tiene que tener 5 digitos.
  + localidad: El campo tiene que tener al menos 2 caracteres y no ser nulo.
  + provincia: El campo tiene que tener al menos 2 caracteres y no ser nulo.
  + móvil: El campo tiene que tener 9 digitos.
  + fijo:El campo tiene que tener 9 digitos.
* **IAmigable.java**: interfaz que define los métodos para el DAOAmigo. Estos métodos a implementar serán los siguientes:
  + boolean insertarAmigo(Amigo amigo)
  + ArrayList<Amigo> getAll()
  + ArrayList<Amigo> getByName(String value)
  + Amigo obtenerAmigoByID(int id)
  + boolean update(Amigo amigo)
  + delete(int id)
* **UtilAmigo.java**: Será la clase de Utilidad encargada de testear los diferentes atributos de la clase amigo y en caso de ser incorrecto lanzar las excepciones declaradas en AmigoException.
* **DAOAmigo.java**: clase que implementa el interfaz IAmigable y cuyo cometido es implementar sus métodos y realizar las conexiones pertinentes a base de datos.
* **ModeloAmigo.java**: esta clase es la encargada de crear el modelo del amigo para gestionar la base de datos de manera que el usuario no se percate de ello.
* **ConnectionFactory.java:** Clase que se encarga de crear una única instancia de la conexion a la BD y una unica instancia de DAOAmigo y contiene los metodos de obtener la conexion con la BD, obtener la instancia del DAOAmigo y devolverla.

### 2.2 Vista

En principio serán necesarios los siguientes .jsp para la aplicación:

Las JSPs podemos dividirlas en dos secciones, las principales y las complementarias. Las principales serán las asociadas a las 5 páginas principales de la aplicación, es decir, la home, añadir, modificar, eliminar y ver. Todas estas incluyen las JSPs head.jsp, header.sp, footer.jsp, navBar.jsp y algunas incluirán formularios y el buscador.

* **index.jsp**: es la página principal que muestra las diferentes opciones que podremos realizar en la aplicación.
* **ver.jsp**: nos muestra el listado de los contactos de la agenda.

Dentro de la carpeta form se enuentran los siguientes JSPs adicionales:

* **anadir.jsp**: es el formulario necesario para la creación de nuevos contactos.
* **modificar.jsp**: es el formulario necesario para editar los datos de los contactos.
* **eliminar.jsp**: formulario que contiene el buscador y la lista de contactos para eliminar.
* **buscador.jsp**: es el buscador dinámico que irá generando una lista con los contactos según vaya encontrando coincidencias entre los escrito en el y la base de datos.

Además existen cuatro JSPs comunes para todas las páginas:

* **head.jsp**: contiene la cabecera de la página con su título y la carga de estilos y scripts.
* **header.jsp**: contiene el logo más la barra de navegación, esta última siempre y cuando no nos encontremos en la home de la aplicación.
* **footer.jsp**: contiene el pie de la página.
* **navBar.jsp**: esta será la barra de navegación de la aplicación. Si nos encontramos en la home se mostrará como cuatro iconos en grande en medio de la página, mientras que si estamos en las demás se mostrará como una barra con las diferentes opciones en la parte superior de la página.

### 

### 

### 

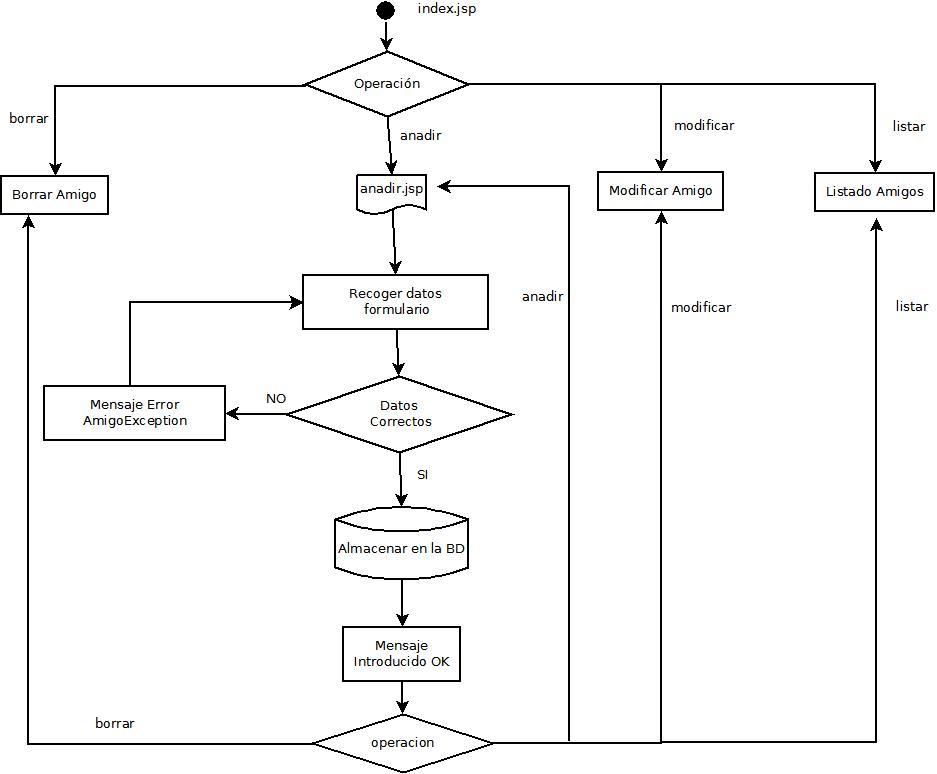
### 2.3 Controlador

En principio se necesitarán unicamente dos servlet:

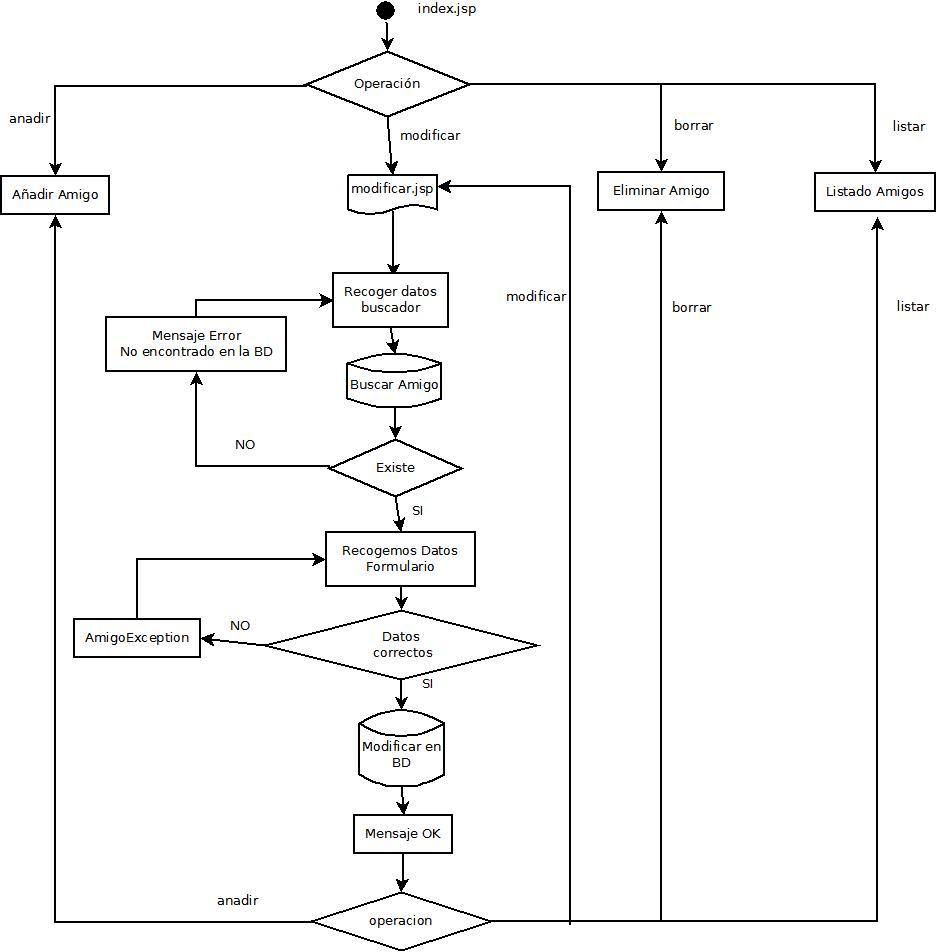
* **AgendaServlet.java:**este será el servlet principal por el que pasarán todas las llamadas y que se encargará de realizar las diferentes operaciones. Dependiendo de los distintos parámetros que le lleguen mediante el get y el post, este realizará los siguientes métodos:
  + anadirAmigo: será el método encargado de guardar un nuevo contacto en la base de datos.
  + modificarAmigo: será el método encargado de modificar el contacto seleccionado.
  + eliminarAmigo: será el método encargado de eliminar un contacto.
  + verAmigos: será el encargado de listar todos los amigos.
  + buscar: realizará en la base de datos la búsqueda de los amigos que concuerden con el nombre y apellido introducidos.
* **ServletMaestro.java:** éste será el encargado de iniciar la configuración de log4java y el del que heredará AgendaServlet.

## 3. Diagramas de Flujo

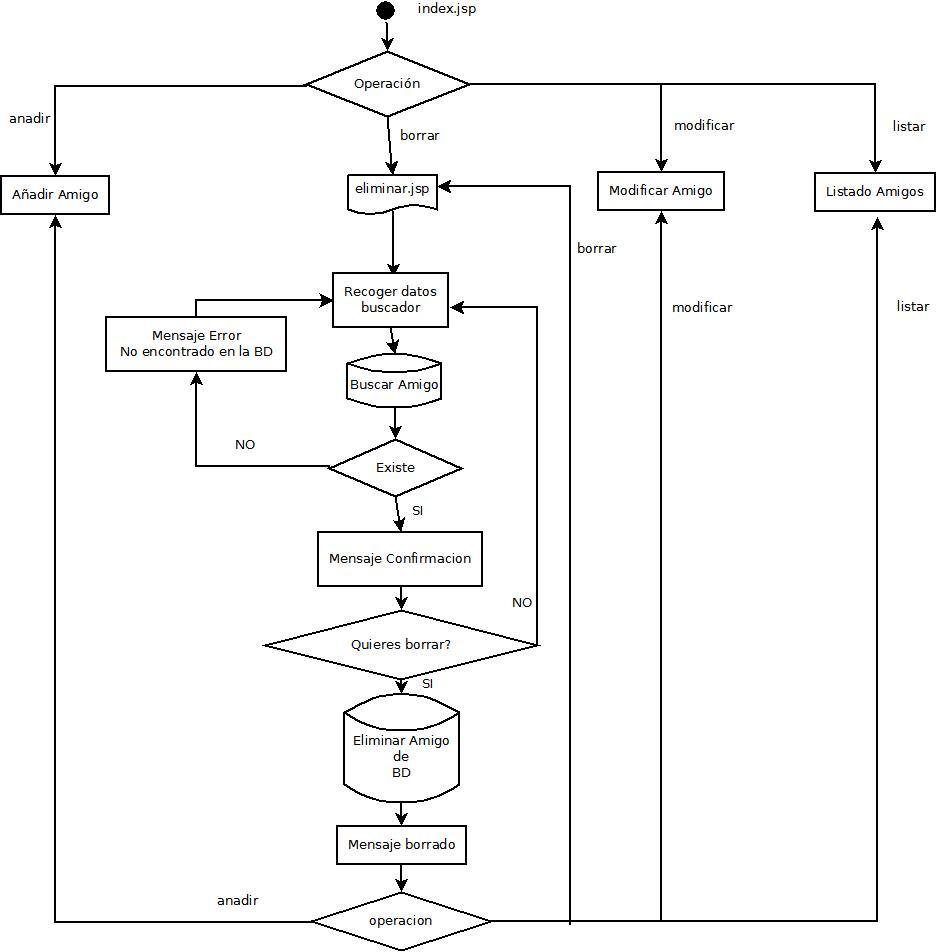
### 3.1 Insertar Amigo



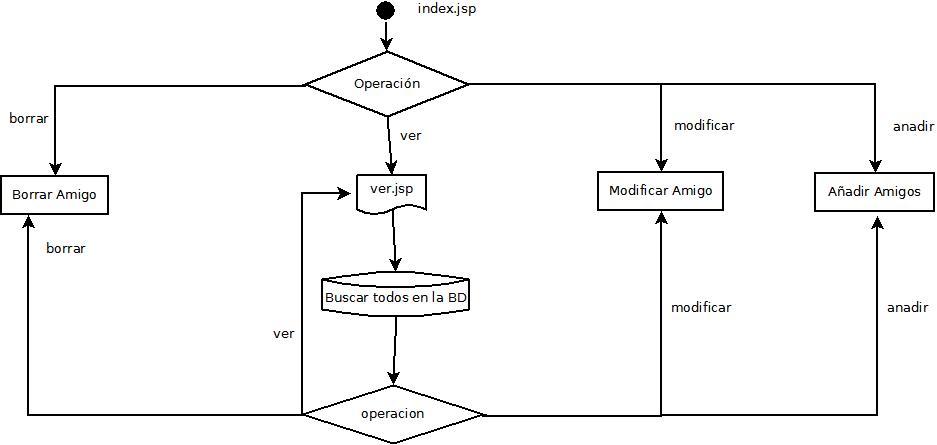
### 3.2 Modificar Amigo



### 3.3 Borrar Amigo



### 3.4 Ver Amigos



## 