

Proyecto Agenda



Autores

Aritz Tellaeche
Kepa Escudero
Jose A. Perez
Susana Costoya

Indice

1. Introducción
2. Elementos que componen la Agenda
 - 2.1. Home (ver todos)
 - 2.1.1. Campo de busqueda y Botón Limpiar
 - 2.1.2. Botón Añadir
 - 2.1.3. Botón Eliminar
 - 2.1.4. Botón Detalle (icono ojo)
 - 2.2. Añadir
 - 2.3. Detalle
3. Elementos Tecnicos
4. Diagramas de Flujo
 - 4.1. Añadir
 - 4.2. Eliminar
 - 4.3. Buscar
 - 4.4. Ver todos
 - 4.5. Modificar
5. Diagramas de Secuencia
 - 5.1. Home
 - 5.1.1. Petición Buscar
 - 5.1.2. Petición Añadir
 - 5.1.3. Petición Eliminar
 - 5.1.4. Petición Detalle

1. Introducción

Vamos a crear una aplicación web de una Agenda.

Esta se va a crear con HTML5 y CSS3 para la parte del cliente y con tecnología Java para la parte del servidor. Los datos de esta agenda se guardarán en una base de datos MySql.

A continuación pasaremos a detallar el proyecto.

2. Elementos que componen la Agenda

2.1. Home

Los JSPs **constarán** de una **cabecera** y un **cuerpo**.

Cabecera: logo, campo de búsqueda, botón limpiar , botón eliminar y botón añadir (Será la misma para todas las páginas de la aplicación, pero ciertos elementos quedarán deshabilitados dependiendo de la página en la que nos encontremos).

Cuerpo: tendrá una tabla en el que se nos mostrará todos los contactos que componen nuestra agenda, que contiene nuestra base de datos. Cada registro tendrá por delante un "checkbox" para poder marcarlos en caso de querer eliminar varios contactos simultáneamente además de contar con un botón "ver detalle" representado por el icono de un ojo, que nos llevará a la página de detalle, allí se podrán modificar los campos de dicho registro.

Podremos visualizar en la parte inferior de la tabla enlaces de paginación, "1,2,3...", que nos indicarán el número de páginas en caso de que el número de registros de nuestra base de datos sea superior al de registros que entran en la tabla. Este lo podremos controlar desde la lista que encontramos en la parte superior de nuestra tabla.

2.1.1. Campo de búsqueda y Botón Limpiar

En el campo de búsqueda podremos realizar las consultas de nuestros registros (contactos), los cuales, se filtraran por nombre o telefono y se enviarán las consultas a nuestra base de datos al pulsar el botón buscar.

Sino existe el contacto al buscar nos saldrá un mensaje; "No existe ningún contacto, por favor cree un contacto nuevo" este se cargará en lugar de la tabla con los contactos.

El botón limpiar nos limpiara los datos del campo de búsqueda para poder realizar una nueva búsqueda.

2.1.2. Botón Añadir

Si hacemos click sobre este botón nos llevará a la pantalla de añadir nuevo contacto. Ver punto 2.2 Añadir.

2.1.3. Botón Eliminar

Si hacemos click sobre este botón y ningún registro tiene marcado el "checkbox" nos aparecerá una ventana modal en la que aparecerá el siguiente mensaje: "No hay ningun contacto seleccionado", además de un botón "continuar" que simplemente cerrará la ventana modal.

Si se hace click sobre éste boton y existen uno o varios registros con el "checkbox" marcado nos aparecerá una ventana modal con éste mensaje: "desea eliminar los contactos seleccionados" y dos botones, uno "aceptar" que eliminará los registros definitivamente de la base de datos y otro "cancelar" que anulará la operación y volverá a la pantalla principal.

Al hacer click sobre cualquiera de los botones nos llevará a la pantalla principal pero justo debajo de la cabecera aparecerá el siguiente mensaje: "Se han eliminado con éxito los contactos" en caso de haber hecho click sobre "continuar", y con éste otro "Se ha cancelado la operación, ningún registro ha sido borrado" en caso de haber pulsado "cancelar".

Este botón estará deshabilitado en la página de "nuevo", donde carece de sentido.

2.1.3. Botón Detalle

Si hacemos click sobre el boton "detalle" (representado por un icono de un ojo) nos llevará a un formulario relleno prácticamente igual al de "añadir" en el que podremos realizar cambios en los datos, al cambiar cualquier dato se activará el boton "modificar" que por defecto estará deshabilitado. También tendremos el botón "volver" que nos llevará a la pantalla principal.

Si hacemos click sobre el boton "modificar" y algún campo de los obligatorios no está relleno o está relleno incorrectamente nos aparecerá un Tooltip con el siguiente mensaje: "debe rellenar los campos obligatorios" en caso de que alguno de los campos obligatorios no esté relleno o "algunos campos no estan rellenos correctamente, revise los datos introducidos" en caso de no estar rellenos correctamente (números en el nombre, letras en el teléfono...) con un botón "continuar" que cerrará la ventana, exactamente igual que al introducir un nuevo amigo.

Si hacemos click sobre el boton "modificar" y los campos están rellenos correctamente además de estar rellenos los campos obligatorios nos aparecerá otra ventana modal con el mensaje "Se va a proceder a guardar los cambios del contacto" y dos botones, "continuar" que nos llevará a la pantalla principal (Home) donde aparecerá el siguiente mensaje bajo la cabecera: "El contacto se ha modificado con éxito"; y "cancelar" que nos llevará también a la pantalla principal y mostrará éste otro mensaje bajo la cabecera: "Se ha cancelado la operación, no se hizo ninguna modificación".

2.2. Añadir

En la cabecera nos encontraremos el botón de "añadir" y de "eliminar" desactivados ya que carece de sentido.

En el cuerpo de página observaremos un formulario con los siguientes campos a rellenar (*OBLIGATORIO); Nombre*, apellidos, teléfono fijo, teléfono móvil*, domicilio, localidad, provincia, C.P. y anotaciones.

Además de dos botones "añadir" y "cancelar".

Una vez rellenos los campos si realizamos un click sobre el boton "añadir" y algún campo de los obligatorios no está relleno o está relleno incorrectamente nos aparecerá un Tooltip indicándonos los campos que no estan correctamente rellenos o los que quedan por rellenar y son obligatorios.

En el caso que el nuevo contacto tenga todos los datos correctos al hacer click sobre "añadir" nos mostrará una ventana modal indicando un mensaje: "Se va a proceder a crear un contacto nuevo" y con dos botones: "continuar" que nos lo añadirá a nuestra base de datos y nos devolverá a la home con un mensaje de " Su contacto fue creado con éxito", y de "cancelar" que nos enviara también a la home y nos mostrará un mensaje de "Se ha cancelado la operación, no se ha añadido el registro"

2.3. Detalle

En el cuerpo de la página de detalle nos mostrará un formulario exactamente igual que el de añadir con la diferencia que el formulario se nos mostrará con los datos del contacto ya rellenos, los cuales se cargarán previamente de nuestra base de datos.

Los campos de dicho formulario estarán desde el principio editables y al final del formulario encontramos dos botones: “modificar” y “volver”.

Si modificamos algunos de ellos y hacemos click en “modificar”, estos se comprobarán y si no son correctos nos saldrán un Tooltip indicando de que no son correctos y que los revisemos. Por el contrario si estos son correctos al dar modificar nos saldrá una ventana modal que nos muestra un mensaje: “Se va a proceder a guardar los cambios del contacto” y dos botones: “continuar” que nos llevará a la home mostrando el mensaje: “El contacto se ha modificado con éxito”, y el de “cancelar” que también nos llevará a la home mostrando el mensaje: “Se ha cancelado la operación, no se hizo ninguna modificación”.

3. Elementos Tecnicos

Esta aplicación la vamos a realizar con HTML5 y CSS3 por parte del lado del cliente y con tecnología Java por parte del servidor. También utilizaremos como motor de base de datos MySQL.

La base de datos que vamos a crear se llamará "agenda" y la tabla "contacto".

Los archivos necesarios que se van a crear en el paquete de java para que esta aplicación se lleve a cabo serán la siguientes;

- package com.ipartek.formacion.agenda;
 - Constantes.java => para poder tener definidas las constantes y ser reutilizadas
- package com.ipartek.formacion.agenda.bean;
 - Persona.java => creamos el objeto Persona en el cual creamos, recogemos y validamos los siguientes datos;
 - i. Datos de texto; nombre, apellidos, direccion, población, provincia y anotaciones
 - ii. Datos numericos; telFijo, telMovil y cp
 - iii. Dato auto-increment; idcontactoTodos los campos tienen en el método SET su previa comprobación
- package com.ipartek.formacion.agenda.controller;
 - PersonaServlet.java => Para realizar las funciones de mostrar, añadir , modificar o eliminar un contacto
- package com.ipartek.formacion.agenda.modelo.dao;
 - DAOFactory.java => Factoria abstracta que determinara el tipo de base de datos que vamos a utilizar
 - IPersonaDAO.java => Interfaz en la cual definimos las operaciones CRUD contra la tabla "contacto" de la base de

datos

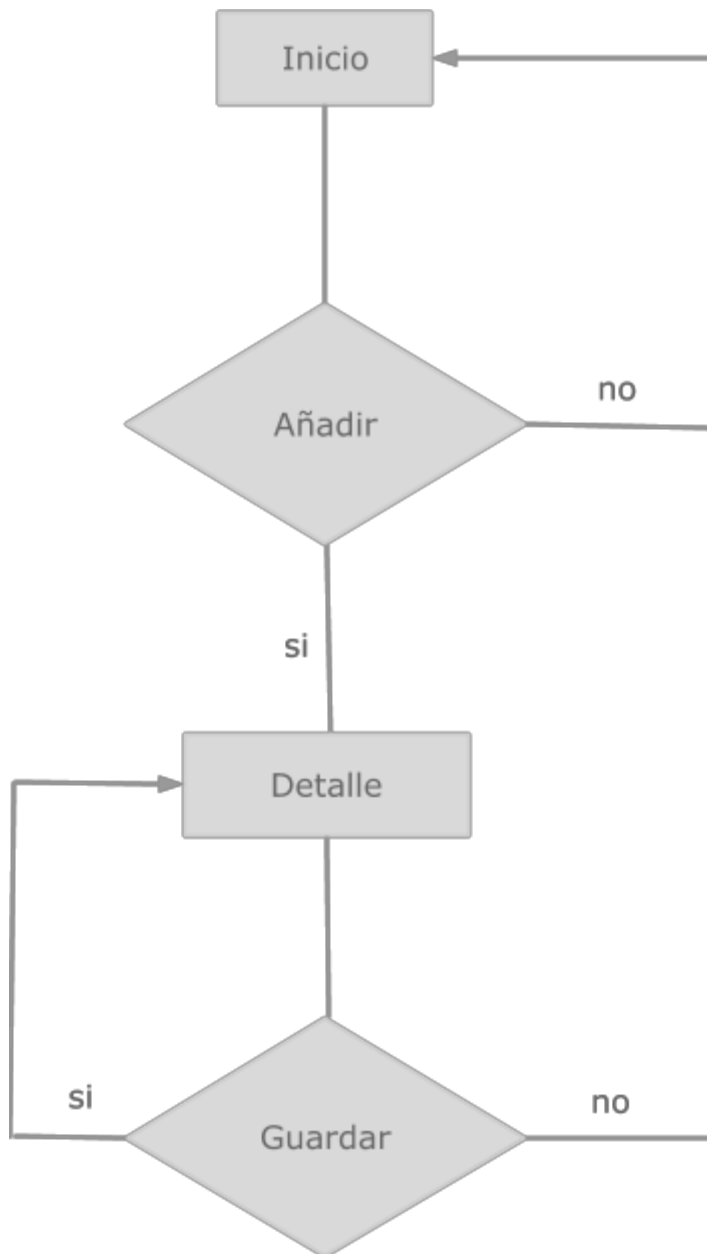
- ModelException.java => Clase para tratar los mensajes de las excepciones
- MySqlIDAOFactory.java => Conexión a la base de datos
- PersonaMySqlDAO.java => Implementa los metodos CRUD de la inteface IPersonaDAO.java

Los archivos JSP necesarios ubicados en la carpeta WebContent;

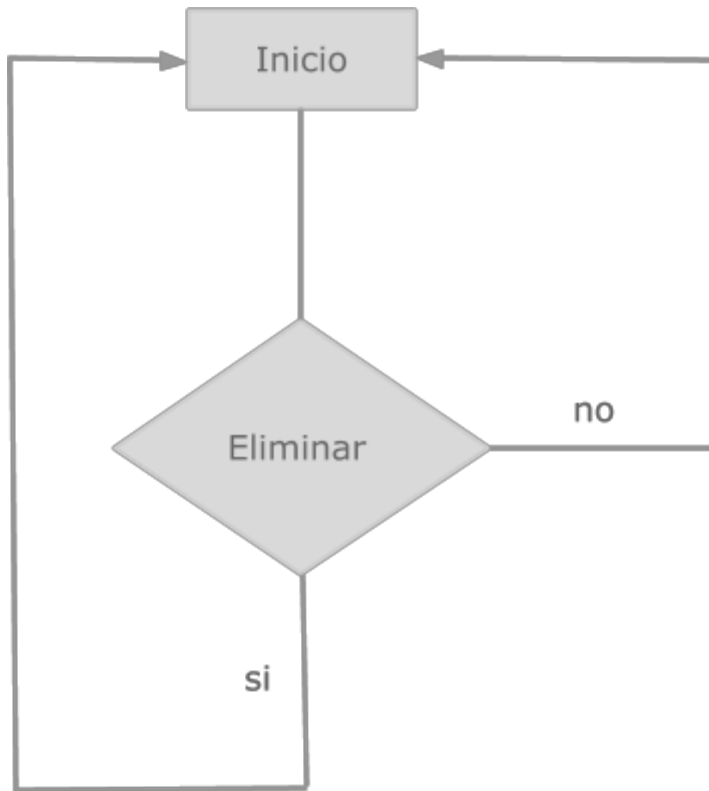
- head.jsp = > (Include) Cabecera del proyecto
- nav.jsp = > (Include) Menu de navegacion del proyecto
- modalDelete.jsp => (Include) Ventana modal que se muestra con la advertencia a la hora de borrar un contacto.
- Detalle.jsp => Nos muestra la tabla con todos los contactos de la base de datos.
- Listado.jsp => Nos muestra un formulario con los campos de nuestros contactos, estos campos estarán vacíos en la opción de añadir y con datos en la opción de modificar

4. Diagrama de Flujo

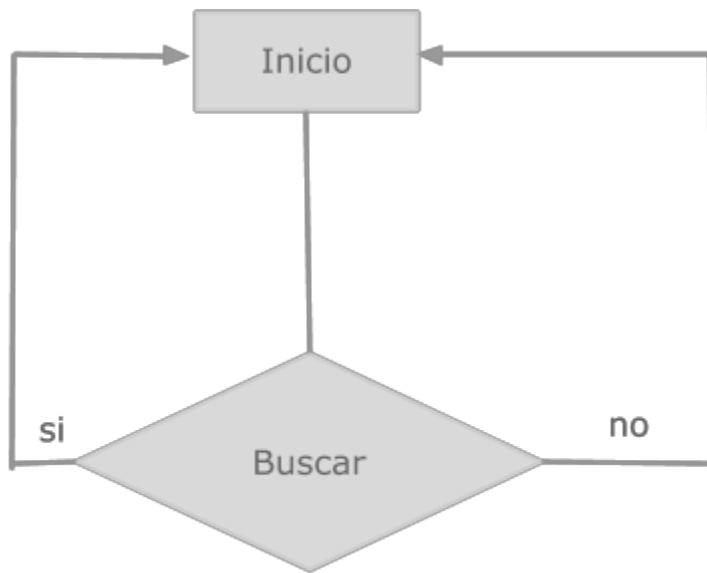
4.1. Añadir



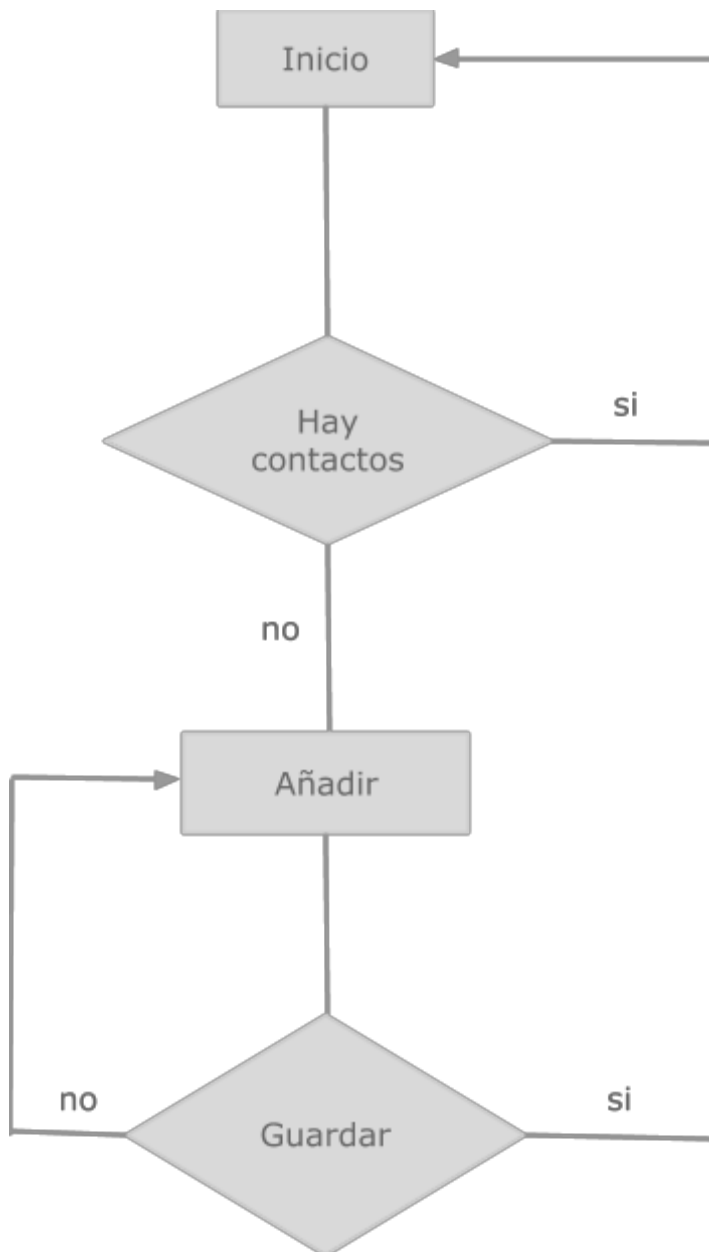
4.2. Eliminar



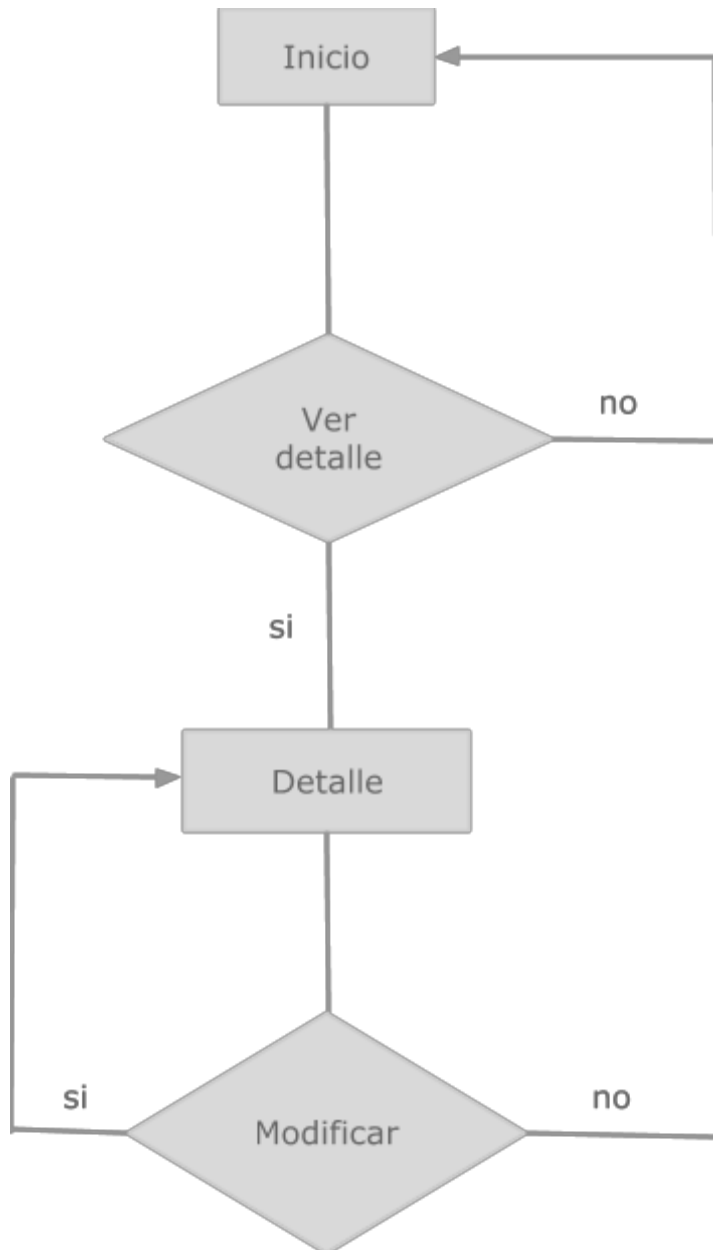
4.3. Buscar



4.4. Ver todos

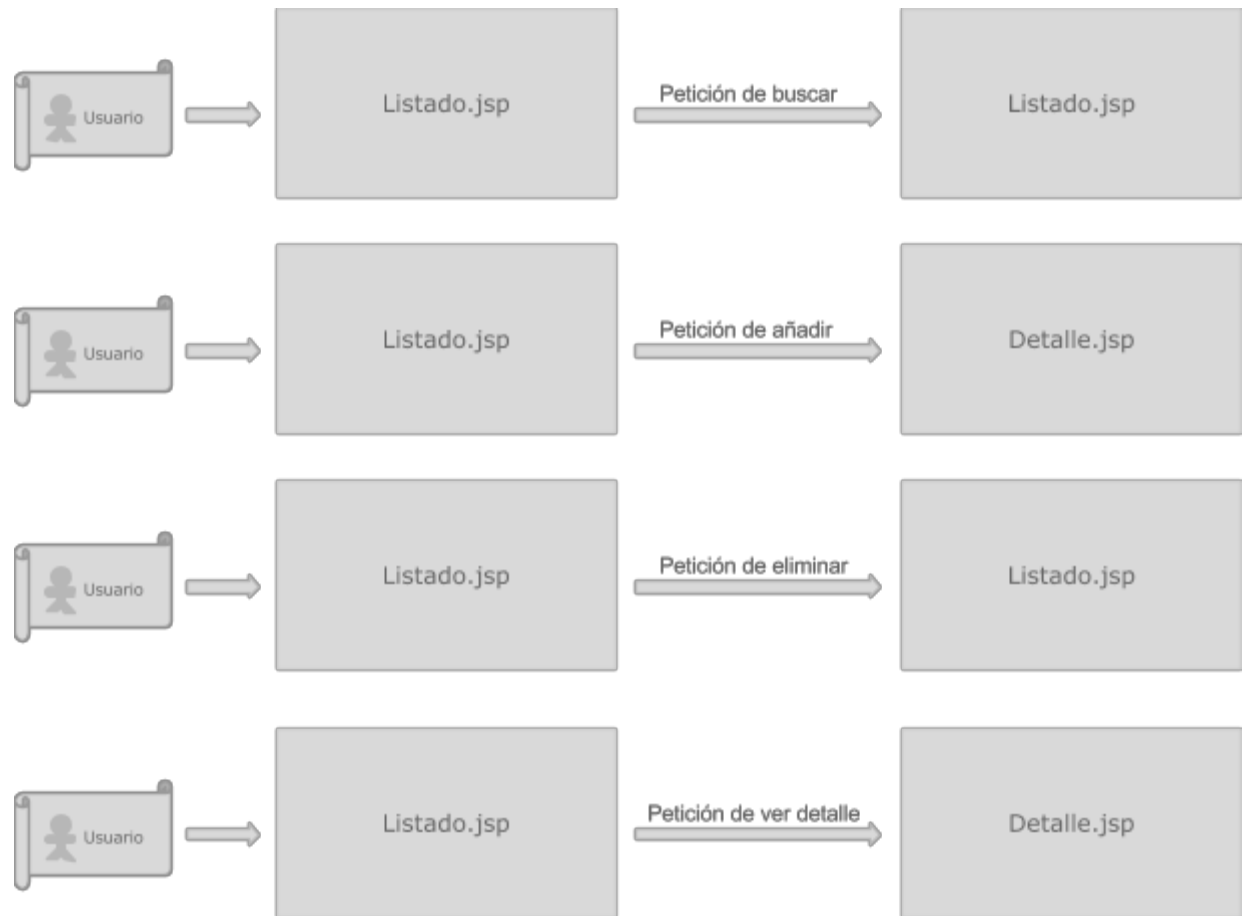


4.5. Modificar

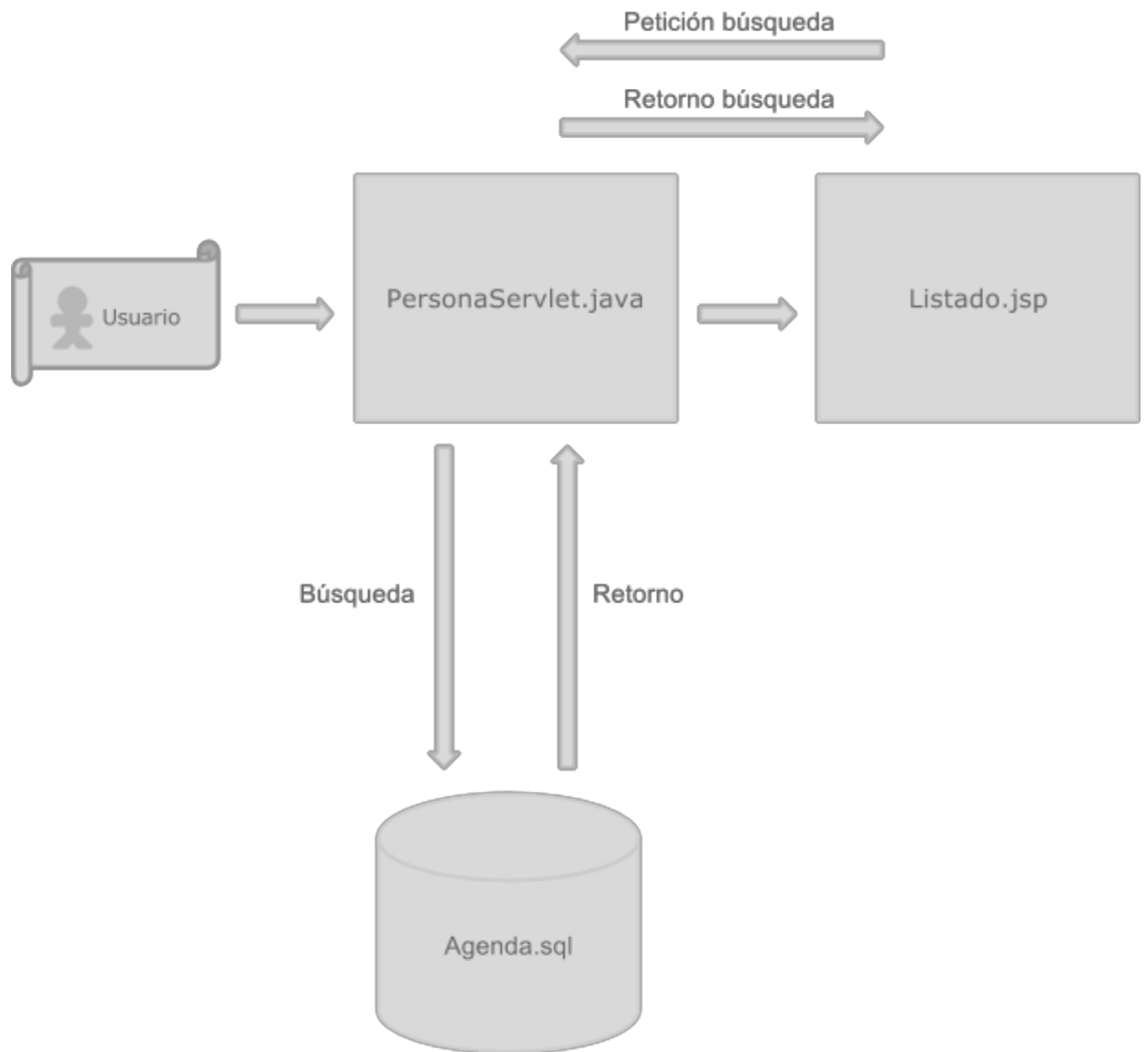


5. Diagramas de Secuencia

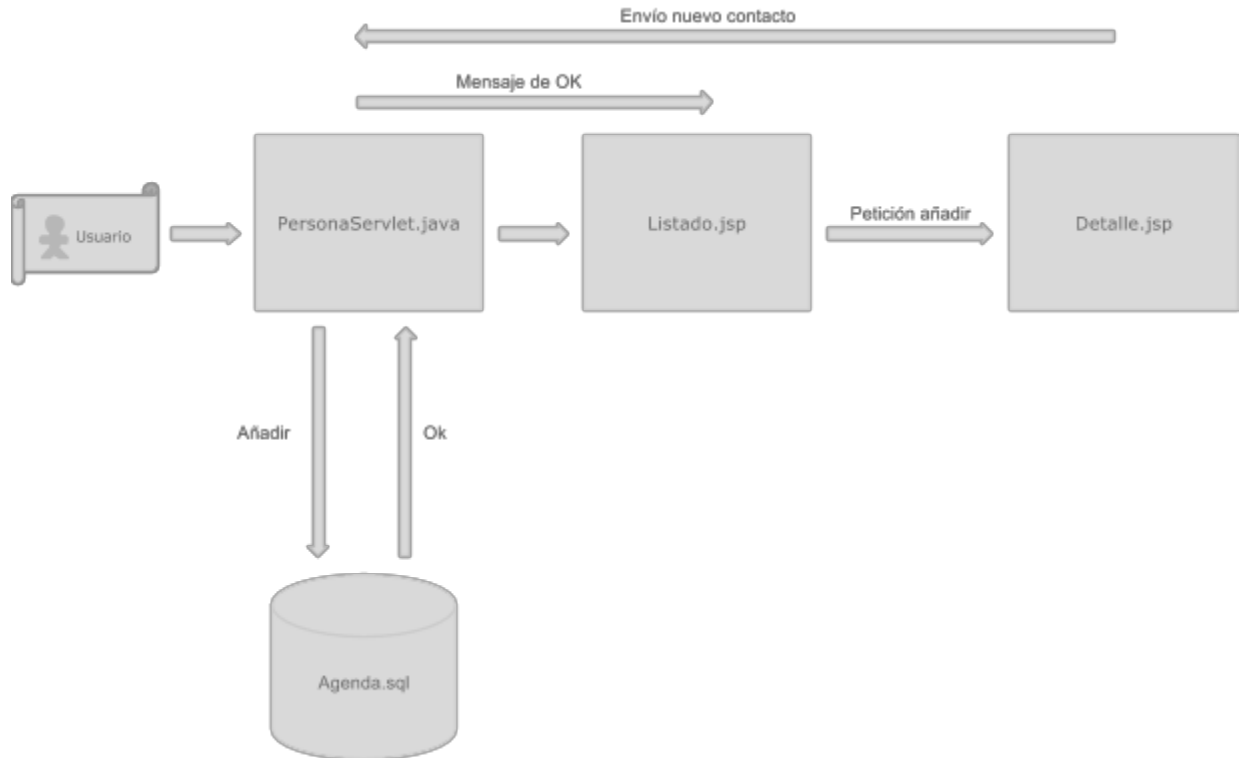
5.1. Home



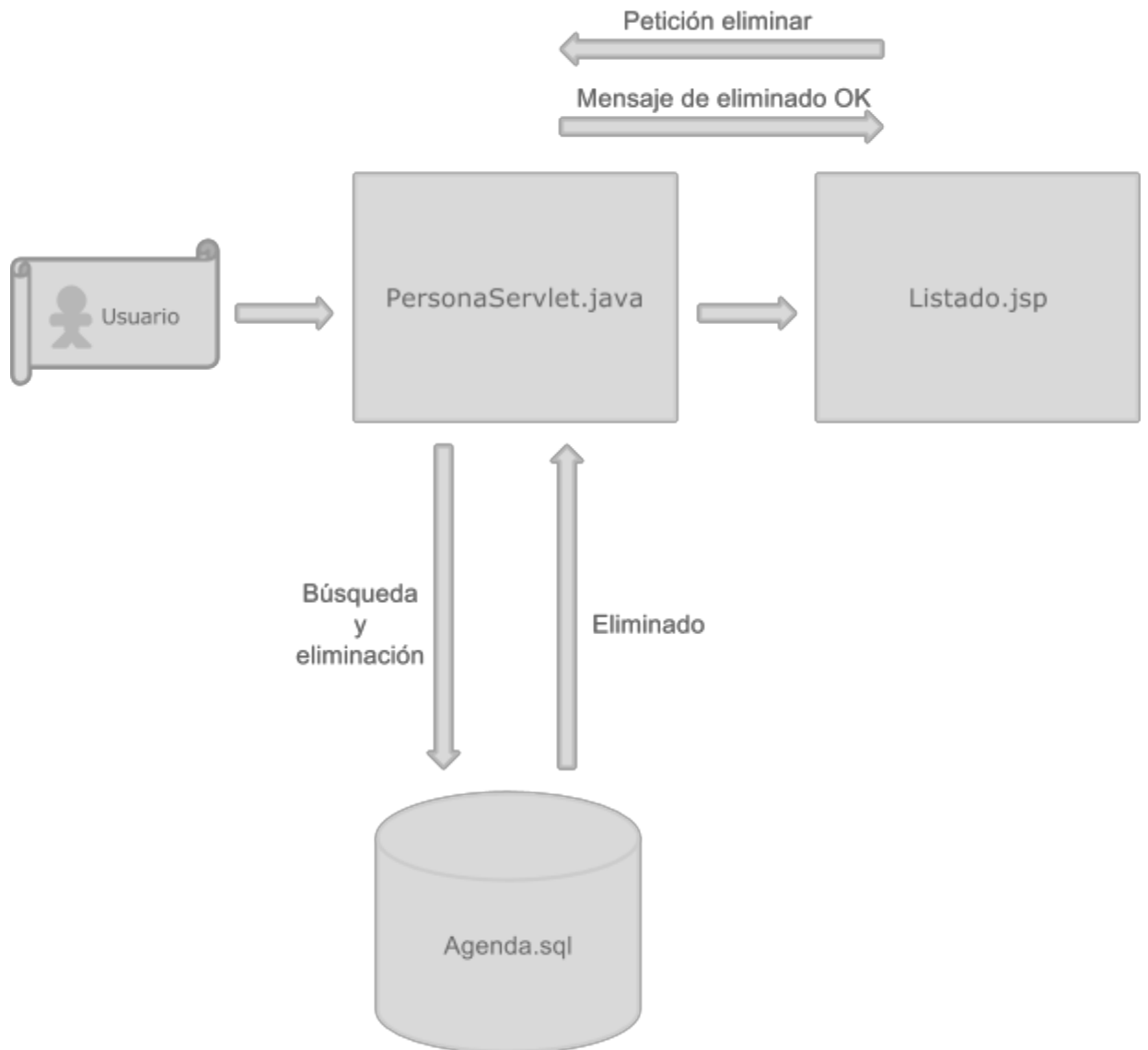
5.1.1. Petición Buscar



5.1.2. Petición Añadir



5.1.3. Petición Eliminar



5.1.4. Petición Detalle

