

# Programación en Lenguaje Java

## Problema 7.1. Uso de ArrayList. Agenda de contactos



**Michael González Harbour**

**Mario Aldea Rivas**

Departamento de Matemáticas,  
Estadística y Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

## Problema 7.1. Uso de ArrayList. Agenda de contactos

### Objetivos

- Practicar la comprensión de un documento de requisitos apoyado por casos de uso.
- Practicar la comprensión de un diagrama de clases sencillo.
- Practicar el uso de la clase `ArrayList`.

### Documento de requisitos

#### Descripción general de la aplicación

Se pretende desarrollar una agenda que permita almacenar nombres de personas junto con su dirección de correo electrónico y su teléfono.

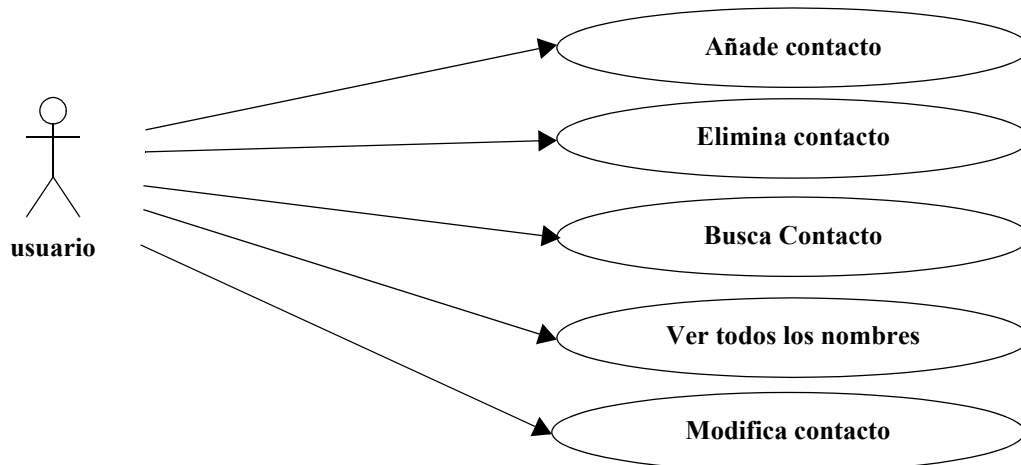
Se denomina “contacto” a cada una de las entradas almacenadas en la agenda. Para cada contacto se almacenan los siguientes datos: nombre, teléfono y dirección de correo electrónico.

Los contactos se identifican por su nombre, por lo que, para evitar confusiones la agenda no podrá contener dos contactos con el mismo nombre (aunque sí con el mismo teléfono o la misma dirección de correo electrónico).

Los contactos de una persona cambian con el tiempo, por lo que la aplicación deberá permitir añadir, borrar y modificar contactos.

#### Casos de uso

Las operaciones que se desea poder realizar sobre la agenda son las descritas en los siguientes casos de uso:



#### Caso de uso “Añade Contacto”

1. El usuario selecciona la opción “añadir contacto”
2. La aplicación muestra una ventana que permite introducir los datos del nuevo contacto
3. El usuario introduce los datos y pulsa aceptar
4. La aplicación añade el contacto a la agenda
  - En el caso de que ya exista otro contacto con el mismo nombre, se notifica el error y no se añade el nuevo contacto.

### Caso de uso “Elimina Contacto”

1. El usuario selecciona la opción “eliminar contacto”
2. La aplicación muestra una ventana que permite introducir el nombre del contacto a eliminar
3. El usuario introduce el nombre y pulsa aceptar
4. La aplicación elimina el contacto con el nombre indicado.
  - Si no existe ningún contacto con ese nombre se notifica.

### Caso de uso “Busca Contacto”

1. El usuario selecciona la opción “buscar contacto”
2. La aplicación muestra una ventana que permite introducir el nombre del contacto
3. El usuario introduce el nombre y pulsa aceptar
4. La aplicación busca un contacto con ese nombre y muestra sus datos (nombre, teléfono y dirección de e-mail).
  - Si no existe ningún contacto con ese nombre se notifica.

### Caso de uso “Ver todos los nombres”

1. El usuario selecciona la opción “ver todos los nombres”
2. La aplicación muestra una ventana con los nombres de todos los contactos almacenados en la agenda.

### Caso de uso “Modifica Contacto”

1. El usuario selecciona la opción “modificar contacto”
2. La aplicación muestra una ventana que permite introducir el nombre del contacto
3. El usuario introduce el nombre y pulsa aceptar
4. La aplicación muestra una ventana que permite modificar los datos del contacto con el nombre indicado
  - Si no existe ningún contacto con ese nombre se notifica y finaliza el caso de uso.
5. El usuario modifica los datos y pulsa aceptar
6. La aplicación modifica los datos del contacto en la agenda

## Requisitos no funcionales

La aplicación contará con una interfaz gráfica implementada con el paquete “fundamentos” (dicha interfaz se supone encargada a otro programador, por lo que no será tarea del alumno su desarrollo).

## Documento de diseño (diagrama de clases)

