

Agenda.java

```
package P26Agenda2;

import java.util.ArrayList;

/**
 * Una clase para mantener un número arbitrariamente grande de notas Las notas
 * están numeradas para facilitar su referencia externa por un usuario En esta
 * versión las notas comienzan con el número 0
 *
 * @author David J. Barnes and Michael Kolling.
 * @version 2008.03.30
 */

class Agenda {
    // Almacenamiento para un número variable de notas
    private ArrayList<String> notes;

    /**
     * Realiza todas las inicializaciones que son necesarias para el uso de la
     * agenda
     */
    public Agenda() {
        notes = new ArrayList<String>();
    }

    /**
     * Almacena una nota nueva en la agenda
     *
     * @param note
     *           La nota que se va a almacenar
     */
    public void storeNote(String note) {
        notes.add(note);
    }

    /**
     * Almacena una nota nueva en la agenda en la posición especificada por un
     * índice
     *
     * @param note
     *           La nota que se va a almacenar
     * @param index
     *           La posición dentro de la agenda la que debe insertarse la nota
     */
    public void storeNote(int index, String note) {
        notes.add(index, note);
    }

    /**
     * @return El número de notas que hay almacenadas en ese momento en la
     *         agenda
     */
    public int numberOfNotes() {
        return notes.size();
    }

    /**
     * Remove a note from the notebook if it exists.
     *
     * @param noteNumber
     *           The number of the note to be removed.
     */
}
```

Agenda.java

```
public void removeNote(int noteNumber) {
    if (noteNumber < 0) {
        // This is not a valid note number, so do nothing.
    } else if (noteNumber < numberOfNotes()) {
        // This is a valid note number.
        notes.remove(noteNumber);
    } else {
        // This is not a valid note number, so do nothing.
    }
}

/**
 * Mostrar una nota
 *
 * @param noteNumber
 *         El número de la nota que se debe mostrar
 */
public void showNote(int noteNumber) {
    if ((noteNumber >= 0) && (noteNumber < numberOfNotes())) {
        // Es un número válido de nota por lo que lo podemos imprimir
        System.out.println(notes.get(noteNumber));
    } else {
        // No es un número válido de nota, así que no hay que hacer nada
    }
}

/**
 * Lista todas las notas que hay almacenadas en la agenda.
 */
public void listNotes() {
    System.out.println("Listado de la agenda con: " + numberOfNotes()
        + " notas");
    for (int index = 0; index < numberOfNotes(); index++) {
        System.out.println(notes.get(index));
    }
}
}
```

AgendaProg.java

```
package P26Agenda2;

public class AgendaProg {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        // Creamos un objeto de tipo agenda
        Agenda agenda = new Agenda();

        // Almacenamos varias notas
        agenda.storeNote("12h Consulta médico");
        agenda.storeNote("15h Comida de trabajo");
        agenda.storeNote(1, "12:30h Llamar por teléfono");

        // Imprimimos las notas
        agenda.listNotes();

        // Eliminamos una nota
        System.out.print("Eliminamos la nota 1: ");
        agenda.showNote(1);
        agenda.removeNote(1);

        // Comprobamos el borrado mostrando la nota 0
        System.out.print("Ahora la nota 1 es: ");
        agenda.showNote(1);

        // Volvemos a imprimir toda la agenda. Ahora con un bucle for en vez del
        // método agenda.listNotes()
        for (int index = 0; index < agenda.numberOfNotes(); index++) {
            agenda.showNote(index);
        }
    }
}
```