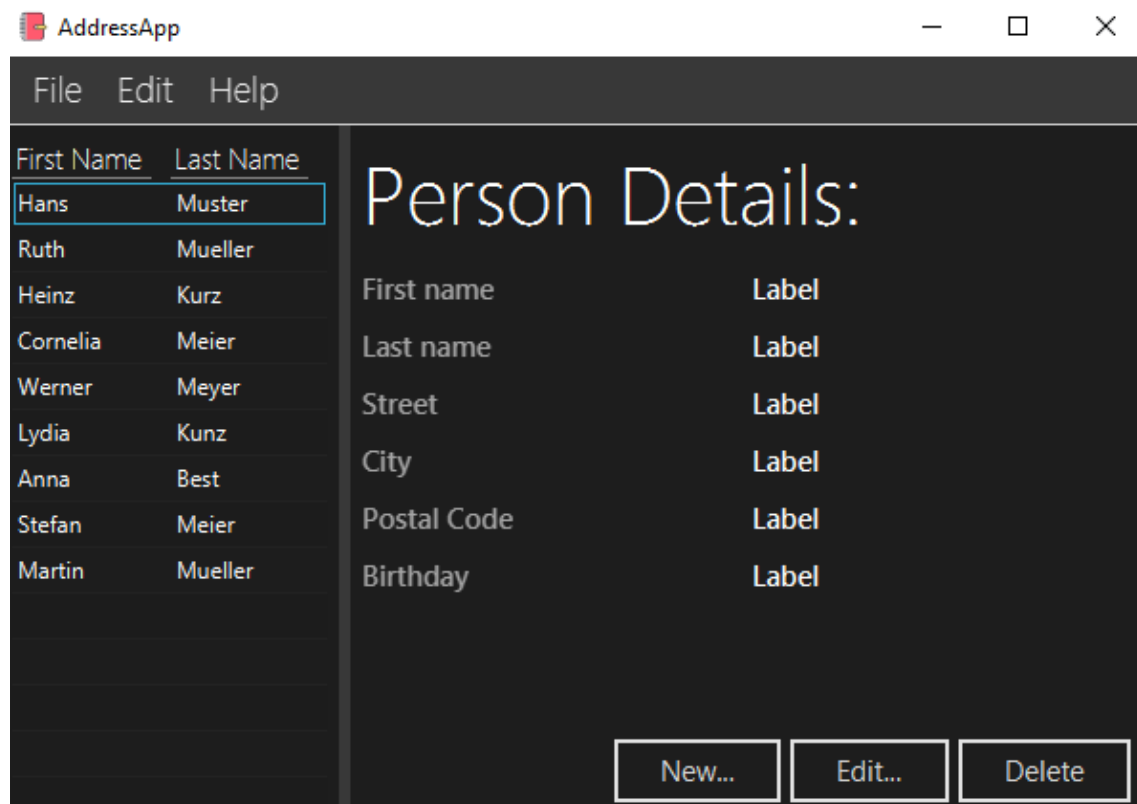


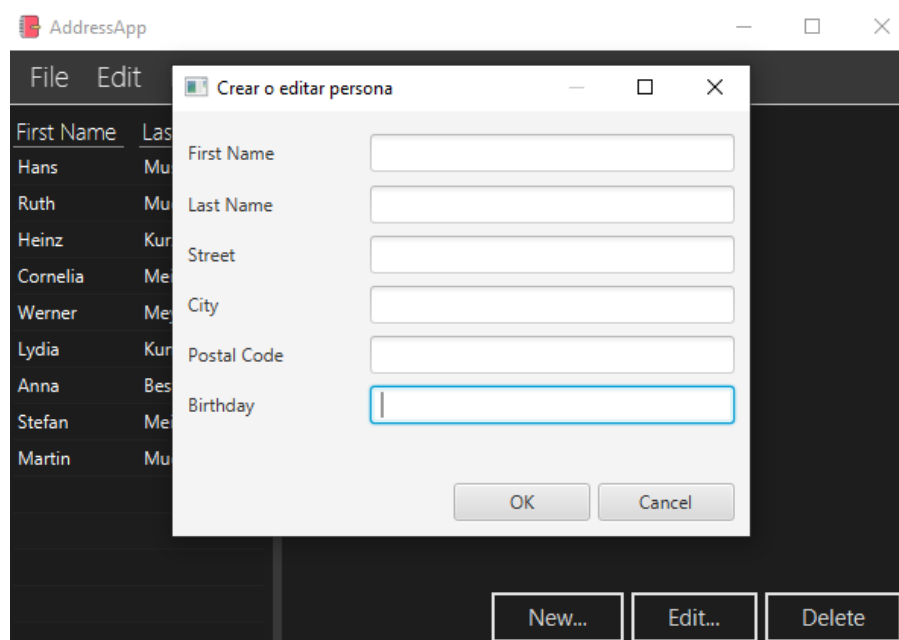
ENUNCIADO

Vamos a continuar con nuestra aplicación de ejemplo, cuya apariencia actual es la siguiente:



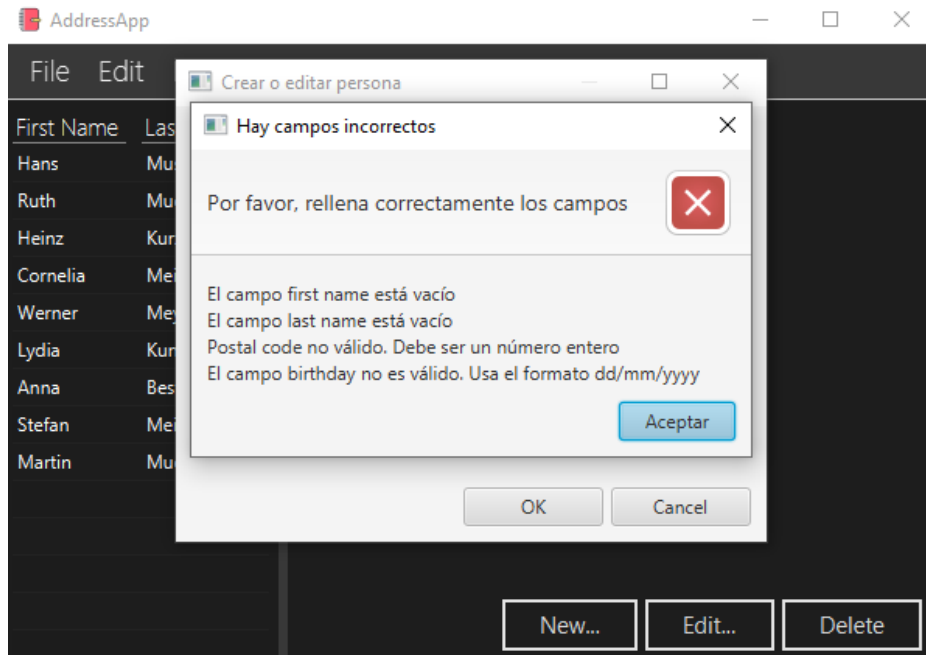
El objetivo es empezar a implementar la funcionalidad. Para ello, vamos a permitir añadir nuevos contactos a la agenda. Esto se llevará a cabo desde el botón “New...”.

Al hacer click en este botón aparecerá el siguiente diálogo personalizado del tipo modal.



Ejercicio sobre diálogos en JavaFX

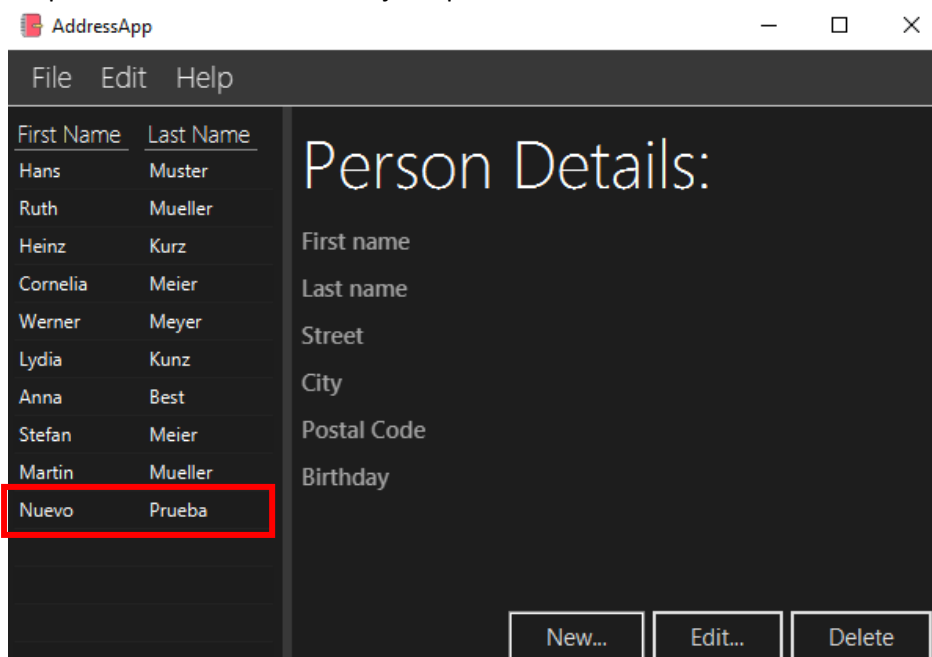
Cuando el usuario pulse el botón “OK” del diálogo, si hay algún dato erróneo, se creará un alert en el que se informa de todos los campos incorrectos como se muestra debajo. Puedes acceder al texto de cada **TextField** con el método **getText()**



Las validaciones a realizar son las siguientes:

- Que no haya campos en blanco.
- Fecha con formato dd/mm/yyyy. Para validar la fecha, se adjunta al final del todo una clase **DateUtils** que puedes añadir en un nuevo paquete **ch.makery.address.util** para organizar mejor el código.
- Código postal numérico. Por ejemplo, se puede validar con `Integer.parseInt`

Si está todo correcto se añadirá una nueva persona al listado. Una posible implementación es crear un controlador para el diálogo en el que se intercambie información con la clase **Main** y así poder transferir el nuevo objeto que se va a añadir al listado asociado a la tabla.



Código de la clase **DateUtil**

```
/**
 * Clase auxiliar para manejar fechas
 *
 */
public class DateUtil {

    /** Patrón empleado para convertir fechas */
    private static final String DATE_PATTERN = "dd/MM/yyyy";

    /** Objeto para formatear fechas */
    private static final DateTimeFormatter DATE_FORMATTER =
        DateTimeFormatter.ofPattern(DATE_PATTERN);

    /** Devuelve la fecha que llega como parámetro como un string
     *
     * @param date es el objeto con la fecha
     * @return String formateado
     */
    public static String format(LocalDate date) {
        if (date == null) {
            return null;
        }
        return DATE_FORMATTER.format(date);
    }

    /**
     * Convierte un String con un formato de fecha en un objeto LocalDate
     * Devuelve null si el formato es incorrecto y no se puede realizar la
     * conversión
     *
     * @param dateString la cadena de texto con la fecha
     * @return devuelve el objeto LocalDate o null en caso de error
     */
    public static LocalDate parse(String dateString) {
        try {
            return DATE_FORMATTER.parse(dateString, LocalDate::from);
        } catch (DateTimeParseException e) {
            return null;
        }
    }

    /**
     * Comprueba si un String es una fecha válida
     *
     * @param dateString
     * @return true si la fecha es válida o false en caso contrario
     */
    public static boolean validDate(String dateString) {
        // Try to parse the String.
        return DateUtil.parse(dateString) != null;
    }
}
```