

# **Programación 1º DAM**

## **Proyecto Final**

Memoria

I.E.S. San Vicente  
San Vicente del Raspeig (Alicante)  
Curso 2018/2019

Alumno:  
Jorge Calzada

Profesor:  
Nacho Cabanes

# 1. Introducción

## Nombre del proyecto

Diary

## Desarrollado por

Jorge Calzada Asenjo

## Github del proyecto

<https://github.com/JorgeCalzadaAsenjo/Diary>

## Descripción breve del proyecto

Se trata de una agenda, la cual incorpora agenda de contactos, calendario, notas y recordatorios. Es una aplicación con interfaz gráfica, creada en lenguaje C# usando la interfaz de programación de aplicación gráfica que se incluye como parte de Microsoft .NET, Windows Forms.

# 2. Funcionalidad del proyecto

Al entrar al programa, aparecerá un pantalla de bienvenida, con un menu para entrar a:

- Agenda de contactos.
- Calendario.
- Notas.
- Recordatorios.
- Ajustes.
- Ayuda.

Si se escoge Agenda de contactos accederá a un menu donde podrá registrar nuevos contactos (nombre, telefono, email, cumpleaños, etc), buscar contactos existentes por algún campo, visualizar la lista de contactos que se tenga actualmente, etc.

En la opción Calendario se accederá a una panatalla en la que se visualizará el calendario del año actual, marcando el dia de hoy, y en la parte inferior una serie de opciones que permitirán introducir nuevos eventos, visualizar eventos de un tipo, crear “Calendarios” que permitan separar categorias de eventos (trabajo, personal, etc), buscar una serie de eventos, etc.

En la opción Notas accederá a una lista con las notas ya creadas, se podrá añadir nuevas notas o consultar y editar notas que ya existiera.

En recordatorios se podrá visualizar y crear nuevos tareas que el usuario deberá realizar. Habra dos tipos principales de recordatorios, los que el usuario debe realizar pero sin fecha o los programados o que el usuario debe realizar antes de una fecha.

En los ajustes se podrá modificar algunas características de la aplicación, como puede ser el idioma, exportar o importar datos, y algunas otras opciones.

Todos los datos se guardarán de forma local, asegurando que al salir de la aplicación no se pierdan y el usuario pueda seguir consultándolos al volver a entrar.

### 3. Prototipo de la pantalla

La apariencia que se persigue es ésta:

2018						
Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio
do lu ma mi ju vi sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	do lu ma mi ju vi sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28	do lu ma mi ju vi sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	do lu ma mi ju vi sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	do lu ma mi ju vi sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31	do lu ma mi ju vi sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	do lu ma mi ju vi sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

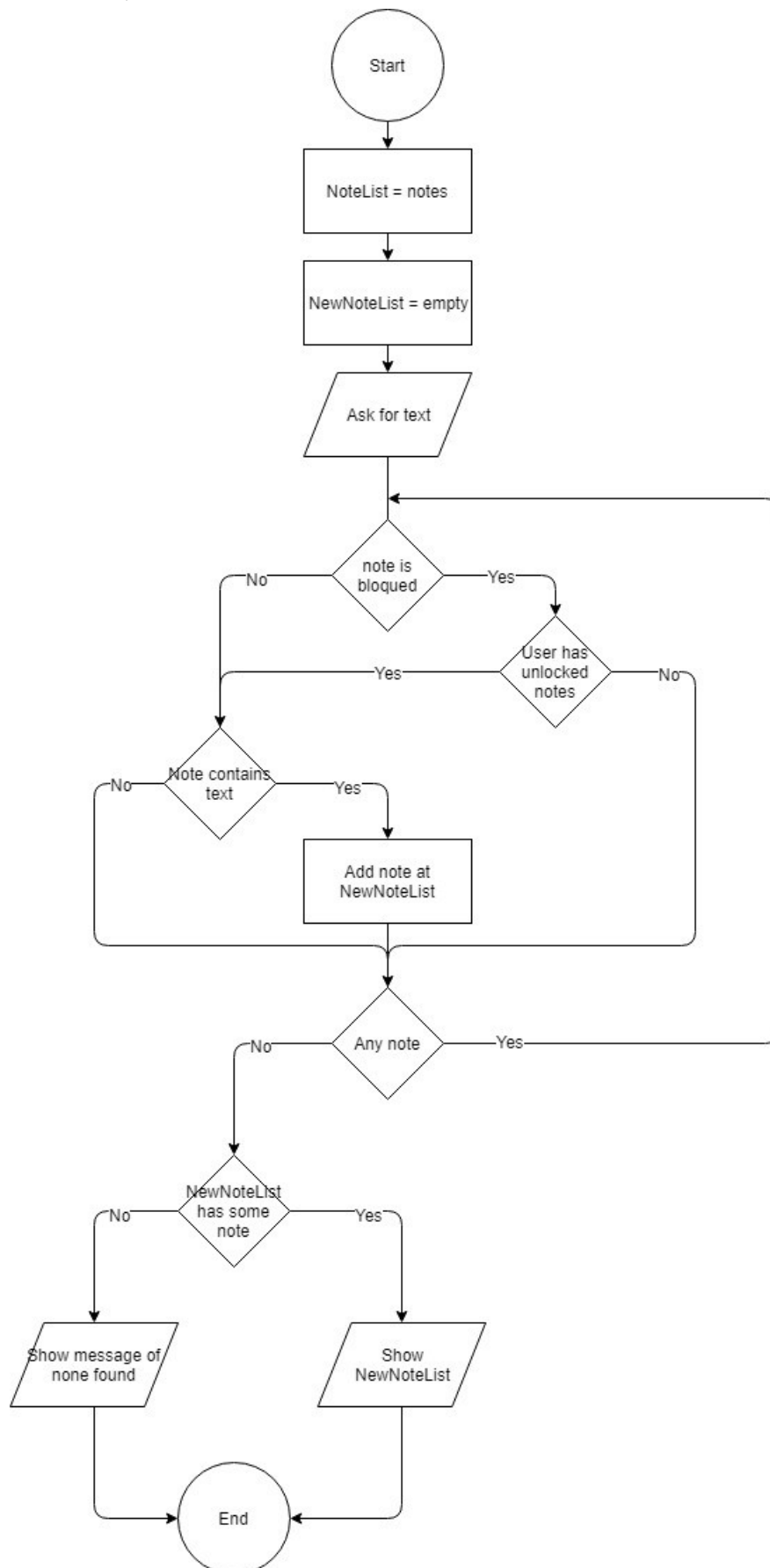


### 4. Entregas previstas

1. Creación del esqueleto del programa. Pantalla de menú simple.
2. Creación, visualización, eliminación, búsqueda de contactos (por varios campos, parcial o completa), ordenado. Los contactos tendrán varios campos como nombre, telefono, cumpleaños, trabajo, etc (solo nombre y telefono serán obligatorias) (no interactivo, solo clase preparada para implementar).
3. Creación, visualización, eliminación, búsqueda de eventos. Los eventos tendrán varios campos como título, tiempo, repeticiones, notas, etc (solo titulo y tiempo serán obligatorias) (no interactivo, solo clase preparada para implementar).
4. Creación, visualización, eliminación, búsqueda de notas. Las notas tendrán varios campos como título, texto, ultima fecha, fecha creación, etc (solo texto será obligatoria y las fechas se rellenarán automáticamente) (no interactivo, solo clase preparada para implementar).
5. Creación, visualización, eliminación, búsqueda de recordatorios. Los recordatorios tendrán varios campos como título, texto, fecha, completar o no, etc (solo titulo será obligatoria (no interactivo, solo clase preparada para implementar). Recordatorios en dos listas (recordatorios y programados)
6. Incorporar todas las clases creadas a las vistas correspondientes.
7. Guardar datos en ficheros y cargar al entrar.
8. Permitir importar y exportar datos desde diferentes ficheros (csv, xml, json)
9. Incorporar a la aplicación seguridad por medio de usuario y contraseña que permita que la aplicación tenga varios usuarios.
10. Proteger notas con contraseña y verificar al acceder que la contraseña sea correcta. Al añadir un contacto con cumpleaños añadir al calendario.
11. En recordatorios mostrar un contador con tareas realizadas y por realizar. Además, poder marcar tareas realizadas y en los ajustes de la aplicación decidir si los eventos completos se borran, se esconden o nada (solo quedna marcados) o preguntar por cada evento.
12. Buscar contactos duplicados, mostrar avisos del sistema para los recordatorios.
13. Pantalla de ajustes de la aplicación (idiomas, colores), guardar ajustes en fichero.
14. Permitir separar los contactos, notas, recordatorios y eventos en categorias o carpetas (instituto, trabajo, personal, amigos, etc).
- 15.

#### **4b. Diagrama de clases**

#### 4c. Diagrama de flujo/ de cajas (Busqueda de notas)



## 5. Trabajo diario realizado

- **17-abr-2019:** Esqueleto del proyecto. Se muestra una pantalla de menú simple.
- **6-may-2019:** Se muestra el menú de contactos, añade y elimina.
- **13-may-2019:** Búsqueda de contactos y retoque de un fallo del menú
- **15-may-2019:** Rehecho parte del esqueleto, menú y clase contactos (adaptación a Windows Forms). Implementados los métodos y atributos que forma la clase calendario
- **16-may-2019:** Creados métodos y atributos de la clase Notas y Nota.
- **20-may-2019:** Preparada la lectura y escritura en ficheros de datos (solo en clase notas, pero el resto será idéntico)
- **22-may-2019:** Terminado cargado, guardado, creación, edición, borrado de notas.
- **23-may-2019:** Clase Reminder y Reminders. Perfeccionando algunos métodos y añadido carga y guardado de ficheros a todas las clases.
- **24-may-2019:** -
- **27-may-2019:** Añadida posibilidad de traducción de la aplicación usando diccionarios.
- **29-may-2019:** -
- **30-may-2019:** -
- **31-may-2019:** -

### 5b. Estructuras utilizadas

#### Obligatorias

- if
- else
- Conectores: && y/o || y/o !
- switch
- ?
- while
- for
- foreach
- try-catch
- (arrays)
- struct
- (clases + herencia)
- (propiedades o getters y setters)
- public, protected, (opcional) private
- ArrayList o List<>
- Hashtable o SortedList o Dictionary
- StreamReader o FileStream o BinaryReader
- ref o out
- manejo avanzado de cadenas: substring, contains, split, replace o similares
- consola avanzada o SDL o Windows Forms o Unity

#### Opcionales

- Operaciones a nivel de bits
- Constantes

- Enumeraciones
- Interfaces, clases selladas
- Manejo de fechas
- Manejo de directorios
- Información del sistema
- LINQ
- Expresiones regulares
- Serialización
- Conexión a bases de datos

## **6. Problemas encontrados durante el desarrollo y sus soluciones**

- Problema para leer texto del usuario con SDL

## **7. Mejoras o restricciones respecto a la idea inicial**

- Se ha adaptado la aplicación a Windows Forms

## **8. Capturas de pantalla del proyecto final**

## **9. Código fuente del proyecto final**