

MANUAL DE PROGRAMADOR

Sistema de generación de reporte de hallazgos



Versión 1.0.0

Tabla de Contenido

1. Introducción.....	2
2. Autores	2
3. Lenguaje	2
4. Librerías utilizadas	2
5. Funcionamiento de la aplicación.....	2
5.1 Módulo de captura de datos.....	2
5.2 Módulo de administración	2
5.3 Formulario “formAdministrarPuestos”	3
5.4 Formulario “formHistorial”	3
5.5 Formulario “formCapturaDeDatos”	3
5.6 Paths	4
6. IDE Utilizado	4
7. CONOCIMIENTOS REQUERIDOS DEL PROGRAMADOR	4

1. Introducción.

En el presente documento se realiza un detalle técnico del funcionamiento del programa de reporte de hallazgos.

2. Autores

Proveedora Mexicana de Monofilamentos S.A. de C.V

Kevin Nataniel Rosas Pérez (Programador de la aplicación)

3. Lenguaje

Se realizó un desarrollo orientado a objetos por medio de C# utilizando el framework 3.5 client Profile debido a que algunas de las computadoras de destino contienen el sistema operativo Windows XP.

4. Librerías utilizadas

Se utilizó Microsoft.Office.Interop.Excel.dll para la búsqueda de información en la base de datos de Excel. También fue utilizada para la apertura de los archivos de excel.

5. Funcionamiento de la aplicación

El sistema cuenta con un módulo de administrador para gestionar algunos atributos. También cuenta con un módulo de captura de datos para generar el archivo de Excel y mandarlo a imprimir. En las siguientes secciones se detallará el funcionamiento de estas dos funcionalidades principales.

5.1 Módulo de captura de datos

Este módulo toma datos por medio de una interfaz gráfica y lo guarda directamente dentro de un objeto instanciado de la clase llamada “classHistorial”.

Una vez que los datos del objeto fueron guardados, se corre un query dentro del mismo objeto para introducir sus datos en la base de datos “dbReporteDeHallazgos.accdb”.

Luego el objeto copia todos sus atributos en una hoja de Excel y manda una vista previa de impresión en la pantalla.

5.2 Módulo de administración

El módulo de administración está separado únicamente en dos formularios diferentes:

- formAdministrarPuestos
- formHistorial

5.3 Formulario “formAdministrarPuestos”

El primer formulario, llamado “formAdministrarPuestos”, contiene un DataGridView que muestra una lista de atributos leídos desde un archivo de texto. Este archivo es actualizado con la información contenida en el DataGridView. Este formulario es utilizado para actualizar los archivos de:

\\192.168.1.244\hallazgos\ayudantes.txt

\\192.168.1.244\hallazgos\enterado.txt

\\192.168.1.244\hallazgos\identificadoPor.txt

\\192.168.1.244\hallazgos\inspectores.txt

\\192.168.1.244\hallazgos\responsables.txt

\\192.168.1.244\hallazgos\operadores.txt

\\192.168.1.244\hallazgos\supervisores.txt

Este formulario toma por medio del constructor un dato de tipo string indicando el archivo de texto que mostrará en pantalla.

5.4 Formulario “formHistorial”

El formulario de historial utiliza un query “Select * from dbReporteDeHallazgos.t_historial” para tomar todos los datos de la tabla de historial de impresiones y lo muestra en un dataGridView. Posteriormente se colocó un botón para realizar la exportación.

Esta exportación a grandes rasgos realiza lo siguiente:

- Pregunta la ruta de guardado del archivo por medio de un openFileDialog.
- Selecciona los datos contenidos en el DataGridView y los copia en el clipBoard.
- Abre un archivo de excel y copia todo lo contenido en el clipboard.
- Invoca una función de formateo (“FormatearHoja()”) para colocar los encabezados con un formato resaltado y colocar bordes en todas las celdas del historial.
- Guarda el archivo en la ruta especificada por el usuario.
- Deja el archivo abierto para que el usuario pueda visualizar el historial ya exportado.

5.5 Formulario “formCapturaDeDatos”

Este formulario sirve para obtener los datos de entrada para el formulario del reporte. Cabe mencionar que en el momento que se corre la aplicación, se carga por completo el contenido de Especificaciones3 en un DataTable de C# para agilizar las búsquedas.

5.6 Paths

Las direcciones de los archivos utilizados para la configuración y funcionamiento de la aplicación están guardados en rutas predefinidas a partir de otro archivo de texto. En este caso el archivo paths estará ubicado en la misma carpeta del archivo ejecutable y contendrá la siguiente información:

```
\\192.168.1.244\hallazgos\ESPECIFICACIONES3.xls  
\\192.168.1.244\hallazgos\plantillaReporteHallazgos.xls  
\\192.168.1.244\hallazgos\ayudantes.txt  
\\192.168.1.244\hallazgos\enterado.txt  
\\192.168.1.244\hallazgos\identificadoPor.txt  
\\192.168.1.244\hallazgos\inspectores.txt  
\\192.168.1.244\hallazgos\responsables.txt  
\\192.168.1.244\hallazgos\operadores.txt  
\\192.168.1.244\hallazgos\supervisores.txt  
\\192.168.1.244\hallazgos\dbReporteDeHallazgos.accdb
```

Cada una de las líneas debe de colocarse en el mismo orden mostrado arriba; de lo contrario el programa tomará de manera equivocada la configuración.

6. IDE Utilizado

Este proyecto se realizó completamente en Visual Studio 2013, el cual es, sin duda alguna, el mejor IDE para proyectos en C# con Windows Forms y WPF. Para obtener una copia de la solución de Visual Studio del proyecto (.sln), favor de consultar con Aseguramiento de Calidad.

7. CONOCIMIENTOS REQUERIDOS DEL PROGRAMADOR

Para que un programador pueda hacer modificaciones o actualizaciones al código de este programa, será indispensable que domine los siguientes conceptos de programación:

1. Programación orientada a objetos.
2. Programación en un entorno Windows Forms o WPF
3. Referencias a librerías externas
4. Manejo de bases de datos (lenguaje SQL)
5. Programación de eventos en C# (la gran mayoría de los eventos utilizados están definidos en el framework .NET)
6. Uso de herramientas de debugging
7. Conocimiento para generación de archivos de Excel por medio de C#.

Se recomienda que la edición al código se haga directamente en Visual Studio, ya que sus herramientas de debugging son las mejores para proyectos de C#.