

***Universidad Nacional del Sur***

***Tecnicatura Universitaria en Sistemas Electrónicos  
Industriales Inteligentes***

# **Principios de Computadoras I**

## **Proyecto Final**

Profesor: Andrés Repetto.

Integrantes de la comisión:  
López Leandro J. - Rodríguez Ezequiel

FECHA: 22/09/2017

## Requerimientos que se nos solicitó:

“Tema 1: Se solicita implementar un programa con ventanas que cumpla los siguientes requerimientos:

- 1) Abrir y mostrar una imagen.
- 2) Generar un recorrido con 40 puntos (x, y) que se encuentren en el rango del tamaño de la imagen.
- 3) Guardar el archivo de recorrido generado.
- 4) Se deberá crear un historial de las imágenes cargadas y de los recorridos generados para cada una de ellas.
- 5) El usuario podrá elegir visualizar el recorrido sobre cualquier imagen generada.”

## Descripción de la Aplicación Desarrollada:


La aplicación desarrollada en el lenguaje Delphi, permite abrir diferentes imágenes de mapa de bits (solo con formato BMP), a las cuales se les puede asociar a cada imagen distintos recorridos / caminos, representados éstos por puntos en el plano coordenado. El usuario tiene la posibilidad de poder visualizar los recorridos sobre las imágenes cargadas, para lo cual se toman los puntos ingresados para poder dibujar distintas líneas sobre la imagen. El usuario podrá luego exportar la nueva imagen generada hacia un nuevo archivo de mapa de bits.

El programa irá creando automáticamente un historial con las últimas imágenes abiertas el cual se mantendrá hasta que se cierra la aplicación.

Si el usuario desea conservar algún recorrido creado para cierta imagen, está la posibilidad de guardarlo en un archivo binario, el cual podrá ser recuperado en cualquier momento posterior.

## Funcionamiento de la Aplicación

A continuación se explicará detalladamente cada característica y cada funcionalidad de la aplicación, seguido de una breve explicación de como se planteó la solución para ello.

Un botón rectangular con un borde gris y el texto "Abrir Imagen" en el centro.

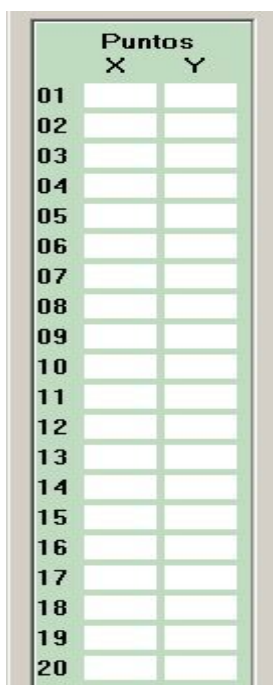
Se puede abrir una imagen con formato BMP desde :

*Para ésta funcionalidad utilizamos un cuadro de diálogo ya predefinido por Delphi (TOpenPictureDialog) el cual permite seleccionar solo archivos con extensión .BMP y visualizar una imagen en miniatura del mismo.*

*Al momento de cargar la imagen y de ubicarla en la pantalla principal de la aplicación obtenemos su tamaño (ancho y alto) en píxeles y las guardamos en dos variables globales para luego hacer control sobre los puntos a ingresar.*

Las líneas a dibujar sobre la imagen estarán definidas por puntos (píxeles en la imagen). El origen de los mismos está ubicado en el vértice superior derecho de la imagen. El punto 0 (X=0, Y=0) se toma como punto inválido y al mismo tiempo como un separador (el mismo sirve por si se quiere dibujar más de un camino sobre la misma imagen).

Desde las dos columnas fijadas a cada lado de la imagen cargada el usuario podrá visualizar y/o ingresar los puntos que desee (siempre dentro del tamaño de la imagen).

Una interfaz de usuario que muestra una lista de 20 puntos, numerados del 01 al 20. Cada punto tiene dos columnas adyacentes para ingresar las coordenadas X e Y. El encabezado de la tabla es "Puntos" con subencabezados "X" y "Y".

Puntos	X	Y
01		
02		
03		
04		
05		
06		
07		
08		
09		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		

Al presionar “ingresar puntos” la aplicación recorrerá los campos de edición de los 40 puntos verificando si son válidos (si es un punto inválido el campo se pondrá en color rojo). Para que un punto sea válido debe estar entre los límites del tamaño de la imagen y no ser 0 en ninguna de sus coordenadas. No es necesario que el usuario ingrese todos los puntos, puede ingresar solo los que necesite, incluso puede utilizar cualquier número de punto, solo hay que tener en cuenta que para dibujar una línea se necesitan ingresar dos puntos en orden consecutivos. El botón “Puntos Random” generará los 40 puntos aleatoriamente, los cuales si son dibujados en la imagen generarán recorridos sin sentido alguno.

Un botón rectangular con un borde gris y el texto "Ingresar Puntos" en el centro.Un botón rectangular con un borde gris y el texto "Puntos Random" en el centro.

*Para representar los puntos se utilizó un arreglo donde cada componente es un registro que contiene dos campos (X e Y). A este arreglo lo acompaña una variable booleana para saber si los puntos fueron ya cargados o no, y se lo asocia a una imagen (se guarda su dirección física completa para volver a cargarla cuando sea necesario).*



El botón “Tamaño Real” tiene varias funcionalidades.

Por un lado permite visualizar la imagen cargada previamente en una nueva ventana y en su tamaño original. Esta mostrará también el recorrido si es que ya fueron cargados puntos y dibujados los mismos con anterioridad.

Por otro lado, al hacer un click izquierdo con el mouse se podrá saber la posición exacta del mismo dentro de la imagen. Esta función es muy útil si ya hemos dibujado algún camino y queremos saber por los puntos que pasa el mismo.

**X: 150, Y: 170**

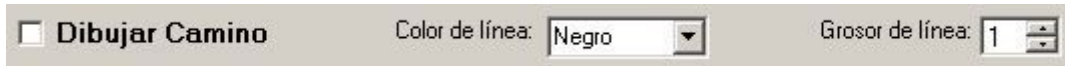
Además haciendo un click derecho tendremos la opción de ir ingresando distintos puntos exactamente en la posición en la cual hayamos hecho el click. Para esto, en cada punto que ingresemos se nos mostrará una ventana mostrándonos las coordenadas del punto y el número del punto a ingresar. Este último el usuario podrá modificar a su necesidad, pero debe tener en cuenta que puede sobrecribir puntos ingresados anteriormente.

Al ingresar al menos dos puntos de ésta forma, las líneas se irán dibujando automáticamente en la ventana de tamaño real. Luego al cerrar ésta ventana los puntos ingresados se actualizarán en la pantalla principal.

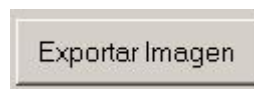
*Para ésta funcionalidad tuvimos que utilizar un nuevo Form el cual lo creamos en el mismo momento en que se acciona el botón. Luego de crearlo le pasamos la imagen cargada en un objeto TImage en la pantalla principal a un objeto TImage de la nueva ventana. También le tenemos que pasar otros parámetros (mediante un procedimiento en la unidad asociada al nuevo Form), como por ejemplo el puntero a la estructura de datos de la imagen actual (junto con sus puntos asociados).*


La opción “Dibujar Camino” permite visualizar el/los recorrido/s generados por los puntos. Cabe aclarar que la imagen original no se verá afectada por esto, solo se modifica la imagen cargada en la aplicación.

Como opción extra se puede elegir el color de las líneas y grosor de las mismas.



*Para realizar ésta función utilizamos la propiedad Canvas de Delphi, que permite dibujar líneas en los mapas de bits. El procedimiento para dibujar hace un recorrido de los 40 puntos y grafica líneas entre dos puntos válidos y consecutivos.*



Mediante el botón  el usuario podrá guardar la imagen actual, junto con sus caminos dibujados si los hubiera. La misma se guardará en formato BMP.

*Para ésta funcionalidad utilizamos un cuadro de diálogo ya predefinido por Delphi (TSaveDialog) el cual permite que el usuario seleccione una ubicación y el nombre del archivo a guardar.*

El botón “Cargar Recorrido” permite cargar un recorrido previamente guardado con el botón “Guardar Recorrido”.

Hay que tener en cuenta que el recorrido cargado va a estar asociado a la imagen cargada inicialmente en la aplicación y para la cual se ingresaron los puntos; por lo cual, si dicha imagen original no existe (se eliminó, se cambió de nombre o de lugar en el disco, etc), no se podrá visualizar la imagen y el programa mostrará una advertencia en este caso. De todas formas los puntos igualmente serán cargados para que el usuario tenga alguna referencia si lo necesita.



*Para ambos casos se utiliza un cuadro de diálogo ya predefinido el cual permite cargar y guardar archivos con extensión .REC que el programa puede interpretar (se guarda la estructura de datos de cada recorrido).*

Para visualizar el historial de las imágenes cargadas previamente hay que utilizar la lista desplegable en la parte superior de la aplicación:



El mismo muestra la dirección completa de las imágenes, y es un historial temporal, o sea que mantendrá la lista hasta cerrar el programa.

Cada vez que el usuario abre una imagen o incluso carga un recorrido previamente guardado, se van agregando a ésta lista en el tope de la misma, es decir lo último abierto / cargado estará en la parte superior de la lista, mientras que lo más antiguo estará en el fondo de la misma.

Al seleccionar un ítem de la lista, se volverá a mostrar la imagen y sus puntos asociados (si fueron cargados previamente), quedando esta como la imagen actual y pudiendo el usuario seguir trabajando con ella.

*Para realizar esta lista temporal primero pensamos en utilizar un arreglo donde cada componente sería toda la estructura de datos que utilizamos (dirección de la imagen y puntos), pero no sabíamos qué tamaño ponerle, ya que éste no tendría un tamaño fijo, sería variable en cada ejecución. Por eso nos pareció que la mejor opción era construir una lista enlazada con punteros, donde cada elemento lo vamos insertando al principio de la misma. Los ítems de la lista enlazada tienen el mismo orden que los ítems en el ComboBox que utilizamos para mostrar la lista, así a la hora de saber cuál fue el seleccionado la búsqueda es directa.*

## Balance final:

Creemos que hemos cumplido con los requerimientos que nos dieron para este proyecto final. Además hemos agregado otras funcionalidades como que el usuario pueda ingresar los puntos, y además de una manera bastante intuitiva como se puede hacer desde la pantalla “Tamaño Real”, que fue la última característica de la aplicación que agregamos y que tuvo sus complicaciones.

Por otro lado, en la mayor parte de el código hacemos varias chequeos y comprobaciones para evitar un mal manejo del usuario y que el programa “se rompa”. Una de los posibles errores que no manejamos es cuando el usuario pueda ingresar por ejemplo una letra en vez de un número, para lo cual una de las funciones predefinidas que usamos lanza una excepción, la cual no sabemos cómo manejarla y creemos que escapa a los temas de éste curso.

Una de las principales complicaciones que tuvimos, fue el no poder cargar las imágenes en la pantalla principal con su tamaño original dentro del recuadro del TImage; necesitábamos poder incorporar barras de desplazamiento pero no lo pudimos lograr de ninguna forma. El motivo de porque queríamos cargarlas en el tamaño real era el de poder cargar los puntos directamente sobre la imagen, ya que el TImage las acomoda a su tamaño (las agranda o las achica), dificultando esto el poder saber sobre qué punto real de la imagen se estaría verdaderamente haciendo click. Esta situación nos motivó a agregar otra ventana, la de “Tamaño Real”, acomodando el tamaño de toda la ventana al mismo tamaño de la imagen, y luego poder agregar los puntos desde ésta ventana.

El problema descrito y resuelto anteriormente nos trajo una nueva complicación. Al crear una nueva ventana (un nuevo form y su unidad), necesitábamos que éste pudiera acceder a la estructura de datos que habíamos definido en la unidad principal del programa, y así una unidad tenía que usar a la otra y viceversa, pero como Delphi no permite ese tipo de vinculación, entonces la única solución que encontramos fue la de ubicar toda la estructura de datos que utilizamos en una nueva unidad, que sirve solamente para mantener esos tipos de datos y poder utilizarlos desde ambas unidades.

Si bien desarrollamos una lista para manejar un historial temporal, nos quedó pendiente haber podido agregar una lista de archivos abiertos anteriormente (como tienen la mayoría de los programas populares), ya sea que el usuario haya trabajado con ellos o los haya guardado o no, y que ésta lista quede siempre disponible, aún cuando se cierre e inicie nuevamente la aplicación. Para esto hubiéramos necesitado crear algún tipo de archivo que exista siempre y que acompañe al archivo ejecutable de la aplicación, o bien guardarlo en alguna parte del sistema operativo, pero ésto último no sabíamos cómo realizarlo.

Otra función que no incorporamos y que se nos ocurrió a último momento, es la de haber agregado una opción de visualización a pantalla completa en la cual se pudiera hacer algún zoom en la imagen cargada, ir cambiándole el tamaño como el usuario lo requiera.