

Segunda práctica
Funciones Matemáticas

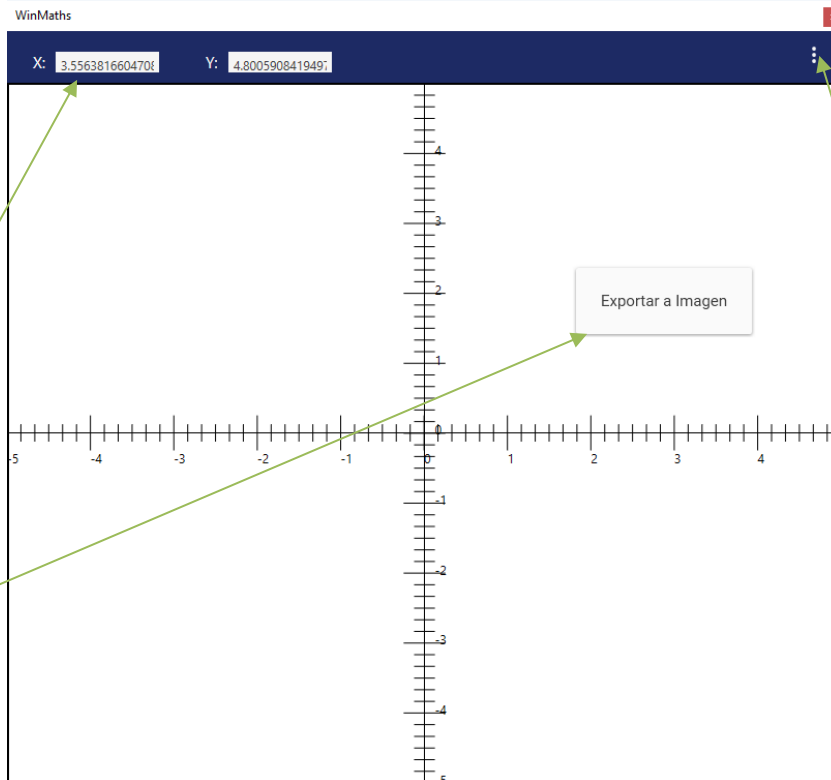
Luis Blázquez Miñambres (i0910465)

1 de Diciembre de 2018

1. Manual de Usuario

La aplicación WinMaths se compone de cuatro ventanas:

- **Menú principal:** aquí se mostrará la representación de las gráficas que se dibujen.



Coordenadas de la Representación: mostrarán las coordenadas sobre las que el puntero se colocará

Opción Exportar Imagen: si se pulsa botón derecho en la representación se abrirá un menú contextual con la opción de exportar a imagen la representación dibujada.

Menu de Control Desplegable: menú que muestra los botones de 'Ajustes' que lanzará la Ventana de ajustes de límites de representación y 'Salir' que permitirá al usuario salir de la aplicación.

- **Panel de Ajustes de Límites:** aquí se ajustarán los límites de representación de las gráficas seleccionando los valores para los cuales se acotarán las representaciones.

- **Preferencias:** este menú incluye dos submenús dentro de él.
 - **Definición de la gráfica:** aquí se añadirán todos los parámetros necesarios para crear y añadir una nueva gráfica a la tabla

Definición

Función: en este submenú se añadirán unade las funciones disponibles y un nombre para la gráfica

Apariencia: en este submenú se seleccionará el color que tendrá la gráfica (Puede dejarse el que está por defecto)

Parámetros: en este submenú se añadirán los valores para los parámetros a, b o c de cada función dependiendo que parámetros contenga la función seleccionada

- **Parámetros generales (Tabla):** aquí se mostrará una tabla con todas las gráficas añadidas y listas para dibujar, así como la información de cada una, la posibilidad de modificar una de ellas o eliminar o dibujar una de ellas o varias a la vez (manteniendo pulsado el botón **CTRL**+seleccionar todas las que se deseen dibujar).

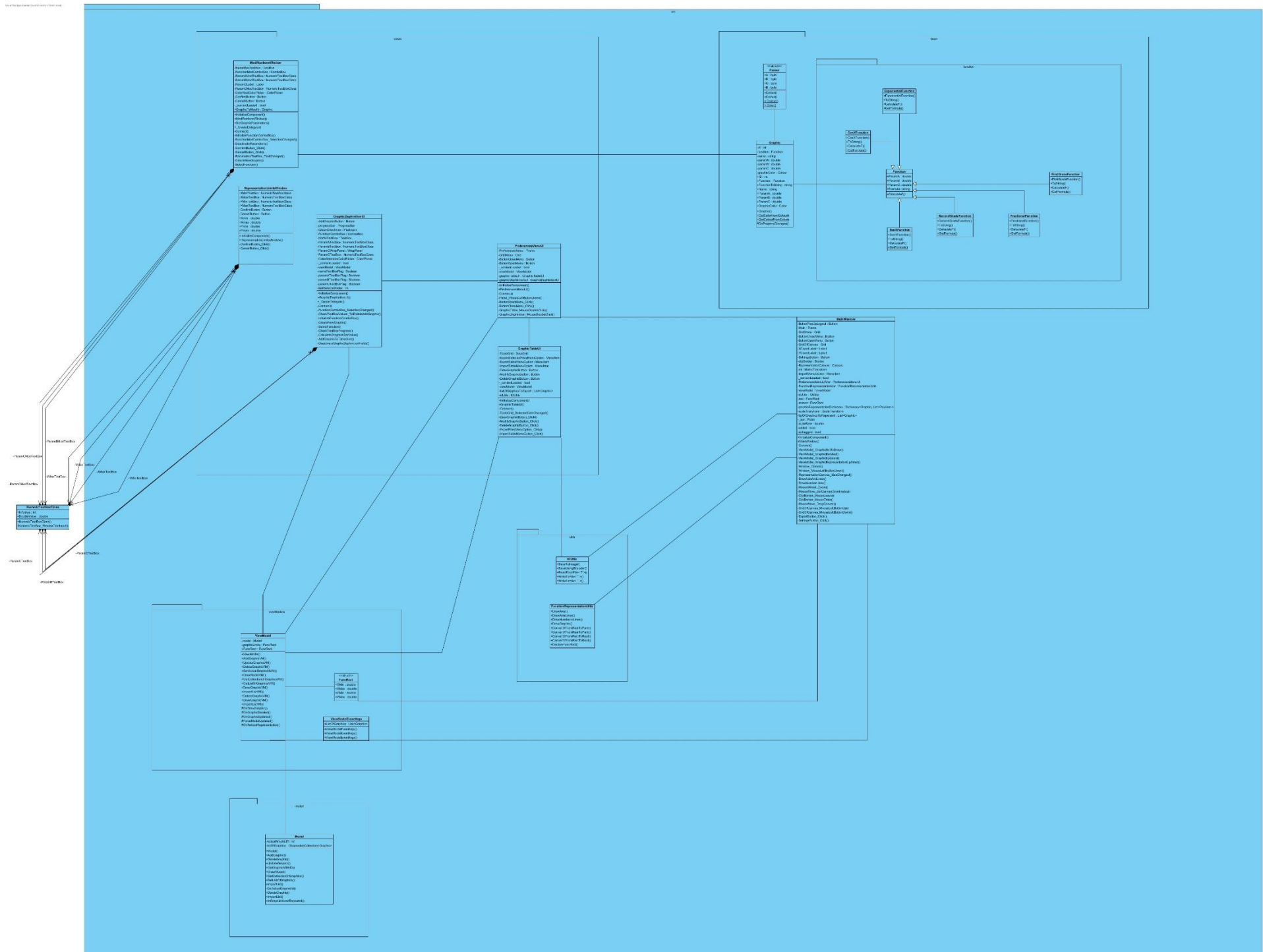
Tabla de gráficas: aquí estarán las graficas que se podrán dibujar modificar, dibujar o modificar. Así como ordenar por orden alfabético pulsando en la cabecera de la columna

Menú Contextual I/O: en este submenú que se desplegará al hacer botón derecho sobre la tabla, se permite al usuario exportar o importar gráficas de fichero a la tabla.

INFORMACIÓN ADICIONAL:

- En las casillas de los parámetros A, B, C sólo podrán introducirse números reales positivos o negativos, cualquier otro carácter será rechazado.
- Las casillas de parámetros A, B, C se activarán según la función seleccionada. Por ejemplo: si seleccionamos la función $a \cdot x^b$, la aplicación solo activará y mostrará las casillas A y B.
- Se podrá hacer zoom sobre la representación de la gráfica ajustando el acercamiento o alejamiento de la vista con la rueda del ratón. También se podrá mover la representación pulsando con el botón izquierdo y arrastrando mientras lo mantenemos pulsado.
- No se podrá exportar ninguna gráfica de la tabla de gráficas hasta que no se añada al menos una gráfica en la tabla del panel '*Preferencias*'.
- No se permitirán gráficas con nombres repetidos con el fin de mantener un orden estricto en la tabla de gráficas así como la coherencia y orden de las graficas dentro de esta.

2. Manual del programador



❖ **Métodos importantes (MainWindow.xaml.cs)**

- ***ViewModel_GraphicSetToDraw:*** éste metodo es llamado cuando se lanza el evento enlazado a cuando el usuario pide dibujar una o más gráficas de la tabla y se van dibujando una a una las gráficas solicitadas que se han pedido de la lista de gráficas. Si alguna de las gráficas que se han solicitado dibujar, ya está dibujada en el Canvas es porque está guardado en un Dictionary que toma como clave un objeto de tipo Graphic y como valor devuelve un array de polilíneas correspondientes al dibujo de esa gráfica, de manera que no se redibuja sino que simplemente se recoge esa polilínea de nuevo. En caso contrario, se aclara la representación del canvas, se dibujan los ejes y las líneas y se llama a un método de la clase '*FunctionRepresentationUtils*' para que dibuje y devuelva un array de colecciones de puntos que forman la polilínea referente a cada una de las gráficas que se dibujarán en el canvas.
- ***ViewModel_GraphicDeleted:*** éste metodo es llamado cuando se lanza el evento enlazado a cuando el usuario elimina una o más gráficas de la table de manera que cuando se eliminan de la table se eliminan de la representación del canvas a través de su referencia en el Dictionary.
- ***ViewModel_GraphicUpdated:*** éste metodo es llamado cuando se lanza el evento enlazado a cuando el usuario modifica una gráfica de la table de manera que cuando se hace efectiva la modificación a partir del panel de modificación en la Ventana de Preferencias, se eliminará la representación anterior (llamando desde el viewModel al evento que se encarga de eliminar una o más gráficas) y se añadirá la nueva representación de la gráfica modificada.
- ***ViewModel_GraphicRepresentationUpdated:*** éste método es distinto al anterior en cuanto a su funcionamiento, ya que éste método es llamado cuando se lanza el evento que modifica los límites de la representación a través de la Ventana '*Modification Window*', de manera que solo se reajustan las gráficas ya dibujadas en el canvas (las que están guardadas en el Dictionary)

❖ **Métodos importantes (PreferencesMenuUI.xaml.cs)**

- ***GraphicTable_MouseDoubleClick* y *GraphicDephinition_MouseDoubleClick:*** cargan el page que contendrá la interfaz para la definición de los parametros de la función y el page que contendrá la table de gráficas.

❖ **Métodos importantes (GraphicDephinitionUI.xaml.cs)**

- ***CheckTextBoxValues_ToEnableAddGraphic :*** método a través del cuál se llama a calcular el valor de ese campo en la barra de progreso y se va comprobando cada vez que el usuario escribe sobre cada uno de los TextBox si se cumplen las condiciones necesarias para habilitar el botón '*Añadir gráfica*' y permitir al usuario añadir una gráfica al DataGridView. Este método está controlado por el evento de acción de los TextBox (*TextChanged*).

❖ **Métodos importantes (GraphicTableUI.xaml.cs)**

- ***ImportTableMenuOption_Click***: éste método es llamado cuando el usuario pide importar a la tabla un fichero con las extensiones válidas que contiene una serie de gráficas y las guarda en el observableCollection del modelo.
- ***ExportFilesMenuOption_Click***: éste método es llamado cuando el usuario pide exportar , diferenciando cuando el usuario desea exportar unicamente las filas que ha seleccionado de la tabla o bien toda la tabla si pulsa la opción del menú contextual para exportar todas, a las extensiones permitidas a través de la ventana de guardado (XML, Binario, HTML,...)
- ***ExportTableExcelMenuOption_Click***: éste método es llamado cuando el usuario pide exportar la tabla a un documento Excel, de manera que al pulsarlo se abrirá la aplicación Excel instalada en el ordenador mostrando las graficas encolumnadas en un nuevo archivo .xls. (**NOTA:** al estar cargando una aplicación de escritorio como Excel desde esta aplicación puede haber un ligero retardo al cargar el archivo .xls junto con la aplicación Excel).
- ***ModifyGraphicButton_Click***: éste método es llamado cuando el usuario pide modificar sólo una de las gráficas a la vez, recogiendo el valor de la gráfica que se desea modificar a través de una propiedad y pasandoselo a la ventana de modificaciones '*ModificationsWindow*' y una vez pulsado el botón de confirmación de dicha ventana se recogerá del mismo modo, a través de una propiedad, y se pasará al Modelo que la insertará en la posición en la que estaba la anterior gráfica.

❖ **Métodos importantes (ModificationsWindow.xaml.cs)**

- ***SetGraphicParameters***: método que será llamado cuando se utilice el set de la propiedad '*GraphicToModify*' y que introduce en los campos correspondientes (TextBox, ComboBox,...) los valores de la gráfica que se va a modificar para poder mostrar al usuario los cambios que quiere realizar de esa gráfica antes de hacerlos efectivos.

❖ **Métodos importantes (FunctionRepresentationUtils.cs)**

- ***DrawGraphic***: éste método devuelve un array de polilienas que se ha creado pasando la gráfica que se desea representar, las dimensiones del Canvas (alto y ancho) y los valores genéricos de representación a través del struct de tipo FuncRect. Aquí hay dos casos a distinguir:
 - Por un lado, si se van a representar funciones cuya formula sea de tipo exponencial con exponente negativo o fraccionaria, está se dividirá en dos polilíneas de manera que así evitamos que se dibuje una línea vertical del mismo color que la gráfica en el centro de los ejes.
 - Por otro lado, si se va a representar cualquier otra función se calcularán los puntos reales y a partir de ellos y las formulas proporcionadas en los apuntes se conseguirán los puntos de pantalla que serán los utilizados en la representación de las polilíneas.

❖ **Métodos importantes (IOUtils.cs)**

- **WriteToExcel:** este método es llamado desde la tabla de gráficas cuando el usuario desea exportar la tabla de gráficas a una aplicación Excel , de manera que internamente en este método se creará una instancia de la clase 'ExportToExcel' que lleva tipos parametrizados o tipos genéricos de manera que se puede exportar cualquier objeto de C# a dicho formato. En esta aplicación, se exportarán objetos de tipo 'Graphic', de manera que a esta clase se le pasarán los datos a mostrar en el documento , así como se llamará a un método de esta clase (**GenerateReport**) que generará las filas, columnas, estilo, margen y demás parámetros para llevar a cabo el lanzamiento de la aplicación con los datos correspondientes.
- **WriteToFile:** este método es llamado desde la tabla de gráficas cuando el usuario desea exportar uno ,varias o todas las gráficas de la tabla de gráficas a un archivo (con extensión .bin, .html, .xml,...) de manera que el contenido del archivo exportado estará en binario o cifrado, de manera que hace imposible su edición y su uso se limita exclusivamente a la importación o exportación desde esta aplicación u otras que permitan recoger la información de gráficas de representación o que utilicen su contenido para fines matemáticos. Utiliza tipos parametrizados o tipos genéricos <T> de manera que se puede escribir cualquier tipo de objeto C# a estos ficheros. Su homólogo, 'ReadToFile' aplica la misma configuración que este método pero para recoger información de este tipo de ficheros exportados pero cuando el usuario desea importarlos para añadirlos a la tabla de gráficas.
- **SaveToImage:** este método es llamado desde la ventana 'MainWindow' al hacer click derecho sobre el Canvas y pulsar la opción del menú contextual para exportar el contenido del Canvas o su representación a una imagen. Internamente éste método convierte el Panel Canvas en un objeto de tipo 'RenderBitmap' de manera que aplica un renderizado al contenido de puntos del canvas y lo codifica de manera que pueda aplicarse y exportarse a un formato determinado (ya se haya elegido .png, .gif, .jpeg,...).

❖ **Métodos importantes (Function.cs)**

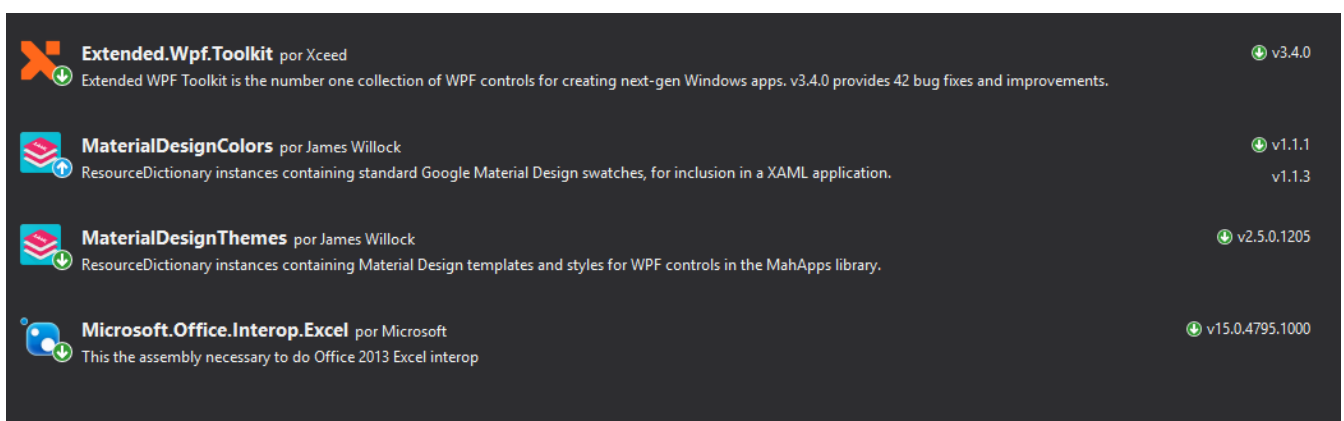
- He creado una clase abstracta Función de la que heredan varias subclases que serán cada una de ellas ,clases concretas u objetos agregados, las funciones matemáticas cuya representación se va a representar en el Canvas. Haber abstraído la implementación de estas subclases permite que cada una de ellas tenga un valor de a ,b y c distintos y su correspondiente fórmula matemática así como un método que cada una de ellas redefine para devolver la fórmula con los datos que ha introducido el usuario (**ToString**) así como otra que devuelve la fórmula que utilizarán los métodos para calcular los puntos de representación de las gráficas (**CalculateF**). Todas ellas serializables para su posterior exportación ya que cada objeto gráfica tendrá un atributo de tipo Función.

3. Bibliografía o referencias

El material que principalmente he utilizado para buscar información sobre las distintas funcionalidades que permite mi aplicación las he encontrado en diversas páginas como repositorios públicos del servicio web GitHub, así como distintos foros de ayuda a programadores como StackOverFlow o AppCoda, entre otros , y la página de documentación de Microsoft [API Microsoft](#) . Como bibliografía principal de libros he utilizado los apuntes de clase sobre WPF.

Bibliotecas usadas (Administración de paquetes NuGet):

- **Extended.Wpf.Toolkit** : lo he utilizado exclusivamente para el uso de uno de los objetos XAML que contiene ‘ColorPicker’.
- **MaterialDesignColors y MaterialDesignThemes**: lo he utilizado para tener diseños más allá de los que ofrece Windows para WPF con respect a las Ventanas y que me ha permitido diseñar el estilo y diseño de la Ventana ‘Preferencias’.
- **Microsoft.Office.Interop.Excel**: lo he utilizado para poder exportar a Excel que es una aplicación externa ofrecida por Microsoft Office, de manera que he conseguido exportar las gráficas de la tabla a un documento de Microsoft Excel lanzando la aplicación directamente con los datos.



- Datagrid y sus características principales_

<https://www.csharpcorner.com/UploadFile/deveshomar/addin-g-checkbox-column-in-datagridview-in-C-Sharp-window-forms/>
<https://stackoverflow.com/questions/4978723/grid-table-in-wpf>
<https://stackoverflow.com/questions/30306385/not-getting-values-from-grid-table-when-checkbox-is-checked>
<https://stackoverflow.com/questions/12491885/how-to-use-datagridcomboboxcolumn>
<https://stackoverflow.com/questions/24334451/unable-to-select-multiple-rows-in-a-wpf-datagrid>
<https://stackoverflow.com/questions/13472321/remove-multiple-items-from-observablecollection-in-silverlight>

<https://stackoverflow.com/questions/35513194/dynamically-hide-a-wpf-tabitem>

- Interfaz, botones y XAML

<https://materialdesignicons.com/>

<https://www.youtube.com/watch?v=nSgd3QNEL3s&list=PLrW43fNmjaQVYF4zgsD0oL9lv6u23PI6M&index=43>

[https://www.youtube.com/redirect?event=video_description&v=YQ1EJJZBHyE&redir_token=kDpUo-](https://www.youtube.com/redirect?event=video_description&v=YQ1EJJZBHyE&redir_token=kDpUo-dwDkl44QuHrbQcsD4GH0l8MTU0MjM4Njc5NUAxNTQyMzAwMzk1&q=ht)

[dwDkl44QuHrbQcsD4GH0l8MTU0MjM4Njc5NUAxNTQyMzAwMzk1&q=ht](https://www.youtube.com/watch?v=YQ1EJJZBHyE)
<tp%3A%2F%2Fmaterialdesigninxml.net%2F>

<https://www.youtube.com/watch?v=YQ1EJJZBHyE>

<https://patconroy.wordpress.com/2009/01/02/restyling-all-wpf-controls/>

<https://stackoverflow.com/questions/29872663/making-a-progress-bar-update-in-real-time-in-wpf>

<https://stackoverflow.com/questions/45558011/chow-to-change-button-icon-in-mvvm>

<http://labloguera.net/javier/2011/09/how-to-move-a-wpf-window-without-window-style/>

<https://stackoverflow.com/questions/37442024/how-to-avoid-repeated-xaml-code>

<https://docs.microsoft.com/en-us/dotnet/csharp/codedoc>

<https://stackoverflow.com/questions/46076559/xamarin-xaml-issue-inheriting-type-not-found-in-xmlns-namespace>

- TextBox

<https://stackoverflow.com/questions/23266723/wpf-border-thickness-bottom-only?rq=1>

<https://stackoverflow.com/questions/11873378/adding-placeholder-text-to-textbox/11873521>

- Binding

<https://msdn.microsoft.com/en-us/magazine/mt736453.aspx>

- Dictionary

<https://docs.microsoft.com/es-es/dotnet/api/system.collections.generic.dictionary-2?view=netframework-4.7.2>

<https://stackoverflow.com/questions/29797489/how-to-get-observablecollection-from-dictionary-values>

- Eventos de ratón

[https://social.msdn.microsoft.com/Forums/vstudio/en-US/5a970445-663e-434d-9214-](https://social.msdn.microsoft.com/Forums/vstudio/en-US/5a970445-663e-434d-9214-f9593bcb45f1/getting-the-mouse-cursor-to-change-to-a-hand-or-anything-when-mouse-)

<f9593bcb45f1/getting-the-mouse-cursor-to-change-to-a-hand-or-anything-when-mouse->

[is-over-gridsplitter?forum=wpf](#)

- Gestión del ComboBox

<https://stackoverflow.com/questions/25328761/deselect-combobox-item>

<https://stackoverflow.com/questions/11878217/add-items-to-combobox-in->

- Zoom y eventos sobre el canvas

<https://searchwindevelopment.techtarget.com/tip/Panning-and-zooming-in-WPF>

<https://social.msdn.microsoft.com/Forums/en-US/eb6c3604-7ae7-4d47-945a-8fb4d9901d62/panning-canvas?forum=wpf>

https://www.youtube.com/watch?v=Ft_qLAY2dEk

<https://github.com/wieslawsoltes/PanAndZoom/tree/master/src>

https://www.youtube.com/watch?v=5X3yCoH_SSw

- Barra de progreso

<https://www.wpf-tutorial.com/misc-controls/the-progressbar-control/>

- Colores

<https://stackoverflow.com/questions/21961920/converting-brush-to-color>

<https://stackoverflow.com/questions/3315088/how-do-i-serialize-a-system-windows-media-color-object-to-an-srgb-string>

<https://stackoverflow.com/questions/34155234/wpf-toolkit-colorpicker-remove-focus>

- Guardar el contenido de un Canvas en una imagen

<https://blogs.msdn.microsoft.com/kirillosenkov/2009/10/12/saving-images-bmp-png-etc-in-wpfsilverlight/>

<https://stackoverflow.com/questions/8881865/saving-a-wpf-canvas-as-an-image>

Serializar y Exportar

<https://social.msdn.microsoft.com/Forums/vstudio/en-US/b3aa38fb-df9f-4dd0-b3dc-a767f8c8b620/problem-in-serialize-a-class-that-implements-inotifypropertychanged?forum=wpf>

<https://stackoverflow.com/questions/6115721/how-to-save-restore-serializable-object-to-from-file>

- Listas y ArrayList

<https://stackoverflow.com/questions/2369673/how-to-add-list-to-a-list-in-asp-net>

<https://stackoverflow.com/questions/32248024/export-to-excel-from-observablecollection-c-sharp>

- Canvas

<https://social.msdn.microsoft.com/Forums/en-US/cb8e1e12-6ab4-43e4-8f4e-fb8437d857a4/drawing-graphics-text-to-a-wpf-canvas?forum=wpf>