

INTERFACES GRAFICAS DE USUARIO

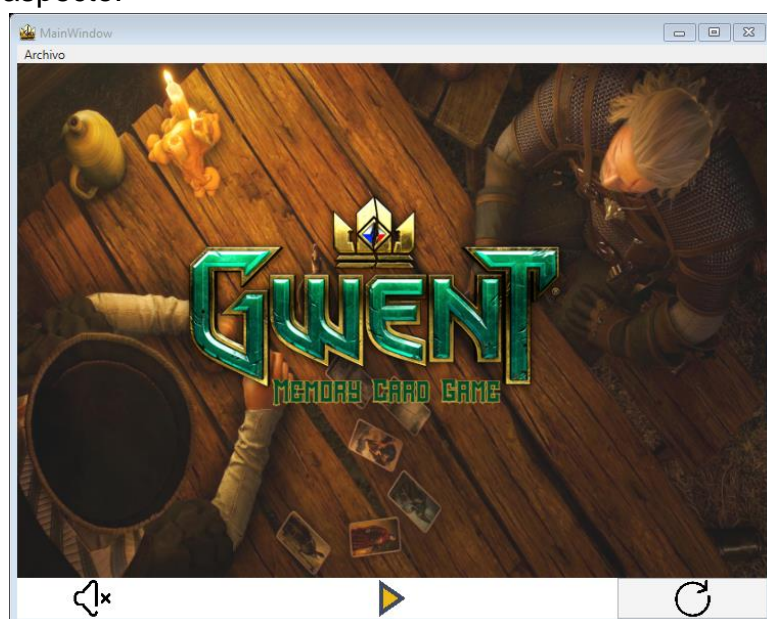
PRACTICA WINDOWS MEMORIA DE LA PRACTICA

Adrián Antonio González Pazos

70922606J

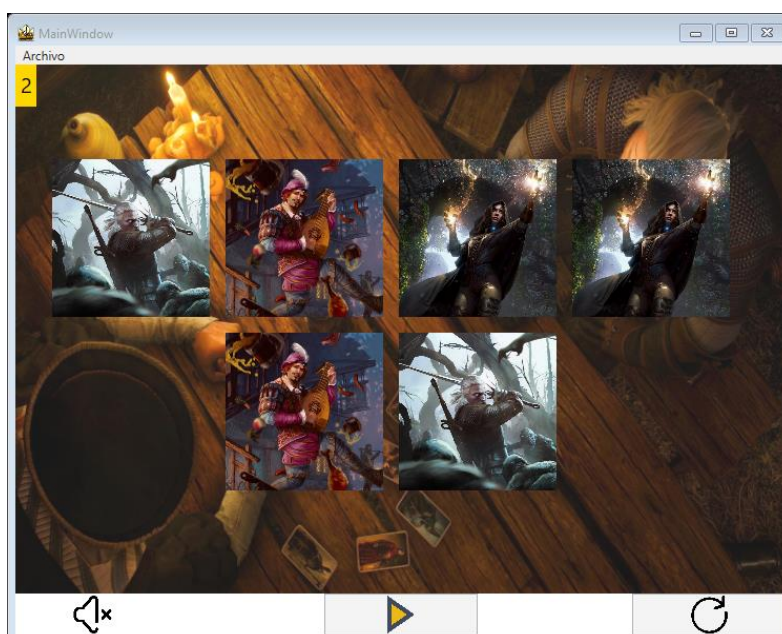
Manual de Usuario

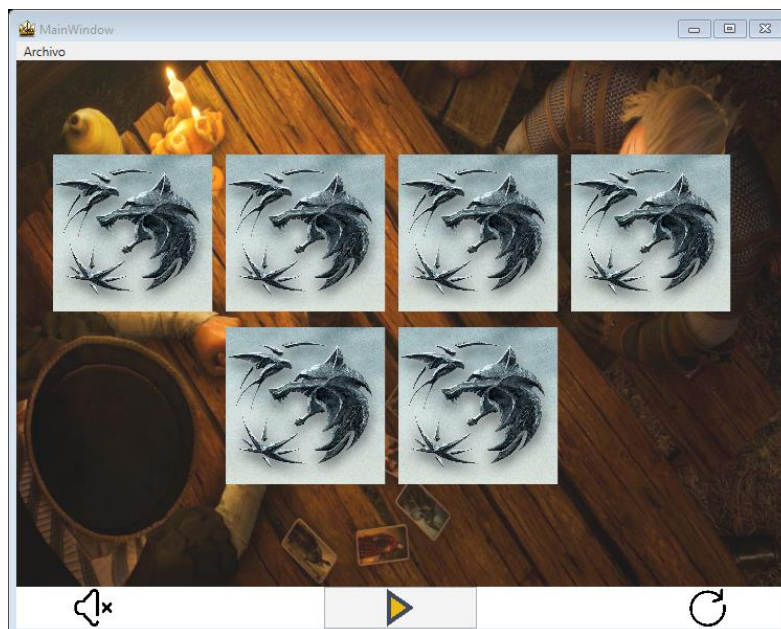
Al arrancar la aplicación se nos mostrará la ventana principal que tendrá el siguiente aspecto:



El botón más a la izquierda, el del icono del altavoz, nos permitirá controlar la música de fondo del juego, si está reproduciéndose música, la parará, en caso contrario, la reproducirá. El botón del medio arranca el juego. Y el botón más a la derecha, de refrescar, permite que el juego vuelva a comenzar en medio de una partida. En “Archivo” nos encontramos dos opciones, “Preferencias” y “Acerca de”, la primera opción nos abre el cuadro de dialogo de las preferencias y la segundo opción abre un panel con información sobre el juego.

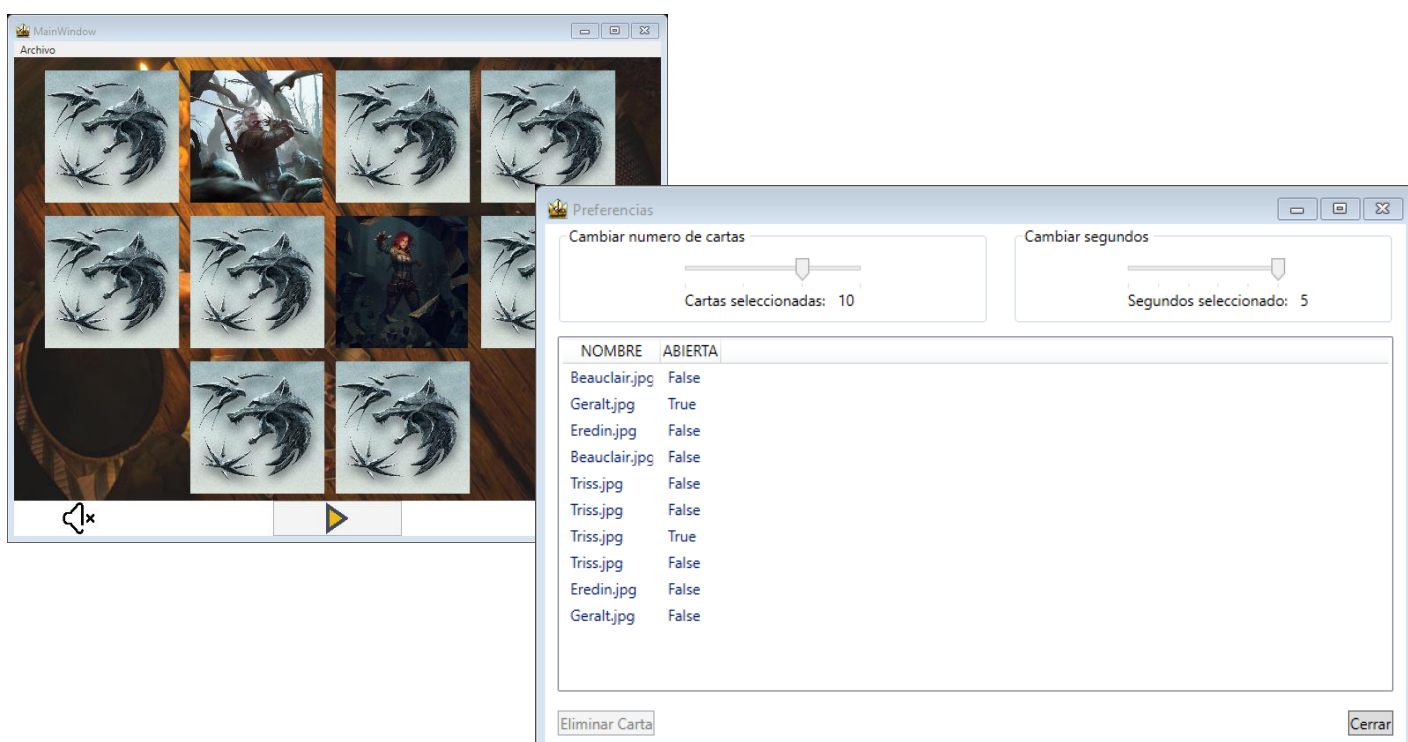
Si no hemos cambiado ningún parámetro en la configuración, al iniciar el juego, nos mostrara 6 cartas, pasado 5 segundos, las cartas se darán la vuelta. Las pantallas antes y después de los segundos serán las siguientes:



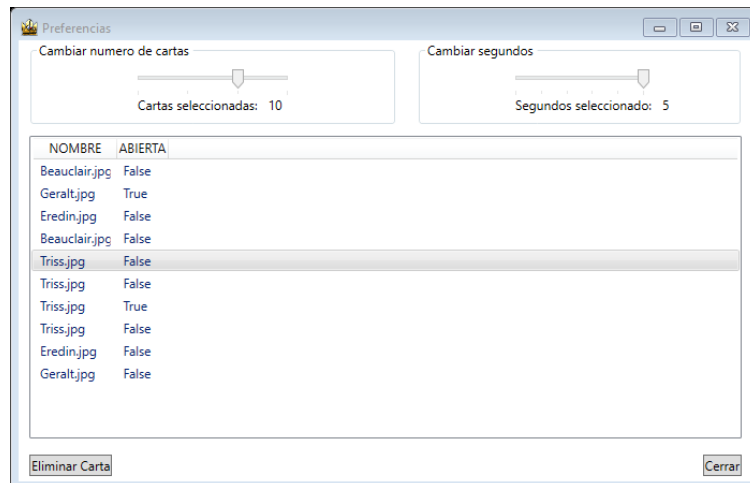


El juego consiste en encontrar la pareja de cada carta. Para ello al hacer click en una, la carta se abre, se da la vuelta, permitiéndote darle la vuelta a otra para saber si es la pareja o no. Si las dos cartas son la misma, se eliminan del tablero. En caso contrario, las cartas se dan la vuelta.

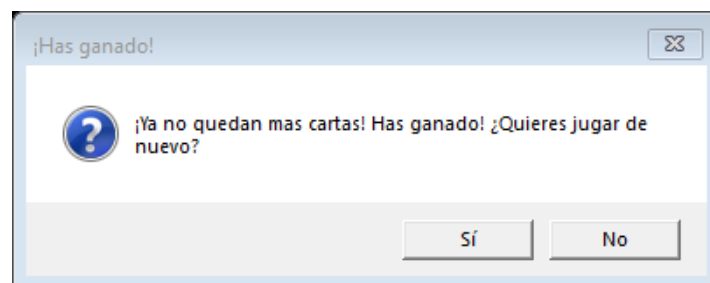
La tabla del panel de preferencias muestra las cartas que hay en el tablero, y si están abiertas o no. También podemos cambiar el número de cartas con las que queremos jugar o el tiempo que queremos para memorizarlas antes de que se den la vuelta. El tiempo no se podrá modificar mientras estemos en el tiempo de memorizar. La nueva configuración se aplicará al darle al botón de refrescar o al comenzar una nueva partida.



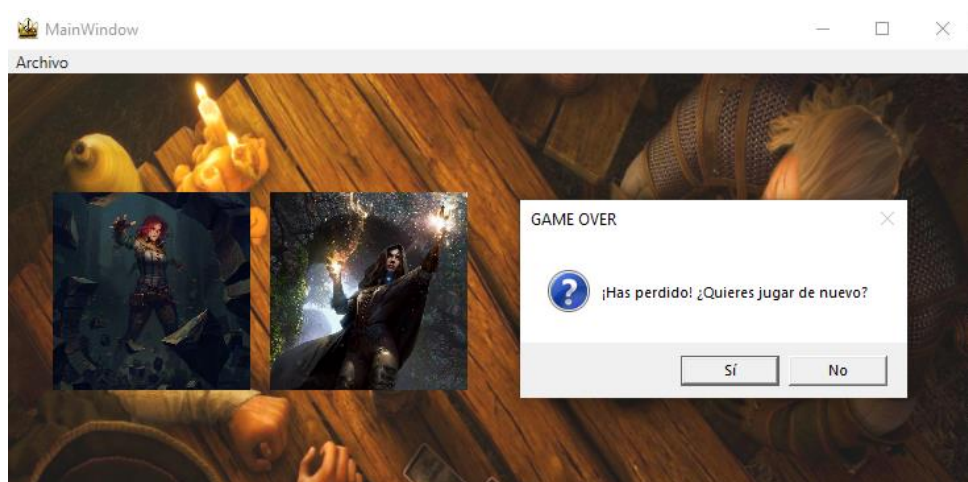
Si seleccionamos un elemento de la tabla, el botón de eliminar se activa, permitiendo borrar del tablero la carta seleccionada:



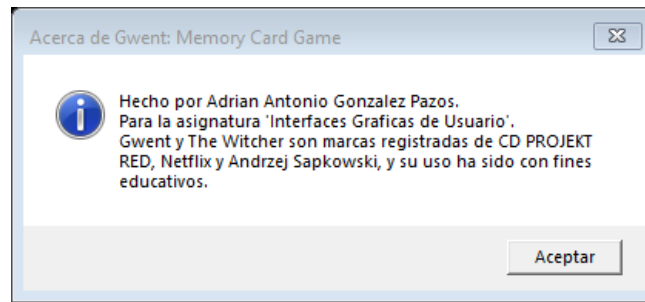
Si conseguimos encontrar las parejas de todas las cartas, nos saldrá un mensaje avisándonos y preguntando si queremos jugar otra partida. En caso afirmativo, comenzara otra partida, y en caso negativo, regresa a la pantalla principal.



Si borramos cartas desde la tabla puede darse el caso de que nos quedemos con una carta, o con dos cartas que no son iguales, entonces en ese caso el programa le dará la vuelta a las cartas que quedan y nos mostrara un mensaje de *game over* con una funcionalidad parecida al anterior:

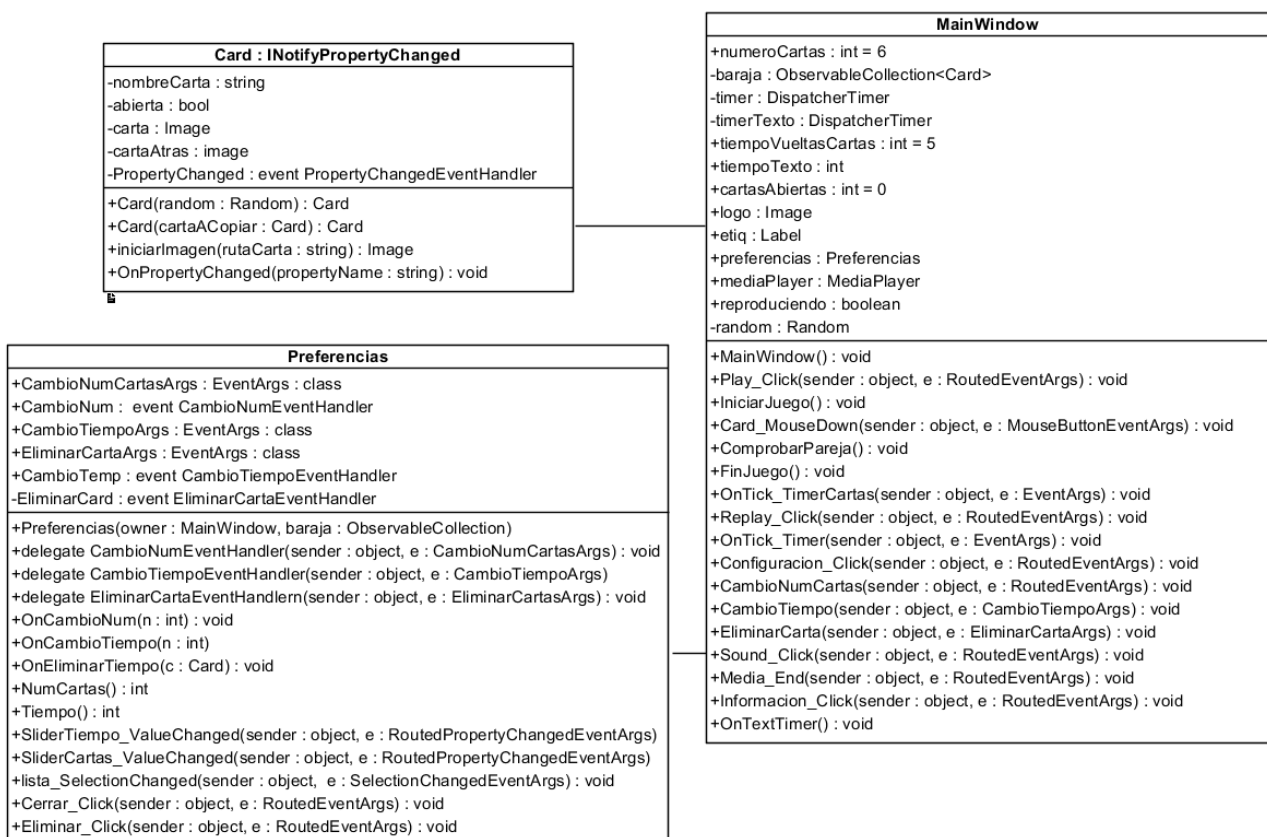


El botón de “Acerca de” que podemos encontrar en “Archivo” nos abre un mensaje con información relativa al juego:



Manual del Programador

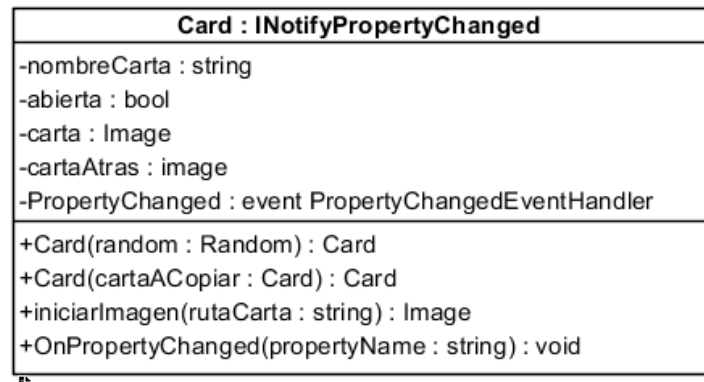
Las clases del programa, con sus atributos y métodos son las siguientes:



La clase Card representa el modelo de la aplicación.

MainWindow y Preferencias son las clases controladoras de ventana principal y del cuadro de dialogo de preferencias, respectivamente.

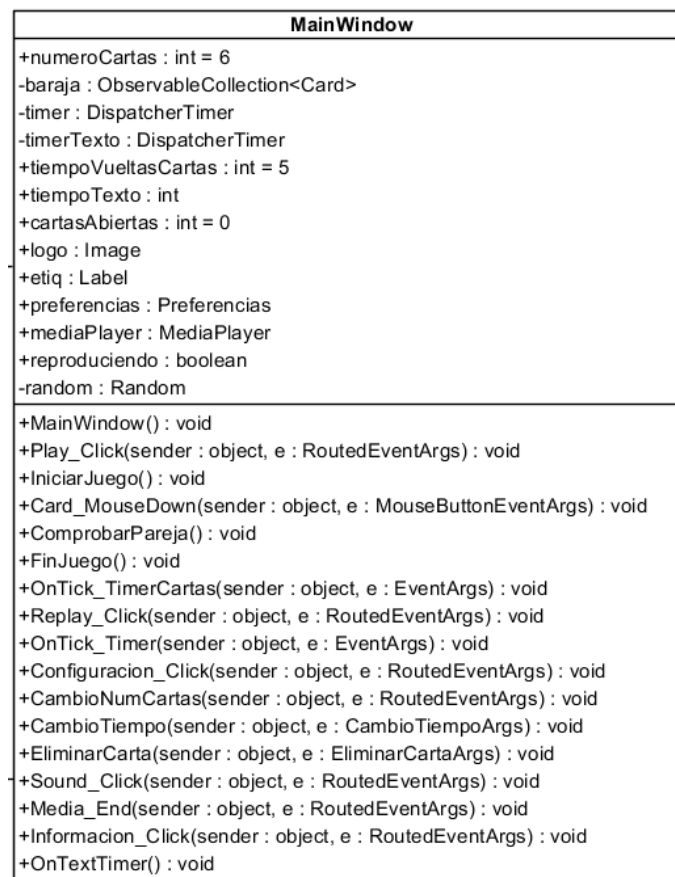
Clase Card:



Destacan los siguientes métodos:

- Card: crea una instancia de la clase dándole un valor aleatorio al atributo nombreCarta, que será el nombre de la foto que representa a la carta.
- iniciarImagen: Inicia la imagen que representara a la carta con el nombre que se le pase como parámetro.
- OnPropertyChanged: Servirá para notificar a la observableCollection que la propiedad abierta ha cambiado.

Clase MainWindow



Destacan los siguientes métodos:

- IniciarJuego: Este método será llamado cuando el juego comience, ya sea llamado por el botón de jugar o el botón de volver a comenzar el juego. Se encarga de iniciar el temporizador, borrar, si fuera necesario, la colección baraja y la colección *children* del *canvas*, crear las diferentes instancias de la cartas y añadirlas a la colección baraja y al canvas.
- Card_MouseDown: Le da la vuelta a la carta que el usuario haya pulsado, y en caso de que el usuario vaya a abrir una tercera carta, le da la vuelta a las dos anteriores.
- ComprobarCarta: Es llamado por Card_MouseDown si hay dos cartas abiertas, se encarga de comprobar si son iguales, y de eliminarlas en caso afirmativo.
- FinJuego: También es llamado por Card_MouseDown() comprueba las cartas que quedan, si no quedan cartas, lanza el mensaje de que has ganado, en caso de que quede 1 o 2, que no sean pareja, le dará la vuelta y mostrara el mensaje de game over.
- EliminarCarta: Atiende el evento lanzado desde el panel de preferencias, y elimina del canvas y de la colección la carta que el usuario haya borrado en el panel de preferencias.

Clase Preferencias:

Preferencias
+CambioNumCartasArgs : EventArgs : class +CambioNum : event CambioNumEventHandler +CambioTiempoArgs : EventArgs : class +EliminarCartaArgs : EventArgs : class +CambioTemp : event CambioTiempoEventHandler -EliminarCard : event EliminarCartaEventHandler
+Preferencias(owner : MainWindow, baraja : ObservableCollection) +delegate CambioNumEventHandler(sender : object, e : CambioNumCartasArgs) : void +delegate CambioTiempoEventHandler(sender : object, e : CambioTiempoArgs) +delegate EliminarCartaEventHandlern(sender : object, e : EliminarCartasArgs) : void +OnCambioNum(n : int) : void +OnCambioTiempo(n : int) +OnEliminarTiempo(c : Card) : void +NumCartas() : int +Tiempo() : int +SliderTiempo_ValueChanged(sender : object, e : RoutedEventArgs) : void +SliderCartas_ValueChanged(sender : object, e : RoutedEventArgs) : void +lista_SelectionChanged(sender : object, e : SelectionChangedEventArgs) : void +Cerrar_Click(sender : object, e : RoutedEventArgs) : void +Eliminar_Click(sender : object, e : RoutedEventArgs) : void

Destacan los siguientes métodos:

- OnCambioNum, OnCambioTiempo, OnEliminarCarta son los métodos de los eventos que notifican a MainWindows sobre un cambio en el número de cartas, de tiempo o la carta a eliminar.

Referencias

- Apuntes y ejercicios de clase
- Webs y documentación (Consultadas el 12/12/19):
 - Poner un icono al juego:
 - <https://docs.microsoft.com/es-es/visualstudio/ide/how-to-specify-an-application-icon-visual-basic-csharp?view=vs-2019>
 - Uso de la clase *random*:
 - <https://docs.microsoft.com/es-es/dotnet/api/system.random?view=netframework-4.8>
 - Uso de la clase *Image*:
 - https://docs.microsoft.com/es-es/dotnet/api/system.windows.controls.image.source?view=netframework-4.8#System_Windows_Controls_Image_Source
 - <https://stackoverflow.com/questions/32429464/c-sharp-relative-path-to-image-will-not-work>
 - Uso de la clase *MediaPlayer*:
 - <https://docs.microsoft.com/es-es/dotnet/api/system.windows.media.mediaplayer?view=netframework-4.8>
 - <https://stackoverflow.com/questions/24320487/how-to-run-the-mediaplayer-on-repeat-mode-in-c-sharp-wpf>
 - Barajar aleatoriamente una colección:
 - <https://stackoverflow.com/questions/273313/randomize-a-list>