* 1. **Diagrama de clases**

En el diagrama de clases o UML mostraremos el modelado del sistema software visualizando todas las funciones creadas.

El diagrama ha sido clasificado en columnas siguiendo un orden de mayor a menor importancia dentro de la columna.

* + 1. **Primera columna**

Contiene las funciones pertenecientes al formulario las cuales se pueden dividir en funciones asociadas a botones, funciones asociadas a los timer propios de las interfaces de visual studio, a funciones de actualización de la interfaz o funciones asociadas a elementos especiales como imágenes o sliders.

* + 1. **Segunda columna**

Contiene funciones específicas de la api

**Core functions** : Funciones pertenecientes al núcleo del programa, son las funciones que transforman a MSP y envían o hacen la recepción de los datos.

**Basic functions** : Funciones básicas de manejo del drone.

**General functions** : Realizan acciones más generales dentro de los movimientos que puede realizar el drone.

**Missions** : Contiene las funciones pertenecientes a las misiones.

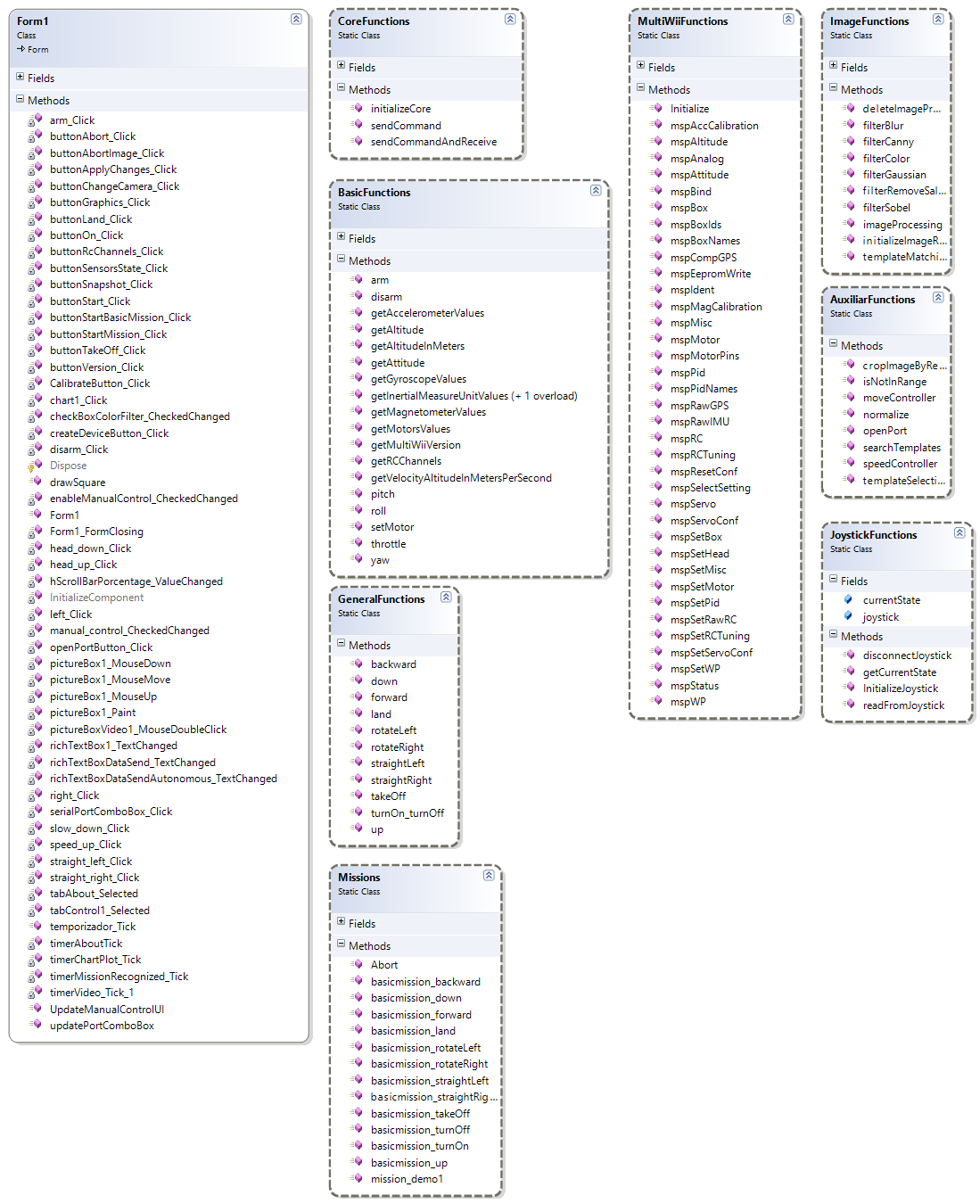
* + 1. **Tercera columna**

Contiene las MultiwiiFunctions, estas funciones hacen la traducción de las funciones propias de Multiwii a nuestro lenguaje de proramación C#

* + 1. **Cuarta columna**

Contiene otras funciones no clasificables o clases individuales, como las funciones propias para el manejo de joysticks, funciones auxiliares utilizadas en las demás clases las funciones y las funciones pertenecientes al reconocimiento de imágenes y preparación del video.

A continuación se muestra el diagrama de clases generado una vez completada la aplicación:



* 1. **Control manual**

El control manual está compuesto por las funciones que hacen posible el correcto funcionamiento del joystick asociado al programa. Para que esto sea posible, es necesario utilizar una librería externa, SlimDX la cual facilita la conexión de elementos "plug and play" para sistemas Windows ofreciendo funciones básicas de llamada.

A continuación explicamos a grandes rasgos cada una de las funciones contenidas en la clase JoystickFunctions:

1. InitializeJoystick()

Es la función que inicializa el dispositivo una vez conectado, recoge el elemento haciendo una llamada al sistema operativo y creando una nueva instancia del objeto Joystick. Si el objeto ha sido recogido de forma satisfactoria, obtiene las propiedades del mismo.

1. readFromJoystick()

Si el Joystick ha sido recogido de manera satisfactoria, guarda en una variable los valores del propio joystick, entre ellas los valores de cada uno de los canales en el momento actual .

1. getCurrentState()

Muestra los valores actuales del joystick, entre ellos los valores de giro de cada uno de los ejes.

1. disconnectJoystick()

Si el joystick estaba habilitado y conectado, se desconectará el elemento.