

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE  
TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



Proyecto Fin de Carrera

**Captura Gesticular orientada a la Docencia a través de tecnología**

**Kinect**



AUTOR: Gilbert Silva Quituisaca  
DIRECTOR: Felipe García Sánchez  
Septiembre / 2014



---

<b>Autor</b>	Gilbert Silva Quituisaca
<b>E-mail del Autor</b>	zgsilva80@gmail.com
<b>Director</b>	Felipe García Sánchez.
<b>E-mail del Director</b>	felipe.garcia@upct.es
<b>Título de PFC</b>	Captura Gesticular orientada a la Docencia a través de tecnología Kinect.

### **Resumen:**

#### 1. Planteamiento inicial del Proyecto

Aplicación orientada a la docencia en donde el objetivo es de captura, reconocimiento y ejecución de la acción requerida por el usuario – docente.

Tenemos la utilización de gestos y no de botones lo cual facilitaría y haría una docencia mucho más dinámica de lo que hoy en día tenemos.

Pensado para que exista la posibilidad mediante la gesticulación de paso de transparencias de presentaciones docentes. Kinect permite a los usuarios controlar e interactuar con el dispositivo sin necesidad de tener contacto físico con un controlador mediante una interfaz natural de usuario que reconoce gestos, comandos de voz y objetos e imágenes.

#### 2. Objetivos del Proyecto

Reconocimiento de las distintas gesticulaciones realizadas por el docente en sus clases tales como movimientos de pantallas, paso de diapositivas, etc.

#### 3. Fases del Proyecto

- Programación de la aplicación en Visual Basic 2012.
- Integración de dispositivo (Kinect) con el software.
- Testing de recepción de datos mediante reconocimiento del movimiento gesticular.
- Salida a producción del proyecto al completo.
- Documentación de la memoria del proyecto.

---

#### 4. Bibliografía

- Programming with the Kinect for Windows Software Development Kit Add gesture and posture recognition to your applications.
- <http://www.kinectfordevelopers.com/category/programacion/c-sharp/>
- <http://juank.black-byte.com/varios-programar-k4w-kinect-windows/>
- <http://msdn.microsoft.com/en-us/library/hh855355.aspx>
- <https://universidadreyjuancarlos.wordpress.com/2011/10/17/>
- <http://msdn.microsoft.com>
- [http://openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page)

<b>Titulación</b>	Ingeniería Técnica en Telecomunicaciones especialidad Telemática.
<b>Departamento</b>	Tecnología de la Información y Comunicaciones
<b>Fecha de Presentación</b>	Sept. 2014

---

# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>8</b>
<b>CAPITULO I. TECNOLOGÍAS EMPLEADAS .....</b>	<b>15</b>
Visualización externa del Sensor .....	16
Visualización Interna del Sensor. ....	17
Instalación del SDK para el desarrollo de Kinect. ....	20
<b>CAPITULO II. DESARROLLO DEL PFC .....</b>	<b>26</b>
Cómo administrar el Stream de Color.....	27
Características del Stream de Color .....	28
Cómo administrar el Stream de esqueleto.....	29
Características del Stream de Profundidad .....	31
<b>CAPITULO III. INTERFAZ DE USUARIO .....</b>	<b>36</b>
<b>CAPITULO IV. CONCLUSIONES Y LINEAS FUTURAS .....</b>	<b>37</b>
<b>BIBLIOGRAFIA Y REFERENCIAS .....</b>	<b>40</b>





# INTRODUCCIÓN

## Motivación

Se llega a realizar este proyecto orientado a la docencia ya que se necesita capturar, reconocer, controlar y ejecutar acciones requeridas de los docentes como pueden ser un desarrollo de presentaciones en clase, o exposiciones, mediante aplicaciones como pueden ser un típico fichero de \*.ppt para impartir clases sin necesidad de tener contacto físico con algún periférico como puede ser un ratón de PC o puntero laser (presentadores) como tradicionalmente se realiza. Tenemos la posibilidad de utilizar una interfaz natural de usuario (Natural User Interface) que reconoce gestos, comandos de voz, y objetos llegando a utilizar la tecnología *Kinect* es la que nos permite hacerlo.

## SISTEMAS ANTERIORES:

Aunque Microsoft se lleva todos los puntos por la innovación de Kinect enfocada en su mayoría a los videojuegos para su consola Xbox 360, no es el primer intento de hacer una NUI en que las manos estén libres.

Es interesante descubrir que prácticamente todas las viejas consolas intentaron construir un periférico con el nivel de interactividad del ingenio de Microsoft.

## U-Force (NES)

Es grande y difícil de manejar, utiliza sensores infrarrojos para detectar movimiento dentro de cuadrado de acción. Fue creado por *Broderbund* para la *Nintendo Entertainment System* en los años 80. La detección de movimiento por



Figura 1. Ejemplar de U-Force(NES)

infrarrojos, no se perfeccionó hasta 1989, y el resultado final fue una lucha para



que el controlador reconociera algún tipo de movimiento delante de sus “ojos digitales”.

También vino con un yugo aéreo de plástico, que transformó el controlador de movimiento de manos libres que no funcionó, en una palanca de mando débil que tampoco funcionó.

### Activator (Sega Genesis)

En 1993, SEGA lanzó el *SEGA activator*, un curioso periférico para la consola *MegaDrive* que 'teóricamente' permitía controlar el juego mediante sensores de infrarrojos que detectaban el movimiento del jugador, sustituyendo los botones y el stick del mando.

SEGA Activator era un anillo octogonal dentro del cual se situaba el jugador. Cada uno de los ocho lados emitía un haz de infrarrojos, de manera que cuando los brazos o piernas del jugador se interponían, el sistema lo detectaba como una pulsación. Cada lado o panel correspondía con un botón (A, B, C) y



Figura 2. Ejemplo del dispositivo Sega Activator

con los cuatro movimientos (arriba, abajo, derecha, izquierda). Pero en la práctica el sistema de control era muy deficiente pues la señal de infrarrojos no es la mejor elección para algo que requiere una respuesta rápida y precisa. Había mucha interferencia con las luces brillantes o las superficies reflectantes en el uso del periférico. Además del engorro del montaje y tener que recalibrarlo cada vez que empezaba el juego.

## Power Glove

*Power glove* es un controlador de accesorios para NES, y el primer controlador de interfaz periférico para recrear movimientos de mano sobre una pantalla de televisión u ordenador en tiempo real. Impopular, impreciso y dificultad para el uso de los controles.



Figura 3. Ejemplar de Power Glove

## Pantomation

Es un sistema de vídeo que aprovechaba el croma en 1977, lograba reconocer un objeto, seguir su trayectoria, aislarlo según su color, enviar esa información a un ordenador y provocar que sucedieran acciones relativas a ese movimiento.

El concepto del *croma* es una capacidad que todavía tienen algunas mezcladoras de vídeo y cuyo principio base, el de separar ciertas frecuencias de color, permite que cuando un actor se viste de verde, el ordenador lo elimine y sustituya por su versión digital.




*En la reciente historia, ¿Qué existía en el mercado?*

En la competencia de las consolas de última generación, fue Nintendo quien tomó la delantera al lanzar su **Wii** (estrenada el 19 de noviembre de 2006) la característica más distintiva de esta consola es su mando inalámbrico, el Control Remoto Wii (o **Wiimote**), el cual puede ser usado como un dispositivo de mano con el que se puede apuntar, además de poder detectar la aceleración de los movimientos en tres dimensiones. Pero tras el reciente lanzamiento del **Play Station Move** el 15 de septiembre de 2010 (que usa un mando principal con sensores de movimiento y una esfera en su extremo que se ilumina y la cámara **PlayStation Eye**, que se encarga de detectar la posición del mando

principal), llegó al mercado *Kinect* el 4 de noviembre de 2010 salió a la venta en los Estados Unidos y México, Europa el 10 de noviembre de 2010 que busca superar a la competencia con un novedoso sistema que *no requiere controles* para dirigir a los personajes en la pantalla. Esta vez el cuerpo del usuario/jugador es el control.

**PlayStation Move** funciona de forma similar al *Wii mote*, es decir, necesitamos de un mando para que la consola pueda detectar los movimientos, mientras que *Kinect* es libre, es decir, que la cámara reconoce nuestro cuerpo a la perfección y es capaz de independizar las extremidades y captar su movimiento de forma independiente.

## COMPARATIVA DE ESTOS ULTIMAS TRES DISPOSITIVOS

	KINECT	WII	MOVE
			
<b>Video-Consola</b>	Xbox 360 (Microsoft)	Wii (Nintendo)	PlayStation3 (Sony)
<b>¿Es un accesorio?</b>	Sí	No	Sí
<b>Necesidad de mando</b>	No	Sí	Sí
<b>Cámara</b>	Sí	No	Sí (aparte)
<b>Reconocimiento de voz</b>	Sí	No	No
<b>CPU</b>	Tres procesadores Power PC a 3,2 GHz	Power PC IBM Broadway a 729 MHz	Procesador Cell Power PC a 3,2 GHz
<b>Procesador gráfico</b>	ATI a 500 MHz, 10 MB de DRAM	ATI Hollywood a 600 MHz	RSX@550 MHz
<b>Procesador de polígono</b>	0,5 TeraFLOPS	sin confirmar	1,8 TeraFLOPS
<b>Resolución</b>	1080 p	480 p	1080 p
<b>Memoria</b>	512 MB GDDR3 RAM. 700 MHz de DDR	512 MB	256 MB XDR RAM @3.2 GHz, 256 MB GDDR3 VRAM @700MHz.

<b>Almacenamiento</b>	Disco duro de 20, 60, 120 y 256 GB (dependiendo del modelo) extraíble. Soporte para unidades de memoria	Memoria flash de 512 MB ampliable por SD o disco USB	Disco duro 160GB, MemoryStick, SD, CompactFlash
<b>Puertos y conexiones</b>	Dos ranuras de memoria, tres puertos USB 2.0	Dos USB traseros, WiFi	Bluetooth, USB, WiFi (PSP)
<b>Conexión a Internet</b>	Puerto Ethernet, conexión a Xbox Live!, preparada para WiFi	Nintendo WiFi Connection	Ethernet, WiFi
<b>Formatos discos</b>	CD y DVD; lector HD DVD independiente	DVD, sistema propietario, discos GameCube	CD, DVD, Blu-ray
<b>Sonido</b>	320 canales. Procesador de sonido de 32 bits	Compatible con Dolby Pro Logic 2. Mando con altavoz	Dolby 5.1
<b>Peso</b>	3,5 kg	1,8 kg	5 kg
<b>Dimensiones (cm)</b>	30x25x8	15x20x5	33x28x10
<b>Precio consola</b>	249,90€	149,90 € (consola + 1(con accesorio) ó 2 juegos)	249,90€
<b>Precio accesorio</b>	Sensor Kinect Xbox 360 + Kinect Adventures: 149,90 €	Mando Remote Plus: 54,90 €	Mando de movimiento: 39,90 €
<b>Precio mandos secundarios</b>		Mando Nunchako: 9,90 €	Mando secundario: 29,90 € cámara: 39,95 €

Figura 4. Comparativa entre Wii Move y Kinect

## Objetivos

Lo que tratamos de hacer es que exista el reconocimiento de las distintas articulaciones mediante las diferentes gesticulaciones realizadas por el docente (cuerpo humano) en sus clases tales como el paso de diapositivas, etc. El objetivo principal del proyecto es utilizar la tecnología Kinect para poder desarrollar aplicaciones dedicadas exclusivamente al sector de la docencia. Hay muchos más sectores en donde también se desarrolla aplicaciones utilizando Kinect. Podremos ver el desarrollo de las actividades cotidianas en clase con la ayuda de esta potente herramienta y éstas sean más fluidas y más centradas llegando a obtener el mejor provecho de las clases en día a día.

## Fases del Proyecto

- Programación de la aplicación en Visual Basic 2012
- Integración de dispositivo (Kinect) con el software.
- Testing de recepción de datos mediante reconocimiento del movimiento gesticular.
- Salida a producción del proyecto al completo.
- Documentación de la memoria del proyecto.

## CAPITULO I. TECNOLOGÍAS EMPLEADAS

Microsoft presentó su “*Natal Project*” en el que prometía controlar los menús con el cuerpo y la voz, escanear la propia ropa y con reconocimiento facial, una IA (Inteligencia Artificial) mucho más elaborada y que pueda interactuar con el usuario.

El sensor Kinect se puede definir como una fuente de múltiples corrientes/flujo (streams) de datos que puede proporcionar un flujo para cada tipo de datos que recoge. Debido a que el sensor Kinect reconoce streams de color / profundidad / audio. Además, el SDK de Kinect para Windows puede calcular un seguimiento completo de esqueletos detectados en frente del sensor usando sólo estos streams.

Kinect en un principio fue «un controlador de juego libre y entretenimiento» creado por Alex Kipman. Es una combinación muy innovadora de cámaras, micrófonos y software que convierte su cuerpo en el controlador de aplicaciones, videojuegos, etc. Se puede programar con C++, C # o Visual Basic para crear soluciones que soportan el gesto y reconocimiento de voz utilizando el sensor Kinect para Windows y un ordenador o dispositivo integrado.

## Componentes del SDK

Estos componentes incluyen los siguientes:

1. **Hardware de Kinect**– Los componentes de hardware, incluyendo el sensor de *Kinect* y el controlador USB a través del cual el sensor está conectado al ordenador.

### Estructura física del sensor Kinect

Visualización externa del Sensor

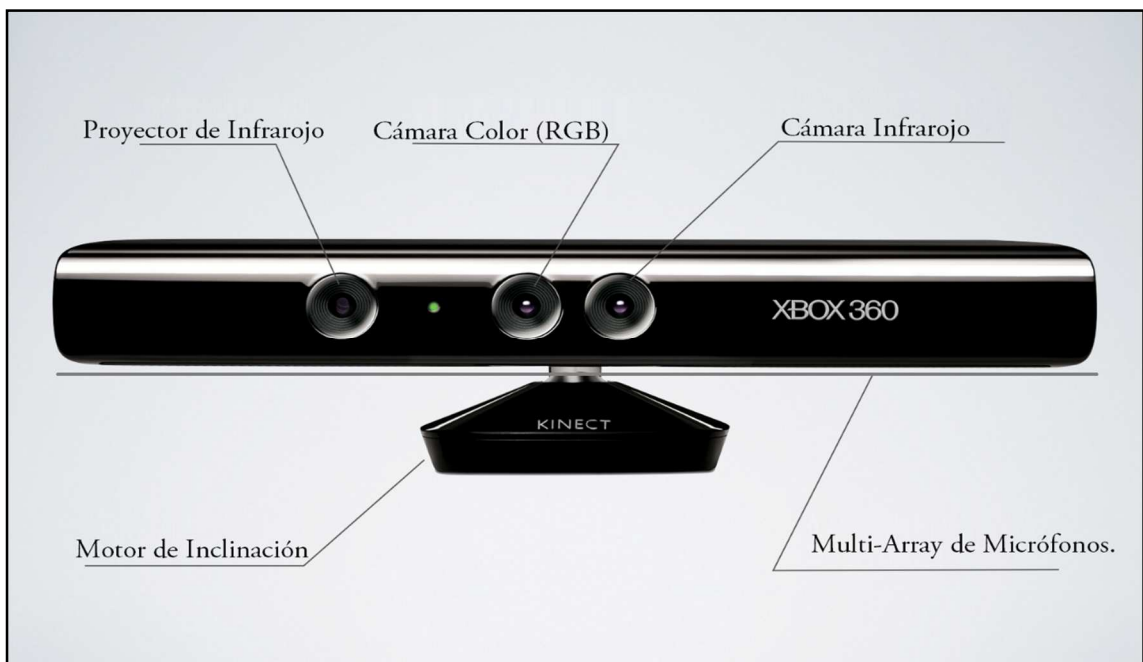


Figura 5. Sensor Kinect.



## Visualización Interna del Sensor.



**Figura 6. Kinect en Piezas.**

Se puede visualizar que el dispositivo tiene cierta complejidad, la placa principal se divide en tres secciones diferentes (en comparación con el enfoque estándar de una sola tarjeta) y requiere un ventilador de refrigeración propio.



**Figura 7. Vista superior trasera del Sensor**

La lista completa de los componentes se administra en forma consistente en los siguientes:

- Motor Eléctrico
- 3 Secciones De Placa Base
- Ventilador De Refrigeración
- Dos Cámaras
- Proyector de luces infrarrojas
- 4 Micrófonos
- Cuatro Tipos Diferentes De Tornillos
- conversor estéreo analogico digital Wolfson con micrófono preamplificador
- paquete chip sap
- Fairchild N-Channel Powertrench Mosfet
- Hub Controlador Nec Usb 2.0
- Controlador de interfaz de la cámara
- Controlador De Interfaz de Cámara Marvell Soc
- Hynix (semiconductor) 512MB DDR2 SDRAM
- Dispositivo Analógico Cmos "Rail to Rail" De Salida Del Amplificador Operacional (Rail to Rail indica que la tensión de salida del circuito integrado puede llegar a ser muy próxima a la tensión de alimentación)
- Ti 8-Bit, 8-Canal De Muestreo De Un Convertidor A / D
- Conductor Allegro Low Voltage Stepper sencillo / doble Motor
- Memoria flash Nv St 8Mbit
- Procesador del Sensor de Imágenes PrimeSense
- Controlador USB de Audio Ti
- Acelerómetro Kionix



**Figura 8. Vista superior trasera del Sensor**

**2. Drivers de Kinect**– Los controladores de Windows para el Kinect, que se instalan como parte del proceso de instalación del SDK. Los drivers de Kinect son compatibles con:

- El conjunto de micrófonos Kinect como un dispositivo de audio en modo de núcleo que se puede acceder a través de las API de audio estándar de Windows.
- Audio y controles de transmisión de video para el streaming de audio y vídeo (color, profundidad, y el esqueleto).
- Funciones de enumeración de dispositivos que permitan una aplicación para utilizar más de un Kinect.
- Visual Studio 2012 o superior.
- SDK de Kinect (<http://www.microsoft.com/en-us/download/details.aspx?id=40278>).
- El SDK proporciona una biblioteca de software sofisticado y herramientas para ayudar a los desarrolladores utilizar la forma natural basado en Kinect, que detecta y reacciona a los acontecimientos del mundo real. El Kinect y la biblioteca de software pasan a interactuar con la aplicación, como se muestra en la Figura.

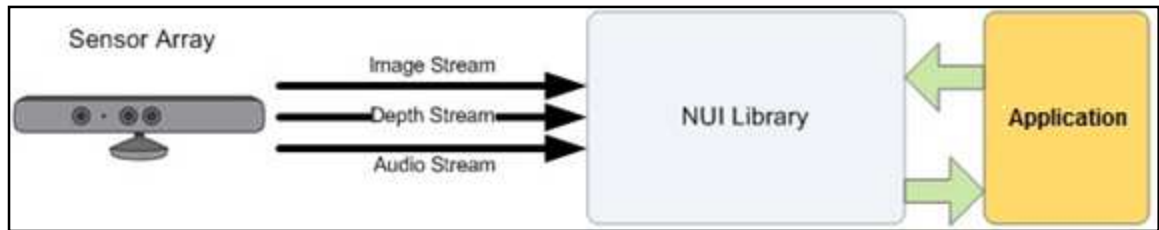


Figura 9. Interacción entre sensor y biblioteca NUI y aplicaciones.

### 3. Componentes de Audio y Video

- La interfaz natural de usuario para el seguimiento de esqueleto, de audio, y el color y la profundidad de imagen.

4. **DirectX Media Object (DMO)** para la formación de conjunto de micrófonos y localización de la fuente de audio.

5. **Windows 8 API estándar** – El audio, voz y API's multimedia de Windows 8, como se describe en el SDK de Windows 8 y el Microsoft Speech SDK.

### Instalación del SDK para el desarrollo de Kinect.

Se instala el SDK (<http://msdn.microsoft.com/es-es/windows/desktop/hh852363.aspx>) de la interfaz de usuario natural (NUI), la interfaz de programación de aplicaciones (API) y los controladores de Microsoft Kinect para integrar el sensor Kinect en Windows. Cabe destacar que el SDK de Kinect para Windows es compatible con Windows 7 y Windows 8 y con arquitecturas x86 y x64.

## Arquitectura SDK de Kinect para Windows.

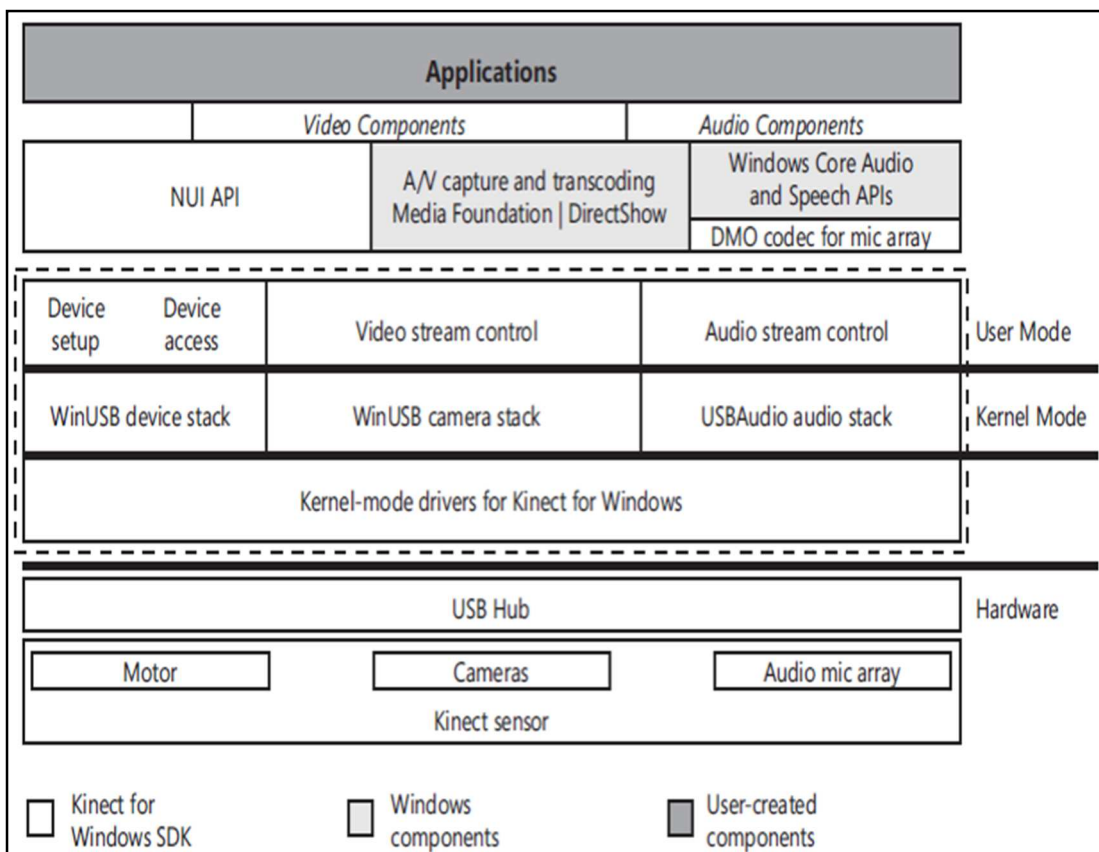


Figura 10. Arquitectura SDK de Kinect para Windows.

## Requisitos necesarios:

- Windows 8, 8.1 (32 - bit o 64-bit edition), Windows 7(cada vez menos).
- Microsoft Visual Studio 2012(recomendada) u otra edición de Visual Studio.
- NET Framework 4
- XNA Game Studio 4
- Procesador Doble núcleo de 2,66 GHz o superior
- USB 2.0
- 2 GB de RAM
- Tarjeta gráfica compatible con DirectX 9.0c
- Sensor *Kinect* de Windows.

## Pantallazos de instalación de Kinect for windows SDK

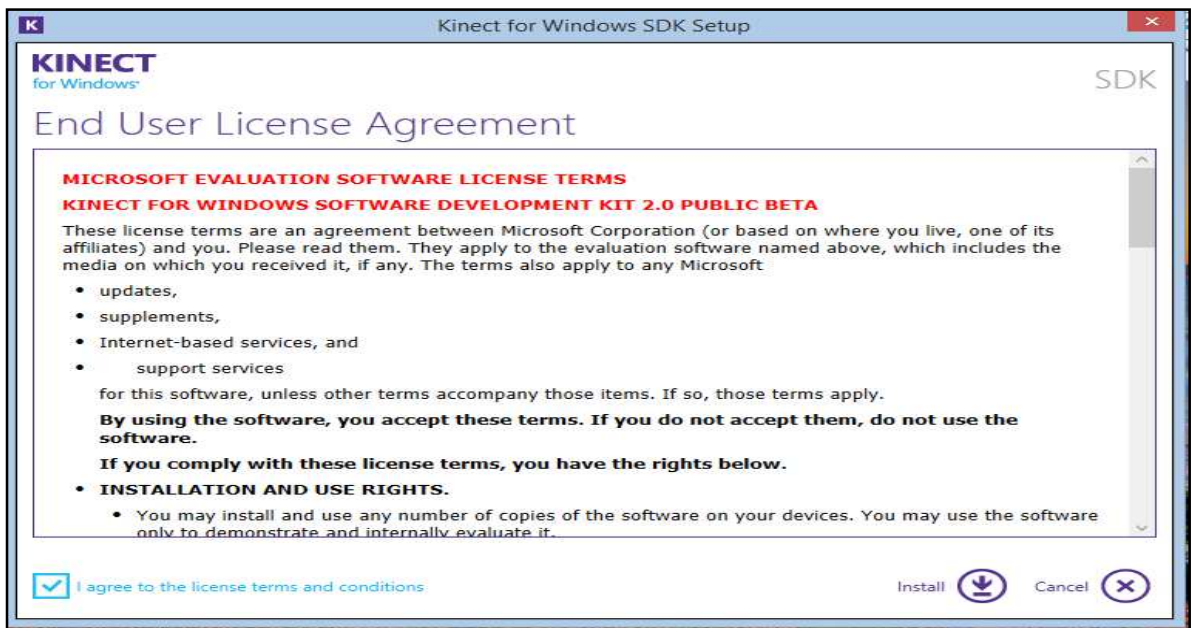


Figura 11. Pantalla de instalación SDK de Kinect.

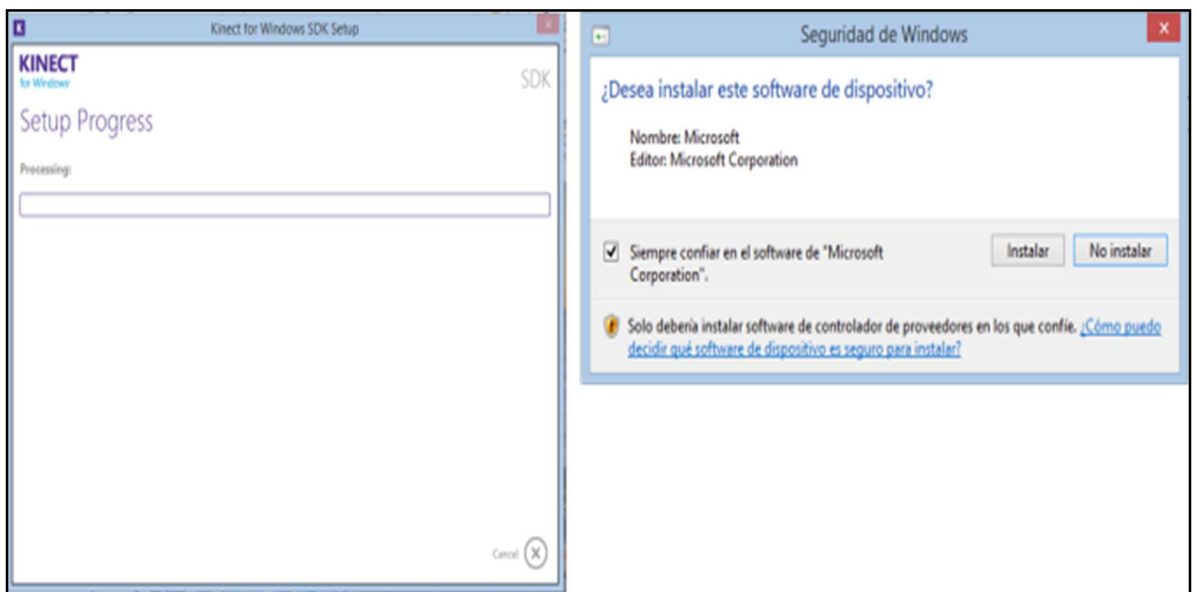


Figura 12. Pantalla de instalación SDK de Kinect.

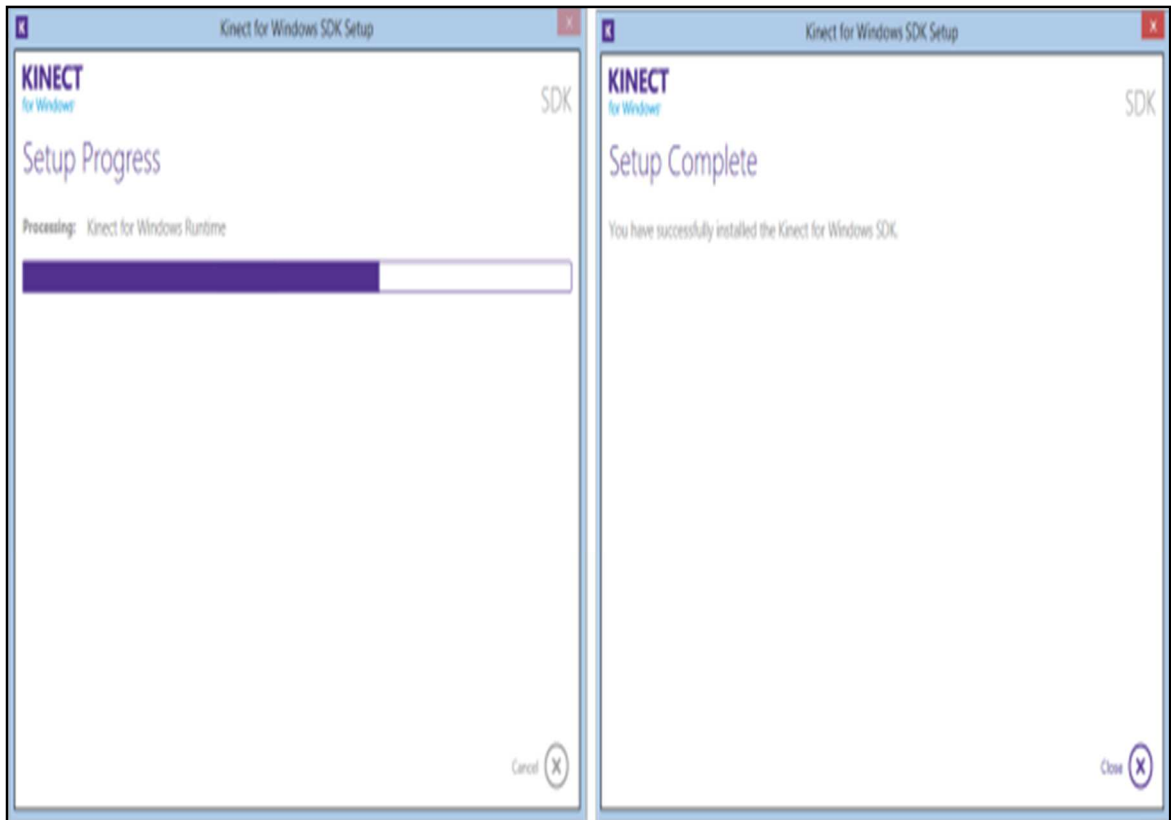


Figura 13. Pantalla de instalación SDK de Kinect.

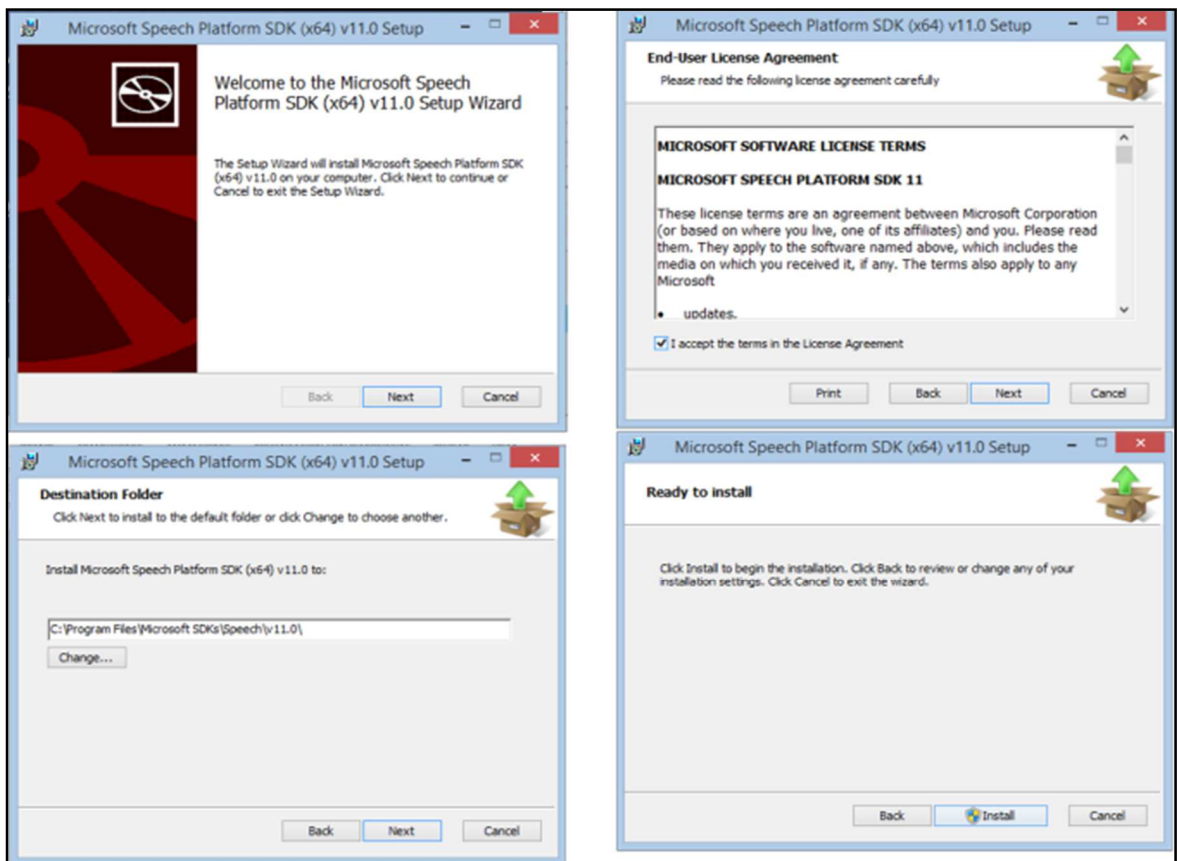


Figura 14. Pantalla de instalación SDK de Kinect.

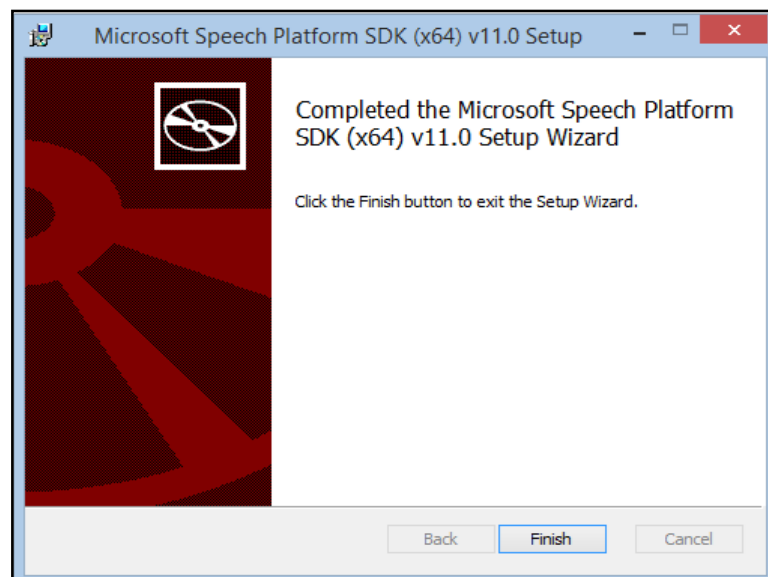


Figura 15. Pantalla de instalación SDK de Kinect.



## SDK en Visual Studio

En Visual Studio se habilitan referencias para agregar en la aplicación para Windows 8.

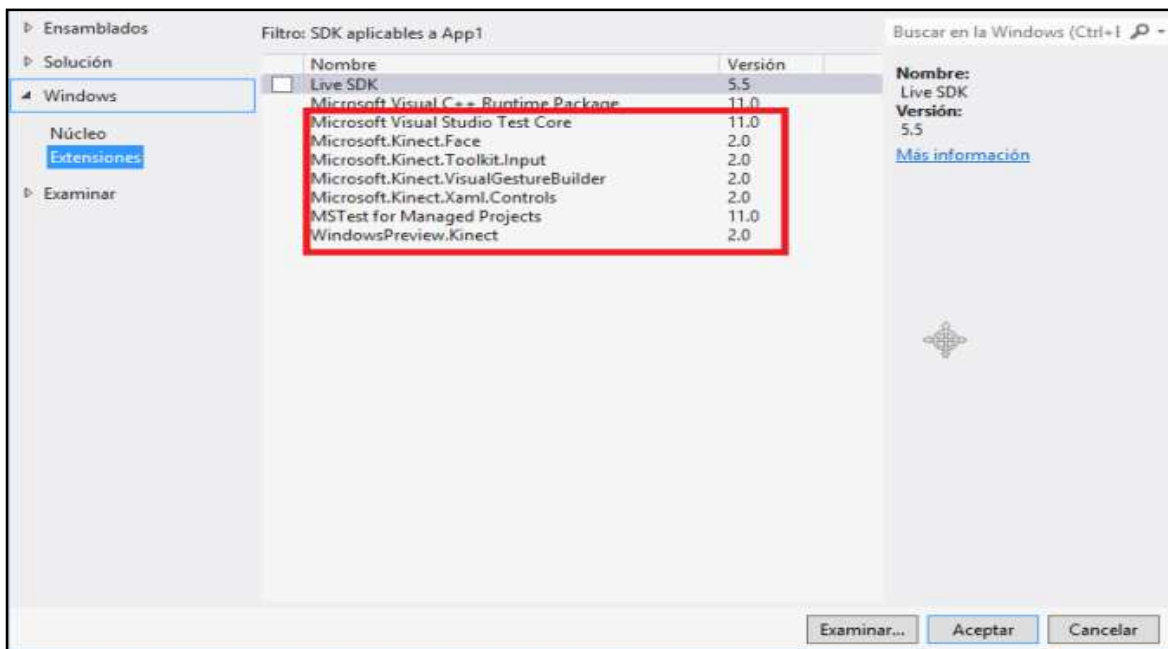


Figura 16. Habilitación de referencias en Visual Studio 2012.

## CAPITULO II. DESARROLLO DEL PFC

Primeramente se prepara el escenario para la realización de una aplicación de WPF (Windows Presentation Foundation) utilizando la plataforma que nos ofrece Microsoft, Visual Basic Studio 2012. En ella desarrollamos la idea de la siguiente manera:

Diagrama de Flujo de la aplicación:

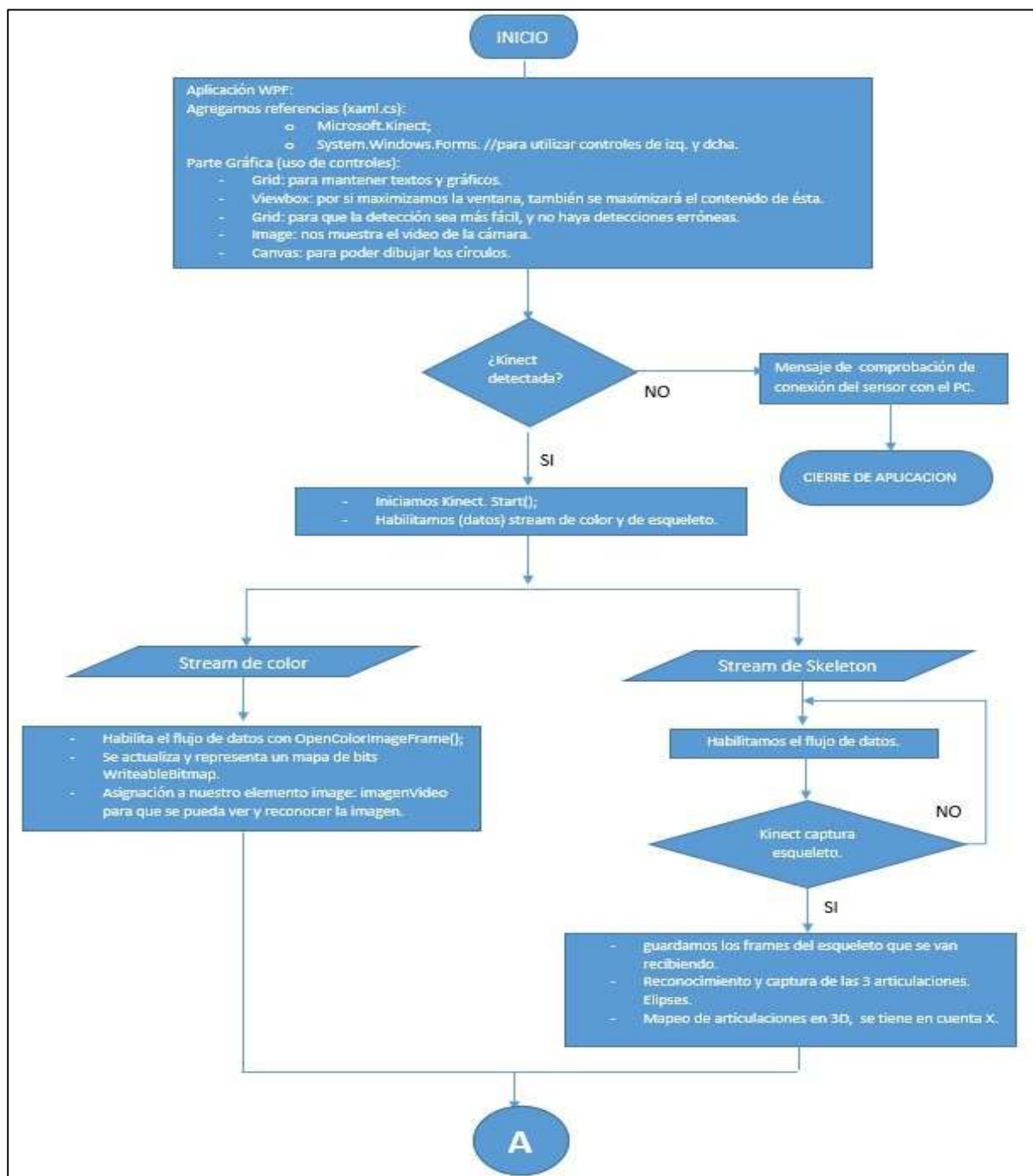


Figura 17. Diagrama de flujo implementación principal

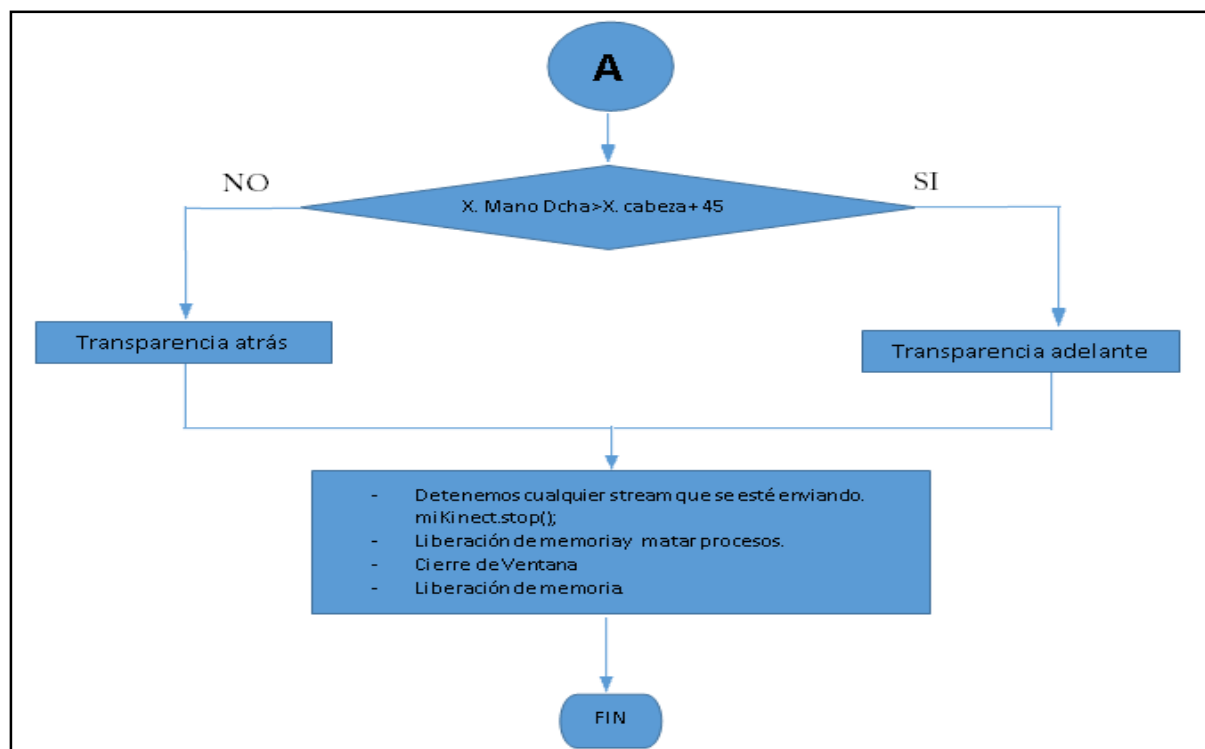


Figura 18. Diagrama de flujo implementación principal.

Primeramente pasamos a detectar lo que es la Kinect, verificamos que no contenga algún valor nulo si no llega a detectar, mostramos el mensaje de la que se necesita conectar el sensor de Kinect y se cerrará la aplicación.

Iniciamos el sensor y pasamos a habilitar el *stream de color y de profundidad*.

### Cómo administrar el Stream de Color

Una vez habilitado el stream de datos, utilizamos el evento de colorFrame para capturar lo datos que nos envían los streams para ello pasamos a crear el mapa de bits (writeablebitmap) que es donde se reflejará el reconocimiento de la imagen, en el que se concreta algunos valores de los parámetros como son:

- Alto (height) del frame.
- Ancho (width) del frame.

- Los “ppp” puntos por pulgada (dpi = dots per inch) horizontal (dpiX) del mapa de bits.
  - Los “ppp” puntos por pulgada (dpi ) vertical (dpiY) del mapa de bits.
  - El formato de los píxeles, que en este caso será Bgr32.
  - El valor que define la paleta del bitmap que son los colores disponibles para un tipo de imagen compatible.
- También pondremos el array de bytes los cuales representan el contenido de la imagen.
- Y por último el *stride* que es lo que va a ocupar una fila de píxeles en memoria (en bytes) o sea en donde tenemos el ancho de la imagen (número de píxeles por fila) y lo que ocupa cada píxel en bytes.

Con esto llegamos a asignar el mapa de bits de imagen d color y ya tenemos de esta manera el establecido la imagen a color de la Kinect.

### Características del Stream de Color

Obviamente, los primeros datos proporcionados por el sensor Kinect es el flujo de vídeo. A pesar de que funciona como una cámara 3D, en su nivel más básico, Kinect es una cámara estándar que puede capturar secuencias de vídeo con las siguientes resoluciones y velocidades de fotogramas:

- 640 × 480 a 30 fotogramas por segundo. (FPS) usando rojo, verde y azul, (RGB en inglés)
- 1280 × 960 at 12 FPS usando formato RGB
- 640 × 480 at 15 FPS usando formato YUV2 (o raw YUV3)

El formato RGB es un formato de 32 bits que utiliza un formato lineal

X8R8G8B8 de color en un espacio de color RGB estándar. (Cada componente puede variar de 0 a 255, ambos inclusive.)

El formato Yuv es de 16 bit, el formato YUV es eficiente porque usa solo 16 bits/píxel, mientras que RGB utiliza 32 bits por píxel, por lo que el controlador necesita menos memoria. Los datos de vídeo se pueden utilizar para dar retroalimentación visual a los usuarios para que puedan verse a sí mismos interactuar con la aplicación.

## Cómo administrar el Stream de esqueleto

Estos esqueletos son creados por la cámara de color y de profundidad, la cámara de profundidad especialmente, ya que ésta captura la distancia de cada píxel. Los creadores del SDK de Kinect se guiaron primero separando los objetos que detectaban estableciendo los más grandes como los más importantes como puede ser la figura de una persona y basando en patrones se llegó a reconocer si lo detectado era humano u otro objeto, y observando el movimiento se llegó a estructurar un esqueleto digital muy parecido al esqueleto humano. Se llega a definir la captura 19 huesos y 20 articulaciones, no todos los 206 huesos del cuerpo humano ya que sería muy complicado los reconocimientos de éstos.

Para capturar los esqueletos:

- Habilitar y capturar y almacenar el flujo de datos de esqueletos
- Copiamos los datos de los esqueletos a los arrays anteriormente creados.
- Recorremos, obtenemos y verificamos ( la calidad de captura) los datos del array con un loop.
- Capturamos y comprobamos las articulaciones que deseamos manejar

del esqueleto, utilizando alguna propiedad ya sea por su posición tridimensional, calidad de captura o tipo de articulación.

Y el SDK nos devolverá los valores de acuerdo a los criterios del paso anterior.

Utilizamos en este caso 3 elipses (de cara a las presentaciones no se verían) para poder comprobar que nos ha reconocido las articulaciones que en este caso son la mano derecha, mano izquierda y la cabeza, ésta última la utilizamos para poder calcular la distancia que las manos recorren en su eje horizontal desde el centro (cabeza, 45cm en X desde el centro del cuerpo).

Se crea una función por cada articulación que nos dice si tenemos un movimiento activo tanto atrás como delante (con un variable booleana) teniendo en cuenta esos 45 cm.

En el caso de que el sensor haya detectado la articulación en posición X de la mano derecha con punto de referencia la posición X de la cabeza entonces se activa y se asocia la flecha derecha a través de la clase SendKeys que proporciona métodos para enviar pulsaciones de tecla a una aplicación como por ejemplo: `SendKeys.SendWait("{Right}");` indicando al sensor que “avance” a la derecha. El mismo procedimiento pasa para lo que es la parte de la flecha izq.

Aquí tenemos algunos ejemplos de las teclas con sus respectivos códigos:

TECLA	CÓDIGO
RETROCESO	{BACKSPACE}, {BS}, o {BKSP}
INTER	{BREAK}
BLOQ MAYÚS	{CAPSLOCK}
INICIO	{HOME}
INS o INSERT	{INSERT} o {INS}
FLECHA IZQUIERDA	{LEFT}
FLECHA DERECHA	{RIGHT}
SUPR o SUPRIMIR	{DELETE} o {DEL}
FLECHA ABAJO	{DOWN}
FIN	{END}
ENTRAR	{ENTER} o ~
ESC	{ESC}
AYUDA	{HELP}

Figura19. Ejemplos de las teclas pulsadas que proporciona la Clase *SendKeys*

Con esto se llega a cumplir el objetivo. Luego en cada caso fallido ya sea de reconocimiento, detección de sensor, o cualquier percance se procede a cerrar la ventana de la aplicación, liberamos memoria y paramos cualquier tipo de stream que esté recibiendo el sensor.

### Características del Stream de Profundidad

El sensor SDK de Kinect además de ser una cámara de color, también es una cámara de profundidad. De hecho, el sensor puede enviar una stream compuesta de la distancia entre el plano de la cámara y el objeto más cercano encontrado. Cada píxel de la imagen resultante contiene la distancia dada expresada en milímetros.

La Figura muestra una pantalla el stream de profundidad estándar con la identificación del jugador.



**Figura 20. Reconocimiento del stream de profundidad estándar**

Para inicializar y utilizar el stream de profundidad, se utiliza código similar a lo que se utiliza para la transmisión de vídeo.

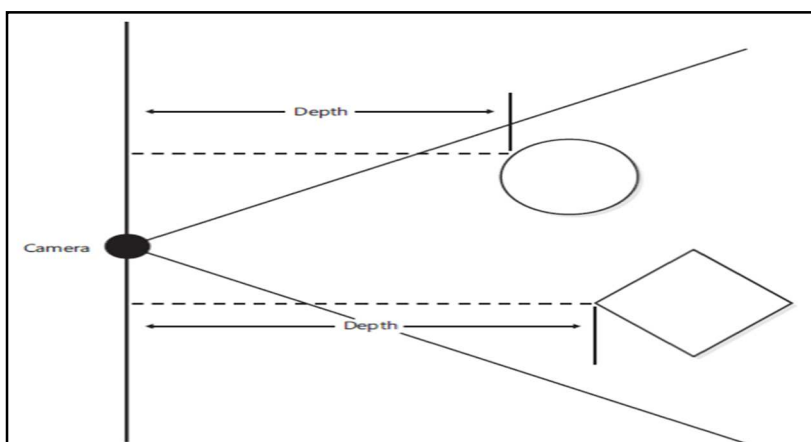
Resoluciones que son compatibles:

- 640 × 480
- 320 × 240
- 80 × 60

### **Calculo Datos de profundidad.**

Hay dos posibles modos para los valores de profundidad.

Podemos utilizar el modo de cerca o estándar. Usando el modo estándar, los valores se encuentran entre 800 mm y 4000 mm inclusive; usando el modo de cerca, los valores se encuentran entre 400 mm y 3000 mm.



**Figura 21. Calculo de los datos de profundidad**



Además, el SDK retornará valores específicos para los valores fuera de rango:

Los valores menores de 400 mm o mayores de 8000 mm son marcados como [Desconocido]. En el modo de cerca, valores entre 3000mm y 8000mm están marcados como [Demasiado lejos].

En el modo estándar, los valores entre 4000mm y 8000mm están marcados como [Demasiado lejos] y valores 400mm y 800mm están marcados como [Demasiado cerca].

El flujo de datos con respecto a la profundidad se almacenan utilizando valores de 16 bits. Los 13 bits de orden superior indexado de cada píxel contienen la distancia efectiva entre el plano de la cámara y el objeto más cercano en milímetros. Los tres bits de orden inferior indexado de cada píxel contienen la representación del mapa de segmentación jugador último píxel.

El mapa del jugador está construido por el sistema Kinect cuando el seguimiento de esqueleto se activa y hay un mapa de bits en el que cada píxel corresponde al jugador indexado de la persona más cercana en el campo de visión de la cámara. Un valor de cero indica que no hay ningún jugador detectado. Los tres bits de menor orden deben ser tratados como un valor entero.

### **Seguimiento del Esqueleto**

La API de NUI utiliza el stream de profundidad para detectar la presencia de los seres humanos frente al sensor. El seguimiento esquelético está optimizado para reconocer a los usuarios que se colocan en frente el Kinect, por la colocación lateral dan algún que otro problema porque las partes del cuerpo no son visibles para el sensor.

Hasta seis personas pueden ser detectados, y una o dos personas pueden ser rastreados a la vez con el sensor Kinect. Para cada persona seguida, la NUI producirá un conjunto completo de los puntos clave posicionados diferenciando un esqueleto. Un esqueleto contiene 20 articulaciones tal como vemos en la figura.

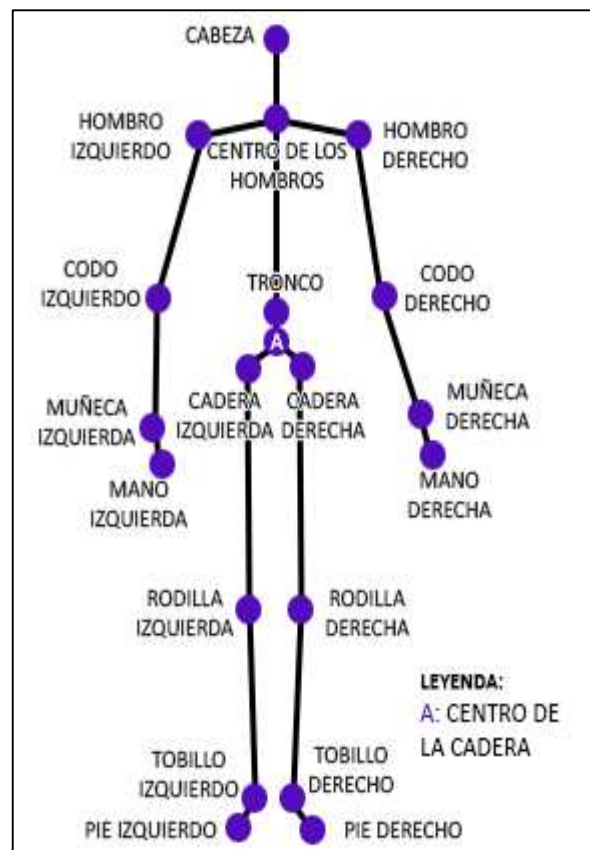


Figura 22. Articulaciones que reconoce el sensor.

Cada punto de control se define por una posición  $(x, y, z)$  expresada en el espacio esqueleto.

El "espacio del esqueleto" se define frente al sensor, que se localiza en  $(0, 0, 0)$ : el punto donde los ejes  $X$ ,  $Y$  e  $Z$  se encuentran.

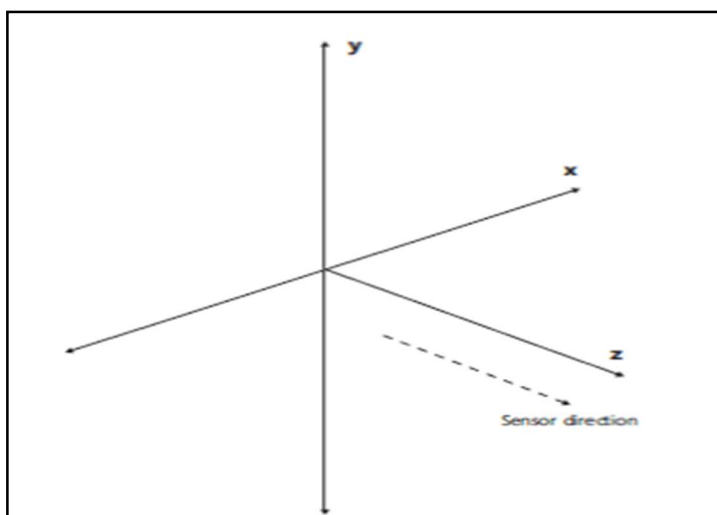


Figura 23. Esquema del punto de control definida por una posición (x, y, z)

### Características del Stream de Audio (no utilizado en éste proyecto)

El sensor Kinect incluye un conjunto de micrófonos que consta de cuatro micrófonos a pocos centímetros de distancia y dispuestos en un patrón lineal. Esta estructura permite funcionalidades muy interesantes:

- Supresión de ruido efectiva.
- Cancelación de eco acústico (AEC).
- Beamforming (Conformación de haces) y localización de la fuente. Cada micrófono en la matriz recibirá un sonido específico en un momento ligeramente diferente, por lo que es posible determinar la dirección de la fuente de audio. También puede utilizar la matriz de micrófono como un micrófono direccional orientable.

El SDK de Kinect para Windows permite:

- Captura de audio de alta calidad.
- Usar el Audio de la Kinect integra algoritmos DMO para controlar el "haz" de sonido y proporcionar la dirección de la fuente su código.
- Usa la API de Microsoft.Speech para reconocimiento de voz.

### Conexión del Sensor con el PC.

- o Es una conexión muy simple ya que se realiza mediante el puerto USB como se ve en la figura.

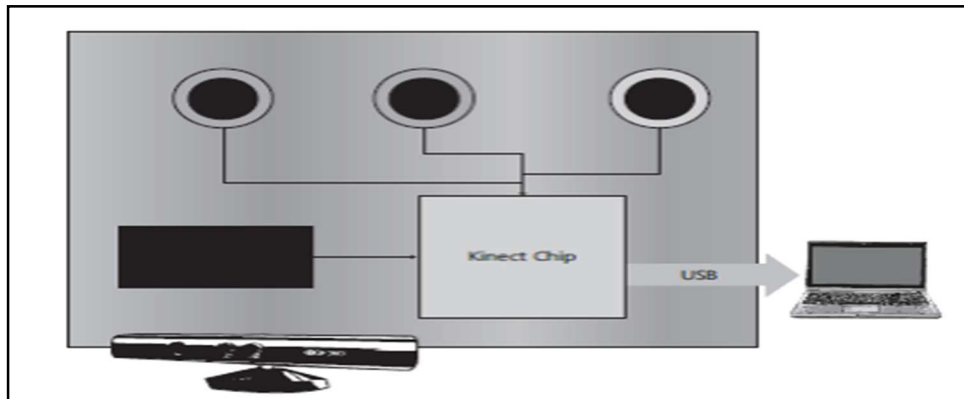


Figura 24. Conexión del Sensor Kinect con el PC.

## CAPITULO III. INTERFAZ DE USUARIO

NUI (Natural User Interface): Interfaz Natural de Usuario, es aquella en las que se interactúa con un sistema, aplicación, etc. sin utilizar sistemas de mando o dispositivos de entrada de las GUI como sería un ratón, teclado alfanumérico, lápiz óptico, Touchpad, joystick etc. y en su lugar, se hace uso de movimientos gestuales tales como las manos o el cuerpo es el mismo mando de control, en el caso de pantallas capacitivas multitáctiles la operación o control es por medio de la yemas de los dedos en uno o varios contactos, también se está desarrollando control de sistemas operativos por medio de voz humana y control cercano a la pantalla pero sin tocarla.

La integración del PC con el dispositivo Kinect da paso a lo que es la interfaz de usuario, lo podemos definir y clasificar sus distintos componentes tanto Hardware como Software.

## CAPITULO IV. CONCLUSIONES Y LINEAS FUTURAS

El gran potencial de la Kinect está en la aplicación de esta tecnología en otros campos, como la telemedicina, la rehabilitación médica, los televisores inteligentes o la educación, entre muchos otros. Los usos de un sensor de reconocimiento (que incluye interpretación de gestos, seguimiento del rostro, profundidad del campo y captura del audio) son tan amplios como la imaginación de los desarrolladores.

- Resumen del trabajo hecho, objetivos alcanzados y futuras mejoras.

En el proyecto se ha cumplido lo que se pensó desde un primer momento, que el sensor Kinect reconozca la gesticulación para que el docente pueda pasar transparencias de las presentaciones con el simple hecho de mover una mano según la dirección hacia donde quiera avanzar.

Las futuras mejoras pueden ser varias ya que existen varios métodos en la API que nos ofrece y cada vez existen mucha más información, se podría implementar para detectar varias personas, o detectar las manos como para escribir notas en las transparencias, eventos clicks de ratón.

También tenemos que el sensor se ha actualizado y pasa a ser Kinect v2, el sensor ha sido rediseñado con grandes mejoras en la fidelidad del color, definición de vídeo, campo visual, percepción de profundidad, y el seguimiento del esqueleto.

En concreto, el *Kinect* para Windows incluirá.

- Vídeo HD de 1080p, que permite nítidas y de alta calidad de escenarios aumentados.
- Un campo de visión más amplio, lo que significa que los usuarios pueden estar más cerca del sensor por lo que es posible utilizar el sensor en habitaciones más pequeñas.

- Mejora del seguimiento del esqueleto, lo que abre aún mejores escenarios para aplicaciones de salud, fitness y soluciones educativas.
- Nueva detección de infrarrojos activa, que proporciona un mejor seguimiento facial y detección de las posturas, incluso en situaciones de poca luz.

## Nuevo Sensor Kinect V2



Figura 25. Kinect V2

El Kinect v2 para Windows SDK aporta nuevas capacidades al sensor:

- **Mayor resolución.** 1920 x 1080 Full HD (antes 640 x 480). Más resolución, más detalle.
  - Permite detectar con más precisión todo el entorno.
  - Capacidad de diferenciar la orientación del cuerpo incluyendo sus manos y pudiendo diferenciar sus dedos.
  - El face tracking tiene mucho más detalle y permite captar los gestos de la cara.
  - Más calidad de imagen.

- **USB 3.0**
  - Al aumentar la velocidad de la comunicación con el ordenador los datos fluyen más rápido y esto disminuye la latencia del sensor. Pasa de 90ms a 60ms.
- **Mejora de la captación de sonidos.**
  - Se ha mejorado la eliminación del ruido ambiente y esto permite captar con más detalle las instrucciones vocales.
- **Captación de movimiento a oscuras.**
  - Ahora Kinect 2 es capaz de reconocer y captar los movimientos aunque la sala este a oscuras.
- **Desarrollo de aplicaciones en la Windows Store:** Será capaz de integrar la tecnología del Kinect a las aplicaciones de Windows y publicarlas en la Windows Store lo que dará a nuestros la capacidad de llegar a más clientes y abrir el acceso a la experiencia de usuario natural en el hogar.
- Kinect 2 permite calcular/analizar la fuerza de nuestros músculos y medir el ritmo cardíaco.
- **Unidad de Apoyo:** Donde encontrarás una mezcla de lenguajes de desarrollo, marcos y protocolos con las herramientas que ya conoces.
- **Mejora de la precisión anatómica:** Con el SDK de primera generación, los desarrolladores teníamos la capacidad de realizar el seguimiento de hasta dos personas al mismo tiempo; Ahora podemos rastrear hasta seis personas simultáneamente. Y el número de articulaciones que podían ser rastreados aumentaron de 20 a 25 articulaciones por persona y no solo eso la orientación conjunta es mucho mejor lo que nos permitirá realizar aplicaciones con un mejor seguimiento del esqueleto.
- **Simultánea, soporte multi-app:** Con esta mejora será posible ejecutar más de una aplicación simultáneamente que utilicen el *Kinect v2*.

## **BIBLIOGRAFIA Y REFERENCIAS**

<http://msdn.microsoft.com/en-us/library/hh855355.aspx>

<https://universidadreyjuancarlos.wordpress.com/2011/10/17/>

<http://msdn.microsoft.com>

[http://openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page)

<http://es.wikipedia.org/>

**Programming with the Kinect for Windows Software Development Kit Add  
gesture and posture recognition to your applications.**

<http://www.kinectfordevelopers.com/category/programacion/c-sharp/>

<http://juank.black-byte.com/vari0s-programar-k4w-kinect-windows/>