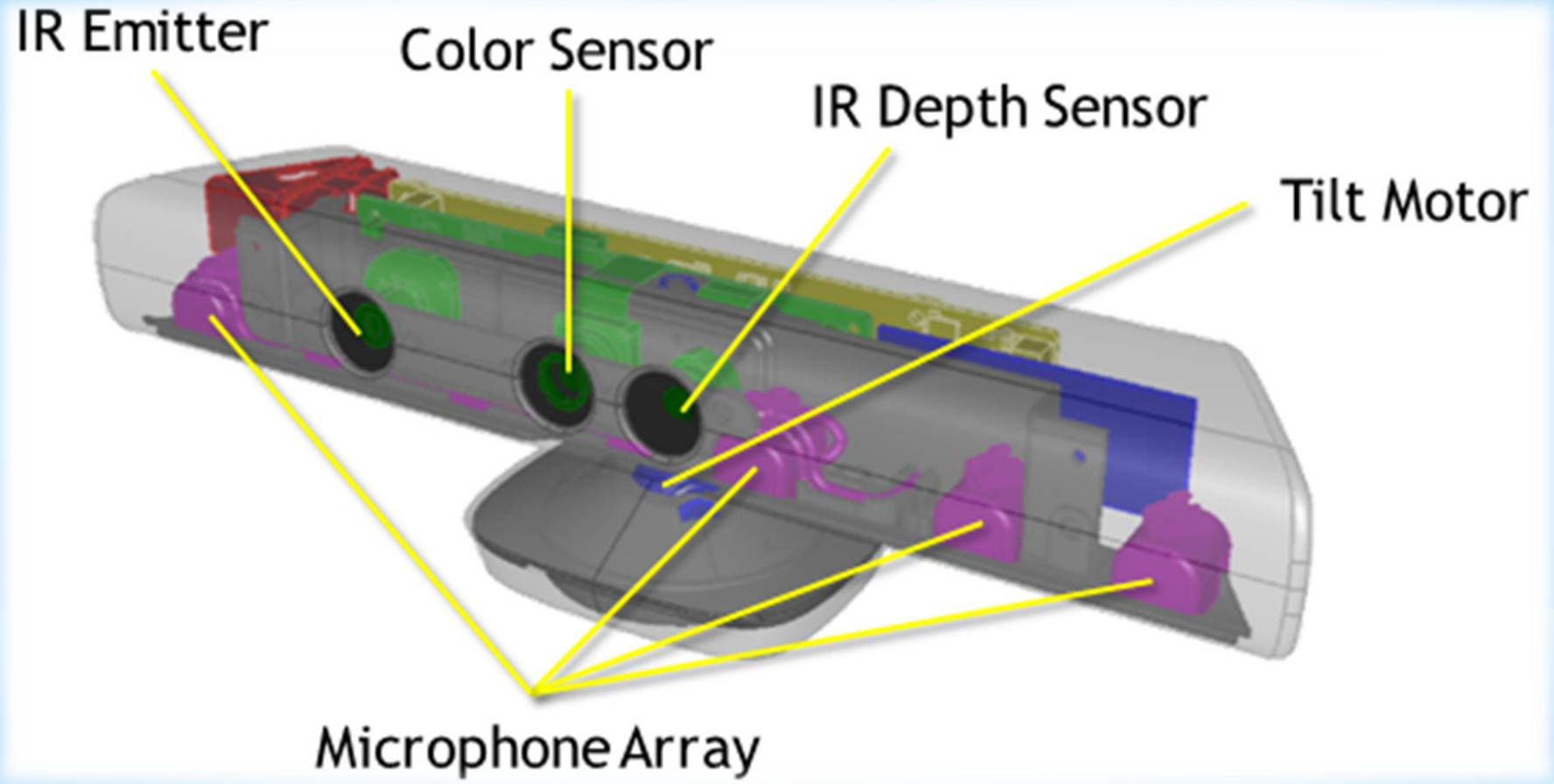
Tutorial
MS Kinect (v1)

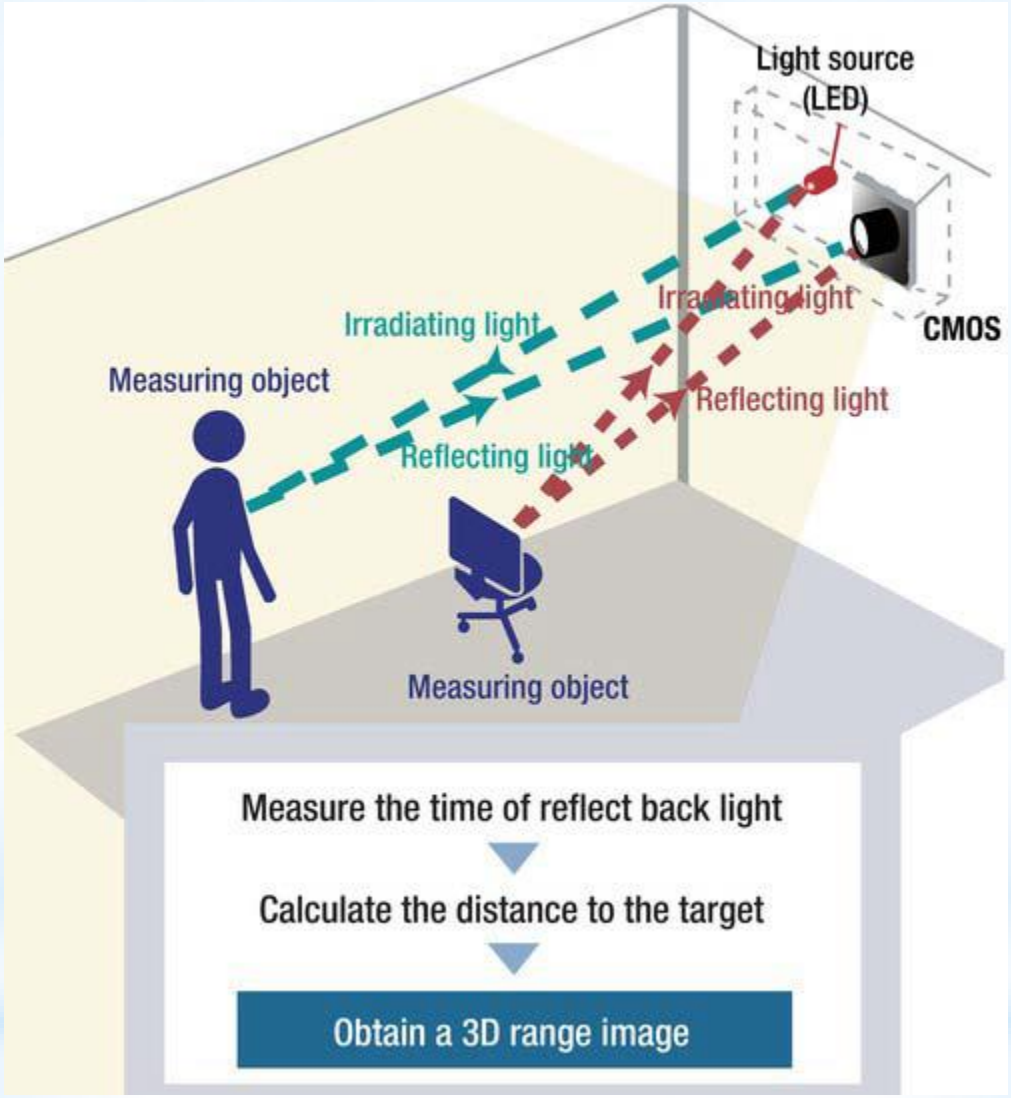
2013/2014

*Índice

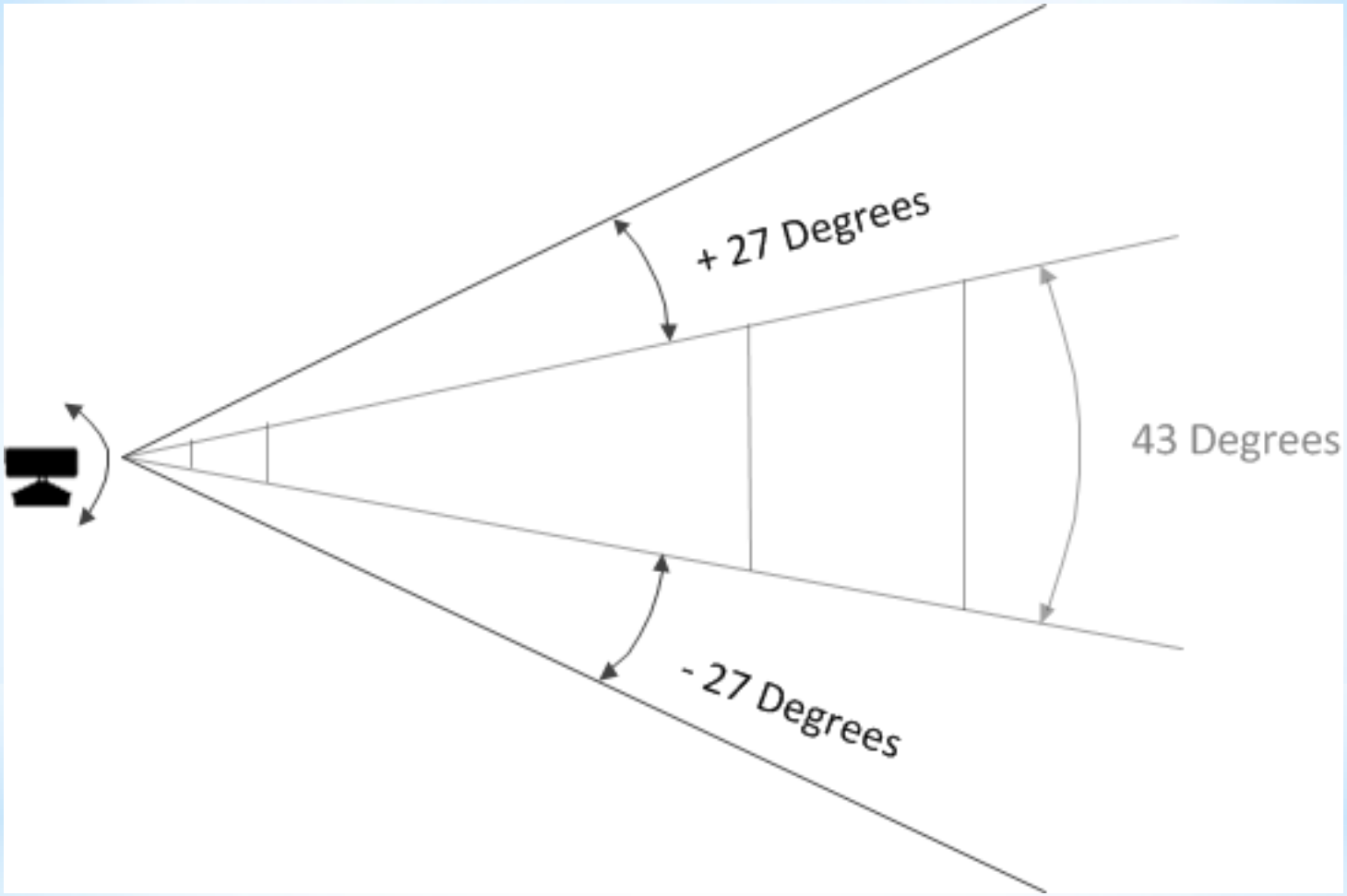
1. Características.
2. Instalación SDK.
3. Tipos de canales.
4. Crear un proyecto con Kinect.
5. Programación básica de los canales de Color, Profundidad (IR) y Skeleton.



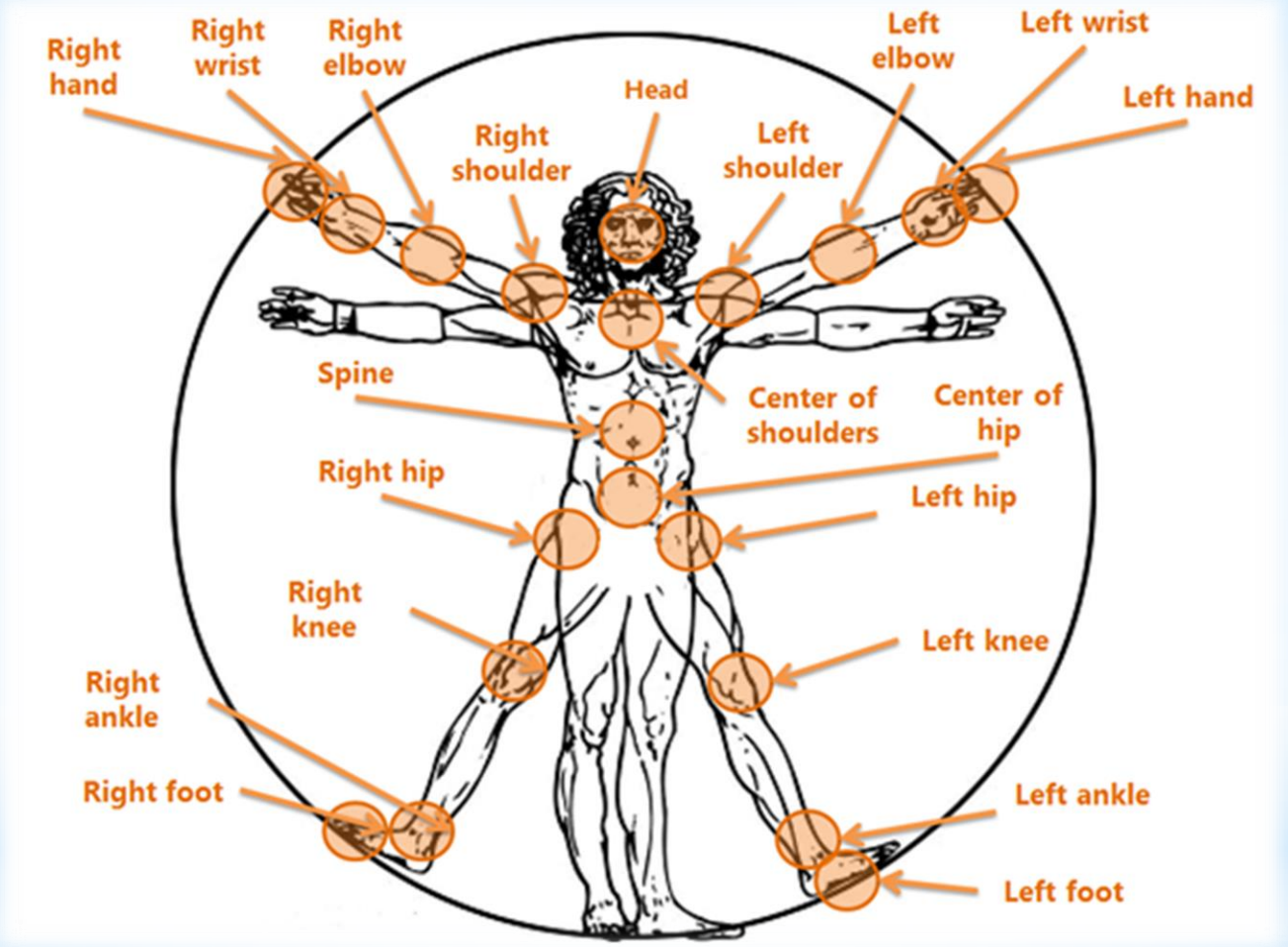
*Componentes



*Espacio de Interacción



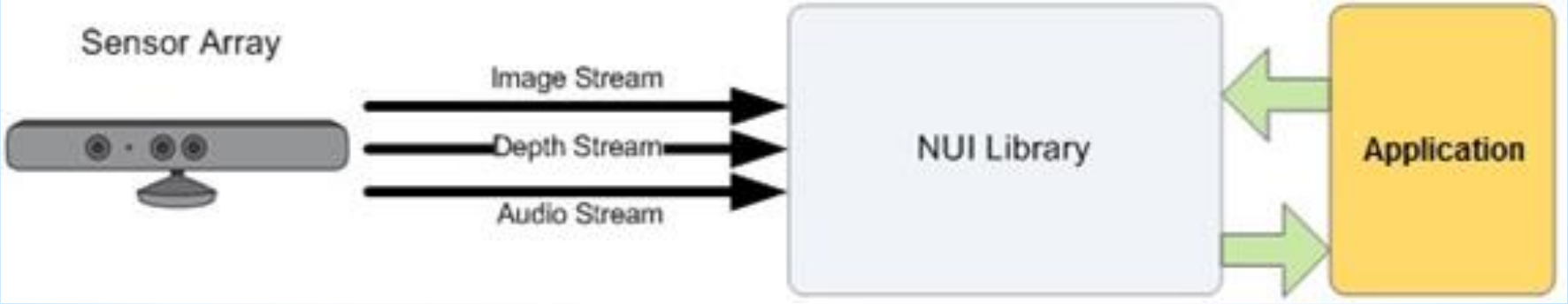
*Espacio de Interacción



*Puntos de detección

Instalación SDK

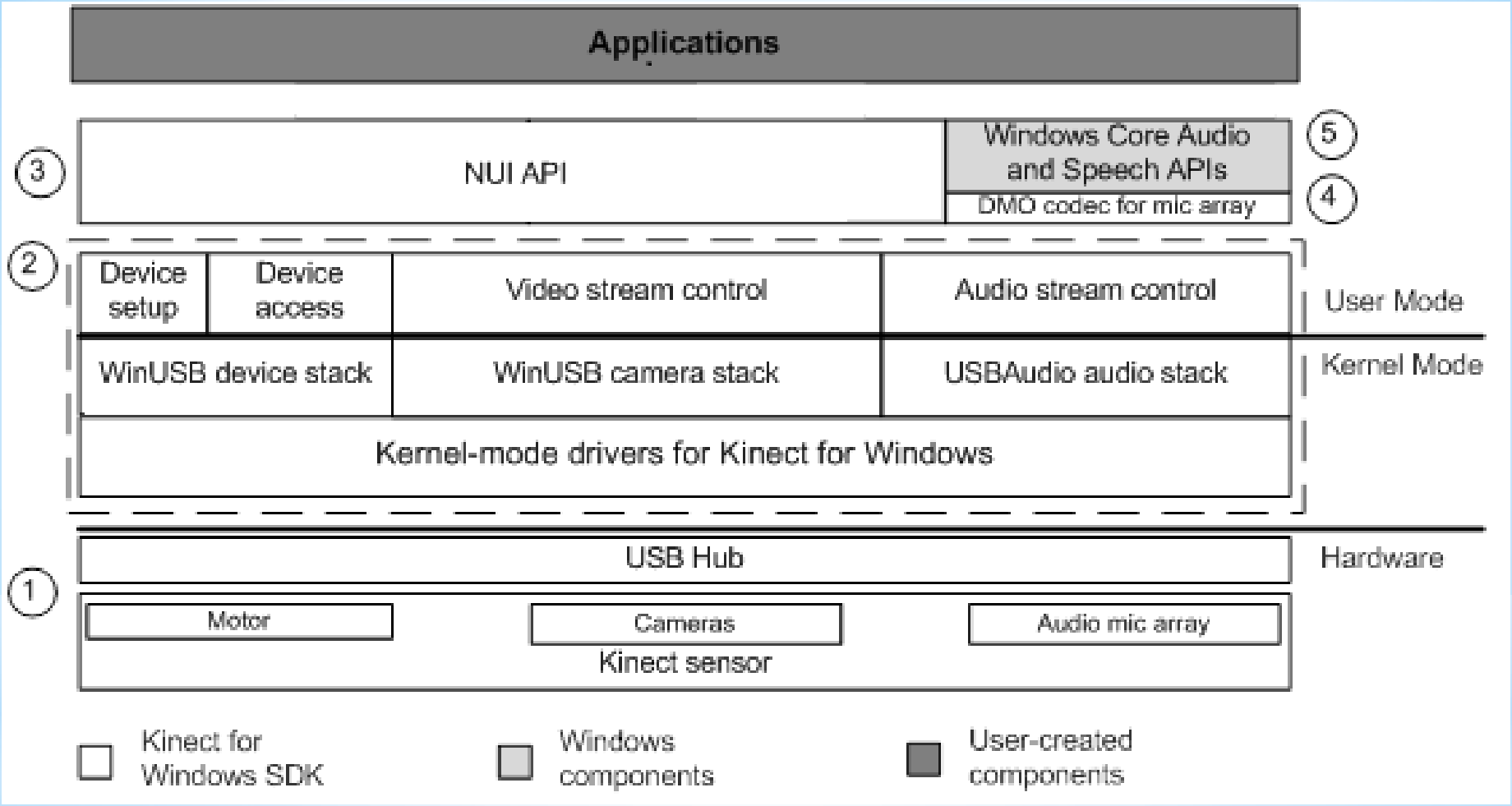
1. Visual Studio 2010 o superior
2. KinectSDK-v1.8-Setup
3. KinectDeveloperToolkit-v1.8.0-Setup
4. Conectar MS Kinect
5. Comprobar la instalación. Ejecutar ejemplos SDK



* Conexión App-Kinect

* Streams

- **Color:** igual webcam, resolución 640x480 - 1280x720
- **Depth:** datos de profundidad proporcionados por la cámara IR.
- **Skeleton:** recibe las posiciones de los puntos del cuerpo detectados.
- **Interaction:** acciones de usuario como cerrar o abrir la mano.



* Arquitectura SDK

1. Crear proyecto Visual Studio
2. Añadir a las referencias de nuestro proyecto Microsoft.Kinect
3. En nuestro programa incluir el namespace

```
using Microsoft.Kinect;
```

4. Instanciar e inicializar el sensor

```
private KinectSensor sensor;
foreach (var potentialSensor in KinectSensor.KinectSensors)
{
    if (potentialSensor.Status == KinectStatus.Connected) {
     this.sensor = potentialSensor;
     break;
    }
}
```

*Primer proyecto

```
//Color stream
this.sensor.ColorStream.Enable(
      ColorImageFormat.RgbResolution640x480Fps30);
// Depth stream
this.sensor.DepthStream.Enable(
      DepthImageFormat.Resolution640x480Fps30);
```

*Inicializar cámaras

```
// Espacio reservado para pixels de color
this.colorPixelsC = new byte[
      this.sensor.ColorStream.FramePixelDataLength];
// Espacio reservado para pixels de profundidad
this.depthPixels = new DepthImagePixel[
      this.sensor.DepthStream.FramePixelDataLength];
// Espacio reservado para pixels de color (profundidad)
this.colorPixelsD = new byte[
      this.sensor.DepthStream.FramePixelDataLength
      * sizeof(int)];
```

*Reservar espacio

```
// Bitmaps para mostrar en pantalla
this.colorBitmapC = new WriteableBitmap(
      this.sensor.ColorStream.FrameWidth,
      this.sensor.ColorStream.FrameHeight,
      96.0, 96.0, PixelFormats.Bgr32, null);
this.colorBitmapD = new WriteableBitmap(
      this.sensor.DepthStream.FrameWidth,
      this.sensor.DepthStream.FrameHeight,
      96.0, 96.0, PixelFormats.Bgr32, null);
```

*Creamos BitMaps

```
// ImageC e ImageD son elementos de IU

this.ImageC.Source = this.colorBitmapC;

this.ImageD.Source = this.colorBitmapD;
```

*Asignamos a IU

```
// Serán llamados cuando cambien las imágenes

this.sensor.ColorFrameReady +=
      this.SensorColorFrameReady;

this.sensor.DepthFrameReady +=
      this.SensorDepthFrameReady;
```

*Añadimos Event Handlers

```
// Comienza a capturar imágenes
try {
      this.sensor.Start();
} catch (IOException) {
      this.sensor = null;
}
```

*Iniciamos el sensor

```
// Activa skeleton stream
this.sensor.SkeletonStream.Enable();

// Añadir un event handler
this.sensor.SkeletonFrameReady +=
      this.SensorSkeletonFrameReady;

// Cambia el sistema de Tracking
this.sensor.SkeletonStream.TrackingMode =
      SkeletonTrackingMode.Seated;

this.sensor.SkeletonStream.TrackingMode =
      SkeletonTrackingMode.Default;
```

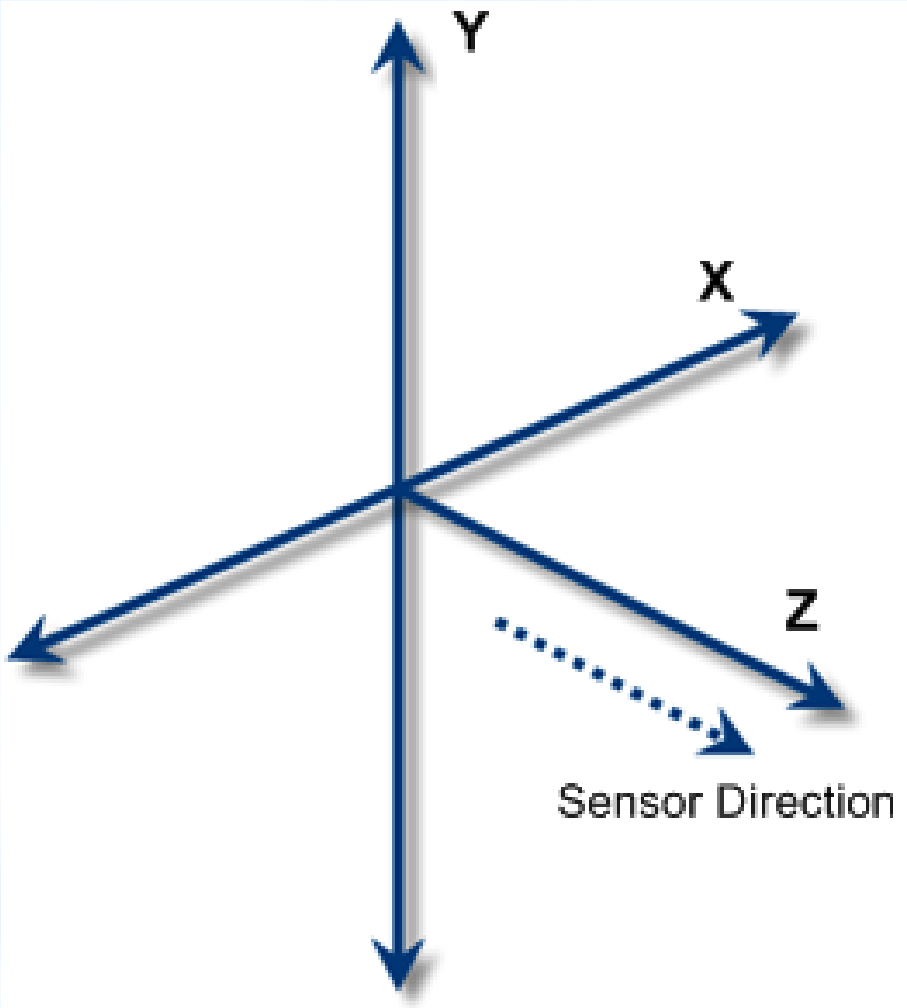
*Capturar Skeleton

```
// Donde vamos a dibujar
this.drawingGroup = new DrawingGroup();

// Imagen para mostrar el dibujo
this.imageSource =
      new DrawingImage(this.drawingGroup);

// Asignar a IU
this.ImageS.Source = this.imageSource;
```

*Asignar espacio

```
DepthImagePoint depthPoint =
this.sensor.CoordinateMapper.
    MapSkeletonPointToDepthPoint(
        skelpoint,
        DepthImageFormat.Resolution640x480Fps30
    );
return new Point(depthPoint.X, depthPoint.Y);
```

*Escalar puntos a pantalla



*Coordenadas del sensor

```
skeleton.Joints[JointType.Head]

foreach (Joint joint in skeleton.Joints) {

if (joint.TrackingState == JointTrackingState.Tracked) {...}
else
if (joint.TrackingState == JointTrackingState.Inferred) {...}

… joint.Position …

}
```

*Tipo, detección y coordenadas