

Resumen

En el presente documento se explica la función de un software que permite simular las características de un pizarrón electrónico. Este utiliza información tridimensional proporcionado por un sensor de movimiento, el cual permite detectar la posición de las articulaciones del cuerpo humano. La aplicación tiene la finalidad de ayudar a un expositor en su presentación audiovisual, con funciones como: avanzar o retroceder, zoom (ampliar o reducir), y pintar. Cabe mencionar que dichas funciones se pueden realizar únicamente con movimientos determinados de las extremidades superiores de un humano, mismos que son interpretados por el sensor debido a la programación que se le ha proporcionado.

El software es apto para el control de una presentación, para el avance y retroceso de la misma, además de realizar acercamientos de la imagen proyectada, así también poder escribir y borrar con solo los movimientos de las manos del usuario.

Para su creación se ha utilizado el sensor Kinect de la consola de videojuegos XBOX 360, el cual se ha programado bajo el lenguaje de desarrollo de C# y la utilización de la librería SDK para el dispositivo.

Índice General

Resumen

1. Introducción	3
2. Planteamiento del Problema	4
3. Justificación	5
4. Objetivo	6
5. Estado del Arte	7
6. Caracterización del Área en que se Participó.....	15
7. Problemas a Resolver Priorizándolos	18
8. Alcances y Limitaciones.....	19
9. Fundamento Teórico	20
10. Procedimiento y Descripción de las Actividades Realizadas	31
11. Resultados, Planos, Gráficas, Prototipos y Programas	43
12. Conclusiones y Recomendaciones	48
13. Bibliografía.....	50
Anexos	51

1. Introducción

En el sector educativo es donde se realizan actividades de retroalimentación visual, es decir una actividad donde el profesor utiliza una herramienta de apoyo para plasmar o simplemente mostrar las ideas o conceptos de un tema en específico con la finalidad de facilitar el aprendizaje en sus espectadores y es a través de este método como se trasmite la información de una manera eficaz, teniendo como consecuencia la generación de conocimiento a terceras personas, pero en ocasiones las clases suelen ser pesadas para el público espectador debido a las formas tradicionales en las que se desempeñan, es decir que el expositor no se apoya de algún elemento con el cual pueda hacer más que una presentación.

Dentro del Instituto Tecnológico de Tuxtla Gutiérrez, en su mayoría maestros y alumnos utilizan las proyecciones audiovisuales para exponer un tema, pero limitados a ir mostrando únicamente la secuencia del mismo sin poder interactuar de manera dinámica con su material de exposición, además se necesita portar un dispositivo que coadyuve en la presentación o en su defecto de una segunda persona que apoye al presentador.

Por lo anterior surge la idea de crear una herramienta de apoyo para que responda a las necesidades de mejorar la exposición de una clase, esto es, Software denominado “Pizarrón Virtual Multifuncional Controlado a Distancia por Sensores de Movimiento para la Presentación de Materiales Audiovisuales”, el cual ayuda al ponente a poder manipular su material por sí solo únicamente con determinados gestos de las manos, además dicho software tiene funciones de zoom y subrayado así como poder trazar líneas, cuadrados y triángulos, cabe mencionar que dichas funciones se realizan únicamente con movimientos de las manos que son leídos con el sensor Kinect e interpretados por el programa principal.

2. Planteamiento del Problema

En la actualidad existen métodos para dar a conocer un tema ante un público, desde láminas que muestran información de manera gráfica, pizarrones para poder desenvolver un tema y las proyecciones audiovisuales con ayuda de un equipo de cómputo y un video proyector, siendo esta última la más utilizada debido al crecimiento de la tecnología, además de hacer más atractiva una presentación.

Sin embargo durante la presentación del material audiovisual, es necesario de una segunda persona que se encargue del avance o retroceso de la presentación a menos que el ponente sea quien controle su material, causando con ello que pierda el contacto con sus espectadores por detenerse un momento para decirle a su colaborador que proceda o en su defecto el mismo expositor tenga que hacerlo.

Además estas formas tradicionales de explicar un tema debe contener un gran sentido de creatividad para tener la atención del público, debido a que el expositor no interactúa de manera dinámica con su material de exposición, es decir que se encuentra limitado a ir mostrando la secuencia de su tema a través de las proyecciones.

3. Justificación

Muchas veces se ha perdido la continuidad de una presentación audiovisual debido a que el expositor se debe apoyar de una segunda persona para que controle su presentación o en su defecto que sea el mismo quien sobrelleve su exposición.

Hoy en día existen dispositivos que ayudan en la manipulación remota de una presentación audiovisual, dispositivos como: el mouse conectado vía Bluetooth, un apuntador, incluso los teléfonos móviles han desarrollado aplicaciones que ayudan en dicha actividad, mas sin embargo para su utilización mantienen ocupada la mano del ponente, y su uso se limita al avance o retroceso según sea el caso.

La creación de un software que sea capaz de interpretar los movimientos corporales del usuario por medio del sensor Kinect, ofrece una alternativa adecuada cuando se trata de manipulación remota de una presentación, pues a través del sensor se pueden interpretar determinados movimientos (gestos), los cuales pueden ser interpretados para el control del material de exposición, al mismo tiempo de poder interactuar dinámicamente con funciones como: zoom (acercar, alejar), pintar y trazos de figuras geométricas.

Alcanzando lo antes mencionado habrá un mejor dominio del material, gracias a las herramientas o utilerías que se acaban de mencionar, en consecuencia el expositor ya no dependerá de terceras personas que le ayuden a controlar la presentación, estará enfocado a personas profesionales, docentes, alumnos y para todos aquellos que quieran impartir un tema ante un grupo.

4. Objetivo

General

Desarrollar una aplicación capaz de simular un pizarrón virtual con la ayuda de un video proyector y el sensor de movimiento, el cual permita la presentación de materiales audiovisuales.

Específicos

1. Analizar los requisitos funcionales del sistema.
2. Diseñar el sistema en base al método de desarrollo por prototipos.
3. Mostrar en la pantalla del sistema los movimientos de ambas manos del usuario, así como los de la cabeza.
4. Calibrar el dispositivo para que solo reconozca los movimientos de la persona que este exponiendo.
5. Avanzar la presentación con un movimiento específico de la mano derecha, mismo que será interpretado por el sensor.
6. Retroceder la presentación con un movimiento específico de la mano izquierda.
7. Simular la escritura, borrado y el zoom únicamente con los movimientos de las manos.
8. Agregar diapositivas en formato JPG.
9. Hacer barridos con ventanas dinámicas que se diseñaran y programarán para el pizarrón.
10. Guardar las presentaciones (pizarrones) para que el usuario pueda ver lo que explico anteriormente.

5. Estado del Arte

Con el fin de poder conocer más a fondo el proceso de manipulación a distancia y aprovechar la tecnología de sensores, se realizó la búsqueda por diferentes medios para recaudar información. Se encontró que existen diferentes herramientas de control de la información que utilizan distintos tipos de dispositivos, siendo estos los más similares al proyecto de residencia de este documento.

La Mesa Mágica: Pizarra Basada en Visión por Computadora Aumentada para Reuniones Creativas (The Magic Table: Computer-Vision Based Augmentation of a Whiteboard for Creative Meetings)

La mesa mágica es un pizarrón avanzado el cual sirve como apoyo en reuniones creativas. La mesa mágica usa la visión por computadora para la digitalización y especialmente la organización de los textos y dibujos en la superficie. La información digital se organiza a través de la manipulación de símbolos (pequeños discos de plástico).

La interpretación consiste en gestos rápidos y fáciles de aprender que soportan múltiples usuarios simultáneos reconociendo las dos manos de cada uno. La aplicación tiene dos principales componentes: el modelo de color en función del token tracker y el escáner basado en técnicas de mosaicos. [1]

Sistema de la mesa mágica

Configuración de Hardware

Los principales componentes de la Mesa Mágica es una calibración de la pizarra con sus lápices y goma de borrar, un proyector de vídeo que estará proyectando a la pizarra, y una cámara digital conectada a la computadora.

El proyector es un video DMD 1024x768 proyector. La cámara es un equipo controlado por Pan / Tilt / Zoom PAL (768x576 a 25 fps.) cámara de vídeo. El equipo de la computadora es un estándar de bi-procesador equipado con gráficos (para la salida del proyector) y una tarjeta de vídeo (Por entrada de cámara).

El sistema fue experimentado con orientación de la pizarra en forma vertical y posteriormente en horizontal esta última orientación favorece porque permite más personas que interactúan al mismo tiempo.

La mesa permite que al menos cuatro personas (una en cada lado de la mesa) puedan trabajar al mismo tiempo sin obstruir mutuamente actividad. Así, el nombre del sistema tiene la CAPS de "pizarra mágica", la "mesa de magia". Una foto de la Mesa Mágica se muestra en la figura 1.

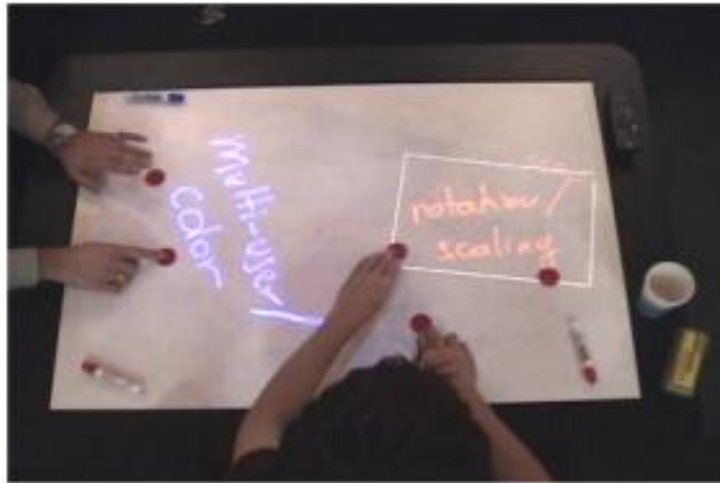


Figura 1: La mesa mágica en uso, dos usuarios Interactuando con ambas manos

La configuración de la mesa mágica ha sido recientemente mejorada con la adición de una segunda cámara de video, como se muestra en figura 2, a fin de permitir el funcionamiento en paralelo de exploración y servicios de seguimiento. Sólo una cámara tiene que ser dirijible: una dedicado a la exploración.



Figura 2: Posición de las cámaras y el proyector para la utilización horizontal

La cámara de seguimiento tiene un fijo campo de visión establecido para abarcar toda la imagen proyectada sobre la mesa. Se ha estado trabajando en el seguimiento ya sea con dedos o fichas de plástico, aunque hasta hace muy poco el seguimiento de los dedos no era lo suficientemente robusta como para su uso en el sistema.

Interacción de la Mesa Mágica

Los usuarios pueden escribir, dibujar y borrar en la tabla mágica como en una pizarra normal. Cuando se necesita acceder a los servicios digitales más avanzados, se escanean los parches de los usuarios. Un parche es un rectángulo que contiene una copia digital de los códigos impresos.

Una vez creado, el parche y su contenido se pueden mover alrededor de la superficie. Los videos del sistema en uso están disponibles para su descarga desde la web (<http://iihm.imag.fr/demos/magicboard/>).

Los usuarios interactúan con la mesa mágica mediante la manipulación de las fichas. En el diseño de la interacción, se intenta que coincida con la sencillez y la rapidez de trabajo de forma regular como una pizarra.

Se define un pequeño conjunto de gestos que permiten tres operaciones fundamentales de los parches, figura 3.

a) Creación, b) Eliminación y c) Transformación.

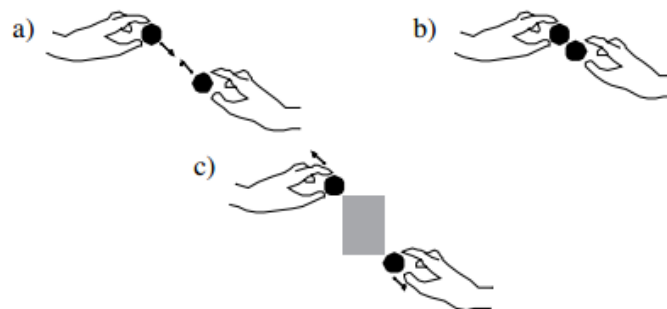


Figura 3: Conjunto de gestos definidos para la pizarra mágica.

Pizarrón Virtual: Sistema de Iteración Hombre Maquina de bajo costo multi-tacto (Virtual Board: A low cost Multi Touch Human Computer Interaction System)

Este pizarrón es un prototipo que utiliza un sistema de control remoto a través de computar la técnica de infrarrojos para detectar múltiples puntos infrarrojos en un lugar, mostradas por los marcadores activos.

El sistema utiliza como marcador de activos, dos guantes con infrarrojos en la punta del dedo índice y un botón en el extremo del dedo medio para que con el pulgar se puedan cambiar de infrarrojos.

Los marcadores han sido diseñados por el movimiento natural de la mano del usuario, dando a los usuarios más fácil manipulación.

El sistema puede manipular imágenes para acercar, alejar el zoom, la rotación y la traducción y utiliza dos marcadores de activos, la cámara web detecta estos marcadores después que el software tiene la posición de los marcadores y con cálculos matemáticos muestra la manipulación de imágenes.

Dispositivos utilizados para el control remoto.

Varios dispositivos como el Kinect y Wiimote, que ofrecen la tecnología por control remoto para la computadora se puede encontrar en el mercado, cada dispositivo utiliza diferentes técnicas para manipular el equipo con sus respectivas características y limitaciones. [2]

Pizarra interactiva eficiente de bajo costo (Low-Cost Efficient Interactive Whiteboard)

Pizarra interactiva eficiente de bajo costo, utiliza la profundidad de fusión y de vídeo información proporcionada por una cámara de profundidad de bajo costo, es capaz de detectar y rastrear los movimientos del usuario.

El Dispositivo pizarrón interactivo (IWB) ha sido ampliamente utilizado en ambientes de enseñanza o en las salas de conferencias. IWB presenta varias ventajas con respecto a una pizarra convencional: permite guardar contenido de la pizarra, la integración de otros contenidos multimedia (videos, e imágenes) y ayuda a captar mayor la atención de la audiencia con la interactiva herramienta de dibujos. IWB puede estar basada en diferentes tecnologías: superficies activas resistiva o capacitiva, tecnologías ópticas sobre la base de los sistemas de láser o dispositivos activos lápiz; ultrasónicos transmisores, aplica generalmente en las esquinas de pizarra.

A pesar de que estas tecnologías garantizan un seguimiento eficaz de las interacciones del usuario con el sistema de la pizarra, por lo general son caros y no puede ser fácilmente adaptada o movido en diferentes ambientes.

Además, estos sistemas sólo pueden detectar la interacción del usuario con la pizarra apoyándose en contacto con la superficie, y no contempla la captura de gestos a mano alzada.

En este trabajo se presenta un sistema de bajo costo (IWB) basado en un sensor de profundidad comercial de bajo costo. La principal ventaja de este sistema es su alto nivel de portabilidad: puede ser integrados con los equipos comunes, como monitores o proyector y equipos fuera de la plataforma. Además, también permite integrar el reconocimiento de gestos, por lo tanto, amplía las posibilidades de fácil interacción con la pizarra.

El esquema del sistema IWB propuesto se muestra en la figura 4. La cámara profundidad adquiere la información de vídeo y la profundidad de la escena y lo envía a una unidad de procesamiento (PU) que combina la información de profundidad y de vídeo para extraer la silueta del usuario y para detectar y rastrear los movimientos de la mano. Entonces, ya extraídos los movimientos se utilizan para actualizar el contenido de pizarra.

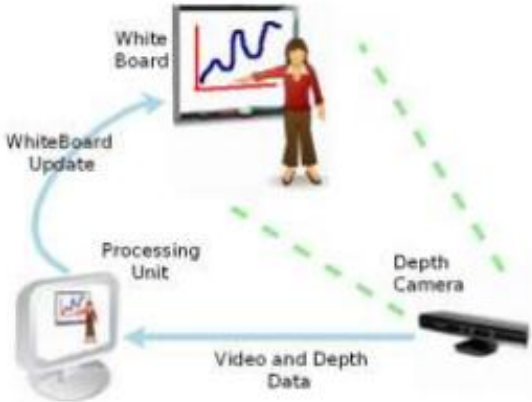


Figura 4: Esquema del sistema Pizarra Iterativa

Las imágenes se representan por un proyector conectado a la PU (que es una PC fuera de la plataforma, ya que no es hardware especializado necesario). El papel de la PU es fundamental para que funcione correctamente la gestión de vídeo y los datos de profundidad y para garantizar la fiabilidad el seguimiento y el reconocimiento de gestos.

El diagrama de bloques de tareas de procesamiento de la unidad (la UP) se muestra en la figura 5:

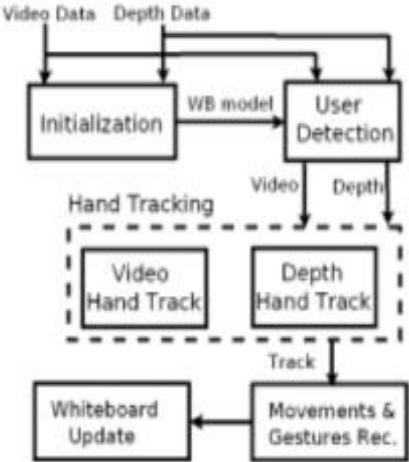


Figura 5: Diagrama de bloques para el funcionamiento de la pizarra

El sistema se inicializa con los datos de vídeo y la profundidad para identificar correctamente la zona en donde están situadas las imágenes en la pizarra. Además, la mezcla de algoritmos gaussianos, se emplea para obtener una fiable estimación del mapa de profundidad de la pizarra y modelo de color. Cuando el sistema se inicializa, el cuerpo completo del usuario se detecta mediante el uso de una profundidad de segmentación basado en primer plano ligero al algoritmo (disponible en las librerías).

Los ensayos han mostrado que este algoritmo es lo suficientemente robusto como para identificar la región ocupada por el usuario cuando se lleva a cabo el rastreo de las manos.

El análisis combinado de datos de profundidad y video permite realizar de manera eficiente el seguimiento de la mano y permite la interacción nueva entre el usuario y la pizarra gracias a la detección de gestos a mano libre. La información de profundidad se utiliza para detectar diferentes gestos de los usuarios que se emplean para activar específicas funciones de la pizarra y eventos (es decir, el movimiento que se emplea para limpiar la pizarra o haga clic para seleccionar las diferentes herramientas de pintura).

La información de vídeo se utiliza para rastrear la interacción mano / superficie, que no puede ser discriminado por usar sólo la información de profundidad. Por lo tanto, el módulo de seguimiento de la mano propuesto se basa en la combinación de dos seguidores independientes: módulo de seguimiento a base de la profundidad que hace uso del algoritmo eficiente presentado en [OpenNI libraries. www.openni.org]; y de vídeo basado en el seguimiento que se realiza con el algoritmo KLT [B.D. Lucas and T. Kanade. "An iterative image registration technique with an application to stereo vision," In *Proceedings of the 7th international joint conference on Artificial intelligence (IJCAI'81)*, Vol.2.1981.], que se inicializa con los puntos extraídos desde la región de la mano basado por el seguidor profundidad.

Los seguidores se combinan teniendo en cuenta la distancia relativa entre el usuario y la pizarra: cuando el usuario está cerca de la Pizarra virtual e interactúa con la superficie de la base de vídeo los resultados del seguidor se utilizan para actualizar el contenido de pizarra. Por el contrario, cuando el usuario está lejos de la superficie de la pizarra, el seguidor de profundidad se utiliza para detectar gestos de movimientos con la mano libre.

Además, la información de vídeo puede ser útil para refinar los contornos de la silueta de la mano obtenidos con el mapa de profundidad que suele ser muy ruidosa cerca de los bordes, por lo tanto, mejora la calidad de seguimiento, gracias a una mayor precisión de estimación del centro de la masa de la mano. La mano que resulte en posiciones de escritura se analiza para detectar los movimientos correspondientes a determinados gestos de la mano libre o de dibujo. [3]

6. Caracterización del Área en que se Participó

El instituto tecnológico de Tuxtla Gutiérrez, se encuentra ubicado en Carretera Panamericana Km. 1080, cuenta con la carrera de Ingeniería en Sistemas Computacionales.



Inicia actividades en enero de 1991. Acreditado por CONAIC desde el 17 de diciembre de 2012. Cuenta con el cuerpo académico "Tecnología Computacional para el Desarrollo Regional, ITTUXG-CA-4" (desde 2011). Programa de titulación integral consolidado. [4]

El área de esta carrera se encuentra ubicada en su mayoría en el Edificio D-1 planta alta, ofrece sus servicios a los alumnos en tiempo corrido de 8:00 a 21:00 horas. La carrera cuenta con: área administrativa, sala de juntas, laboratorios de cómputo, laboratorio Cisco y aulas.

Misión

Formar de manera integral profesionales de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

Visión

Ser una institución de excelencia en la educación superior tecnológica del sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

Valores

- El ser humano
- El espíritu de servicio
- El liderazgo
- El trabajo en equipo
- La calidad
- El alto desempeño

Ingeniería en sistemas computacionales

Objetivo general de la carrera

Formar profesionales capaces de diseñar y desarrollar sistemas de software que les permita proporcionar el fortalecimiento de la tecnología nacional; administrar proyectos de desarrollo de software y especificar y evaluar configuraciones de sistemas de cómputo en todo tipo de organizaciones donde se utilicen sistemas computacionales.



Quehacer profesional

Las necesidades actuales de desarrollo hacen necesario que en el estado se cuente con profesionistas, capaces de diseñar, crear y dar mantenimiento a modernos e innovadores productos. La transformación de los sistemas computacionales ha sido muy rápida; la combinación de software y hardware con tecnología de comunicación, de interfaz con el medio ambiente y otras han permitido el desarrollo de poderosos sistemas. Se espera, que los tres sectores productivos del estado apoyen aún más en los sistemas computacionales, telecomunicaciones y redes.

Campo de trabajo

El ingeniero en sistemas computacionales puede prestar sus servicios de manera independiente, trabajar en todo tipo de empresas industriales, de servicios, públicas o privadas como podrían ser industrias extractivas, de transformación, empresa de servicios, comerciales, exportadoras, de distribución, de desarrollo informático, de inversión o crédito.

Organigrama

Departamento de Ingeniería en Sistemas Computacionales



Croquis de ubicación

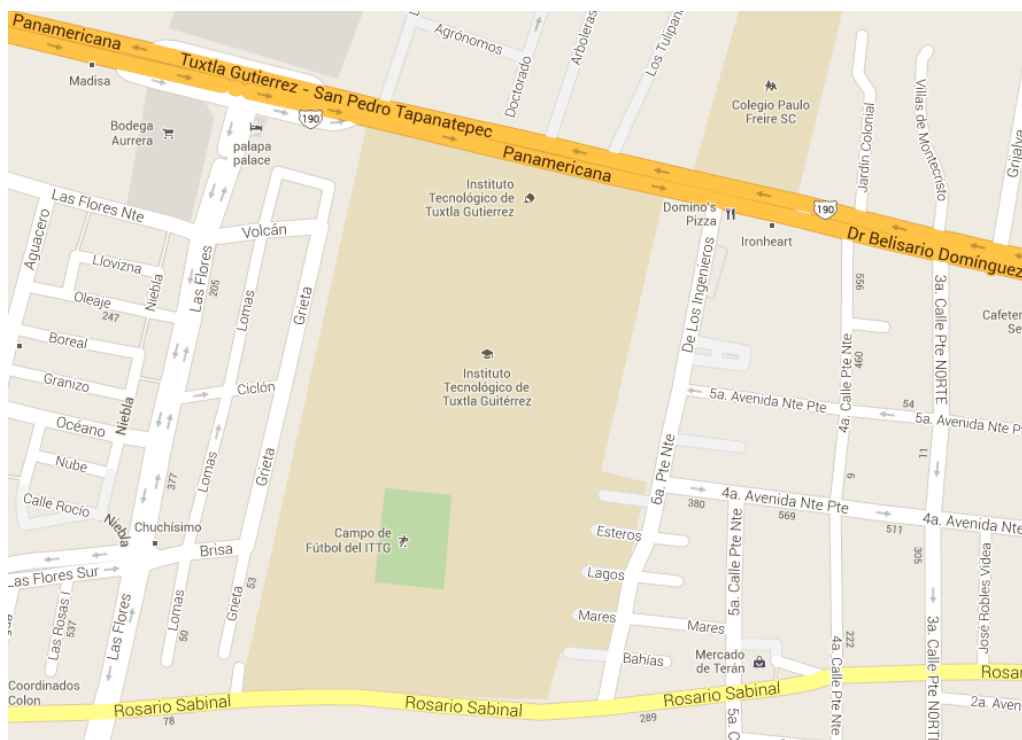


Figura 6: Mapa de ubicación del instituto Tecnológico de Tuxtla Gutiérrez.

Domicilio y teléfonos de la dependencia

Carretera Panamericana Km.1080 C.P.29050, Tuxtla Gutiérrez, Chis. Tels. (961) 615 0461 y 615 1687.

7. Problemas a Resolver Priorizándolos

Conforme a la definición de los objetivos específicos antes mencionados, se hace una lista de los problemas que se atacaran de manera prioritaria, o los que se han considerado que son los elementos claves para el desarrollo del proyecto.

1.- Interactuar con los movimientos de ambas manos para manipular remotamente los objetos que se esté proyectando, todos los elementos se podrán manejar mediante gestos sin la necesidad de tener un dispositivo sujeto a la mano, los movimientos que se harán serán naturales y fáciles de recordar.

El Kinect a través de sus sensores toma estas imágenes y las hace pasar por una serie de filtros para que determine qué es una persona y qué no lo es. De esta manera se procesa la imagen buscando patrones, como pueden ser el tamaño del cuerpo humano, cantidad de extremidades que éste tiene, etc. Mediante este proceso es capaz de distinguir incluso entre hombro derecho y hombro izquierdo. Todo ello para evitar que objetos del lugar donde nos encontramos sean reconocidos como otros jugadores.

2.- Simular la escritura y borrado en un pizarrón, este es el punto importante que destaca el proyecto puesto que desde un principio se mencionó la necesidad de poder simular la escritura y trazo de figuras a través del software y la ayuda de sensores.

3.- Hacer zoom en algún punto en específico.

Como se ha venido resaltando la clave de la realización del proyecto, es además de poder avanzar o retroceder la presentación con los movimientos de las manos, mediante el software poder interactuar con los objetos que se estén proyectando, para hacer la presentación más emotiva, teniendo como consecuencia que el público espectador preste más atención al expositor.

8. Alcances y Limitaciones

Alcances

- 1.- El pizarrón virtual puede controlar en su totalidad de manera remota, en un área entre 2 y 3 metros de frente al sensor Kinect.
- 2.- Controlar diapositivas que estén en formato JPG, para poder lograr esto, estas tienen que estar cargadas en un proyecto que se puede crear en el mismo programa.
- 3.- Adelantar y atrasar las diapositivas, haciendo gestos con los brazos, avanzar (gesto con la mano derecha) y atrasar (gesto con la mano izquierda).
- 4.- Zoom digital utilizando las dos manos posicionándolos frente del sensor y posicionando ambas manos en el centro del pecho desplazando hacia afuera logrando hacer la ampliación.
- 5.- Para el caso de reducir el Zoom, se tiene que hacer el mismo procedimiento en forma inversa como se explicó en el punto 4.

Limitaciones

- 1.- Se necesita un sensor que detecte movimientos, un video proyector y una computadora de alto rendimiento para que pueda ejecutarse el programa sin problema alguno.
- 2.- El programa puede simular la escritura y borrado a mano, se limita a borrar todo lo que se ha pintado en el pizarrón, es decir que no se puede borrar por partes lo que sea escrito.
- 3.- El software solo puede funcionar en áreas donde no exista la posición directa del sol, es decir que para el buen funcionamiento se necesita de un área con poca iluminación, se recomienda un aula con cortinas.

9. Fundamento Teórico

Microsoft Kinect

Kinect es un dispositivo de control de movimiento que fue inicialmente creado para jugar con la videoconsola XBOX 360 sin la necesidad de ningún tipo de mando o controlador. De este modo, el jugador hace uso de su propio esqueleto para interactuar con el juego, creando una experiencia de usuario más realista y dando la opción al jugador de ser el protagonista de la historia.

Aunque el sensor Kinect ha sido desarrollado oficialmente por Microsoft, el diseño y la tecnología fue creada por la empresa israelí PrimeSense. Kinect fue anunciado por primera vez el 1 de Junio de 2009 en la Electronic Entertainment Expo 2009 bajo el nombre de “Project Natal” y posteriormente, en noviembre de 2010 sale al mercado como accesorio de la XBOX 360, en la figura 7 se muestra el sensor Kinect.

Con el lanzamiento de Kinect surgió una gran oportunidad para científicos, aficionados e inventores que comenzaron a utilizar y “piratear” Kinect para construir nuevas aplicaciones y conseguir aprovechar el potencial del que dispone este periférico. Tanto es así, que al poco de su lanzamiento, la empresa Adafruit ofreció una recompensa a la primera persona que consiguiera hackear Kinect (Adafruit, 2010), siendo el ganador el español Héctor Martín. Poco más tarde, la empresa PrimeSense lanzó el primer SDK no oficial para Kinect.



Figura 7: Sensor Kinect de Microsoft

La particularidad de este periférico, es la posibilidad de interacción entre videoconsola y usuario sin necesidad de tener contacto físico con ningún controlador de videojuegos.

Hardware: Partes Fundamentales

El sensor Kinect dispone de cuatro partes fundamentales que podemos ver en la siguiente imagen y que se explican a continuación:



Figura 8: Componentes del Sensor Kinect

1. **Sensores 3D de profundidad.** Los sensores tridimensionales hacen un seguimiento del cuerpo dentro del área del juego. Dispone de un proyector de profundidad (retícula izquierda) y un sensor de profundidad (retícula derecha), que calculan la distancia en función del tiempo que tarda en reflejar la luz.
2. **Cámara RGB.** Una cámara RGB (Red, Green, Blue) que ayuda identificar y captar imágenes y videos con una resolución máxima de 640x480 a 30 fps.
3. **Array de micrófonos.** Se usa un conjunto de cuatro micrófonos en el borde frontal inferior del sensor Kinect para el reconocimiento de voz.
4. **Inclinación motorizada.** Se trata de un impulso mecánico en la base del sensor Kinect que inclina de manera automática el sensor hacia arriba o abajo según sea necesario hasta un máximo de 27°.

Además de estas partes que podemos observar en el propio sensor, Kinect dispone de lo siguiente:

Memoria RAM de 512 MB.

Acelerómetro, para estabilizar la imagen cuando se mueve.

Ventilador, que no se encuentra encendido continuamente para no interferir con los micrófonos. [5]

Sistema de rastreo del Kinect

Este cuenta con un emisor de infrarrojos y dos cámaras. El emisor de infrarrojos llena la sala de cientos de puntos. Los píxeles que Kinect recibe como ruido en el receptor de infrarrojos son convertidos en una escala de colores, haciendo que los cuerpos, dependiendo de la distancia, se capturen como rojos, verdes, azules hasta llegar a tonos grises, que representan a objetos muy lejanos.

El software toma estas imágenes y las hace pasar por una serie de filtros para que Kinect determine qué es una persona y qué no lo es. De esta manera se procesa la imagen buscando patrones, como pueden ser el tamaño del cuerpo humano, cantidad de extremidades que este tiene. Esto se hace con el fin de evitar que el sensor interprete como una persona los objetos que se encuentren en el área de visión del Kinect.

Como se puede apreciar en la Figura 9, el Sensor dispara los rayos infrarrojos formando una maya en la persona, y con la ayuda de la cámara de profundidad se puede observar los puntos que toca a cada parte de la persona que se posiciona frente del sensor.



Figura 9: Sistema de rastreo infrarrojo

Una vez que la información es separada del resto, es decir que el sensor ha identificado a una persona, se convierte cada identificación del cuerpo en un esqueleto con articulaciones móviles. Además, está precargado con una base de datos de 200 poses, para llenar los espacios en caso de hacer un movimiento que obstruya la visión de la cámara (como echar los brazos hacia atrás).

El sistema completo es capaz de seguir hasta a 6 personas con 2 jugadores activos, y monitorizar 20 articulaciones por cada jugador.

Como factor negativo, nos encontramos que las manos se toman como un objeto agrupado en lugar de capturar los dedos por separado.

En noviembre de 2010, Industrias Adafruit ofreció una recompensa para un controlador de código abierto para Kinect. El 10 de noviembre de ese mismo año, se anunció al español Héctor Martín como el ganador, que en dos horas usó métodos de ingeniería inversa con Kinect y desarrolló un controlador para GNU/Linux, que permite el uso de la cámara RGB y las funciones de profundidad. [6]

¿Qué es una cámara RGB?

Una cámara RGB (ver Figura 10) proporciona los componentes básicos de tres colores (rojo, verde y azul) en tres hilos diferentes. Este tipo de cámara a menudo utiliza tres sensores CCD independientes para adquirir las tres señales de color. Las cámaras RGB se utilizan para la adquisición de la imagen muy precisa del color. [7]



Figura 10: Cámara RGB del sensor

¿Cómo funciona el sensor de profundidad de Kinect?

La profundidad de los objetos es captada por la cámara gracias al rebote de los haces laser por el campo de juego, creando así un “campo de profundidad” que permite al sensor diferenciar entre los objetos estáticos de la sala y las personas utilizándolo. Este “campo de profundidad” consiste, básicamente, en que Kinect recibe este haz de luz como infrarrojos que varían en mayor o menor grado de color dependiendo de la distancia a la que se encuentran del sistema. De este modo, los cuerpos aparecen como rojo, verde, etc.; y los objetos más alejados aparecen en gris.

Con los datos obtenidos en esta imagen, el software aplica una serie de filtros para que Kinect pueda saber qué es una persona y qué no, basándose en una serie de directrices como “una persona tiene dos brazos y dos piernas”.

Una vez tenemos la información ordenada, se identifican las partes del cuerpo y crea un esqueleto en movimiento. Kinect tiene unas 200 posturas precargadas, de manera que se puedan llenar los espacios en blanco en caso de que se realicen movimientos que obstruyan la visión de su esqueleto. El principal inconveniente que podemos encontrar, es que los dedos no se asignan de forma individual en el esqueleto, impidiendo con ello una serie de movimientos.

Todo esto es realizado por el sistema continuamente a 30 fps.

En la figura 11 se muestra el sensor de profundidad del sensor Kinect.



Figura 11: Sensor de profundidad



Figura 12: Sensor de profundidad (infrarrojos) en funcionamiento

¿Cómo es posible extraer información de la profundidad de la única imagen de infrarrojos del patrón de especificaciones?

Para la obtención de los datos de profundidad, el sensor Kinect hace uso de la triangulación estéreo gracias a las posiciones en la que se encuentran el láser infrarrojo y la cámara de profundidad.

La Triangulación Estéreo requiere dos imágenes para obtener la profundidad de cada punto (especificaciones) ver figura 13. Aquí está el truco: en realidad no hay una, sino dos imágenes. Una imagen es lo que vemos en la foto, la imagen de las especificaciones capturados por el sensor de infrarrojos. La segunda imagen es invisible, es un patrón de cableado de las especificaciones que se proyectan con láser. La segunda imagen debe ser codificada en la lógica del chip.

Esas imágenes no son equivalentes, hay una cierta distancia entre el láser y sensor, de modo que las imágenes corresponden a las posiciones de cámara diferentes, y que permite el uso de la triangulación estéreo para calcular la profundidad de cada especificación. El Sensor Kinect emplea la visión estereoscópica para la obtención de la profundidad de los objetos que encuentra en su área de visión.

Visión Estereoscópica

La visión humana es estereoscópica debido a que la posición frontal de los ojos en la cara hace que el ser humano disponga de este tipo de visión. Los ojos son los encargados de obtener imágenes desde puntos de vista diferentes, dentro de los cuales existen puntos en común y otros puntos que no tienen relación unos con otros.

La visión estereoscópica es la fusión de imágenes que son captadas sobre un objeto desde dos perspectivas distintas. Por lo tanto la vista humana es estereoscópica debido a que el hombre dispone de ojos que forman imágenes captadas ligeramente distintas en perspectivas al mundo.

La diferencia relativa en posiciones de los objetos se denomina disparidad y está provocada por las diferencias en distancia del observador. El cerebro es el responsable de analizar las imágenes relacionando los puntos idénticos, produciendo esa sensación tridimensional que se tiene con los objetos cercanos. El cerebro es capaz esta disparidad y usarla para estimar las distancias relativas que nos separan de los objetos.

Estéreo Visión

El termino estéreo visión es aplicado cuando se realiza la recuperación de la estructura tridimensional de una escena u objeto usando dos o más imágenes adquiridas por cámaras [Fofi]. Cada adquisición es realizada desde un punto de vista diferente en el espacio. Las imágenes pueden ser obtenidas usando múltiples cámaras o moviendo la cámara. [8]

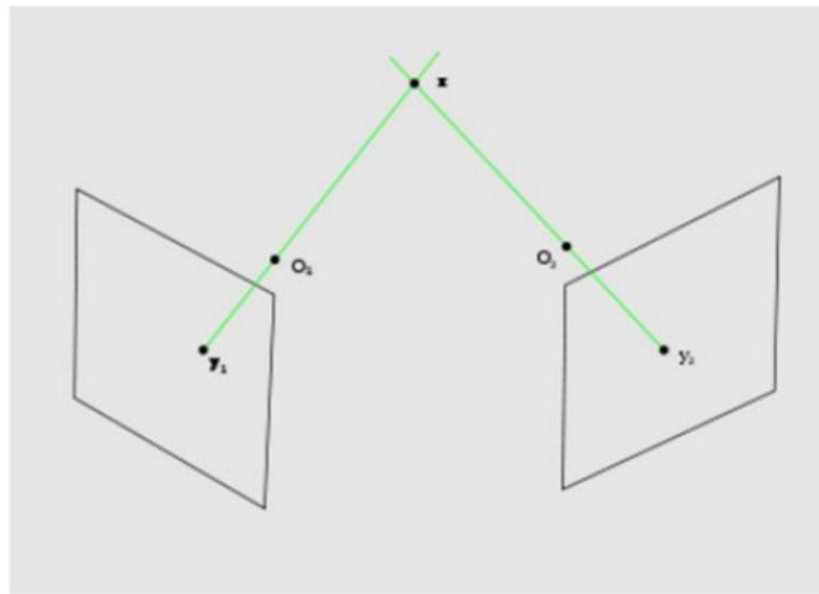


Figura 13: Triangulación estéreo.

La diferencia aquí es que la segunda imagen es "virtual", la posición del segundo punto Y2 ya está codificado en la memoria. Debido a que el láser y el sensor están alineados hacen la tarea aún más fácil: lo que uno tiene que hacer es medir el desplazamiento horizontal de la especificación de la primera imagen respecto a la posición codificada (después de la corrección de distorsión de la lente).

Los Patrones Pseudoaleatorios hacen más fácil el ajuste de las especificaciones de dos imágenes, ya que cada determinación se da de un vecindario local diferente. Puede ser llamado "luz estructurada" de sensor. [9]

Sistema de luz estructurada

El sistema de luz estructurada se aplica en la visión estereoscópica. En este sistema el emisor es un video proyector o láser y el receptor una cámara CCD. Cuando se inicia una digitalización el proyector emite sobre el objeto una serie de imágenes sobre la superficie de un objeto y que son registradas por la cámara CCD.

Este sistema se caracteriza por ser un método directo y activo. Los sistemas de luz estructurada se basan en estudiar la deformación que sufre un patrón de luz al ser reflejada por cualquier objeto. Este es el problema principal de este tipo de herramienta debido a que se necesita de un tipo de luz concentrada en un punto que esté compuesto por ondas de la misma frecuencia para que el haz de luz no se deforme.

Todo el funcionamiento del Kinect depende de controladores que hacen que se comunique con la computadora, para poder usar las funciones existen librerías especiales que se desarrollaron, estas se encuentran en la página oficial de Microsoft, el paquete completo para desarrollo se denomina SDK.

¿Qué es el SDK?

El SDK de Windows para Kinect es una herramienta de programación para desarrolladores de aplicaciones. Las capacidades ofrecidas por el dispositivo de Microsoft Kinect está abierta para las comunidades académicas y entusiastas que se dediquen a desarrollar aplicaciones ejecutadas en equipos con sistema operativo Windows 7.

El SDK incluye controladores, API rica para los flujos de los sensores sin procesar y el seguimiento del movimiento humano, los documentos de la instalación, y recursos materiales de Kinect proporciona capacidades para los desarrolladores que crean aplicaciones con C++, C# o Visual Basic mediante Microsoft Visual Studio 2010. [10]

El SDK de Windows contiene un conjunto de herramientas, ejemplos de código, documentación, compiladores, encabezados y las bibliotecas que los desarrolladores pueden utilizar para crear aplicaciones que se ejecutan en Microsoft Windows. Se puede utilizar el SDK de Windows para escribir aplicaciones que utilizan los nativos (Win32/COM) o administrador (.NET Framework).

¿Qué significa el SDK para Windows 7 y .NET Framework4?

Sistemas Operativos: Puede instalar este SDK y/o crear aplicaciones para Windows 7, Server 2008 R2, Server 2008, XP SP3, Vista y Windows Server 2003 R2.

Arquitectura: se puede instalar este SDK y/o crear aplicaciones para los chipsets de la plataforma X86, X64 y IA64(Itanium).

NET Framework: Puede utilizar los recursos SDK para crear aplicaciones destinadas a versiones NET Framework 2.0, 3.0, 3.5, 4.

Visual Studio: puede utilizar los recursos de este SDK con las versiones de Visual Studio 2005, 2008 y 2010, incluyendo las ediciones Express. (No todas las funciones de trabajo con todas las versiones de Visual Studio. Por ejemplo, no se puede utilizar la red. Herramientas de Visual Studio 2008.)

Windows SDK Requisitos de espacio en disco

La instalación completa del SDK requiere menos de 600 MB, es un tercio del tamaño de la RTM de Windows 7 SDK, que se instala rápido y ocupa menos espacio. [11]

El SDK de Kinect para Windows Incluye lo siguiente:

- Drivers para el uso de dispositivos de sensores kinect en un equipo con Windows 7, Windows Embedded Standard 7 y Windows 8 Developer Preview (solo aplicaciones de escritorio).
- API y las interfaces del dispositivo.
- Documentación técnica.
- Ejemplos de código fuente.

10. Procedimiento y Descripción de las Actividades Realizadas

Para el desarrollo de este Software se utilizó el modelo de construcción de software basado en prototipos.

Cabe mencionar que el método elegido es el más adecuado ya que la creación de prototipos es una forma de desarrollo de software muy completa y permite desarrollar aplicaciones en las que son necesarias modificaciones o mejoras durante las pruebas que se realicen, el modelo es de suma importancia en ocasiones en las que no se tienen definidos los límites o se tiene una idea muy simple de las funcionalidades que puede tener un software. En nuestro caso se decide utilizar el modelo debido a que las tecnologías utilizadas para el desarrollo del proyecto se encuentran en pleno descubrimiento.

La Figura 14 muestra las etapas de las cuales consta el paradigma de construcción de prototipos.



Figura 14: Modelo de construcción de Prototipos.

Comunicación

Es el proceso en el cual el desarrollador establece contacto con el cliente para definir los requerimientos del software, para nuestro caso contamos con los objetivos por alcanzar, los objetivos se describen en este documento en el apartado Objetivos.

En esta etapa se mantiene comunicación con la Jefe del departamento de Sistemas Computacionales para llegar a acuerdos en el paso con la recolección de requerimientos.

Recolección de requerimientos

Se han recolectado todos los requerimientos a necesitar para la realización del proyecto. Los elementos necesarios servirán para realizar el primer prototipo para su posterior análisis y búsqueda de refinamientos para su mejora.

Requerimientos necesarios:

Hardware

Sensor de movimiento: se necesita el sensor Kinect de Microsoft.

Dicho sensor requiere de cables para su instalación y también necesita de controladores que se descargan en línea.

Computadora de alto rendimiento: se necesitó de una máquina que tuviera las características mínimas como: Procesador de doble núcleo, 4 Gb de memoria RAM, tarjeta de video de 512 Mb, sistema operativo Windows 7.

Video proyector: Una fuente para poder proyectar lo que se realice en el programa.

Estos requerimientos de hardware son vitales para que pueda funcionar el prototipo.

Software

Librerías: se hace una búsqueda exhaustiva donde se llegó a la conclusión de que la mejor librería o kit de desarrollo para la elaboración del prototipo es el SDK de Kinect, ya que es el mas completo, donde se pueden sacar clases y métodos para que se puedan implementar en el programa que se desarrollará más adelante.

Cabe aclarar que el SDK de Kinect provee de funciones básicas, pero esta carece de poder tener incorporados métodos para reconocer gestos.

La plataforma de desarrollo: El proyecto se realizara en la plataforma .Net, en C# con aplicación WPF.

Esta plataforma es muy completa y funciona en la mayoría de los sistemas operativos Windows.

Requerimientos de funcionalidad

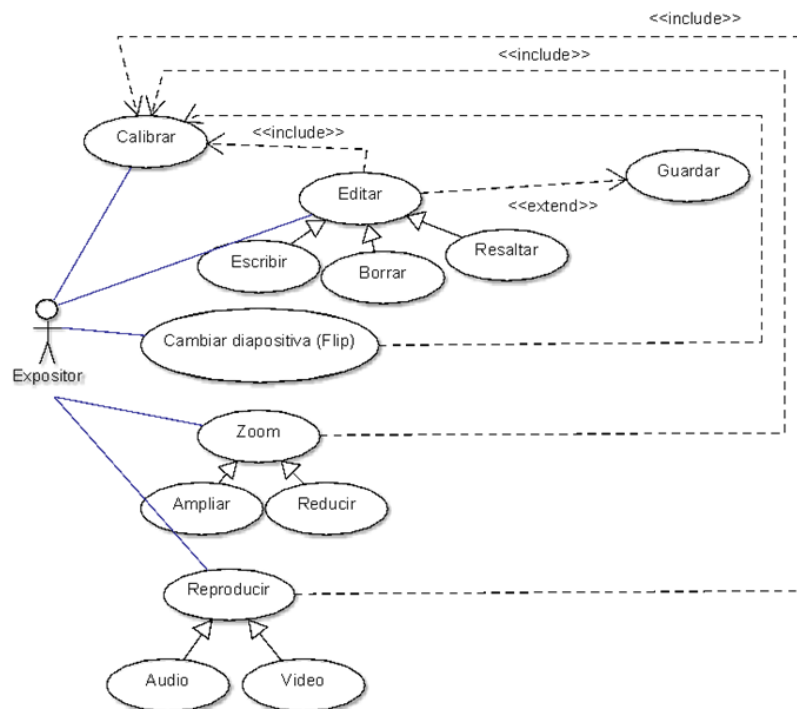


Figura 15: Diagrama de casos de uso

En el diagrama de casos de uso se especifican los principales requerimientos que serán necesarios para el que proyecto pueda funcionar correctamente, en este caso al actor principal se le ha nombrado Expositor. Y se ha documentado en tablas con formato de descripción de los casos de uso que describen las acciones que el usuario puede realizar con el sistema. La siguiente tabla muestra un ejemplo de documentación de casos de uso del sistema.

Documento de descripción de caso de uso.		
Elaborado por: José Antonio Sánchez Alfonso, Leonardo Cabrera García.		Fecha: :10/02/2013
Nombre:	Cambiar Diapositiva (Flip)	
Autor:	Usuario	
Descripción:	Describe el proceso de cambio de diapositiva (Transición).	
Flujo principal:	Eventos ACTOR	Eventos SISTEMA
	1.- Hace un Flip moviendo la mano derecha de frente al sensor de derecha a izquierda	1. Cambia diapositiva siguiente.
	2.- Hace un Flip moviendo la mano izquierda de frente al sensor de izquierda a derecha	2. Cambia diapositiva anterior.
Alternativa.	1.- El sensor no está conectado.	1. muestra en pantalla un mensaje de error.
Precondición	Que el usuario haya iniciado su presentación y que el sensor Kinect esté debidamente conectado.	
Pos condición	El usuario puede avanzar o retroceder su presentación con los movimientos de las manos derecha e izquierda respectivamente	

La documentación es muy esencial para saber qué es lo que se necesita y qué no, además sirve para disipar dudas o suposiciones en cuanto la construcción del Software. Para efectos de entendimiento de la documentación se ha plasmado un único caso de uso más sin embargo se han documentado todos los casos de uso.

Una vez que se tienen los requerimientos necesarios para la construcción del Software se procede al paso Plan y Diseño rápido.

Plan y Diseño rápido

En esta etapa se crea un diseño rápido, ver Figura 16, en el cual se representan todos aquellos aspectos básicos del software que serán visibles para el usuario, así como los componentes necesarios para el buen funcionamiento de la aplicación.

En esta fase del desarrollo del proyecto se implementan los requerimientos para realizar el diseño rápido del primer prototipo.

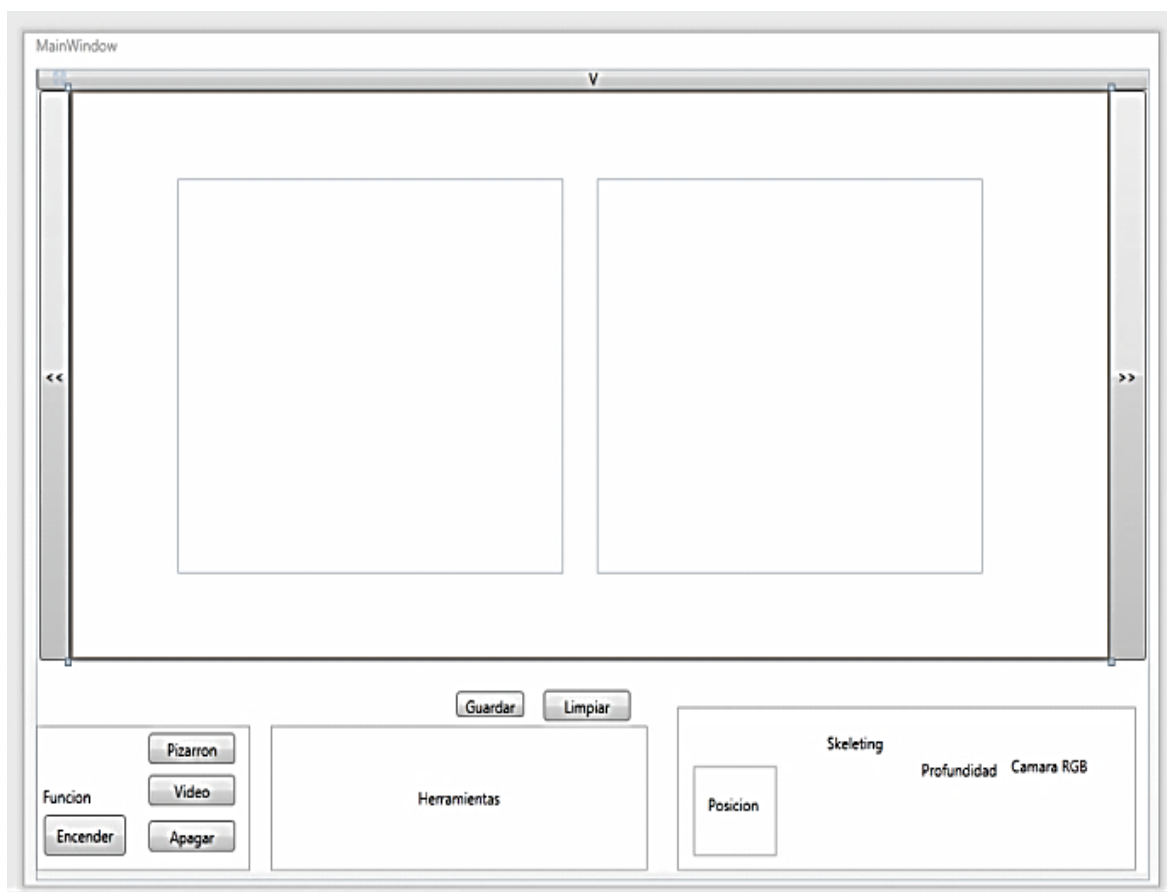


Figura 16: Diseño rápido

En la figura 16 se muestra el área de trabajo del usuario, además de algunos componentes de la aplicación que servirán para el uso de la misma, este diseño servirá para la construcción del prototipo número 1.

Construcción de prototipos

Prototipo numero 1

En esta etapa se construye el prototipo número 1, el cual se basa en el diseño rápido, cabe señalar que durante la construcción solo se estructuró la interfaz gráfica, por lo consiguiente el prototipo aún no utilizaba los recursos del sensor Kinect.

Prototipo número 2

En esta etapa se realiza la mejora del primer diseño propuesto, el cual ahora es el segundo prototipo del software, cabe recalcar que este último no debe alejarse del diseño rápido, es decir, conservar la idea principal que se tiene. Por lo anterior el segundo prototipo queda de la siguiente manera, ver Figura 17.

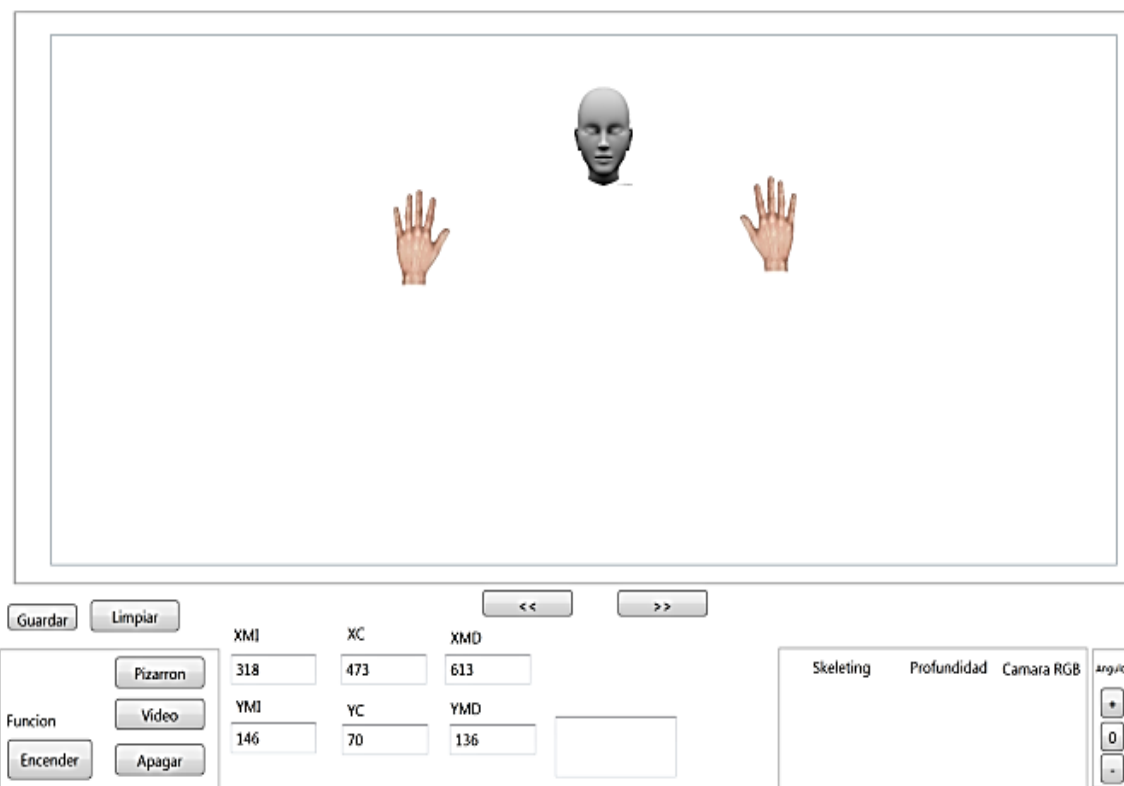


Figura 17: Diseño del primer prototipo y su funcionamiento

En el diseño anterior se observa que la interfaz fue modificada, en ella se implementaron los algoritmos necesarios para obtener los puntos de referencia de las manos del usuario así como la cabeza, logrando mover las figuras que representan las partes del cuerpo antes mencionadas, estas a su vez se mueven constantemente siguiendo los movimientos que el usuario realice. Los puntos son obtenidos a través de sensor Kinect y son interpretados con la ayuda del SDK.

Desarrollo y retroalimentación

Con los datos recabados anteriormente se crea un visualizador de imágenes dentro del área de trabajo, con la capacidad de avanzar o retroceder con los movimientos de las manos. Es decir que se toma de referencia la altura de la cabeza del usuario, de las manos se considera un deslice en el eje X ya sea a la derecha o izquierda para avanzar o retroceder respectivamente.

En seguida se muestra el diagrama de flujo muy general de la lógica aplicada para la creación del primer prototipo detección de gestos (Figura 18).

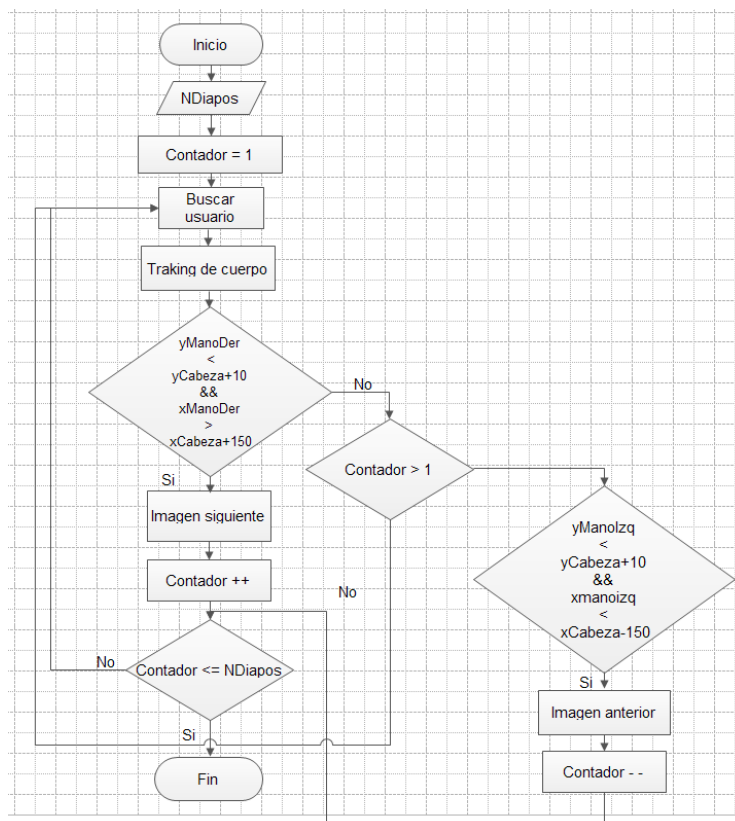


Figura 18: Diagrama de flujo para detectar gestos.

Evaluar y refinar requerimientos

Los requerimientos que se documentan al inicio del proyecto, cumplen con las expectativas y se acoplan debidamente al software que se está construyendo. Hasta el segundo prototipo cumple con 6 objetivos de 8:

1. Mostrar en la pantalla del sistema los movimientos de ambas manos del usuario, así como los de la cabeza.
2. Calibrar el dispositivo para que solo reconozca los movimientos de la persona que este exponiendo.
3. Avanzar la presentación con un movimiento específico de la mano derecha, mismo que será interpretado por el sensor.
4. Retroceder la presentación con un movimiento específico de a mano izquierda.
5. Simular la escritura, borrado y el zoom únicamente con los movimientos de las manos.
6. Agregar diapositivas en formato JPG.

Los objetivos faltantes se programan en la segunda fase, se pretende mejorar el reconocimiento de gestos.

El campo de visión del Sensor está restringido y el reconocimiento de personas se ha complicado, dado que el expositor puede ver o no al público, para ello se creó un método especial para que el sensor evite hacer funciones que no las realice el expositor.

El método es obtener las posiciones de profundidad de manos y cabeza, es decir se extraen los vectores Z, también se calculan los nuevos puntos X y Y de la cadera, hombros y muñecas, estos métodos se mencionan más adelante de este documento y son aplicados en el prototipo número 3.

El método de seguimiento de puntos funciona correctamente, se puede apreciar en la Figura 19 la interpretación que hace el programa, obteniendo los puntos de los ejes X y Y de la mano derecha, de la mano izquierda y la cabeza, estos puntos son importantes para el proyecto debido a que solo se toman estos tres para las posteriores funciones que se pretendan realizar.

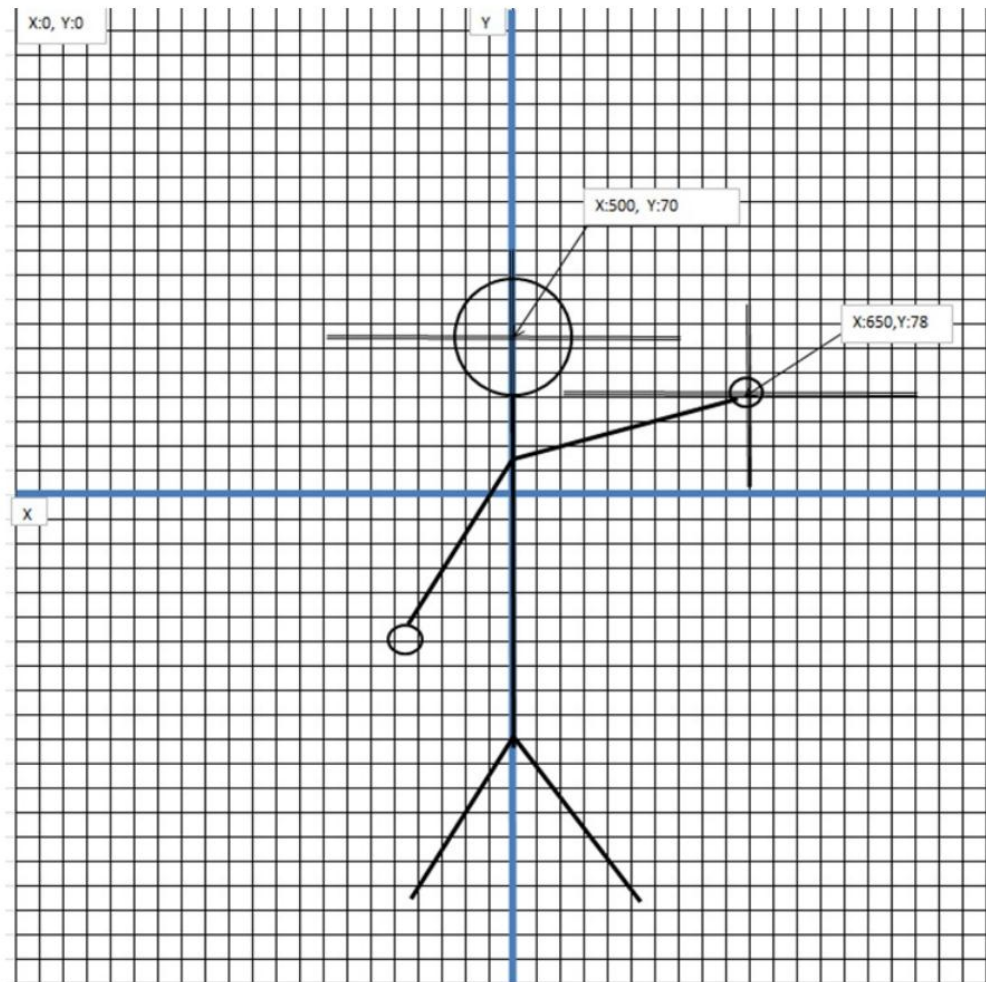


Figura 19: Coordenadas de puntos del expositor.

Con los puntos de la cabeza y manos, se pueden tomar los puntos X,Y y Z , de cada punto "Joint". Después que se haya redefinido el algoritmo de reconocimiento estos puntos servirán de referencia para que el algoritmo haga su trabajo y utilice los vectores.

En la figura 20 se puede observar como los “Joints” de cabeza y manos mueven los dibujos que están en la ventana de la presentación, estos dibujos son necesarios para que el usuario pueda saber dónde está posicionado.

En la misma venta se puede apreciar los valores de la cabeza y manos en el programa.

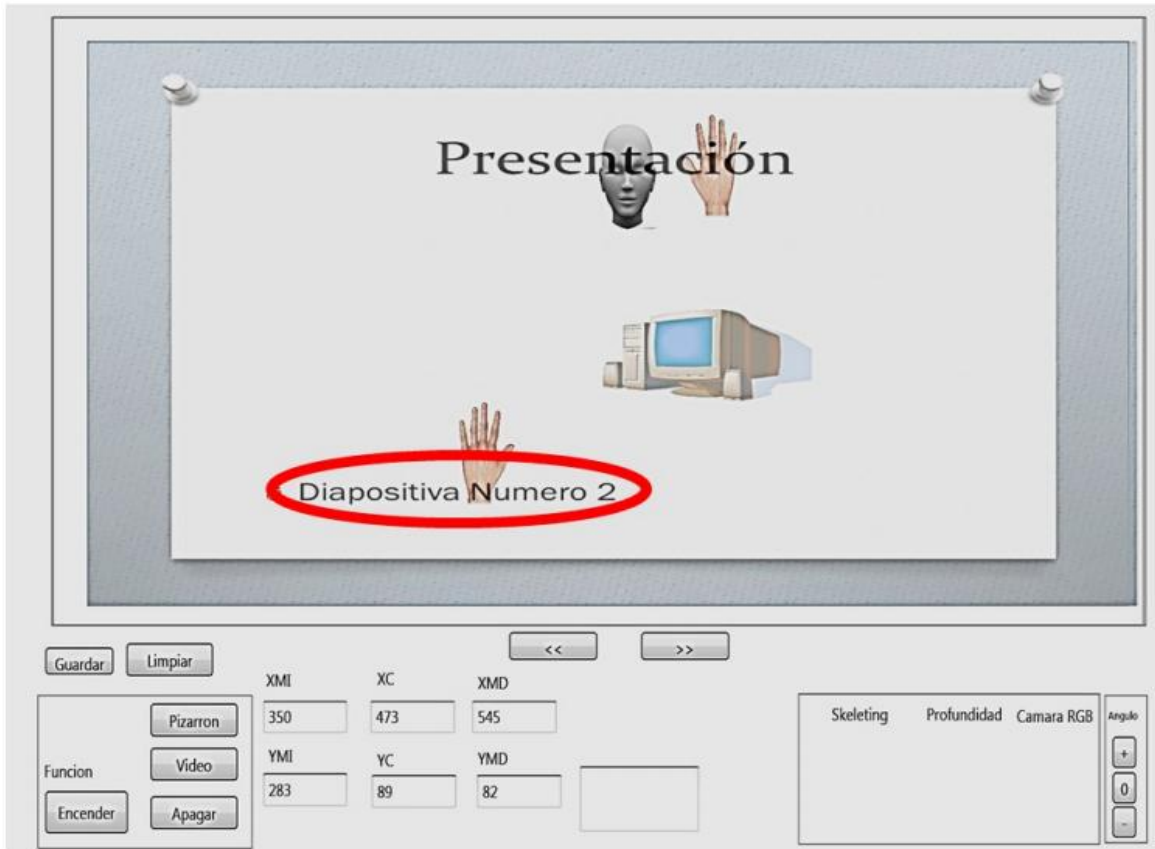


Figura 20: Programa en ejecución controlando una diapositiva.

La obtención de puntos (ver Figura 21), se realiza utilizando las funciones de tracking (seguimiento de un objeto en movimiento) y la creación de los puntos de la clase Skeleton con la que cuenta la librería del Kinect, esto permite extraer los puntos de cada articulación que unen las extremidades del cuerpo de una persona.



Figura 21. Esqueleton completo de una persona, obtenido con el Kinect

Esto es posible debido a la interpretación de los sensores de profundidad y a los filtrados que hace el SDK, como se puede apreciar en la figura 22.

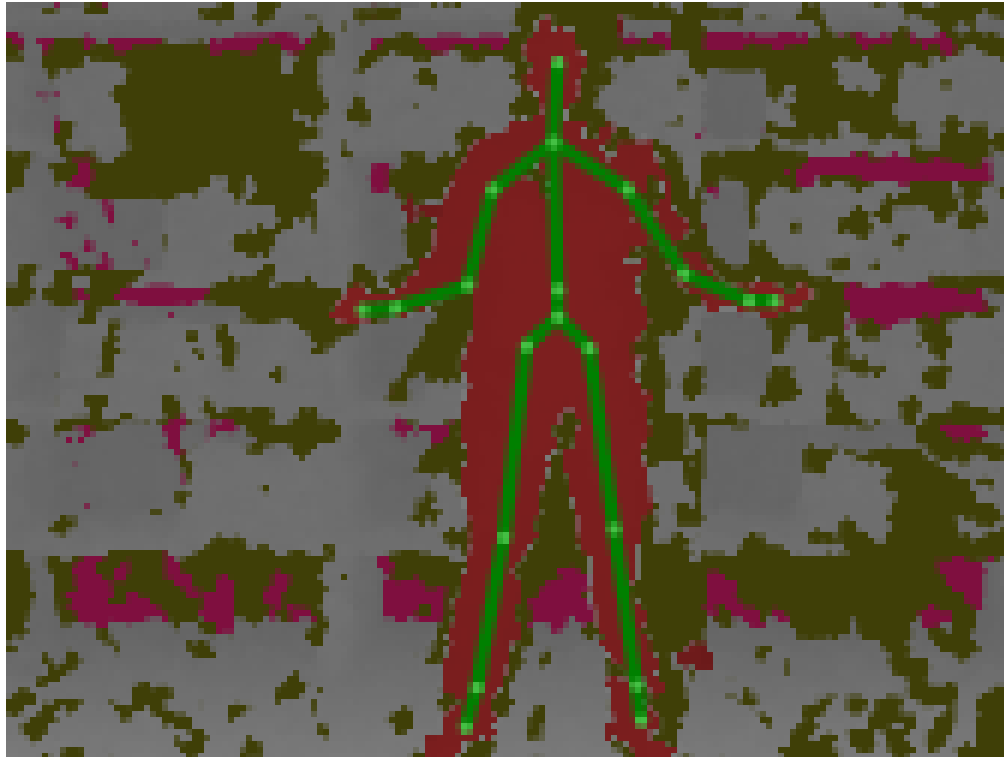


Figura 22: Imagen de profundidad con Skeleton

Los requerimientos no cambiarán, pero si se harán mejoras en la interfaz gráfica, debido a que éste fue un diseño rápido, para el final de la residencia se entregará el último prototipo que tendrá una interfaz más gráfica e intuitiva.

11. Resultados, Planos, Gráficas, Prototipos y Programas

Pantalla de inicio: su función es informar al usuario que el programa se está iniciando.

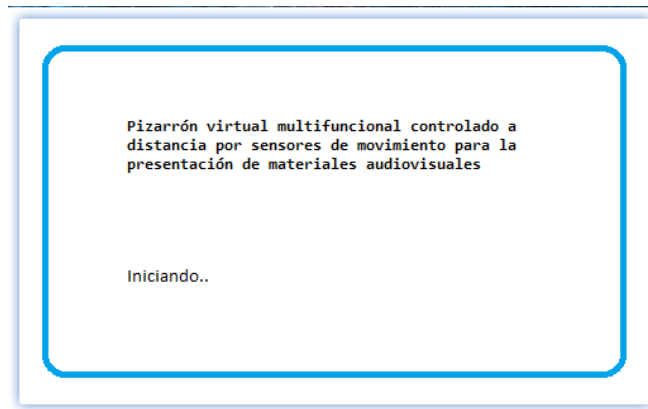


Figura 23: Pantalla de inicio

Pantalla de edición del proyecto: es la pantalla en donde el usuario puede abrir proyectos existentes o crear nuevos, además de poder ver la descripción completa del proyecto con que se encuentre trabajando.

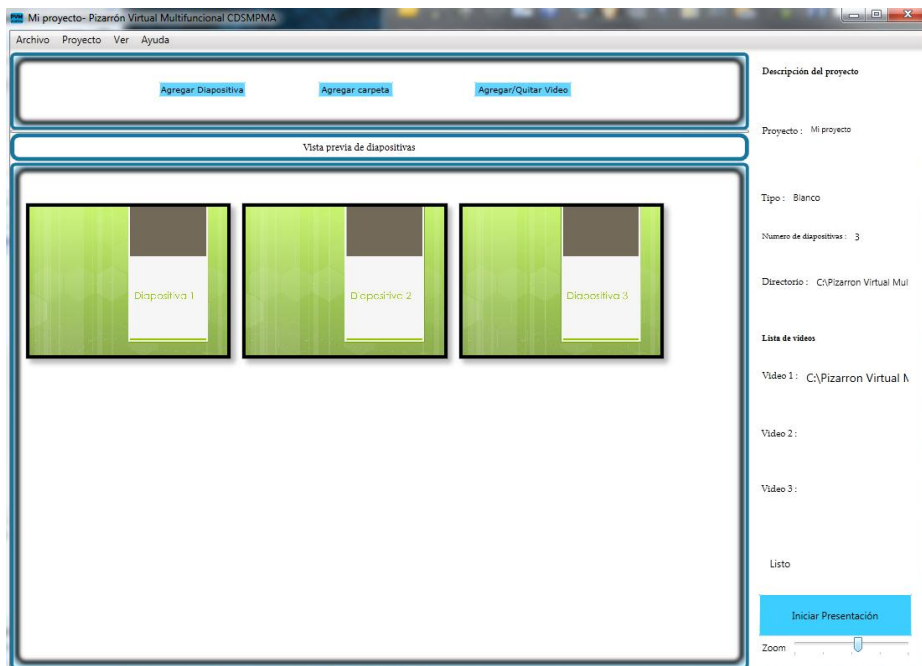


Figura 26: Pantalla de edición de proyecto

- 1.- El cuadro central es donde se visualizan las diapositivas.
- 2.- Menú.
- 3.- Botones para agregar una carpeta completa de diapositivas al proyecto o en forma individual, también se cuenta con el botón agregar video.
- 4.- Área donde se encuentra localizado el botón iniciar presentación y el zoom para cambiar el tamaño de la vista previa de las diapositivas.
- 5.- Área de información del proyecto como: nombre, tipo, número de archivos (diapositivas), etc.

Pantalla modo presentador: su función es primordial, en ella se visualizan las pantallas que se hayan cargado previamente en la pantalla de edición del proyecto.

El modo presentador contiene la programación donde se reconocen los gestos de una persona.

Contiene los botones básicos para controlar una presentación cuando no se haya detectado el sensor y botones para cerrar o iniciar el Kinect en caso que se conecte después de iniciar la presentación.

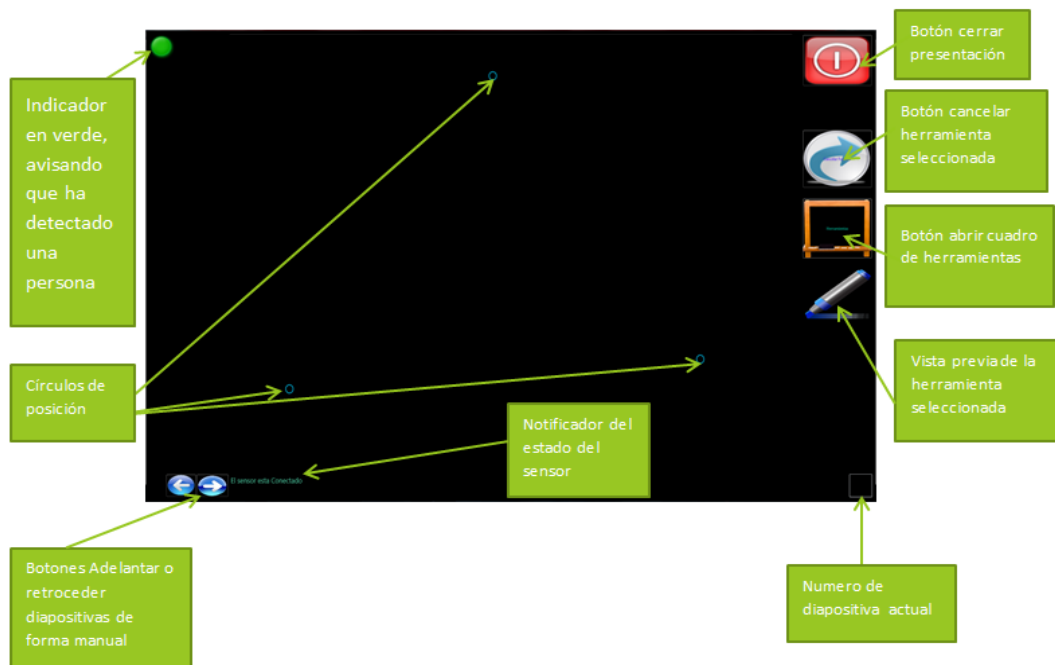


Figura 27: Pantalla en modo presentador



Figura 28: Visualizando una diapositiva

Ventana cuadro de herramientas: su función es brindarle al usuario las diferentes herramientas con la que cuenta el software, en total se cuenta con nueve herramientas como; pintar, Marca texto, Reproducir video, trazar cuadro, limpiar, Círculo, trazar línea, encerrar, Triángulo.

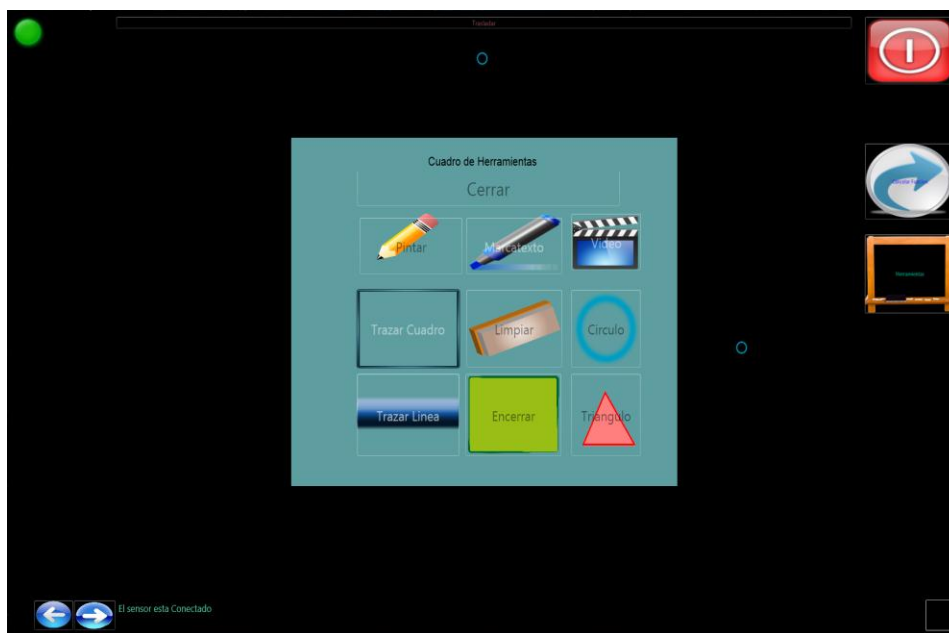


Figura 29: Ingresando al cuadro de herramientas

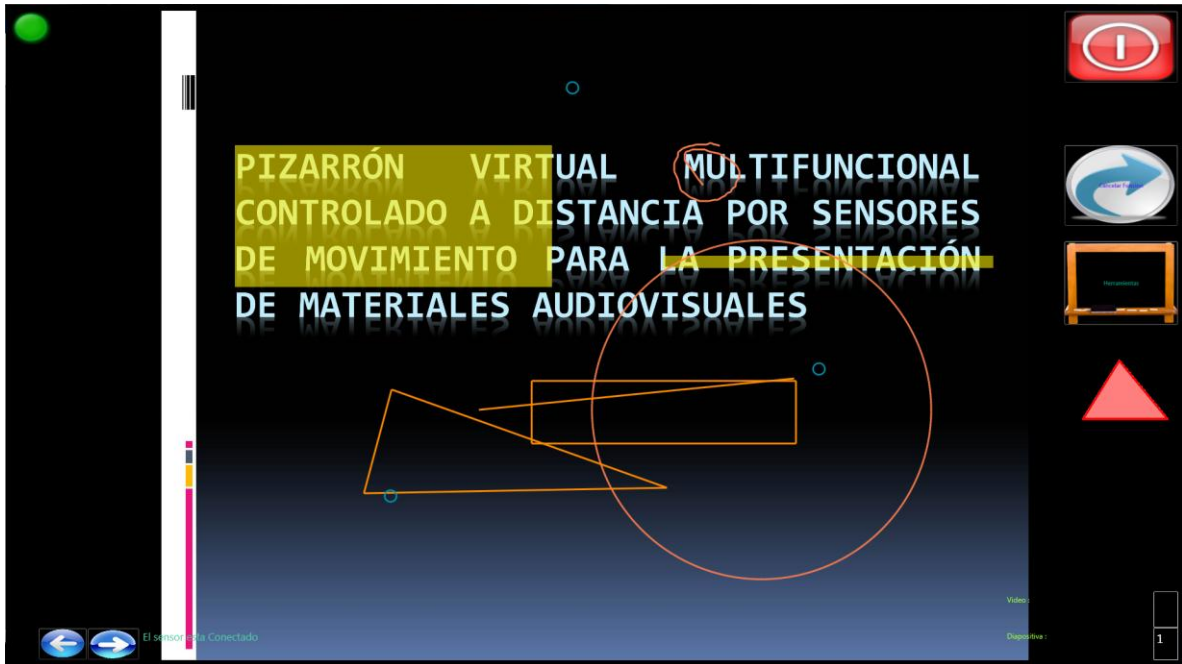


Figura 30: Utilizando la herramienta pintar



Figura 31: Usuario controlando el avance de una presentación



Figura 32: Usuario Realizando Zoom digital



Figura 33: Kinect interpretando los gestos del usuario

12. Conclusiones y Recomendaciones

Conclusión

Con las pruebas que se realizaron en el proceso de elaboración del software, se puede observar que se han cumplido los objetivos que se estipularon para la realización de la aplicación de software, cumpliendo de esta manera con el apoyo en el manejo de materiales visuales con funciones como:

- Avance de una diapositiva
- Retroceso de la diapositiva
- Zoom (ampliación y reducción de una imagen)
- Pintar y borrar

Cabe recalcar que las funciones antes mencionadas se pueden llevar acabo únicamente con movimientos de las manos, que para cada función fueron debidamente programadas y han sido utilizadas para comprobar las deficiencias encontradas y a la vez corregirlas para que el usuario final pueda utilizar el software sin problema alguno.

Algunas de las funciones fueron realizadas con ciertas limitantes, mas sin embargo esto no quiere decir que la aplicación no es eficiente, por el contrario actualmente se está trabajando en el mejoramiento de cada una de las funciones, haciendo del proyecto un tema de investigación para tesis.

Puesto que la aplicación es para el Instituto Tecnológico de Tuxtla Gutiérrez, se recomienda continuar mejorando esta herramienta que será de gran apoyo para los profesores, alumnos y directivos, actualmente no se cuenta dentro de la institución con una herramienta con las características que se han expuesto en este documento, además de motivar a sus alumnos en la creación e innovación de tecnologías que apoyen hoy en día a las actividades cotidianas de las personas.

Recomendación

Cabe destacar que durante el desarrollo de este proyecto, se observó que la codificación para el seguimiento del cuerpo del expositor, es completamente funcional con una persona que se encuentre en silla de ruedas, dado que el programa hace un tracking de las manos y cabeza.

El proyecto lo pueden utilizar personas que puedan estar de pie o estén discapacitadas, esto impacta puesto que no se pierden las características de pizarrón virtual multifuncional.

El proyecto se puede retomar enfocándolo a diferentes aplicaciones como: aplicar para el control de objetos en 3 dimensiones, control de voz, interpretación de gestos corporales (utilizando inteligencia artificial o visión por computadora). Así como para el desarrollo de aplicaciones en diferentes ramas como: la medicina, ingeniería o en donde se tenga la necesidad de controlar materiales audiovisuales de manera remota sin la necesidad de tocar un dispositivo.

Se recomienda afinar el desempeño de las funciones básicas y la utilización del software en el aula. Esto se realizará en el seguimiento del proyecto concluyendo con una tesis.

13. Bibliografía

[1] François Bérard, CLIPS-IMAG. *“The Magic Table: Computer-Vision Based Augmentation of a Whiteboard for Creative Meetings”*. University of Grenoble, Francia, Octubre de 2012.

[2] Wan-Chen Lee, Dora Adriana Merino-Schettino, Ignacio López-Martínez, Ruben Posada-Gómez, Ulises Juárez-Martínez, *“Virtual Board: A low cost Multi Touch Human Computer Interaction System”*. Division of Research and Postgraduate Studies, Instituto Tecnológico de Orizaba 94320, Veracruz, Mexico, 2012.

[3] Massimo Camplani, Luis Salgado and Romolo Camplani. *“Low-Cost Efficient Interactive Whiteboard”*. Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, Spain. Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy.

[4] <http://www.sistemastectuxtla.net/>

[5] Estefanía Fernández Sánchez. *“Control de Software Educativo Mediante Kinect de Microsoft”*. Universidad Carlos III de Madrid, 2012.

[6] Manol Zabalza Grau, Losu Azkue Odriozola. *“Tecnologías de interacción avanzada y realidad aumentada”*. Escuela técnica superior de Ingenieros Industriales y de telecomunicación. Pamplona, 17 de Junio de 2011.

[7] <http://digital.ni.com/public.nsf/allkb/46BE3912D75112C58625652C005B8422>

[8] Néstor A. Morales Navarro. *“Reconstrucción 3D de objetos con precisión subpixel”*. Instituto Tecnológico de Tuxtla Gutiérrez. 2010.

[9] <http://mirror2image.wordpress.com/2010/11/30/how-kinect-works-stereo-triangulation/>

[10] <http://microsoft.com/en-us/kinectforwindows/develop/new.aspx>

[11] <http://microsoft.com/en-us/kinectforwindows/develop/overview.aspx>

Anexos

Manual técnico.

1.- Plataforma tecnológica

La plataforma tecnológica del pizarrón virtual se presenta en la siguiente figura 34.



Figura 34: Plataforma tecnológica.

El ambiente consta de un sensor Kinect, una computadora de escritorio o portátil y el usuario que interactúa con el sistema.

2.- Características de Hardware y software utilizados para el desarrollo del proyecto.

Las características de hardware se dividen en dos, el de la computadora y el del sensor.

Hardware de la computadora

Referencia: Sony VAIO VGN-FW280FY

Procesador(es): Intel Centrino 2 de 2.2 GHz.

Memoria RAM: 4GB.

Tarjeta de video: ATI de 512 MB

Disco duro: 500 GB

Hardware del Sensor

El sensor Kinect dispone de cuatro partes fundamentales que podemos ver en la siguiente imagen y que se explica a continuación (ver figura 35).



Figura 35: Componentes del sensor Kinect

- 1. Sensores 3D de profundidad.** Los sensores tridimensionales hacen un seguimiento del cuerpo dentro del área del juego. Dispone de un proyector de profundidad (retícula izquierda) y un sensor de profundidad (retícula derecha), que calculan la distancia en función del tiempo que tarda en reflejar la luz.
- 2. Cámara RGB.** Una cámara RGB (Red, Green, Blue) que ayuda identificar y captar imágenes y videos con una resolución máxima de 640x480 a 30 fps.
- 3. Array de micrófonos.** Se usa un conjunto de cuatro micrófonos en el borde frontal inferior del sensor Kinect para el reconocimiento de voz.
- 4. Inclinación motorizada.** Se trata de un impulso mecánico en la base del sensor Kinect que inclina de manera automática el sensor hacia arriba o abajo según sea necesario hasta un máximo de 27°.

Además de estas partes que podemos observar en el propio sensor, Kinect dispone de lo siguiente:

- Memoria RAM de 512 MB.
- Acelerómetro, para estabilizar la imagen cuando se mueve.
- Ventilador, que no se encuentra encendido continuamente para no interferir con los micrófonos.

Software

- Sistema operativo: Windows 7 Ultimate 64 bits.
- Herramienta de desarrollo: Visual Studio 2010.
- Leguaje: C#(C Sharp).
- Tipo de proyecto: WPF (Windows Presentation Foundation).

3.- Funcionalidad del pizarrón.

3.1 Abrir proyecto

Descripción: Abrir proyecto.

Tipo de procedimiento: Procedimiento abrir nuevo proyecto de tipo .saca.

Datos de entrada: Dirección del proyecto.

Procedimiento:

1.- Evento al presionar el botón abrir proyecto.

```
private void MenuItem_Click_AbrirProyecto(object sender, RoutedEventArgs e)
{
    String texto = null;
    System.Windows.Forms.OpenFileDialog file = new OpenFileDialog();
    file.Filter = "Proyecto CEG (*.ceg) | *.CEG";
```

2.- Se abre un cuadro de dialogo para seleccionar el proyecto con extencion .saca.

```
        if(file.ShowDialog() == System.Windows.Forms.DialogResult.OK)
        {
```

3.- Procedimiento para abrir el archivo y extraer el contenido.

```
            btn_agregar.IsEnabled = Convert.ToBoolean(true);

            btn_AgregarCarp.IsEnabled = Convert.ToBoolean(true);
            btn_Video.IsEnabled = Convert.ToBoolean(true);
            Nfile = file.FileName.ToString();
```

```

System.IO.StreamReader fi = new System.IO.StreamReader(Nfile);
//System.Windows.Forms.MessageBox.Show(Nfile);
texto = fi.ReadToEnd();
fi.Close();
String cad1 = texto;
String[] words = cad1.Split('|');
try
{
    NombreProyecto = words[0];
    TipoDeProyecto = words[1];
    NoArchivos = words[2];
    TipoAnimacion = words[3];
    DirMultimedia = words[4];
    NumeroArchivos = Convert.ToInt32(words[5]);
}
catch (Exception )
{
    System.Windows.Forms.MessageBox.Show("Archivo dañado");
}

```

4.- Actualización de la dirección multimedia.

```

        Fotos.Path = DirMultimedia;
    }

```

5.- Actualización de la descripción en la ventana.

```

NombreProy.Content = NombreProyecto;
TipoProy.Content = TipoDeProyecto;
Numero.Content = NumeroArchivos;
Animación.Content = TipoAnimacion;
Directorio.Content = DirMultimedia;
Title = "Archivo -" + NombreProyecto;
}

```

3.2 Agregar diapositiva

Descripción: Agrega una nueva diapositiva al proyecto

Tipo de procedimiento: Insertar diapositiva y actualizar el archivo de proyecto.

Datos de entrada: Dirección fuente de la diapositiva.

Procedimiento:

1.- Se produce el evento al presionar el botón agregar

```

private void btn_agregar_Click(object sender, RoutedEventArgs e)
{
    String NomOrigi = null;

    System.Windows.Forms.OpenFileDialog file = new OpenFileDialog();
    file.Filter = "Archivo de imagen (*.jpg) | *.JPG";
    if (file.ShowDialog() == System.Windows.Forms.DialogResult.OK)
    {
        String dir = file.FileName.ToString();
        totlArch();
        NomOrigi = obtNom(dir);
    }
}

```

2.- Se manda la dirección y nombre de la diapositiva de tipo .jpg a la función agregarArchivo

```
agregarArchivo(dir, NomOrigi);
```

3.- Después de completarse la tarea se actualizar el visualizador

```
    actualizarVisualizador();  
  }  
}
```

Función agregarArchivo.

```
public void agregarArchivo(String directorio, String NombreOriginal) //Agrega el nuevo archivo a la carpeta de trabajo  
{  
    string fileName, destFile = null;  
    TotalArchivos = TotalArchivos + 1;  
    NumeroArchivos++;  
    //Copiar  
    fileName = System.IO.Path.GetFileName(directorio);  
    destFile = System.IO.Path.Combine(DirMultimedia, fileName);  
    System.IO.File.Copy(directorio, destFile, true);  
    //System.Windows.Forms.MessageBox.Show(destFile);  
    //Renombrar  
    System.IO.File.Move(destFile, DirMultimedia + "\\Images_CEG " + NumeroArchivos + ".jpg");  
    //Incrementar variables de control  
    ModifArchivo();  
}
```

Función actualizarVisualizador

```
public void actualizarVisualizador()  
{  
    Photos.Path = DirMultimedia;  
}
```

3.3 Iniciar presentación

Descripción: Se inicia la presentación y se inicia el sensor Kinect

Tipo de procedimiento: iniciar presentación e iniciar sensor Kinect

Datos de entrada:

Procedimiento:

1.- Evento al presionar el botón iniciar presentacion

```
private void Iniciar_Click(object sender, RoutedEventArgs e)  
{  
    IniPresn();  
}
```

2.- Se crea la nueva instancia.

```
private void IniPresn()
{
    Thread.Sleep(1000);
    string[] filePaths = Directory.GetFiles(DirMultimedia, "*.jpg");
    NoArchivos = Convert.ToString(filePaths.Length);
    ModoPresentador MP = new ModoPresentador();
    MP.setDir(DirMultimedia, NoArchivos); MP.ShowDialog(); }
}
```

3.- Iniciando el sensor kinect

```
/* 1 */
#region Iniciar_Kinect
```

4.- Iniciamos las funciones para utilizar el sensor Kinect

```
public void Iniciar_Kinect()
{
    try
    {
        Estado = 1;
        KinectSensor.KinectSensors.StatusChanged += KinectSensors_StatusChanged; //checa el estado del
sensor Kinect
        this.KinectDevice = KinectSensor.KinectSensors.FirstOrDefault(x => x.Status ==
KinectStatus.Connected); //Tomamos es reporte del estado del sensor
    }
    catch (Exception)
    {
        Estado = 0;
        System.Windows.Forms.MessageBox.Show("Error, verifique el sensor");
    }
}
#endregion Iniciar_Kinect
```

5.- Se evalúa el estado del sensor.

```
/* 2 */
#region Evaluar_Estado_Kinect
private void KinectSensors_StatusChanged(object sender, StatusChangedEventArgs e)
{
    switch (e.Status)
    {
        case KinectStatus.Initializing:
            textBoxMensajes.Text = "Iniciando sensor";
            break;
        case KinectStatus.Connected:
            textBoxMensajes.Text = "El sensor esta Conectado";
            Iniciar_Kinect(); //Iniciamos el sensor
            break;
        case KinectStatus.NotPowered:
            textBoxMensajes.Text = "El sensor no esta conectado con el Adaptador de corriente";
            break;
        case KinectStatus.NotReady:
            textBoxMensajes.Text = "El sensor no se ha iniciado";
            break;
        case KinectStatus.DeviceNotGenuine:
            textBoxMensajes.Text = "El sensor no es Genuino";
            this.KinectDevice = e.Sensor;
            break;
        case KinectStatus.Disconnected:
            textBoxMensajes.Text = "El sensor esta desconectado";
            botonRojo.Visibility = Visibility.Collapsed;
            this.KinectDevice = null; // En caso de que el sensor no esté conectado devolvemos null, Este
valor será evaluado en las propiedades del kinect
            break;
        default:
            textBoxMensajes.Text = "***error estado desconocido***";
            break;
    }
}
#endregion Evaluar_Estado_Kinect
```


6.- después de la evaluación se procede a iniciar el sensor con el tracking de las articulaciones

```

/* 3 */
#region PropertiesKinect
private KinectSensor KinectDevice
{
    get { return this._KinectDevice; }
    set
    {
        if (this._KinectDevice != value)
        {
            //Uninitialize
            if (this._KinectDevice != null) //Detenemos el sensor
            {
                this._KinectDevice.Stop();
                this._KinectDevice.SkeletonFrameReady -= KinectDevice_SkeletonFrameReady;
                this._KinectDevice.SkeletonStream.Disable();
                this._FrameSkeletons = null;
            }

            this._KinectDevice = value;

            //Initialize
            if (this._KinectDevice != null) //Iniciamos el sensor
            {
                if (this._KinectDevice.Status == KinectStatus.Connected)
                {
                    this._KinectDevice.SkeletonStream.Enable(); //Habilitamos el Skeleton
                    this._FrameSkeletons = new
                    Skeleton[this._KinectDevice.SkeletonStream.FrameSkeletonArrayLength];
                    this.KinectDevice.SkeletonFrameReady += KinectDevice_SkeletonFrameReady; //Tomamos el
                    Skeleton

                    this._KinectDevice.Start(); //Iniciamos el sensor
                    KinectDevice.ElevationAngle = 0;
                    textBoxMensajes.Text = "El sensor esta Conectado";
                }
            }
        }
    }
}
#endregion PropertiesKinect

```

7.- Se Inicia el Tracking de las articulaciones

```

/* 4 */
#region Kinect_Skeleton_FrameReady

private void KinectDevice_SkeletonFrameReady(object sender, SkeletonFrameReadyEventArgs e)//Cuando se
detecta una persona se obtiene el Skeleton
{
    using (SkeletonFrame frame = e.OpenSkeletonFrame())
    {
        if (frame != null)
        {
            try
            {
                frame.CopySkeletonDataTo(this._FrameSkeletons);
            }
            catch (InvalidCastException)
            {
            }
        }

        Skeleton skeleton = GetPrimarySkeleton(this._FrameSkeletons);// Tomamos el skeleton del usuario

        if (skeleton == null)// No se encontro ningun usuario
        {
            skeletonUsuario = 0;
            ocultarCirculos();
            botonRojo.Visibility = Visibility.Visible;
            botonVerde.Visibility = Visibility.Collapsed;
        }
    }
}

```

```

else if (skeleton != null && skeletonUsuario == 1) //Tomamos el esqueleto de un solo usuario
{
    /* Obtenemos los Joints que se utilizaran en el programa*/
    Joint HandLeft = GetHandLeft(skeleton);
    Joint HandRight = GetHandRight(skeleton);
    Joint Head = GetHead(skeleton);
    Joint homIzq = GetHombroIzq(skeleton);
    Joint homDer = GetHombroDer(skeleton);
    Joint homCen = GetHombroCen(skeleton);
    Joint CentroCadera = GetCadCen(skeleton);
    /* Obtenemos los Vectores de profundidad "Z" de los Joints que utilizaran en el programa*/
    float zManoDerecha = skeleton.Joints[JointType.HandRight].Position.Z;
    float zManoIzquierda = skeleton.Joints[JointType.HandLeft].Position.Z;
    float zCabeza = skeleton.Joints[JointType.Head].Position.Z;

    if (zCabeza > 1 && zCabeza < 2.4) //Evalua que el usuario no esta alejado a 2.5 metros de
    distancia y que no se encuentre a menos de un metro de distancia del sensor
    {
        botonVerde.Visibility = Visibility.Visible;
        botonRojo.Visibility = Visibility.Collapsed;
        IniTracking(Head, HandLeft, HandRight, homIzq, homDer, homCen, CentroCadera,
zManoDerecha, zManoIzquierda, zCabeza);
    }
    else
    {
        botonRojo.Visibility = Visibility.Visible;
        botonVerde.Visibility = Visibility.Collapsed;
        reiniciarAngulo();
        ocultarCirculos();
    }
}
}
}
}

#endregion Kinect_Skeleton_FrameReady

```

8.- Se toman los nodos a utilizar

```

#region Tomar_Nodos_Sekeleton
//Mano Derecha
private static Joint GetHandRight(Skeleton skeleton)
{
    Joint HandRight = new Joint();

    if (skeleton != null)
    {
        HandRight = skeleton.Joints[JointType.HandRight];
    }
    return HandRight;
}

//Mano izquierda
private static Joint GetHandLeft(Skeleton skeleton)
{
    Joint HandLeft = new Joint();

    if (skeleton != null)
    {
        HandLeft = skeleton.Joints[JointType.HandLeft];
    }
    return HandLeft;
}

//Cabeza
private static Joint GetHead(Skeleton skeleton)
{
    Joint Head = new Joint();
    if (skeleton != null)
    {
        Head = skeleton.Joints[JointType.Head];
    }
    return Head; }
}

```

```

private static Joint GetHombroIzq(Skeleton skeleton)
{
    Joint hombroIzq = new Joint();
    if (skeleton != null)
    {
        hombroIzq = skeleton.Joints[JointType.ShoulderLeft];
    }
    return hombroIzq;
}

private static Joint GetHombroDer(Skeleton skeleton)
{
    Joint hombroDer = new Joint();
    if (skeleton != null)
    {
        hombroDer = skeleton.Joints[JointType.ShoulderRight];
    }
    return hombroDer;
}

private static Joint GetHombroCen(Skeleton skeleton)
{
    Joint hombroCen = new Joint();
    if (skeleton != null)
    {
        hombroCen = skeleton.Joints[JointType.ShoulderRight];
    }
    return hombroCen;
}

private static Joint GetCadCen(Skeleton skeleton)
{
    Joint CentroCadera = new Joint();
    if (skeleton != null)
    {
        CentroCadera = skeleton.Joints[JointType.ShoulderRight];
    }
    return CentroCadera;
}
#endregion Tomar_Nodos_Sekeleton

```

9.- Captura del Skeleton del usuario

```

#region Kinect_Get_Skeleton

private Skeleton GetPrimarySkeleton(Skeleton[] skeletons)
{
    Skeleton skeleton = null;
    skeletonUsuario = 1;
    if (skeletons != null)
    {
        for (int i = 0; i < skeletons.Length; i++)
        {
            if (skeletons[i].TrackingState == SkeletonTrackingState.Tracked)
            {
                if (skeleton == null)
                {
                    skeleton = skeletons[i];
                }
            }
        }
    }
    return skeleton;
}
#endregion Kinect_Get_Skeleton

```

Cuando el usuario este interactuado con el kinect y haga los gestos avanzar, retroceder etc. Se evalúan el recorrido de los puntos.

```
#region Tracking_y_EvaluaciondeGestos
//Se inicia el tracking completo de cada Joint en los vectores X,Y y Z
private void IniTracking(Joint head, Joint handLeft, Joint handRight, Joint ShoulderLeft, Joint
ShoulderRight, Joint ShoulderCenter, Joint CadCen, float zManoDerecha, float zManoIzquierda, float zCabeza)
{
    int xmIzq, ymIzq, xmDer, ymDer, xh, yh, xhomIzq, yhomIzq, xhomDer, yhomDer, xhomCen, yhomCen, xcadCen,
ycadCen;

    HeadCursorElement.Visibility = System.Windows.Visibility.Visible;
    DepthImagePoint pointH =
this.KinectDevice.CoordinateMapper.MapSkeletonPointToDepthPoint(head.Position,
DepthImageFormat.Resolution640x480Fps30);
    pointH.X = (int)((pointH.X * LayoutRoot.ActualWidth / this.KinectDevice.DepthStream.FrameWidth) -
(HeadCursorElement.ActualWidth / 2.0));
    pointH.Y = (int)((pointH.Y * LayoutRoot.ActualHeight / this.KinectDevice.DepthStream.FrameHeight) -
(HeadCursorElement.ActualHeight / 2.0));
    xh = pointH.X;
    yh = pointH.Y;

    HandIzqCursorElement.Visibility = System.Windows.Visibility.Visible;
    DepthImagePoint pointManIzq =
this.KinectDevice.CoordinateMapper.MapSkeletonPointToDepthPoint(handLeft.Position,
DepthImageFormat.Resolution640x480Fps30);
    pointManIzq.X = (int)((pointManIzq.X * LayoutRoot.ActualWidth /
this.KinectDevice.DepthStream.FrameWidth) - (HandIzqCursorElement.ActualWidth / 2.0));
    pointManIzq.Y = (int)((pointManIzq.Y * LayoutRoot.ActualHeight /
this.KinectDevice.DepthStream.FrameHeight) - (HandIzqCursorElement.ActualHeight / 2.0));
    xmIzq = pointManIzq.X;
    ymIzq = pointManIzq.Y;

    HandDerCursorElement.Visibility = System.Windows.Visibility.Visible;
    DepthImagePoint pointManDer =
this.KinectDevice.CoordinateMapper.MapSkeletonPointToDepthPoint(handRight.Position,
DepthImageFormat.Resolution640x480Fps30);
    pointManDer.X = (int)((pointManDer.X * LayoutRoot.ActualWidth /
this.KinectDevice.DepthStream.FrameWidth) - (HandDerCursorElement.ActualWidth / 2.0));
    pointManDer.Y = (int)((pointManDer.Y * LayoutRoot.ActualHeight /
this.KinectDevice.DepthStream.FrameHeight) - (HandDerCursorElement.ActualHeight / 2.0));
    xmDer = pointManDer.X;
    ymDer = pointManDer.Y;

    DepthImagePoint pointHomI =
this.KinectDevice.CoordinateMapper.MapSkeletonPointToDepthPoint(ShoulderLeft.Position,
DepthImageFormat.Resolution640x480Fps30);
    pointHomI.X = (int)((pointHomI.X * LayoutRoot.ActualWidth /
this.KinectDevice.DepthStream.FrameWidth));
    pointHomI.Y = (int)((pointHomI.Y * LayoutRoot.ActualHeight /
this.KinectDevice.DepthStream.FrameHeight));
    xhomIzq = pointHomI.X;
    yhomIzq = pointHomI.Y;

    DepthImagePoint pointHomD =
this.KinectDevice.CoordinateMapper.MapSkeletonPointToDepthPoint(ShoulderRight.Position,
DepthImageFormat.Resolution640x480Fps30);
    pointHomD.X = (int)((pointHomD.X * LayoutRoot.ActualWidth /
this.KinectDevice.DepthStream.FrameWidth));
    pointHomD.Y = (int)((pointHomD.Y * LayoutRoot.ActualHeight /
this.KinectDevice.DepthStream.FrameHeight));
    xhomDer = pointHomD.X;
    yhomDer = pointHomD.Y;

    DepthImagePoint pointHomC =
this.KinectDevice.CoordinateMapper.MapSkeletonPointToDepthPoint(ShoulderCenter.Position,
DepthImageFormat.Resolution640x480Fps30);
    pointHomC.X = (int)((pointHomC.X * LayoutRoot.ActualWidth /
this.KinectDevice.DepthStream.FrameWidth));
    pointHomC.Y = (int)((pointHomC.Y * LayoutRoot.ActualHeight /
this.KinectDevice.DepthStream.FrameHeight));
    xhomCen = pointHomC.X;
    yhomCen = pointHomC.Y;
}

```

```

        DepthImagePoint pointCadCen =
this.KinectDevice.CoordinateMapper.MapSkeletonPointToDepthPoint(CadCen.Position,
DepthImageFormat.Resolution640x480Fps30);
        pointCadCen.X = (int)((pointCadCen.X * LayoutRoot.ActualWidth /
this.KinectDevice.DepthStream.FrameWidth));
        pointCadCen.Y = (int)((pointCadCen.Y * LayoutRoot.ActualHeight /
this.KinectDevice.DepthStream.FrameHeight));
        xcadCen = pointCadCen.X;
        ycadCen = pointCadCen.Y;

        MoverCirculos(xh, yh, xmIzq, ymIzq, xmDer, ymDer);
        //Se aumentaron 20 puntos en x de mano derecha y 10 en y para centrar con e cursor
        EvaluarGesto(xmIzq + 4, ymIzq + 4, xmDer + 20, ymDer + 10, xh, yh, xhomIzq, yhomIzq, xhomDer, yhomDer,
xhomCen, yhomCen, xcadCen, ycadCen, zManoDerecha, zManoIzquierda, zCabeza);
    }
}

```

La función que se encarga de evaluar cada gesto que el usuario haga se llama “EvaluarGesto”, y recibe como parámetros los nodos de manos, cabeza, hombros, cadera y cadera central, en sus vectores X, Y y Z.

```

private void EvaluarGesto(int xmIzq, int ymIzq, int xmDer, int ymDer, int xh, int yh, int xhomIzq, int yhomIzq,
int xhomDer, int yhomDer, int xhomCen, int yhomCen, int xcadcen, int ycadCen, float zManderecha, float
zManizquierda, float z_Cabeza)
{
    if (Estado == 1) //Aplicacion no finalizada
    {
        MoverMotor(yh); //Movemos el motor en el vector y, se toma como referencia la altura de la cabeza
respecto al margen superior del programa

        float centro, punto;
        centro = (yhomCen - ycadCen) / 2;
        punto = ycadCen + centro;

        EventoClickEnBotonManoDerecha(xmDer, ymDer, zManderecha, zManizquierda, z_Cabeza); //Enviamos
lecturas de puntos al cursor

        if (zManderecha + .4 <= z_Cabeza) // Gesto pintar con mano derecha, si el BloquearClick de pintar
esta desactivado
        {
            pintarConManoDerecha(xmDer, ymDer);
        }
        else
        {
            band = 0;
        }

        if (zManizquierda + .4 <= z_Cabeza) // Gesto pintar con mano izquierda, si el BloquearClick de
pintar esta desactivado
        {
            pintarConManoIzquierda(xmIzq, ymIzq);
        }

        /*Modo presentador*/

        if (ymDer > yh && xmDer < xh + 150 && zManderecha + .5 <= z_Cabeza && BloquearGesto == false)
//Gesto avanzar diapositiva
        {
            Thread.Sleep(500);
            clickAudio();
            ScaleTransform scale = new ScaleTransform(1, 1);
            Image_1.RenderTransform = scale;
            try
            {
                Avanzar();
            }
            catch
            {
                System.Windows.MessageBox.Show("No se han cargado diapositivas en el proyecto");
            }
        }
    }
}

```

```

        if (ymIzq > yh && xmIzq > xh - 150 && zManizquierda + .5 <= z_Cabeza && BloquearGesto ==
false)//Gesto retroceder diapositiva
    {
        Thread.Sleep(500);
        ScaleTransform scale = new ScaleTransform(1, 1);
        Image_1.RenderTransform = scale;
        try
        {
            Retroceder();
        }
        catch
        {
            System.Windows.MessageBox.Show("No se han cargado diapositivas en el proyecto");
        }
    }

    /*Modo presentador video*/
    if (ymDer > yh && xmDer < xh + 150 && zManderecha + .5 <= z_Cabeza && BloquearGestovideo == false)
    {
        Thread.Sleep(500);
        if (contadorv >= 3)
        {
            this.inkCanvas1.Strokes.Clear();
            inkCanvas1.Children.Clear();

            contadorv = 0;
            reproducirVideoContenedor(++contadorv);
            textBox2.Text = contadorv.ToString();
        }
        else
        {
            this.inkCanvas1.Strokes.Clear();
            inkCanvas1.Children.Clear();

            reproducirVideoContenedor(++contadorv);
            textBox2.Text = contadorv.ToString();
        }
    }

false) if (ymIzq > yh && xmIzq > xh - 150 && zManizquierda + .5 <= z_Cabeza && BloquearGestovideo ==
    {
        Thread.Sleep(500);
        if (contadorv <= 1)
        {
            this.inkCanvas1.Strokes.Clear();
            inkCanvas1.Children.Clear();

            contadorv = 1;
            reproducirVideoContenedor(contadorv);
            textBox2.Text = contadorv.ToString();
        }
        else
        {
            this.inkCanvas1.Strokes.Clear();
            inkCanvas1.Children.Clear();

            reproducirVideoContenedor(--contadorv);
            textBox2.Text = contadorv.ToString();
        }
    }

    if (ymIzq-150 < punto && ymDer-150 < punto) //Gesto Zoom
    {
        float distance = Math.Abs(xmDer - xmIzq);
        image_Zoom(distance, xh, yh);
    }

    if (ymDer == ymIzq && xmDer == xmIzq)//Gesto Salir de Modo presentador
    {
        try
        {
            Estado = 0; // Aplicacion finalizada
            reiniciarAngulo();
            this._KinectDevice.Stop();
            this.Close();
        }
        catch (Exception)
        { this.Close(); } } }

```

Manual de usuario

Pizarrón virtual multifuncional controlado a distancia por sensores de movimiento para la presentación de materiales audiovisuales

Versión 1.0.

1.-Instalación

Instalación de Pizarrón virtual multifuncional CDSMPMA (controlado a distancia por sensores de movimiento para la presentación de materiales audiovisuales)

Descripción:

Herramienta de apoyo para la presentación de materiales audiovisuales con distintas herramientas para interacción con el material de exposición controlado con el sensor Kinect de Microsoft.

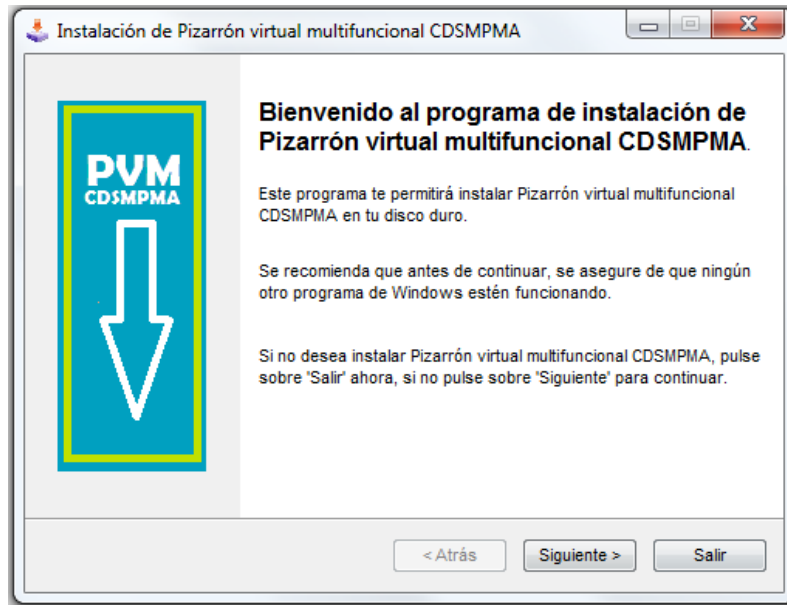
El software incluye un instalador que se puede ejecutar en un sistema operativo Windows 7 64 o 32 Bits.

Dentro del instalador contiene los drivers del sensor Kinect y el Framework que son necesarios para el buen funcionamiento del software. El driver del sensor y su paquetería se instalan al momento de dar continuidad el instalador que se ejecuta la primera vez que se inicia el programa Pizarrón Virtual Multifuncional CDSMPMA.

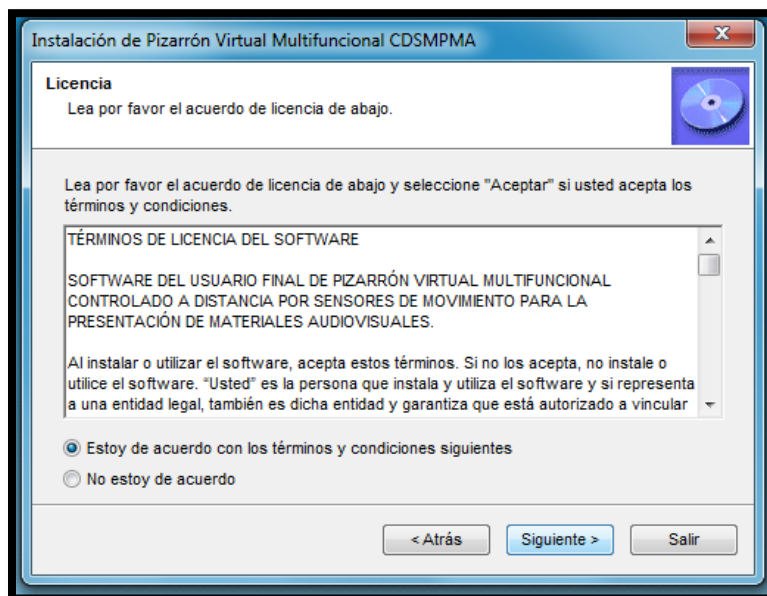
El software incluye en su instalación el programa principal donde se tiene la vista previa del proyecto, en él se muestran la información y estado del proyecto, también cuenta con una barra de menú que contiene diferentes opciones para crear, cerrar y modificar el proyecto.

Para instalar el producto ejecute Pizarrón Virtual CDSMPMA.exe y siga los pasos durante la instalación.

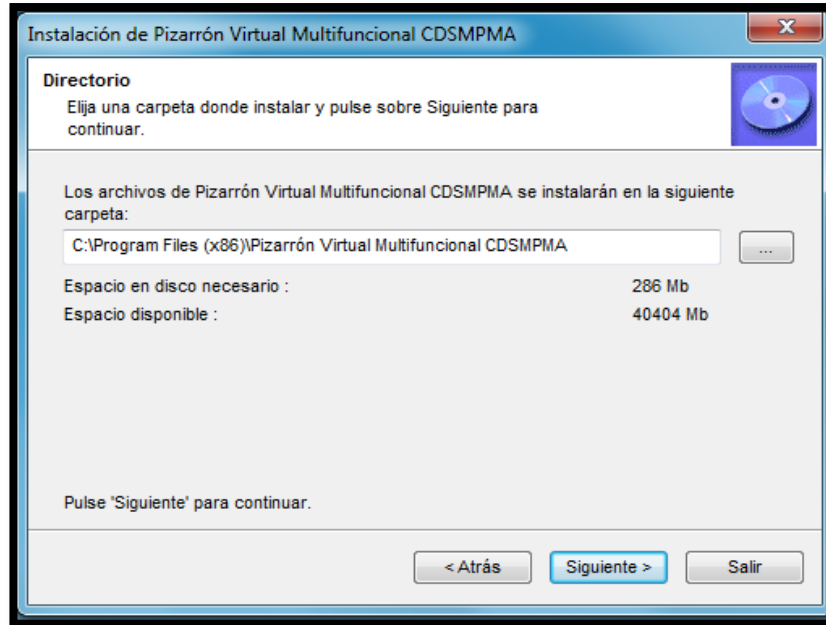
1.- Haga Clic en siguiente.



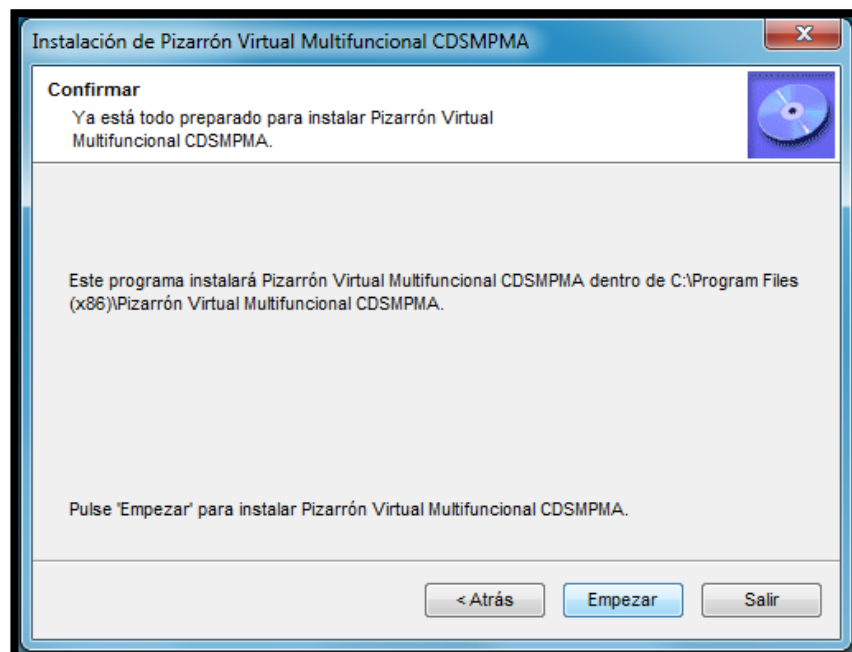
2.- Lea los términos y condiciones del programa



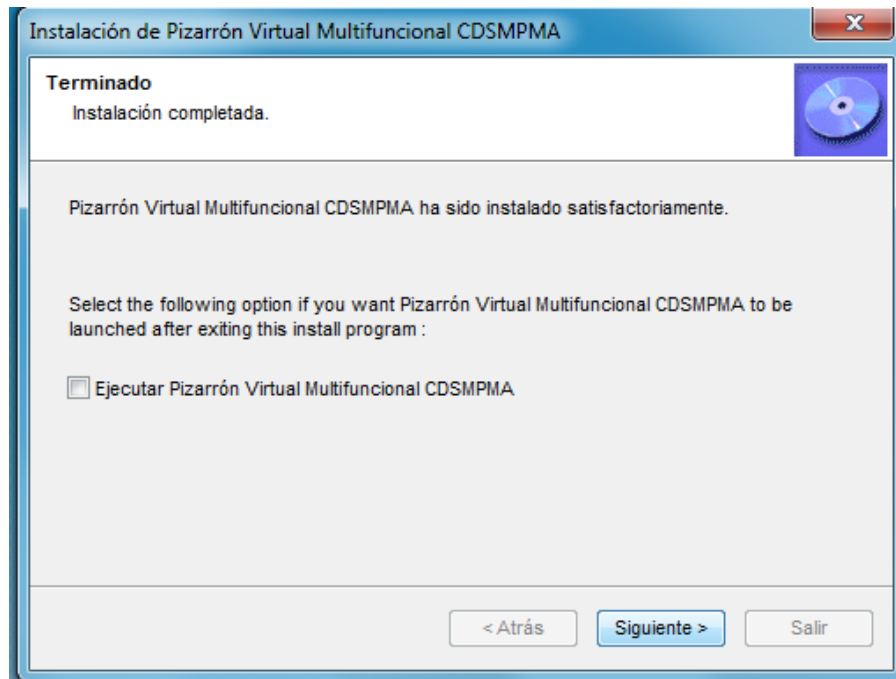
3.- Seleccione la carpeta donde instalara el programa y sus componentes



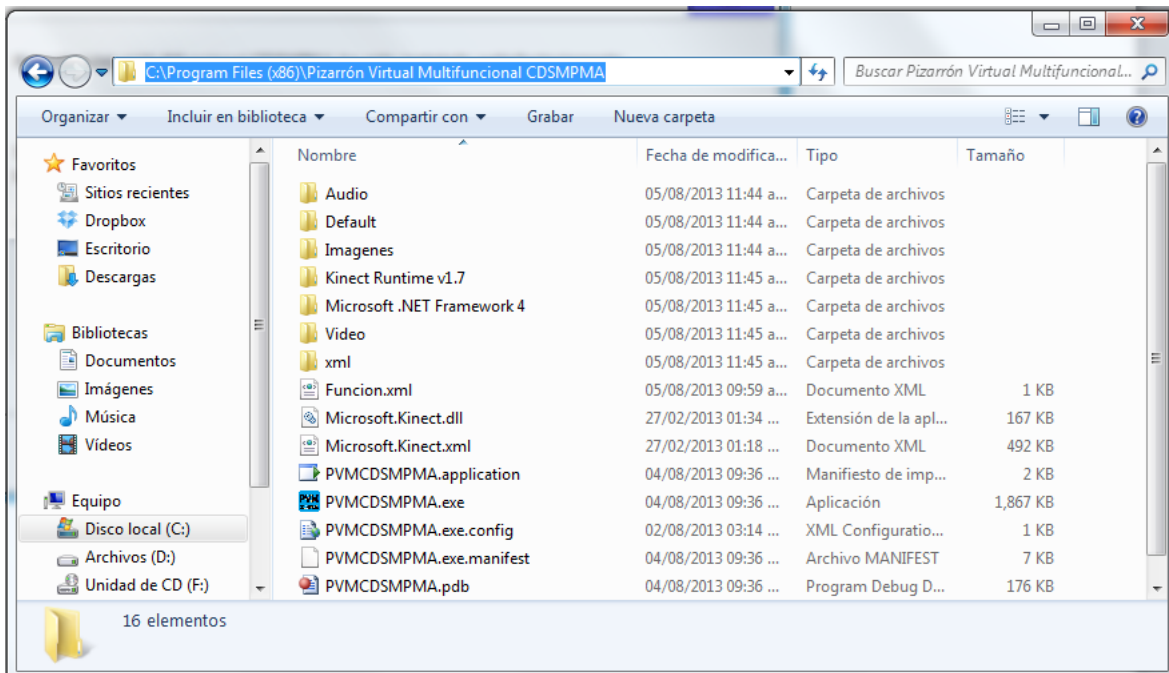
4.- Confirme para empezar la instalación



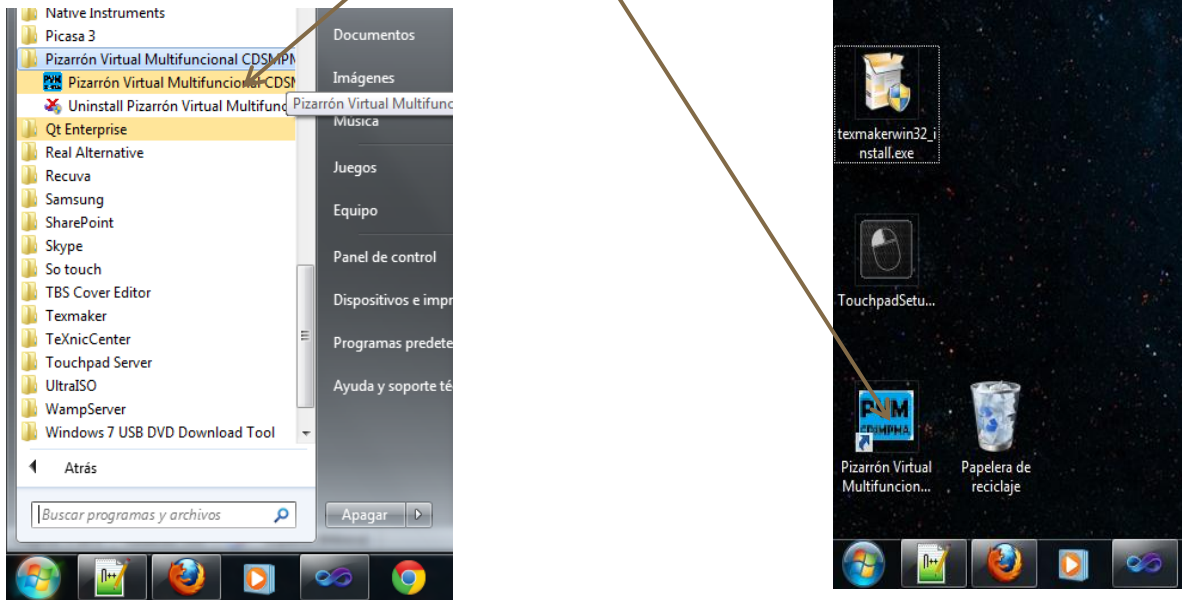
5.- Des seleccione ejecutar y continúe para finalizar



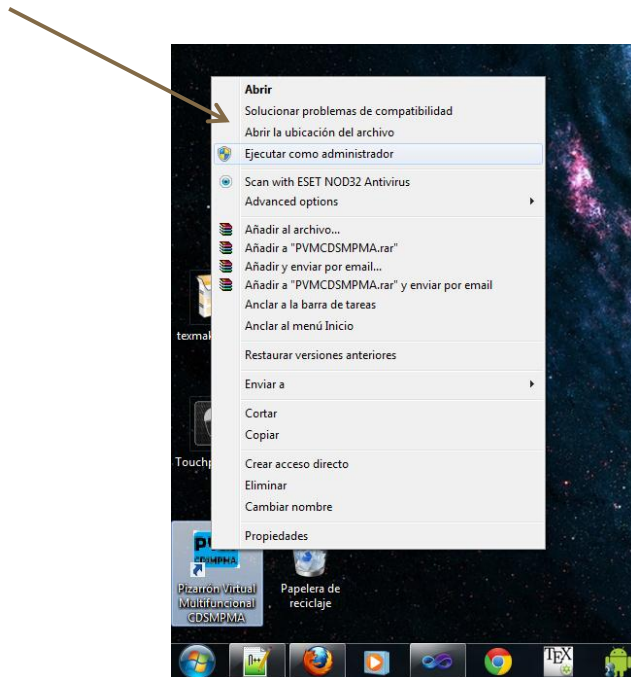
Durante la instalación se crearon los directorios necesarios para el programa.



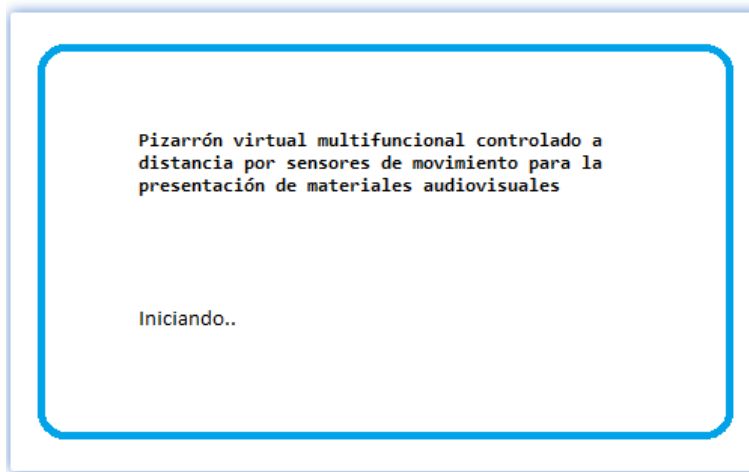
También se crearon los accesos directos dentro del menú inicio de Windows y el escritorio.



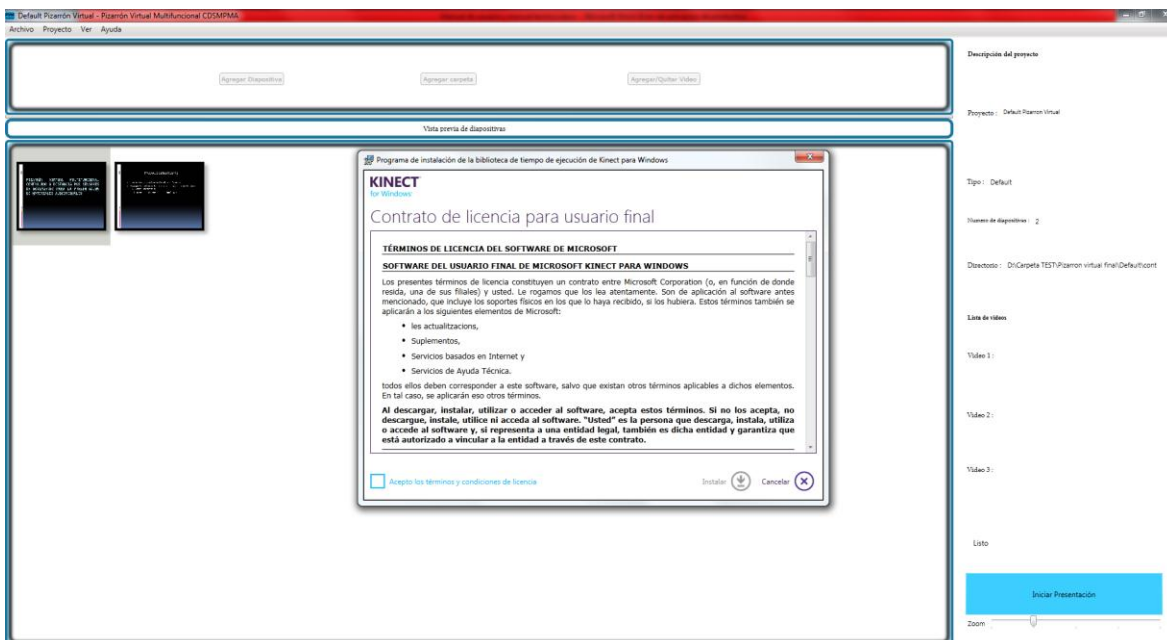
Para ejecutar el programa es necesario hacer Clic derecho sobre el acceso directo y ejecutar como administrador.



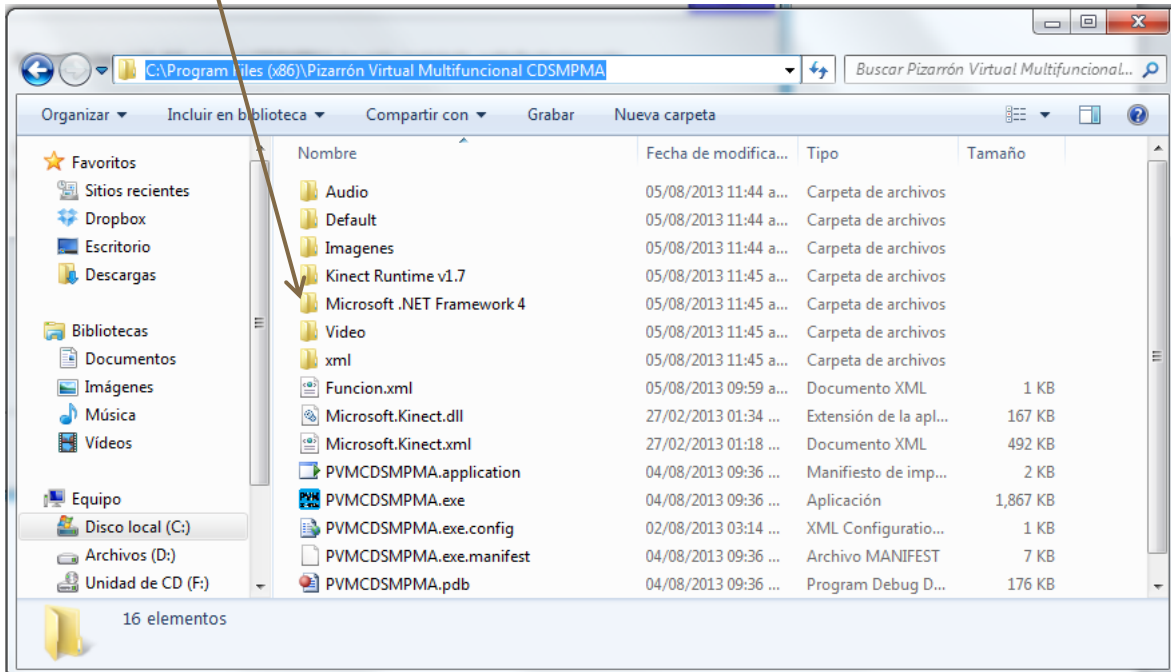
A continuación se iniciara la carga del programa, espere un momento.



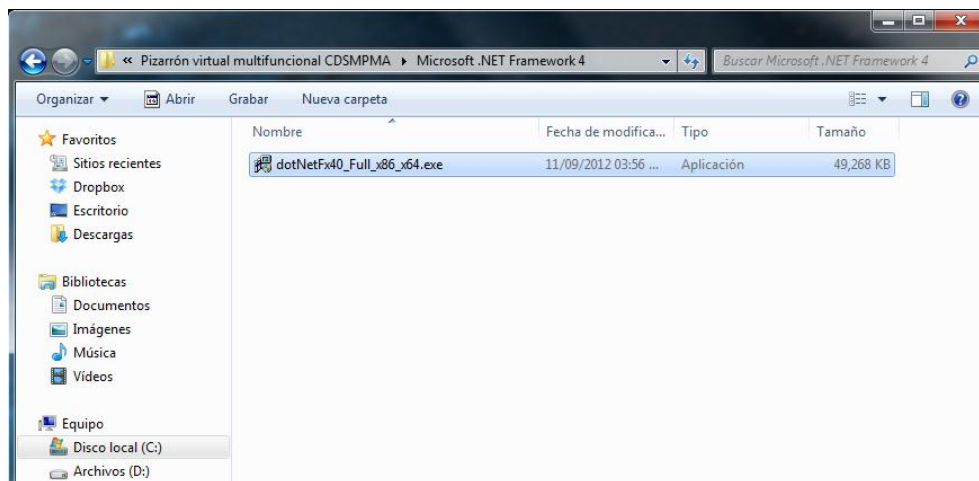
Durante el primer arranque si no se tiene instalado los drivers del Kinect se ejecutara Kinect Runtime. La instalación de este programa es primordial, puesto en él se incorporan los drivers y librerías que se necesitan para poder utilizar el programa Pizarrón Virtual Multifuncional CDSMPMA.



Después de haber instalado Kinect Runtime, se procede a instalar Microsoft .NET Framework 4, el instalador se encuentra ubicado en la dirección donde se instaló el programa.



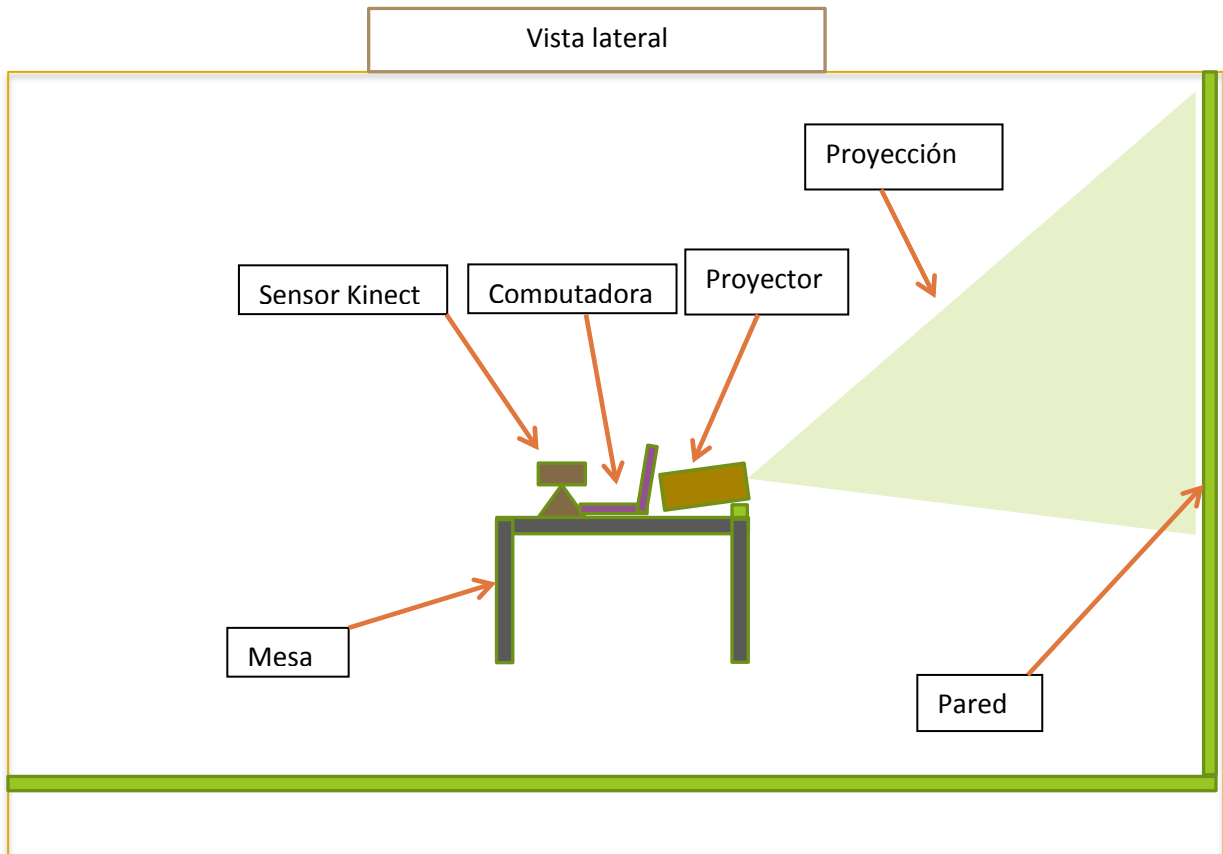
Haga doble clic sobre el instalador y siga los pasos.



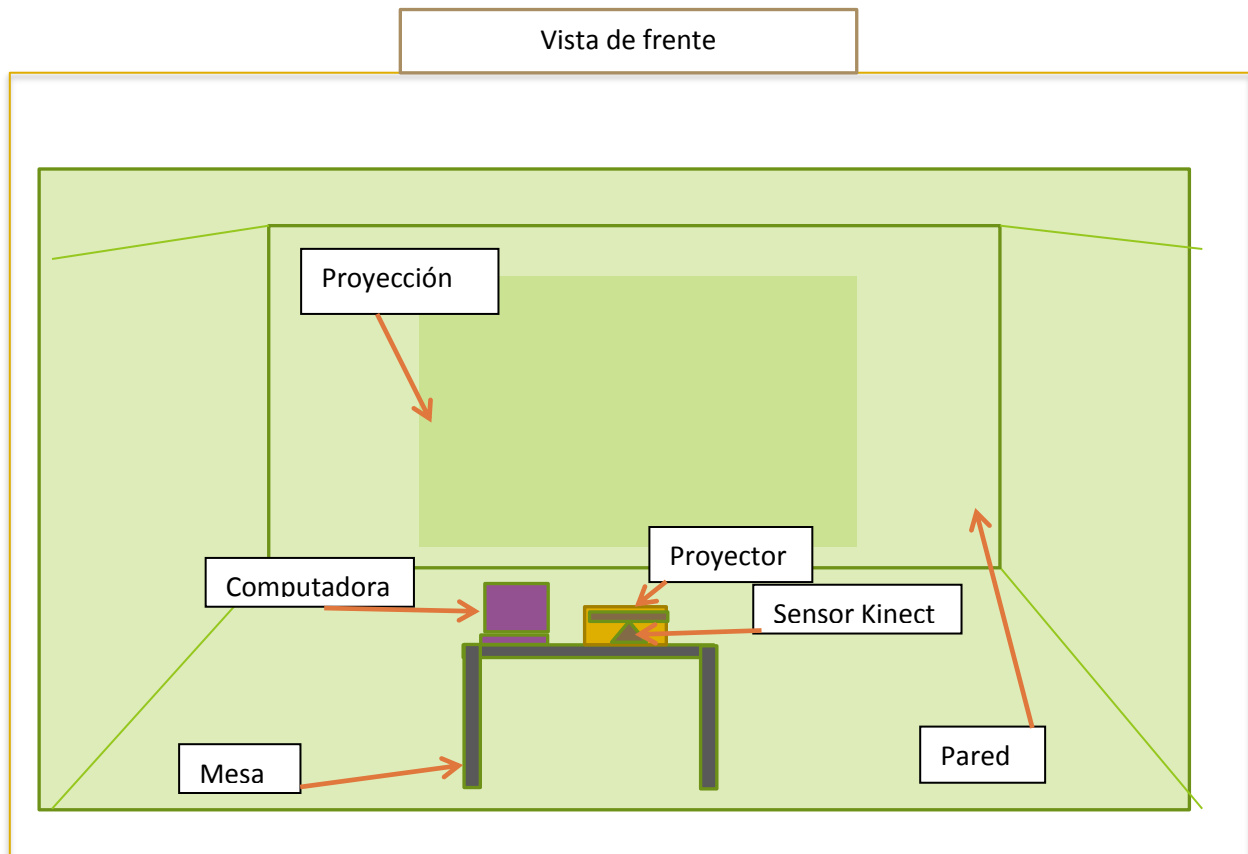
Después de haber finalizado con la instalación del Framework, el programa Pizarrón Virtual Multifuncional CDSMPMA está listo para su utilización.

2.- Instalación del sensor Kinect, proyector y computadora.

Instale los componentes como se representa en el siguiente diagrama



Para la instalación de los componentes se requiere de un espacio adecuado y sobre todo el espacio donde el usuario pueda moverse e interactuar con el programa.



Conexión de cables

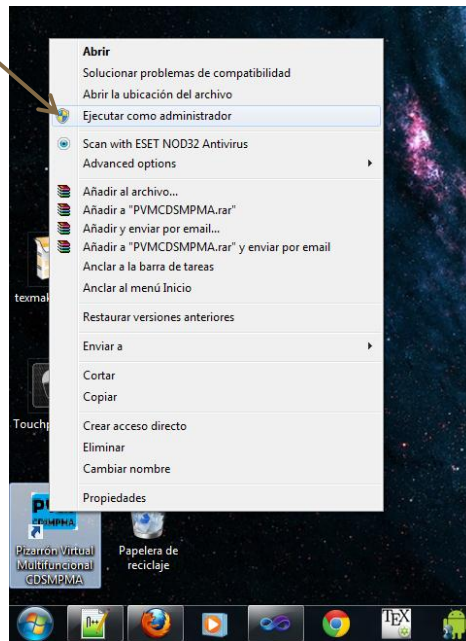
- 1.- Conecte el cable de alimentación del Kinect al suministro eléctrico.
- 2.- Conecte el cable USB del Kinect a la computadora.
- 3.- Conecte el cable de video del proyector a la computadora y al suministro eléctrico.

Estos pasos son necesarios para tener completa la interfaz del programa.

El programa solo funciona con el sensor Kinect para Windows, si trata de conectar el Kinect de la consola Xbox360 el programa no lo detectara y por lo tanto no funcionara.

3.-Ejecucion del programa.

Para iniciar el programa haga clic derecho sobre acceso directo y haga clic en ejecutar como administrador



La ejecución como administrador es importante haga siempre este paso antes de iniciar el programa.

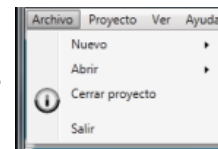
4.- Ventana de proyecto.

En esta ventana se encuentran los botones y un menú para la modificación de un nuevo proyecto.

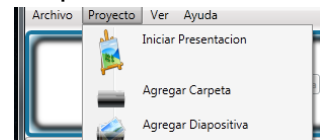
Al iniciar el programa se abre el proyecto default con el cual se puede comenzar a explorar el funcionamiento del programa con el Kinect.

El menú consta de:

Archivo: contiene el submenú nuevo, abrir cerrar proyecto y salir.



Proyecto: contiene Iniciar presentación, agregar carpeta y agregar diapositiva



En el menú ayuda contiene la descripción del programa así como su versión y fecha de terminación

Botones agregar diapositiva, agregar carpeta y agregar video.

Descripción del proyecto

Proyecto : Default Pizarra Virtual

Tipo : Default

Numero de diapositivas : 2

Directorio : C:\Program Files (x86)\VP

Lista de videos

Video 1 :

Video 2 :

Video 3 :

Listo

Iniciar Presentación

Zoom

Visualizador de diapositivas.

Área de descripción del proyecto

Lista de videos que contiene el proyecto

Botón para iniciar la presentación

Control para aumentar la vista previa de las diapositivas

Diapositivas

Vista previa de diapositivas

Archivos

Default Pizarra Virtual - Pizarra Virtual Multifuncional CDSMPMA

Archivo Proyecto Ver Ayuda

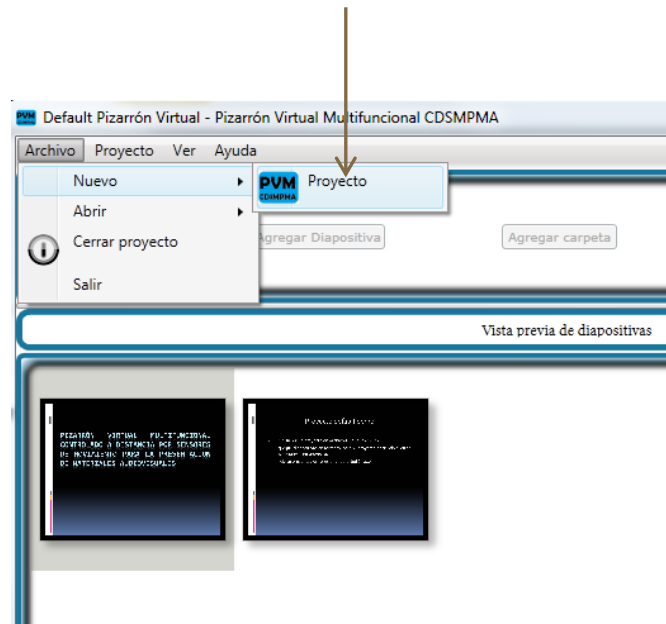
Agregar Diapositiva

Agregar Carpeta

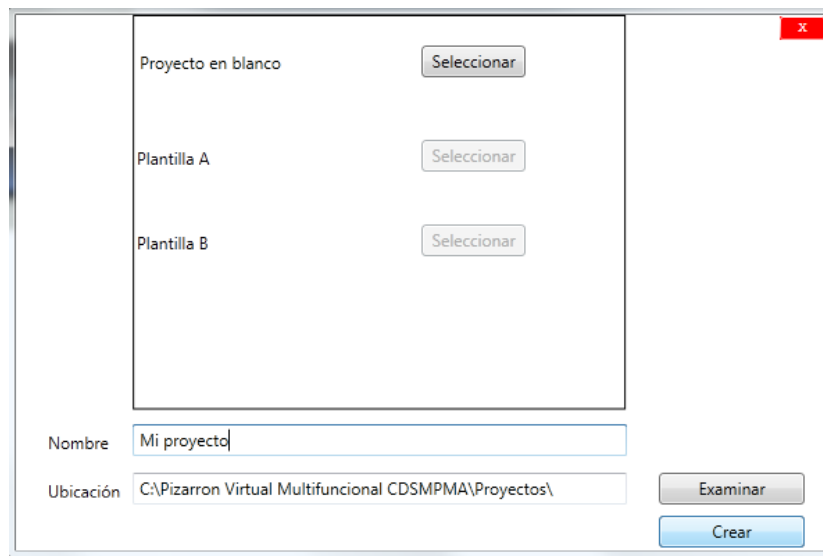
Agregar/Quitar Video

5.- Crear nuevo proyecto.

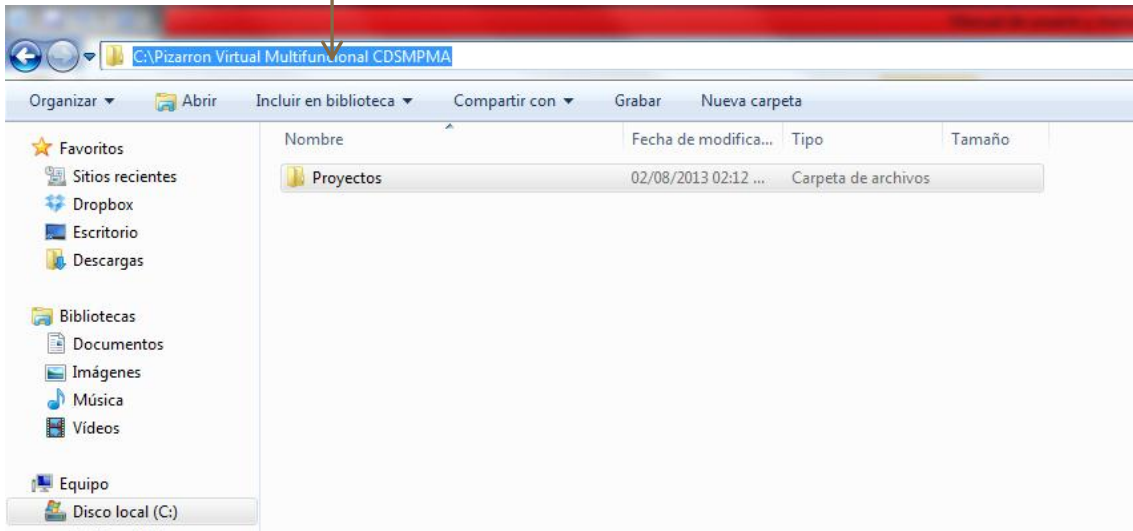
Para crear un nuevo proyecto haga clic en: Archivo -> Nuevo -> Proyecto



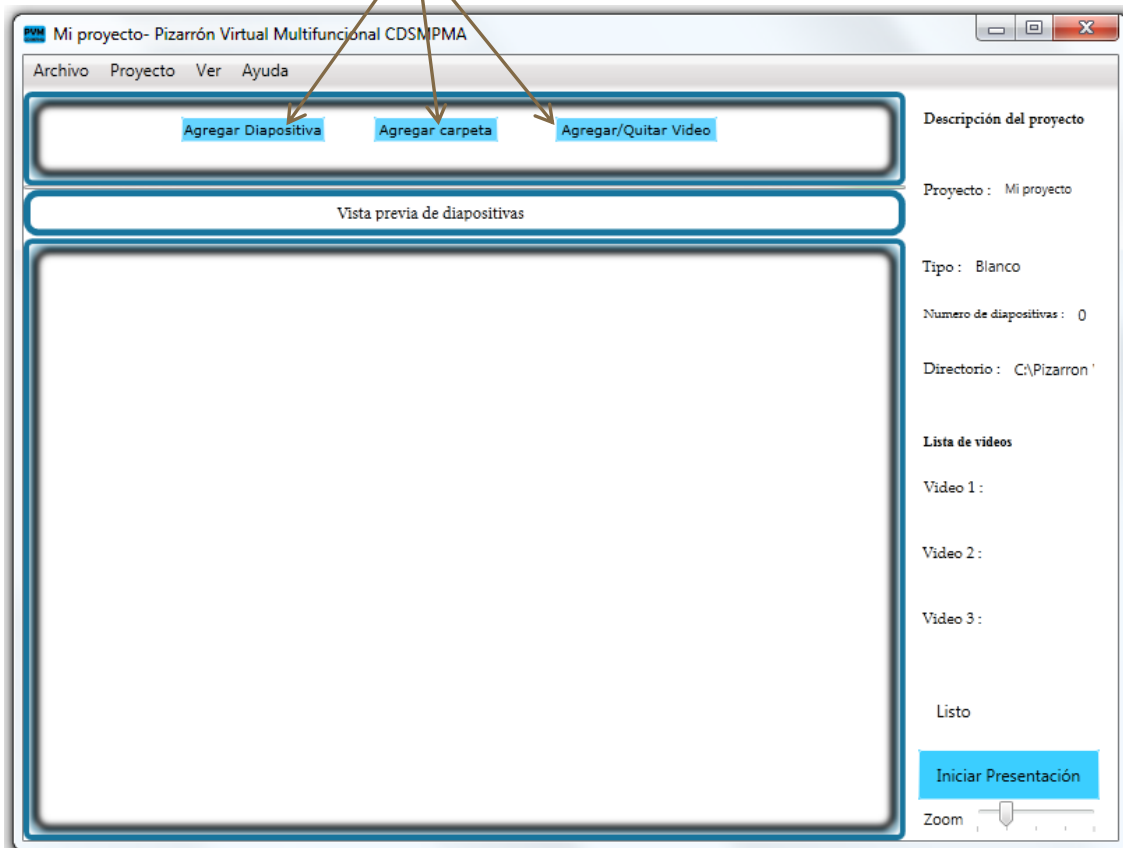
Escriba el nombre de su proyecto y haga clic en Crear.



Si el directorio principal donde se alojara su proyecto no existe, este se creara.



Cuando se finaliza la creación del nuevo proyecto, se habilitan los botones para que se proceda a modificar el proyecto agregando diapositivas o videos.



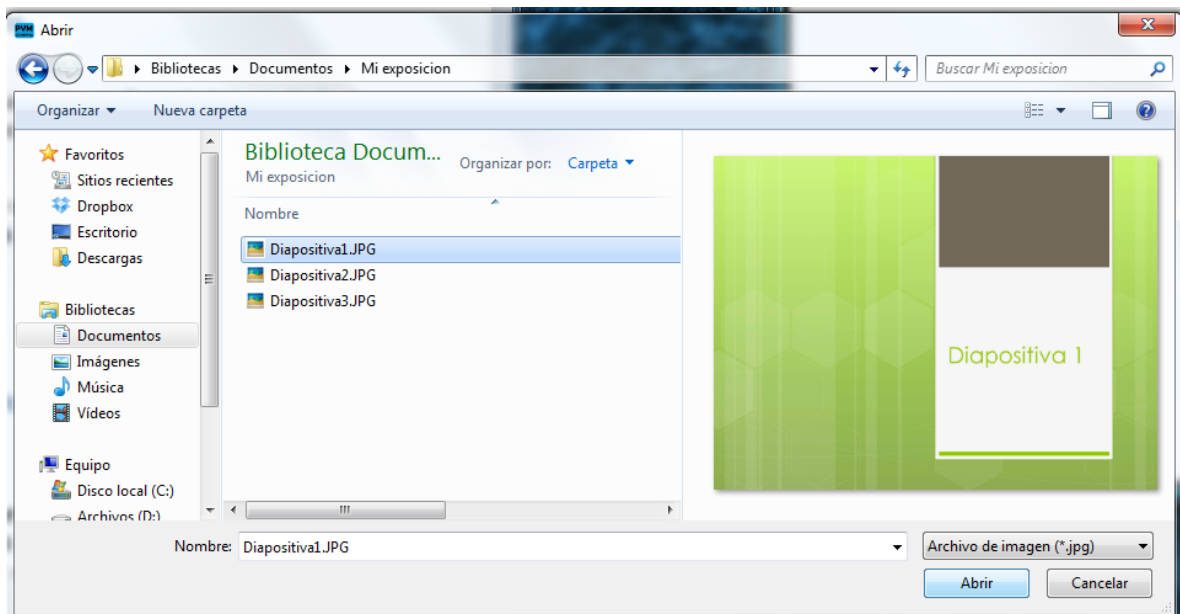
6.- Agregar diapositivas en él proyecto.

Nota: el proyecto solo admite diapositivas que estén en formato jpg, es decir si desea visualizar un proyecto que se haya hecho en PowerPoint primero tendrá que convertir sus diapositivas a formato jpg.

Existen dos formas para agregar diapositivas en el proyecto, agregando una diapositiva individual y agregando una carpeta entera.

Agregando diapositiva en forma individual.

- 1.- Haga clic en el botón Agregar Diapositiva
- 2.- Acceda al directorio donde se encuentra su diapositiva
- 3.- Seleccione su diapositiva
- 4.- Haga clic en Abrir

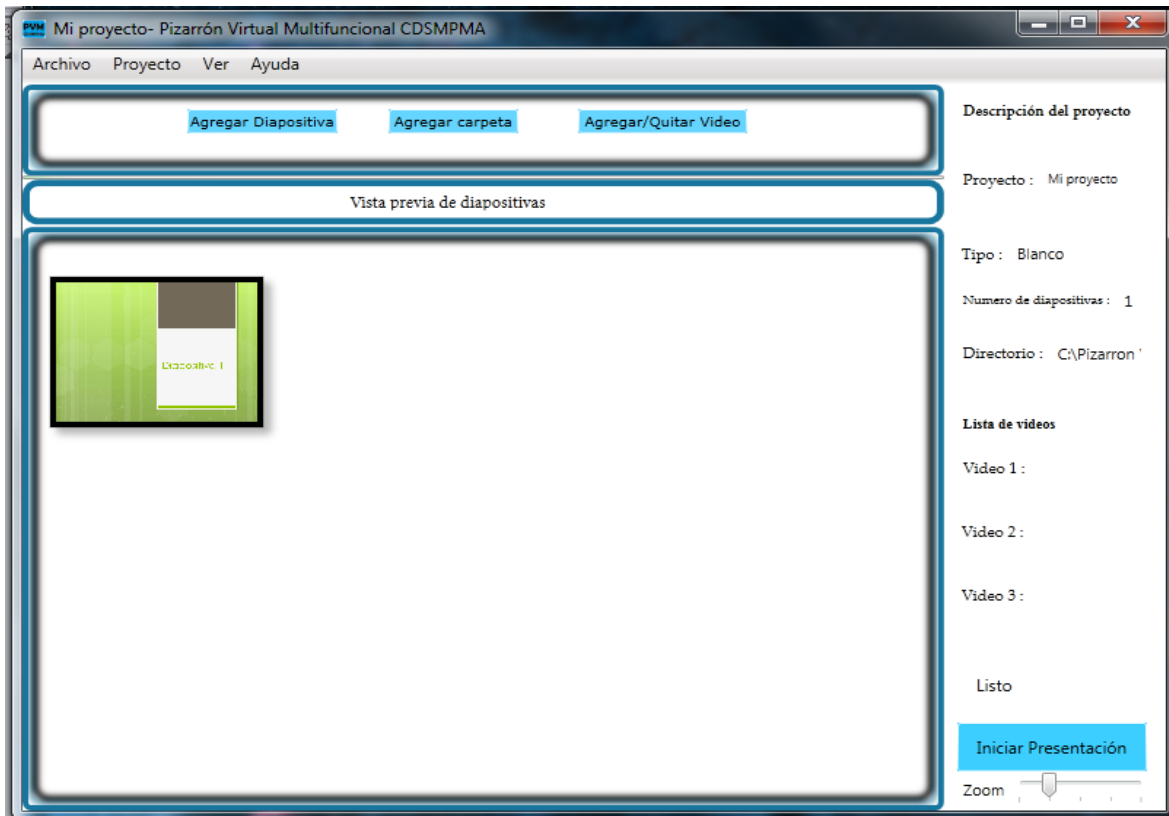


También se puede hacer este mismo procedimiento desde el menú Proyecto.

El número de diapositivas que puede contener el proyecto es ilimitado, pero se recomienda no rebasar más de 100 diapositivas.

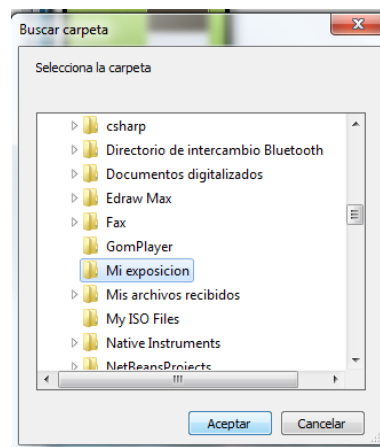
El tamaño de las diapositivas en formato .jpg es indefinida.

Al finalizar la vista previa cambiara y se mostrara la nueva diapositiva que se agregó al proyecto.



Agregar una carpeta completa al proyecto

- 1.- Haga clic en el botón Agregar Carpeta
- 2.- Acceda al directorio donde se encuentra la carpeta que desea agregar
- 3.- Seleccione su carpeta de diapositivas en formato jpg y haga clic en Aceptar



Al terminar de agregar aparecerá “Listo” dentro del área de descripción del proyecto, guíese de estos mensajes de estado para saber en qué condición se encuentra el programa.

El programa al estar agregando una carpeta completa puede tardar dependiendo de la cantidad de elementos que se encuentren en la carpeta que se está agregando.

Durante este procedimiento aparecerá “Agregando Carpeta” en el área de descripción del proyecto, al finalizar cambiar a “Listo”.

7.- Convertir diapositivas de PowerPoint a Formato de intercambio de archivos JPEG (jpg).

Debido a que el programa solo admite diapositivas en formato jpg, se explicara en este apartado como se deben de exportar una presentación completa hecha en PowerPoint aun conjunto de imágenes.

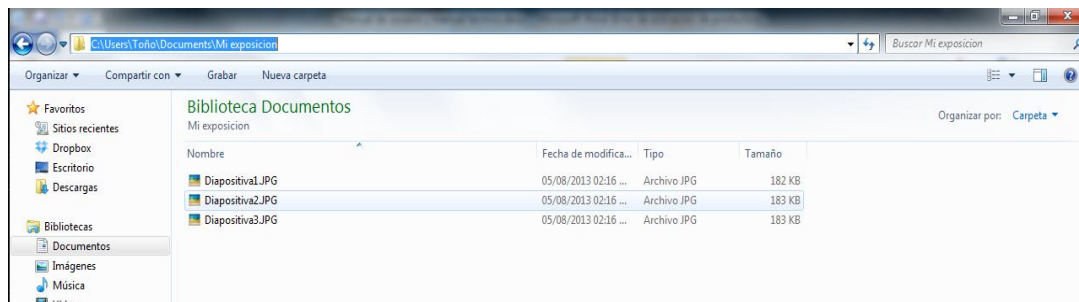
1.- Una vez que tenga completa su presentación en el programa PowerPoint haga clic en guardar como

2.- Seleccione el tipo Formato de intercambio de archivos JPEG

3.- Haga clic en guardar y después haga clic en exportar cada diapositiva, espere mientras finaliza el proceso.

Este paso es necesario si desea utilizar una presentación completa en PowerPoint en el pizarrón virtual.

Al final se creara una carpeta donde contendrá toda su presentación



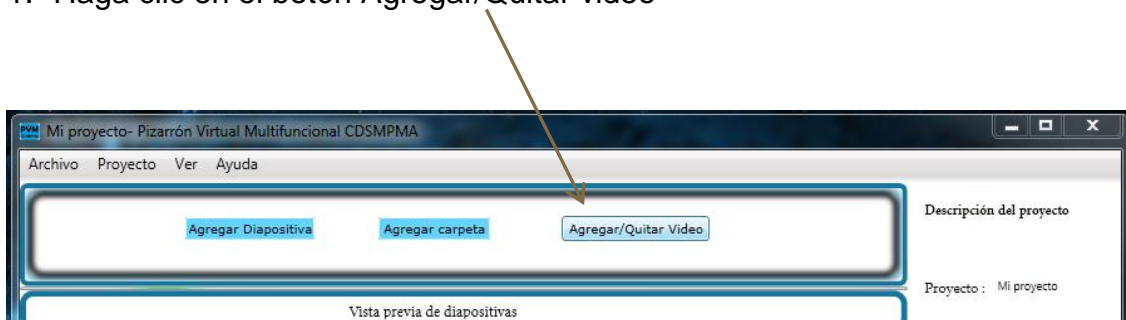
8.- Agregar/Quitar videos al proyecto.

En el proyecto se pueden agregar videos en formato MP4, de duración y tamaño indefinido. La cantidad de videos que se pueden contener es de tres.

Cada proyecto almacena la cantidad de tres videos que pueden reemplazarse por otros.

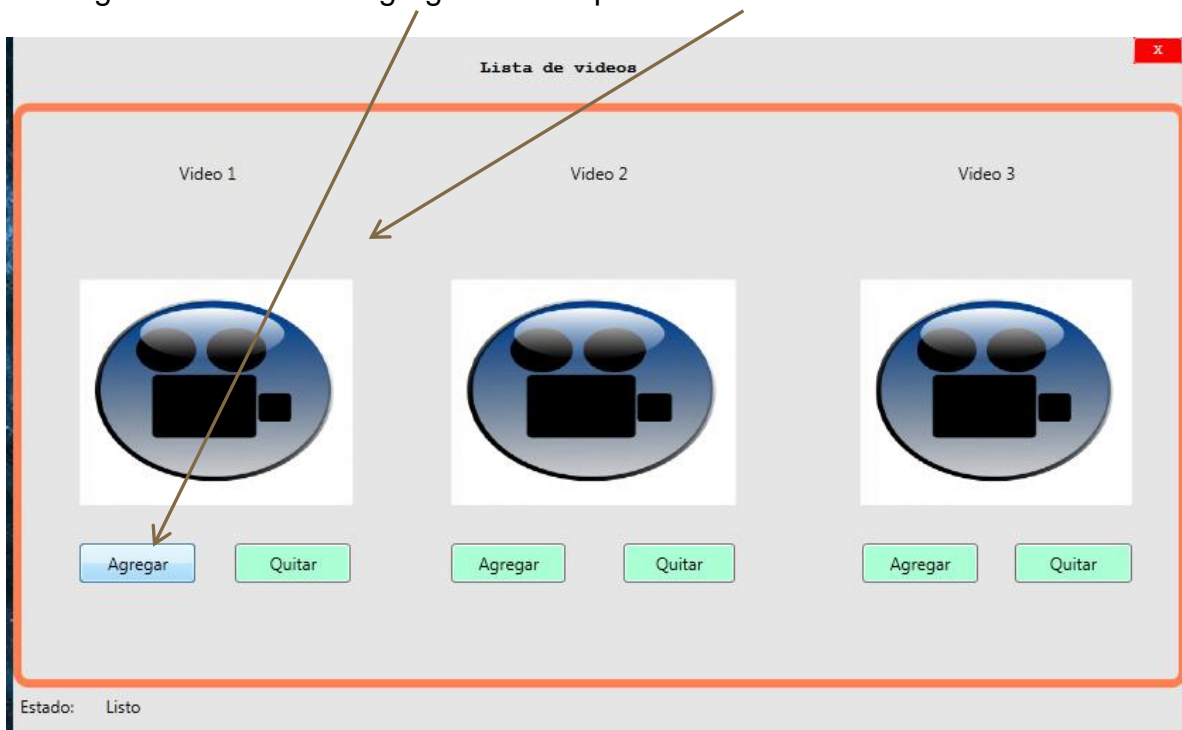
Para agregar videos a su proyecto siga los siguientes pasos:

1.- Haga clic en el botón Agregar/Quitar video



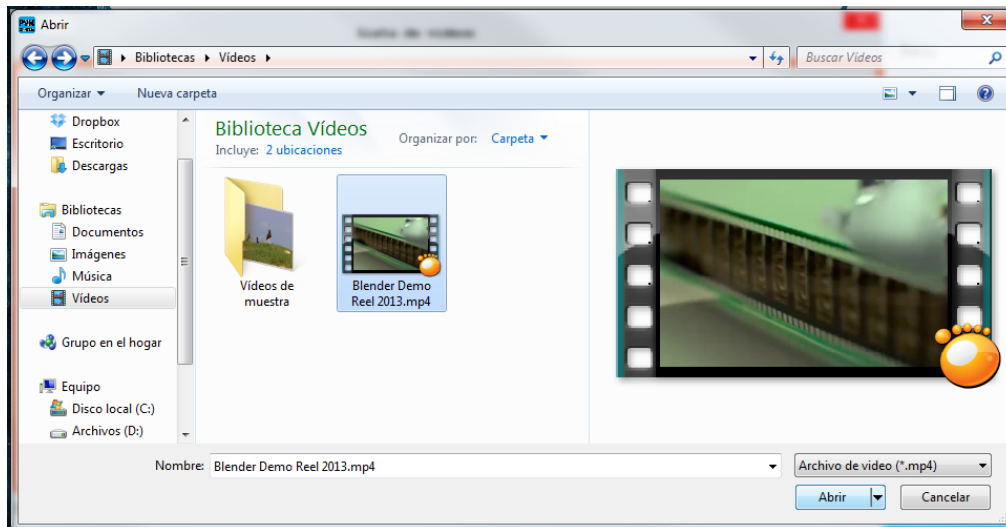
2.- Aparecerá una nueva ventana donde estarán los tres contenedores de video.

3.- Haga clic en el botón agregar en cualquier contenedor.



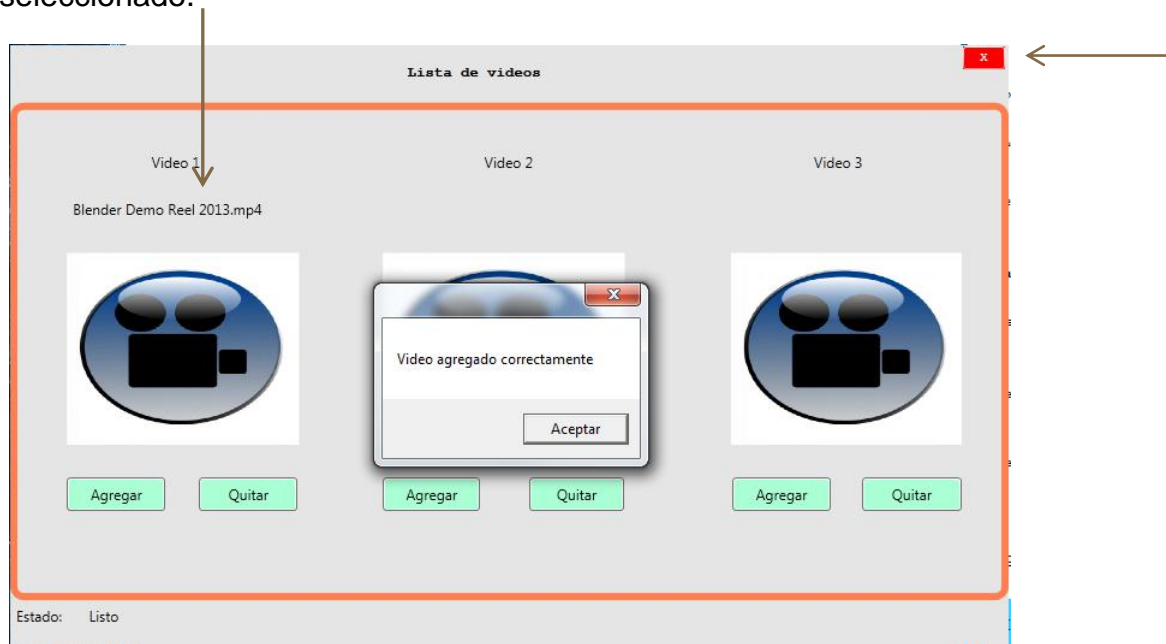
4.- Aparecerá una ventana donde podrá elegir el video que desee agregar a su proyecto.

5.- Seleccione el video y haga clic en Abrir.



6.- El programa le avisara cuando el video haya sido agregado al proyecto.

7.- Aparecerá el nombre del video que acaba de agregarse en el contenedor que haya seleccionado.



8.- Cuando haya terminado de agregar los videos necesarios haga clic en el botón cerrar.

9.- En el área de descripción del proyecto aparecerá la dirección donde se almaceno el video, los videos que se agregan son copiados al directorio especial dentro de una carpeta del proyecto que se esté utilizando.

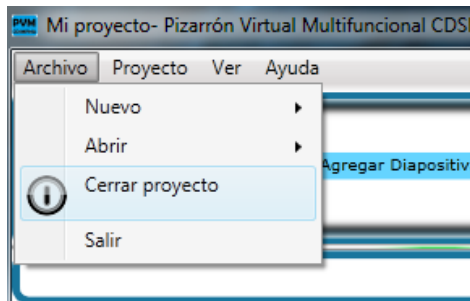


10.- Cuando desee eliminar un video de su proyecto haga clic en Agregar/quitar video y a continuación haga clic en el botón quitar para eliminar el video que se encuentra en el contenedor que quiere dejar en vacío.

11.- Al finalizar haga clic en el botón cerrar para regresar a la ventana de proyecto.

9.- Cerrar proyecto.

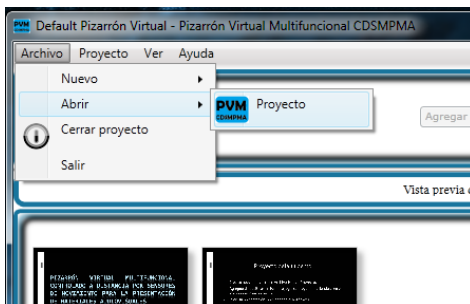
1.- Haga clic en Archivo y cerrar proyecto.



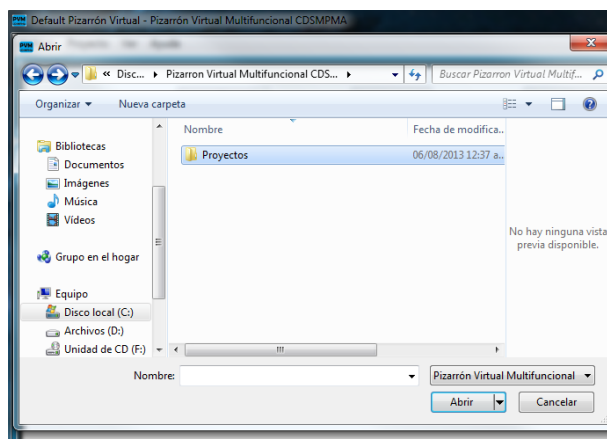
Al cerrar el proyecto que tenía abierto regresa al proyecto default automáticamente.

10.- Abrir proyecto.

1.- Para abrir un proyecto haga clic en Archivo, Abrir y Proyecto.

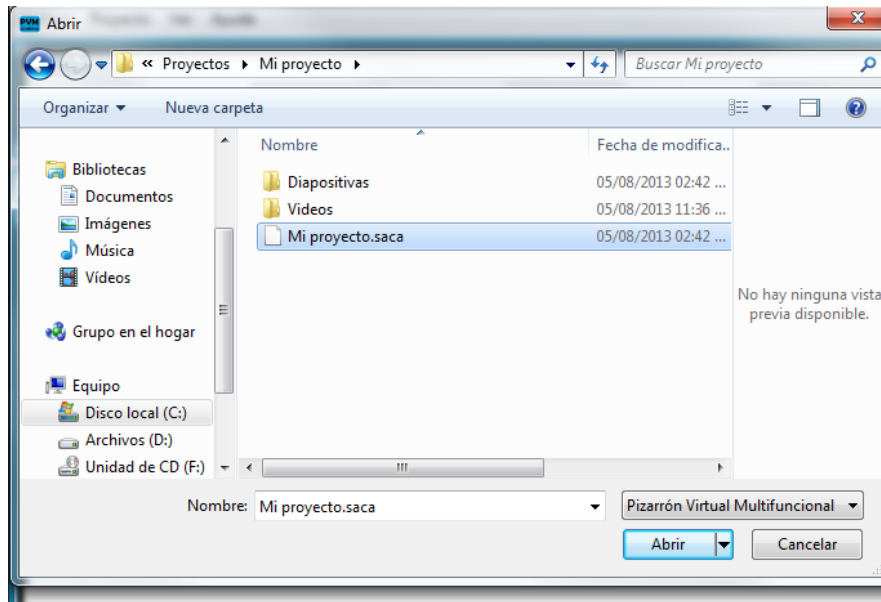


2.- Los proyectos quedan guardados por default en el directorio: C:\Pizarron Virtual Multifuncional CDSMPMA



3.- Abra la carpeta donde están los proyectos y después abra nuevamente la carpeta de su proyecto.

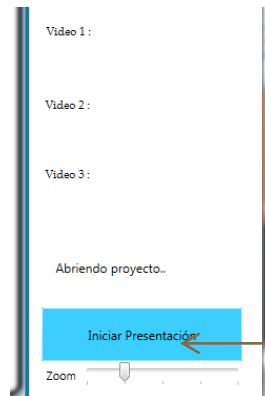
4.- Dentro de la carpeta de su proyecto se encuentra todo el contenido carpeta de diapositivas, carpeta de videos y el archivo con extensión .saca.



5.- Abra su proyecto seleccionando el archivo con extensión .saca, selecciónelo y haga clic en Abrir.

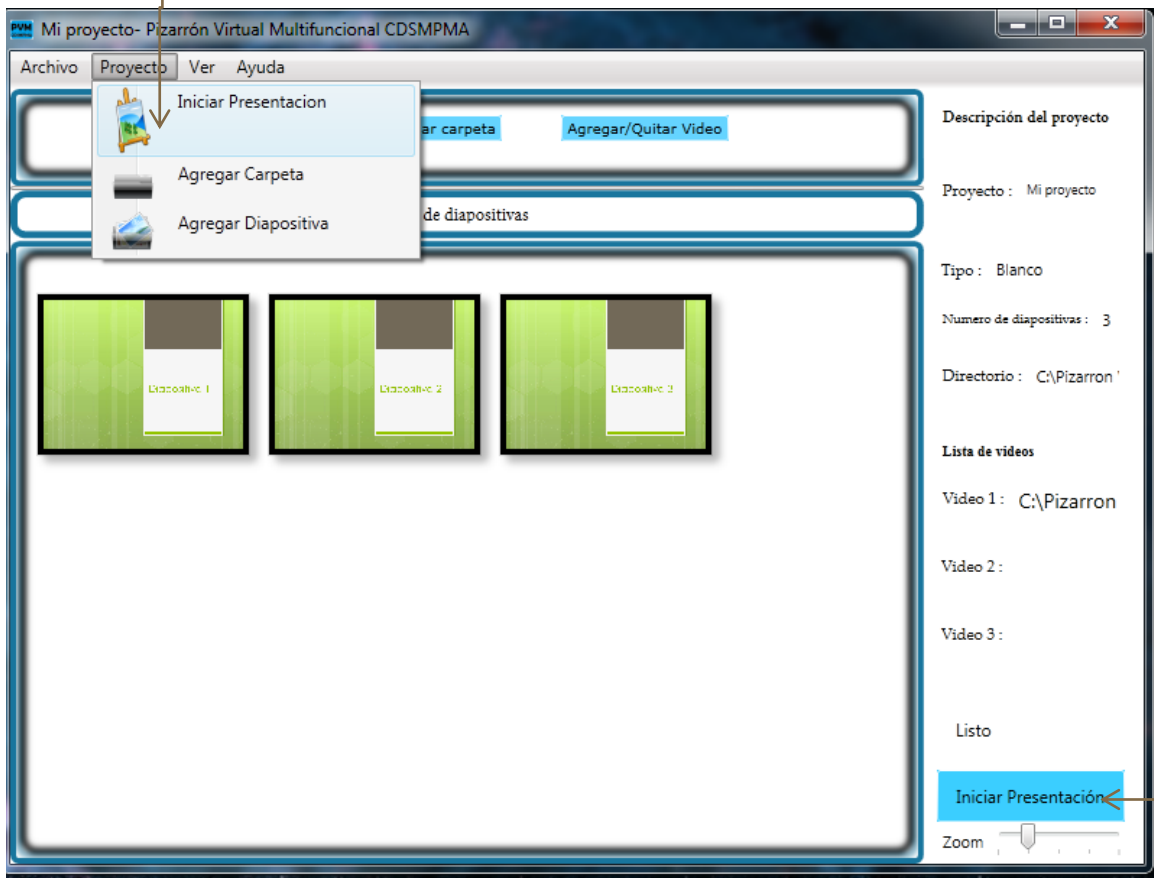
6.- Dentro del área de descripción del proyecto aparecerá el estado del programa.

7.- Espere mientras finaliza la apertura de su proyecto, cuando el estado del programa cambie a "Listo" se puede continuar con la edición del proyecto.



11.- Iniciar presentación.

Para iniciar su presentación haga clic en el botón iniciar presentación que se encuentra ubicado en el área de descripción del proyecto o bien en el menú Proyecto.



12.- Ventana modo presentador.

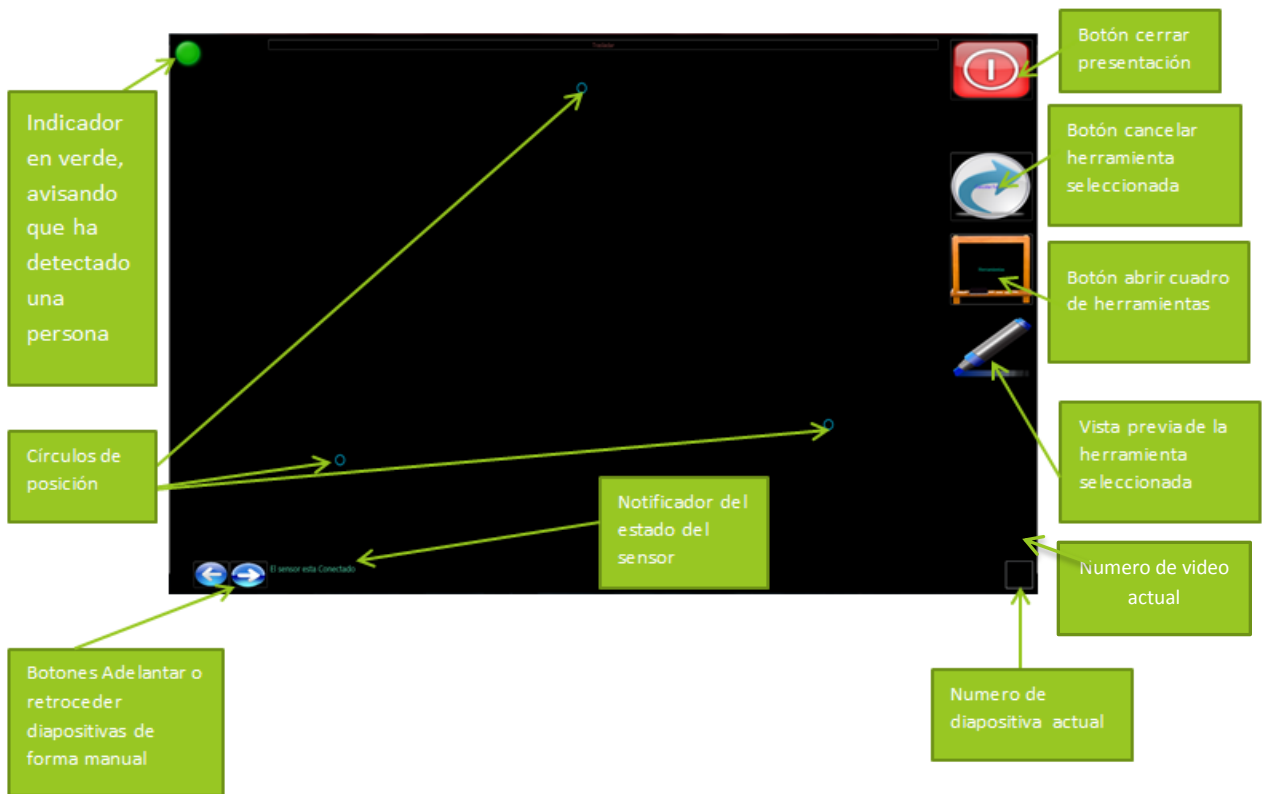
La ventana presentador es la encargada de visualizar las diapositivas y videos que se agregaron al proyecto.

Al iniciar el programa se reproduce una presentación al mismo tiempo que se cargan los componentes y se detecta el sensor Kinect.



La ventana modo presentador está compuesto de botones, iconos y marcadores que le ayudan a saber en todo momento en que numero de video o diapositiva se está utilizando, además de informar el estado en la que se encuentra el sensor Kinect, también cuenta con los iconos que señalan si existe una persona delante del sensor así como la herramienta que se esté utilizando en el momento.

A continuación se nombran los elementos de la ventana modo presentador:



Indicador: El icono cambia automáticamente en verde cuando detecta a una persona en el área de visión del sensor y cambia en rojo cuando no detecta a ninguna.

Círculos de posición: Estos círculos le ayudaran a saber en qué posición se encuentran su mano derecha, mano izquierda y cabeza, además sirven de señaladores al momento de presentar su material de exposición.

Botones adelantar o retroceder: Cumplen con la tarea de adelanta o retroceder las diapositivas que se estén proyectando de manera manual.

Notificador del estado del sensor: Su función es desplegar mensajes sobre el estado en que se encuentra el sensor. Las notificaciones que manda el programa son: Iniciando sensor, el sensor está conectado, el sensor no está conectado con el adaptador de corriente, el sensor no se ha iniciado, el sensor no es Genuino, el sensor esta desconectado, no tiene conectado el sensor Kinect para Windows.

Botón cerrar presentación: Botón para cerrar la ventana modo presentador, este botón funciona cuando se hace clic sobre él, no funciona utilizando los gestos de clic a menos que esté utilizando una herramienta y tenga activada la opción de clic.

Numero de diapositiva y de video: Como su nombre lo indica muestran el número actual de la diapositiva o del video.

Icono de herramienta seleccionada: Los iconos cambiarán automáticamente cuando usted seleccione una herramienta.

Botón abrir cuadro de herramienta: Cuando necesite utilizar una herramienta solo tendrá que mover el círculo de su mano derecha hasta el botón, con solo pasara su mano sobre el botón se abrirá la ventana de herramientas.

Botón cancelar herramienta o continuar: La función de este botón es deshabilitar la herramienta que se esté utilizando, una vez que se haya dejado de utilizar y se requiera continuar con la presentación, se tiene que activar, posicione el cursor y círculo azul sobre el botón, automáticamente desaparecerá el icono de la herramienta que se esté utilizando.

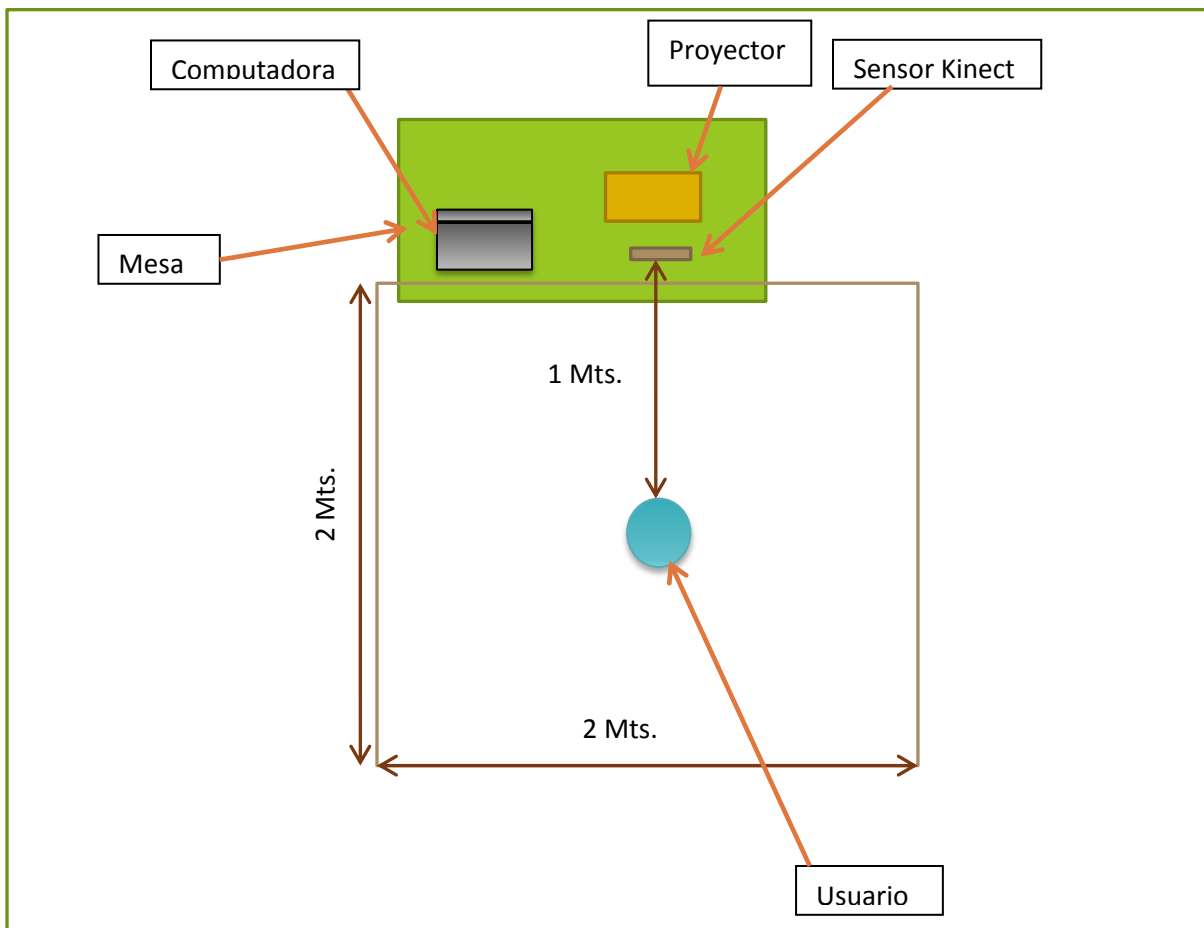
13.- Utilizar la ventana modo presentador con gestos.

Antes de empezar asegúrese de que tiene el suficiente espacio para moverse libremente, se recomienda dos metros cuadrados.

Para controlar el programa mediante gestos es necesario seguir estos pasos:

- 1.- Posiciónese a 1 metro de distancia del sensor.
- 2.- Asegúrese de que no tiene a otras personas cerca del área de Tracking del sensor.
- 3.- Extienda sus ligeramente sus brazos en forma lateral.
- 4.- Cuando el indicador este en color verde significa que puede empezar a utilizar el programa, notara que se moverán los círculos azules de las manos y cabeza.

Vea el siguiente diagrama, visto desde arriba:



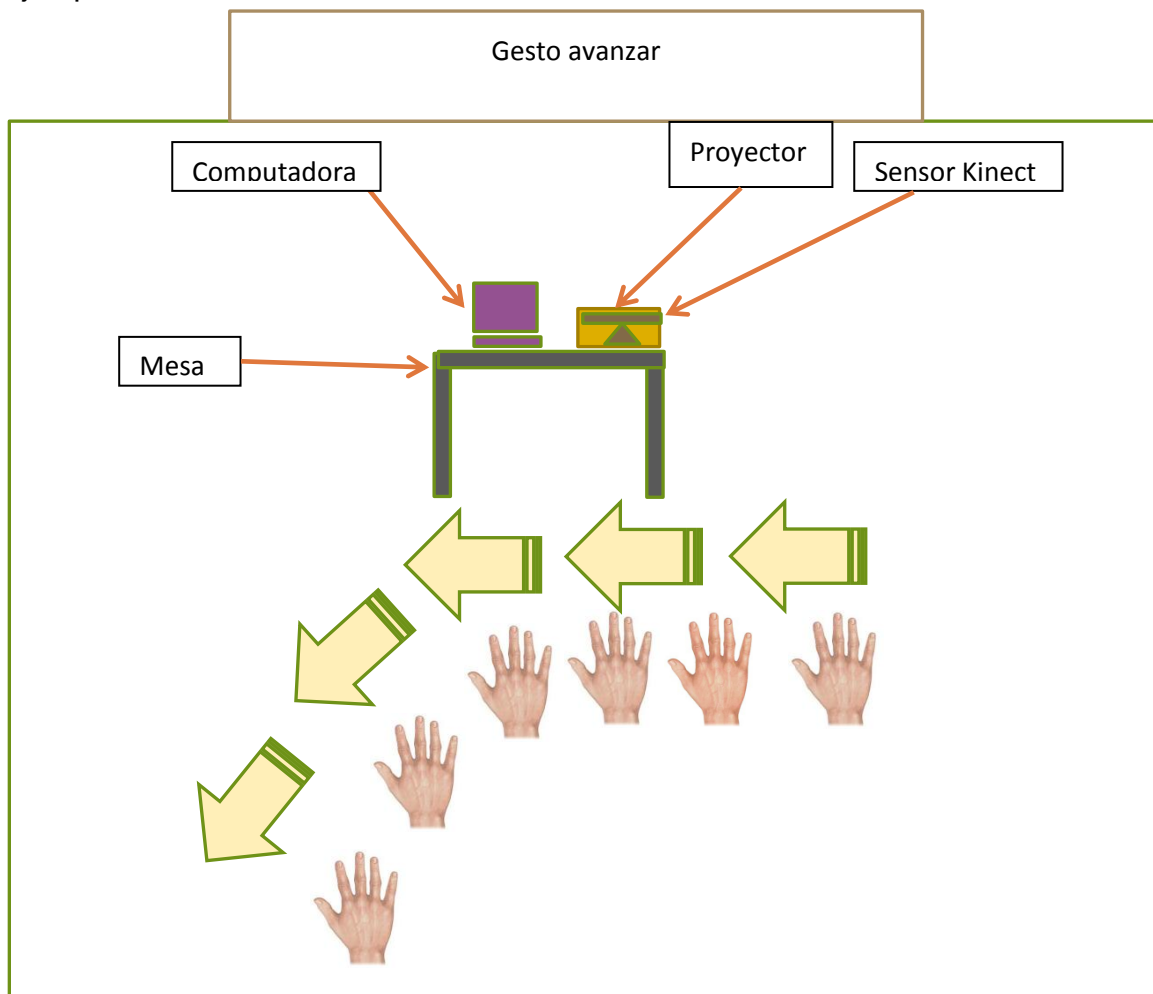
Cuando se acerque al sensor a una distancia inferior a un metro, el programa dejara de reconocerlo automáticamente y si se aleja a una distancia superior a dos metros, el indicador cambiara a rojo y los círculos azules dejaran de moverse.

Gesto Avanzar

Para avanzar una diapositiva haga el siguiente movimiento.

1.- Posicione su mano derecha frente al sensor y haga un movimiento deslizante rápido de derecha a izquierda bajando a la vez su mano para que el programa reconozca el gesto.

Ejemplo:

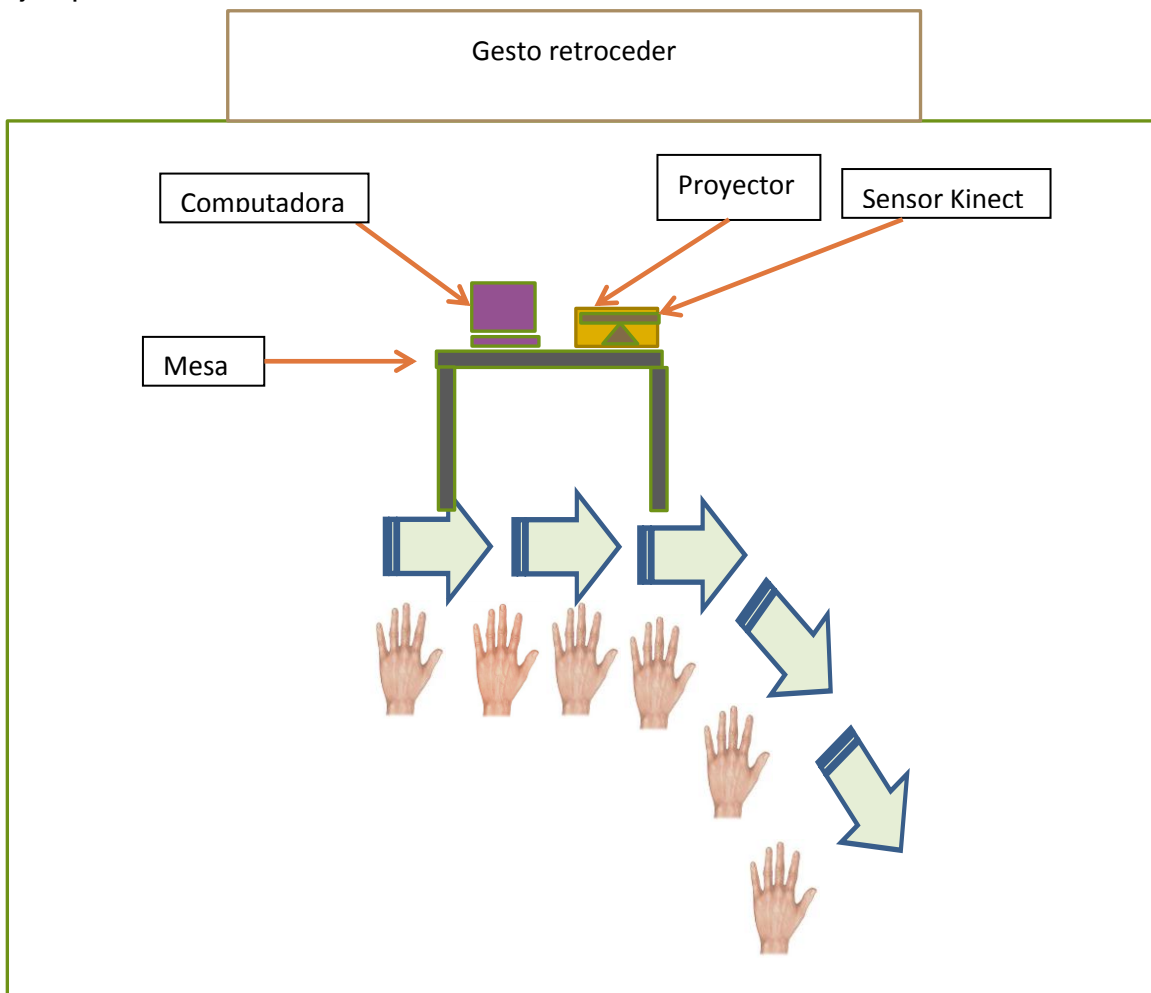


Gesto Retroceder

Para retroceder una diapositiva haga el siguiente movimiento.

1.- Posicione su mano izquierda frente al sensor y haga un movimiento deslizante rápido de izquierda a derecha bajando a la vez su mano para que el programa reconozca el gesto.

Ejemplo:



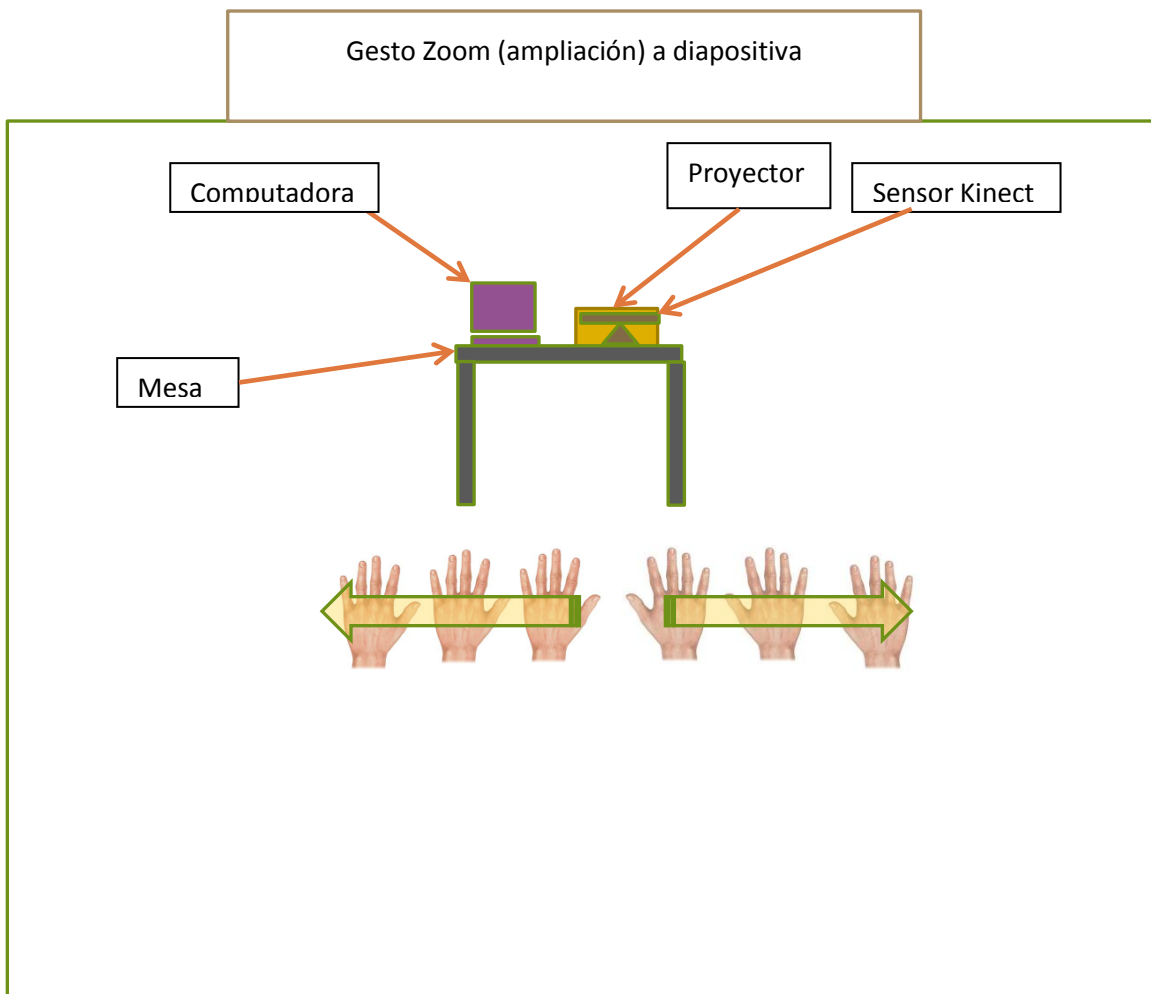
Gesto Zoom (ampliar o reducir diapositiva)

El Zoom funciona únicamente con la diapositiva que se está proyectando, las figuras que se hayan hecho con las herramientas no cambiarán, tampoco tendrá cambio la reproducción de video.

Para realizar Zoom sobre la diapositiva siga los siguientes pasos.

1.- posicione su mano derecha e izquierda al par a la altura de su pecho, para hacer el zoom (ampliación) haga una abertura de sus manos llevando su mano derecha e izquierda hacia fuera como si estuviese extendiendo los brazos.

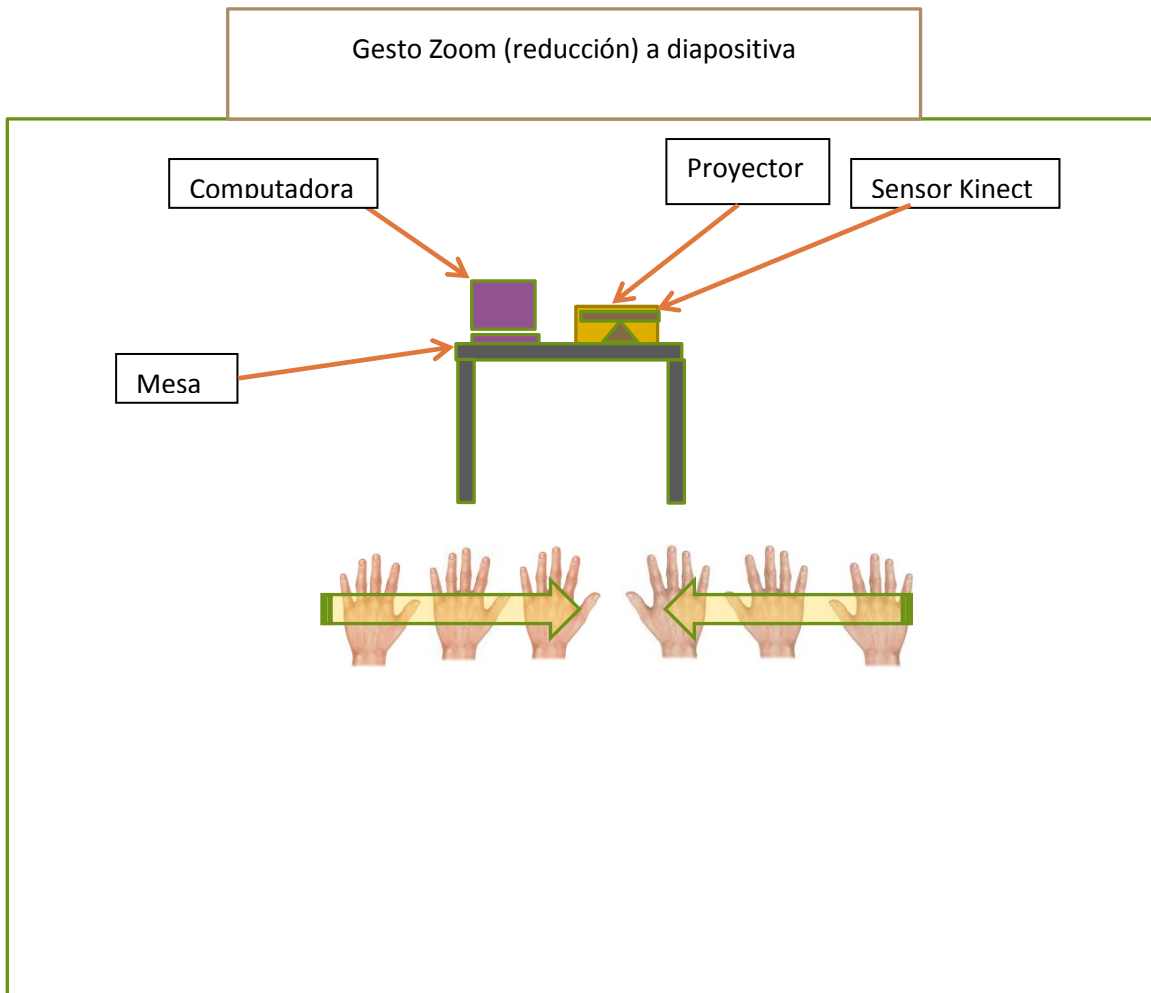
Ejemplo:



Para realizar Zoom (reducir) sobre la diapositiva siga los siguientes pasos.

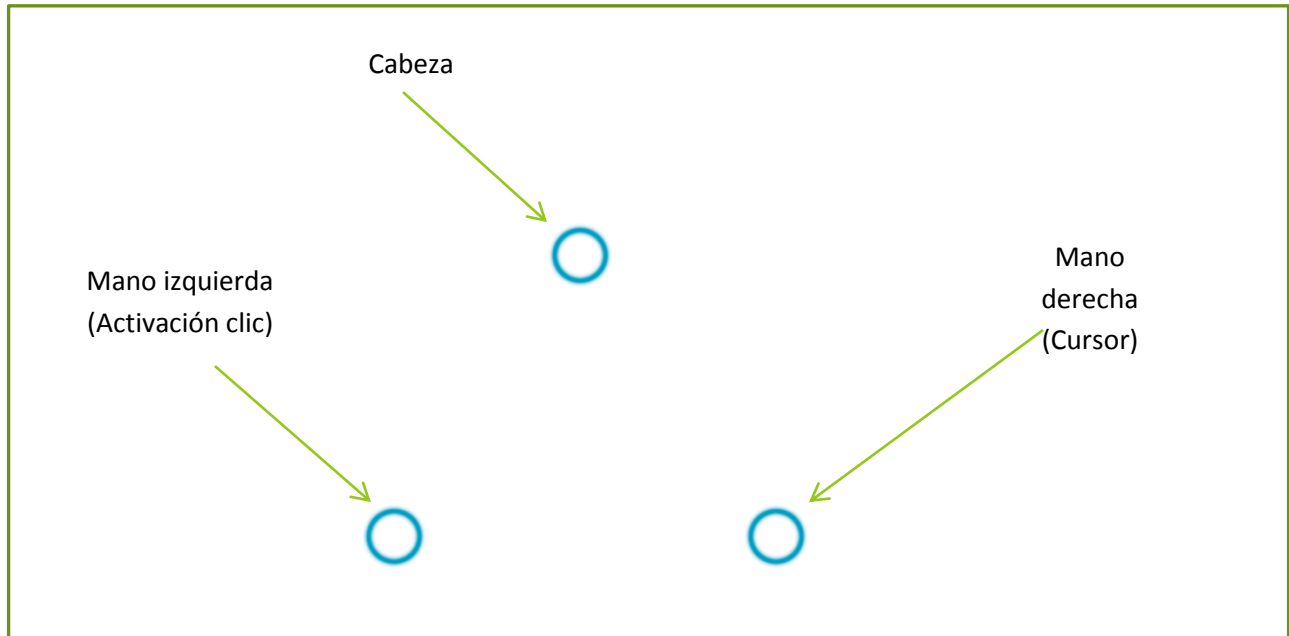
1.- posicione sus mano derecha e izquierda extendidas al par a la altura de su pecho, para hacer el zoom (reducción) haga un cierre de sus manos llevando su mano derecha e izquierda hacia dentro tratando de unir las dos manos

Ejemplo:



Gesto Seleccionar y hacer clic

El programa incorpora de un cuadro de herramientas el cual contiene botones y cada uno de ellos representa una herramienta como tal. Para seleccionar y activar una de ellas es necesario utilizar las dos manos, la mano derecha representa el cursor y la mano izquierda representa la activación del clic.

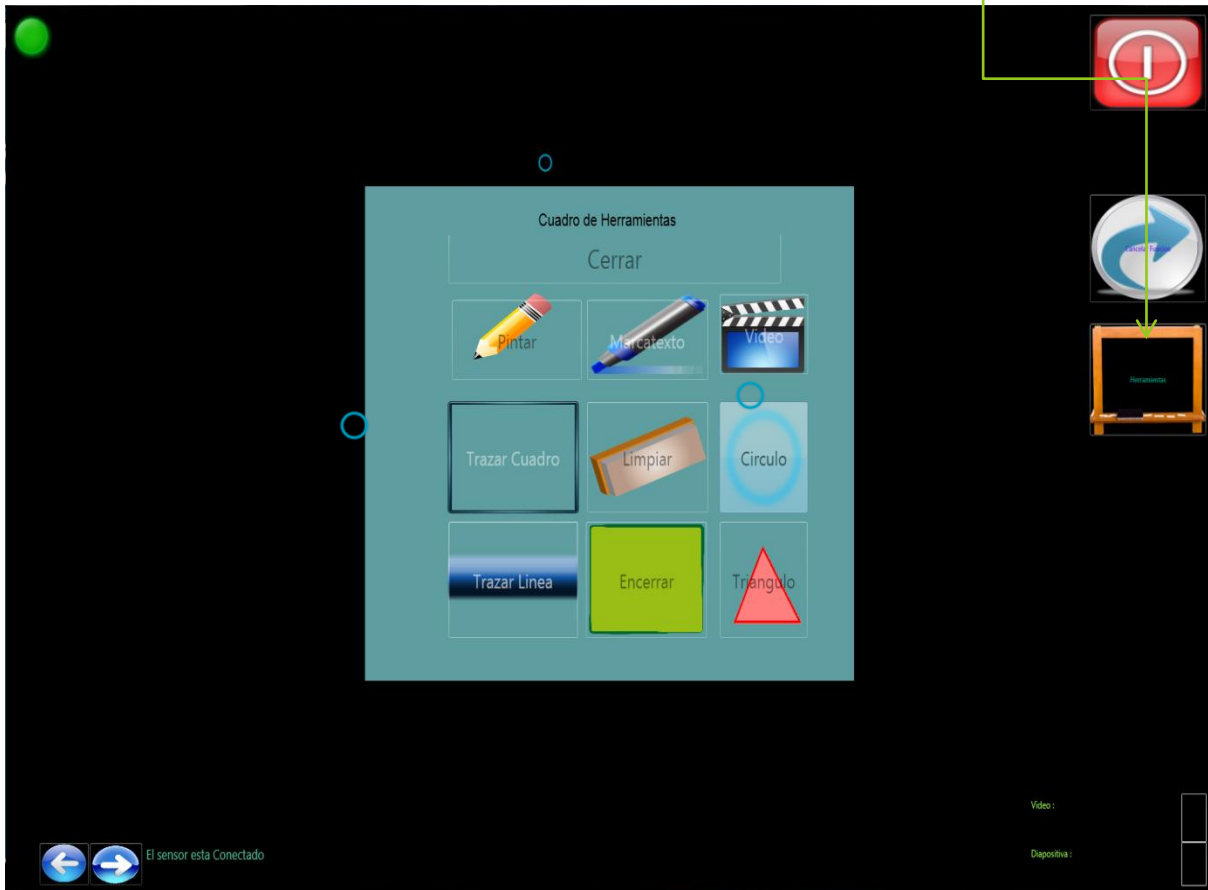


Para mover el cursor solo tendrá que mover su mano derecha y dirigir el cursor en el lugar que quiera, en el caso de activar la selección o hacer clic , levante ligeramente su mano izquierda y haga el gesto como si empujara el botón, este movimiento es rápido y no llevara más de un segundo realizarlo.

14.- Cuadro de herramientas.

Para seleccionar una herramienta es necesario que lleve su mano al icono que tiene forma de pizarrón, cuando pase su cursor sobre él, aparecerá automáticamente el cuadro donde contendrá cada una de las herramientas.

Cuando necesite utilizar una herramienta haga los pasos



Herramientas

El cuadro consta de las nueve herramientas:

Pintar, Marcatexto, Video, Trazar Cuadro, Limpiar, Circulo, Trazar Línea, Encerrar y Triangulo

Pintar: Usted podrá pintar libremente con la mano derecha y con la mano izquierda podrá dejar líneas segmentadas de color blanco, mismas que servirán como borrado a mano alzada. Para pintar es necesario tener su mano ligeramente hacia delante de todo su cuerpo, como si estuviese haciendo clic, cuando baje su mano automáticamente dejara de pintar.

Puntos de referencia

Para utilizar las herramientas Marcatexto, Trazar Cuadro, Circulo, Trazar Línea y Encerrar, es necesario dejar puntos en el área de proyección.

Cada herramienta trabaja utilizando los puntos de referencia, para dejar los puntos haga el gesto **Seleccionar y hacer clic** en el área que desee dejar el punto. Cuando deje los puntos aparecerán automáticamente puntos de color rojo.

Marcatexto: Esta herramienta utiliza dos puntos de referencia, el inicial y el final, para utilizarlo deje sus puntos de referencia sobre la pantalla haciendo el gesto **Seleccionar y hacer clic**.

Trazar Cuadro: Utiliza dos puntos de referencia, el inicial y el final, para utilizarlo deje sus puntos de referencia sobre la pantalla haciendo el gesto **Seleccionar y hacer clic**. El ancho y el largo usted lo definirá con forme retire el punto final del eje X o deleje Y.

Circulo: Utiliza dos puntos de referencia, el inicial y el final, para utilizarlo deje sus puntos de referencia sobre la pantalla haciendo el gesto **Seleccionar y hacer clic**. EL tamaño del circulo lo definirá dependiendo del largo del radio que usted defina, el radio se representa del punto inicial al punto final.

Trazar Línea: Utiliza dos puntos de referencia, el inicial y el final, para utilizarlo deje sus puntos de referencia sobre la pantalla haciendo el gesto **Seleccionar y hacer clic**. El largo y la posición de la línea, horizontal, vertical etc, usted lo definirá dependiendo de la posición que deje los puntos de referencia.

Encerrar: Utiliza dos puntos de referencia, el inicial y el final, para utilizarlo deje sus puntos de referencia sobre la pantalla haciendo el gesto **Seleccionar y hacer clic**. El ancho y el largo usted lo definirá con forme retire el punto final del eje X o deleje Y.

Video: Esta herramienta reproduce los videos que tenga el contenedor, para adelantar o retroceder la lista de los videos haga el gesto **Adelantar** o **Retroceder**.

La herramienta provee de botones extra de control de reproducción (reproducir, pausa y detener), para activarlos pase ligeramente el cursor de su mano derecha sobre ellos.



Cancelar herramienta seleccionada o continuar:

Este botón funciona para cancelar la herramienta que esté utilizando actualmente, al pasar su cursor sobre ella todas las herramientas se cancelan, incluso si está reproduciendo un video este se detendrá automáticamente. Utilícela cuando necesite controlar sus diapositivas, para adelantar, retroceder o hacer zoom.





INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

**SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE**

SEGUIMIENTO DE PROYECTO DE RESIDENCIAS PROFESIONALES

ALUMNO: Sánchez Alfonso José Antonio No. DE CONTROL: 09270248
 NOMBRE DEL PROYECTO: Pizzarrón Virtual Multifuncional Controlado a Distancia por Sensores de Movimiento para la Presentación de Materiales Audiovisuales. EMPRESA: Instituto Tecnológico de Tuxtla Gutiérrez.

ASESOR EXTERNO: _____ ASESOR INTERNO: MC. Aida Guillermina Cossio Martínez
 PERIODO DE REALIZACIÓN: ENERO-JUNIO DE 2013

ACTIVIDAD	SEMANAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Recolección de requerimientos	P														
Diseño Rápido	R														
Construir prototipo 1	P														
Evaluar y refinar requerimientos (Prototipo 1)	R														
Construir prototipo 2	P														
Evaluar y refinar requerimientos (Prototipo 2)	R														
Producto final	P														
R															
OBSERVACIONES: La residencia se concluye en la semana 20 Con el producto terminado															
ENTREGA DE REPORTES	Febrero 27 y 28														
	Abril 8 y 9														
	Mayo 8 y 9														
Docente : MC Aida Guillermina Cossio Martínez Alumno : Sánchez Alfonso José Antonio Jefe Depto. : MC Aida Guillermina Cossio Martínez															

ITTG-AC-PO-007-05 Rev.1



INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

Departamento: GESTION TEC. Y VINC
No. de Oficio: DGTyV /378
Fecha: 18/02/13

ASUNTO: PRESENTACIÓN DEL ALUMNO
Y AGRADECIMIENTO

M.C.A. José Luis Méndez Navarro
Director
Instituto Tecnológico de Tuxtla Gutiérrez
Tuxtla Gutiérrez, Chiapas
PRESENTE

El Instituto Tecnológico de Tuxtla Gutiérrez, tiene a bien presentar a sus finas atenciones al (la) alumno (a): **José Antonio Sánchez Alfonso** número de control: **09270248** carrera de: **Ing. En sistemas Computacionales** quien desea desarrollar en ese organismo el proyecto de Residencias Profesionales denominado **Pizarron Virtual multifuncional controlado a distancia por sensores de movimiento para la presentación de materiales audiovisuales** cubriendo un total de 640 horas, en un período de cuatro a seis meses, en el periodo Febrero-Junio 2013.

Es importante hacer de su conocimiento que todos los alumnos que se encuentran inscritos en esta institución cuentan con un seguro contra accidentes personales con la empresa **MetLife**, Según póliza No. **AE1489**, e inscripción en el IMSS.

Así mismo, hacemos patente nuestro sincero agradecimiento por su buena disposición y colaboración para que nuestros alumnos, aún estando en proceso de formación, desarrollen un proyecto de trabajo profesional, donde puedan aplicar el conocimiento y el trabajo en el campo de acción en el que se desenvolverán como futuros profesionistas.

Al vernos favorecidos con su participación en nuestro objetivo, sólo nos resta manifestarle la seguridad de nuestra más atenta y distinguida consideración.

ATENTAMENTE

ING. RODRIGO FERRER GONZÁLEZ
JEFE DEL DEPARTAMENTO DE GESTIÓN TECNOLÓGICA Y VINCULACIÓN

C.c.p. Archivo
C.c.p. Alumno

SECRETARÍA DE EDUCACIÓN
PÚBLICA
Instituto Tecnológico de Tuxtla Gutiérrez
Departamento de Gestión Tecnológica y Vinculación



ITTG-AC-PO-007-03

Carretera Panamericana Km. 1080, Tuxtla Gutiérrez, Chiapas. C. P. 29050, apartado Postal 599
Teléfonos: (961) 615-0380, 615-0461 Fax: (961) 615-1687
www.ittxtlagutierrez.edu.mx

Rev.1



Tuxtla Gutiérrez, Chiapas: 15/Abril/2013

DSC/031/2013

ASUNTO: CARTA DE ACEPTACION

C. C.D. JOSE ERASMO CAMERAS MOTA
JEFE DEPTO. GESTION TECNOLOGICA Y VINCULACION
PRESENTE.

Por este medio me permito informarle que el C. **José Antonio Sánchez Alfonso**, estudiante de la carrera de: **Ingeniería en Sistemas Computacionales**, con núm. de Control: **09270248**, ha sido aceptado para realizar su Residencia Profesional en este Departamento denominado: **"Pizarrón virtual multifuncional controlado a distancia por sensores de movimiento para la presentación de materiales audiovisuales"**, con fecha de inicio a partir del 18 de Febrero de 2013, cubriendo un periodo mínimo de seis meses y no mayor a dos años, haciendo un total de **640 horas**.

Sin mas por el momento quedo de Usted.

ATENTAMENTE
"Ciencia y Tecnología con Sentido Humano"

M.C. AIDA GONZALEZ MARTINEZ
JEFA DEL DEPTO. SISTEMAS Y COMPUTACION

C.p. Archivo
AGCM/fylc.

ESTADOS UNIDOS MEXICANOS
SECRETARIA DE EDUCACION
PUBLICA
INSTITUTO TECNOLÓGICO
TUXTLA GUTIERREZ
DEPARTAMENTO DE SISTEMAS
Y COMPUTACION



Carretera Panamericana Km. 1080, C.P. 29050, Apartado Postal 599
Tuxtla Gutiérrez, Chiapas: Tels. (961) 61 54285, 61 50461
www.itg.edu.mx





Subsecretaría de Educación Superior
Dirección General de Educación Superior Tecnológica
Instituto Tecnológico de Tuxtla Gutiérrez

Tuxtla Gutiérrez, Chiapas; 08/Agosto/2013

DSC/080/2013

ASUNTO: CARTA DE LIBERACION

C. C.D. JOSE ERASMO CAMERAS MOTA
JEFE DEPTO. GESTION TECNOLOGICA Y VINCULACION
PRESENTE.

Por medio de la presente me dirijo a Usted, con la finalidad de informarle que la **C. José Antonio Sánchez Alfonso**, con número de control **09270248**, de la carrera de **Ingeniería en Sistemas y Computación**, ha concluido satisfactoriamente su residencia profesional realizado en el proyecto: **"Pizarrón virtual multifuncional controlado a distancia por sensores de movimiento para la presentación de materiales audiovisuales"**, con fecha de inicio a partir del 18 de Febrero de 2013 y finalizando el 08 de Agosto de 2013, cubriendo un total de **640 horas**, bajo el asesoramiento del C. **M.C. Aida Guillermina Cossío Martínez**.

No teniendo otro particular que tratar, me despido enviándole un cordial saludo

ATENTAMENTE
"Ciencia y Tecnología con Sentido Humano"

M.C. AIDA GUILLERMINA COSSIO MARTINEZ
JEFA DEL DEPTO. SISTEMAS Y COMPUTACION

C.p. Archivo
AGCM/fyle.

ESTADOS UNIDOS MEXICANOS
SECRETARIA DE EDUCACION PUBLICA
INSTITUTO TECNOLÓGICO TUXTLA GUTIERREZ
DEPARTAMENTO DE SISTEMAS Y COMPUTACION



Carretera Panamericana Km. 1080, C.P. 29050, Apartado Postal 509
Tuxtla Gutiérrez, Chiapas. Tels. (961) 61 54285. 61 50461
www.ittg.edu.mx





INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

**SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE
SEGUIMIENTO DE PROYECTO DE RESIDENCIAS PROFESIONALES**

ALUMNO: Cabrera García Leonardo No. DE CONTROL: 09270198
 NOMBRE DEL PROYECTO: Pizarrón Virtual Multifuncional Controlado a Distancia por Sensores de Movimiento para a Presentación de Materiales Audiovisuales. EMPRESA: Instituto Tecnológico de Tuxtla Gutiérrez.

ASESOR EXTERNO: ASesor INTERNO: MC. Aida Guillermina Cossio Martínez
 PERIODO DE REALIZACIÓN: ENERO-JUNIO DE 2013

ACTIVIDAD	SEMANAS														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Recolección de requerimientos	P	R													
Diseño Rápido	P	R													
Construir prototipo 1															
Evaluar y refinar requerimientos (Prototipo 1)															
Construir prototipo 2															
Evaluar y refinar requerimientos (Prototipo 2)															
Producto final															
OBSERVACIONES: La residencia se concluye en la semana 20 Con el producto terminado															
ENTREGA DE REPORTES	Docente : MC Aida Guillermina Cossio Martínez														
	Alumno : Cabrera García Leonardo														
	Jefe Depto. : MC Aida Guillermina Cossio Martínez														

Rev.1



INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

Departamento: GESTION TEC. Y VINC
No. de Oficio: DGTyV /378
Fecha: 18/02/13

ASUNTO: PRESENTACIÓN DEL ALUMNO
Y AGRADECIMIENTO

M.C.A. José Luis Méndez Navarro
Director
Instituto Tecnológico de Tuxtla Gutiérrez
Tuxtla Gutiérrez, Chiapas
PRESENTE

El Instituto Tecnológico de Tuxtla Gutiérrez, tiene a bien presentar a sus finas atenciones al (la) alumno (a): **Leonardo Cabrera García** número de control: **09270198** carrera de: **Ing. En sistemas Computacionales** quien desea desarrollar en ese organismo el proyecto de Residencias Profesionales denominado **Pizarron Virtual multifuncional controlado a distancia por sensores de movimiento para la presentación de materiales audiovisuales** cubriendo un total de 640 horas, en un período de cuatro a seis meses, en el periodo Febrero-Junio 2013.

Es importante hacer de su conocimiento que todos los alumnos que se encuentran inscritos en esta institución cuentan con un seguro contra accidentes personales con la empresa **MetLife**, Según póliza **No. AE1489**, e inscripción en el IMSS.

Así mismo, hacemos patente nuestro sincero agradecimiento por su buena disposición y colaboración para que nuestros alumnos, aún estando en proceso de formación, desarrollen un proyecto de trabajo profesional, donde puedan aplicar el conocimiento y el trabajo en el campo de acción en el que se desenvolverán como futuros profesionistas.

Al vernos favorecidos con su participación en nuestro objetivo, sólo nos resta manifestarle la seguridad de nuestra más atenta y distinguida consideración.

ATENTAMENTE

SECRETARÍA DE EDUCACIÓN
PÚBLICA
Instituto Tecnológico de Tuxtla Gutiérrez
Departamento de Gestión Tecnológica y Vinculación

ING. RODRIGO FERRER GONZÁLEZ
JEFE DEL DEPARTAMENTO DE GESTIÓN TECNOLÓGICA Y VINCULACIÓN

C.c.p. Archivo
C.c.p. Alumno



ITTG-AC-PO-007-03

Rev.1

Carretera Panamericana Km. 1080, Tuxtla Gutiérrez, Chiapas. C. P. 29050, apartado Postal 500
Teléfonos: (961) 615-0380, 615-0461 Fax: (961) 615-1687
www.ituxtlagutierrez.edu.mx



Tuxtla Gutiérrez, Chiapas; 15/Abril/2013

DSC/030/2013

ASUNTO: CARTA DE ACEPTACION

C. C.D. JOSE ERASMO CAMERAS MOTA
JEFE DEPTO. GESTION TECNOLOGICA Y VINCULACION
PRESENTE.

Por este medio me permito informarle que el C. **Leonardo Cabrera García**, estudiante de la carrera de: **Ingeniería en Sistemas Computacionales**, con núm. de Control: **09270198**, ha sido aceptado para realizar su Residencia Profesional en este Departamento denominado: **"Pizarrón virtual multifuncional controlado a distancia por sensores de movimiento para la presentación de materiales audiovisuales"**, con fecha de inicio a partir del 18 de Febrero de 2013, cubriendo un período mínimo de seis meses y no mayor a dos años, haciendo un total de **640 horas**.

Sin mas por el momento quedo de Usted.

ATENTAMENTE
"Ciencia y Tecnología con Sentido Humano"

M.C. AIDA GUILFRERMA COSSIO MARTINEZ
JEFA DEL DEPTO. SISTEMAS Y COMPUTACION

C.p. Archivo
AGCM/fyle.



SECRETARIA DE EDUCACION
PUBLICA
INSTITUTO TECNOLÓGICO
TUXTLA GUTIERREZ
DEPARTAMENTO DE SISTEMAS
Y COMPUTACION



Carretera Panamericana Km. 1080, C.P. 29050, Apartado Postal 599
Tuxtla Gutiérrez, Chiapas; Tels. (961) 61 54285, 61 50461
www.ittg.edu.mx





Subsecretaría de Educación Superior
Dirección General de Educación Superior Tecnológica
Instituto Tecnológico de Tuxtla Gutiérrez

Tuxtla Gutiérrez, Chiapas; 08/Agosto/2013

DSC/081/2013

ASUNTO: CARTA DE LIBERACION

C. C.D. JOSE ERASMO CAMERAS MOTA
JEFE DEPTO. GESTION TECNOLÓGICA Y VINCULACION
PRESENTE.

Por medio de la presente me dirijo a Usted, con la finalidad de informarle que la **C. Leonardo Cabrera García**, con número de control **09270198**, de la carrera de **Ingeniería en Sistemas y Computación**, ha concluido satisfactoriamente su residencia profesional realizado en el proyecto: **"Pizarrón virtual multifuncional controlado a distancia por sensores de movimiento para la presentación de materiales audiovisuales"**, con fecha de inicio a partir del 18 de Febrero de 2013 y finalizando el 08 de Agosto de 2013, cubriendo un total de **640 horas**, bajo el asesoramiento del **C. M.C. Aida Guillermina Cossío Martínez**.

No teniendo otro particular que tratar, me despido enviándole un cordial saludo

ATENTAMENTE
"Ciencia y Tecnología con Sentido Humano"

M.C. AIDA GUILLERMINA COSSIO MARTINEZ
JEFA DEL DEPTO. SISTEMAS Y COMPUTACION

ESTADOS UNIDOS MEXICANOS
SECRETARIA DE EDUCACION PUBLICA
INSTITUTO TECNOLÓGICO TUXTLA GUTIERREZ
DEPARTAMENTO DE SISTEMAS Y COMPUTACION



C.p. Archivo
AGCM/fyle.



Carretera Panamericana Km. 1080, C.P. 29050, Apartado Postal 500
Tuxtla Gutiérrez, Chiapas. Tels. (961) 61 54285, 61 50461
www.itg.edu.mx





SUBSECRETARÍA DE EDUCACIÓN SUPERIOR
DIRECCIÓN GENERAL DE EDUCACIÓN SUPERIOR TECNOLÓGICA
INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ



INGENIERÍA EN SISTEMAS COMPUTACIONALES

INFORME FINAL DE RESIDENCIA PROFESIONAL

“Pizarrón Virtual Multifuncional Controlado a Distancia por Sensores de Movimiento para la Presentación de Materiales Audiovisuales”

Presentan	N° Control	Semestre
José Antonio Sánchez Alfonso	09270248	9°
Leonardo Cabrera García	09270198	9°

Asesor	Firma
M.C Aida Guillermina Cossío Martínez	

Revisores	Firma
M.C. José Alberto Morales Mancilla	
M.C. Néstor Antonio Morales Navarro	