

# Aplicando la Metodología PSP para el Desarrollo de un Sistema de Reconocimiento de Gestos

Erik Ramos Pérez, Carlos Alberto Fernández-y-Fernández, Edwin León Hernández

Instituto de Computación  
Universidad Tecnológica de la Mixteca  
Huajuapán de León, Oaxaca, México  
{erik, caff}@mixteco.utm.mx, leon.hernandez.edwin@gmail.com

**Resumen**— MUDI es un sistema reconocedor de gestos, capaz de detectar el acercamiento, rotación y cambiar de pieza arqueológica modeladas en 3D existentes en el MureH. El sistema está contextualizado para un museo interactivo en el que los usuarios puedan descubrir las características de las piezas de una forma entretenida y divertida. Para desarrollar el sistema se utilizó PSP, el cual es una metodología encaminada al proceso personal, establecido y regido por etapas bien establecidas, una de ellas es el diseño que permite tener una buena estructura del sistema traduciéndose en una fácil implementación y control de los errores de codificación, así como el manejo de los requerimientos. Para medir la interacción, efectividad y facilidad de uso de MUDI se usaron técnicas de usabilidad, obteniendo resultados favorables, específicamente el acercamiento tuvo una calificación de 4.8, rotación de 4.0 y cambiar pieza de 4.4 en una escala del 0 al 5. Los usuarios manifestaron gran aceptación y buenos comentarios puesto que con MUDI se olvidaban de las visitas tradicionales.

**Palabras clave**— usabilidad, reconocimiento de patrones, psp

## I. INTRODUCCIÓN

México es un país rico en cultura, gran parte de esta se encuentra resguardada en los 1227 museos registrados en el país [1]. Tan solo en el estado de Oaxaca se encuentran 50 museos, específicamente hablando en la ciudad de Huajuapán de León, Oaxaca se encuentra el Museo Regional de Huajuapán (MureH).

El MureH es el espacio donde se conserva, difunde y promueve la arqueología y etnografía de la cultura Mixteca.

Sin embargo, los registros de visitas a los museos son bajos, en el estado de Oaxaca asiste el 9.2% de la población total [2], no obstante en Huajuapán se tiene una asistencia del 8.5% según datos obtenidos por el patronato del MureH.

El número de visitas al año es muy bajo, este dato muestra que los oaxaqueños tienen poco interés en conocer el patrimonio cultural. Esto se puede deber a la falta de organización, financiamiento, promoción e innovación tecnológica de los museos.

En términos generales existen museos interactivos en gran parte del país, En México se encuentra el Papalote, el cual en 2008 obtuvo un poco más de 2 millones de visitantes, lo cual representa más del 5% del total de visitas a nivel nacional en todos los museos [3].

Por tanto, se puede notar que los museos interactivos son más atractivos que los que carecen de tecnología, puesto que la visita a un museo, por ejemplo, en el caso de una exposición de piezas arqueológicas, sólo se presenta la exposición y una leyenda explicando el acontecimiento, pero jamás existe interacción entre el visitante y la pieza arqueológica.

Por tanto se realizó un proyecto utilizando la tecnología que Kinect ofrece, dicha aplicación es capaz de reconocer gestos

con las manos para la manipulación de piezas arqueológicas existentes en el MureH, dichos gestos son rotar y cambiar la escala de tamaño a la misma, también se reconoce el gesto de deslizar para cambiar de pieza arqueológica.

De acuerdo a lo anterior y motivado por el desarrollo rápido de las tendencias tecnológicas en el uso de realidad aumentada, el control mediante el movimiento de las manos y la visualización e interacción en los museos, este proyecto intenta satisfacer una necesidad de innovación para el acercamiento y vinculación con la cultura a través de una interfaz de promoción de la cultura mixteca en el Museo Regional de Huajuapán de una forma agradable, estructurada y fácil de usar.

En la sección II se describe que son las interfaces naturales de usuario, la interacción humano computadora y como puede ser aplicado a una aplicación que interactúa con gestos de la mano. La sección III se describe la metodología utilizada, en la sección IV se muestra el diseño de cada uno de los gestos. La sección V indica cómo se aplicó PSP en el desarrollo de la aplicación. En la sección VI se indica el software necesario para el desarrollo de la aplicación. La sección VII muestra los resultados de las pruebas realizadas. Finalmente en la sección VIII se muestran las conclusiones.

## II. MARCO TEÓRICO

### A. Interfaz Natural de Usuario

La Interfaz Natural de Usuario (NUI por sus siglas en inglés) [4] surge motivada por el estudio e investigación de la forma en que se interactúa con el ambiente de trabajo en el contexto computacional y las interfaces naturales de usuario.

Las NUI's son aquellas en las que se interactúa con un sistema y/o aplicación, sin utilizar dispositivos de entrada como ratón, teclado, lápiz óptico [5]. En lugar de esto se utilizan las manos o las yemas de los dedos.

La NUI se enfoca en la construcción de una interfaz adaptada a nuestras capacidades, que permita concentrar la atención del usuario para realizar una tarea exitosa, en lugar de como realizarla.

La historia de la NUI se remonta en los años 80's, en la búsqueda de tratar de cambiar la forma de interactuar con las computadoras, mediante estrategias de interfaces de usuario con el mundo real [6].

Una de las primeras interfaces que utilizaron el movimiento del cuerpo y la voz para interactuar fue la aplicación desarrollada por Richard A. Bolt llamada Put-that-there [7], al igual que el Eye Tap de Steve Mann [8].

En los últimos años la utilización de la NUI se ha expandido debido a su integración con el mercado de las consolas de video juego, tales como: Kinect de Microsoft, WiiMote.

Diseñar interfaces NUI en forma adecuada significa involucrarse con el usuario de manera más directa, por tal motivo se debe hacer énfasis en el área de Interacción Humano Computadora (HCI por sus siglas en inglés).

Además para que una interfaz sea considerada natural, debe cumplir con las siguientes consideraciones [9] [10]:

- Crear una experiencia, que de la sensación de ser una extensión del cuerpo.
- Que sea natural tanto a usuarios expertos como a los usuarios nuevos.
- Auténtica al medio.
- Considerar el contexto.

### B. Interacción Humano Computadora (HCI)

La interfaz de usuario es el espacio más importante de cualquier aplicación y/o sistema debido que es ahí donde se desarrolla la interacción entre el humano y la computadora.

HCI es el estudio y practica del uso de las computadoras; vista como la capacidad que tiene la computadora para ser usada mediante el entendimiento, la creación del software y otras tecnologías que serían útiles para los usuarios, a fin de encontrar una solución efectiva a una situación que involucre el manejo y manipulación de información [11].

El usuario en esta área se vuelve el actor principal, pues se requiere entender su comportamiento durante el proceso de construcción del software para comprender los procesos y capacidades que están asociados a las tareas que desempeñan.

Para ello, se requiere desarrollar conocimientos sobre memoria, cognición, oído, tacto y habilidades motrices [12].

HCI se componen de elementos necesarios para que el diseño del sistema sea de calidad [13], por medio de la aplicación sistemática de conocimientos sobre las características y metas humanas (Proceso de desarrollo), sobre las capacidades y limitaciones de la tecnología (Computadora). Relacionándose entre sí, con aspectos sociales, organizacionales y físicos del entorno de trabajo del usuario (Uso y Contexto).

Si la interfaz es pobre o difícil de usar, el usuario simplemente ignorará el producto o la tecnología, los conceptos más importantes en HCI son [14]:

- Funcionalidad
- Facilidad de uso

Facilidad de uso es cuando un usuario utiliza las funciones del sistema fácilmente, correctamente y con claridad [15]. La funcionalidad y la facilidad de uso pueden variar de un sistema a otro. Un sistema se dice que es exitoso si hay equilibrio entre la funcionalidad y la facilidad de uso.

Para garantizar que un sistema es fácil de usar, intuitivo y amigable es necesario evaluarlo enfocándose en el contexto, apoyándose de diferentes áreas de conocimiento como [11]: psicología social, ingeniería, diseño, psicología, ciencias de la computación, inteligencia artificial, lingüística, filosofía, sociología entre otras, para construir una prueba de usabilidad.

Hablar de usabilidad es referirse a una medida de que tan “usable” es el uso de un sistema, por tanto algunos autores definen a la usabilidad como [15] [16]:

“La usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido, usado y ser atractivo para el usuario, en condiciones específicas de uso, además de ser intuitivo y fácil de usar”.

Para el desarrollo de la prueba de usabilidad es necesario definir usuarios, tareas y contexto de uso [17].

Las pruebas de usabilidad son diferentes a las pruebas de caja negra, caja blanca, o unitarias realizadas habitualmente por los desarrolladores, tampoco se trata de hacer una

demostración para los usuarios o demostrar que la aplicación y(o sistema funcional lo que se busca y se desea lograr es aprender acerca de los usuarios, como piensan y reaccionan a lo que ocurre en el sistema [18].

### C. Kinect

Kinect es una tecnología de control para la consola XBOX 360 de Microsoft que permite interactuar con un videojuego sin la necesidad de usar un control.

La tecnología empleada por este dispositivo, se basa en el entendimiento del movimiento del cuerpo, es decir, la computadora como receptora debe interpretar y entender cada uno de los movimientos del usuario antes que el videojuego pueda responder. Por tanto, el procesamiento debe ser en tiempo real para que el usuario no perciba una demora durante la interacción.

La estructura del Kinect se compone (Ver Fig. 1.) de una cámara de profundidad, una cámara RGB y una matriz de 4 micrófonos, que aíslan el ruido ambiental permitiendo utilizar el Kinect como un dispositivo capaz de recibir comandos de voz a la consola XBOX.

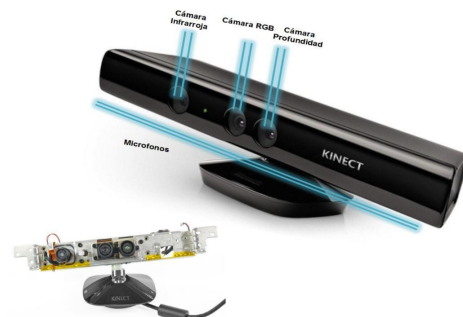


Fig. 1. Componentes de Kinect.

Lo integra también una base motorizada, controlada por un acelerómetro de 3 ejes que le permite rotar horizontalmente para ajustar el campo de visión de las cámaras para visualizar el cuerpo completo del usuario.

Las especificaciones de Kinect mencionan que el rango de especificación del Kinect a la que el usuario debe colocarse es de 0.4m a 3m. Donde el ángulo de visión del Kinect está dado por: Horizontal 57.5 grados, Vertical 43.5 grados, inclinación de -27 a 27 grados de abajo a arriba.

El sensor Kinect es capaz de capturar 30 cuadros por segundo (FPS), para la cámara de color (Ver Fig. 2.) y la de infrarrojo.

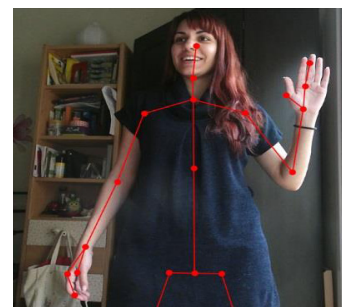


Fig. 2. Cámara de color de Kinect.

La cámara de profundidad (Ver Fig. 3.) se compone por la cámara infrarroja y el proyector infrarrojo de luz estructurada, utiliza una tecnología de luz llamada LightCoding que actúa como escáner 3D para realizar una construcción tridimensional de la escena.

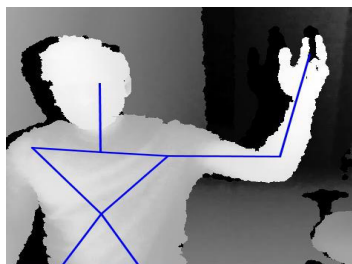


Fig. 3. Cámara de profundidad del Kinect.

Kinect se enfoca en reconocer el esqueleto humano como materia prima, para obtener dicho esqueleto es necesario procesar las imágenes de profundidad obtenidas de Kinect para detectar formas humanas e identificar partes del cuerpo del usuario en el conjunto de imágenes procesadas.

Cada parte del cuerpo es abstraída como coordenada 3D o articulación, formando un esqueleto virtual (Ver Fig. 4. ), según la biblioteca de Kinect que se utilice es el número de articulaciones detectadas en este caso se obtienen 20 articulaciones.

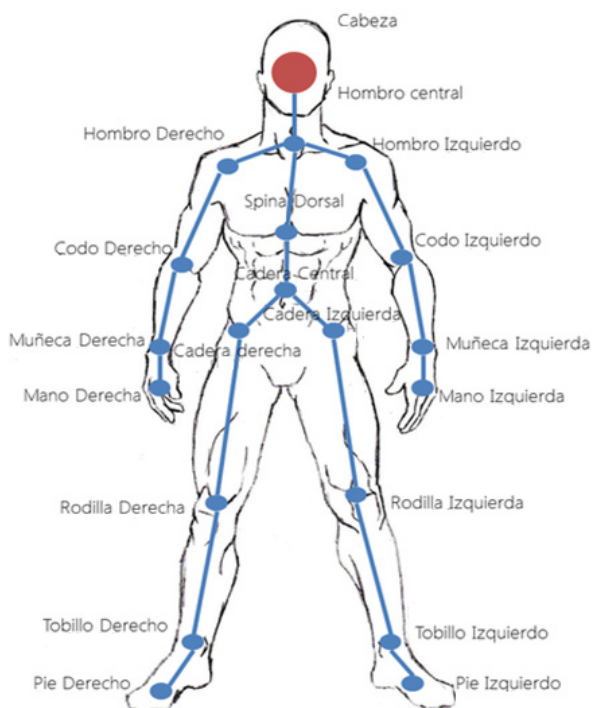


Fig. 4. Articulaciones del Esqueleto de Kinect.

#### D. Gestos de interacción

El sistema construido se basa en el reconocimiento de gestos con las manos para manipulación de un objeto virtual, específicamente un modelo tridimensional de una pieza arqueología existente en el MureH.

Por tal motivo, se define un gesto, el cual puede llegar a ser un término especializado para diferentes disciplinas de estudio. Por ejemplo: en semiótica (estudio de los signos), los gestos pueden significar palabras, imágenes, fórmulas matemáticas, mapas.

En las artes, un gesto es usado para describir aspectos expresivos de la danza.

Finalmente en el diseño de interacción, los gestos se enfocan en la manipulación de la experiencia de usuario en NUI.

Eric Hulteen y Gord Kurenbach en 1990 en su publicación llamada: "Gestures in Human Computer Interaction", lo definieron como:

"Un gesto es un movimiento del cuerpo que contiene información. Mover la mano para decir adiós es un gesto. Presionar la tecla en un teclado no es un gesto porque el movimiento del dedo para presionar la tecla no es observado y es insignificante."

Etimológicamente la palabra gesto viene del latín *gestus*, el cual era usado para referirse al lenguaje corporal de las personas, es decir, la expresión facial, movimientos de las manos y postura del cuerpo.

Para poder reconocer un gesto existen diferentes técnicas: la primera es utilizar redes neuronales, las cuales deben entrenarse para tener mayor precisión y calidad de detección. La segunda técnica es definir algorítmicamente cada gesto y la última técnica es comprobando con una serie de plantillas preestablecidas.

### III. METODOLOGÍA

Las metodologías de desarrollo de software desde el punto de vista de la ingeniería de software son vistas como un marco de trabajo que permite estructurar, planificar y controlar el proceso de desarrollo de sistemas.

Las metodologías permiten al programador llevar un control de su desarrollo, así como tenerlo organizado por etapas.

El proceso personal de software (PSP por sus siglas en inglés), es un conjunto de prácticas para la gestión del tiempo y mejora de la productividad personal de los programadores e ingenieros de software. Propuesta por Watts Humphrey en 1995, en sus inicios dirigida a estudiantes.

Las fases del ciclo PSP son: Planificación, Desarrollo y Postmortem.

Lo interesante de esta metodología es el enfoque que le da a la etapa de diseño, pues el diseño es la idea global del sistema en la cual se plasman los requerimientos, en primera instancia no se llega a tocar detalles técnicos o de bajo nivel.

El objetivo de tener un diseño, es evitar cometer errores en el desarrollo, logrando tener un mejor conocimiento que se quiere lograr con el sistema, a su vez el diseño se vuelve en una representación aprovechada para quien va implementarlo, por lo cual debe ser un documento claro, sin ambigüedades y además que descienda a un nivel de detalle de componentes.

Por tanto se deben cumplir con 4 plantillas donde se plasma el trabajo (Ver TABLA I. DISEÑOS DE PSP.):

- Plantilla de Especificación Operacional (OST por sus siglas en inglés).
- Plantilla de Especificación Funcional (FST por sus siglas en inglés).
- Plantilla de Especificación de Estados (SST por sus siglas en inglés).
- Plantilla de Especificación Lógica (LST por sus siglas en inglés).

TABLA I. DISEÑOS DE PSP.

	Interno	Externo
Estático	LST	FST
Dinámico	SST	OST

### IV. DISEÑO DEL SOFTWARE

Básicamente cuando se menciona reconocimiento de gestos con las manos, se trata de asignar un número de movimientos de la mano a determinada acción en este caso el sistema construido es capaz de rotar, agrandar/reducir y elegir una pieza arqueológica.

Samsung con su SMART TV, en la integración de comandos empleando gestos con las manos, los cuales ayudaron a definir cada uno de los gestos que se emplearon en dicha aplicación, puesto que son similares a los gestos que maneja Kinect para la interacción con los videojuegos.

Para elaborar el diseño de cada uno de los gestos que la aplicación fue capaz de reconocer existen diferentes técnicas. La primera técnica es utilizar redes neuronales, las cuales deben entrenarse para tener mejor precisión y calidad de detección. La segunda técnica es definir algorítmicamente cada gesto y la última técnica es comparando con una serie de plantillas preestablecidas.

**A. Gesto Agrandar/Reducir**

Agrandar/Reducir es hacerle zoom a una pieza arqueológica, es decir el escalamiento de la misma en los tres ejes  $(x, y, z)$ , según la longitud del gesto.

Este gesto se definió por una par de puntos obtenidos de cada una de las posiciones de las manos en el tiempo, si las manos se encuentran alineadas se toma la coordenada de la mano derecha e izquierda respectivamente  $((x_d, y_d), (x_i, y_i))$ .

Inicialmente para agrandar la pieza (Ver Fig. 5.) se deben mantener alineadas la manos en línea recta y separarlas hasta lograr el tamaño deseado.



Fig. 5. Agrandar pieza arqueológica.

Para reducir la pieza arqueológica (Ver Fig. 6.) se deben mantener las manos alineadas horizontalmente y juntarlas hasta lograr el tamaño deseado.

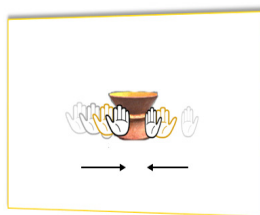


Fig. 6. Reducir pieza arqueológica.

Para obtener la proporción del escalamiento se calcula la distancia euclidiana entre la mano izquierda y derecha mediante la fórmula de la distancia euclidiana entre dos puntos, para después sacar la proporción según el tamaño de la pantalla (verFig. 7.) contemplado de 640\*480 px.

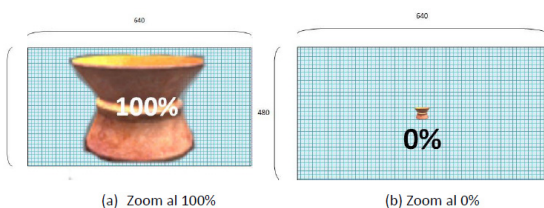


Fig. 7. Comportamiento del zoom.

**B. Gesto Cambiar**

La aplicación es capaz de detectar un swipe o deslizamiento. Este gesto básicamente, trata de mover la mano simulando una barra de desplazamiento vertical para que el usuario pueda elegir la pieza arqueológica de su preferencia y poder manipularla.

Para que este gesto pueda ser procesado se deben capturar 20 coordenadas de la mano derecha o izquierda según sea el gesto que se desea realizar.

El gesto se compone de un deslizamiento de abajo hacia arriba ejecutado por la mano derecha, moviendo está en línea recta (Ver Fig. 8. ).



Fig. 8. Deslizamiento abajo-arriba.

El segundo gesto implica solo el movimiento de la mano izquierda, está desplazándose de arriba hacia abajo, en línea recta (Ver Fig. 9. ).

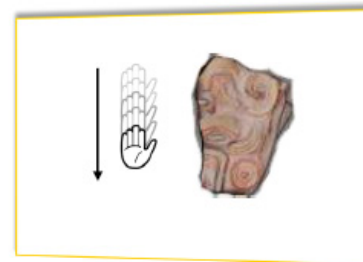


Fig. 9. Deslizamiento arriba-abajo.

En la sección VII Evaluación y Resultados, se muestra con más detalle que después de concluir las pruebas, los resultados arrojaron que este gesto no era comprendido, por tal motivo el gesto se tuvo que definir nuevamente.

Según las recomendaciones de los usuarios el gesto se definió utilizando solo una mano, específicamente la mano derecha.

El gesto debe tener una duración entre 500 y 1500 ms. Con una distancia promedio de 20cm.

Del mismo modo se tomaran 20 coordenadas, el gesto, consiste en mover la mano de izquierda a derecha o viceversa (Ver Fig. 10. ).



Fig. 10. Modificación del gesto deslizamiento.

### C. Gesto Rotación

La rotación se utiliza manipulando cada una de las coordenadas de la mano izquierda y derecha respectivamente, a un ángulo obtenido por el movimiento de las manos.

Por ahora solo se implementó la rotación sobre el eje Z. el cual se define de la siguiente manera:

Las manos deben estar alienadas de manera horizontal (Ver Fig. 11. ), una vez alineadas, estas deben moverse de izquierda a derecha o viceversa, simulando mover un volante, en todo momento las manos deben estar alineadas formando una línea recta.

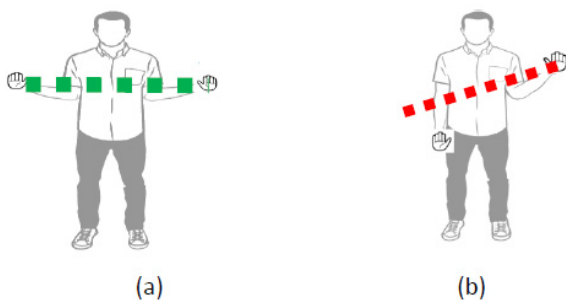


Fig. 11. a) Manos alineadas b) Manos desalineadas.

Del mismo modo al ejecutar las pruebas, se decidió cambiar el gesto puesto que era complicado de ejecutarse, por tal motivo ahora se toman las siguientes consideraciones:

- Quitar la línea verde y roja.
- Capturar 20 coordenadas de ambas manos.
- Indicar el comienzo y fin del gesto.
- Rotar  $\pm 45^\circ$  si el gesto es aceptado.

Con la modificación (Ver Fig. 12. ), para iniciar el gesto el usuario debe mantener sus manos alineadas, en ese momento se mostraran unas manos azules, las cuales indican que el gesto ha comenzado, el usuario mueve sus manos simulando mover un volante ya sea derecha a izquierda o viceversa hasta conseguir la rotación deseada.



Fig. 12. Modificación del gesto rotación.

La modificación de dicho gesto trajo consigo la implementación de una red neuronal multicapa con 40 neuronas en la capa de entrada, 5 neuronas en la capa intermedia y una única neurona en la capa de salida.

### V. METODOLOGÍA DE DESARROLLO (PSP)

El motivo central por el cual se eligió PSP como metodología de desarrollo es por la serie de procesos establecidos y la serie de recomendaciones, y su énfasis en el diseño del software.

El diseño debe plasmarse en la serie de plantillas preestablecidas, en la cual se plasma la estructura estática y el

comportamiento dinámico del software englobando tanto características internas y externas.

PSP por ser una metodología personal es ideal para el desarrollo del proyecto.

El primer paso es realizar el diseño conceptual (Ver Fig. 13. ), en el cual se plasman todos los módulos y las relaciones entre sí, así como el flujo de datos.

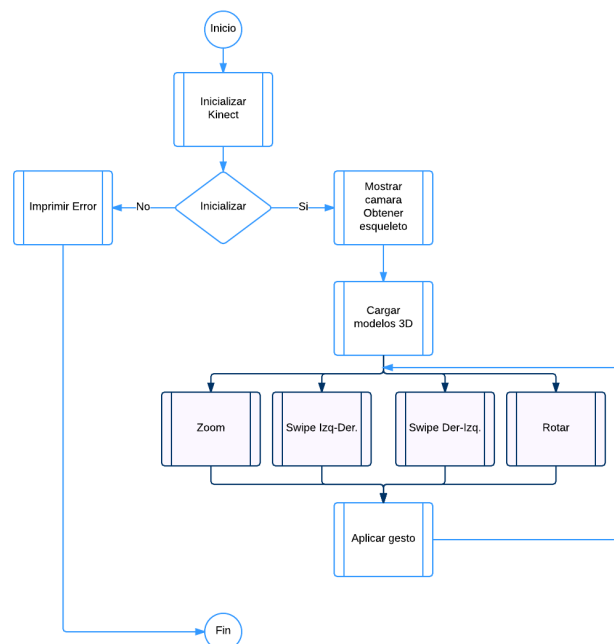


Fig. 13. Diseño conceptual del sistema.

### A. Plantilla especificación operacional (OST)

En el OST se describieron cuatro escenarios por cada gesto, dos para probar el funcionamiento correcto del sistema y dos que contemplaban casos de error.

#### 1) Acercar/Alejar

El primer escenario estará sujeto a la detección del zoom (Ver TABLA II).

TABLA II. OST DETECCIÓN DEL ZOOM.

fuente	Paso	Acción	Comentarios
Escenario #:1	Objetivo Usuario: Gesto de zoom detectado		
Objetivo del Escenario: Dadas las coordenadas de cada una de las manos el sistema detectara que es un zoom in o zoom out			
Usuario	1	Ejecuta el sistema	
Sistema	2	Obtiene las coordenadas de cada una de las manos del usuario.	Estas coordenadas se obtienen cada segundo.
Sistema	3	Verifica que las manos estén alineadas horizontalmente.	
Sistema	4	Calcula la distancia que existe entre las manos para determinar si es un zoom in o zoom out.	El zoom máximo y mínimo está determinado por la resolución de la cámara de 640x480.
Sistema		Aplica el zoom al modelo tridimensional.	

El segundo escenario se contempla cuando no se detecte el gesto de zoom como válido (Ver TABLA III).

TABLA III. OST ZOOM NO DETECTADO.

Escenario #:2		Objetivo Usuario: Gesto de zoom no detectado	
Objetivo del Escenario: Dadas las coordenadas de cada una de las manos el sistema detectara que no es zoom			
Fuente	Paso	Acción	Comentarios
Usuario	1	Ejecuta el sistema.	
Sistema	2	Obtiene las coordenadas de cada una de las manos del usuario.	Estas coordenadas se obtienen cada segundo.
Sistema	3	Verificar que las manos no estén alineadas.	

### 2) Deslizamiento

El OST para este gesto lo definimos en dos partes por tanto es importante tener para cada una su OST correspondiente.

Primero se plantea el funcionamiento normal de detección del gesto deslizamiento con la mano derecha e izquierda dado que es similar el comportamiento del sistema (Ver TABLA IV).

TABLA IV. OST SWIPE DOWNTOP/SWIPE UPTODOWN DETECTADO.

Escenario #:1		Objetivo Usuario: Gesto de SwipeDowntoUp(Deslizar Abajo-Arriba) y SwipeUptoDown(Deslizar Arriba-Abajo)	
Objetivo del Escenario: Dadas 20 coordenadas de la mano derecha detectar el SwipeDowntoUp o 20 coordenadas de la mano izquierda para detectar el SwipeUptoDown.			
Fuente	Paso	Acción	Comentarios
Usuario	1	Ejecuta el sistema.	
Sistema	2	Obtiene las coordenadas de la mano derecha o mano izquierda.	Estas coordenadas se obtienen cada segundo.
Usuario	3	Mueve la mano derecha de abajo hacia arriba de forma recta o la mano izquierda de arriba hacia debajo de forma recta.	
Sistema	4	Almacena 20 coordenadas de la mano derecha o 20 coordenadas de la mano izquierda.	Estas coordenadas serán almacenadas en un vector.
Sistema	5	Analiza y determina si las coordenadas corresponden a un SwipeDowntoUp o a un SwipeUptoDown.	
Sistema	6	Cambia la pieza arqueológica en la pantalla.	

A continuación se describe el escenario cuando el sistema no sea capaz de detectar el gesto debido que el movimiento de la mano derecha o izquierda no es el adecuado para poder definir el gesto de deslizamiento (Ver TABLA V).

TABLA V. OST DESLIZAMIENTO NO DETECTADO.

Escenario #:2		Objetivo Usuario: Gesto de SwipeDowntoUp(Deslizar Abajo-Arriba) y SwipeUptoDown(Deslizar Arriba-Abajo) no detectado	
Objetivo del Escenario: Dadas 20 coordenadas de la mano derecha o 20 coordenadas de la mano izquierda verificar que no es un gesto deslizamiento valido.			
Fuente	Paso	Acción	Comentarios
Usuario	1	Ejecuta el sistema.	
Sistema	2	Obtiene las coordenadas de la mano derecha o mano izquierda.	Estas coordenadas se obtienen cada segundo.
Usuario	3	Mueve la mano derecha de abajo hacia arriba de forma recta o la mano izquierda de arriba hacia debajo de forma recta.	
Sistema	4	Almacena 20 coordenadas de la mano derecha o 20 coordenadas de la mano izquierda.	Estas coordenadas serán almacenadas en un vector.
Sistema	5	Analiza y determina si las coordenadas corresponden a un SwipeDowntoUp o a un SwipeUptoDown.	
Sistema	6	Cambia la pieza arqueológica en la pantalla.	

Los escenarios OST de deslizamiento servirán para verificar que este gesto se implementó de forma adecuada.

### 3) Rotación

La rotación es uno de los gestos más complicados a reconocer puesto que las manos deben permanecer alineadas durante cada movimiento para lograr la rotación, el OST (Ver TABLA VI) describe el funcionamiento normal para la detección del gesto.

TABLA VI. OST ROTACIÓN DETECTADA.

Escenario #:1		Objetivo Usuario: Detección de rotación izquierda-derecha o derecha izquierda	
Objetivo del Escenario: Dadas las coordenadas de la mano derecha e izquierda verificar que es una rotación valida			
Fuente	Paso	Acción	Comentarios
Usuario	1	Ejecuta el sistema.	
Sistema	2	Obtiene las coordenadas de la mano derecha y mano izquierda las cuales deben estar alineadas.	Estas coordenadas se obtienen cada segundo.
Usuario	3	Mueve las manos simulando un volante dando vuelta a la izquierda o vuelta a la derecha.	Las manos deben estar alineadas en todo momento
Sistema	4	Registra si es vuelta derecha o izquierda y a su vez las manos estén en línea recta.	
Sistema	5	Aumenta el ángulo de rotación de la pieza arqueológica.	El ángulo aumenta o disminuye según sea el movimiento.
Sistema	6	Modifica la visualización de la pieza arqueológica.	

Cuando las manos no estén alineadas el reconocedor de gestos no hará cambio alguno a la pieza arqueológica tal como se describe en el OST (Ver TABLA VII) cuando el gesto no es válido.

TABLA VII. OST ROTACIÓN NO DETECTADA.

Escenario #:2		Objetivo Rotación no detectada Usuario:	
Objetivo del Escenario:		Dadas las coordenadas de la mano derecha e izquierda verificar que el gesto de rotar no es valido	
Fuente	Paso	Acción	Comentarios
Usuario	1	Ejecuta el sistema.	
Sistema	2	Obtiene las coordenadas de la mano derecha y mano izquierda.	Estas coordenadas se obtienen cada segundo.
Usuario	3	Mueve las manos simulando un volante dando vuelta a la izquierda o vuelta a la derecha.	El sistema muestra que las manos no están alineadas
Sistema	4	Las manos no están alienadas por tanto no es un gesto valido.	
Sistema	5	La pieza arqueológica no sufre cambio alguno.	

**B. Plantilla Especificación Funcional (FST)**

El FST documenta las clases y las relaciones, se describen los métodos y la interacción de los mismos con los resultado producidos. Se contemplaron variables a usarse y su tratamiento para cada gesto.

**1) Acercar/Alejar**

Para poder detectar si los movimientos de las manos corresponden a un zoom y además aplicarlo a la pieza arqueológica se requieren de las siguientes variables y métodos descritos en la TABLA VIII.

El procedimiento parte de la fórmula matemática de la distancia entre dos puntos, tales puntos son A(x1, y1) y B(x2, y2), en este caso el punto A representa las coordenadas de la mano derecha y el punto B de la mano izquierda.

TABLA VIII. FST ZOOM.

Clase de Prueba		ZOOM	
		Variables	
Declaración	Descripción		
Float manoizqx	Almacena el valor de la posición de la mano izquierda en el eje X		
Float manoizqy	Almacena el valor de la posición de la mano izquierda en el eje Y		
Float manoderx	Almacena el valor de la posición de la mano derecha en el eje X		
Float manodery	Almacena el valor de la posición de la mano derecha en el eje Y		
Float delta=20	Almacena el margen de error para la alineación		
Float pspinex	Almacena el valor del punto central del esqueleto humano en el eje X		
Float pspiney	Almacena el valor del punto central del esqueleto humano en el eje Y		
Float escala_inicial=0.5	Almacena el valor de la escala de la pieza arqueológica		
		Métodos	
Float distancia(float ix, float iy, float dx, float dy)	Calcula la distancia existente entre la posición de la mano derecho y la mano izquierda.		
$d = \sqrt{(ix - dx)^2 + (iy - dy)^2}$			

**2) Deslizamiento**

El deslizamiento maneja dos métodos en la TABLA IX se muestran las variables y métodos para la detección del SwipeDowntoUp:

TABLA IX. FST SWIPEDOWNTOUP DETECTADO.

Clase de Prueba		SwipeDowntoUp	
		Variables	
Declaración	Descripción		
public struct VectorG { public float X; public float Y; public float Z; public DateTime date; }	Estructura de datos que almacena las coordenadas de la mano derecha en los ejes X, Y Z, además del tiempo en el que se capturan dichas coordenadas.		
const int SwipeMininalDuration = 250	Almacena la duración mínima del gesto en milisegundos.		
const int SwipeMaximalDuration = 1500	Almacena la duración máxima del gesto en milisegundos.		
const float SwipeMinimalLength = 0.4f	Almacena la longitud mínima del gesto.		
const float SwipeMaximalHeight = 0.2f	Almacena la elevación máxima del gesto.		
List positionList	Lista que almacena los puntos de la mano derecha en la estructura de datos VectorG.		
Boolean swipe	Bandera que indica si el gesto fue aceptado.		
		Métodos	
Void Swipe1(Boolean s)	Aplica el cambio de pieza arqueológica cuando el gesto es aceptado.		

De forma similar la TABLA X muestra el FST para el reconocimiento del gesto SwipeUptoDown:

TABLA X. FST SWIPEUPTODOWN DETECTADO.

Clase de Prueba		SwipeDowntoUp	
		Variables	
Declaración	Descripción		
public struct VectorG { public float X; public float Y; public float Z; public DateTime date; }	Estructura de datos que almacena las coordenadas de la mano izquierda en los ejes X, Y Z, además del tiempo en el que se capturan dichas coordenadas.		
const int SwipeMininalDuration = 250	Almacena la duración mínima del gesto en milisegundos.		
const int SwipeMaximalDuration = 1500	Almacena la duración máxima del gesto en milisegundos.		
const float SwipeMinimalLength = 0.4f	Almacena la longitud mínima del gesto.		
const float SwipeMaximalHeight = 0.2f	Almacena la elevación máxima del gesto.		
List positionList2	Lista que almacena los puntos de la mano izquierda en la estructura de datos VectorG.		
Boolean swipe	Bandera que indica si el gesto fue aceptado.		
		Métodos	
Void Swipe2(Boolean s)	Aplica el cambio de pieza arqueológica cuando el gesto es aceptado.		

### 3) Rotación

Para poder generar el gesto de rotación son requeridos las siguientes variables y métodos identificados en la TABLA XI.

Para determinar cuándo las manos se encuentran alineadas basta con calcular los coeficientes de la ecuación general de la recta que pasa por dos puntos, en este caso el punto A, dado por las coordenadas de la mano derecha y el punto B que representa las coordenadas del punto central del esqueleto.

La ecuación de la recta permitirá determinar si las coordenadas del punto C que son las de la mano izquierda, si pasan por la recta calculada entonces se podrá determinar la alineación entre las manos.

TABLA XI. FST ROTACIÓN DETECTADA.

Clase de Prueba	Rotar
<b>Variables</b>	
Declaración	Descripción
Float manoizqx	Almacena el valor de la posición de la mano izquierda en el eje X
Float manoizqy	Almacena el valor de la posición de la mano izquierda en el eje Y
Float manoderx	Almacena el valor de la posición de la mano derecha en el eje X
Float manodery	Almacena el valor de la posición de la mano derecha en el eje Y
Float pspinxex	Almacena el valor del punto central del esqueleto humano en el eje X
Float pspineyey	Almacena el valor del punto central del esqueleto humano en el eje Y
Float rotarzinicial=0	Almacena el valor del ángulo para la rotación respecto al eje Z
Float dpr=0	Almacena la distancia de un punto a una recta
Float a	Almacena la constante a de la ecuación general de la recta
Float b	Almacena la constante b de la ecuación general de la recta
Float c	Almacena la constante c de la ecuación general de la recta
<b>Métodos</b>	
Vector calcula_recta(float mx, float my, float px, float py)	Esta función calcula los coeficientes de la forma general de la recta que pasa por dos puntos: la mano derecha y el punto central del esqueleto. $AX - BY + C = 0$ Donde A, B y C son numero reales dados por: $A = manodery - pspiney$ $B = pspinxex - manoderx$ $C = (-A * pspinxex) + (-B * pspineyey)$
Float calcula_y(float mx, float a, float b, float c)	Esta función dado un punto en x, en este caso la coordenada x de la mano derecho, calcula su correspondiente coordenada y según los coeficientes de la ecuación de la recta calculada. $y = \frac{(-A * mx) - C}{B}$
Float dpunto_recta(float mix, float miy, float a, float b, float c)	Esta función calcula la distancia de un punto a una recta dada por la siguiente ecuación: $d = \frac{ A * mix + B * miy + C }{\sqrt{A^2 + B^2}}$

El módulo de rotación fue cambiado después de los resultados obtenidos en las pruebas integrales, por tal motivo a continuación se muestra los cambios realizados (Ver TABLA XII).

TABLA XII. FST MODIFICACIÓN A ROTACIÓN.

Clase de Prueba	Rotar
<b>Métodos</b>	
Declaración	Descripción
Public void ordena();	Ordenar de menor a mayor la coordenada X de los puntos capturados de las manos, internamente algoritmo de la burbuja.
Public void depurar();	Se eliminan coordenadas repetidas

### C. Plantilla de Especificación de Estados (SST)

En el SST se plasmó en una máquina de estados, con la finalidad de seguir el flujo del sistema, con sus métodos.

#### 1) Acercar/Alejar

El zoom está compuesto por 5 estados los cuales se describen en la TABLA XIII:

TABLA XIII. SST PARA ZOOM.

Nombre Estado	Descripción	
Inicio	Define el estado inicial de la máquina de estados, obteniendo las coordenadas de las manos.	
Validar	Valida que las coordenadas de las manos estén alineadas de forma horizontal.	
Calcular_distancia	Calcula la distancia de las manos, para aplicar el zoom.	
Aplicar_zoom	Aplica el valor del zoom a la pieza arqueológica.	
Fin	Termina la detección del gesto.	
Función/Parámetro	Descripción	
Float manoderx	Almacena la coordenada x de la mano derecha.	
Float manodery	Almacena la coordenada y de la mano derecha.	
Float manoizqx	Almacena la coordenada x de la mano izquierda.	
Float manoizqy	Almacena la coordenada y de la mano izquierda.	
Float escala	Almacena el valor de escalamiento que tendrá la pieza arqueológica.	
Boolean gesto	Almacena si el gesto es válido.	
Estados/Siguientes Estados	Condición de transición	Acción
Inicio		Obtener coordenadas de cada una de las manos.
Validar	Tener las coordenadas de las manos.	
Calcular_distancia	No hay transición	
Aplicar_zoom	No hay transición	
Fin	No hay transición	
Validar		Verificar que las coordenadas Y de cada mano tengan el mismo valor $\neq$ Error.
Inicio	No hay transición.	
Calcular_distancia	Si la coordenada Y de ambas manos es igual $\neq$ Error se calcula distancia	Calcular la distancia entre la coordenada de manoderx y manodery, para obtener el valor de la escala.
Aplicar_zoom	No hay transición	
Fin	Si la coordenada Y de ambas manos es diferente $\neq$ Error	



Calcular_distancia		
Inicio	No hay transición	
validar	No hay transición	
Aplicar_zoom	Si la distancia está calculada	Asignar a escala el valor calculado de zoom correspondiente.
fin	No hay transición	
Aplicar_zoom		La pieza arqueológica toma el tamaño de la escala calculada
Inicio	No hay transición	
validar	No hay transición	
Calcular_distancia	No hay transición	
Fin	No hay transición	

## 2) Deslizamiento

El deslizamiento se compone por 6 estados de inicio a fin para completar el gesto, los cuales se muestran en la TABLA XIV:

TABLA XIV. SST PARA DESLIZAMIENTO.

Nombre Estado	Descripción	
Inicio	Define el estado inicial de la máquina de estados, obteniendo las coordenadas de las manos.	
Verificar_Swipe	Verifica que ya se tengan almacenados 20 puntos de la mano derecha o izquierda.	
Swipe1	Verifica que los puntos de la mano derecha sean válidos para SwipeDowntoUp.	
Swipe2	Verifica que los puntos de la mano izquierda sean válidos para SwipeUpDown.	
Aplicar_Swipe	Muestra la siguiente pieza arqueológica	
Fin	Termina la detección del gesto	
Función/Parámetro	Descripción	
List positionList	Lista que contiene los valores de las coordenadas X, Y Z correspondientes a la mano derecha.	
List positionList2	Lista que contiene los valores de las coordenadas X, Y Z correspondientes a la mano izquierda.	
Boolean Swipe	Almacena el valor para saber qué tipo de Swipe se detectó.	
Swipe1()	Verifica que sea un SwipeDowntoUp.	
Swipe2()	Verifica que sea un SwipeUptoDown.	
Estados/Siguientes Estados	Condición de transición	Acción
Inicio		Capturar 20 puntos de mano derecha o mano izquierda
Inicio	No hay transición	
Verificar_Swipe	Tener 20 puntos almacenados	
Swipe1	No hay transición	
Swipe2	No hay transición	
Aplicar_Swipe	No hay transición	
Fin	No hay transición	
Verificar_Swipe		Tener 20 puntos de la mano derecha o izquierda almacenados.

Inicio	No hay transición	
Verificar_Swipe	No hay transición	
Swipe1	Verificar si los 20 puntos almacenados en positionList de la mano derecha corresponden al SwipeDowntoUp	
Swipe1	Verificar si los 20 puntos almacenados en positionList de la mano derecha corresponden al SwipeDowntoUp	
Aplicar_Swipe	No hay transición	
Fin	No hay transición	
Swipe1		Validar punto a punto para saber si es swipe=1
Inicio	No hay transición	
Verificar_Swipe	No hay transición	
Swipe1	No hay transición	
Swipe2	No hay transición	
Aplicar_Swipe	Si Swipe=1	
Fin	Si swipe=0	
Swipe2		Validar punto a punto para saber si es swipe=2
Inicio	No hay transición	
Verificar_Swipe	No hay transición	
Swipe1	No hay transición	
Swipe2	No hay transición	
Aplicar_Swipe	Si swipe=2	
Fin	Si swipe=0	
Aplicar_Swipe		
Inicio	No hay transición	
Verificar_Swipe	No hay transición	
Swipe1	No hay transición	
Swipe2	No hay transición	
Aplicar_Swipe	Swipe=1 o swipe=2	Mostrar la pieza anterior o siguiente de las piezas arqueológicas disponibles
Fin	No hay transición	

## 3) Rotación

La rotación se compone por 4 estados de inicio a fin para completar el gesto, los cuales se muestran en la TABLA XV:

TABLA XV. SST PARA ROTACIÓN.

Nombre Estado	Descripción
Inicio	Define el estado de inicio, se obtienen los puntos de las manos y el de la mitad del esqueleto humano.
Validar	Valida que sea un gesto de rotación válido.
Aplicar	Aplica la rotación según el ángulo del gesto.
Fin	Termina el gesto.
Función/Parámetro	Descripción
List manoder	Almacena las coordenadas X y Y de la mano derecha.
List manoizq	Almacena las coordenadas X y Y de la mano izquierda.
List spine	Almacena las coordenadas X y Y de la mano spine.
Float dpr=0	Almacena la distancia del punto a la recta.
Vector	Esta función calcula los coeficientes de la

calcula_recta(float mx, float my, float px, float py)	forma general de la recta que pasa por dos puntos: la mano derecha y el punto central del esqueleto. $AX - BY + C = 0$ Donde A, B y C son numero reales dados por: $A = \text{mano}dery - \text{psp}fney$ $B = \text{psp}fmex - \text{mano}derx$ $C = (-A * \text{psp}fmez) + (-B * \text{psp}fmez)$	
Float calcula_y(float mx, float a, float b, float c)	Esta función dado un punto en x, en este caso la coordenada x de la mano derecho, calcula su correspondiente coordenada y según los coeficientes de la ecuación de la recta calculada. $Y = \frac{(-A * mx) - C}{B}$	
Float dpunto_recta(float mix, float miy, float a, float b, float c)	Esta función calcula la distancia de un punto a una recta dada por la siguiente ecuación: $d = \frac{ A * mix + B * miy + C }{\sqrt{A^2 + B^2}}$	
<b>Estados/Siguientes Estados</b>	<b>Condición de transición</b>	<b>Acción</b>
Inicio		Obtener coordenadas de las manos y mitad del esqueleto humano.
Inicio	No hay transición	
Validar	Puntos almacenados	
Aplicar	No hay transición	
Fin	No hay transición	
Validar		Calcular los coeficientes de la recta: calcula_recta, calcular_y, y finalmente calcular la distancia de un punto a una recta.
Inicio	No hay transición	
Validar	No hay transición	
Aplicar	Si dpr>0 y dpr<25	El gesto es aceptado.
Fin	Si dpr>0 o dpr>25	El gesto no es aceptado.
Aplicar		
Inicio	No hay transición	
Validar	No hay transición	
Aplicar	No hay transición	
Fin	Fin del gesto	Aplicar la rotación según sea el movimiento de las manos.

#### D. Plantilla de Especificación Lógica (LST)

##### 1) Acercar/Alejar

El LST se plasmó la lógica interna de cada uno de los algoritmos empleados en pseudocódigo para explicar de forma clara y concisa su funcionamiento.

El zoom se obtiene al tener alineadas la mano derecha e izquierda respecto a la coordenada Y, para poder calcular el zoom correspondiente, se toma el tamaño máximo de la pantalla, puesto que la pantalla será de 640x480 pixeles.

Por tanto el escalamiento máximo de la pieza arqueológica tendrá un valor de 2, que equivale al 100% de escalamiento que se le va a permitir al usuario, el escalamiento mínimo tendrá un valor de 0.5 equivalente al 0%.

En la TABLA XVI se muestra el LST, el cual muestra el comportamiento del gesto de zoom para poder hacer la manipulación de cada una de las piezas arqueológicas:

TABLA XVI. LST PARA ZOOM.

Parámetros	Descripción
Float manoizqx	Almacena el valor de la posición de la mano izquierda en el eje X
Float manoizqy	Almacena el valor de la posición de la mano izquierda en el eje Y
Float manoderx	Almacena el valor de la posición de la mano derecha en el eje X
Float manodery	Almacena el valor de la posición de la mano derecha en el eje Y
Float delta=20	Almacena el margen de error para la alineación
Float pspinex	Almacena el valor del punto central del esqueleto humano en el eje X
Float pspiney	Almacena el valor del punto central del esqueleto humano en el eje Y
Float escala_inicial=0.5	Almacena el valor de la escala de la pieza arqueológica
Declaración de librerías: Microsoft.Kinect, System; Definir constantes: int delta = 20;	
Obtener las coordenadas de las manos usando Kinect. void Sensor_Skeleton_FrameReady(object sender, SkeletonFrameReadyEventArgs e);	
<pre> Si (manoizqy &lt; (manodery + delta) &amp;&amp; manoizqy &gt; (manodery - delta) &amp;&amp; (manoizqy &lt; (pspine.Y + delta) &amp;&amp; manoizqy &gt; (pspine.Y - delta)    manodery &lt; (pspine.Y + delta) &amp;&amp; manodery &gt; (pspine.Y - delta))) Hacer distancia = <math>\sqrt{(\text{mano}izqx - \text{mano}derx)^2 + (\text{mano}izqy - \text{mano}dery)^2}</math> Si (distancia &gt; 25) { escalainicial = (distancia * 2) / 640; } </pre>	

Las demás plantillas lógicas se omitieron por el espacio que ocupa cada una,

Mediante el seguimiento de cada una de las plantillas se pudo tener una administración constante del avance del proyecto, así como identificar errores de programación.

La continua verificación de los diseños de PSP, adaptándolos de la mejor forma antes de la implementación, permite crear un desarrollo iterativo interno, en el cual se diseña, ejecuta y prueba.

#### VI. IMPLEMENTACIÓN DEL SOFTWARE

En esta etapa se implementó cada uno de las plantillas descritas en los diseños PSP, la etapa de codificación fue sencilla puesto que ya se tenían pseudocódigos.

El sistema se desarrolló en lenguaje C#, conjuntamente con el Kinect SDK 1.8, dicha librería nos permite obtener las

coordenadas del esqueleto de Kinect previamente obtenidas de la cámara de color y profundidad del Kinect.

Se utilizó la librería de Microsoft XNA Game Studio para la manipulación de los modelos tridimensionales y características de la propia aplicación como sonidos.

La implementación surge de los siguientes pasos:

1. Inicializar el Kinect (Ver Fig. 14. )
  - a. Verificar si se encuentra conectado tanto a la computadora como a la corriente eléctrica.
  - b. Verificar que sea un Kinect valido
2. Obtener el esqueleto de Kinect
  - a. En específico las coordenadas de las manos.
3. Cargar modelos tridimensionales
4. Cargar audios con la descripción de cada una de las piezas utilizadas.



Fig. 14. Inicializar kinect

Una vez implementados los diseños del sistema contemplando los casos de éxito y error, se obtuvo la primera versión del sistema (Ver Fig. 15. ).

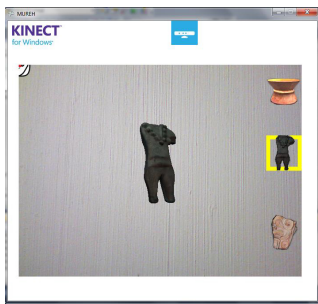


Fig. 15. Primera versión del sistema

Según los resultados de las pruebas, se cambió el menú de piezas arqueológicas y se incluyó el audio a cada uno de las piezas disponibles (Ver Fig. 16. ).

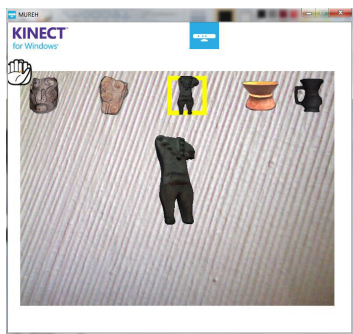


Fig. 16. Versión probada del sistema

## VII. EVALUACIÓN Y RESULTADOS

Existen muchas formas para probar un software. Desde construir el software en su totalidad y posteriormente probarlo, tratando de encontrar errores, aunque a primera vista es muy atractivo para los desarrolladores este implica más esfuerzo al corregirlo [19].

La documentación del Kinect SDK, recomienda hacer un proceso iterativo [20]: diseñar, probar y evaluar, lo cual coincide con la metodología PSP.

Se realizaron dos pruebas de usabilidad, para las cuales fue indispensable definir los usuarios, casos de prueba y el contexto de uso.

### A. Definición de usuarios

Se seleccionaron 5 usuarios para cada prueba, puesto que Nielsen en su artículo menciona que las mejores pruebas provienen de no más de 5 usuarios [15].

Se requirieron usuarios con las siguientes características, obtenidas del libro de visitas del año 2012 otorgado por el patronato del MureH:

- Estudiantes universitarios
- 18 a 25 años.

### B. Casos de prueba

El objetivo de las pruebas es por medio de la interfaz y con ayuda de Kinect poder manipular piezas arqueológicas del MureH, ajustándose a los usuarios.

Por tanto se definió al sistema como MUDI (Museo Digital Interactivo).

Los módulos a probar son:

- Acercar/Alejar pieza arqueológica.
- Cambiar pieza arqueológica.
- Rotar pieza arqueológica.

### C. Contexto

Se requirió un ambiente controlado en el cual se le pidió al usuario que pensara como si se encontrara en el siguiente escenario:

<Un día te encuentras con un amigo de la Universidad y te cuenta de un nuevo sistema que están empleando en el MureH, este sistema es utilizado para poder manipular piezas arqueológicas usando nueva tecnología, él te cuenta que es una nueva experiencia además que es novedoso>

Las pruebas se realizaron en el UsaLab (Laboratorio de Usabilidad) perteneciente a la Universidad Tecnología de la Mixteca (Ver Fig. 17. ).

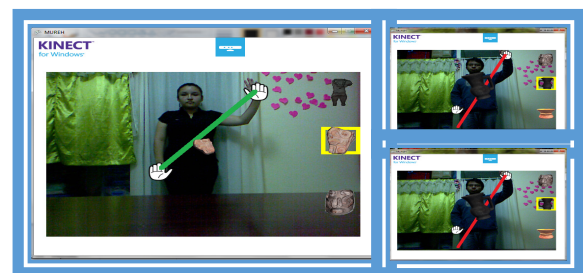


Fig. 17. Pruebas de usabilidad 1

### D. Resultados

Las métricas para cada caso de prueba son: velocidad, interfaz, funcionalidad e integración, en una escala del 0 al 10,

donde 10 representa muy sencillo y 0 muy complicado (Ver TABLA XVII).

TABLA XVII. RESULTADOS PRUEBA DE USABILIDAD 1.

Caso de prueba	Velocidad	Interfaz	funcionalidad	integración
Acercar/Alejar	10	8	9	9
Deslizar	8	8	8	8
Rotar	6	7	5	8

Los usuarios en los dos primeros casos de prueba manifestaban su aceptación, puesto que el tiempo en el que se realizó dicha tarea fue menor a dos minutos, no así para el gesto de rotar el cual les tomaba mucho tiempo realizarlo y manifestaban que era complicado y confuso.

La segunda interfaz probada con el cambio en la forma de hacer el gesto de rotación, al cual además se le añadió un video de instrucciones puesto que los usuarios no están acostumbrados a leer y omitían ese paso (Ver Fig. 18. ).



Fig. 18. Pruebas de usabilidad 2

Los resultados de esta prueba indican que los usuarios fueron capaces de completar cada uno de los casos de prueba en menos de 3 minutos ( Ver Fig. 19. ).



Fig. 19. Calificaciones obtenidas por tarea

Además se midió la usabilidad, si este era agradable y la funcionalidad (Ver Fig. 20. ).

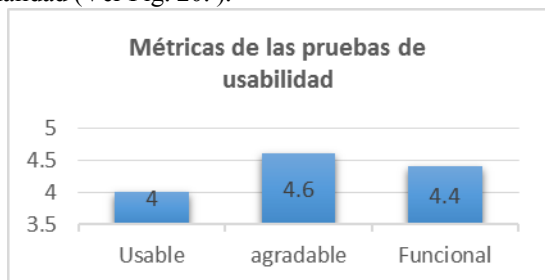


Fig. 20. Métricas pruebas de usabilidad 2

Las escalas manejadas fueron de 0 a 5, los índices muestran que la interfaz mejoró al igual que la implementación de cada uno de los gestos.

Tomando en cuenta cada una de las recomendaciones de los usuarios, se mejoró el diseño contextualizándolo (Ver Fig. 21.

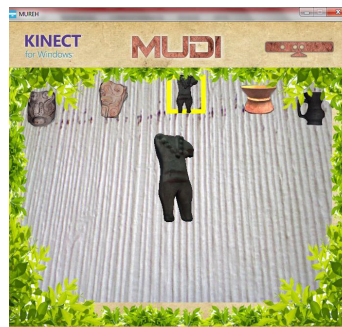


Fig. 21. MUDI versión Final

Además se muestran iconos de manos, las cuales están presentes en toda la interacción situadas en las palmas de las manos del usuario (Ver Fig. 22. ).



a) Manos Blancas b) Manos Azules

Fig. 22. Iconos de manos de MUDI final

## VIII. CONCLUSIONES

Normalmente los usuarios no leen instrucciones y si lo llegan a hacer, es muy difícil que lo recuerden por un lapso de tiempo grande, por lo tanto, se optó por introducir las instrucciones y explicaciones de las piezas arqueológicas con audio, logrando la aceptación de los usuarios al sistema.

La información de las piezas arqueológicas fue obtenida a través del Instituto Nacional de Antropología e Historia (INAH) por el arqueólogo investigador Iván Rivera Guzmán.

Después de haber realizado las pruebas de usabilidad y los resultados obtenidos por las mismas, los usuarios afirmaron que asistirían y recomendarían la visita al mismo, por lo tanto, se espera que el MureH tenga mayor afluencia, sin embargo, esto dependerá si se realiza la debida difusión y promoción.

El ciclo de uso de pruebas de usabilidad permitió ir mejorando la interfaz, hasta que finalmente fue aceptada por los usuarios, dichos usuarios por ser estudiantes universitarios son potenciales asistentes del MureH, es por eso que se consideró a los mismos como parte importante del desarrollo.

La metodología PSP permite tener un control del diseño del sistema lo cual es importante, puesto que es fácil de encontrar errores, añadir nuevos requerimientos y sobre todo tener la visión total del sistema que se construyó.

## REFERENCIAS

- [1] Conaculta, Cultura en México-museos-Sistema de Información Cultural--CONACULTA, [En línea]. Available: [http://www.sic.gob.mx/index.php?table=museo&disciplina=&estado\\_id=20&municipio\\_id=0](http://www.sic.gob.mx/index.php?table=museo&disciplina=&estado_id=20&municipio_id=0). [Último acceso: 21 Febrero 2014].
- [2] Conaculta, Infraestructura y Patrimonio, Hábitos y Prácticas Culturales., [En línea]. Available: [http://mapa.sic.gob.mx/index.php?estado\\_id=20&tema=museo](http://mapa.sic.gob.mx/index.php?estado_id=20&tema=museo). [Último acceso: 25 Febrero 2014].

- [3] Conaculta, Museos interactivos, museos para las vacaciones, [En línea]. Available: <http://www.conaculta.gob.mx/>. [Último acceso: 22 Febrero 2014].
- [4] A. Valli, Notes on natural interaction, 2005. [En línea]. Available: <http://www.idemployee.id.tue.nl/g.w.m.rauterberg/Movies/NotesOnNaturalInteraction.pdf>. [Último acceso: 12 agosto 2014].
- [5] I. SOMMERVILLE y T. J. A. D. TORRES, Ingeniería del Software, Mexico: Pearson Education, 2002.
- [6] S. Mann, Inteligent image processing, USA, Noviembre 2001.
- [7] R. A. Bolt, Put-that-there: Voice and gesture at the graphics interface, de In proceedings of the 7th annual conference on Computer Graphics and interactive techniques, SIGGRAPH, Washington, USA, 1980.
- [8] S. Mann, Steve Mann's Blog, 2012. [En línea]. Available: <http://eyetap.blogspot.ca/2012/07/physical-assault-by-mcdonalds-for.html>. [Último acceso: 21 Octubre 2014].
- [9] P. Santana, Interfaces Naturales de Usuario - La experiencia de la Universidad de Colima | SG, Abril 2014. [En línea]. Available: <http://sg.com.mx/revista/43/interfaces-naturales-usuario-la-experiencia-la-universidad-colima#.VGJl7vmG-b9>. [Último acceso: 23 Junio 2014].
- [10] P. V. R. C. V. F. Peredo V. Iván, «Manipulación de un robot Lego Mindstorms NXT a través de una NUI avanzada con reconocimiento de voz y gestos.» Revista Electronica de Divulgacion de la Investigacion, Revista Universitaria del SABES, Universidad Politécnica de Querétaro, 2014.
- [11] P. A. Booth, An Introduction to Human-Computer Interaction, New York: Psychology Press, 1989.
- [12] J. Preece, Human Computer Interaction, Addison Wesley, 1994.
- [13] T. J. R. Flor E. Narciso, La interaccion Humano-Computadora (MODIHC), de XXVII Conferencia Latinoamericana de Informática CLEI2001, 2001.
- [14] A. Mathew, Human-Computer Interaction (HCI): An overview, de Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference, Shanghai, 10-12 June 2011.
- [15] J. Nielsen, Usability engineering (Interactive Technologies), Morgan Kaufmann/Academic Press, 2014.
- [16] I. 9126-1:2001, International Standard (2001). 1. Software engineering-Product quality-Part 1: Quality model, 2005.
- [17] G. i. Eduardo Mercovich, Taller: Sitios Web de Bibliotecas Universitarias, Junio 2004. [En línea]. Available: <http://www.inspiro.com.ar/articulos/bibliotecas.html>. [Último acceso: 21 Septiembre 2014].
- [18] H. Belson, Usability, 2012.
- [19] R. S. Pressman, Ingeniería del Software Un enfoque practico Septima Edicion, Mexico: Mc Graw Hill ISBN: 978-607-15-0314-5, 2010.
- [20] M. D. Network, Kinect for Windows Programming Guide, 2014. [En línea]. Available: <http://msdn.microsoft.com/en-us/library/hh855348.aspx>. [Último acceso: 13 Agosto 2014].



**Erik Germán Ramos Pérez** es ingeniero en computación egresado de la Universidad Tecnológica de la Mixteca, actualmente se encuentra terminando la maestría en tecnologías de cómputo aplicado en la especialidad de inteligencia artificial, es profesor e investigador de tiempo completo del instituto de computación de la Universidad Tecnológica de la Mixteca, actualmente es responsable del cuerpo académico de Redes y Sistemas Distribuidos, certificado en PSP (Personal Software Process) ante el SEI.



**Carlos Alberto Fernández y Fernández** es egresado de la Facultad de Informática de la Universidad Veracruzana. Más tarde realizó la Maestría en Ciencias de la Computación en la Fundación Arturo Rosenblueth. Recibió el grado de Doctor en Ciencias de la Computación en la Universidad de Sheffield, Inglaterra. Se encuentra adscrito al Instituto de Computación de la Universidad Tecnológica de la Mixteca, desempeñándose como profesor e investigador de tiempo completo. Ha sido coordinador de la Universidad Virtual y de la Maestría en Computación con especialidad en Sistemas Distribuidos. Recientemente finalizó una estancia sabatical en el Laboratorio Nacional de Informática Avanzada. Trabaja dentro del área de Ingeniería de Software, particularmente en las líneas de modelado visual, métodos de desarrollo y especificación formal de software. Ha sido responsable del Cuerpo Académico de Ingeniería de Software en la UTM y miembro del Verification and Testing Research Group en la Universidad de Sheffield. En la actualidad es miembro de los núcleos académicos de los estudios de posgrado de Maestría y Doctorado en Tecnologías de Cómputo Aplicado y de la Maestría en Computación con especialidad en Sistemas Distribuidos.



**Edwin Iván León Hernández** es Ingeniero en Computación (Especialidad en Ing. De Software) por la Universidad Tecnológica de la Mixteca (UTM), certificado en PSP (Personal Software Process) ante el SEI. Se ha capacitado en distintos cursos de ingeniería de software y Usabilidad, así mismo como manejo de herramientas de Gestión de Proyectos de Software, ha participado en congresos de innovación en ingeniería de software. Actualmente se encuentra ejerciendo su profesión en el sector de telecomunicaciones.