



UNIVERSIDAD PRIVADA BOLIVIANA
FACULTAD DE INGENIERÍA Y ARQUITECTURA
Ingeniería de Sistemas Computacionales

PRÁCTICA INTERNA

Desarrollo de un Sistema Interactivo con Body Tracking (POSE IT!)

Estudiantes:

Ricardo Fernández

Juan Diego García

Brami Prudencio

Tutor: Ingeniero Marcelo Bernardo de la Rosa López

Cochabamba, Diciembre 2020

Índice general

| | |
|--|-----------|
| 1. Introducción | 4 |
| 1.1. Antecedentes | 4 |
| 1.2. Descripción del Problema | 4 |
| 1.3. Justificación | 5 |
| 1.4. Delimitación | 6 |
| 2. Estudio de Diagnóstico | 7 |
| 2.1. Seguimiento de la Pandemia | 7 |
| 2.2. Investigación Sobre Los Dispositivos Pioneros en el área de Body Tracking | 9 |
| 2.2.1. Kinect desarrollado por Microsoft | 9 |
| 2.2.2. PlayStation Camera desarrollado por Sony Computer Entertainment | 11 |
| 2.2.3. Ámbito Clínico | 11 |
| 2.2.4. Ámbito Pedagógico | 13 |
| 2.2.5. Ámbito De Entretenimiento y Entrenamiento | 14 |
| 3. Objetivos | 18 |
| 3.1. Objetivo General | 18 |
| 3.2. Objetivos Específicos | 18 |
| 4. Marco Teórico | 19 |
| 4.1. Body Tracking/Motion Capture | 19 |
| 4.1.1. Skeletal Tracking | 21 |
| 4.2. Base Teórica | 23 |

| | |
|---|-----------|
| 4.2.1. Cámara y Desambiguación | 24 |
| 4.2.2. Inteligencia Artificial (IA) | 25 |
| 4.2.3. Point Cloud | 29 |
| 5. Estudio de alternativas | 31 |
| 5.1. Análisis de Alternativas actualmente disponibles en el mercado | 31 |
| 5.1.1. Entretenimiento | 32 |
| 5.2. Análisis de Alternativas En Herramientas Disponibles Para el Desarrollo | 33 |
| 5.2.1. Trajes con sensores de movimiento | 33 |
| 5.2.2. TensorFlow | 34 |
| 5.2.3. Wrnch | 36 |
| 5.2.4. OpenPose | 36 |
| 5.2.5. DeepMotion | 37 |
| 5.2.6. Unity y Kinect | 37 |
| 5.2.7. Resumen de Accesibilidad de las Alternativas | 37 |
| 6. Metodología | 40 |
| 6.1. Metodología Ágil | 40 |
| 6.2. Metodología SCRUM | 41 |
| 6.2.1. Componentes de SCRUM | 41 |
| 7. Análisis y Desarrollo del Diseño del Sistema Interactivo | 48 |
| 7.1. Análisis de Requerimientos | 49 |
| 7.1.1. Requerimientos Funcionales | 50 |
| 7.1.2. Requerimientos No Funcionales | 52 |
| 7.1.3. Funcionalidad Esperada del Sistema Interactivo | 53 |
| 7.1.4. Diagrama de Flujo de Datos | 54 |
| 8. Desarrollo e Implementación del Sistema Interactivo | 55 |
| 8.1. Funcionamiento Base | 55 |
| 8.1.1. Configuración del Entorno | 55 |
| 8.1.2. Creación del Menú | 57 |

| | |
|---|-----------|
| 8.1.3. Creación de un Mapa | 58 |
| 8.1.4. Comparación entre dos Poses | 60 |
| 8.2. Normalización de Puntos Clave | 61 |
| 8.2.1. Cálculo a partir de la Normalización de un punto | 62 |
| 8.2.2. Cálculo a partir de la Normalización de Cuadro de Point Cloud | 64 |
| 8.2.3. Cálculo a partir de la Normalización de la distancia entre dos puntos | 66 |
| 8.3. Diseño De GUI | 68 |
| 8.4. Modificación y Requerimientos Adicionales | 70 |
| 8.4.1. Requerimientos Adicionales | 70 |
| 8.4.2. Diseño de GUI | 73 |
| 8.4.3. Diagrama de Flujo de Datos Modificado | 76 |
| 9. Plan de Actividades | 78 |
| 9.0.1. Cronograma | 79 |
| 10. Conclusiones | 80 |
| 11. Recomendaciones | 83 |
| 12. Anexos | 91 |
| 12.1. Lista de Abreviaturas | 91 |
| 12.2. Glosario | 92 |
| 12.3. Anexo: Control de Sprint | 96 |
| 12.4. Anexo: Product Backlog | 99 |
| 12.5. Anexo: Gráfico Burn-Up Burn-Down | 102 |
| 12.6. Anexo: Proceso de Instalación de OpenPose y Unity para el Desarrollo del Proyecto | 103 |
| 12.7. Anexo: Archivos Generados | 107 |
| 12.8. Anexo: Mensajes de Error | 108 |
| 12.9. Anexo: Cálculos de Comparación de Poses | 109 |

| | |
|---|-----|
| 12.9.1. Ejemplo de Cálculo de Normalización Basada en el Centro de la Cadera | 109 |
| 12.10 Diseño de GUI Modificados | 120 |

Índice de cuadros

| | |
|--|-----|
| 2.1. Categorías y características del Sistema PEGI | 16 |
| 2.2. Categorías y características del Sistema ESRB | 17 |
| 4.1. OpenPose Output JSON Content, Which are Given X, Y and C values | 23 |
| 5.1. Análisis de Alternativas según Accesibilidad | 38 |
| 6.1. Secciones de un Sprint Backlog | 45 |
| 6.2. Secciones de una tarea dentro del Sprint Backlog | 45 |
| 7.1. Interpretación de la solicitud inicial del cliente | 49 |
| 12.1. Product Backlog | 101 |
| 12.2. Puntos Clave T Original - Frame 80 | 112 |
| 12.3. Puntos Clave Frame 95 Curva - Frame 147 Dab | 113 |
| 12.4. Puntos Clave Frame 155 Curva Cerrada - Frame 158 Curva Cerrada . . | 114 |
| 12.5. Puntos Clave Frame 72 Desfase - Frame 188 Desfase | 115 |
| 12.6. Puntos Clave T Original - Frame 80 Desfasada | 116 |
| 12.7. Puntos Clave Frame 95 Curva - Frame 147 Dab | 117 |
| 12.8. Puntos Clave Frame 155 Curva Cerrada - Frame 158 Curva Cerrada . . | 118 |
| 12.9. Puntos Clave Frame 155 Curva Cerrada - Frame 158 Curva Cerrada . . | 119 |

Índice de figuras

| | |
|--|----|
| 2.1. Dispositivo Kinect y un Ejemplo de su Uso en un Videojuego de Volleyball | 10 |
| 2.2. Dispositivos Azure Kinect y PlayStation Camera | 11 |
| 2.3. Ejemplo de un PlayStation Eye y su uso con un videojuego “The Eye Of The Judgement” | 12 |
| 2.4. Ejemplo de Rehabilitación empleando Body Tracking | 13 |
| 2.5. Ejemplo de Clase de Educación Física empleando Body Tracking | 14 |
| 2.6. Sistema De Clasificación PEGI | 16 |
| 2.7. Sistema De Clasificación ESRB | 17 |
| 4.1. Ejemplo de Clasificación de Imagen de TensorFlow | 20 |
| 4.2. Ejemplos de Body Tracking | 21 |
| 4.3. Diversos Tipos de Seguimiento al Esqueleto seleccionados por OpenPose para el desarrollo del proyecto | 22 |
| 4.4. Ejemplo de Visión de una Cámara con sensores RGB-D | 25 |
| 4.5. Ejemplo de Neural Network | 27 |
| 4.6. Ejemplo de un Point Cloud | 29 |
| 4.7. Ejemplo de Point Set Registration | 30 |
| 5.1. Traje con Sensores de Movimiento | 33 |
| 5.2. Ejemplos de Datos en la Base de Datos COCO para la estimación de pose | 35 |
| 6.1. Ejemplo de Clasificación de imagen de TensorFlow | 41 |
| 7.1. Ilustración de Requerimientos en la GUI | 53 |

| | |
|---|-----|
| 7.2. Diagrama de Datos de Flujo | 54 |
| 8.1. Pasos a Seguir Para la Creación de Mapas | 58 |
| 8.2. Primer GUI para el Desarrollo, Cambiada para el Diseño de GUI Final | 61 |
| 8.3. Sobre posición de coordenadas de las poses al Centro | 63 |
| 8.4. Ejemplos de Encuadre de una Point Cloud | 65 |
| 8.5. Redimensionamiento de la Point Cloud A al tamaño de la Point Cloud B | 65 |
| 8.6. Menú de el Sistema Interactivo POSE IT! | 68 |
| 8.7. Menú de Opciones | 69 |
| 8.8. Menú de Listas para Mapas de POSE IT! | 69 |
| 8.9. Pasos a Seguir Para la Creación de Mapas | 71 |
| 8.10. Ilustración de Requerimientos en la GUI | 71 |
| 8.11. Diseño Final de GUI para el Modo Play | 74 |
| 8.12. Diseño Final de GUI para el Modo de Creación de Mapa | 74 |
| 8.13. Diagrama de Datos de Flujo | 77 |
| 12.1. Sprint 1 concluida | 96 |
| 12.2. Sprint Backlog del día 5 de Noviembre | 97 |
| 12.3. Tarea de la Sprint 2, Mostrar Acciones a Copiar | 98 |
| 12.4. Gráfico Burn-Up | 102 |
| 12.5. Gráfico Burn-Down | 102 |
| 12.6. Elección Personal del Estudiante para Configurar Visual Studio 2019 . | 104 |
| 12.7. Version Deseada de Unity Encerrada en Rojo | 105 |
| 12.8. Captura de Pantalla de la Configuración Inicial de OpenPose Plug-in . | 105 |
| 12.9. Prueba de Ejecución de Demo de OpenPose | 106 |
| 12.10Archivos Generados por la Conversión usando el Script con OpenPoseDemo | 107 |
| 12.11Archivos Generados por la Conversión usando el Script con OpenPoseDemo | 108 |
| 12.12Tabla de Diferencias entre Poses | 109 |
| 12.13Poses: T Original - Frame 80 - Frame 95 Curva - Frame 147 Dab . . . | 110 |
| 12.14Poses: Frame 155 Curva Cerrada - Frame 158 Curva Cerrada - Frame 72 | |
| Desfase - Frame 188 Desfase | 111 |

| | |
|---|-----|
| 12.15GUI del Menú Principal | 120 |
| 12.16GUI del Menú de Opciones | 121 |
| 12.17GUI del Menú de Opciones | 122 |
| 12.18GUI del modo de Creación de Mapas | 123 |
| 12.19GUI del modo de Creación de Mapas Sección de Llenar Datos del Mapa | 124 |
| 12.20GUI del modo de Creación de Mapas Sección de Tiempo Entre Poses Guardadas | 125 |
| 12.21GUI del modo de Juego, Fondo Azul Previo Inicio de Start | 126 |
| 12.22GUI del modo de Juego, Fondo Rojo en Error | 127 |
| 12.23GUI del modo de Juego, Fondo Verde Correcto | 128 |

Resumen Ejecutivo

El constante desarrollo de la tecnología crea oportunidades de expandir posibles resoluciones a los mismos problemas con métodos distintos, que con el tiempo mejoran o requieren menos capacidades.

Los sistemas interactivos son una gran alternativa para varias tareas que los usuarios comunes puedan resolver con su uso, en este proyecto se presenta un prototipo de un sistema interactivo con Body Tracking para que el usuario sea capaz de generar coreografías o pasos a seguir que otro usuario sea capaz de emplear para imitar las acciones que se realicen. Se la considera como una alternativa a sistemas interactivos como Just Dance o Shape up, los cuales son dirigidos al sector de entretenimiento, en baile y entrenamiento. Como característica adicional, esta alternativa es libre de la necesidad de una cámara de profundidad de campo, como un Kinect o un PlayStation Camera, los cuales son comúnmente usado para estas actividades. El prototipo esta basado en OpenPose, una API Open Source disponible para el uso con lenguaje C Sharp y Python, así mismo se emplearan conceptos de Point Cloud para la comparación de las poses y Unity, tanto para la creación de una interfaz gráfica con el principio KISS y la unión base del proyecto.

Abstract

Constant development of technology creates opportunities to expand possible solutions to the same problems with different methods, which over time improve or require fewer capabilities.

Interactive systems are a great alternative for various tasks that regular users can solve with their use, in this project a prototype of an interactive system with Body Tracking is presented so that the user is able to generate choreographies or steps to follow so that other user is able to use it to imitate the actions that are carried out. It is considered as an alternative to interactive systems such as Just Dance or Shape up, which are aimed at the entertainment, dance and training sectors. As an additional feature, this alternative is free of the need for a depth of field camera, such as a Kinect or a PlayStation Camera, which are commonly used for these activities. The prototype is based on OpenPose, an Open Source API available for use with C Sharp and Python language, as well as Point Cloud concepts to compare the poses and Unity, both for the creation of a graphical interface with the principle KISS and the base union of the project.

Introducción

1.1. Antecedentes

Durante 10 años, el desarrollo de sistemas interactivos con mira en el Body Tracking se encuentra en el mercado global, siendo una herramienta muy popular para la actividad física en casa, la rehabilitación de pacientes postrados en cama durante un largo tiempo y por supuesto, para el entretenimiento de los consumidores.

A nivel Bolivia, el desarrollo de sistemas interactivos es comúnmente dejado de lado, debido a la creencia banal de ser gravoso y poco gratificante, también de la necesidad de los habitantes de desarrollar diversos sectores más beneficiarios para la población y economía. Sin embargo, el desarrollo de sistemas interactivos con Body Tracking pueden ser aplicados a muchas áreas, potencialmente en la medicina, la educación y especialmente en el entretenimiento.

El constante desarrollo de la inteligencia artificial y Machine Learning revolucionó la forma de desarrollar Software, la liberación de información e inclusión de su educación, kits de desarrollo de Software, alternativas Open Source y librerías que permiten su uso en distintas áreas pueden ser ampliamente usadas para cumplir con las expectativas de los usuarios, buscando alternativas más accesibles.

1.2. Descripción del Problema

La brecha tecnológica y económica entre los países del mundo, la diferencia entre las culturas acostumbradas a la tecnología y los países en vías de desarrollo, sumado a

la actitud de la población de países en vías de desarrollo sobre la tecnología, ve como dispendioso la compra de un Kinect o una cámara con esas características (profundidad de campo y sensores RGB-D) ya que en su mayoría, los usuarios no la necesitan ni la emplean para múltiples tareas, solo para el entretenimiento, por lo que termina acumulando polvo.

La visión inicial del proyecto fue obstruida por este motivo, ya que se quería desarrollar un sistema interactivo empleando Kinect, sin embargo, esto no fue posible, por tanto, la perspectiva se debe centrar en la falta de herramientas comunes para el desarrollo de sistemas interactivos con Body Tracking, función que volvió comercial a Kinect y explorar la posibilidad de no necesitar una cámara especial que solo se usará para una actividad y posiblemente sea desecharlo al dejar de usarlo.

Existen varios sistemas interactivos con Body Tracking, como Shape Up o Just Dance, que van dirigidos a públicos distintos, pero si bien existe la posibilidad de un sistema interactivo que los une, no es conocido o un éxito en ventas, por tanto se requiere de una alternativa que solucione la necesidad de contar con varios programas diferentes para varias actividades distintas.

1.3. Justificación

El presente proyecto busca analizar las distintas herramientas que existen y puedan proporcionar una alternativa asequible y funcional que cumplan con los requerimientos que se esperan. Adicionalmente, busca dar un aporte a las funciones dadas por distintas alternativas para la atracción de los usuarios a esta alternativa.

En Bolivia, el uso de la inteligencia artificial puede ser la clave para mitigar el costo del desarrollo de sistemas interactivos con Body Tracking y conlleve a extender su uso y popularidad. Un primer paso puede ser el cambio de perspectivas de la población que puede permitirse estos dispositivos a través de un sistema interactivo con Body Tracking dirigido al entretenimiento, proporcionando una nueva alternativa con una visión diferente a la del común como es Just Dance o Shape It.

Existe una necesidad de promover y crear nuevas opciones para diferentes tareas

y el Body Tracking tiene múltiples posibilidades, dándolas a conocer, estudiando y produciendo resultados, se generará en el futuro atención a este prometedor campo.

1.4. Delimitación

El presente proyecto de práctica interna limitará su presentación de diseño y desarrollo a una versión beta con únicamente el producto mínimo viable, que consta de únicamente el desarrollo de las partes vitales que debe constituir el sistema interactivo para ser considerado como un desarrollo exitoso.

Debido a la naturaleza del proyecto y los recursos disponibles, no se garantiza que el producto desarrollado sea asequible para toda la población, mas bien busca atraer a las personas interesadas a ver el potencial que presenta este campo del desarrollo y decidan invertir en ello, así como busca atraer a usuarios que al emplearlo cambien su perspectiva respecto al tema.

Debido a la naturaleza de las herramientas empleadas y su constante mejora y continuo crecimiento que prosigue hoy en día, es probable que existan errores con los que no se pueda lidiar dentro de la versión beta presentada.

Estudio de Diagnóstico

2.1. Seguimiento de la Pandemia

A causa del COVID-19, a nivel global se han registrado aproximadamente 34 millones de personas contagiadas y 1 millón de muertes confirmadas hasta la fecha 1 de octubre del año 2020, cuyo desenlace causa pánico y desorden público a través del mundo. La Organización Mundial de la Salud (OMS) declaró al COVID-19 como una emergencia de salud internacional, dando recomendaciones para prevenir y controlar el contagio de la misma. En Bolivia se tiene registrados 135,311 enfermos, de los cuales 7.965 han fallecido y 31.817 casos activos restantes, además de 95.529 pacientes dados de alta, dando al COVID-19 un ratio de mortalidad de 5,89 % en Bolivia. Eventualmente se tomaron medidas de prevención, las cuales son fuertemente criticadas independientemente de la calidad de resultados, siendo la principal la declaración de una cuarentena que ha impactado servicios de transporte, comercio, entretenimiento, educativo y muchos más; por lo cual se han estado viendo sofocadas las posibilidades de salir libremente y realizar actividades que solían ser consideradas como cotidianas.

Esta cuarentena implicó que las ciudades del mundo impongan nuevas leyes de confinamiento, implementando advertencias, multas y cárcel para quienes incumplan las mismas, así mismo, se cancelaron eventos festivos, se cerraron escuelas y pospusieron clases, incluso negocios que no eran considerados necesarios para la primera necesidad de la población tuvieron que cerrar temporalmente durante largo tiempo hasta que las

personas adoptaron costumbres de higiene personal. El deporte ha sufrido un duro golpe debido al COVID-19, un ejemplo determinante son los Juegos Olímpicos, que fueron pospuestos al año 2021 y tentativamente tengan que ser cancelados en caso que no se lleven a cabo ese año, por ello las personas, especialmente los deportistas precisan de mantener sus rutinas diarias de ejercicio físico y las empresas deben planear nuevos modelos de negocio en orden de ajustarse a los cambios.

Se recomienda hacer ejercicio durante al menos 30 minutos cada día o al menos 20 minutos de exigencia vigorosa. En cuanto a niños, ancianos o enfermos crónicos, consultar con un médico es recomendado. Mantenerse en casa es efectivo para evitar la expansión de esta enfermedad, pero mantener la actividad física es un punto importante.

El ejercicio en casa puede realizarse de demasiadas formas y muy variadas, basta con buscar un ambiente que respete las medidas de prevención para el COVID-19 y sea un espacio mínimamente grande para moverse cómodamente con los brazos y piernas extendidos, aunque no sean de las mejores prácticas, es incluso suficiente un metro cuadrado para entrenar en distintos deportes o disciplinas. Se pueden exhibir múltiples ejemplos de ejercicio, por lo que es casi imposible tener una excusa, la población que no puede darse el lujo de ejercitarse, está en una situación desplorable económicamente, carece del tiempo debido a un exceso insano de trabajo, problemas de salud u otros. Se pueden realizar entrenamientos desde flexiones, sentadillas, abdominales, trotar en el mismo lugar y hacer juego de pies, así mismo se puede practicar Yoga, Tai Ji Quan, Karate, Tae Kwon Do y diversas artes marciales que poseen técnicas y prácticas tanto en ambientes grandes como pequeños y puede ser practicado en relativamente cualquier momento, solo con la presencia de voluntad y un tiempo bien administrado. Existen innumerables videos y guías de ejercicio para realizar en Internet y televisión e incluso juegos o aplicaciones para poder ejercitarse lo necesario o ir más allá.

Esto no significa que el deporte o el ejercicio deba ser obligatoriamente limitado o que se deben restringir si no se cumple con estas restricciones, sino que esta es una

medida más en contra de la expansión del COVID-19 hasta que exista una vacuna o resistencia efectiva ante ella [21].

La tutoría de entrenadores a deportistas siempre ha sido a partir de órdenes y tutoriales cuya base teórica es el enfoque cognitivo (aprender procesando la información a partir de lo que vemos y adquirimos de esa experiencia). Esto significa que la experiencia personal y el entrenamiento previo o visualización de resultados de los entrenadores se verá reflejado en su metodología de enseñanza, señalizando principalmente órdenes de ejecución de movimiento, entrenamiento mental (cuyo objetivo es desarrollar la autoestima y pensamiento/visualización positiva) y la retroalimentación para corregir errores y perfeccionar la técnica[42].

2.2. Investigación Sobre Los Dispositivos Pioneros en el área de Body Tracking

En principio, el Body Tracking con cámaras ha ido evolucionando y existen varias herramientas y entornos que se explicaran a futuro en el Estudio de Alternativas, que permiten introducir a campos de estudio y entretenimiento a experimentar con el uso de las herramientas de Body Tracking, como el médico y educativo.

2.2.1. Kinect desarrollado por Microsoft

Para exponer un ejemplo revolucionario de esta nueva rama, se menciona el dispositivo Kinect, siendo uno de los dispositivos más famosos, pero desacreditado del potencial que una vez tuvo, fue desarrollado por Microsoft por 20 años, se publicó el año 2010 y su venta se descontinuo en abril del 2016, debido a la llegada de nuevas alternativas como la realidad virtual, su trágica implementación al mercado que decanto a los usuarios y su sobre coste consumaron el hecho, a pesar de ello, desde el 2011 se lo



(a) Dispositivo Kinect



(b) Ejemplo de Uso de Kinect

Figura 2.1: Dispositivo Kinect y un Ejemplo de su Uso en un Videojuego de Volleyball

Fuente: [4] [27]

adapto para PC desde Windows 7 y permitió el desarrollo libre.

Una fortaleza del desarrollo de esta tecnología son las opciones que enriquecen las posibilidades de los usuarios en su experiencia con la tecnología, siendo el principal beneficiario la industria de los videojuegos. “Kinect ofrece una increíble cantidad de diversión a jugadores casuales, y su creativo concepto de ser libre de un control es innegablemente atractivo” denota una reseña realizada por IGN. Si bien los sistemas interactivos que emplean Body Tracking son duramente criticados tanto por críticos y usuarios, no se puede dudar de su éxito en ventas, se distinguió por ser el dispositivo electrónico de consumo más vendido en el Guiness World Records durante ese año.[20] Posteriormente, al mencionar Sistemas Interactivos con Body Tracking, se requiere de denotar la dificultad de mostrar resultados, debido a la pobreza de condiciones en muchos ambientes que los usuarios tienen, subiendo las expectativas de su desarrollo.

Azure Kinect

Es una versión moderna y reciente por parte de Microsoft, consta de un sensor de profundidad con opciones de reducción del campo de visión (FOV) para optimización de la aplicación, posee un micrófono para mejorar la captura de sonido, una cámara RGB, un acelero metro, un giroscopio y permite la sincronía de lectura de múltiples dispositivos Kinect para el desarrollo de un proyecto. Sin embargo, este equipo no está disponible comercialmente, por lo que su sola mención es suficiente.



(a) Dispositivo Azure Kinect de Microsoft

(b) Dispositivo PlayStation Camera

Figura 2.2: Dispositivos Azure Kinect y PlayStation Camera

Fuente: [7] [11]

2.2.2. PlayStation Camera desarrollado por Sony Computer Entertainment

El modelo comercial actual es la PlayStation Camera, la cual es un sensor de movimiento y un accesorio de cara para la PlayStation 4 y la PlayStation 5, fue desarrollado por Sony Computer Entertainment,.

Su modelo previo fue el PlayStation Eye, siendo lanzado al mercado desde el 2007, si bien fue lanzado al mercado antes que el Kinect, sus aplicaciones se distinguían en distintos campos y utilidades. Su visión estaba concentrada en una investigación de chat audio visual con quien tengan conexión, además del reconocimiento facial, de gestos, colores y proporcionar con ello una realidad mixta para socializar.

2.2.3. Ámbito Clínico

Realizando un enfoque principal en el papel de la rehabilitación, desde la perspectiva médica, se realizaron estudios sobre un uso alternativo a Body Tracking, distinguiendo su accesibilidad y fácil desarrollo, además que se reportan altos niveles de gozo con la interacción y ejercicio con familiares y amigos.

Se realizaron múltiples estudios, para distintas enfermedades, lesiones y situaciones medicas, se mencionaran un par de ejemplos para señalar su eficiencia. Un grupo de investigadores, estudiaron el tratamiento de deficiencia neurológica empleando consolas,



(a) Dispositivo PS Eye



(b) Ejemplo de Uso de PS Eye

Figura 2.3: Ejemplo de un PlayStation Eye y su uso con un videojuego “The Eye Of The Judgement”

Fuente: [8] [3]

tales como el Sony Playstation®2 [43] y Nintendo Wii [29] como punto de apoyo para empleo en terapias, lo que niega la objetividad negativa hacia la iniciativa del empleo de sistemas interactivos de esta índole.

La detección de movimiento y la tecnología gráfica empleada en software comerciales no se limitan al movimiento de un control convencional, permiten al usuario sentir estímulos necesarios para optimizar las habilidades motoras precisas que se buscan rehabilitar, más allá de agitarse de un lado a otro. Como teoría no cumplen los requerimientos de intervención precisa de una manera sistemática, necesarios desde el punto de vista tradicional, sin embargo, su potencial puede ser útil en el camino de la rehabilitación [32].

Siendo un estudio en el ámbito, la rehabilitación basada en Simuladores Interactivos, creando desafíos de bajo costo y desarrollados de manera Amateur, para secundar la motivación de personas hospitalizadas, particularmente aquellas que necesitan recuperarse de una lesión de trauma cerebral, médula espinal o amputaciones [32]. La retroalimentación indica que en general, se enmarcaron los desafíos como divertidos y retadores, viendo con positivismo su uso posterior en el hogar de estar disponible [33].



Figura 2.4: Ejemplo de Rehabilitación empleando Body Tracking
Fuente: [6]

2.2.4. Ámbito Pedagógico

Si bien se suspendieron las clases de este año académico a nivel primaria y secundaria en Bolivia, una obligación que consume una sustancial cantidad de tiempo es la asistencia escolar y una de las facetas importantes, la implementación de actividad física, conlleva un potencial impacto en la capacidad del estudiante de evadir una vida sedentaria reclusa y comprometerlo a actividades moderadamente intensas de práctica fuera de la labor estudiantil [23].

Se realizaron estudios sobre el potencial de los sistemas interactivos y discusiones sobre la facilidad que proporcionan en la educación y enseñanza, sin embargo, el presupuesto necesario no es necesariamente bajo para un colegio, lo cual ha privado a Bolivia de considerar estas opciones. Si este caso no impusiera un impacto negativo en la decisión de implementarlos, podría ser considerado limpiamente como oportuno.

Los sistemas interactivos utilizan una tecnología basada en acciones de movimiento, el cual puede envolverse con un importante aspecto interactivo pedagógico, el beneficio a la inteligencia corporal kinestésico. Se resalta el potencial que ofrece para mejorar la interacción y discusión saludable entre el alumnado y las habilidades de los profesores para manipular y presentar equipos y material multimedia. Como una herramienta educativa, tiene la capacidad de impulsar la motivación, promover el aprendizaje e interés de las actividades, crear un entorno agradable y competitivo en la clase[41].



Figura 2.5: Ejemplo de Clase de Educación Física empleando Body Tracking
Fuente: [25]

Sin embargo, no se puede descartar las dificultades del espacio requerido, la necesidad de calibrar los dispositivos y el cambio de metodología pedagógica para incluir el sistema interactivo, por encima de todo, la persuasión a las unidades estudiantiles y a los profesores de modificar la manera tradicional de enseñar educación física, ya que los estudios aún no son claros en su totalidad de que tan tangibles son los resultados en comparación al tradicional [23].

2.2.5. Ámbito De Entretenimiento y Entrenamiento

Un campo popular es aquel que requiere de una forma lúdica de realizar actividad física y ejercicio, ya que, al igual que en el ámbito pedagógico y clínico, su objetivo es incitar al usuario a realizar ejercicio de una u otra forma, sin embargo, esta vez el enfoque es distinto, buscando desgastar el cuerpo de los usuarios para poder relajarse o cansarse.

Este ámbito tiene una característica importante y es su amplio margen de usuarios potenciales, los cuales, han sido mermados debido a que hasta el momento, han tenido la necesidad de adquirir dispositivos como Kinect o PlayStation Camera para acceder a estos juegos interactivos.

Clasificación de Edades Para el Uso de Sistemas Interactivos de Entretenimiento y Entrenamiento

Los sistemas interactivos están diseñados para distintos públicos según las temáticas que presenta el producto, en el caso de entretenimiento y entrenamiento, la población dirigida no tiene restricciones de edad, ya que títulos como Just Dance o Shape Up, ambas de la empresa Francesa Ubisoft, que clasifican sus títulos como apto para todo público en la clasificación por edades, exponiendo los sistemas de clasificación Europeo y Estadounidense, se encuentran el sistema PEGI y ESRB respectivamente.

En Latinoamericano, el año 2014, Colombia ofrece en la ley No. 1554, un nuevo sistema para clasificación de Sistemas Interactivos que se comercialicen de una u otra forma, como el alquiler o venta.

- Circulación Abierta, la clasificación para el público general. En principio de temática deportiva, educativa, informativa y fantástico.
- Circulación Restringida, clasificada para un público mayor a 18 años. Su contenido referencia actos discriminatorios, conflicto, consumo de sustancias controladas, consumo de bebidas alcohólicas, desnudez, sexo o sexualidad, lenguaje soez, derramamiento de sangre, muerte, lesiones humanas o apuestas por dinero o propiedades.

Sistema PEGI

En el sistema Pan European Game Information (PEGI), cuyo objetivo es clasificar el contenido de sistemas interactivos y software de entretenimiento.

El sistema PEGI fue desarrollado por la Federación Europea de Software Interactivo (ISFE) y se puso en práctica desde el 9 de abril de 2003, empleado por 25 países. El sistema cuenta con la evaluación del Instituto Holandés para la Clasificación de Contenido Audiovisual (NICAM), como historial de trabajo, NICAM fue responsable también del sistema holandés para la clasificación de edades de películas.

Todas las categorías contienen un modo de juego en línea y de compras dentro del juego. En la figura 2.6 se encuentran las categorías del sistema PEGI, los cuales son:



Figura 2.6: Sistema De Clasificación PEGI

Fuente: [2]

| Categoría | Características |
|------------------------------|---|
| Apto para todo público | Violencia en un grado moderado |
| Apto para mayores de 7 años | Violencia moderada, uso de armas y terror moderado. |
| Apto para mayores de 12 años | Violencia moderada, uso de armas, lesiones, muerte censurada, lenguaje soez, terror moderado, referencias vagas a sexo y apuestas moderadas y vigilancia. |
| Apto para mayores de 16 años | Violencia, uso de armas, lesiones, muerte, derramamiento de sangre, lenguaje soez, terror, temas sexuales, sustancias controladas y apuestas. |
| Apto para mayores de 18 años | Además de todo lo anterior, se añade la discriminación y erotismo, solo para personas con criterio o conscientes. |

Cuadro 2.1: Categorías y características del Sistema PEGI

Fuente: Elaboración propia basado en [2]

Sistema ESRB

Con el mismo objetivo que el sistema PEGI, el sistema Entertainment Software Rating Board (ESRB), es la versión estadounidense para clasificación de contenido de sistemas interactivos, para asignar categorías que dependen del contenido y temática. Se estableció en 1994 por la Entertainment Software Association (ESA), la cual es una asociación comercial de la industria de sistemas interactivos en Estados Unidos. Fundada inicialmente con el nombre de Interactive Digital Software Association (IDSA), renombrado el 16 de julio de 2003. Es conocido mundialmente por ser empleado por empresas que son miembros de ESA, entre ellos Capcom, Square Enix, Ubisoft, Bandai Namco Entertainment, Nintendo. La ESA también prepara la exposición anual de videojuegos Electronic Entertainment Expo (E3) en California, Los Ángeles.

En la figura 2.7 se encuentran las categorías del sistema ESRB, los cuales son:



Figura 2.7: Sistema De Clasificación ESRB

Fuente: [1]

| Categoría | Características |
|--------------------|---|
| Early Childhood | Contenido apto para niños, específicamente desarrollados para público infantil. Se retiró la categoría en 2018. |
| Everyone | Una selección de títulos diseñados para todo público, minimizando violencia fantástica y empleo de insultos leves. Aplica al deporte. |
| Everyone 10 and up | Permite violencia leve o fantástica, sangre animada, insultos leves. Apto para mayores de 10 años. |
| Teen | Contiene derramamiento de sangre, temas sugerentes, violencia moderada o humor negro, en varios casos, juegos de azar o apuestas. |
| Mature 17+ | Contienen violencia, sangre, horror, temas sexuales e insultos. Apto para mayores de 17 años. |
| Adults Only 18+ | Escenas prolongadas de violencia o temas sexuales, apuestas y azar con dinero real o ficticio y sangre o desnudez frecuentes. Usualmente contiene temas de controversia o exageraciones que requieren de restricciones. |
| Rating Pending | Se desconoce su categoría, debido a su reciente lanzamiento, usualmente depende de la intensidad de la temática presentada por el sistema interactivo. |

Cuadro 2.2: Categorías y características del Sistema ESRB

Fuente: Elaboración propia basado en [1]

Objetivos

3.1. Objetivo General

Desarrollar un sistema interactivo con Body Tracking, empleando una cámara común para seguimiento de posiciones, entrenamiento, técnicas de arte marcial y baile.

3.2. Objetivos Específicos

1. Utilizar Software existentes para el seguimiento corporal utilizando una cámara.
2. Implementar una función para registrar mapas de movimiento propios del usuario.
3. Proveer una alternativa factible al mercado de sistemas interactivos con Body Tracking tales como Just Dance.

Marco Teórico

El proyecto es un sistema interactivo que busca proporcionar al usuario independientemente de sus intereses, el cumplimiento de sus expectativas, siendo un desarrollo a conciencia, pero sin tomarlo demasiado en serio, busca también romper el estereotipo que imponen los proyectos de esta modalidad, que si bien, ofrecen un producto de relativa buena calidad y mantenimiento, son o bien productos comerciales como es el caso de Just Dance o productos privatizados como son los desarrollados para la medicina.

El desarrollo del proyecto empleará los recursos disponibles y al alcance de cualquier desarrollador, por tanto, no empleara equipo especializado, esta aclaración es necesaria, ya que la calidad del producto final puede ser muy variable al de proyectos similares y es una característica más por la que sobresaldría este proyecto, ya que reduciría el presupuesto necesario para el consumidor.

A continuación, se debe mencionar factores importantes sobre la implementación del proyecto, como las propiedades del Body Tracking, la verdadera forma que tiene esta herramienta que tiene para ofrecer al software, por que no sobra mencionar las distinciones entre una cámara normal y una de profundidad.

4.1. Body Tracking/Motion Capture

El seguimiento corporal del cuerpo en tiempo real, normalmente conocido como Body Tracking o Motion Capture hace referencia al seguimiento del cuerpo humano a través de una cámara, existen dos acercamientos a este estudio, el enfoque de ajuste del

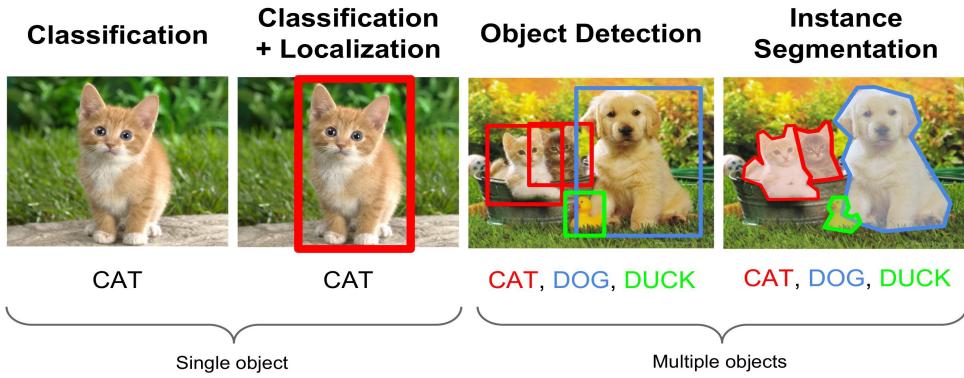


Figura 4.1: Ejemplo de Clasificación de Imagen de TensorFlow
Fuente: [46]

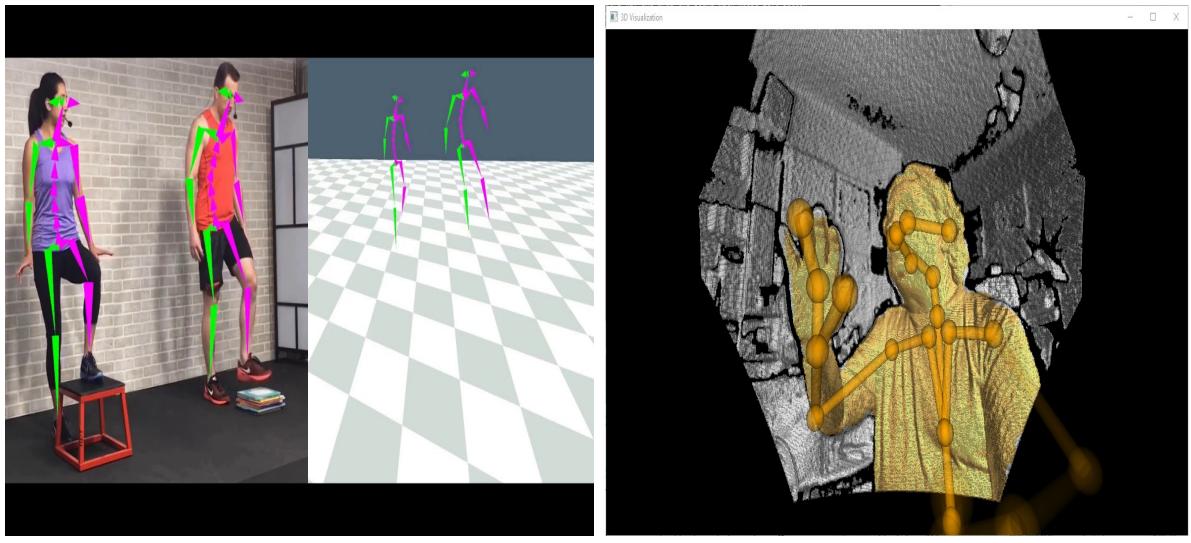
modelo y el enfoque de aprendizaje. El enfoque de ajuste del modelo involucra ajustar el modelo formulado según imágenes previas cargadas, estimando parámetros de puntos especificados de la imagen, sin embargo, es demasiado dependiente de extremos locales y la inicialización adecuada, lo cual lo vuelve inservible en ambientes nuevos. Este modelo comparte una similitud al método Monte Carlo basados en cadenas de Markov [47].

En cambio, el enfoque de aprendizaje requiere de grandes cantidades de imágenes con notas y especificaciones del esqueleto, incrementando las dimensiones de espacio y memoria requerido. Un ejemplo de este tipo de reconocimiento es la herramienta PoseNet, derivado de TensorFlow, empleado para identificar objetos a partir de una Dataset que clasifique los elementos que se buscan identificar con silueta y nombre.

Finalmente, la iteración del punto más cercano [28], el cual dispone de un enfoque de inicialización del esqueleto a través de fotogramas subsecuentes, clasificándolos con vértices y segmentos en un modelo 3D para este propósito.

La estimación de las poses humanas representan una problemática compleja de solucionar, el cual tuvo un largo trayecto hasta salir a la luz. La complejidad se centra en las múltiples limitantes, como las mascotas, los objetos del área, las personas de alrededor, la variación del escenario, los parámetros del cuerpo (el tamaño, longitud de las extremidades, torso y otras partes del cuerpo) y la iluminación.

Empleando las herramientas de seguimiento del Esqueleto, sensores de profundidad y sensores RGB, a medida que el tiempo corre, la necesidad de herramientas como los sensores va volviéndose obsoleta con el nacer de herramientas como PoseNet y Open-



(a) Ejemplo 1

(b) Ejemplo 2

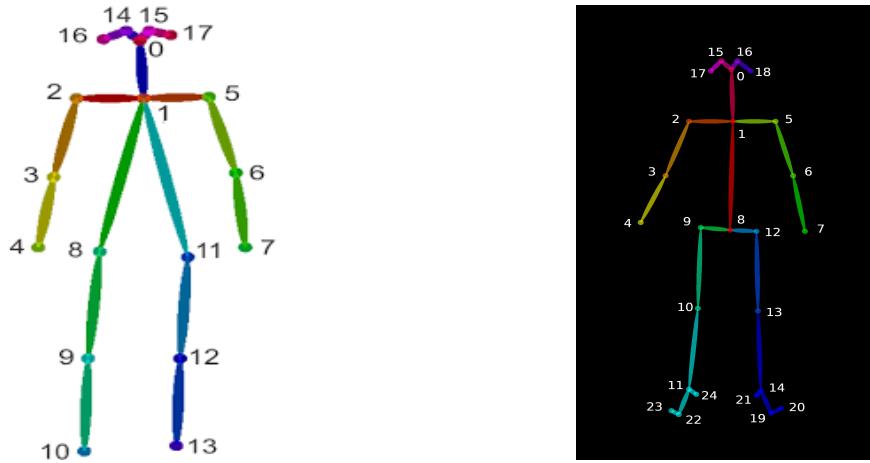
Figura 4.2: Ejemplos de Body Tracking
Fuente: [10] [12]

Pose, que con el apoyo del Hardware mínimo necesario, son capaces de proporcionar la misma información con una calidad apenas inferior de seguimiento corporal.

4.1.1. Skeletal Tracking

Este fue una innovación brindada por el controlador Kinect al mercado comercial, Su demanda fue elevada en la época y hasta el día de hoy sigue siendo empleado, el reconocimiento de una persona desde cualquier angulo o distancia, tomando en cuenta su figura, tamaño, color, cabello, ropa y el ambiente. Se emplea el escaneo de la imagen para reconocer puntos importantes del cuerpo que representan al cuerpo, tales como la cabeza, cuello, hombros, brazos, piernas y otros 10 a 20 puntos dependiendo la herramienta de reconocimiento que se emplee.

La herramienta seleccionada para el proyecto es OpenPose empleando este modelo de Formato de salida, para los datos del seguimiento del esqueleto es emplear la flag write_json para guardar la información dentro un JSON, el cual contiene un objeto de esqueleto de persona dentro, que contiene un vector pose_keypoints_2d con puntos 2D para localizar y detectar cada punto de unión $x_1, y_1, c_1, x_2, y_2, c_2, \dots$. Las coordenadas x,y en el rango de $[0,1], [-1,1]$.



(a) Reconocimiento OpenPose Tipo COCO (b) Reconocimiento OpenPose Tipo BODY_25

Figura 4.3: Diversos Tipos de Seguimiento al Esqueleto seleccionados por OpenPose para el desarrollo del proyecto

Fuente: OpenPose Models Image Output [17] [16]

Además de la existencia de los vectores `face_keypoints_2d`, `hand_left_keypoints_2d` y `hand_right_keypoints_2d`, análogos a `pose_keypoints_2d`, los cuales debido a la masiva carga de memoria que requieren y la falta de ella (mínimo 4GB de Memoria dedicada estimada, contando solo con 2GB en el equipo proporcionado) serán ignorados, pero empleando `pose_keypoints_2d`.

Como dato, los vectores análogos `body_keypoints_3d`, `face_keypoints_3d`, `hand_left_keypoints_2d` y `hand_right_keypoints_2d` (si se habilitase), en vez de $x_1, y_1, c_1, x_2, y_2, c_2, \dots$, el formato sería $x_1, y_1, z_1, c_1, x_2, y_2, z_2, c_2, \dots$, donde c sería 1 o 0 dependiendo si la reconstrucción 3D es exitosa.

Se empleará el modelo BODY_25 de Caffe, para mostrar el esqueleto, que consiste en un modelo formal para mostrar 25 puntos del cuerpo dándole forma, cada punto con su conexión en los siguientes puntos clave:

Estimación de Poses

La estimación de poses como su nombre indica, tiene la labor de estimar la pose de una persona a partir de modelos guardados en una base de datos, que proporcionan en ubicaciones específicas puntos clave que en su conjunto forman el esqueleto, además los datos sobre sus puntos se guardan como es explicado en el Análisis de Alternativas de

| Posición | Punto Clave | Posición | Punto Clave |
|----------|------------------|----------|------------------------|
| 0 | Nariz | 13 | Rodilla Izquierda |
| 1 | Cuello | 14 | Tobillo Izquierdo |
| 2 | Hombro Derecho | 15 | Ojo Derecho |
| 3 | Codo Derecho | 16 | Ojo Izquierdo |
| 4 | Muñeca Derecha | 17 | Oreja Derecha |
| 5 | Hombro Izquierdo | 18 | Oreja |
| 6 | Codo Izquierdo | 19 | Dedo Gordo Izquierdo |
| 7 | Muñeca Izquierda | 20 | Dedo Meñique Izquierdo |
| 8 | Cadera Central | 21 | Talón Izquierdo |
| 9 | Cadera Derecha | 22 | Dedo Gordo Derecho |
| 10 | Rodilla Derecha | 23 | Dedo Meñique Derecho |
| 11 | Tobillo Derecho | 24 | Talón Derecho |
| 12 | Cadera Izquierda | 25 | Escenario |

Cuadro 4.1: OpenPose Output JSON Content, Which are Given X, Y and C values

Fuente:[16]

PoseNet.

Se podría mencionar que su diferencia parte también de las palabras que lo conforman, Body Tracking/Capture Motion hace referencia a seguir el movimiento de una persona empleando Estimación de Pose, mientras que la Estimación de Poses hace referencia a marcar los puntos clave de una persona y construir un esqueleto de una persona a partir de esos puntos[40].

4.2. Base Teórica

Depth of Field (DOF)

La profundidad de campo es el producto del deseo de producir una imagen más clara del objeto o escenario deseado. Normalmente, una cámara debe enfocar su lente para tener mayor precisión, por tanto, se adjuntaron formas para ajustar la lente. Actualmente, se desarrollaron sistemas automáticos para el ajuste de la lente, los cuales incluyen la medición de la distancia al objetivo, empleando esta para ajustar la lente y obtener mayor claridad [37].

Sensores RGB-D

Es el término dado al conjunto de sensores RGB y Depth (de profundidad). Los sensores de RGB y profundidad son una adición a los dispositivos como Kinect, PlayStation Camera y otros.

El sensor de profundidad cumple la función de proyectar luz infrarroja y un sensor infrarrojo.

Divide su funcionalidad en dos pasos, la proyección de rayos de luz infrarroja, los cuales rebotan en el ambiente en el que se encuentra y posteriormente son detectados por el sensor de infrarrojo. Esta información es decodificada y enviada al usuario para poder emplearla como se lo estime.

RBG es un modelo de colores basado en la obtención de un color empleando una mezcla entre otros colores, todas ellas con los colores primarios, que son el rojo, verde y azul. El sensor RBG analiza el escenario que tiene en frente, determinando la cantidad de luz que requiere la producción de una imagen de calidad, optimiza y ajusta la velocidad de obturación, apertura y sensibilidad de captura de luz. Se encarga de analizar cada pixel del Frame y crear una imagen meticulosa.

En el desarrollo del proyecto, no se empleará un sensor de profundidad, ya que una cámara Web no lo tiene equipado.

La funcionalidad del sensor RGB-D es combinar la información colectada por el sensor infrarrojo y el sensor RGB, proporcionan datos de color y profundidad a cada pixel en cada Frame que analiza. Dentro de la memoria del Kinect, existen patrones de referencia, que poseen colores y profundidades ya conocidas, que facilitan la estimación del escenario que lo necesite[35].

4.2.1. Cámara y Desambiguación

Una cámara se define como un dispositivo que permite el registro y reproducción de imágenes. A través del tiempo, se desarrollaron muchos tipos de cámara, por tanto es necesario determinar el tipo de cámara del cual se dispone para la elaboración del proyecto, siendo el selecto la cámara Web. Sin embargo, en la mayoría de proyectos

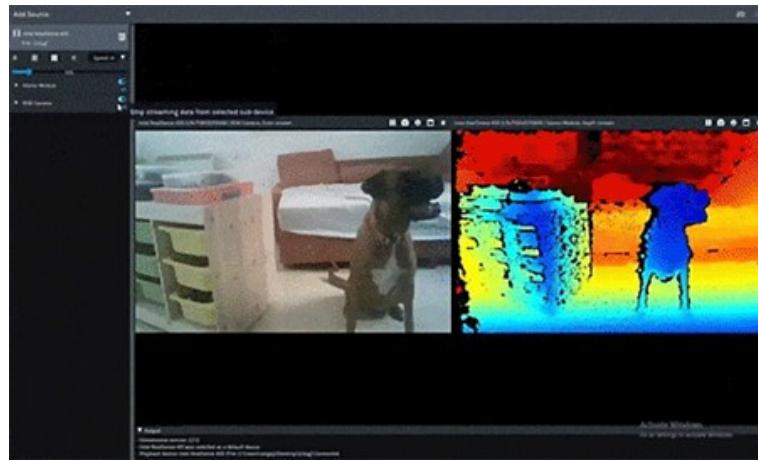


Figura 4.4: Ejemplo de Visión de una Cámara con sensores RGB-D
Fuente: [9]

relacionados al Body Tracking, se menciona el término Depth of Field (DOF), mención a una cámara de profundidad de campo.

La cámara web es un modelo pequeño de una cámara digital conectada a una computadora, normalmente una portátil, es muy común tener una en una laptop o notebook. Tiene la capacidad de capturar imágenes y transmitirlas a través de Internet. Un punto importante es que pueden ser empleadas para el desarrollo de aplicaciones y programas de cierta índole, entre ellas, este proyecto.

4.2.2. Inteligencia Artificial (IA)

Una faceta importante del proyecto, es la estimación de poses, elemento producto de la Inteligencia Artificial.

La inteligencia artificial (IA) es una rama de la ciencia de computación, que se define como “Un programa que en un mundo arbitrario, no se las arregle peor que un humano” [24]. Esta definición se somete a muchos factores, para empezar requiere de asumir tres factores importantes. El primer factor es que cualquier dispositivo de cálculo puede ser modelado por un programa. El segundo factor es que el IA es un dispositivo que permite ingresar información de afuera y dar una respuesta de acuerdo a ello. El tercer factor es asumir que la IA, al estar en contacto con el mundo este recibe información que da una respuesta interactiva con el mundo. Se asume que el mundo en que

se encuentre será influenciado por la IA[24]. El mundo se considera como el ambiente con el que la IA tiene interacción, aquel que proporciona los valores que la IA procesará y aquel que reaccionará a las acciones y/o valores de salida de la IA, en un proceso de emisión y recepción continuo.

Machine Learning (ML)

El aprendizaje de la máquina o Machine Learning es una rama de la ciencia de computación dirigida en algoritmos Computacionales diseñada para emular la inteligencia humana, a través del aprendizaje del mundo que lo rodea.

Los modelos ML ganan popularidad debido a su capacidad de proporcionar resultados y conocimientos de un tipo específico en escenarios similares a los que ha entrenado, pero nunca antes visto. Los modelos ML provee información relevante sobre los datos relacionados a lo que se requiere, traduciendo los datos de entrada, en datos que se desean obtener. Por ejemplo, un médico, para diagnosticar a un paciente con un síntoma tan común como fiebre, requerirá de un extenso conocimiento en enfermedades que tienen este síntoma y basándose en todo su conocimiento teórico, indagara en información que pueda relacionar con la fiebre, especificando otros problemas o falta de ellas que tenga y llegará a una conclusión sobre que enfermedad padece [38].

Una versión simplificada es decir que Machine Learning emplea algoritmos para clasificar y filtrar la información recibida, aprender de ella, guardarla y tomar una decisión basándose en lo aprendido.

Neural Network

Neural Network inicialmente es inspirado por la compleja red de neuronas del cuerpo humano, el cual transmite información, ordenes a través de impulsos eléctricos, donde miles de millones de conexiones existen, con ese fin[49].

Formalmente, el modelo de Neural Network consiste en un set de unidades Computacionales y un set de conexiones de una vía unidas. A través del ingreso de datos, las unidades Computacionales lo examinan y computan, generando una activación como

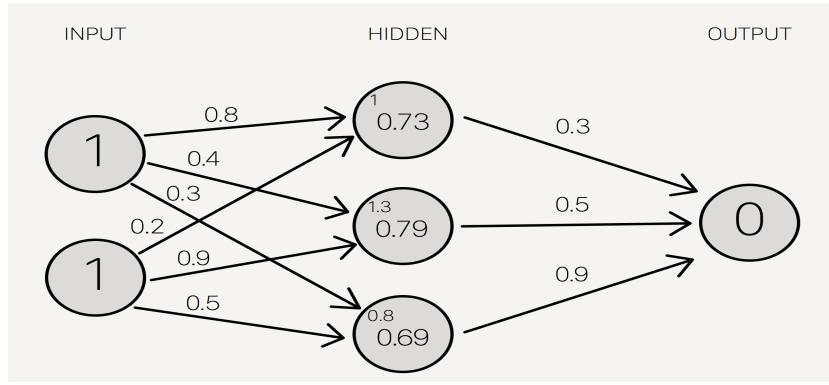


Figura 4.5: Ejemplo de Neural Network
Fuente:[16]

dato de salida. La activación atraviesa la conexión que tiene la unidad Computacional con otra unidad y procede a examinarlo y computarlo para generar otra activación. Este proceso se repite, con el fin de aumentar el llamado peso que determina la influencia que se obtiene al examinar y computar el valor recibido, dependiendo el objetivo, mientras mayor o menor termine siendo el peso en las conexiones a medida que se computan las activaciones en las unidades Computacionales, mayor efecto tendrá en el dato de salida. Un ejemplo de ello se observa en la figura 4.5 [26], donde los círculos representan unidades Computacionales, las flechas las conexiones, los valores los pesos, ingreso de datos es el 1 y el dato de salida es 0. El 0.73, 0.79 y 0.69 es el resultado de una Activación después del primer computo.

Esta herramienta es empleada por OpenPose y PoseNet para el entrenamiento de las herramientas de estimación de pose, que lo utilizan para que con los Dataset de miles de imágenes que sirven como imágenes de ingreso y sus etiquetas, puedan ser analizadas por extensas capas de unidades Computacionales que evaluaran cada imagen y proporcionaran pesos, que a mayor peso, mayor precisión en la estimación de pose tendrá.

Deep Learning (DL)

El aprendizaje profundo o Deep Learning es una subrama de Machine Learning que se basa en el uso de Neural Network, empleando numerosas capas y nodos para el aprendizaje, que en conjunto a toda la base de datos previa da lugar las decisiones en

las que se basa.

El Deep Learning utiliza múltiples capas de decisión, siendo el dato de ingreso interpretado por el logaritmo en diferentes niveles, cada uno repercute en el anterior y produce un aprendizaje basado en intento y error para aprender, corrigiendo los datos de salida de cada capa para llegar a una respuesta cada vez más aproximada a la deseada. Una diferencia importante entre Machine Learning y Deep Learning, es que, Machine Learning en general mejora con el tiempo, pero requiere de corrección de vez en cuando, si se desvía de los resultados que se buscan. En cambio, Deep Learning determina si su propia predicción es correcta haciendo uso de su red neuronal, no requiere de intervención.

En el desarrollo del proyecto se hace mención al modelo CAFFE y COCO, los cuales son parte fundamental en el proyecto, ya que estas son los modelos de Deep Learning empleados para el Body Tracking.

Modelo DL CAFFE

El modelo CAFFE es un Framework opensource trabajado con las librerías C++ y CUDA para Deep Learning, tiene interfaces en Command Line, Python y MathLab, incluye soporte en problemas con el modelo, tiene referencias, demos y herramientas para poder usarlo, además de la característica de usarse con CPU y GPU [30] [30].

Además de poseer una comunidad abierta a ampliar el desarrollo y trabajar con la herramienta, ventaja característica del código open source. El modelo CAFFE ofrece el poder definir modelos, optimizar opciones para el desarrollo y pre-entrenar para poder realizar el aprendizaje.

En la sección del proyecto, CAFFE realizó sus estudios para poder guardar Dataset de miles de cuerpos humanos, otorgándole puntos clave para formar el esqueleto. La herramienta OpenPose selecta hace uso de la CMU Panoptic Dataset, que junto a sus miles de miles de esqueletos 3D permite estimar la pose humana, produciendo el Modelo BODY_25 visto previamente en la figura 4.3a.

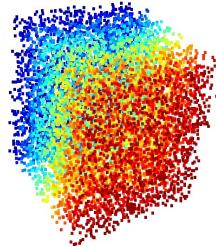


Figura 4.6: Ejemplo de un Point Cloud
Fuente: [14]

Modelo DL COCO

EL modelo COCO es un Dataset reciente del año 2014, desarrollado para el reconocimiento de objetos en el contexto requerido, a través de la obtención de imágenes complejas que contengan la información requerida en su estado natural. En su mayoría, posee objetos reconocibles por niños pequeños, tiene 2.5 millones de etiquetas en más de 300000 imágenes.

En el contexto, el modelo COCO posee miles de imágenes relacionadas a la identificación del cuerpo humano y la construcción de su esqueleto, marcando con etiquetas sus puntos claves del cuerpo y la confiabilidad de esos datos[34].

4.2.3. Point Cloud

Una nube de puntos o en inglés como es comúnmente conocido, es una colección de datos definido por un sistema de coordenadas, este normalmente es empleado para almacenar la forma de un objeto o sistema físico. Se utilizan para crear modelos 2D o 3D y es empleado por distintos campos, como la arquitectura, manufactura, videojuegos, realidad virtual y otros [45].

En el proyecto, este es relevante para el almacenamiento del Skeletical Tracking que

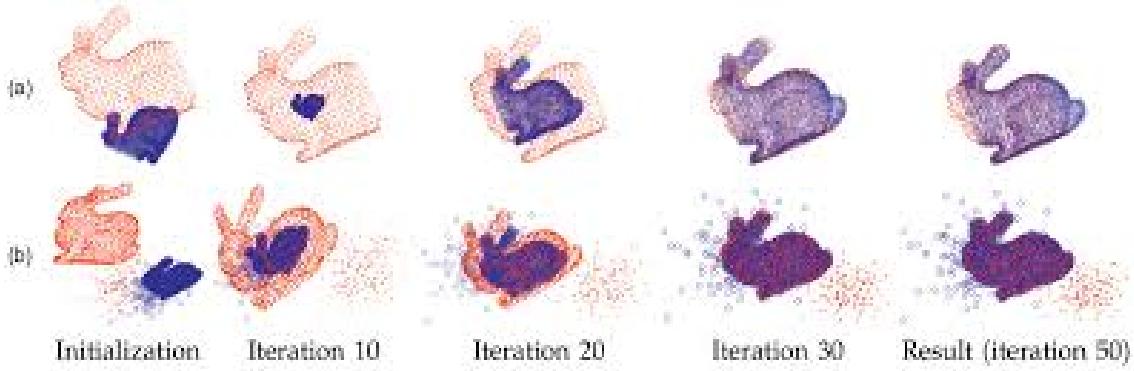


Figura 4.7: Ejemplo de Point Set Registration

Fuente: [44]

se requerirá en el proyecto, donde se guardan los 25 puntos del modelo CAFFE en una Point Cloud.

Point Cloud Registration

Registro de nubes de puntos o Point Cloud Registration es un componente clave para el proyecto, consiste en asignar la correspondencia entre dos sets de puntos y así recuperar la forma de un mapa transformando las dimensiones de una Point Cloud a las dimensiones del otra. Normalmente es empleado cuando se tienen distintos objetos del mismo tipo pero en distintos tamaños y se desea ajustar su escala a una sola.

Suelen existir múltiples factores que considerar al realizar esta comparación, por lo que es una labor con un margen de error que se define en base al proyecto que se desea realizar y existen varios algoritmos que generalizan la comparación, sin embargo, para el proyecto, se tomará solo el concepto básico y a partir de los datos de la herramienta que se seleccione, se usará un calculo propio [39].

Estudio de alternativas

5.1. Análisis de Alternativas actualmente disponibles en el mercado

En la actualidad, en la extensión de la investigación, existen productos comerciales y proyectos privados que se asemejan a las características ofrecidas por el proyecto, son en su mayoría de índole privatizada especialmente en el campo de la medicina, donde solo se hace mención a sus resultados, sin embargo, ninguna resalta por su accesibilidad gratuita, distinguiendo el proyecto del factor común. Siendo las aplicaciones y programas desarrollados con un enfoque principal en entretenimiento, como es el caso de videojuegos y medicina, en analizadores médicos y programas de rehabilitación.

Analizadores Médicos

En la rama de la medicina, la información es más privatizada o proporciona un menor alcance al público, solo reporta resultados y posibles cambios y amplitudes afirmando la falta de pruebas para dar mayor información, por lo que profundizar en ello no otorgara profundidad al proyecto, no obstante, se ha encontrado que se utiliza la tecnología planeada para controlar la pose del usuario, ya sea en pruebas médicas para hallar anomalías físicas en la posición del cuerpo al realizar diversas actividades y en entrenamiento físico para mantener una posición estable y evitar dañarse a uno mismo.

5.1.1. Entretenimiento

Cuando se referencia a un sistema de estimación de pose del cuerpo, se nombran varios títulos de videojuegos de baile, siendo las primeras y más importantes sagas, las desarrolladas por Ubisoft como Just Dance o Dance Experience, enfocadas en realizar coreografías pre-diseñadas para canciones populares en la época del lanzamiento de sus entregas; a pesar de que en Just Dance han existido niveles con temática de artes marciales, estas no contaban con la intención de ser un entrenamiento educativo, sino un baile con el propósito de entretener utilizando movimientos basados del arte marcial. Además de las mencionadas, existe otro género en los videojuegos enfocado al entrenamiento físico, como Shape Up de Ubisoft y Wii Fit de Nintendo, estas enfocadas más a un entrenamiento físico casual como es el caso de Shape Up, donde se incentiva al jugador a realizar ejercicios anaeróbicos de alta intensidad como son flexiones, abdominales o sentadillas y Wii Fit enfocado a rutinas de Yoga, equilibrio y aeróbicos; demostrando las diferentes aplicaciones y posibilidades de desarrollo y resultados.

Just Dance

Just dance es una serie de sistemas interactivos con Body Tracking, es un producto pionero en el tema desarrollado por Ubisoft París que debutó en octubre del 2010. La serie de Just Dance cuenta con un total de 30 títulos con un amplio rango de categorías de edades para llegar a la audiencia, es ya una serie que libera un nuevo título al menos una vez al año, demostrando ser un éxito comercial. Gran parte de la inspiración del proyecto actual es este sistema interactivo, que emplea al máximo el controlador Kinect y el Body Tracking con comandos de voz para ofrecer un uso casual y entretenido, sin embargo, la carencia del equipo y la frustración de no ser capaces de aprovecharlo, se desvían en cambio a la posibilidad de crear una alternativa, el cual se espera que con el equipamiento provisto para el proyecto, carente de una cámara de profundidad (elemento sobresaliente en el controlador Kinect), permita el desarrollo adecuado de las características de Body Tracking.

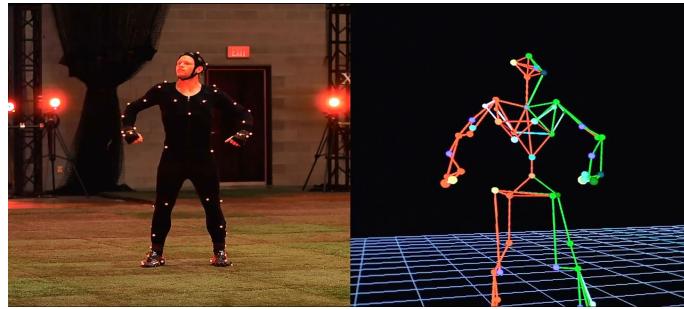


Figura 5.1: Traje con Sensores de Movimiento
Fuente: [5]

5.2. Análisis de Alternativas En Herramientas Disponibles Para el Desarrollo

5.2.1. Trajes con sensores de movimiento

La industria actual de cine, sistemas interactivos, deportes e incluso la medicina cuentan con una inversión que llega a niveles millonarios en recrear animaciones del cuerpo humano, empleando las herramientas externas que faciliten la lectura de movimiento, su uso es bastante extendido y llega a crear obras de gran calidad, sin embargo, en su mayoría esta tecnología no está disponible para el público en general debido a su elevado costo, y ya que no es un campo prioritario de investigación en la Universidad Privada Boliviana, por tanto, esta alternativa queda descartada.

Los trajes cuentan con 19 sensores distribuidos alrededor del cuerpo y recogen movimientos de un actor para transmitirlos en tiempo real. Una de sus ventajas es la reducción de costos en las producciones para motion capture, además de no requerir un set de estudio demasiado restrictivo para su implementación. Existen además, varios modelos del traje proporcionados por empresas como Xsens, Holosuit, Teslasuit y otros.

Creación de Software propio para Body Tracking

Uno de los objetivos implícitos del desarrollo de este proyecto es el siguiente “El objetivo no es volver a crear la rueda, sino, crear algo nuevo utilizando esa rueda”, palabras del supervisor del proyecto, por tanto, a pesar de ser un posible acercamiento

a la solución, queda completamente descartada.

5.2.2. TensorFlow

Tensorflow es una herramienta open source para Machine Learning provista de soporte por Microsoft, la comunidad la ha empleado extensamente en proyectos y estudios de múltiples campos de la inteligencia artificial, de Body Tracking con el uso de PoseNet, derivado de TensorFlow, cuenta con soporte y una documentación clara, siendo además sus requisitos recomendados para su uso relativamente bajos.

TensorFlow se basa en redes neuronales para Deep Learning, es una herramienta interfaz que implementa algoritmos para Machine Learning, puede ser ejecutada con una variedad de sistemas, desde celulares hasta Tablets, e incluso sistemas a gran escala. Tiene un amplio rango de algoritmos de entrenamiento, incluyendo entrenamiento e modelos de inferencia para Neural Network Models e investiga ramas de múltiples ramas, incluyendo recolección de información, proceso de entendimiento de un lenguaje natural, geografía, descubrimiento de drogas, ciencias Computacionales y otros [13].

PoseNet

PoseNet es una herramienta desarrollado por Google Creative Lab basada en TensorFlow que permiten demostrar una estimación a tiempo real de estimación de poses (Body Tracking) en tiempo real. Esta herramienta puede ser empleada tanto para una persona a la vez como para varias personas a la vez dependiendo del algoritmo que se emplee, la diferencia es en la velocidad y simpleza en su función [31].

Debido a que el proyecto será para un uso singular, se mencionara su posibilidad a través del algoritmo (del cual se desconoce su funcionamiento interno, ya que no es explícitamente necesario para el proyecto, es parte de una herramienta que se empleará) de Body Tracking para una persona.

Emplea una imagen RGB que alimenta a la red neuronal y emplean un decodificador de poses, designando valores de confianza, posiciona puntos clave y valores de confianza para los puntos clave para el aprendizaje de la red neuronal en la lectura de imágenes con las cuales aprendió a estimar las posiciones en tiempo real[40].

En cuanto los términos previos, son propios de la estimación de pose, por tanto se los debe explicar y mostrar visualmente para su entendimiento, los cuales se denotan en la figura 5.2:

- Pose: Retorna un objeto que contiene una lista de puntos clave y valores de confidencia para cada persona detectada.
- Valor de confianza de pose: Determina la confianza que tiene la estimación de la pose, en un rango de 0 a 1 y puede ser usado en caso que las partes del cuerpo bloqueadas por el cuerpo mismo o elementos externos represente un obstáculo para su lectura exacta.
- Punto Clave: Son literalmente un grupo de puntos, es la parte que se estima del cuerpo humano para formar el esqueleto, como se observa claramente en la figura 4.3b. En la base de datos de modelo COCO, existen 17 puntos de lectura.
- Valor de Confianza de un punto clave: Determina en un rango de 0 al 1 la confianza que se tiene de la precisión del punto determinado, bajo el mismo concepto de la pose, en caso que existan obstáculos.
- La posición del punto clave: Las coordenadas (x,y) de la imagen en las que se encuentran los puntos claves.

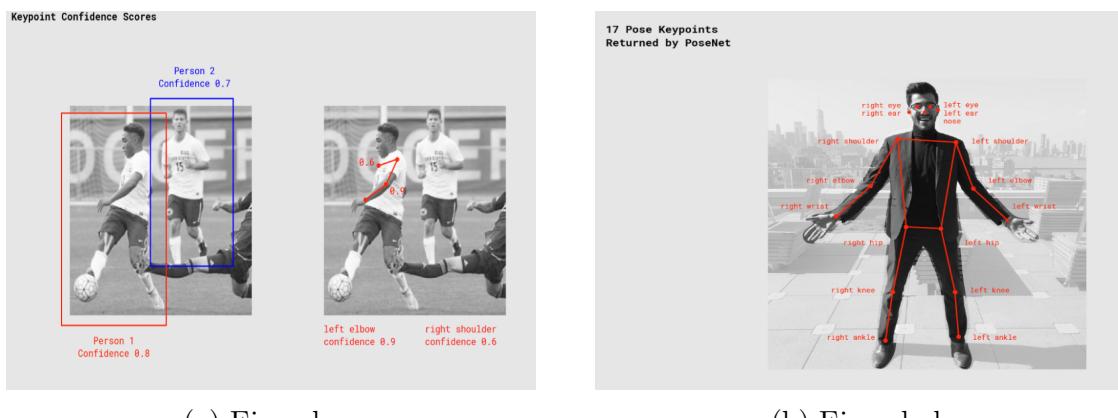


Figura 5.2: Ejemplos de Datos en la Base de Datos COCO para la estimación de pose
Fuente: [31] [40]

Si bien esta es una herramienta con soporte constante por parte de Google y una documentación clara y amplia, la herramienta requiere de características altas por parte de los equipos empleados para usarlo, su instalación y uso no es nada sencillo y es imposible cubrir todos sus requisitos en la medida de lo posible con el corto tiempo dado por la materia, concluyendo que se requerirá de buscar otras opciones para el proyecto.

5.2.3. Wrnch

<https://wrnch.ai/>) Wrnch es una herramienta de calidad para el desarrollo de aplicaciones con Body Tracking, posee un gran potencial para el desarrollo del proyecto, contando con el esqueleto que se forma al seguir los movimientos de la persona, además cuenta con una opción de múltiples cámaras en distintos ángulos, capaz de seguir el movimiento de los dedos al mismo tiempo que el cuerpo completo casi en tiempo real. Como debilidad, la dependencia del Hardware y sus elevados requisitos para emplear al máximo esta herramienta con el equipamiento disponible, limitó sus posibilidades y uso en el proyecto.

5.2.4. OpenPose

OpenPose es una herramienta open source desarrollada y mantenido principalmente por Gines Hidalgo, junto a su equipo de 6 personas y el apoyo de CMW Panoptic Studio Dataset. Es una herramienta principalmente programada en C++, empleando Cuda, Cmake y Shell, parte de ello para su instalación por parte de terceros que deseen desarrollar Software a partir de esta herramienta. Incluye APIs para desarrollo en Python y C++, posee un plug-in para Unity desarrollado en el 2018, que a pesar de estar desactualizado, posee los mínimos requerimientos necesarios para el desarrollo del proyecto[50][17][48][16]. Es una herramienta similar a PoseNet en cuanto a los requerimientos del proyecto, es constantemente actualizado y permite incluso el reconocimiento de puntos clave del rostro y las manos, que si bien, merece ser mencionado, no se empleara a lo largo del proyecto.

Esta es la herramienta seleccionada para el desarrollo del proyecto, por tanto, fue previamente explicado y mencionado en el Marco Teórico, en la sección de Capture Motion y Skeletal Tracking, para más información, favor de revisarlo nuevamente.

5.2.5. DeepMotion

DeepMotion es una herramienta de Body Tracking proporcionada por la compañía DeepMotion, la cual, empleando Machine Learning, proveen de una solución para crear animaciones de modelos 3D con recursos mínimos en requerimientos de Hardware y experiencia requerida. Si bien, esta herramienta es rápidamente descartada debido a su alto coste y la lentitud de respuesta por parte del equipo en proporcionar el conocimiento para su adquisición, el soporte que se espera es amplio.

5.2.6. Unity y Kinect

El Kinect SDK provee a los desarrolladores de herramientas para el reconocimiento de voz empleando el uso de los sensores del dispositivo Kinect con el que funciona, además de una lectura de profundidad e infrarrojo para facilitar la Motion capture. Al emplear este recurso y utilizar Unity para poder desarrollar tanto los modelos necesarios para el proyecto como la interfaz de la aplicación, se podría llegar a desarrollar el proyecto sin contratiempos e incluso con más funciones.

Sin embargo, hoy en día, el dispositivo Kinect fue descontinuado y las consolas a las que se conectan ya no incluyen el adaptador para el dispositivo. Además, se recuerda que la falta del Kinect y la búsqueda de una solución alternativa a su uso obligan a rechazar la consideración de emplear esta posible solución.

5.2.7. Resumen de Accesibilidad de las Alternativas

Durante la exploración de las alternativas, se encontró que la mayoría estaban sujetas a un pago previo a su uso, privatizando su contenido, sus herramientas y únicamente mostrando un precio o incluso ni siquiera estando disponible a la venta. En cambio,

| Herramientas | Disponibilidad | Documentación disponible | Detalles |
|-----------------------------------|----------------|--------------------------|--|
| Trajes con sensores de Movimiento | Pago + Envió | Privatizada | Demandante, esperar por el traje, no cumple con el deseo del cliente y es elegido por empresas profesionales, requiere de formación previa implementación. |
| PoseNet | Gratis | Disponible | Fácil aprendizaje, tedioso, inexacto y extenso labor por realizar. |
| Wrnch | Pago | Privatizada | Desconocido, se intento contactar con la empresa, no se obtuvo una respuesta en una semana |
| OpenPose | Gratis | Disponible | Difícil aprendizaje, fácil implementación, no es tan exacto |
| DeepMotion | Pago | Privatizado | Desconocido, sin embargo, es posiblemente una herramienta excepcional según se observa en la página oficial |
| Unity y Kinect | Pago + Envío | Disponible | Si bien se necesita el equipo Kinect, es de fácil aprendizaje y disponibilidad |

Cuadro 5.1: Análisis de Alternativas según Accesibilidad

Fuente:[16]

existen otras alternativas que son de fácil acceso y conocimiento, pero carecen del atractivo en el uso.

Para evitar la indagación de la disponibilidad, tomando en cuenta números sobre el coste, ya que tienden a variar o necesitar equipos externos como un traje, sensores, pantallas verdes y otros, o variar según el tipo de empresa o investigación, se los clasificara en 3 tipos, Gratis, Desconocido y Pago (+ Envío).

En cuanto a la documentación disponible, se considerarán dos tipos Disponible y Privatizada, ya que en ciertos casos, se obtiene la documentación posterior al pago de los servicios, incluso con cursos disponibles cuyo acceso requiere del flujo de efectivo.

Finalmente en detalles, se mencionan aspectos de la disponibilidad, documentación y la visión subjetiva de los estudiantes sobre el desarrollo con la herramienta.

Finalmente **se decidió emplear OpenPose**, debido a que es una herramienta para el desarrollo de Software libre para fines de investigación, si bien tiene poca documentación, es intuitivo y ofrece más que suficiente para el desarrollo. Cabe resaltar que parte del problema inicial es la falta de equipo y recursos para la investigación e

implementación de las alternativas, por tanto, se optó por la opción más prometedora.

Metodología

En el desarrollo de proyecto, al tratarse del desarrollo de una aplicación de Software, sumado al corto tiempo asignado para la actividad, se implementará una metodología ágil, las cuales están diseñadas para concentrarse más en la elaboración del producto que en la elaboración del documento.

6.1. Metodología Ágil

Día a día, el Software se vuelve más complejo y el manejo de herramientas ha evolucionado, volviéndose más sencillas y productivas de manejar, por ello, se requiere de una metodología de desarrollo de Software que se adapte a estos cambios. La metodología de desarrollo de Software es un conjunto de “mejores prácticas” que si no se lleva a cabo, resulta inútil[19]. Los recursos humanos son considerados como el factor más importante del proyecto de Software.

La metodología ágil de desarrollo de Software se fundamenta en los 4 principios básicos de la figura 6.1, que permite al equipo priorizar el Software que funciona a la documentación, rezagarse del uso de un diagrama de flujo de datos hasta el final del desarrollo y una respuesta a cambios más versátil que en una planificación estricta.

De las metodologías ágiles conocidas, la más apropiada es la metodología SCRUM, debido a su seguimiento y comunicación, forzando un reporte diario y a un avance continuo con resultados productivos, con la posibilidad de definir metas en un tiempo específico, maximizando la efectividad de los recursos humanos.

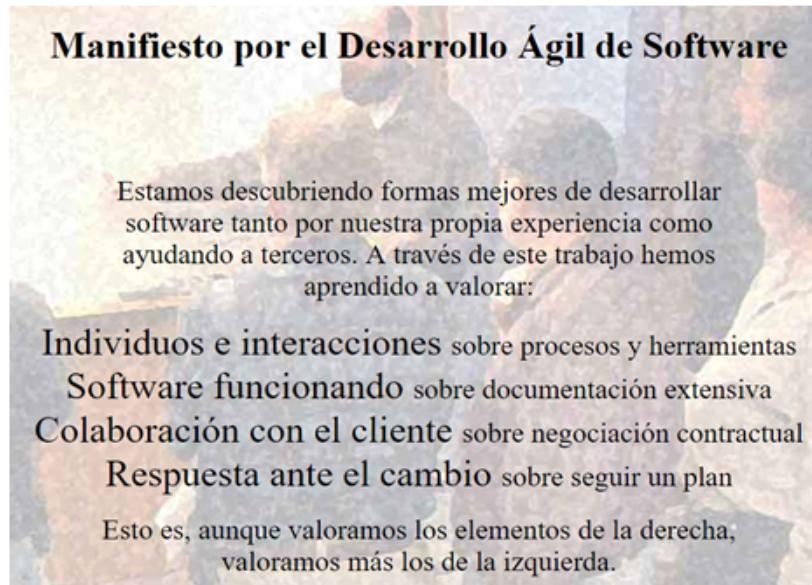


Figura 6.1: Ejemplo de Clasificación de imagen de TensorFlow

Fuente: [15]

6.2. Metodología SCRUM

La metodología SCRUM es una estrategia de gestión donde se aplica un conjunto de prácticas de manera regular para mejorar el trabajo colaborativo y obtener el mejor proyecto de Software posible[36].

Basado en la adaptabilidad, orientación a las personas y no los procesos y es iterativo e incremental. Requiere de la comunicación directa del cliente con el equipo desarrollador, el cual que será un miembro del equipo, al ser quien expuso, propuso y lideró el desarrollo.

La metodología SCRUM recibe los requisitos y los involucra como parte de la discusión de una reunión, para ser implementada. Se debe garantizar el desarrollo de los requisitos y permite detectar si estos producen conflictos.

6.2.1. Componentes de SCRUM

Para realizar el seguimiento de la metodología SCRUM, se requiere tomar nota de todas las decisiones realizadas, en general, la metodología fue seguida al pie de la letra, sin embargo, se destaca en Plan de Actividades, que hubo una pausa al avance del

proyecto durante vasto tiempo, por tanto, se produjeron diversos cortes y extensiones de tiempo al momento de realizar las actividades, posteriormente explicado en dicho subtítulo.

Roles de la Metodología SCRUM

- **Product Owner:** Es la persona que conoce el panorama completo del producto y el entorno del negocio, representa a los usuarios del producto y es responsable del seguimiento del proyecto.
- **SCRUM Master:** Encargado de garantizar el funcionamiento de la metodología y evaluar los procesos, es una responsabilidad del funcionamiento del modelo, es un rol flexible del que depende el éxito del proyecto.
- **Project Manager:** Es el encargado de dirigir el equipo a cumplir con los objetivos del equipo, se asegura de que los requerimientos se cumplan.
- **SCRUM Team:** Se le conoce así a los integrantes del equipo de desarrollo, que produce resultados, cuyas habilidades son aptas para el desarrollo del proyecto de Software.

Se asigna a cada uno de los integrantes del equipo un rol distintivo y aparte el rol de SCRUM Team.

Sprint

Un Sprint es también considerado una iteración y se organiza en la revisión de requisitos del usuario, el cual es la designación de los requisitos del proyecto, traduciéndose tanto en tareas de usuario como en tareas para los desarrolladores, que se distribuyen en los distintos Sprint y deben ser realizadas por el Rol de SCRUM Team. En un Sprint Backlog se realiza el seguimiento de una Sprint, es una pizarra donde se muestran las tareas a realizar en el proyecto, se detallará su funcionamiento.

Requerimientos

Los requisitos del usuario, también llamados requerimientos se los considera como un estatuto abstracto en alto nivel de un servicio. Además existen las restricciones del sistema para asegurar una funcionalidad detallada. Los requerimientos tienen las caracterizadas de ser completos, consistentes y precisos. La definición de requerimientos tiene que involucrar al cliente, usuario y los roles distintivos (no requiere al SCRUM Team), se debe definir e incluir en la documentación los requerimientos para que no existan conflictos entre los interesados y los desarrolladores del proyecto. La especificación de requerimientos se basa en la formalización y anotación escrita de los requerimientos, normalmente oficializa un contrato entre el cliente y el contratista sobre que se realizará específicamente en el proyecto para evitar demandas y huecos legales, debido a la mala interpretación o subjetividad de los requerimientos solicitados. Finalmente se realiza la especificación del Software, que es una descripción más detallada del Software, que servirá como base para el diseño e implementación de los requerimientos, esta es manejada por el SCRUM Team.

Tareas y Estimación de Poker

Los requerimientos se transforman en tareas, básicamente es reescrito para ser entendido y manejado por el SCRUM Team que realizaran las tareas, para realizar un seguimiento del tiempo que el equipo trabaja y se toma en realizar cada tarea, se debe reunir al SCRUM Team e indagar cuanto tiempo se estimará a cada tarea, para este proyecto, se emplea el método Estimación de Poker.

Estimación de Poker es una técnica de riesgo elevado en metodologías Ágiles, es una herramienta de asignación de tiempo requerido para realizar una tarea, que reúne a todo el SCRUM Team para realizar una estimación individual sobre la cantidad de tiempo que llevaría realizar esa actividad a través de cartas y una escala de tiempo, en este caso, de horas de 1 a 13 horas, para determinar los motivos que tiene esa persona para dar ese tiempo y llegar a un consenso sobre el tiempo requerido [36]. Este método de estimación se emplea también para definir el tiempo de trabajo diario en que se

trabajaría el proyecto.

Posteriormente, durante el proceso de elaboración de cada tarea asignada, se descubrió una realidad totalmente distinta a los números estimados, siendo indefinidamente mayores a los estimados, por tanto se tuvo que trabajar al menos 3 veces más el estimado para la mayoría de las tareas, detallado en el Product Backlog.

Reuniones

Las reuniones deben ser diarias y por Sprint, las cuales tienen el objetivo de definir cual es el trabajo que se debe realizar durante el Sprint. Para una reunión con la metodología SCRUM, deben participar todos los roles para poder generar la lista de tareas a realizar, además de determinar el objetivo de la Sprint.

Durante el transcurso del proyecto, en las reuniones diarias realizadas a horas **09:45-10:00**, se debe reportar 3 puntos del avance de los integrantes del SCRUM Team, que son el trabajo realizado, el trabajo a realizar y los problemas encontrados. Este punto de la metodología fue realizado con éxito.

Sprint Backlog

El Sprint Backlog es una pizarra donde se muestran las tareas a realizar en el proyecto, para el proyecto, se definió el uso de la herramienta HacknPlan, que es una página web dedicada a la administración de proyectos, brinda un servicio excepcional para la creación de un Sprint Backlog de la metodología ágil SCRUM, la cual fue recomendada por el Docente de Ingeniería de Software.

La Sprint Backlog tiene 4 secciones descritas en la tabla 6.1 y visualmente representadas en las figuras 12.1 y 12.2 que se encuentra en el Anexo: Control de Sprint.

Las tareas requieren de un formato para poder ser manejadas por el SCRUM Team y supervisadas por los demás roles, las secciones que componen una tarea son descritas en la tabla 6.2. En la figura 12.3 de Anexo: Control de Sprint, se puede observar un ejemplo de una tarea completada.

| | |
|-------------|---|
| Planificado | Las tareas traducidas para la facilidad del SCRUM Team y su resolución. |
| En Progreso | Tareas que actualmente están siendo elaboradas por al menos un miembro del SCRUM Team. |
| Pruebas | Tareas que han sido terminadas, pero requieren de una búsqueda de errores o aprobación del SCRUM Master. |
| Completado | Son las tareas finalizadas, aquellas que ya no requieren cambios y han sido aprobadas por el SCRUM Master y confirmadas por el Project Manager. |

Cuadro 6.1: Secciones de un Sprint Backlog

Fuente: Elaboración Propia basado en la Herramienta HacknPlan.

| | |
|---------------------|--|
| Título | El alias de la historia (tarea), se le asigna un nombre corto. |
| Importancia | Se determina su importancia en 3 niveles, bajo, normal o alta. |
| Estimación Inicial | Se determina el tiempo que toma la actividad, previamente definido con Planning Poker |
| Descripción | Lo que se busca realmente, el requerimiento funcional que se necesita para la elaboración de la tarea, se detalla lo que se requiere, esta sección es empleada solo para las tareas de programación. |
| Comentarios | Se describen los avances, problemas encontrados y varios a medida que se desarrolla o termina la tarea. |
| Registro de trabajo | Acompañado de un comentario donde se registra que se hizo, se anota cuanto tiempo estuvieron que miembros del SCRUM Team en la tarea. |

Cuadro 6.2: Secciones de una tarea dentro del Sprint Backlog

Fuente: Elaboración Propia basado en la Herramienta HacknPlan.

Product Backlog

El Product Backlog es una lista de requisitos o tareas que representa la visión y expectativa del cliente respecto a objetivos y entregas del proyecto. Se debe mencionar el valor y el costo de su finalización. Se puede observar el Product Backlog en la tabla 12.1 de Anexo: Product Backlog posterior a la finalización del proyecto.

Gráfica Burn-Up

Es una herramienta de seguimiento que se usa en la metodología ágil SCRUM, sirve para determinar el tiempo estimado del proyecto en general, para tener una referencia visual de cuanto progreso se realiza realmente, se establecieron los siguientes puntos:

- Versión 0.1: Tiempo en que se estima tendrá un prototipo para presentar.
- Horas Estimadas de Trabajo: Cantidad de horas totales estimadas para la realización del proyecto.
- Horas Acumuladas: Cantidad de horas registradas durante los Sprint para cada tarea que tiene progreso, no necesariamente terminadas para sumarlas a la gráfica.
- Trayectoria Estimada: La estimación optimista de la distribución de horas, determina la trayectoria de las horas acumuladas deseada.

Para estimar la cantidad de horas de trabajo se empleará un calculo simple. Primero se estiman 3 horas en promedio en un rango de 2 a 5 horas de trabajo diario mientras la Sprint este activa, se toma en cuenta que se trabajarán todos los. La estimación de tiempo toma en cuenta que la fecha de inicio del proyecto es el 17 de septiembre y la fecha de entrega estimada es el 14 a 17 de diciembre, determinando un total de 88 días mínimo, se las multiplica y redondea al dígito más grande como se ve en la ecuación 6.1.

$$528 \text{ Horas Estimadas de Trabajo} = 2 \text{ Horas Diarias} * 3 \text{ Personas} * 88 \text{ Días disponibles} \quad (6.1)$$

La cantidad de horas estimadas se redondea a 500 horas, este es un calculo pesimista, tomando en cuenta que el desconocimiento inicial de las herramientas con las que se trabajan y el tiempo que tome resolver problemas sea muy elevado, que durante el desarrollo se demuestra que es necesario.

El control de la gráfica es apreciado en la figura 12.4 en el Anexo: Gráfico Burn-Up Burn-Down posterior a la.

Gráfica Burn-Down

Es una herramienta de seguimiento que se usa en la metodología ágil SCRUM, sirve para determinar el tiempo estimado que se tiene en total para cada tarea y todas las tareas, para tener una referencia visual de cuanto progreso se ha completado en comparación al ideal, se establecieron los siguientes puntos:

- Remanente Ideal: En una trayectoria descendiente deseada para estimar resultados, se observa la cantidad de horas que faltan para llegar a la meta.
- Esfuerzo Remanente: Es la cantidad de horas que todavía faltan por registrar por parte del SCRUM Team.
- Tareas Remanentes: Cantidad de tareas totales de las Sprint que faltan por completar.
- Tareas Completadas: Tareas realizadas durante ese espacio de tiempo específico.

El control de la gráfica es apreciado en la figura 12.5 en el Anexo: Gráfico Burn-Up Burn-Down.

Análisis y Desarrollo del Diseño del Sistema Interactivo

El diseño propuesto es elaborado a partir de los requerimientos recopilados por parte de un cliente inicial externo a la universidad y se busca satisfacer las necesidades del usuario que lo emplee. La propuesta será diseñada a partir de las bases de la ingeniería de Requerimientos aplicada en la metodología ágil SCRUM.

La ingeniería de requerimientos del proyecto implica establecer lo que el cliente y usuario requiere del sistema interactivo [36], además de la funcionalidad esperada del proyecto, se debe definir, especificar y validar los requerimientos para poder aprobar su elaboración.

De la siguiente manera la solicitud del cliente inicia con un pedido ambiguo y coloquial por parte del instructor de artes marciales de uno de los integrantes del grupo al compartir su idea con el, juntos anotaron sus pensamientos con las siguientes palabras: “Me gustaría un juego con el cual pueda entrenar las poses y técnicas básicas de Karate siguiendo lo que hace en la pantalla y que me diga si lo hago bien o mal, pero no tengo un Kinect o una consola para jugar, solo mi laptop”, a lo cual el estudiante añadió “Pero también sería genial que puedas bailar e imitar lo que hace la gente, si va por esa línea, podrías generalizarlo para atraer más público”. Es de conocimiento general que las ideas suelen venir de lugares inesperados, en este caso, fue una conversación coloquial entre dos conocidos lo que inspiró el proyecto, a partir de esas palabras, se determina la base del proyecto, por tanto, se debe recabar entre las frases, se las dividirá e interpretará dividiéndolo en 4 secciones como se ven en la tabla 7.1. Si bien, el

| Solicitud en lenguaje coloquial | Interpretación inicial como requerimiento |
|--|--|
| Me gustaría un juego con el cual pueda entrenar las poses y técnicas básicas de Karate siguiendo lo que hace en la pantalla. | Desarrollar un sistema interactivo. Existe un vídeo o instrucción en una pantalla para poder entrenar artes marciales. El usuario debe moverse acorde lo visto en la pantalla. |
| Que me diga si lo hago bien o mal. | Se requiere de un sistema de calificaciones para los movimientos hechos por el usuario. |
| No tengo un Kinect o una consola para jugar, solo mi laptop. | Tiene que funcionar en una laptop. |
| Sería genial que puedas bailar e imitar lo que hace la gente. | No debe limitarse a artes marciales. |

Cuadro 7.1: Interpretación de la solicitud inicial del cliente

Fuente: Elaboración Propia.

instructor perdió el interés en el proyecto con el pasar del tiempo, el estudiante tomará el rol de cliente para el desarrollo del sistema interactivo.

Posteriormente, se vio conveniente ser capaz de usar vídeos proporcionados por el usuario para poder crear niveles propios, ya que si inicialmente el instructor lo deseaba para entrenar poses y técnicas de Karate, él habría provisto de los vídeos, sin embargo, se emplearan recursos de terceros para realizar las pruebas del prototipo que se desarrolle.

7.1. Análisis de Requerimientos

En un principio, se debe definir como se llamará a los distintos vídeos que se desean imitar, formalmente se lo conocerá en el contexto como: **Mapa, mapa es un conjunto de poses que el usuario debe imitar en tiempo cronometrado de un determinado vídeo.**

Se desarrolla un sistema interactivo que permite al usuario poder jugar distintos mapas y además crear mapas propios, para ello se requieren dos funciones fundamentales.

- Play: Refiere al acto de entrar a una lista de mapas de usuario, donde se seleccionará un mapa y se realizará el seguimiento de las acciones provistas por el sistema interactivo a imitar.

- Crear un nuevo mapa: Término dado a la elaboración de mapas, donde el usuario provee un vídeo que quiera convertir en un mapa del sistema interactivo.

Se debe dar a conocer la forma en que los usuarios interactúan con el sistema interactivo, definir ciertas limitación, realizar consideraciones respecto a la interfaz gráfica y a las modificaciones externas del producto.

Características de Usuario

El sistema interactivo contará con una categoría Apto para todo público según el sistema PEGI y Early Childhood en el sistema ESRB, por tanto los usuarios puede ser cualquier persona mayor a los 3 años que pueda emplear el sistema interactivo. Sin embargo, debido a que está diseñado para realizar actividad física, se recomienda un control de salud previo uso para no resultar herido o lastimado en caso de ser mayor de edad o tener problemas que limiten las actividades físicas. La creación de los mapas requiere de la lectura de un ligero manual de usuario y perseverancia para la actividad tediosa, por tanto, si bien puede hacerlo cualquiera, es recomendado que lo hagan mayores de 12 años.

7.1.1. Requerimientos Funcionales

Los requerimientos funcionales hacen referencia a todas las funciones que tiene disponible el usuario para realizar dentro del sistema interactivo, según los módulos y componentes externos que lo integran. Se mencionaron dos requerimientos funcionales, los cuales, se descomponen en:

Play

1. Jugar: La opción de afirmar que se jugara, debe dirigirte a seleccionar el mapa deseado.
2. Opciones: El usuario es capaz de interactuar con las opciones descritas en 3 y 4.
3. Subir/Bajar el volumen: Una de las opciones es poder subir y bajar el volumen del audio del vídeo en una escala de 0 a 1.

4. Cambiar la resolución: Una opción más, esta permitirá el cambio del tamaño de la ventana del sistema interactivo, debe tener múltiples resoluciones disponibles.
5. Seleccionar el nivel deseado: Una vez se elige jugar, el usuario puede seleccionar el mapa con el que interactuar.
6. Salir: La opción de dejar de ejecutar el sistema interactivo.
7. Mostrar calificación al terminar el mapa: Un usuario debe ser capaz de visualizar la calificación que va acumulando a medida que realiza el nivel.
8. Salir del mapa: La capacidad del usuario para interrumpir el seguimiento del mapa.
9. Pausar juego: El usuario debe ser capaz de detener el mapa en proceso cuando este haya sido selecto para ejecutar.

Crear un nuevo mapa

Para la creación de un mapa, se tiene que seguir una serie de 5 pasos que involucran al usuario:

1. Ejecutar Script de conversión de vídeo a JSON e imágenes en formato JPG: El usuario selecciona un vídeo, usará un Script para el primer paso de conversión a un mapa.
2. Ejecutar Script de JSON a .txt: Posteriormente, se vuelve a usar otro Script para una segunda conversión a un formato intuitivo de entender para el usuario
3. Selección manual de poses: Para crear un mapa, solo se requieren poses específicas para cada movimiento, se borran las que se consideren redundantes de entre los txt.
4. Guardar el resultado en una carpeta designada para los mapas del usuario: Se guarda en una carpeta el nuevo mapa, en preparación al siguiente paso.

5. Añadir un margen de tiempo para realizar las poses: Todas las poses de cada nivel tienen que tener una variable que determine el tiempo límite que se tiene para realizar esa acción.

7.1.2. Requerimientos No Funcionales

Los requerimientos no funcionales indican los límites del sistema o el proceso de desarrollo, entre ellos se emplearan:

1. Interfaz Gráfica de Usuario: Desarrollar una interfaz de usuario, para poder emplear fácilmente los requerimientos funcionales, el sistema interactivo de una manera más efectiva.
2. Funcionalidad: Que la capacidad del sistema interactivo sea capaz de funcionar acorde lo mencionado en los requerimientos funcionales.
3. Confidabilidad: Inicialmente, el plug-in de Unity tiene una tendencia a cerrarse repentinamente y al abrirlo por primera vez viene con archivo corruptos que se tienen que arreglar, se espera solucionar o al menos reparar esta debilidad de ser posible.
4. Intuitividad: Se espera que el cliente sea capaz de entender que actividad hacer con solo ver las funciones disponibles en la interfaz gráfica.

7.1.3. Funcionalidad Esperada del Sistema Interactivo

Se prepara un resumen visual de los requerimientos solicitados por el cliente, donde se separan los dos tipos de usuario que emplearán el producto final

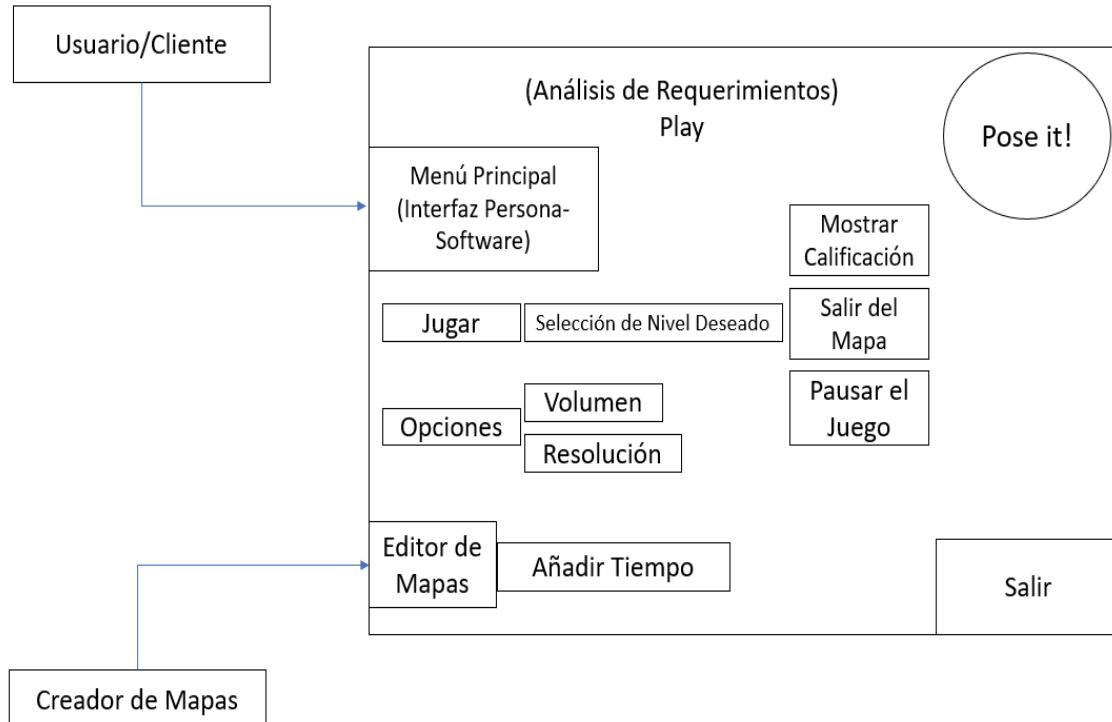


Figura 7.1: Ilustración de Requerimientos en la GUI

Fuente: Elaboración Propia

7.1.4. Diagrama de Flujo de Datos

Un diagrama de flujo de datos provee una función gráfica de los datos que se tienen dentro de un sistema de información, esencialmente se puede disponer de una vista a las variables y métodos esenciales, en este caso se aprecia el diagrama presentado por OpenPose que será modificado en una fase posterior del proyecto.

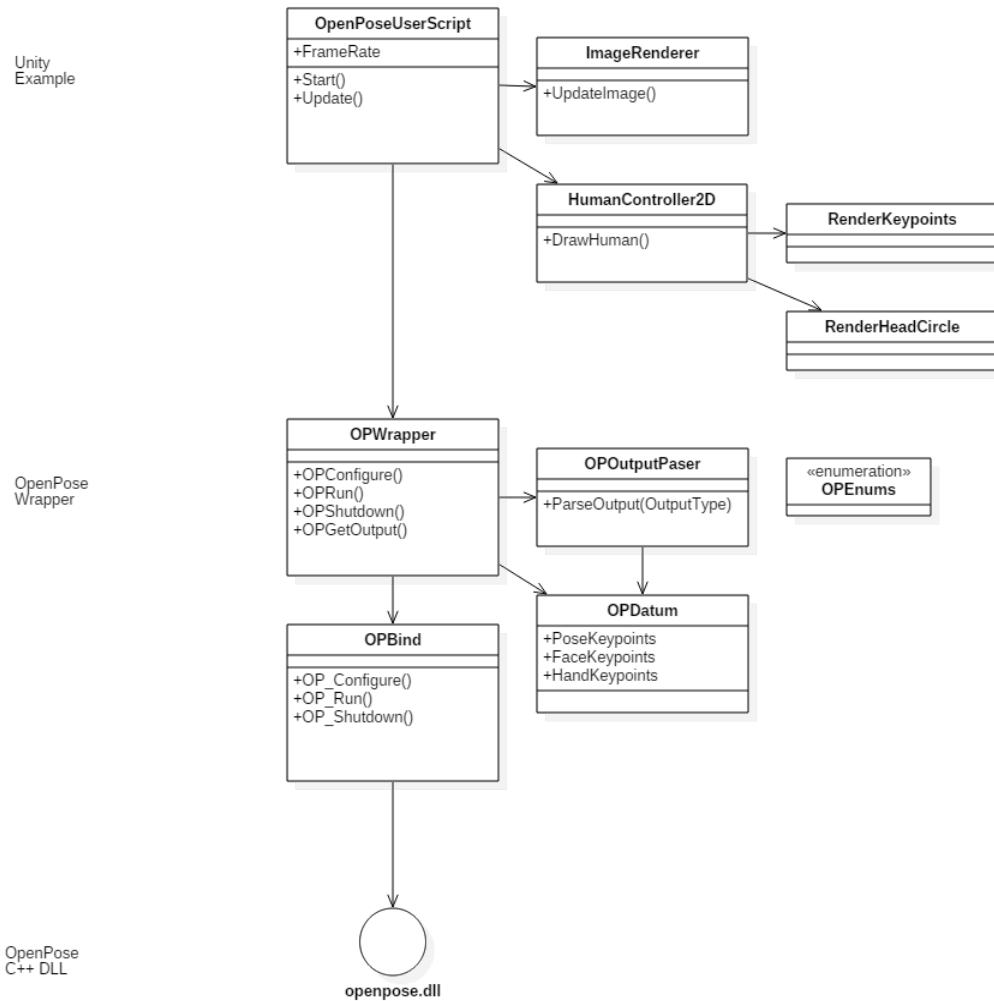


Figura 7.2: Diagrama de Datos de Flujo

Fuente: Versión original del Diagrama de Datos de Flujo OpenPose [16]

Desarrollo e Implementación del Sistema Interactivo

8.1. Funcionamiento Base

El sistema interactivo tomará en cuenta los requerimientos previos como la base de la elaboración del proyecto, si fuera necesario modificar, eliminar o añadir tareas según la exigencia de la metodología SCRUM, será en el transcurso del desarrollo.

Si bien el orden de las tareas y el procesamiento lógico puede resultar caótico, busca realizar las tareas más sencillas con el fin de profundizar el aprendizaje y enfocarse en procesos más complejos.

Para todas las pruebas iniciales realizadas se emplea el video I'm Han Solo Remastered, creación de Cinematic Captures, para un sistema interactivo alternativo, seleccionado arbitrariamente por el SCRUM Team para la creación de mapas, todos los derechos a su correspondiente autor [18].

8.1.1. Configuración del Entorno

Características de los Equipos Personales Empleados

La elaboración de este proyecto es realizada en 3 dispositivos distintos, dos laptops y una computadora de escritorio enfocadas para un alto rendimiento, cuyas tarjetas gráficas son de la marca nVidia Corporation, contando la GeForce GTX 960M, y dos con GeForce RTX 2060. La importancia de estos dispositivos es su capacidad de pro-

cesamiento gráfico o GPU, especializada para el desarrollo de sistemas 3D interactivos y gráficas de alta calidad, pueden llegar a ser varias veces más eficientes que la unidad central de procesamiento o CPU, en el caso de OpenPose, ejecutarlo con el CPU requiere de 8 GB de RAM, en cambio, las tarjetas gráficas tienen incorporadas memoria dedicada para el procesamiento gráfico, en el caso de OpenPose, ejecutarlo con el GPU requiere de 2 GB de memoria dedicada mínimo, siendo la cantidad disponible por la GeForce GTX 960M, cumpliendo el requisito mínimo, en cuanto a las GeForce RTX 2060, tienen una cómoda cantidad de 4 GB de memoria dedicada.

Para las cámaras de los dispositivos se emplean en el caso de las laptops, las cámaras de fábrica, que emplea la configuración genérica USB2.0 HD UVC WebCam y HD Webcam, en cuanto a la cámara de la computadora de escritorio, se emplea una cámara web genérica de orígenes desconocidos según el estudiante que la adquirió.

El Body Tracking está implementado por la Demo.unity de manera estándar, para poder ejecutarlo en todos los dispositivos de trabajo, se debe modificar el valor de una variable que cambie la resolución del OpenPose de net_resolution -1x176, caso contrario saltara el error más común de OpenPose 12.11 en el Anexo: Mensajes de Error.

Configuración Inicial de las SDK y herramientas

El desarrollo de la aplicación emplea Unity y Visual Studio, ya que Visual Studio es una SDK que permite programar en C Sharp, que tiene una biblioteca disponible para el desarrollo en Unity. Además se empleará la herramienta selecta, que es el plug-in de OpenPose para poder desarrollar, el proceso de instalación se explicará en la sección de anexos. Las versiones instaladas son Unity 2019.4.14f1, Visual Studio 2019 y OpenPose 1.5.0 (la versión del plug-in de Unity, por tanto no existen más opciones disponibles actualmente).

Las configuraciones son necesarias para disponer de los archivos dentro de la carpeta openpose_plugin\OpenPosePlugin\Assets, en su interior se encuentran las carpetas StreamingAssets y OpenPose, dentro de la carpeta StreamingAssets se encuentran los modelos BODY_25 y COCO, que serán necesarios para la estimación de poses. Dentro la carpeta OpenPose están las carpetas de Plugins, que contiene todas los DLL nece-

sarios para ejecutar OpenPose, de Scripts, contiene archivos para traducir los modelos e importar los DLL respectivos y la más importante Examples, que contiene Scenes y Scripts, siendo los más relevantes. Dentro de Scripts se encuentra el código fuente del plug-in, que es modificable por el usuario y en Scenes se encuentra el Demo.unity, que provee el formato de la figura 12.8 en Anexo: Proceso de Instalación de OpenPose y Unity para el Desarrollo del Proyecto.

Una vez inicializada la Demo.unity, se abre también el proyecto en C Sharp, abierto por defecto en Visual Studio 2019. Con ello, se tienen las herramientas para desarrollar el proyecto.

Siendo el primer paso la creación de un nuevo archivo.unity para el proyecto, copiando los objetos básicos del Demo.unity y relacionando la configuración de los Scripts para tener la base, se puede empezar su modificación.

8.1.2. Creación del Menú

El menú tiene 3 funciones principales:

- Jugar: Un botón Play para poder entrar a la pantalla de lista de mapas.
- Opciones: Un botón Options para cambiar el menú principal al menú opciones cuando se lo aprieta.
- Salir: Un botón Quit para terminar correctamente el funcionamiento del sistema interactivo.

Menú de Opciones

El menú de opciones tiene dos funciones:

- Una opción para modificar el volumen, deslizar la barra para alterar el nivel del volumen en una escala del 0 al 1 o para el usuario de 0 a 100 %.
- Una opción para modificar las resoluciones, Unity permite un número de resoluciones que están registradas en su documentación, se mostrarán las disponibles con el método Screen.resolutions;, su función es cambiar el largo y ancho del sistema interactivo.

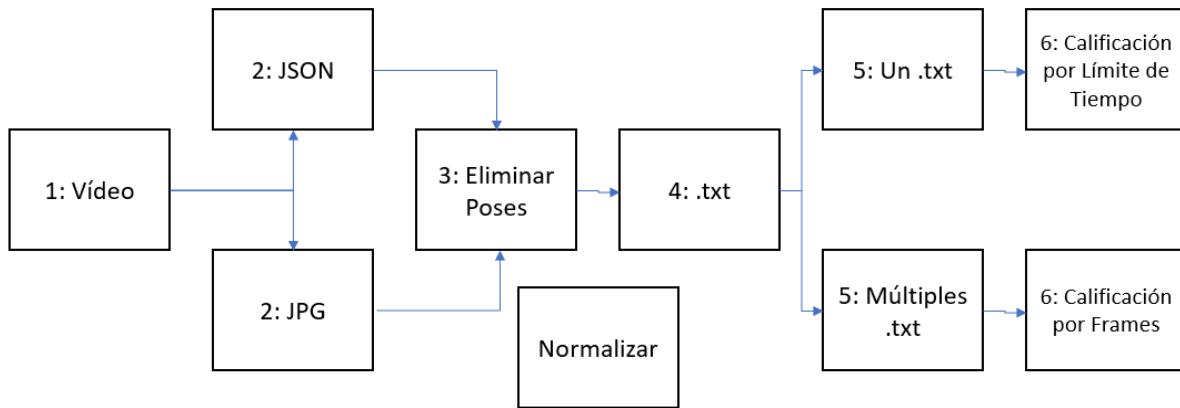


Figura 8.1: Pasos a Seguir Para la Creación de Mapas
Fuente: Elaboración Propia

8.1.3. Creación de un Mapa

La idea original sobre la creación de mapas es que el proceso de creación sea externo al sistema interactivo, de tal forma que solo tenga que añadir una carpeta con el contenido del mapa, fácilmente comprensible para el usuario que los creo y disponible para compartir entre los demás.

El método para la creación inicial de los mapas consta en 6 pasos necesarios y uno que es posiblemente útil, sin embargo, se determinará si hacerlo a futuro. Se desarrollarán dos métodos diferentes desde el paso 5, tal y como se observa en la figura 8.1. Dependiendo las dificultades y los resultados se determinará cual método conservar.

Conversión de Vídeo a JSON y JPG

El primer paso es la conversión de un vídeo grabado para la creación de un mapa y emplear un Script para ejecutar la OpenPoseDemo con la opción `--write_json output/`, `--number_people_max 1`, `--write_images output_images`, `--disable_blending`, `--frame_step 30` (por comodidad) y `--video ./examples_media_hansolo.mp4`, el resultado son dos carpetas llenas de archivos JSON y JPG como se ve en la figura 12.10 en el Anexo: Archivos Generados.

Las opciones empleadas por la herramienta OpenPoseDemo tienen las siguientes

funciones:

- `-write_json` dirección: Guarda los 25 puntos clave del esqueleto en un archivo JSON, se debe definir la dirección de la carpeta donde se guardarán.
- `-write_images` dirección: Guarda los 25 puntos clave del esqueleto en un archivo JPG, se debe definir la dirección de la carpeta donde se guardarán (si no existe la carpeta donde guardar, no funcionará).
- `-number_people_max X`: La cantidad X límite de personas que la herramienta va a reconocer, no se reconocerán más que X, pero si menos que X si no se los reconoce.
- `-disable_blending`: El vídeo grabado se torna negro, mostrando únicamente la pose que se reconoce.
- `-frame_step X`: Cada cuantos Frames se guarda un JSON y JPG de una pose.
- `-video` dirección: Dirección del vídeo que se va a convertir.

Eliminación de Poses

La eliminación de poses consiste en literalmente observar los archivos JSON y JPG de cada Frame guardado, ya que comparte el número del Frame elegido al final de su nombre, se observa la imagen JPG del Frame, si no es electo para el mapa, es eliminado el JSON y JPG. Este es un proceso lento y manual ejecutado por el creador de mapas, ya que el definirá la exactitud del mapa y de las poses que desea imitar en el nivel.

Conversión de JSON a txt y Almacenamiento

Independientemente si es previa o posterior a la eliminación de poses (de preferencia posterior, ya que existen menos archivos, por tanto, es más rápido), se ejecuta un Script diseñado para convertir los archivos JSON a un formato más reconocible por el usuario. Una vez realizada la conversión, el SCRUM Team vio dos alternativas para los archivos txt, los cuales almacenan una pose cada uno. Conservarlo como esta, siendo la carpeta

donde se guardan los txt el recurso principal para recurrir a las poses del nivel o convertir todos los txt en uno solo, almacenando todas las poses en un solo txt empleando un Script, si bien almacenarlos en un solo Script es más fácil, puede que los creadores de mapas encuentren molesto la ejecución de un Script más.

Adición del Tiempo Límite para Realizar las Poses

8.1.4. Comparación entre dos Poses

Esta es considerada la tarea más compleja y laboriosa de elaborar, ya que requiere del estudio previo del modelo BODY_25 selecto, manejo de los Scripts en C Sharp, estudio relacionado a Point Set Registration y Point Cloud.

Visualizar Movimientos del Usuario y Poses a Imitar

La idea original consta de tres partes, observar desde el Output de la cámara al usuario y sus movimientos y mostrar al costado un vídeo cualquiera para copiar los movimientos, al estar apagada la cámara se verá en azul, como se nota en la figura 8.2. Se realizan modificaciones de GUI para lograrlo, este no será el producto final, es más dirigido a la facilidad de pruebas, para observar los movimientos del usuario del lado izquierdo y el vídeo a imitar del lado derecho.

Adicionalmente a la visión del usuario y la pose a imitar, se debe visualizar la precisión que se tiene entre el usuario y la pose que se debe imitar, durante el desarrollo se lo empleará para revisar con los resultados de la normalización de los puntos clave la exactitud en tiempo real, calculo realizado en Normalización de Puntos Clave.

Posteriormente, durante la inclusión del tiempo límite, se visualizará solo una comparación, siendo o la precisión adecuada para una buena calificación o la última calificación antes de pasar a la siguiente pose.

Finalmente, una vez decidido el método de normalización y la valoración de la precisión, se mostrará al usuario únicamente si sus movimientos son perfectos, buenos, malos o erróneos.

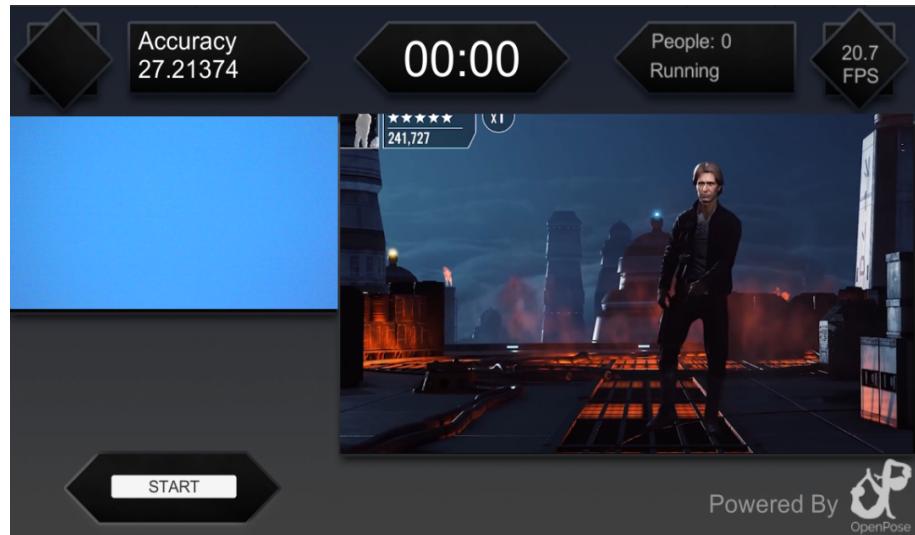


Figura 8.2: Primer GUI para el Desarrollo, Cambiada para el Diseño de GUI Final
Fuente: Elaboración Propia

Control de Tiempo del Mapa

Se determina a través del límite de tiempo para realizar la posición que se asignó al crear el mapa, la serie de poses a imitar están en un orden específico del primero al final, por tanto el tiempo límite de cada pose es más sencillo de determinar. Durante el tiempo total del vídeo del mapa, el usuario va a poder ver una pequeña visualización de las imágenes JPG de las poses mientras ve el vídeo para poder llevar el ritmo de las poses a imitar.

Se califica la posición en el tiempo respectivo y se debe proceder a la siguiente pose cuando termine y así se mueva al nuevo tiempo límite de la siguiente pose.

8.2. Normalización de Puntos Clave

Se requiere de desarrollar un método para comparar las poses del usuario en tiempo real con las poses del mapa, para ello se obtienen en tiempo real los puntos clave del cuerpo del usuario, pero todas las personas somos distintas, por mencionar el tamaño y aparte estamos en entornos distintos, algunos pueden alinearse a la cámara, otros suelen estar o muy cerca o muy lejos, sin mencionar el espacio que tienen para moverse, pero siempre y cuando se puedan observar todos los puntos claves necesarios, se puede

realizar una comparación entre poses correcta. Se debe considerar que se debe igualar lo más posible el tamaño de las poses independientemente de la posición que tengan en la pantalla y el tamaño de la persona.

El modelo BODY_25 tiene un conjunto de puntos guardados en forma de coordenadas, lo cual se considera como un Point Cloud, por tanto, en concepto se tienen dos Point Cloud muy distintas, pero similares (ya que ambos representan un cuerpo humano) que se deben comparar, para eso existe el Point Set Registration. Durante la exploración de algoritmos inicial no se encontró ningún algoritmo para facilitar el desarrollo de la aplicación, por lo que se desarrollaron distintas formas de medir la similitud entre las poses y la precisión de los movimientos que se necesitan representar.

El método presentado es el calculo de la precisión a partir de la normalización de los puntos dentro de la Point Cloud de tres formas distintas (presentadas por cada integrante del equipo), de las cuales se definirá la más adecuada. La normalización involucra ajustar la distancia entre los puntos para que tengan distancias proporcionalmente correctas a su forma original y permitan su comparación entre dos Point Clouds distintas.

8.2.1. Cálculo a partir de la Normalización de un punto

La normalización a partir de un punto busca un punto clave como base para normalizar el resto de los puntos, del modelo BODY_25 de la figura 4.3b se especificó el centro del cuerpo, el punto de la cadera central.

Se mueve el punto de la cadera central al centro de la imagen y se alinean los puntos de la persona de acuerdo a la distancia movida correspondiente como se ve en la figura 8.3. Se realizara una redimensión previa de todos los puntos existentes dentro de cada txt de las poses del mapa, se emplea la resolución 1280x720 al guardar archivos txt, cada uno de ellos será normalizado, por tanto a cada punto de las coordenadas se les aplica las ecuaciones 8.1 8.2, donde la constante $\beta = 24$ determinado por la cantidad de puntos clave, A_x es el largo y A_y la altura de la resolución, P es el vector de duplas que conforman las coordenadas de la Point Cloud de la pose del usuario y Q al de la pose del mapa.



Figura 8.3: Sobre posición de coordenadas de las poses al Centro
Fuente: Elaboración propia basado en [16]

$$P = \left\{ \left(\frac{x_{p0}}{A_x}, \frac{y_{p0}}{A_y} \right), \dots, \left(\frac{x_{p\beta}}{A_x}, \frac{y_{p\beta}}{A_y} \right) \right\} \quad (8.1)$$

$$Q = \left\{ \left(\frac{x_{q0}}{A_x}, \frac{y_{q0}}{A_y} \right), \dots, \left(\frac{x_{q\beta}}{A_x}, \frac{y_{q\beta}}{A_y} \right) \right\} \quad (8.2)$$

Se calculará la distancia N_x y N_y 8.3 que son la distancia que separa el punto de la cadera central de la otra, esos valores es la distancia que se debe recorrer en los ejes x,y para sobreponer las Point Cloud en el mismo espacio y el punto de la cadera central sobre el mismo.

$$N_x = x_{q8} - x_{p8} \quad N_y = y_{q8} - y_{p8} \quad (8.3)$$

Se calcula el promedio del porcentaje de la diferencia de la pose del usuario a la otra con la ecuación 8.4, donde se calcula el porcentaje de diferencia de cada punto en los ejes x,y se divide el total por la cantidad de puntos clave. En la ecuación 8.5 se calcula el promedio esperado de la pose del mapa.

$$Prom = \left[\sum_{n=0}^{\beta} (|(x_{pn} + N_x) * 100\%| + |(y_{pn} + N_y) * 100\%|) \right] / \beta \quad (8.4)$$

$$ExpProm = \left(\sum_{n=0}^{\beta} (|(x_{qn} + y_{qn}) * 100\%|) \right) / \beta \quad (8.5)$$

Finalmente se calcula el porcentaje de error que existe entre las dos poses en la ecuación 8.6.

$$Accuracy = 100 - \left(\frac{Prom}{ExpProm} \right) * 100 \% \quad (8.6)$$

Un ejemplo es desarrollado en el Ejemplo de Cálculo de Normalización Basada en el Centro de la Cadera, donde dadas 8 poses de 8 Frames de una grabación de prueba donde se realizan distintas pruebas, son comparadas y se concluye que calcular el promedio de diferencia entre dos poses no es una solución optima, pues un brazo puede estar totalmente fuera de lugar, pero al promediar todos los porcentajes de error se minimiza el efecto de la diferencia del brazo, que es muy importante. Por tanto este método de cálculo queda descartado.

8.2.2. Cálculo a partir de la Normalización de Cuadro de Point Cloud

Para la normalización del cuadro de Point Cloud, se busca los mínimos X_{min} y los máximos X_{max} del conjunto de puntos en las Point Clouds del usuario y de la pose guardada en el mapa 8.7 8.8 8.9 8.10.

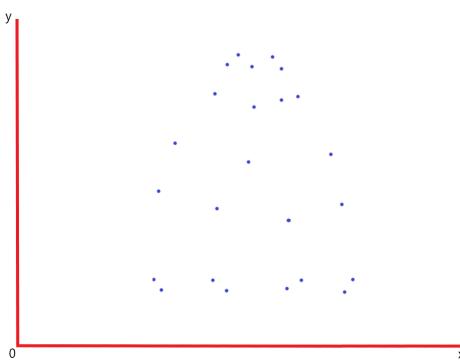
$$X_{pmax} = Max(P(x_p)) \quad Y_{pmax} = Max(P(y_p)) \quad (8.7)$$

$$X_{pmin} = Min(P(x_p)) \quad Y_{pmin} = Min(P(y_p)) \quad (8.8)$$

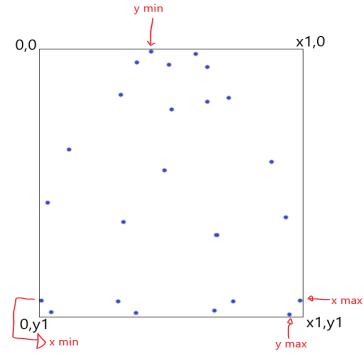
$$X_{qmax} = Max(Q(x_q)) \quad Y_{qmax} = Max(Q(y_q)) \quad (8.9)$$

$$X_{qmin} = Min(Q(x_q)) \quad Y_{qmin} = Min(Q(y_q)) \quad (8.10)$$

Los mínimos y máximos se convierten en los bordes limitantes de un cuadro imaginario que los encierra, como se ve en la figura 8.4 otorgando un tamaño del conjunto de los puntos, con ello se puede redimensionar el Point Cloud de la pose del usuario al tamaño de la Point Cloud de la pose del mapa con que se compara como se ve en la figura 8.5, para evaluar el desfase porcentual que existen en los ejes x,y de cada punto.



(a) Ejemplo de Point Cloud



(b) Cuadro de la Point Cloud

Figura 8.4: Ejemplos de Encuadre de una Point Cloud

Fuente: Elaboración propia

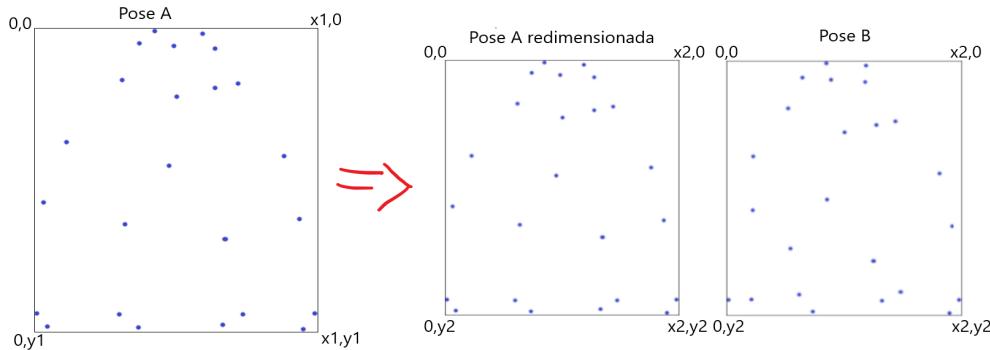


Figura 8.5: Redimensionamiento de la Point Cloud A al tamaño de la Point Cloud B
 Fuente: Elaboración propia

Se calcula el límite entre los dos extremos en x_{size} y y_{size} de cada pose 8.11 8.12 y se calcula una constante de proporción σ y φ 8.13 para los ejes x,y que permitirá comparar la distancia de cada pose respecto a la de la otra pose.

$$(x_{psize}, y_{psize}) = ((x_{pmax} - x_{pmin}), (y_{pmax} - y_{pmin})) \quad (8.11)$$

$$(x_{qsize}, y_{qize}) = ((x_{qmax} - x_{qmin}), (y_{qmax} - y_{qmin})) \quad (8.12)$$

$$\sigma = \frac{x_{psize}}{x_{asize}} \quad \varphi = \frac{y_{psize}}{y_{asize}} \quad (8.13)$$

Se calcula la precisión en la variable *Accuracy* 8.14, la cual será la sumatoria del cálculo de la diferencia porcentual de cada punto modificada para tener un valor expo-

nencial de acuerdo al porcentaje de diferencia. Donde la constante $\beta = 24$ debido a la cantidad de puntos clave, C 8.17 y D 8.18 son la varianza exponencial de los valores A 8.15 y B 8.16, que son el porcentaje de diferencia entre dos poses tomando en cuenta la proporcionalidad.

$$Accuracy = \sum_{n=0}^{\beta} (|C| + |D|) \quad (8.14)$$

$$A = \left(\frac{x_{pn} - x_{qn} * \sigma}{x_{qsize}} * 100 \% \right) \quad (8.15)$$

$$B = \left(\frac{y_{pn} - y_{qn} * \varphi}{y_{qsize}} * 100 \% \right) \quad (8.16)$$

$$C = \begin{cases} A^{\log A}, & \text{si } A \geq 1 \\ A, & \text{si } A < 1 \end{cases} \quad (8.17)$$

$$D = \begin{cases} B^{\log B}, & \text{si } B \geq 1 \\ B, & \text{si } B < 1 \end{cases} \quad (8.18)$$

El rango de la precisión de cada punto sería de $0 \leq x \geq 10000$ para un total de $0 \leq x \geq 250000$ del valor de *Accuracy*, siendo el 0 la comparación entre dos poses iguales y 250000 siendo todos los puntos de una pose en un extremo y los de la otra pose en el otro extremo en ambos ejes, lo cual es imposible en el contexto. Sin embargo, la volatilidad al tener un error en un punto específico no es lo suficientemente alta, por tanto, este método de cálculo fue descartado.

8.2.3. Cálculo a partir de la Normalización de la distancia entre dos puntos

La normalización a partir de la distancia cuenta con una constante $\beta = 24$ Key Points (ya que va del 0 al 24) entre dos puntos elige dos puntos del modelo BODY_25 de la figura 4.3b, el punto del cuello (Punto 1) y el punto de la cadera central (Punto 8), ya que conforman la línea principal del cuerpo que une extremidades inferiores y superiores.

res. Se obtiene una constante a partir de la división de las distancias de ambas poses a comparar que permite acomodar la posición de los demás puntos

Se convertirá a las Point Cloud como un vector de duplas x,y de la siguiente forma $P = \{(x_{p0}, y_{p0}), \dots, (x_{p\beta}, y_{p\beta})\}$ y $Q = \{(x_{q0}, y_{q0}), \dots, (x_{q\beta}, y_{q\beta})\}$, donde P es el vector del usuario y Q es de la pose del mapa, x_{pn} es un valor del vector, que respeta el orden de la tabla 4.1. Se debe calcular las distancias d_p y d_q que hay entre los puntos clave mencionados (cuello y cadera central) con la ecuación $d_p = distancia((x_{p1}, y_{p1}), (x_{p8}, y_{p8}))$ y $d_q = distancia((x_{q1}, y_{q1}), (x_{q8}, y_{q8}))$.

Se calcula una constante que determina la proporción entre las distancias entre el usuario respecto a la pose del mapa 8.19, esto nos permitirá definir la diferencia entre las distancias del punto del cuello de la pose del usuario respecto a la de la pose del mapa en las ecuaciones 8.20 y 8.21. La distancia D_x y D_y determinan la lejanía del punto del cuello respecto al otro punto del cuello en el cuadro de la pose basándose en sus proporciones.

$$\gamma = \frac{d_p}{d_q} \quad (8.19)$$

$$D_x = (x_{p1} * \gamma - x_{q1}) \quad (8.20)$$

$$D_y = (y_{p1} * \gamma - y_{q1}) \quad (8.21)$$

Finalmente, se calculará la diferencia que existe entre cada punto normalizado de la pose del usuario y los puntos de la pose del mapa, se resta D_x y D_y para sobreponer los ejes de las coordenadas de cada punto en la ecuación 8.22 para obtener la diferencia entre una pose de otra.

$$Accuracy = D_{sum} = \sum_{n=0}^{\beta} (|x_{pn} * \gamma - x_{qn} - D_x| + |y_{pn} * \gamma - y_{qn} - D_y|) \quad (8.22)$$

El resultado varía bastante dependiendo de la resolución de la cámara. Se estima un rango de Accuracy entre $500 < x \geq 1000000$, al ser tan amplio el rango, cuando se mueve un punto fuera de lugar el cambio de la precisión es elevado, por tanto, este es el método de calculo seleccionado.

8.3. Diseño De GUI

El diseño de la GUI de la figura 7.1 en la fase de diseño se tomó en cuenta para su creación, donde se tienen los botones y funciones esperadas. Para el diseño también se toma en cuenta el principio KISS (“Keep it Simple, Stupid!”), que establece que gran parte de los sistemas son mejores si se mantienen simple que complejos, este se vuelve un objetivo clave para el diseño de la GUI.

La licencia de OpenPose especifica la necesidad de mantener en claro que el uso de la API esta dirigido a investigación y desarrollo privado, perteneciendo a su creador todos los programas desarrollados con su uso. Se resalta también que a estas etapas del desarrollo recién **se determinó el nombre del sistema interactivo POSE IT!**, en la figura 8.6 se observan los logos de POSE IT! y de OpenPose, así mismo las 3 funciones mencionadas.



Figura 8.6: Menú de el Sistema Interactivo POSE IT!

Fuente: Elaboración Propia

Al presionar Options en la figura 8.6 se observan las dos funciones observadas funcionales marcadas en rojo de la figura 8.7, así mismo se añadió la posibilidad de poner la pantalla en modo “Fullscreen”, que en español es pantalla completa.

Al apretar Play de la figura 8.6 se pueden observar la lista de mapas disponibles, existen dos botones marcados en rojo en la figura 8.8 para “Volver al menú principal 8.6



Figura 8.7: Menú de Opciones

Fuente: Elaboración Propia

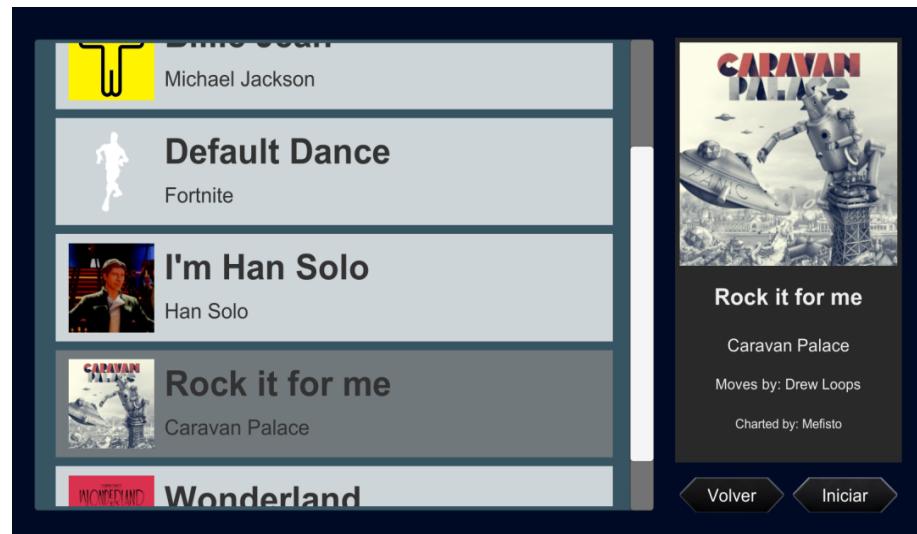


Figura 8.8: Menú de Listas para Mapas de POSE IT!

Fuente: Elaboración Propia

o para iniciar a interactuar con el mapa elegido. Marcado en verde, se puede apreciar la lista de mapas disponibles, que al seleccionar un mapa, se escuchará levemente el audio del mapa, si es configurado, en la parte marcada con azul, se verá también el nombre del mapa, el autor que creo la pieza, el grupo que lo toco y quien creo el mapa.

8.4. Modificación y Requerimientos Adicionales

La metodología SCRUM ofrece las posibilidades de cambiar rápidamente la visión de algunas tareas, las transforma y añade nuevas tareas continuamente para la mejora de las aplicaciones que se desarrollan, este caso no es una excepción, pues en el transcurso del desarrollo se tuvieron cambios respecto a la creación del mapa y el modo de juego. La creación de mapas originalmente era hecha a partir de un grupo de Script externos a la aplicación principal, sin embargo, esta actividad sería mucho más cómoda para el usuario si el sistema interactivo mismo lo tuviera implementado en su GUI. Para esto se debe modificar algunos requerimientos funcionales y el diseño de la GUI, además de añadir una GUI para la creación de mapas.

En el modo de Play para imitar poses, se modifica la GUI a una versión final, pero en la funcionalidad de sus requerimientos no se sufren cambios directos, no son relevantes para el manejo del sistema para el usuario.

8.4.1. Requerimientos Adicionales

En el modo de Creación de Mapas, se debe tomar en cuenta el modelo de pasos a seguir para crear un mapa mostrado en la figura 8.1 y cambiarlo para poder generar los mapas con el sistema interactivo como se ve en la figura 8.9, ya no se requiere el paso 2, donde el usuario tenía que seleccionar las poses que deseaba imitar, si no que se hace de manera automática, ya no se requiere de la eliminación manual de poses, se filtran las poses para imitar. Las poses selectas pasaran por un proceso de normalización de valores de la Point Cloud para facilitar su uso en el mapa y se guardaran en un solo archivo .txt, que gracias a la creación de mapas con interfaz gráfica, ya no requiere ser manejado por el usuario y es más sencillo de manejar, la calificación por límite de tiempo es manejada por el usuario en la interfaz gráfica proporcionada.

En el menú principal se debe tener la opción de entrar al modo de Creación de mapas.

Así mismo se tiene en consideración el análisis de los requerimientos, que previamente se ilustraba en la figura 7.1, se lo modificara como se ve en la figura 8.10, la función

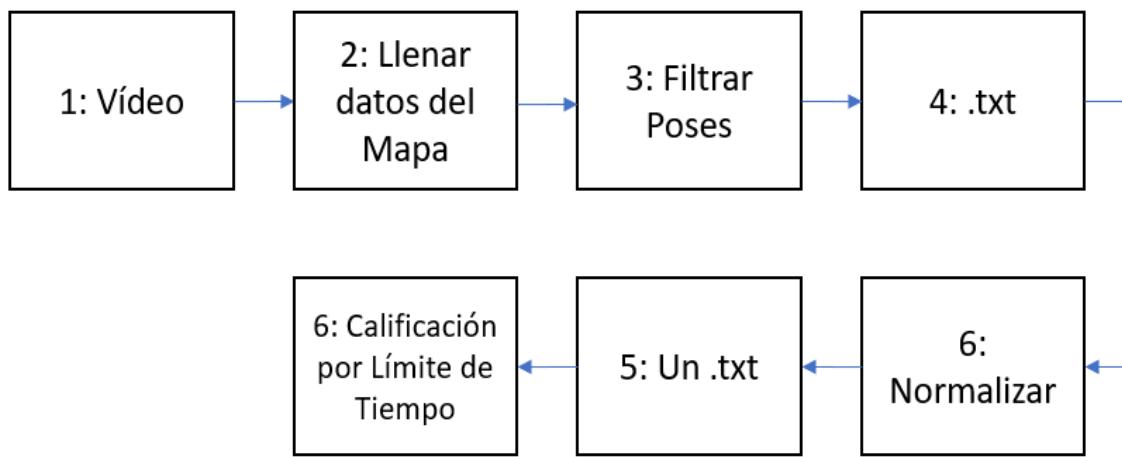


Figura 8.9: Pasos a Seguir Para la Creación de Mapas

Fuente: Elaboración Propia

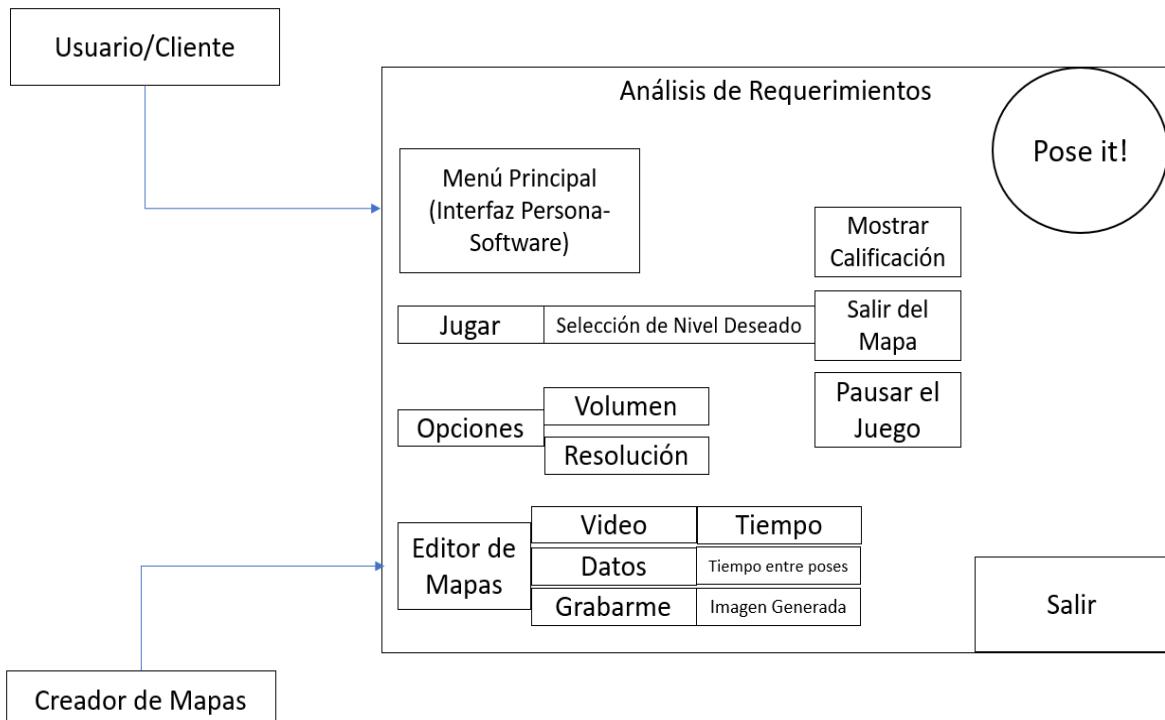


Figura 8.10: Ilustración de Requerimientos en la GUI

Fuente: Elaboración Propia

“Grabarme” será presentada, permite al usuario grabarse a si mismo para construir un nivel.

Nuevas Requerimientos Funcionales de Creación de Mapa

Para la creación del mapa se deben tener en cuenta los siguientes requerimientos funcionales:

- Datos del Mapa: Se debe tener una manera de ingresar los datos del mapa que se creara y visualizan en el menú de listas al seleccionar un mapa^{8.8}, estos son el nombre, el artista, la persona que se mueve y el usuario que crea el mapa.
- Seleccionar vídeo: Único factor que se mantiene externo, se tiene como requisito que el vídeo sea en formato mp4, este es guardado en la dirección ./OpenPose-Plugin/Media, así mismo se tiene previsto un modo de usar cámara para grabarse a si mismo, se hará mención de esta función en recomendaciones.
- Mostrar vídeo y poses de mapa: Se debe mostrar tanto el vídeo seleccionado como las poses que se están guardando.
- Control del Tiempo: Se debe poder especificar en que parte del vídeo se debe comenzar a grabar, así como cuanto tiempo se va a grabar y cada cuanto tiempo se debe generar una pose para guardar en el mapa.
- Start: Se debe tener una función que permita iniciar la creación del nivel con los datos ingresados.
- Usar imagen Generada: En el menú de listas se tiene una imagen junto a los datos del mapa, esta puede ser modificada manualmente o ser auto generada con un campo que lo permita y brinde la imagen del vídeo durante su creación.

Todos los datos e imágenes del mapa se pueden modificar manualmente interviniendo en los archivos del mapa, estos mapas se guardaran en la carpeta ./OpenPosePlugin/Custom/ %NombredelMapa, se tiene el txt con las poses generadas del mapa, la imagen del mapa que se mostrará en el menú de listas y el vídeo que se deberá imitar, también se disponen de sus archivos .meta, cuya única función es proporcionar un formato accesible para Unity.

8.4.2. Diseño de GUI

El menú principal tendrá un nuevo botón para acceder al modo de Creación de Mapas, se podrá observar la versión limpia de la GUI del sistema interactivo en el anexo Diseño de GUI Modificados.

Modo Play

En la GUI del modo Play se tienen los siguientes campos enmarcados con cuadros de colores en la figura 8.11:

1. Datos del Nivel (azul): Se muestra el valor de la precisión calculado en la ecuación 8.22, el tiempo desde que empezó el nivel, si se encuentra el jugador y los FPS a los que va el modo Play.
2. Mostrar Usuario (rojo): Se observa el usuario y su pose a partir de la cámara y el plug-in de OpenPose.
3. Se muestra el vídeo del mapa a imitar (verde): Se observa el vídeo del mapa.
4. Botón de Start (Celeste): Al presionarlo, se tiene 3 segundos para que el usuario se acomode y empiece a imitar las poses.

Se observan en las figuras 12.21, 12.22, 12.23 los diferentes estados de la GUI, que marca continuamente color rojo o verde para las poses imitadas correctamente o erróneamente.

Modo Creación de Mapas

En la GUI del modo de Creación de mapas se tienen en cuenta los siguientes componentes enmarcados con cuadros de colores en la figura 8.12:

1. Start(Gindo): Botón para empezar la creación del nivel, es empleado cuando se llenan los datos del mapa.
2. Elección de vídeo(Verde): En este lugar se seleccione con cual de los vídeos disponibles se creará el mapa.

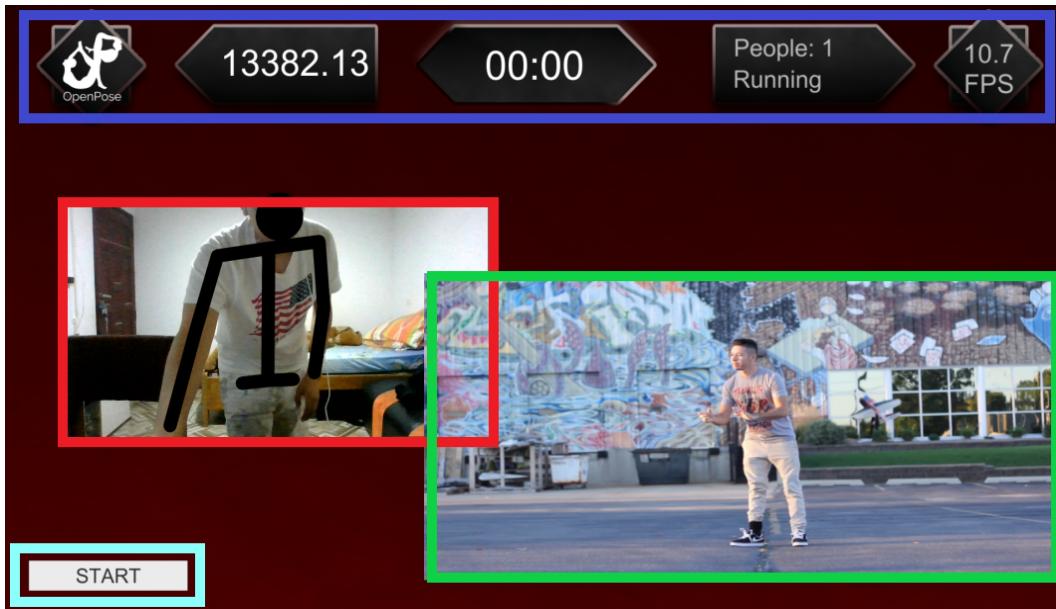


Figura 8.11: Diseño Final de GUI para el Modo Play
Fuente: Elaboración Propia



Figura 8.12: Diseño Final de GUI para el Modo de Creación de Mapa
Fuente: Elaboración Propia

3. Tiempo entre poses guardadas y uso de Imagen Generada (Celeste): El primero para que el usuario seleccione cada cuanto tiempo se debe guardar una pose tras otra a imitar, el segundo para ver si se guarda la imagen auto generada que se verá en la lista.

4. Datos del Mapa (Rojo): Se deben ingresar los datos que se visualizan en el menú de listas al seleccionar un mapa 8.8, estos son el nombre, el artista, la persona que se mueve y el usuario que crea el mapa.
5. Control de Tiempo(Violeta): Se ven los campos de Time to Record y Delay Before Start, el primero cumple la función de cuanto tiempo se va a grabar y el segundo cuanto se debe esperar para comenzar a grabar.
6. Mostrar vídeo o usuario (Amarillo): Al presionar Start se observará el transcurso de la creación del mapa, viendo el vídeo y las poses generadas.
7. Funciones y Datos Extras(Azul): Se tendrá un botón que permitirá al usuario retroceder al menú principal nuevamente, así mismo se vera el logo de OpenPose, el tiempo transcurrido hasta el momento, si se puede observar algo/alguien a quien imitar y los FPS a los que transcurre la creación del mapa.

8.4.3. Diagrama de Flujo de Datos Modificado

En el diagrama de flujo se llevaron a cabo varias modificaciones a nivel de OpenPose desde la parte de OpenPoseUserScript, ya que el nivel de DLL y Wrapper son parte del plug-in de OpenPose.

Se crean presentes dos OpenPoseUserScript que tienen las mismas dependencias, sin embargo tienen funcionalidades distintas, ya que uno es para el modo Play y el otro es para el modo de Creación de Mapa, su función es tener los métodos y atributos necesarios para realizar el funcionamiento de los requerimientos funcionales.

Se crea una clase de Main Menu para la funcionalidad del Menú principal, el cual esta conectado a ListGeneration, que crea la lista visual de mapas del menú de listas y una clase de LevelLoader para poder seleccionar y realizar las transiciones requeridas del menú de listas al modo Play.

Se crea una clase de FPSTarget para los OpenPoseUserScript, que indicarán la velocidad de 60 FPS de los vídeos para igualar el tiempo del vídeo y el del usuario al imitar poses.

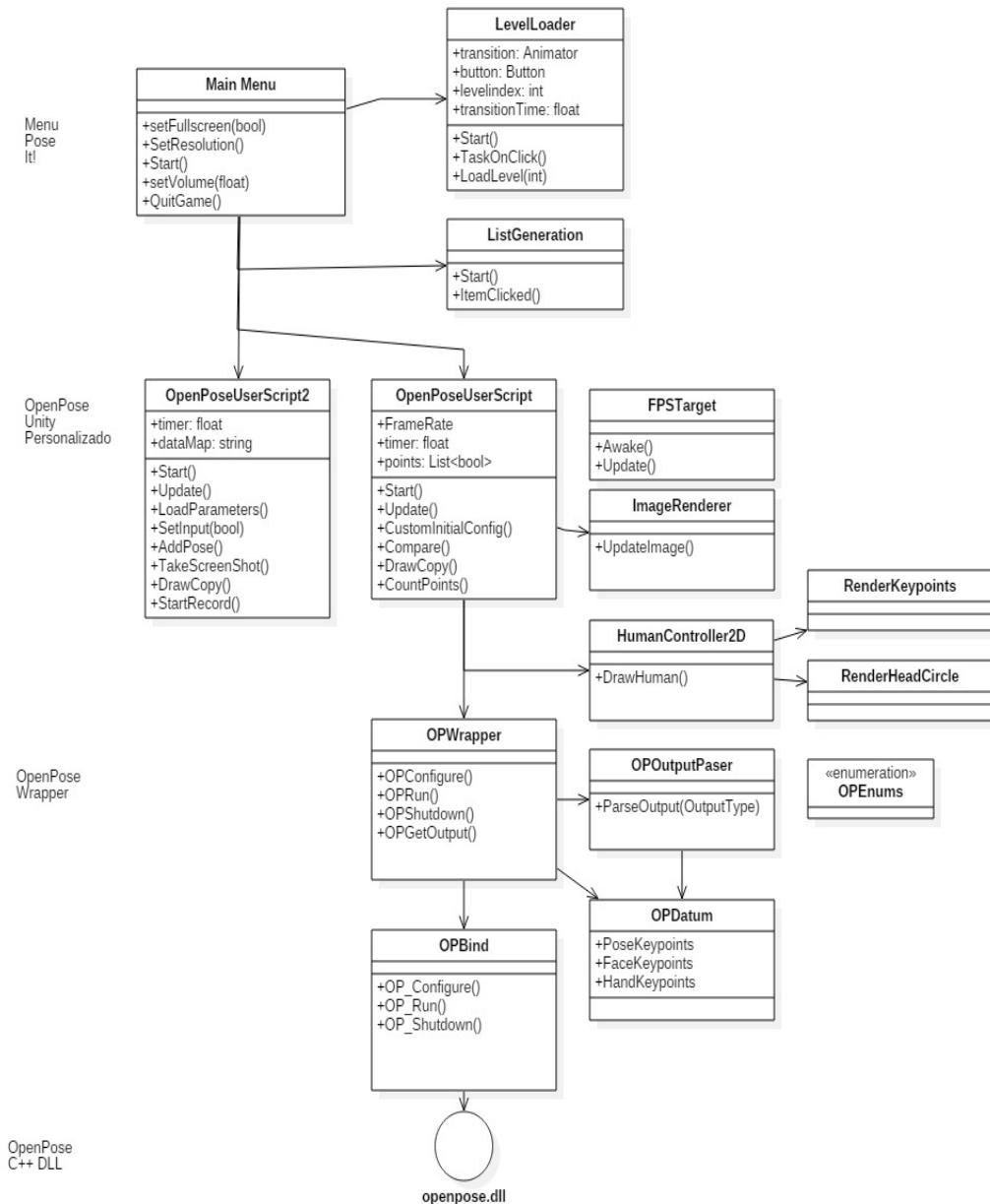


Figura 8.13: Diagrama de Datos de Flujo
Fuente: Modificación de la versión original del plug-in de OpenPose[16]

Plan de Actividades

El desarrollo del proyecto inicia oficialmente el día 17 de Septiembre, con la definición del perfil del proyecto por parte del grupo de estudiantes.

El desarrollo fue interrumpido debido a la concentración del equipo en otras actividades, que exigió el 100 % del tiempo libre y de estudio y progreso en el proyecto, siendo más específicos la materia de Aplicación con Redes, estimando entre 4 a 8 horas diarias (incluyendo fines de semana) de exigencia entre estudio y prácticas para mantener el ritmo solo a esa materia. Debido a ello, desde la fecha 18 de septiembre del 2020, a la fecha 4 de noviembre del 2020, el progreso fue mínimo, reduciéndose a la búsqueda de API's y herramientas, así como la redacción base del proyecto y la oficialización interna del producto deseado.

La metodología SCRUM fue inicializada oficialmente y de manera constante el día 10 de Noviembre del 2020, planteando un rango de horas de trabajo de 2 a 5 horas diarias mínimo (incluyendo fines de semana), con la finalización del Sprint con la conclusión de las tareas designadas al Sprint, en caso de resultar imposible terminar a tiempo una tarea, se la arrastrará a la siguiente Sprint, forzando una finalización del Sprint hasta donde se encuentra el proyecto en ese momento.

9.0.1. Cronograma

| September | | October | | | | | November | | | | | December | | | | |
|-----------|--------|---------|--------|--------|--------|--------|----------|--------|---------|---------|---------|----------|---------|---------|---------|---|
| Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 | Week 13 | Week 14 | Week 15 | Week 16 | |
| 2 | 3 | 4 | 5 | 6 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 0 | 1 | 2 | 3 | 4 |
| T1 | | | | | | | | | | | | | | | | |
| T2 | | | | | | | | | | | | | | | | |
| T3 | | | | | | | | | | | | | | | | |
| T4 | | | | | | | | | | | | | | | | |
| T5 | | | | | | | | | | | | | | | | |
| T6 | | | | | | | | | | | | | | | | |
| T7 | | | | | | | | | | | | | | | | |

Cada TX representa un título

- T1 *Redacción del Perfil.*
- T2 *Investigación.*
- T3 *Redacción del documento.*
- T4 *Implementación de la aplicación.*
- T5 *Revisión final del documento.*
- T6 *Revisión final de la aplicación.*
- T7 *Redacción de la presentación.*

Conclusiones

Durante el trayecto del proyecto se realizaron diferentes actividades y seguimientos de las mismas, además de la aplicación de herramientas externas apropiadamente empleadas y una metodología ágil en su mayoría, si bien se tuvo dificultades de gran proporción con la estimación de tiempo, a fin de cuentas, no fue un factor que determine el fracaso del proyecto, es más, se afirma que los objetivos fueron cumplidos de acuerdo al producto mínimo viable establecido en delimitación.

En función a los objetivos específicos, a lo largo del desarrollo e implementación del prototipo diseñado, se llega a la siguiente conclusión.

Respecto al primer objetivo específico, utilizar Software existente para el seguimiento corporal utilizando una cámara, en el estudio de alternativas se observan varias herramientas de Software que posibilitaban el desarrollo del proyecto, al emplear OpenPose, la mejor alternativa, que sirve para el seguimiento corporal y se la empleo utilizando cámaras web, siendo las pruebas con el prototipo realizadas empleando diferentes modelos de cámara Web, se determina que el objetivo específico es cumplido satisfactoriamente.

Respecto al segundo objetivo específico, implementar una función para registrar mapas de movimiento propios del usuario, fue el objetivo específico más conflictivo, pero a pesar de todos los problemas, con la supervisión de los roles distintivos oportunos, se redirigió adecuadamente el proyecto a su finalización, proyectando los requisitos del producto mínimo viable.

Respecto al tercer objetivo específico, proveer una alternativa factible al mercado de

sistemas interactivos con Body Tracking tales como Just Dance, se determina a través de la precisión del juego, la dificultad de crear mapas y factores menores subjetivos por parte de los usuarios, como la creación de mapas y la dificultad de reconocer adecuadamente las poses del usuario, que si es una alternativa factible, sin embargo, no se aproxima al nivel de calidad que ofrece un sistema interactivo como Just Dance.

Respecto al objetivo general, que especifica el desarrollo de un sistema interactivo con Body Tracking con una cámara común para múltiples propósitos, se considera exitoso. Uno de los mayores desafíos fue la falta de herramientas, característica principal del proyecto, a pesar de la simplicidad de la UI, las funcionalidades principales del producto si cumplen con los requisitos críticos del usuario. Adicionalmente, el cumplimiento de los objetivos específicos producen un efecto en cadena que se redirige directamente al cumplimiento del objetivo general. Por otro lado, el desarrollo de nuevas alternativas, la constante mejora de versión de la herramienta OpenPose aplicada, así como las posibilidades de mayor libertad legal de emplear distintas herramientas, podrían terminar por incrementar la calidad del producto final y su firmeza como una alternativa frente a otros sistemas interactivos.

Durante la elaboración y desarrollo del proyecto, existieron tres principales problemas que acontecieron y demostraron lo caótico que un proyecto puede llegar a ser, los problemas provinieron de el tiempo disponible durante un mes y medio, la estimación de horas requeridas para cada tarea y el conflicto de ideas en la elaboración de tareas.

Un problema que tuvo el inicio del proyecto y su fase temprana fue el cese repentino de las actividades, de tiempo disponible, debido a la inoportuna aparición de una actividad curricular dentro de la universidad, más específicamente la materia de Aplicación de Redes, los estudiantes carecieron del tiempo necesario para realizar el proyecto, ya que la materia exigió toda la atención que se tenía disponible, entre 4 a 8 horas.

Uno de los principales problemas es la infraestimación de tiempo al usar la herramienta de Estimación de Poker que derivo en que la mayoría de las tareas lleguen a triplicar su tiempo estimado, si bien, el total de 500 horas de trabajos fue calculado, no se esperaba que realmente fuera a ser necesario como un tiempo cercano, este problema fue producido por el desconocimiento de las herramientas necesarias para desarrollarlo,

en muchos aspectos, esta fue una experiencia totalmente nueva. Para solucionarlo, se reguló con un factor de ajuste de la estimación inicial en la asignación del tiempo de las tareas que los llevo a ser más realistas y precisos, en general el proyecto tardo aproximadamente 3 veces más del estimado inicial, el cual está escasamente en el margen del tiempo total estimado del trabajo, en la medida de lo aceptable.

Un problema fue la mala asignación de tareas, producto de la negligencia en el seguimiento necesario de la metodología, siendo más específico, miembros del SCRUM Team encontraron conflicto de ideas en el segundo objetivo específico, ya que durante el desarrollo de las tareas relacionadas a registrar mapas del usuario, el choque entre las ideas de implementación ya desarrolladas, el tiempo invertido en ello y la imposibilidad de enlazar sus resultados, retraso el proyecto y por poco colapsa debido al tiempo, una vez fue definido más claramente las labores a realizar por el SCRUM Master, el SCRUM Team se puso de acuerdo y el desarrollo volvió a encaminarse.

Recomendaciones

Durante el desarrollo del sistema interactivo, se descubrieron varias limitantes que considerar en ciertas facetas de la elaboración del proyecto. Además del seguimiento de la elaboración del sistema interactivo.

Requerimientos de los equipos de desarrollo y usuario

Si bien, una de las características del proyecto, es no requerir una cámara de profundidad de campo, como el Kinect o la PlayStation Camera, termina siendo una de sus debilidades en cuanto a equipo se refiere, ya que termina requiriendo un mínimo de 2 GB de GPU. Se recomienda modificar el requerimiento mínimo de GPU a CPU, de tal manera que los requerimientos del equipo sean dirigidos a la CPU y no a la GPU, ya que existen dispositivos que están a la altura de procesamiento, pero no cuentan con una tarjeta gráfica lo suficientemente potente, ya sea por ser un equipo diseñado para trabajo de administración o no haya sido pensada para procesar gráficos de alta calidad.

Otra recomendación sería expandir el rango de sistemas operativos en los cuales se puede ejecutar el sistema interactivo, la herramienta OpenPose puede ser empleada en Linux y Mac, así mismo, Unity cuenta con soporte en Mac, por tanto, sería viable considerar esta posibilidad.

Se recomienda que a medida que salgan futuras actualizaciones de las herramientas de Body Tracking (OpenPose) se vaya actualizando el programa, ya que esta se encuentra en desarrollo y mejora continua.

Uso del sistema interactivo

Se recomienda la integración o mejora de la interfaz de usuario para la facilidad de la creación y edición de mapas creados por el usuario. Además, se desea que en el futuro, el reconocimiento de las poses mejore considerablemente con el tiempo, para que la necesidad de tener un ambiente preparado donde jugar reduzca sus condiciones y se permita en un mayor rango de ambientes.

Desarrollo completo del prototipo

Si bien el prototipo cumple con el producto mínimo viable, también se observaron otras posibles funciones para aumentar la satisfacción del cliente, tales opciones mencionadas en delimitación que son una playlist de mapas y el observar a un jugador experimentado realizar el mapa, incluso mostrar sus puntuaciones pasadas para compararlas con las que siguen.

Además, se recomienda ver posibles modificaciones a la forma en que se realiza el Point Set Registration para la calificación de movimientos, ya que esta todavía tiene muchas variaciones y algoritmos posibles para su mejora. La posibilidad de grabarse a si mismo para la creación de mapas es una opción con vista a futuro.

Bibliografía

- [1] Esrb game ratings. <https://www.esrb.org/>, 1994.
- [2] Las etiquetas de edad pegi. <https://pegi.info/es>, 2003.
- [3] The eye of judgment (ps3). <https://forum.gamesaktuell.de/blogs/imperatcore/361571-augen-zu-und-durch-prev.html>, 2007.
- [4] Holiday gaming gift guide: Xbox 360 kinect. <https://owtk.com/2011/11/holiday-gaming-gife-guide-xbox-360-kinect/>, 2011.
- [5] The last of us - motion capture. <https://www.youtube.com/watch?v=9TxJWY19xxk>, 2013.
- [6] Crearon una tÉcnica de rehabilitaciÓn virtual para pacientes con discapacidad. <https://www.elfederal.com.ar/crearon-una-tecnica-de-rehabilitacion-virtual-para-pacientes-con-discapacidad/>, 2016.
- [7] Playstation camera. <https://www.playstation.com/en-us/accessories/playstation-camera/>, 2016.
- [8] Sony eye-camera, ps3 cámara web 640 x 480 pixeles usb 2.0. <https://icecat.biz/es/p/sony/9473459/webcams-eye-camera-+ps3-1269549.html>, 2016.
- [9] Intel lanza dos cámaras realsense con sensor de profundidad: visión en 3d para cualquier dispositivo. <https://www.xataka.com/realidad-virtual-aumentada/intel-lanza-dos-camaras-realsense-con-sensor-de-profundidad-vision-en-3d-para-> 2018.

- [10] [visual body tracking]. <https://www.youtube.com/watch?v=o1WRrjs0m8o&app=desktop>, 2018.
- [11] Azure kinect dk. developer.microsoft.com/en-us/mixed-reality/, 2020.
- [12] Working with the azure kinect developer kit. <https://www.infoworld.com/article/3562738/working-with-the-azure-kinect-developer-kit.html>, 2020.
- [13] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [14] Anónimo. How to get diagonal line of a cube in a point cloud? <https://stackoverflow.com/questions/57089009/how-to-get-diagonal-line-of-a-cube-in-a-point-cloud>, 2019.
- [15] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, and D Thomas. Manifiesto por el desarrollo ágil de software. 2001.
- [16] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Real-time multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.
- [18] Cinematic Captures. I'm han solo remastered — kinect dance in battlefield ii. <https://www.youtube.com/watch?v=ymV-R5vAuQI>, 2018.
- [19] Dante Carrizo and Andres Alfaro. Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare. Revista chilena de ingeniería*, 26(1):114–129, 2018.

- [20] X. Chang, Z. Ma, M. Lin, Y. Yang, and A. G. Hauptmann. Feature interaction augmented sparse learning for fast kinect motion detection. *IEEE Transactions on Image Processing*, 26(8):3911–3920, 2017.
- [21] N. George H. Peter A. Barbara L. Fuzhong Chen P., L. Mao. Coronavirus disease (covid-19): The need to maintain regular physical activity while taking precautions. *Journal of sport and health science*, 9(2):103–104, February 2020.
- [22] Christensson. Hardware definition. <https://techterms.com/definition/hardware>, 2020.
- [23] Andy J Daly-Smith, Stephen Zwolinsky, Jim McKenna, Phillip D Tomporowski, Margaret Anne Defeyter, and Andrew Manley. Systematic review of acute physically active learning and classroom movement breaks on children’s physical activity, cognition, academic performance and classroom behaviour: understanding critical design features. *BMJ open sport & exercise medicine*, 4(1), 2018.
- [24] Dimiter Dobrev. A definition of artificial intelligence. *arXiv preprint arXiv:1210.1568*, 2012.
- [25] Dominic. Kinect como profesor de educación física: Vicent gadea, profesores innovadores. <https://tecnousuariostv.wordpress.com/2016/01/13/kinect-como-profesor-de-educacion-fisica-vicent-gadea-profesores-innovadores/>, 2016.
- [26] Stephen I Gallant and Stephen I Gallant. *Neural network learning and expert systems*. MIT press, 1993.
- [27] Julen Zaballa García. Euskadi será el escenario de la presentación en primicia de kinect. <https://www.videojuegosvascos.com/eventos-de-videojuegos/euskadi-primicia-kinect/>, 2010.
- [28] Daniel Grest, Jan Woetzel, and Reinhard Koch. Nonlinear body pose estimation from depth images. In *Joint Pattern Recognition Symposium*, pages 285–292. Springer, 2005.

- [29] Nathan B Herz, Shyamal H Mehta, Kapil D Sethi, Paula Jackson, Patricia Hall, and John C Morgan. Nintendo wii rehabilitation (“wii-hab”) provides benefits in parkinson’s disease. *Parkinsonism & related disorders*, 19(11):1039–1042, 2013.
- [30] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014.
- [31] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.
- [32] Belinda Lange, A Rizzo, Chien-Yen Chang, Evan A Suma, and Mark Bolas. Markerless full body tracking: Depth-sensing technology within virtual environments. In *Interservice/industry training, simulation, and education conference (I/ITSEC)*, 2011.
- [33] Belinda Lange, Evan A Suma, Brad Newman, Thai Phan, Chien-Yen Chang, Albert Rizzo, and Mark Bolas. Leveraging unencumbered full body control of animated virtual characters for game-based rehabilitation. In *International Conference on Virtual and Mixed Reality*, pages 243–252. Springer, 2011.
- [34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [35] Krystof Litomisky. Consumer rgb-d cameras and their applications. *Rapport technique, University of California*, 20, 2012.
- [36] Marcelo Lopez. Metodología scrum.
- [37] Ricky J Madsen, Jon A Holmes, and Trygve D Peterson. Depth-of-field indicator for a camera, June 27 2000. US Patent 6,081,670.

- [38] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning: definitions, methods, and applications. *arXiv preprint arXiv:1901.04592*, 2019.
- [39] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.
- [40] D Oved, I Alvarado, and A Gallo. Real-time human pose estimation in the browser with tensorflow. js. *TensorFlow Medium, May*, 2018.
- [41] Bruce Pirie. Meaning through motion: Kinesthetic english. *The English Journal*, 84(8):46–51, 1995.
- [42] Gaetano Raiola, Pio Alfredo Di Tore, et al. Motor learning in sports science: Different theoretical frameworks for different teaching methods. *Sport Science*, 10(S1):50–56, 2017.
- [43] Debbie Rand, Rachel Kizony, and Patrice Tamar L Weiss. The sony playstation ii eyetoy: low-cost virtual reality for use in rehabilitation. *Journal of neurologic physical therapy*, 32(4):155–163, 2008.
- [44] Nishant Ravikumar, Ali Gooya, Alejandro F Frangi, and Zeike A Taylor. Generalised coherent point drift for group-wise registration of multi-dimensional point sets. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 309–316. Springer, 2017.
- [45] Margaret Rouse. point cloud. <https://whatis.techtarget.com/definition/point-cloud>, 2020.
- [46] Pulkit Sharma. Image classification vs. object detection vs. image segmentation. <https://medium.com/analytics-vidhya/image-classification-vs-object-detection-vs-image-segmentation-f36db85fe81>, 2019.

- [47] Matheen Siddiqui and Gérard Medioni. Human pose estimation from a single view point, real-time range sensor. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 1–8. IEEE, 2010.
- [48] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.
- [49] Sun-Chong Wang. Artificial neural network. In *Interdisciplinary computing in java programming*, pages 81–100. Springer, 2003.
- [50] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.

Anexos

12.1. Lista de Abreviaturas

| | |
|-------------------|--|
| API | Application Programming Interface |
| Cmake | Cross Platform Make |
| CUDA | Compute Unified Device Architecture |
| DLL | Dynamic Link Library |
| FPS | Frame per Second |
| MPEG-4/mp4 | Moving Picture Experts Group version 4 |
| RAM | Random Access Memory |
| GPU | Graphic Processing Unit |
| GUI | Graphical User Interface |
| UI | User Interface |
| 3D | 3 Dimensiones |
| CPU | Central Processing Unit |
| JPG-JPEG | Joint Photographic Experts Group |
| JSON | JavaScript Object Notation |
| SDK | Software development kit |
| txt | Plain Text |

12.2. Glosario

Algoritmo: Un algoritmo se define como un conjunto de órdenes definidas y finitas con el objetivo de llegar a una solución a un problema, la realización de un cómputo, proceso de datos u otras actividades. El estudio de los algoritmos busca una constante mejora de la resolución de problemas empleando métodos distintos, con el fin de optimizar los recursos necesarios, así como una solución estandarizada y apropiada para un problema general.

API: Una API es una Interfaz de Programación de Aplicaciones, que es un conjunto de protocolos y definiciones empleada en el desarrollo e integración de Software que se emplea como una biblioteca.

Animación: La animación es el acto de producir imágenes en movimiento; técnica que significa que provee de movimiento a una grabación o una serie de dibujos (Oxford English Dictionary).

Código: El código en el contexto, se define como el conjunto de símbolos y signos que se transmiten de un emisor a un receptor, cuya intención es privar del conocimiento de su contenido a terceros. Un código tiene un conjunto de reglas o normas que se emplean para dificultar la lectura del mensaje, siendo un ejemplo la traducción de un mensaje en español a inglés, el mensaje puede ser solo entendido por quien conozca el idioma inglés.

Cmake: Es una herramienta multiplataforma de automatización o generación de código.

CPU: La unidad central de procesamiento, es un Hardware requerido por una computadora o dispositivo Von Newmann, se encarga interpretar las órdenes de un Software o programa mediante operaciones aritméticas, lógicas y otros.

CUDA: Cuda se emplea para referencia una plataforma de computación en paralelo, con compilador y herramientas de desarrollo creado por la compañía nVidia.

Dataset: Un dataset es una tabla dentro de una base de datos, en la cual se representa una variable particular en la primera columna, y cada fila es un componente del conjunto de datos.

DLL Una biblioteca de enlace dinámico es un archivo con código ejecutable para poder utilizar programas por el sistema operativo.

Decodificador: El decodificador es una herramienta que permite descifrar un mensaje codificado. Se refiere a la actividad inversa de la codificación, implica la recepción de un código y de acuerdo al conjunto de normas establecidas para su traducción, se lo interpreta y se obtiene el mensaje inicial que se deseaba recibir.

Frame: Las grabaciones o serie de dibujos están compuestas por cuadros o Frames en inglés, que son una imagen individual de la secuencia de imágenes que componen la animación para dar la sensación de movimiento. La mayoría de las animaciones emplean de 24 a 30 Frames per second (FPS).

Framework: El Framework es toda estructura conceptual y tecnológica de soporte definido, con artefactos o módulos de Software que sirven para la el desarrollo y organización de Software.

Hardware: El Hardware se entiende como el componente físico de una computadora o dispositivos de Von Neumann. Esta compuesto por el procesador, la tarjeta madre, RAM, disco duro, monitor, dispositivos de entrada y salida, los cuales normalmente se mencionan como componentes y a los dispositivos externos como periféricos[22].

GPU: La unidad de procesamiento gráfico es un procesador suplementario al central, aligera la carga para trabajar con gráficos u operaciones de tipo flotante, normalmente usado en aplicaciones 3D interactivas.

GUI: Una interfaz gráfica de usuario es una interfaz diseñada con imágenes y gráficos para la representación de la información y controles disponibles en la interfaz del usuario.

UI: La interfaz puede ser usada tanto en un contexto de Hardware o de usuario, siendo la segunda la definición relevante. Una interfaz de usuario tiene la función de otorgar al usuario el control de una aplicación de Software o un dispositivo de Hardware. Uno de los objetivos de la UI es ser amigable con el usuario, permitiendo interactuar con la aplicación de manera intuitiva y natural. Ejemplos de UI pueden ser la pantalla de inicio de una página web y de Hardware un control remoto de televisión.

JPG: Es un formato estándar de codificación y compresión de imágenes y archivos

estáticas. JPG es una abreviación de Joint Photographic Experts Group (JPEG), comité que creo dicho formato.

JSON: El JavaScript Object Notation es un formato ligero e independiente de lenguajes para la transmisión de datos, es de simple lectura y fácil para el procesamiento de la máquina.

MPEG-4/mp4: Es conocido como un método de compresión digital de audio y vídeo, es el estándar para audio y vídeo relacionado a la organización internacional de estandarización.

Software: El software esta construido para ejecutar dispositivos von Neumann multipropósito. Un software contienen secuencias de declaraciones de programas abstractos que describen las tareas que deben ser realizadas por una maquina.

Modelo de Von Neumann: Es una arquitectura de computadoras, su objetivo es explicar como funciona una computadora y la interacción entre sus distintos componentes (CPU, RAM y otros).

Interprete: El interprete en el contexto informático es un programa traductor que permite ejecutar otros programas a lenguaje máquina, a diferencia de los compiladores, estos solo ejecutan la sección que requieren emplear.

Pixel: El pixel es la unidad mínima homogénea en color que en su conjunto da forma a una imagen digital.

Plug-in: Un plug-in o complemento es un Software o Aplicación con un grupo de características y funciones específicas para interactuar por medio de una API.

RAM: La memoria de acceso aleatoria se emplea para guardar los datos de trabajo temporal de la computadora y demás para el sistema operativo, el Software y los programas que se ejecutan, la RAM carga con todas las instrucciones que ejecuta el CPU para trabajar con los distintos programas.

Shell: Shell es un interprete de comandos para acceder y emplear los servicios del sistema operativo, son necesarios para ejecutar los programas de una computadora.

Script: El script es una secuencia de comandos, estos son ejecutados primero por un interprete que lee el código fuente o una consola interactiva para que el usuario trabaje. Se los utiliza para automatizar tareas repetitivas, interactuar con el sistema operativo

y otros.

SDK: El kit de desarrollo de Software es un conjunto de herramientas para el desarrollo de Software, permite la creación de una aplicación para sistemas concretos, es una API creada para poder usar ciertos lenguajes de programación y en algunos casos, comunicarse con ciertos sistemas embebidos, en algunos casos, son capaces de detectar errores y aportar utilidades extra.

txt: El txt es el formato de los documentos de texto plano.

12.3. Anexo: Control de Sprint

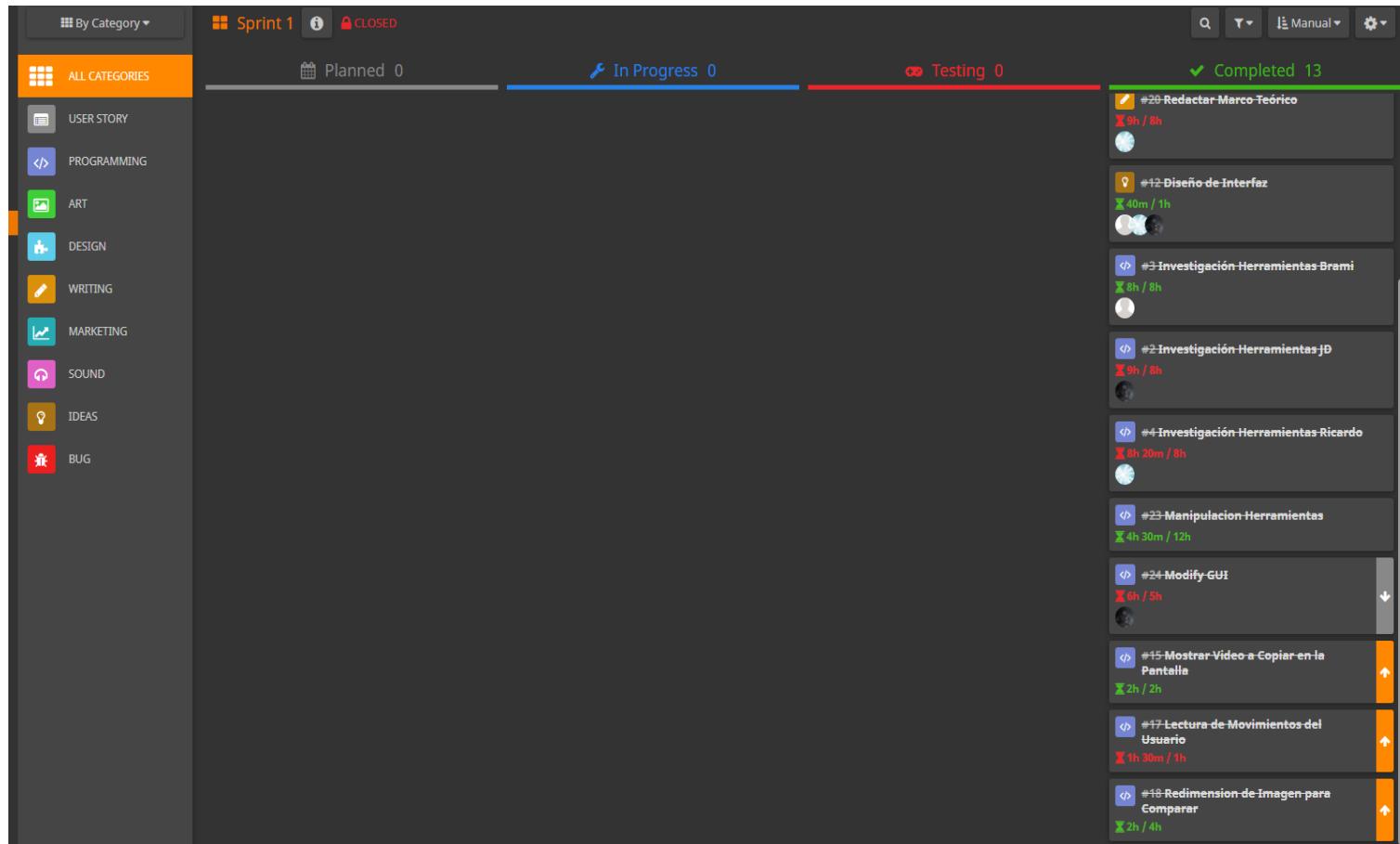


Figura 12.1: Sprint 1 concluida

Fuente: Elaboración Propia empleando la herramienta HacknPlan.

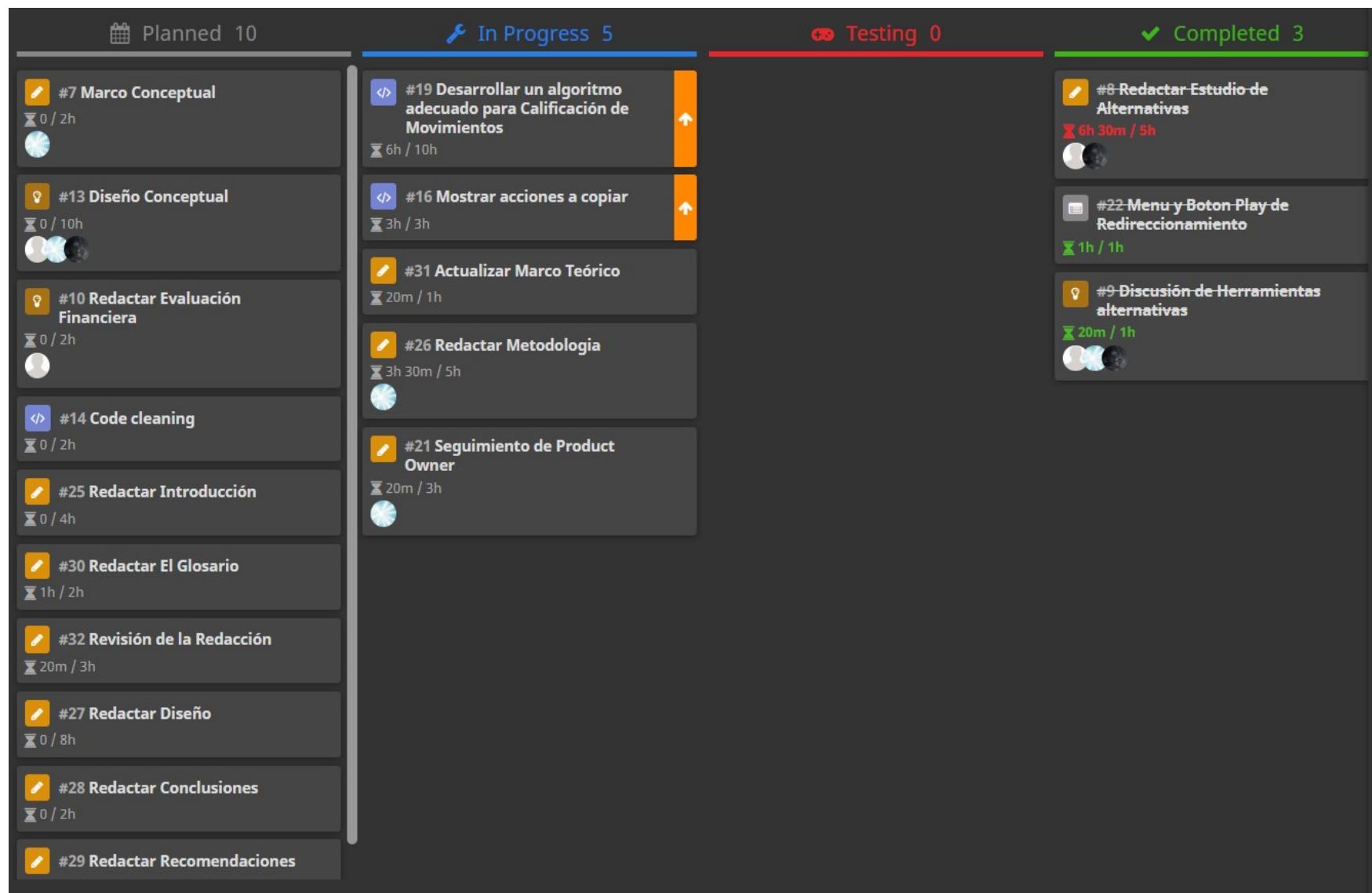


Figura 12.2: Sprint Backlog del día 5 de Noviembre
Fuente: Elaboración Propia empleando la herramienta HacknPlan.

The screenshot shows a task card from the HacknPlan application. The task is titled "#16 | Mostrar acciones a copiar". It is categorized under "Programming" and "Sprint 2", and is currently "In progress".

Importance: High

Estimated cost: 3h (3h logged, 100%)

Description:

El video a mostrar en la pantalla, tiene frames que se tienen que copiar por la persona, estos se muestran en la barra inferior debajo de la imagen
Es un grupo de poses necesarias para poder ser interpretadas por el usuario a copiar.

Comments:

- Brami Prudencio logged 1h 30m of work Dec 5, 2020 4:33:10 PM
esqueletos guardados y se pueden mostrar
- Juan Diego Garcia Vargas logged 1h 30m of work Dec 4, 2020 9:16:18 AM
Se intento mostrar una sucesion de imagenes de la misma carpeta de mapa, no hubo mucho exito aun

Work Logs:

- Brami Prudencio 1h 30m (50.0%)
- Juan Diego Garcia Vargas 1h 30m (50.0%)

Figura 12.3: Tarea de la Sprint 2, Mostrar Acciones a Copiar
Fuente: Elaboración Propia empleando la herramienta HacknPlan.

12.4. Anexo: Product Backlog

En la tabla 12.1 del Product Backlog, se observan los campos:

- ID como identificador de la tarea, debe ser único.
- Tarea, que indica el título dado en Sprint Backlog para facilitar su comprensión.
- Estado(Sprint), si ha de estar Planificada, En Proceso o Hecho (Sprint en que se completo) o si fue Descartado.
- Valor, el valor designado a cada actividad realizada, en total se asignan 1000 puntos a su finalización
- Estimado Inicial, la valoración de las tareas que se mide en horas requeridas.
- Factor Ajuste, determina el fallo que se debe aumentar al tiempo original, se le suma al Valor Estimado en función $ValorEstimado = ValorEstimado + ValorEstimado * FactordeAjuste$
- Ajustado, es la valoración de las tareas posteriormente al ser más realistas sobre el tiempo requerido para resolverlo.
- Sprint, que señala el número de Sprint en que fue realizada
- Prioridad, al ser una metodología ágil, se reconoce que las tareas pueden ser más relevantes unas que otras, en este caso, Baja, Normal y Alta.

Como métrica para calcular el valor se empleará el Estimado Inicial y la Prioridad, se utilizarán cálculos matemáticos, donde el valor total a sumar para el resto de las tareas es 500, se asignara el valor de las prioridades como multiplicadores $Baja = 1$, $Normal = 1,5$, $Alta = 2$ a las horas de la Estimación inicial EI , de tal forma que la sumatoria de $EI * Prioridad$ se divida en 500 y se asigne los valores correspondientes.

La Tarea es el nombre dado a las tareas distribuidas al SCRUM Team, formuladas a partir de los requerimientos del cliente. El valor de las tareas se distribuirá de un valor total de 1000 puntos, se determinara una cuarta parte a la redacción del informe, una

cuarta parte a la investigación dedicada requerido para poder hacer posible el proyecto y el resto lo determinará el rol de Product Owner. El Estimado Inicial es definido con la Estimación de Poker previamente mencionada. Redactar en la tabla 12.1 engloba la redacción del informe y el tiempo estimado de cada tarea de redacción, el Redactar hace referencia a los títulos de cada parte del informe. En Redactar Parte 1, están los títulos de Objetivos, Estudio de Diagnóstico, Marco Teórico, Estudio de Alternativas, Plan de Actividades En Redactar Parte 2, están los títulos de Metodología, Diseño, Conclusiones, Recomendaciones, dentro de la metodología se lleva a cabo la tarea de Seguimiento de Product Owner. En Redactar Parte 3, están los títulos de Introducción, Anexos, Glosario, Revisión de ortografía y Resumen.

Al existir potenciales nuevas tareas, el valor puede variar en la siguiente entrega. Para simplificar la tabla, el proceso de corrección de errores es parte de la tarea correspondiente.

| Product Backlog | | | | | | | | | |
|-----------------|--|----------------|-------|----------------------|---------------|-------------|--------|-----------|--|
| ID | Tarea | Estado(Sprint) | Valor | Estimado Inicial (h) | Factor Ajuste | Ajustado(h) | Sprint | Prioridad | |
| T-0 | Redactar Parte 1 | Hecho(1) | 50 | 18 | 1.5 | 45 | 1 | Normal | |
| T-1 | Redactar Parte 2 | Hecho(2) | 100 | 20 | 2.0 | 60 | 2 | Normal | |
| T-1 | Redactar Parte 3 | Hecho(3) | 100 | 9 | 1.0 | 18 | 3 | Normal | |
| T-2 | Investigación de Herramientas | Hecho(2) | 250 | 40 | 3.0 | 160 | 1 | Normal | |
| T-3 | Manipulación de Herramientas | Hecho(3) | 70 | 12 | 4.5 | 66 | 1 | Normal | |
| T-4 | Modificación de GUI | Hecho(1) | 10 | 3 | 1.0 | 6 | 1 | Baja | |
| T-5 | Mostrar Vídeo a Copiar en la Pantalla | Hecho(1) | 15 | 2 | 0.0 | 2 | 1 | Alta | |
| T-6 | Lectura de Movimientos del Usuario | Hecho(1) | 8 | 1 | 4.0 | 5 | 1 | Alta | |
| T-7 | Redimension de Imagen para Comparar | Hecho(1) | 30 | 4 | 0.0 | 4 | 1 | Alta | |
| T-8 | Menú y Botón Play de Redirecciónamiento | Hecho(2) | 4 | 1 | 0.0 | 1 | 2 | Baja | |
| T-9 | Calificación de Movimientos | Hecho(2) | 75 | 10 | 4.0 | 50 | 2 | Alta | |
| T-10 | Mostrar acciones a imitar | Hecho(2) | 24 | 3 | 9.0 | 30 | 2 | Alta | |
| T-11 | Asignar tiempo a las poses a imitar | Hecho(2) | 30 | 5 | 3.0 | 20 | 2 | Normal | |
| T-12 | Diseño de GUI | Hecho(2) | 60 | 10 | 0.0 | 10 | 2 | Normal | |
| T-13 | Creación de menú y visuales | Hecho(2) | 60 | 10 | 0.0 | 10 | 2 | Normal | |
| T-14 | Funcionalidad de menú y menú de opciones | Hecho(2) | 12 | 2 | 0.5 | 3 | 2 | Normal | |
| T-15 | Menú de lista de Mapas | Hecho(2) | 12 | 2 | 2.0 | 6 | 2 | Normal | |
| T-16 | Modificación de UML | Hecho(2) | 4 | 1 | 0.0 | 1 | 2 | Baja | |
| T-17 | Creación de Nivel a través de GUI | Hecho(3) | 13 | 4 | 2.0 | 12 | 3 | Normal | |
| T-18 | Elección de Grabación o Vídeo | Hecho(3) | 13 | 4 | 1.0 | 8 | 3 | Normal | |
| T-19 | Revisión de Código | Hecho(3) | 24 | 4 | 0.0 | 4 | 3 | Normal | |
| T-20 | Revisión del Informe | Hecho(3) | 36 | 6 | 0.0 | 6 | 3 | Normal | |
| | Entrega final | | 1000 | 171 | 2.082 | 527 | | | |

Cuadro 12.1: Product Backlog

Fuente: Elaboración propia

12.5. Anexo: Gráfico Burn-Up Burn-Down

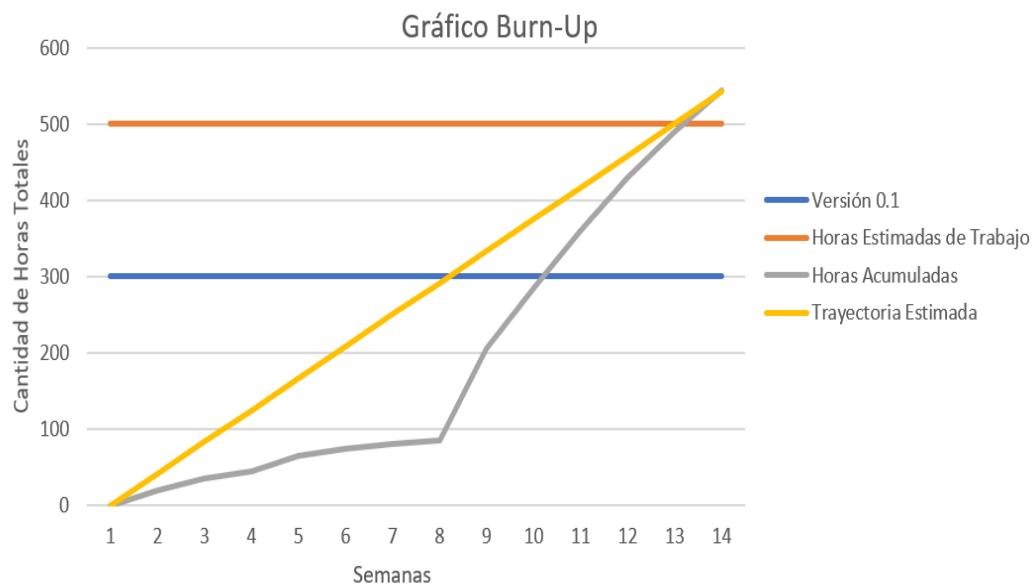


Figura 12.4: Gráfico Burn-Up
Fuente: Elaboración Propia

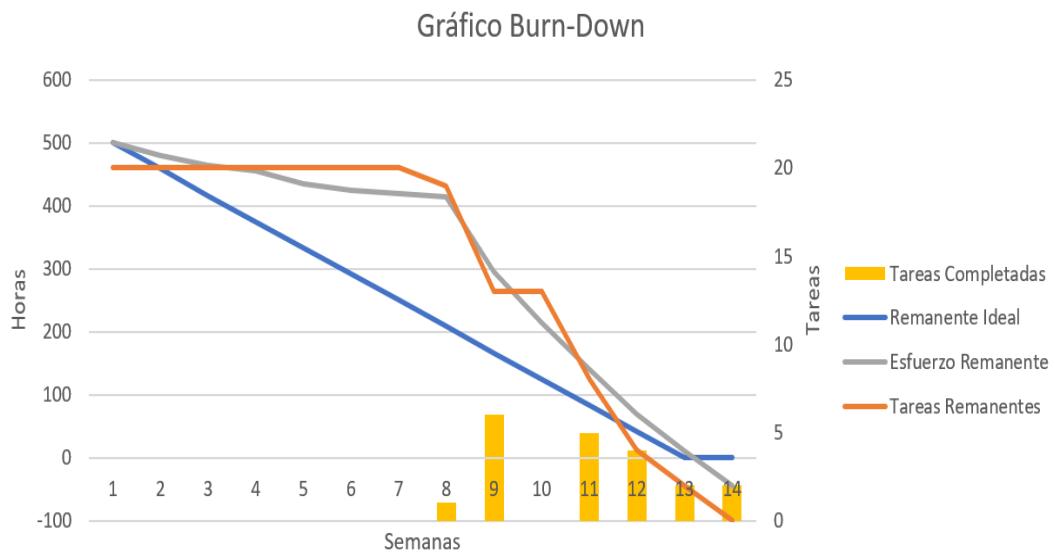


Figura 12.5: Gráfico Burn-Down
Fuente: Elaboración Propia

12.6. Anexo: Proceso de Instalación de OpenPose y Unity para el Desarrollo del Proyecto

Inicialmente se deben instalar la SDK de Visual Studio 2019 , de la pagina oficial visualstudio.microsoft.com, se ejecuta el archivo, se selecciona el paquete de idioma deseado y se lo descarga.

Posteriormente, se debe seleccionar la versión deseada, en este caso se trabajo con “Visual Studio Community 2019”, se deben habilitar las opciones principales de “Desarrollo de .NET”, “Desarrollo para el escritorio con C++”, “Desarrollo de juego con Unity”, para realizar este proyecto. Se tiene que habilitar todas las opciones relacionadas con C++ y las Build Tools VS 2015-2017 en “Desarrollo para el escritorio con C++” como se observa en la figura 12.6 de Anexo: Proceso de Instalación de OpenPose y Unity para el Desarrollo del Proyecto, ya que OpenPose las utiliza, finalmente se lo debe instalar en el lugar deseado y ejecutar.

La instalación de Unity inicia vistiando su pagina oficial de descargas unity3d.com/es/get-unity/download, se selecciona la opción de Descarga Unity Hub, se selecciono la opción “Personalz de “Usuario recurrente”, finalmente se aceptan los términos y condiciones, para descargar el instalador, se lo ejecutara y configurara como se desee, se debe ejecutar el programa instalado “Unity Hub”.

En la página oficial de versiones existentes unity3d.com/get-unity/download/archive, se seleccionara la versión Unity 2019.4.14 y se selecciona la opción “Unity Hub”, finalmente se acepta instalar la versión correcta, si se siguieron correctamente los pasos, se tendrá la instalación de la figura 12.7 de Anexo: Proceso de Instalación de OpenPose y Unity para el Desarrollo del Proyecto al finalizar.

La herramienta de Body Tracking OpenPose tiene requisitos elevados para su instalación y desarrollo, si bien instalar el programa para modificar el código fuente original puede realizarse siguiendo las instrucciones de [openpose/doc/installation/README.md](https://openpose.readthedocs.io/en/latest/installation.html) guardado en el repositorio principal de GitHub (CMU-Perceptual-Computing-Lab/openpose). Sin embargo, se empleara su plug-in de Unity almacenado en el repositorio CMU-

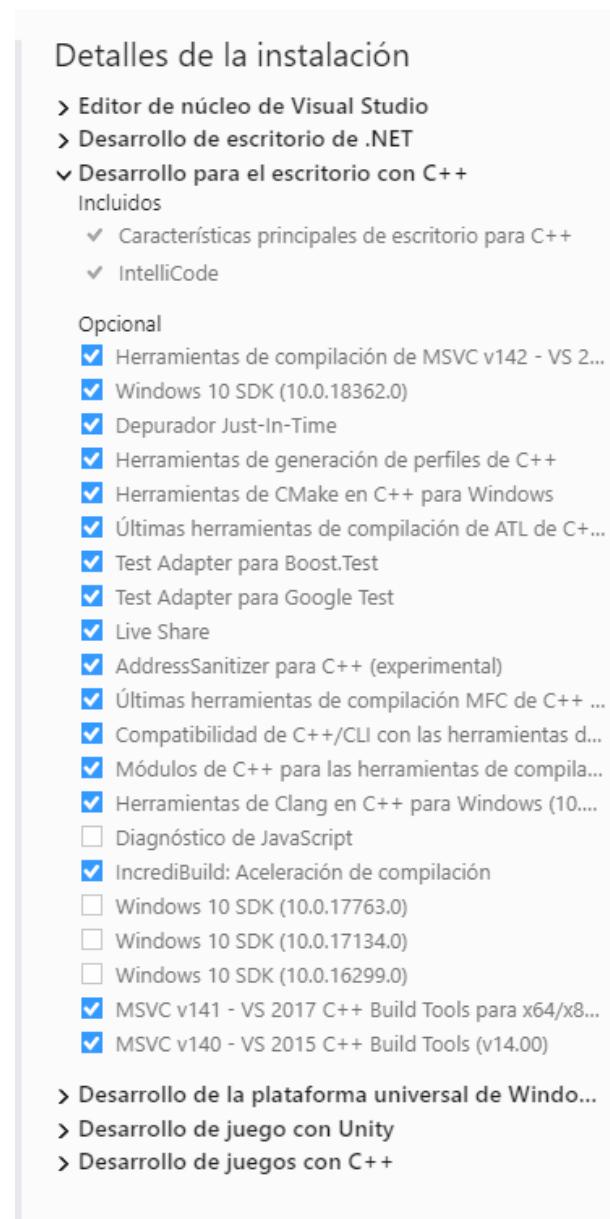


Figura 12.6: Elección Personal del Estudiante para Configurar Visual Studio 2019

Fuente: Instalador de Visual Studio.

Perceptual-Computing-Lab/openpose_unity_plugin de GitHub. Se deben realizar x pasos para instalarlo, básicamente seguir las instrucciones de su documentación de instalación. Primero descargar el repositorio de GitHub, ejecutar el archivo “getPlugins.bat”, “getModels.bat” “testBinary.bat” dentro de la carpeta, para descargar los modelos, los archivos binarios de OpenPose y una Demo que se nombrará como **OpenPoseDemo** con la cual realizar pruebas. Finalmente se abre la escena “Demo.Unity” localizada en ./OpenPosePlugin/Assets/OpenPose/Examples/Scenes/ en Unity y ejecutarla, si se

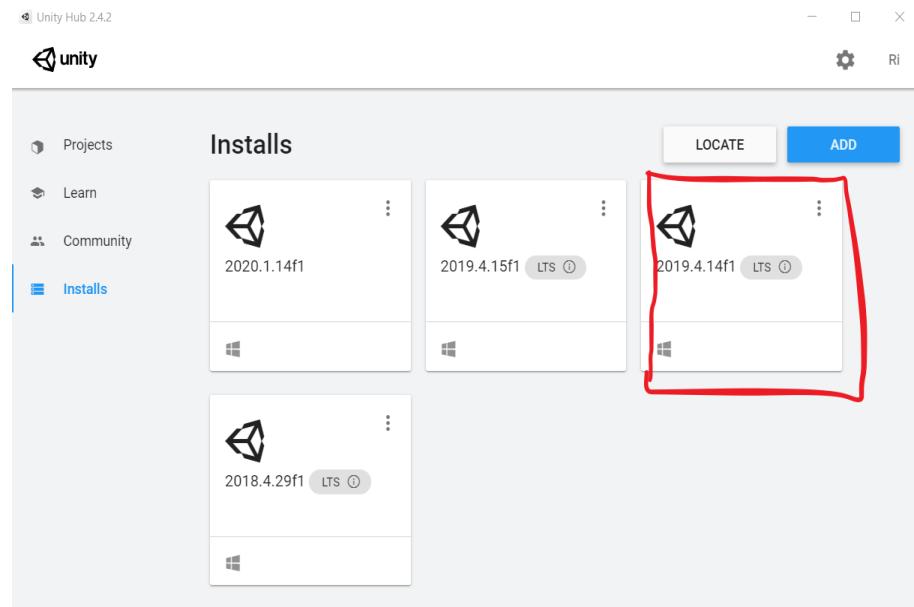


Figura 12.7: Versión Deseada de Unity Encerrada en Rojo
Fuente: Captura de Pantalla de Installs - Unity Hub.



Figura 12.8: Captura de Pantalla de la Configuración Inicial de OpenPose Plug-in
Fuente: Elaboración Propia Basado en [16]

realizo correctamente, se le pedirán aceptar dos problemas debido al versionamiento y será dirigido a la siguiente pantalla de Unity de las figuras 12.8 y 12.9.

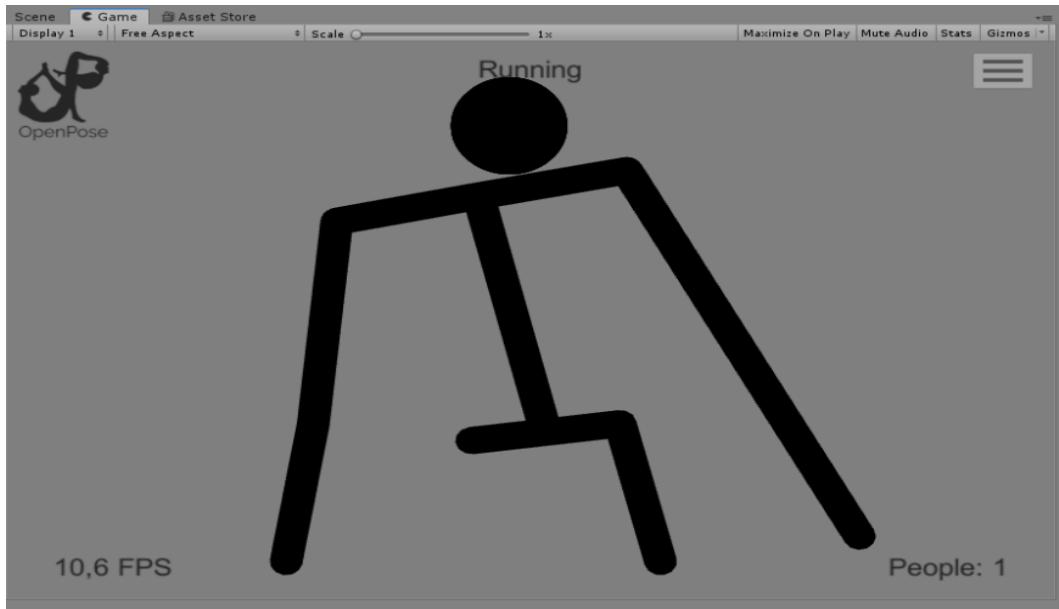


Figura 12.9: Prueba de Ejecución de Demo de OpenPose

Fuente: Elaboración Propia Basado en [16]

12.7. Anexo: Archivos Generados

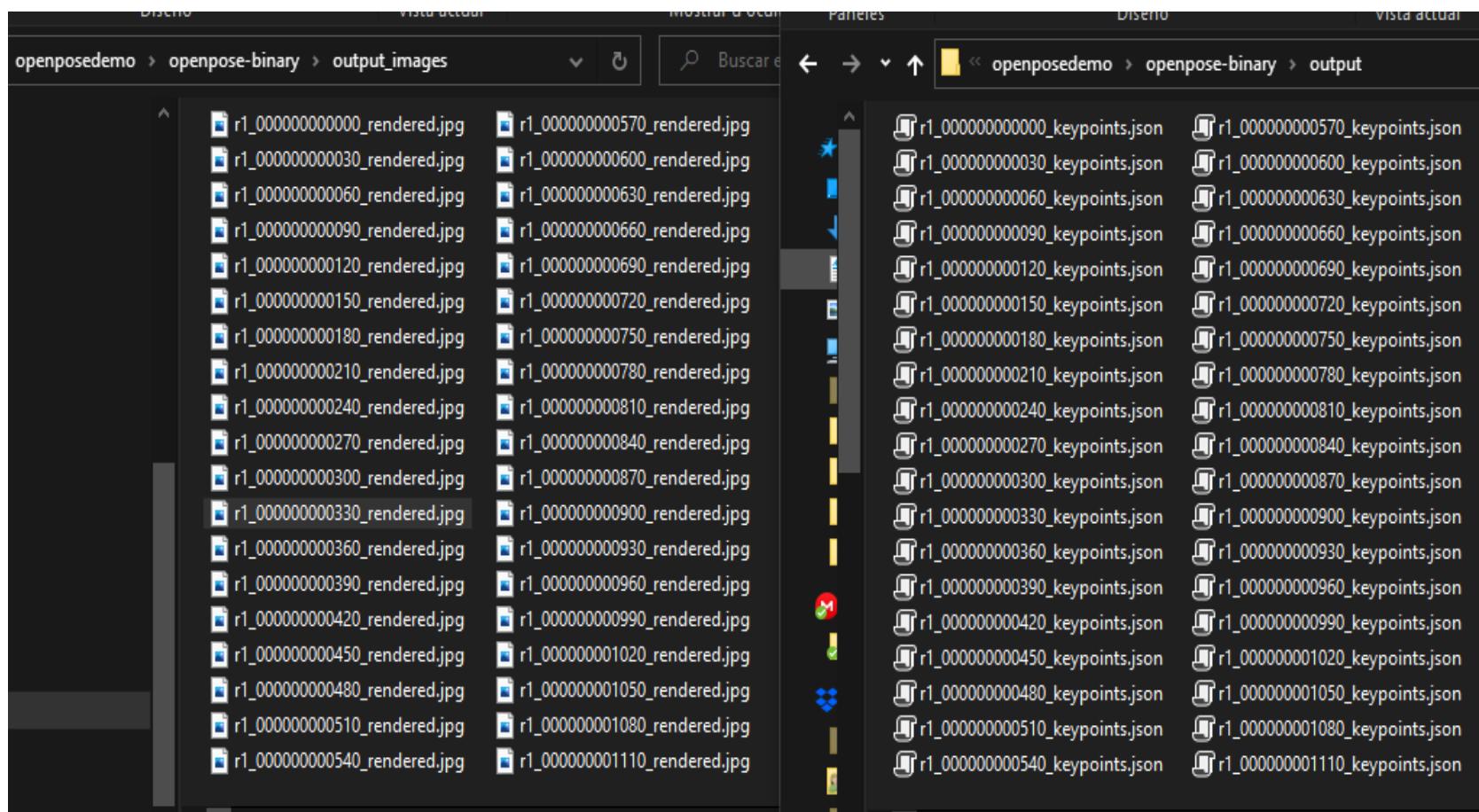


Figura 12.10: Archivos Generados por la Conversión usando el Script con OpenPoseDemo
Fuente: Elaboración Propia Basado en [16]

12.8. Anexo: Mensajes de Error

```
Starting OpenPose demo...
Configuring OpenPose...
Starting thread(s)...
Auto-detecting camera index... Detected and opened camera 0.
Auto-detecting all available GPUs... Detected 1 GPU(s), using 1 of them starting at GPU 0.
F1210 01:11:06.582840 1324 syncedmem.cpp:71] Check failed: error == cudaSuccess (2 vs. 0)  out of memory
*** Check failure stack trace: ***
```

Figura 12.11: Archivos Generados por la Conversión usando el Script con OpenPoseDemo

Fuente: Elaboración Propia Basado en [16]

12.9. Anexo: Cálculos de Comparación de Poses

12.9.1. Ejemplo de Cálculo de Normalización Basada en el Centro de la Cadera

Se tiene un vídeo grabado por un estudiante a medida que la lectura de Frames es dada, se guardaron 8 Frames selectos de manera arbitraria para la comparación de poses ilustrados en las figuras 12.13 y 12.14, los cuales tienen los puntos clave en las tablas 12.2 12.3 12.4 12.5.

Los cálculos del porcentaje de semejanza entre el T Original con cada Frame selecto esta en las tablas 12.6 12.7 12.8 12.9, los cuales se emplearan para la ecuación 8.4 dando los resultados de la tabla 12.12, donde T Original tiene un 100% de similitud y la diferencia independientemente de si es mayor o menor es la diferencia, como se puede ver solo hay un aproximado de 108% en el Frame 147 Dab, lo cual indica solo un aproximado de 8% de diferencia para poses tan distintas, convirtiendo este método de cálculo en uno muy inefectivo para poses distintas, pero bueno para poses similares en distintos lugares.

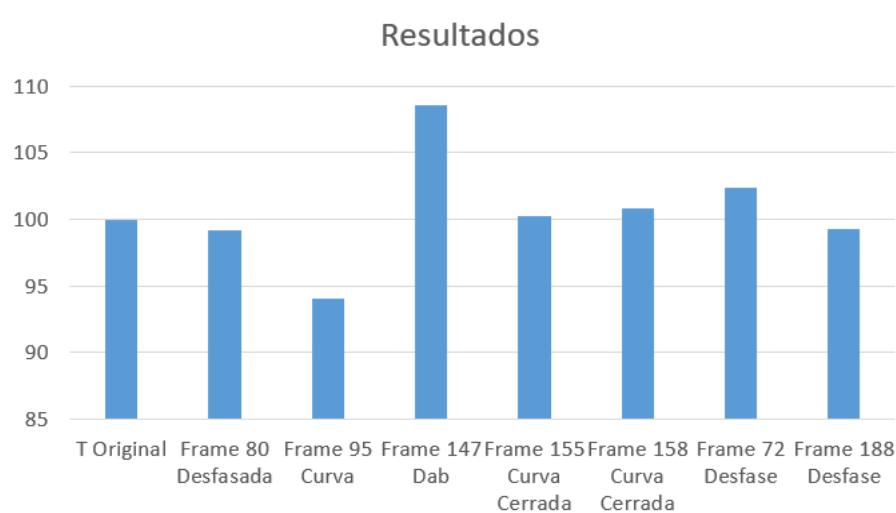


Figura 12.12: Tabla de Diferencias entre Poses
Fuente: Elaboración Propia

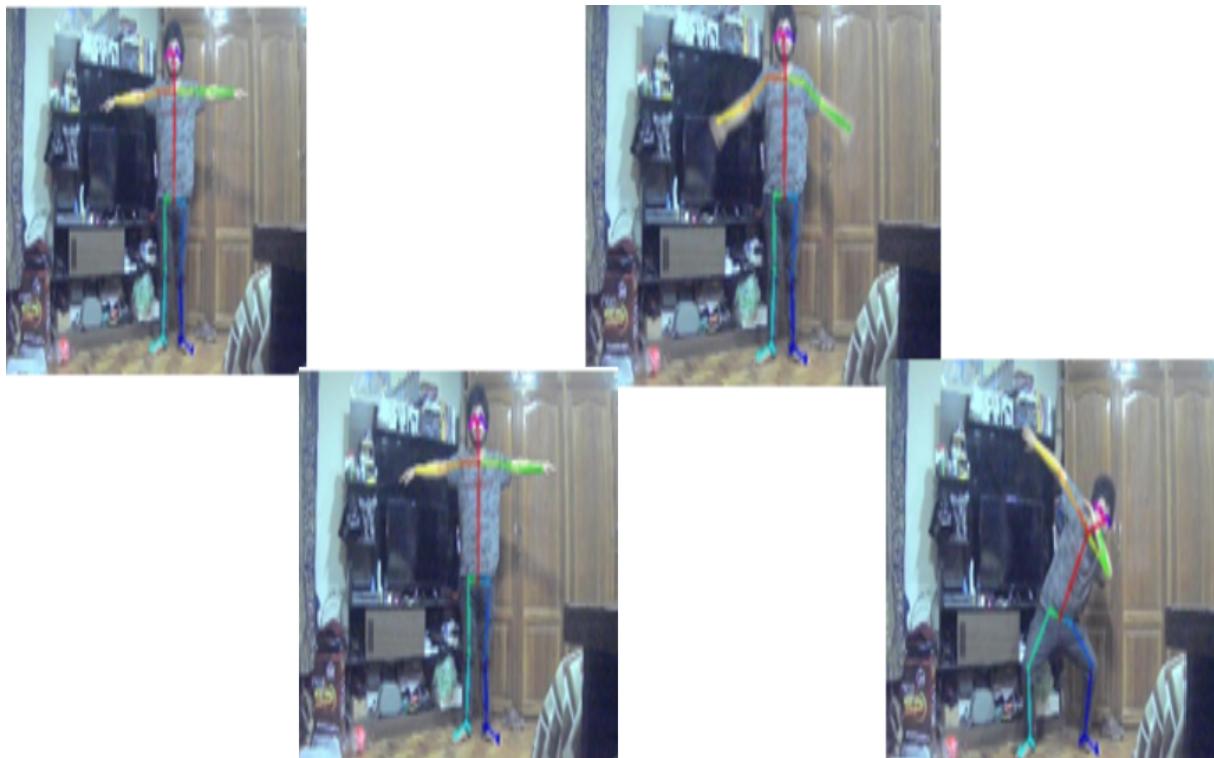


Figura 12.13: Poses: T Original - Frame 80 - Frame 95 Curva - Frame 147 Dab
Fuente: Elaboración Propia

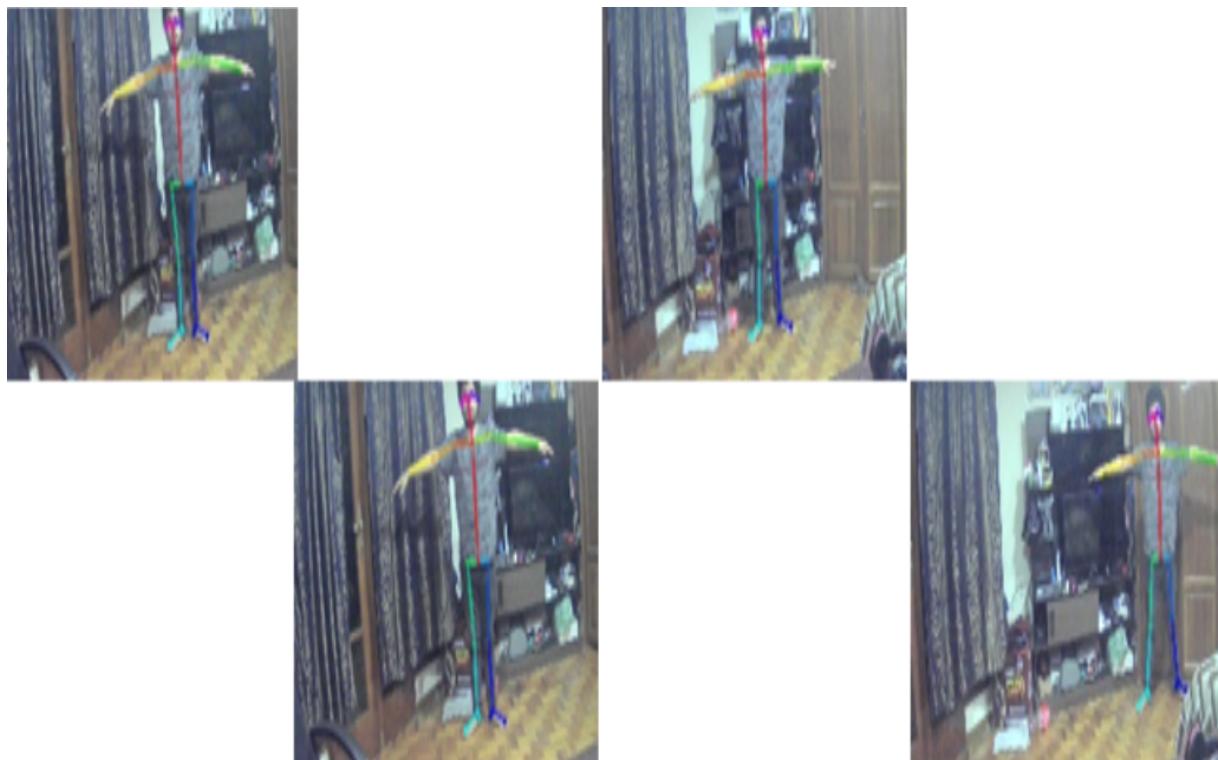


Figura 12.14: Poses: Frame 155 Curva Cerrada - Frame 158 Curva Cerrada - Frame 72 Desfase - Frame 188 Desfase

Fuente: Elaboración Propia

| Coordenadas | T Original | Frame 80 |
|---------------|---------------------|---------------------|
| (x-0 ; y-0) | (713,549 ; 94,5737) | (713,713 ; 82,6751) |
| (x-1 ; y-1) | (717,589 ; 165,122) | (719,531 ; 155,226) |
| (x-2 ; y-2) | (650,932 ; 165,16) | (651,021 ; 155,255) |
| (x-3 ; y-3) | (560,829 ; 178,768) | (560,901 ; 166,911) |
| (x-4 ; y-4) | (474,593 ; 180,826) | (476,567 ; 168,975) |
| (x-5 ; y-5) | (789,967 ; 165,069) | (790,029 ; 155,17) |
| (x-6 ; y-6) | (878,214 ; 165,17) | (880,181 ; 166,925) |
| (x-7 ; y-7) | (964,378 ; 167,014) | (964,378 ; 167,014) |
| (x-8 ; y-8) | (713,711 ; 378,593) | (715,662 ; 370,634) |
| (x-9 ; y-9) | (676,377 ; 374,707) | (678,356 ; 366,775) |
| (x-10 ; y-10) | (678,369 ; 511,872) | (680,362 ; 498,068) |
| (x-11 ; y-11) | (668,569 ; 647,009) | (668,713 ; 631,277) |
| (x-12 ; y-12) | (758,596 ; 380,551) | (758,761 ; 370,707) |
| (x-13 ; y-13) | (743,08 ; 515,78) | (745,016 ; 507,78) |
| (x-14 ; y-14) | (746,985 ; 646,956) | (748,931 ; 635,116) |
| (x-15 ; y-15) | (698,038 ; 84,7163) | (699,969 ; 72,8773) |
| (x-16 ; y-16) | (727,333 ; 84,786) | (729,288 ; 72,9199) |
| (x-17 ; y-17) | (682,286 ; 96,5211) | (684,223 ; 88,577) |
| (x-18 ; y-18) | (745,001 ; 100,351) | (746,995 ; 90,481) |
| (x-19 ; y-19) | (792,074 ; 674,425) | (795,937 ; 662,601) |
| (x-20 ; y-20) | (793,917 ; 666,537) | (795,929 ; 654,715) |
| (x-21 ; y-21) | (739,089 ; 654,823) | (741,057 ; 642,991) |
| (x-22 ; y-22) | (621,554 ; 668,531) | (627,446 ; 654,804) |
| (x-23 ; y-23) | (621,554 ; 664,564) | (625,515 ; 650,806) |
| (x-24 ; y-24) | (680,334 ; 656,804) | (682,303 ; 641,062) |

Cuadro 12.2: Puntos Clave T Original - Frame 80

Fuente: Elaboración Propia

| Coordenadas | Frame 95 Curva | Frame 147 Dab |
|---------------|---------------------|---------------------|
| (x-0 ; y-0) | (729,331 ; 4,366) | (978,154 ; 143,52) |
| (x-1 ; y-1) | (735,144 ; 78,871) | (917,42 ; 163,131) |
| (x-2 ; y-2) | (670,464 ; 76,91) | (884,023 ; 125,923) |
| (x-3 ; y-3) | (586,208 ; 127,81) | (809,644 ; 59,34) |
| (x-4 ; y-4) | (494,206 ; 153,34) | (727,409 ; 14,161) |
| (x-5 ; y-5) | (797,855 ; 84,667) | (956,564 ; 200,389) |
| (x-6 ; y-6) | (876,223 ; 131,712) | (993,837 ; 231,704) |
| (x-7 ; y-7) | (964,378 ; 167,014) | (964,378 ; 167,014) |
| (x-8 ; y-8) | (731,243 ; 290,389) | (813,554 ; 309,996) |
| (x-9 ; y-9) | (694,018 ; 288,428) | (774,314 ; 296,356) |
| (x-10 ; y-10) | (695,949 ; 421,716) | (694,038 ; 394,327) |
| (x-11 ; y-11) | (684,197 ; 556,86) | (699,909 ; 517,78) |
| (x-12 ; y-12) | (774,278 ; 290,431) | (854,677 ; 317,93) |
| (x-13 ; y-13) | (758,684 ; 429,497) | (933,02 ; 390,453) |
| (x-14 ; y-14) | (762,595 ; 556,875) | (905,618 ; 513,83) |
| (x-15 ; y-15) | (715,549 ; -5,509) | (968,38 ; 125,939) |
| (x-16 ; y-16) | (744,9 ; -5,479) | (991,88 ; 137,657) |
| (x-17 ; y-17) | (697,993 ; 8,293) | (944,779 ; 116,08) |
| (x-18 ; y-18) | (764,509 ; 8,270) | (999,703 ; 147,491) |
| (x-19 ; y-19) | (807,706 ; 584,274) | (952,711 ; 541,265) |
| (x-20 ; y-20) | (807,7 ; 576,385) | (950,725 ; 531,482) |
| (x-21 ; y-21) | (754,718 ; 566,639) | (890,01 ; 523,581) |
| (x-22 ; y-22) | (639,157 ; 576,525) | (662,706 ; 545,202) |
| (x-23 ; y-23) | (639,134 ; 572,548) | (662,651 ; 535,4) |
| (x-24 ; y-24) | (695,955 ; 566,639) | (711,751 ; 529,561) |

Cuadro 12.3: Puntos Clave Frame 95 Curva - Frame 147 Dab

Fuente: Elaboración Propia

| Coordenadas | Frame 155 Curva Cerrada | Frame 158 Curva Cerrada |
|---------------|-------------------------|-------------------------|
| (x-0 ; y-0) | (670,562 ; 86,7501) | (676,393 ; 86,7425) |
| (x-1 ; y-1) | (684,291 ; 163,048) | (694,002 ; 163,01) |
| (x-2 ; y-2) | (613,679 ; 167,065) | (623,518 ; 166,961) |
| (x-3 ; y-3) | (513,797 ; 194,375) | (523,567 ; 194,406) |
| (x-4 ; y-4) | (423,646 ; 215,992) | (433,449 ; 216,033) |
| (x-5 ; y-5) | (760,664 ; 151,364) | (766,482 ; 151,366) |
| (x-6 ; y-6) | (858,594 ; 153,373) | (860,57 ; 159,17) |
| (x-7 ; y-7) | (964,378 ; 167,014) | (964,378 ; 167,014) |
| (x-8 ; y-8) | (711,69 ; 388,437) | (717,46 ; 386,404) |
| (x-9 ; y-9) | (666,692 ; 392,345) | (672,49 ; 386,444) |
| (x-10 ; y-10) | (697,925 ; 541,129) | (701,895 ; 539,219) |
| (x-11 ; y-11) | (709,668 ; 666,599) | (713,637 ; 666,601) |
| (x-12 ; y-12) | (756,698 ; 386,451) | (758,674 ; 384,454) |
| (x-13 ; y-13) | (756,694 ; 527,559) | (760,642 ; 527,511) |
| (x-14 ; y-14) | (774,31 ; 664,657) | (780,161 ; 662,628) |
| (x-15 ; y-15) | (654,89 ; 74,965) | (660,731 ; 76,8259) |
| (x-16 ; y-16) | (684,283 ; 72,9228) | (690,103 ; 72,9345) |
| (x-17 ; y-17) | (641,118 ; 90,5857) | (648,935 ; 92,531) |
| (x-18 ; y-18) | (713,559 ; 84,673) | (719,425 ; 84,6355) |
| (x-19 ; y-19) | (823,294 ; 684,268) | (829,214 ; 684,199) |
| (x-20 ; y-20) | (821,401 ; 680,299) | (829,154 ; 682,134) |
| (x-21 ; y-21) | (760,668 ; 678,351) | (766,602 ; 674,4) |
| (x-22 ; y-22) | (658,782 ; 701,867) | (666,562 ; 701,793) |
| (x-23 ; y-23) | (656,829 ; 697,864) | (662,708 ; 697,885) |
| (x-24 ; y-24) | (717,557 ; 676,393) | (723,407 ; 678,336) |

Cuadro 12.4: Puntos Clave Frame 155 Curva Cerrada - Frame 158 Curva Cerrada

Fuente: Elaboración Propia

| Coordenadas | Frame 72 Desfase | Frame 188 Desfase |
|---------------|----------------------|---------------------|
| (x-0 ; y-0) | (709,715 ; 116,0957) | (723,358 ; 92,6803) |
| (x-1 ; y-1) | (725,312 ; 182,622) | (735,158 ; 155,333) |
| (x-2 ; y-2) | (652,882 ; 182,776) | (678,293 ; 157,189) |
| (x-3 ; y-3) | (566,648 ; 198,3) | (590,192 ; 171,032) |
| (x-4 ; y-4) | (472,617 ; 212,045) | (500,019 ; 190,572) |
| (x-5 ; y-5) | (790,104 ; 172,949) | (793,928 ; 155,249) |
| (x-6 ; y-6) | (884,03 ; 166,975) | (876,208 ; 157,253) |
| (x-7 ; y-7) | (964,378 ; 167,014) | (964,378 ; 167,014) |
| (x-8 ; y-8) | (731,241 ; 386,44) | (745,008 ; 343,418) |
| (x-9 ; y-9) | (693,978 ; 388,373) | (711,593 ; 345,332) |
| (x-10 ; y-10) | (705,736 ; 523,581) | (701,793 ; 466,829) |
| (x-11 ; y-11) | (703,793 ; 645,044) | (690,04 ; 580,44) |
| (x-12 ; y-12) | (772,465 ; 386,403) | (780,188 ; 343,364) |
| (x-13 ; y-13) | (776,286 ; 513,813) | (791,898 ; 453,032) |
| (x-14 ; y-14) | (792,027 ; 635,251) | (813,498 ; 562,764) |
| (x-15 ; y-15) | (695,897 ; 104,3542) | (711,638 ; 84,7439) |
| (x-16 ; y-16) | (723,392 ; 102,3882) | (731,278 ; 79,0083) |
| (x-17 ; y-17) | (678,405 ; 114,1074) | (697,914 ; 94,634) |
| (x-18 ; y-18) | (743,027 ; 106,3588) | (750,838 ; 90,6595) |
| (x-19 ; y-19) | (837,053 ; 656,843) | (842,898 ; 582,447) |
| (x-20 ; y-20) | (838,965 ; 650,968) | (848,728 ; 578,417) |
| (x-21 ; y-21) | (786,114 ; 645,036) | (809,528 ; 574,547) |
| (x-22 ; y-22) | (676,332 ; 674,482) | (654,837 ; 611,793) |
| (x-23 ; y-23) | (666,642 ; 668,613) | (652,82 ; 605,869) |
| (x-24 ; y-24) | (711,659 ; 654,798) | (697,94 ; 594,068) |

Cuadro 12.5: Puntos Clave Frame 72 Desfase - Frame 188 Desfase
 Fuente: Elaboración Propia

| Coordenadas | T Original | Frame 80 Desfasada |
|---------------|-------------|---------------------------------------|
| (x-0 ; y-0) | (100 ; 100) | (100,022983705394 ; 87,4187009707773) |
| (x-1 ; y-1) | (100 ; 100) | (100,270628451662 ; 94,006855537118) |
| (x-2 ; y-2) | (100 ; 100) | (100,013672703139 ; 94,0027851780092) |
| (x-3 ; y-3) | (100 ; 100) | (100,012838137828 ; 93,3673811867896) |
| (x-4 ; y-4) | (100 ; 100) | (100,415935338279 ; 93,4461858361077) |
| (x-5 ; y-5) | (100 ; 100) | (100,007848429112 ; 94,00311384936) |
| (x-6 ; y-6) | (100 ; 100) | (100,223977299383 ; 101,062541623782) |
| (x-7 ; y-7) | (100 ; 100) | (100 ; 100) |
| (x-8 ; y-8) | (100 ; 100) | (100,273359945412 ; 97,8977424305257) |
| (x-9 ; y-9) | (100 ; 100) | (100,292588305043 ; 97,8831460314326) |
| (x-10 ; y-10) | (100 ; 100) | (100,293792906221 ; 97,3032320580145) |
| (x-11 ; y-11) | (100 ; 100) | (100,021538539777 ; 97,5685036838746) |
| (x-12 ; y-12) | (100 ; 100) | (100,021750707887 ; 97,4132245086729) |
| (x-13 ; y-13) | (100 ; 100) | (100,260537223448 ; 98,4489511031835) |
| (x-14 ; y-14) | (100 ; 100) | (100,260513932676 ; 98,1698909972239) |
| (x-15 ; y-15) | (100 ; 100) | (100,276632504248 ; 86,0251214937385) |
| (x-16 ; y-16) | (100 ; 100) | (100,26879022401 ; 86,0046469936075) |
| (x-17 ; y-17) | (100 ; 100) | (100,283898541081 ; 91,7695716273437) |
| (x-18 ; y-18) | (100 ; 100) | (100,267650647449 ; 90,164522525934) |
| (x-19 ; y-19) | (100 ; 100) | (100,487706956673 ; 98,2468028320421) |
| (x-20 ; y-20) | (100 ; 100) | (100,253426995517 ; 98,2263550260526) |
| (x-21 ; y-21) | (100 ; 100) | (100,266273750523 ; 98,1930995093331) |
| (x-22 ; y-22) | (100 ; 100) | (100,947946598365 ; 97,9466920756106) |
| (x-23 ; y-23) | (100 ; 100) | (100,637273672119 ; 97,9297704961449) |
| (x-24 ; y-24) | (100 ; 100) | (100,289416668871 ; 97,6032423675861) |

Cuadro 12.6: Puntos Clave T Original - Frame 80 Desfasada

Fuente: Elaboración Propia

| Coordenadas | Frame 95 Curva | Frame 147 Dab |
|---------------|--|------------------------------------|
| (x-0 ; y-0) | (102,211761210513 ; 4,61629395910282) | (137,0847351899 ; 151,7546632341) |
| (x-1 ; y-1) | (102,446386441264 ; 47,7652886956311) | (127,8456191636 ; 98,794224871522) |
| (x-2 ; y-2) | (103,000620648547 ; 46,5669653669169) | (135,8009522346 ; 76,2430370577) |
| (x-3 ; y-3) | (104,525265276938 ; 71,4948984158239) | (144,3673106954 ; 33,19386019933) |
| (x-4 ; y-4) | (104,132593611789 ; 84,7997522480175) | (153,2765087349 ; 7,831285324316) |
| (x-5 ; y-5) | (100,998522723101 ; 51,2918840000243) | (121,0808785557 ; 121,3971127558) |
| (x-6 ; y-6) | (99,7732898815095 ; 79,743294787189) | (113,1657654558 ; 140,2821335563) |
| (x-7 ; y-7) | (100 ; 100) | (100 ; 100) |
| (x-8 ; y-8) | (102,456456464872 ; 76,7021577261069) | (113,9875771286 ; 81,88107017826) |
| (x-9 ; y-9) | (102,608160833381 ; 76,9742759009039) | (114,4746705905 ; 79,090062367308) |
| (x-10 ; y-10) | (102,591509930436 ; 82,3870030007502) | (102,9804840728 ; 77,03625103126) |
| (x-11 ; y-11) | (102,337529858549 ; 86,0668089624719) | (104,76238653 ; 80,02670751238) |
| (x-12 ; y-12) | (102,067240006538 ; 76,3185486308011) | (112,5634936119 ; 83,54464915735) |
| (x-13 ; y-13) | (102,099908488992 ; 83,2713560044981) | (125,1177800506 ; 75,70146135853) |
| (x-14 ; y-14) | (102,089734064272 ; 86,0761782872406) | (121,6437143985 ; 79,42271100365) |
| (x-15 ; y-15) | (102,508602683522 ; -6,5029988325741) | (138,883711202 ; 148,65970332356) |
| (x-16 ; y-16) | (102,415262335134 ; -6,46215177033941) | (136,2198154078 ; 162,3581339773) |
| (x-17 ; y-17) | (102,302113776334 ; 8,59231815634098) | (138,4576016509 ; 120,2638504835) |
| (x-18 ; y-18) | (102,618519975141 ; 8,24117348108143) | (134,114202934 ; 146,9751178143) |
| (x-19 ; y-19) | (101,973552976111 ; 86,6329095155132) | (120,2554594646 ; 80,25577366312) |
| (x-20 ; y-20) | (101,736075685494 ; 86,4745693037296) | (119,7183058179 ; 79,737809038512) |
| (x-21 ; y-21) | (102,114630308393 ; 86,5331547609049) | (120,4868243202 ; 79,95763773605) |
| (x-22 ; y-22) | (102,832095039208 ; 86,2375865891036) | (106,6824578395 ; 81,55223931802) |
| (x-23 ; y-23) | (102,828394636669 ; 86,1539294936229) | (106,6975789714 ; 80,564099687783) |
| (x-24 ; y-24) | (102,296078102814 ; 86,272160340071) | (104,6178865381 ; 80,62694506954) |

Cuadro 12.7: Puntos Clave Frame 95 Curva - Frame 147 Dab

Fuente: Elaboración Propia

| Coordenadas | Frame 155 Curva Cerrada | Frame 158 Curva Cerrada |
|---------------|---------------------------------------|---------------------------------------|
| (x-0 ; y-0) | (93,9756064404827 ; 91,7275098679654) | (94,7927892828664 ; 91,7194738072001) |
| (x-1 ; y-1) | (95,3597393494047 ; 98,7439590121243) | (96,7130209632533 ; 98,7209457249791) |
| (x-2 ; y-2) | (94,2769751679131 ; 101,153426979898) | (95,7885001812785 ; 101,090457737951) |
| (x-3 ; y-3) | (91,6138430787281 ; 108,73030967511) | (93,3559070590144 ; 108,747650586235) |
| (x-4 ; y-4) | (89,2651176903157 ; 119,44742459602) | (91,3306770222064 ; 119,470098326568) |
| (x-5 ; y-5) | (96,2906045442405 ; 91,6974113855418) | (97,0270910050673 ; 91,6986230000788) |
| (x-6 ; y-6) | (97,7659203793153 ; 92,8576618029909) | (97,9909224858634 ; 96,3673790639947) |
| (x-7 ; y-7) | (100 ; 100) | (100 ; 100) |
| (x-8 ; y-8) | (99,7168321631585 ; 102,60015372709) | (100,525282642414 ; 102,063165457365) |
| (x-9 ; y-9) | (98,5681062484384 ; 104,707144515582) | (99,4253204943397 ; 103,132314048043) |
| (x-10 ; y-10) | (102,882796825916 ; 105,715686734184) | (103,468024040014 ; 105,342546574144) |
| (x-11 ; y-11) | (106,147308654754 ; 103,027778593497) | (106,74096465735 ; 103,028087708208) |
| (x-12 ; y-12) | (99,7498009480672 ; 101,550383522839) | (100,010282152819 ; 101,025618116888) |
| (x-13 ; y-13) | (101,832104214889 ; 102,28372561945) | (102,363406362707 ; 102,274419326069) |
| (x-14 ; y-14) | (103,658038648701 ; 102,736043873154) | (104,441320776187 ; 102,422421308404) |
| (x-15 ; y-15) | (93,8186746280289 ; 88,4894642471402) | (94,6554485572419 ; 90,6860899260237) |
| (x-16 ; y-16) | (94,0811155275507 ; 86,0080673696129) | (94,8812992123278 ; 86,021866817635) |
| (x-17 ; y-17) | (93,9661666808347 ; 93,8506709931818) | (95,1118739062505 ; 95,8660852393933) |
| (x-18 ; y-18) | (95,7796029803987 ; 84,3768373010732) | (96,5669844738463 ; 84,3394684656855) |
| (x-19 ; y-19) | (103,941550915697 ; 101,459465470586) | (104,688955829885 ; 101,449234533121) |
| (x-20 ; y-20) | (103,461822835385 ; 102,064701584458) | (104,438373280834 ; 102,340005130998) |
| (x-21 ; y-21) | (102,919675438276 ; 103,593032010177) | (103,722555740919 ; 102,989662855459) |
| (x-22 ; y-22) | (105,989503727753 ; 104,986455377537) | (107,241205108486 ; 104,975386332122) |
| (x-23 ; y-23) | (105,675291286035 ; 105,010804076056) | (106,621146352529 ; 105,01396404259) |
| (x-24 ; y-24) | (105,471283222652 ; 102,982472701141) | (106,331154991519 ; 103,27829915774) |

Cuadro 12.8: Puntos Clave Frame 155 Curva Cerrada - Frame 158 Curva Cerrada

Fuente: Elaboración Propia

| Coordenadas | Frame 72 Desfase | Frame 188 Desfase |
|---------------|---------------------------------------|---------------------------------------|
| (x-0 ; y-0) | (99,4626858141487 ; 122,756855235652) | (101,374677842727 ; 97,9979634930219) |
| (x-1 ; y-1) | (101,076242807512 ; 110,598224343213) | (102,448337418773 ; 94,0716561088165) |
| (x-2 ; y-2) | (100,299570462045 ; 110,666020828288) | (104,203357647189 ; 95,1737708888351) |
| (x-3 ; y-3) | (101,037571166969 ; 110,925892777231) | (105,23564223676 ; 95,6726035979594) |
| (x-4 ; y-4) | (99,5836432480041 ; 117,26466326745) | (105,357432579073 ; 105,38971165651) |
| (x-5 ; y-5) | (100,017342496585 ; 104,773761275588) | (100,501413350178 ; 94,0509726235695) |
| (x-6 ; y-6) | (100,662253163807 ; 101,092813464915) | (99,7715818695671 ; 95,206756674941) |
| (x-7 ; y-7) | (100 ; 100) | (100 ; 100) |
| (x-8 ; y-8) | (102,456176239402 ; 102,072674349499) | (104,385108258104 ; 90,7090199765976) |
| (x-9 ; y-9) | (102,602246971733 ; 103,647116280187) | (105,206563795043 ; 92,1605414363756) |
| (x-10 ; y-10) | (104,034235054963 ; 102,287485933983) | (103,452987975571 ; 91,2003391472868) |
| (x-11 ; y-11) | (105,268566146501 ; 99,6962947965175) | (103,211486024629 ; 89,7112714042618) |
| (x-12 ; y-12) | (101,828245864729 ; 101,537770233162) | (102,846310816298 ; 90,228116599352) |
| (x-13 ; y-13) | (104,468697852183 ; 99,6186358524952) | (106,569682941271 ; 87,83434797782) |
| (x-14 ; y-14) | (106,029839956626 ; 98,190757949536) | (108,904194863351 ; 86,9864411180977) |
| (x-15 ; y-15) | (99,6932831736954 ; 123,180781030333) | (101,94831799988 ; 100,032579326529) |
| (x-16 ; y-16) | (99,4581574052051 ; 120,760738801217) | (100,542392549218 ; 93,1855495010969) |
| (x-17 ; y-17) | (99,4311769551186 ; 118,220161187554) | (102,290535054215 ; 98,0448834503544) |
| (x-18 ; y-18) | (99,7350339127062 ; 105,986786379807) | (100,783488881223 ; 90,3423981823799) |
| (x-19 ; y-19) | (105,678636087032 ; 97,39303851429) | (106,416572188962 ; 86,3620120843682) |
| (x-20 ; y-20) | (105,674144778358 ; 97,6641956860609) | (106,903870303823 ; 86,7794285988625) |
| (x-21 ; y-21) | (106,362562560125 ; 98,5053976418055) | (109,530516622491 ; 87,7408093484804) |
| (x-22 ; y-22) | (108,813071752414 ; 100,890160665698) | (105,354804248706 ; 91,5130338009756) |
| (x-23 ; y-23) | (107,254076073841 ; 100,609271642761) | (105,030295034703 ; 91,1678935362132) |
| (x-24 ; y-24) | (104,604356095682 ; 99,6945816407939) | (102,58784655772 ; 90,4482920323262) |

Cuadro 12.9: Puntos Clave Frame 155 Curva Cerrada - Frame 158 Curva Cerrada

Fuente: Elaboración Propia

12.10. Diseño de GUI Modificados



Figura 12.15: GUI del Menú Principal
Fuente: Elaboración Propia



Figura 12.16: GUI del Menú de Opciones
Fuente: Elaboración Propia



Figura 12.17: GUI del Menú de Opciones

Fuente: Elaboración Propia

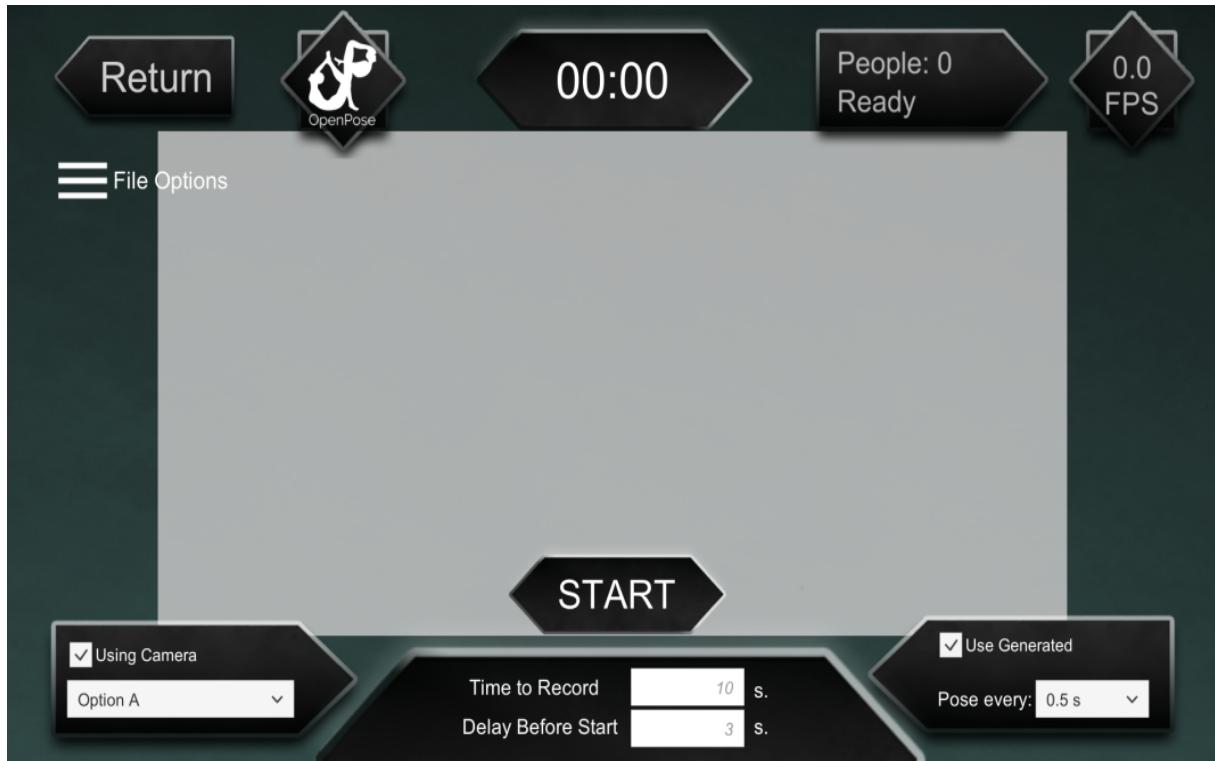


Figura 12.18: GUI del modo de Creación de Mapas
Fuente: Elaboración Propia

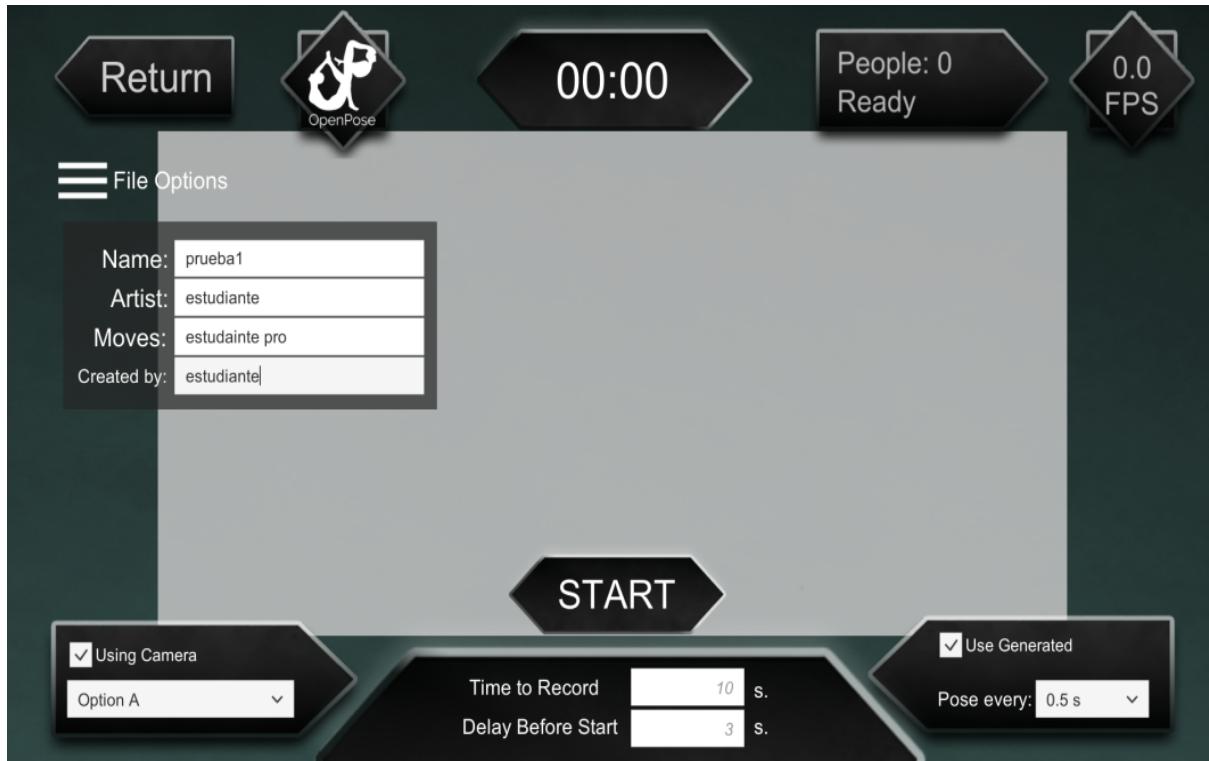


Figura 12.19: GUI del modo de Creación de Mapas Sección de Llenar Datos del Mapa
Fuente: Elaboración Propia

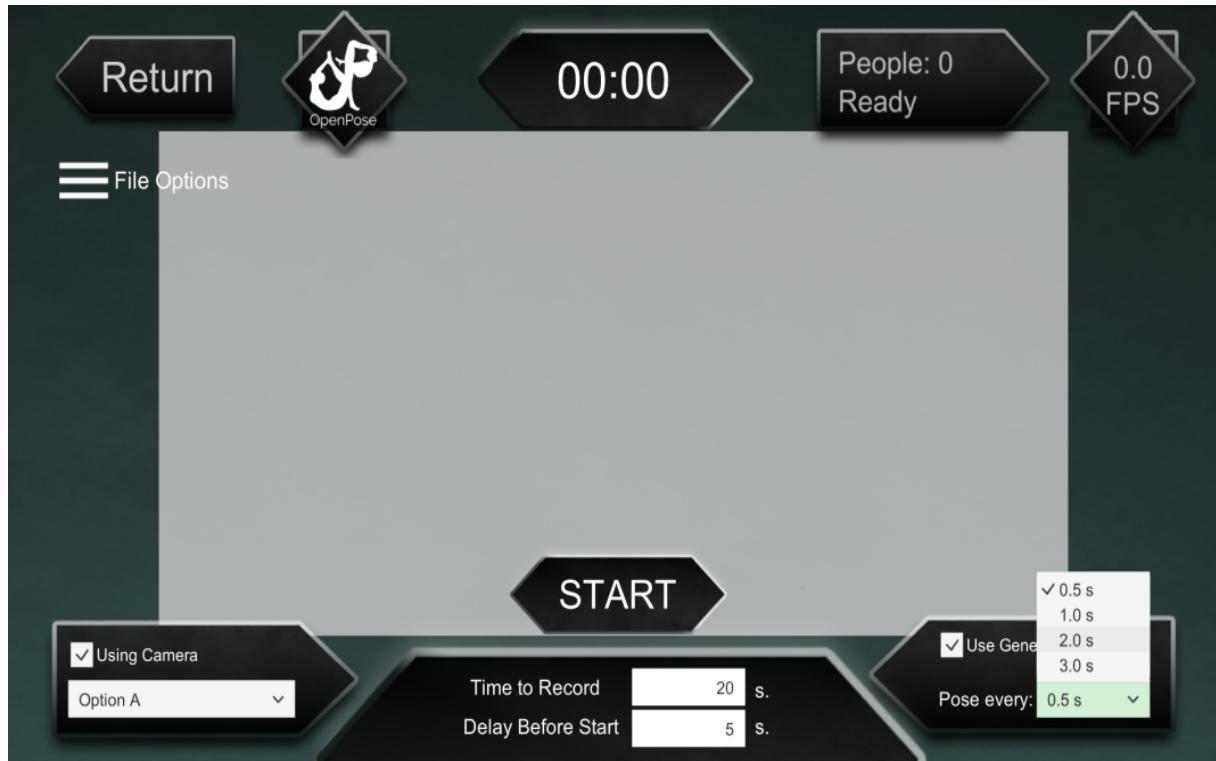


Figura 12.20: GUI del modo de Creación de Mapas Sección de Tiempo Entre Poses Guardadas

Fuente: Elaboración Propia

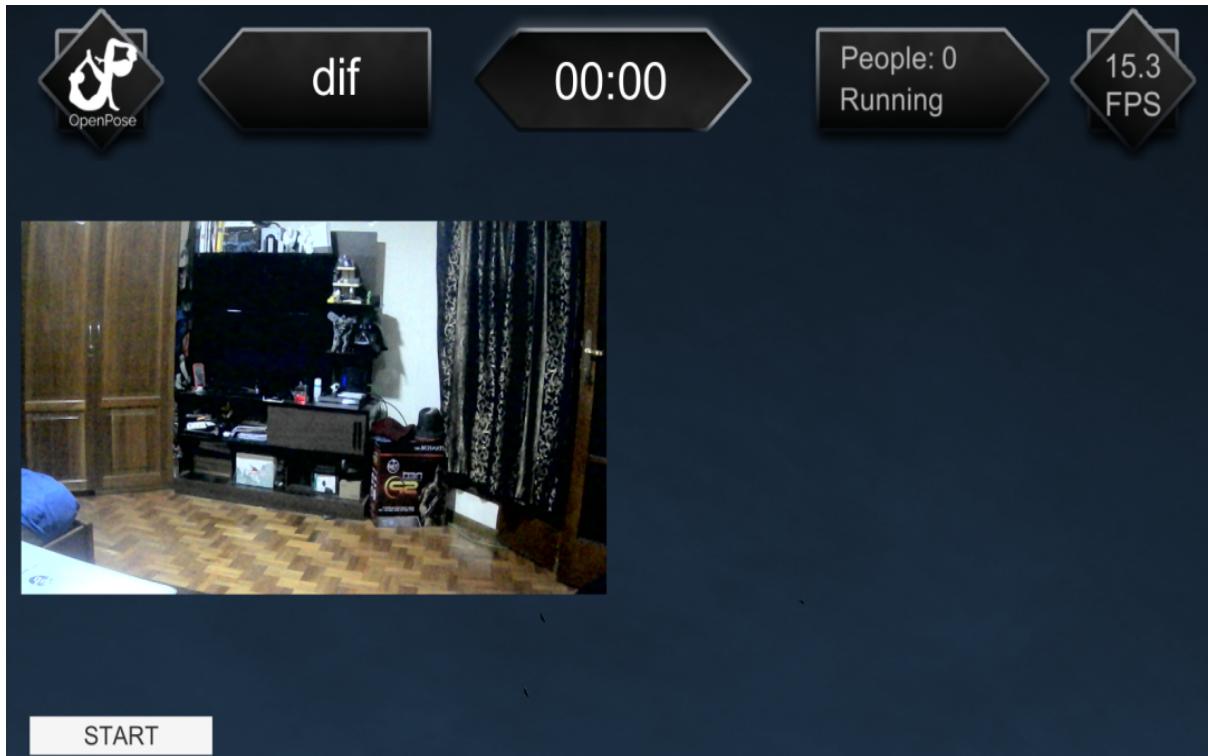


Figura 12.21: GUI del modo de Juego, Fondo Azul Previo Inicio de Start
Fuente: Elaboración Propia

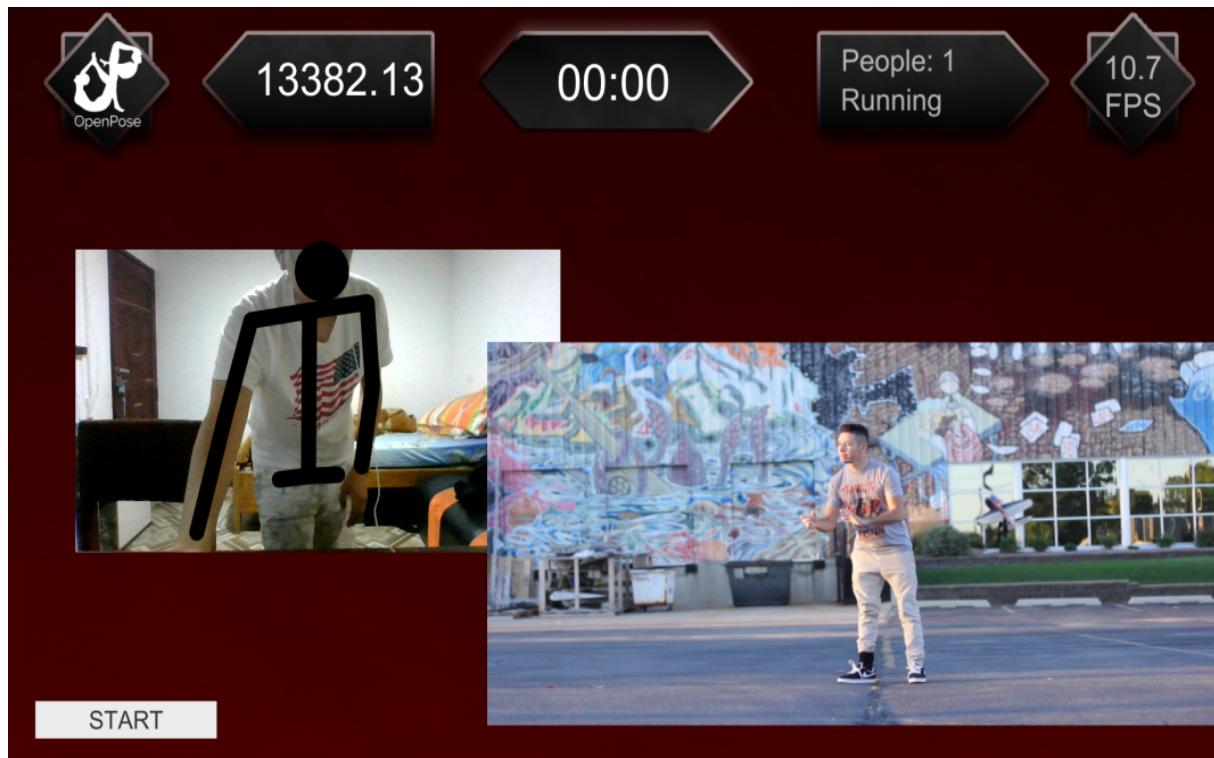


Figura 12.22: GUI del modo de Juego, Fondo Rojo en Error

Fuente: Elaboración Propia

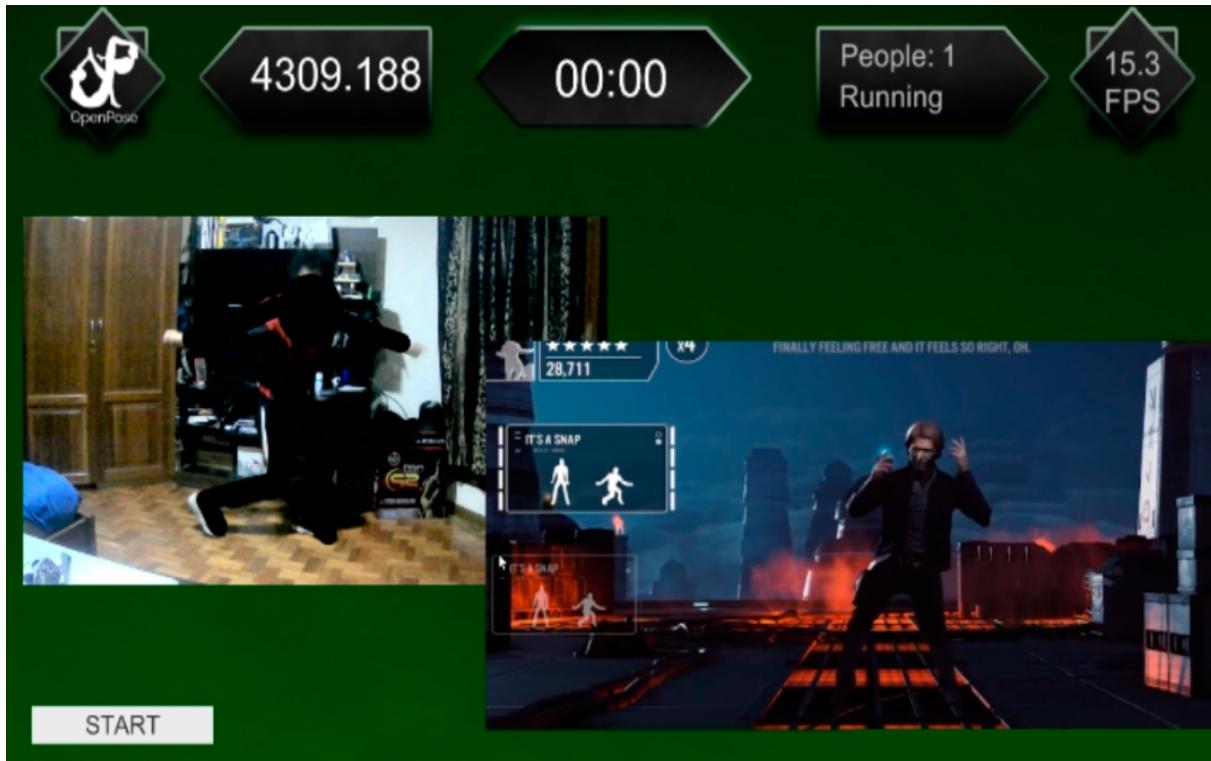


Figura 12.23: GUI del modo de Juego, Fondo Verde Correcto
Fuente: Elaboración Propia