

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

**DISEÑO Y APLICACIÓN DE UN MODELO DE
RECONOCIMIENTO DE 11 GESTOS DE LA MANO
USANDO SEÑALES EMG Y DEEP LEARNING.**

**DESARROLLO DE UNA INTERFAZ HUMANO-MÁQUINA USANDO
RECONOCIMIENTO DE GESTOS DE LA MANO**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SOFTWARE**

CÉSAR ISRAEL LEÓN CIFUENTES

israelleon98@hotmail.com

DIRECTOR: LORENA ISABEL BARONA LÓPEZ

lorena.barona@epn.edu.ec

DMQ, Agosto 2022

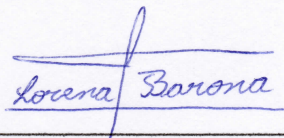
CERTIFICACIONES

Yo, César Israel León Cifuentes declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



César Israel León Cifuentes

Certifico que el presente trabajo de integración curricular fue desarrollado por César Israel León Cifuentes, bajo mi supervisión.



Lorena Isabel Barona López
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

César Israel León Cifuentes

Lorena Isabel Barona López

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	¡Error! Marcador no definido.
DECLARACIÓN DE AUTORÍA	I
ÍNDICE DE CONTENIDO.....	II
RESUMEN.....	III
ABSTRACT.....	IV
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO.....	1
1.1 Objetivo general.....	3
1.2 Objetivos específicos.....	3
1.3 Alcance	3
1.4 Marco teórico	5
2 Metodologías de desarrollo de software.....	7
2.1 Metodologías tradicionales de desarrollo de software.....	7
2.2 Metodologías ágiles de desarrollo de software.....	10
3 METODOLOGÍA.....	16
3.1 Selección de la metodología	16
3.2 Desarrollo de un videojuego musical utilizando la metodología SUM.....	16
4 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	29
4.1 Resultados	29
4.2 Conclusiones.....	34
4.3 Recomendaciones	35
5 REFERENCIAS BIBLIOGRÁFICAS.....	36
6 ANEXOS.....	38
ANEXO I	38
ANEXO II	40
ANEXO III	44
ANEXO IV.....	46

RESUMEN

PALABRAS CLAVE: Desarrollo, gestos de la mano, metodología SUM, videojuego musical.

Para la creación de videojuegos se han planteado diversas metodologías, pero muy pocas son acertadas para acoplarse a las necesidades de un software de esta naturaleza. Por ello, la metodología SUM está conformada de fases que permiten el desarrollo de un videojuego de calidad considerando los diversos elementos que debe tener un videojuego como también el contexto sobre el cual se pretende establecer el desarrollo. En este documento se describe la aplicación de la metodología SUM en el desarrollo de un videojuego musical que utiliza un modelo de inteligencia artificial para procesar el gesto realizado por el jugador mediante un brazalete. La metodología se centra en el desarrollo del videojuego, y no en la integración entre el modelo y el videojuego.

ABSTRACT

KEYWORDS: Development, hand gestures, SUM methodology, music video game

For the creation of video games, various methodologies have been proposed, but very few of these methodologies are able to adapt the software needs to the nature of this kind of applications. Therefore, the SUM methodology is made up of phases that allow the development of a quality video game considering the various elements that a video game must. Also, the context on which the development is intended to be established is considered. This document describes the application of the SUM methodology in the development of a music video game that uses an artificial intelligence model to process the gesture made by the player using a bracelet. The methodology focuses on the development of the video game rather than the integration between the model and the video game.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El componente desarrollado es un videojuego musical de baile el cual utiliza un modelo de reconocimiento de 11 gestos de la mano para comandar las diferentes funcionalidades del videojuego. Para recolectar la información de los movimientos del usuario, el usuario deberá usar un brazalete colocado en el antebrazo, mismo que capturará las señales musculares y de rotación que se realicen con el brazo. El videojuego será controlado según el gesto que haya sido reconocido mediante el modelo de inteligencia artificial, y si existe alguna acción disponible en el videojuego para ese gesto en ese momento. En caso de que no se reconozca ningún gesto definido o no se encuentre una acción disponible para el gesto detectado, el videojuego no realizará ninguna acción.

El videojuego de baile consta de 4 menús: menú principal, tabla de canciones, configuración del volumen e instrucciones. En el menú principal el usuario podrá seleccionar la opción para visualizar los otros menús del videojuego. En el menú de tabla de canciones se puede visualizar el nombre, dificultad y mejor puntuación de las 6 canciones definidas. En este menú el usuario se desplaza mediante los gestos para seleccionar una canción y a medida que lo hace también puede escuchar la canción que está seleccionando. Por otro lado, en el menú de configuración de volumen el usuario podrá controlar el volumen mediante el desplazamiento de una barra que representa el volumen actual del videojuego y de esta forma configurarlo a su preferencia. Con respecto a el menú de instrucciones, el usuario podrá ver videos de los gestos que puede realizar para controlar el videojuego y a su vez las acciones relacionadas a estos.

Además, el videojuego tiene una pantalla donde se le muestra un modelo 3D de un bailarín, el escenario que simula una pista de baile, el nombre de la canción, la puntuación del baile y el video del gesto requerido para que el modelo baile. También el usuario podrá escuchar la canción que ha escogido para bailar. Esta pantalla se la podrá visualizar una vez que el usuario seleccione una canción en la tabla de canciones. En esta pantalla se irá recolectando la información de los gestos realizados por el usuario y se irá contabilizando si se ha realizado el gesto mostrado en pantalla para así aumentar la puntuación. Además, si el gesto realizado es correcto el modelo realizará un paso de baile el cuál es único para el gesto realizado. Cabe recalcar que los pasos de baile requeridos son mostrados aleatoriamente con un tiempo de espera hasta que el usuario realice la acción, por lo cual nunca se repetirán los mismos pasos en una misma o diferente canción. Si el usuario termina de bailar

se le mostrará una pantalla donde se muestra su puntuación y en el caso que haya superado la mejor puntuación para esa canción se mostrará un mensaje mencionando que existe una mejor puntuación y ésta se guardará como la nueva mejor puntuación. Si, por el contrario, el usuario no quiere terminar la canción, tendrá la posibilidad de volver a la tabla de canciones para seleccionar otra canción o ir a los diferentes menús. En este caso la puntuación obtenida hasta el momento no será comparada con la mejor puntuación actual de la canción que se estaba bailando.

1.1 Objetivo general

Desarrollar un videojuego musical que sea manipulado mediante once gestos de la mano utilizando un modelo de reconocimiento basado en inteligencia artificial.

1.2 Objetivos específicos

- Desarrollar el concepto del videojuego musical basado en elementos técnicos y de juego para conseguir una interacción natural usando los 11 gestos de la mano.
- Diseñar e implementar las características principales del videojuego musical tomando como guía de desarrollo la metodología ágil SUM.
- Integrar el videojuego con el componente de modelo de reconocimiento de once gestos de la mano.

1.3 Alcance

Desarrollar un videojuego musical que permita su manipulación utilizando los 11 gestos descritos en la Figura 1:

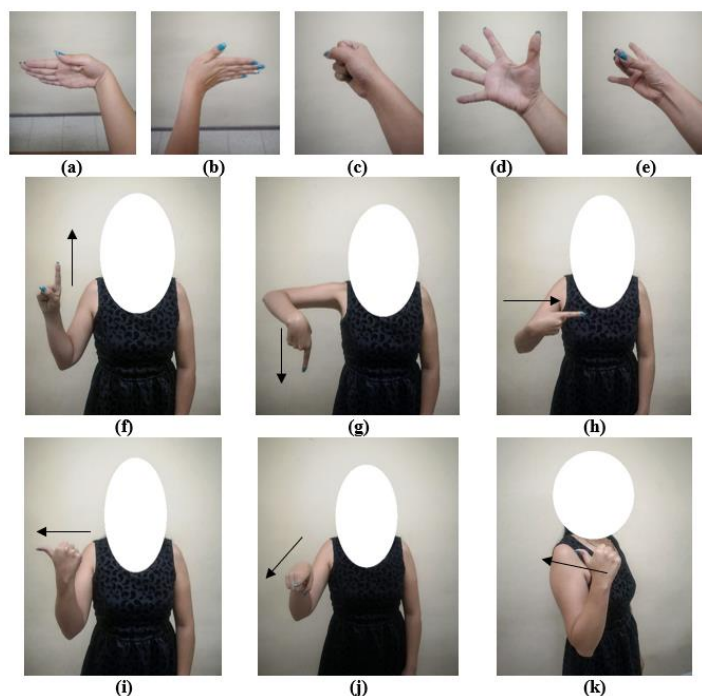


Figura 1. Gestos (a) mano a la izquierda, (b) mano a la derecha, (c) puño, (d) mano abierta, (e) doble golpe de los dedos, (f) arriba, (g) abajo, (h) izquierda, (i) derecha, (j) apuntar al frente, y (k) apuntar hacia atrás.

Las acciones implementadas en el videojuego deberán estar acorde a los gestos descritos para que la manipulación y navegación dentro del videojuego sea lo más

natural posible. Para cumplir este fin, la Tabla 1 describe la relación entre los gestos mostrados en la Figura 1 y las acciones que se realizarán en el videojuego.

Tabla 1: Definición de la relación entre los gestos y sus acciones dentro del videojuego

Gesto	Acciones
(a) mano a la izquierda- (wave in)	<ul style="list-style-type: none"> • Desplazamiento entre canciones a la izquierda • Gesto del baile
(b) mano a la derecha (wave out)	<ul style="list-style-type: none"> • Desplazamiento entre canciones a la derecha • Gesto del baile
(c) puño (fist)	<ul style="list-style-type: none"> • Volver al menú • Parar canción
(d) mano abierta (open)	<ul style="list-style-type: none"> • Seleccionar opción en menú • Seleccionar canción • Gesto del baile
(e) doble golpe de los dedos (pinch)	<ul style="list-style-type: none"> • Seleccionar canción • Gesto del baile
(f) arriba (up)	<ul style="list-style-type: none"> • Desplazamiento en menú hacia arriba • Subir volumen • Gesto del baile
(g) abajo (down)	<ul style="list-style-type: none"> • Desplazamiento en menú hacia abajo • Bajar volumen • Gesto del baile
(h) izquierda (left)	<ul style="list-style-type: none"> • Gesto del baile
(i) derecha (right)	<ul style="list-style-type: none"> • Gesto del baile
(j) apuntar al frente (forward)	<ul style="list-style-type: none"> • Seleccionar opción en menú • Seleccionar canción • Gesto del baile
(k) apuntar hacia atrás (backward)	<ul style="list-style-type: none"> • Gesto del baile

Además, para recolectar la información de los gestos del usuario se deberá usar únicamente el brazalete colocado en el antebrazo. Por otra parte, el videojuego se comunicará con una aplicación realizada en Matlab que recibe esta información y envía al videojuego el gesto reconocido. La Figura 2 presenta la arquitectura propuesta para la comunicación entre el usuario y el aplicativo:

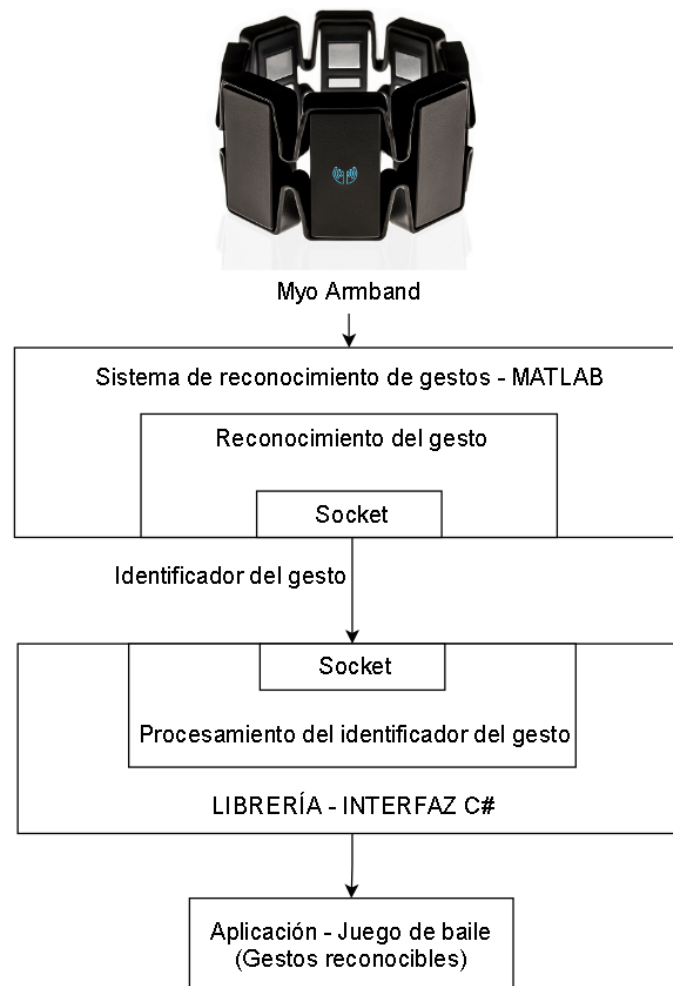


Figura 2. Arquitectura para la comunicación entre el jugador y el aplicativo

1.4 Marco teórico

1.4.1 ¿Qué son los videojuegos?

Los videojuegos son una forma de entretenimiento que implica la interacción entre un usuario denominado jugador, una interfaz y un dispositivo de videojuego [1]. Los videojuegos permiten simular, mediante la virtualidad, situaciones donde uno o varios jugadores pueden controlar personajes que a su vez suelen tener una evolución en base a una serie de objetivos que se deben cumplir basados en unas reglas determinadas [1] [2] [3]. Cabe recalcar que el proceso para cumplir estos objetivos debe garantizar que el jugador se divierta y entretenga [2] [3]. De esta forma, la interacción entre el jugador y el videojuego es explícita y se la realiza mediante varios dispositivos electrónicos como son monitores, televisiones, proyectores, periféricos, etc [1] [2]. En los 2007 [1] [4], se definió que un videojuego puede ser creado por cualquier persona que tenga un computador y conocimientos de lenguaje Ensamblador, y de esta forma bastaba con tener un grupo de 2 a tres personas para desarrollar un videojuego en un tiempo aproximado de 3 meses. Esta concepción ha ido cambiando con el transcurso del tiempo

y de tal manera que actualmente el desarrollo de videojuegos considera varias fases teniendo en cuenta tanto el aspecto técnico como el artístico [1] [4].

1.4.2 Videojuegos musicales

En los últimos años se han desarrollado muchos videojuegos musicales a nivel comercial los cuales han ido evolucionando en sus métodos para el control del videojuego. Al inicio, los videojuegos musicales eran controlados por periféricos como el teclado y ratón, pero en los últimos años se ha podido hacer uso de sensores que perciben el movimiento del cuerpo del jugador y transmiten esta información al videojuego para ser procesada. Cabe mencionar que el objetivo de este tipo de videojuegos no es la de enseñar o educar, pero pueden ser enfocados de tal manera que cumplan esta función [5].

1.4.3 Motor de juego y su arquitectura

El termino motor de juego fue concebido en la última década del siglo anterior con la aparición del muy conocido juego DOOM, desarrollado por idSoftware. Este término se relaciona con una arquitectura que permite la reutilización de componentes para el desarrollo del videojuego a través de la creación de módulos de los componentes fundamentales como son: sistema de renderizado, elementos gráficos y artísticos, escenarios virtuales o las directrices que debía cumplir el propio juego. Esta arquitectura se encuentra dividida en capas, cuyas capas superiores dependen de las inferiores, por lo cual la dependencia es unidireccional [2] [3]. Las capas son descritas en la Tabla 2:

Tabla 2. Definición de las capas de la arquitectura de un motor de videojuegos

Capas de la arquitectura	Definición
Gestor de recursos	Interfaz unificada para acceder a las distintas entidades software que conforma el motor de juegos.
Motor de renderizado	Capa encargada de dibujar las primitivas que son visibles para la cámara.
Motor de física	Capa encargada de la detección de colisiones de los elementos del videojuego y su reacción resultante a dicha colisión.

Interfaces de usuario	Capa que permite la abstracción de la interacción entre el jugador y el videojuego.
Audio	Capa que gestiona el sonido reproducido en el videojuego

2 Metodologías de desarrollo de software

2.1 Metodologías tradicionales de desarrollo de software

En esta sección se describen las metodologías tradicionales que son utilizadas para el desarrollo de software. Se mencionan sus etapas y características importantes sobre las mismas.

2.1.1 Metodología Cascada

Es la primera metodología de desarrollo de software en la cual se debe atravesar por una serie de fases para obtener el producto software terminado. Se debe mencionar que estas fases son secuenciales y ordenadas, donde el resultado de una fase es la entrada de la posterior [6]. A continuación, en la Tabla 3 se describen cada una de las fases y sus actividades clave:

Tabla 3. Descripción de las fases de la metodología en cascada

Fase	Descripción
Preanálisis	Conocer las reglas de negocio, sus necesidades y la adquisición del conocimiento de las actividades y funciones que se desempeña en el negocio.
Análisis	Se propone una solución en base a los hallazgos de la fase anterior, teniendo en cuenta la viabilidad a nivel técnico y administrativo.
Desarrollo	Se ejecuta el diseño de la solución realizada en la fase anterior utilizando uno o varios lenguajes de programación y programas de software ya existentes como apoyo.
Pruebas	Se prueba el producto desarrollado con un enfoque individual e integrado para verificar que la solución desarrollada satisface las necesidades del usuario.
Puesta en marcha	Se entrega al usuario el producto desarrollado y, de ser necesario, se insertan valores y configuraciones para su uso.

Mantenimiento	Se controlan los desajustes que se producen a través del uso del producto y se da mantenimiento a la infraestructura física y lógica como son redes y bases de datos.
---------------	---

En cada una de estas fases existen documentos aprobados por parte del cliente y por parte del equipo de desarrollo. Su principal problema es tener compromisos iniciales con el cliente que generan complicaciones con cambios imprevistos o adición de nuevas funcionalidades [6] [7] [8]. Esta metodología es adecuada cuando los requerimientos son bien conocidos tanto por el cliente como por el equipo de desarrollo y la probabilidad de que existan cambios significativos es baja.

2.1.2 Metodología Prototipo

La metodología del prototipo se centra en el desarrollo continuo de un modelo, maqueta o borrador del sistema que cada vez se mejora para que cumpla con las necesidades del usuario. Estos prototipos pueden tener un grado de fidelidad variable a lo largo de la implementación de la metodología por lo cual el cliente podría confundir que el prototipo ya es el producto desarrollado [6] [7] [8]. Esta metodología permite a los desarrolladores descubrir las necesidades que tienen los usuarios cuando existe mucha incertidumbre. Esto a su vez disminuye el riesgo de desarrollar un producto que no satisfaga al cliente o de realizar cambios significativos. Una vez que el prototipo cubre todos los requisitos del usuario pasa a la fase de desarrollo del sistema. Caso contrario continúa el ciclo de prototipado donde la definición del problema y su solución mediante el prototipado son las principales actividades.

2.1.3 Metodología Incremental

La metodología incremental está basada en incrementos que proveen de funcionalidad al sistema progresivamente mediante la ejecución de una serie de avances a través de fases secuenciales. Cada uno de los incrementos es evaluado por el cliente, por lo cual el cliente no debe esperar a que todo el sistema se encuentre hecho para observar algunas funcionalidades [6] [7] [8]. Los incrementos iniciales pueden servir como prototipos para los requerimientos posteriores. En la figura 3 se muestran las fases a seguir:

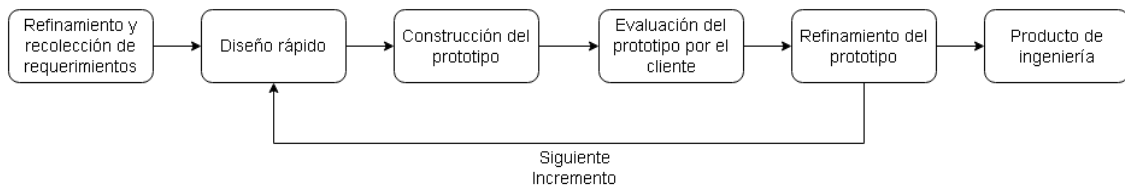


Figura 3. Fases de la metodología incremental

2.1.4 Metodología Espiral

La metodología espiral es un modelo evolutivo que fusiona la realización de prototipos de forma incremental con los aspectos controlados y sistémicos de la metodología en cascada. Esta metodología permite realizar versiones más completas del sistema donde el riesgo es menor y las soluciones son viables y satisfactorias para el cliente [6] [7] [8]. Este modelo se divide en cuatro cuadrantes que se describen a continuación:

- Definición de objetivos: en este cuadrante se definen los objetivos específicos, restricciones y se realiza un plan detallado de gestión tomando en cuenta a su vez estrategias alternativas.
- Evaluación y reducción de riesgos: en este cuadrante se realiza un análisis exhaustivo de los riesgos del proyecto y se define como reducirlos.
- Desarrollo y validación: en este cuadrante se define un modelo adecuado para el desarrollo del sistema o subsistema.
- Planificación: en este cuadrante se revisa el proyecto, se evalúa si se continua con el siguiente ciclo y se planifica las siguientes fases.
-

2.1.5 Metodología en V

La metodología en V se centra en iniciar las pruebas lo más pronto posible en el ciclo de vida, es decir una verificación y validación constante durante todo el desarrollo. Por lo tanto, tanto los técnicos de pruebas como los desarrolladores y analistas deben estar en constante comunicación para producir los entregables de pruebas para cada fase del ciclo de vida [6] [7] [8]. Cabe recalcar que las pruebas de validación son realizadas mayormente en etapas tempranas del desarrollo. En la figura 4 se muestran las fases de verificación validación de esta metodología.

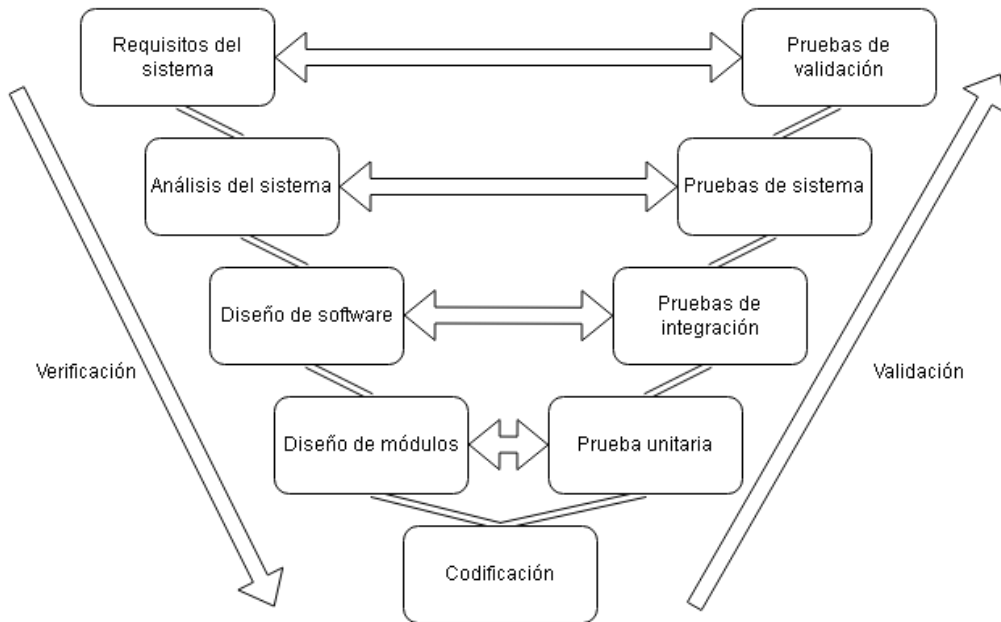


Figura 4. Fases de la metodología en V

2.2 Metodologías ágiles de desarrollo de software

2.2.1 Metodologías SUM para Videojuegos

La metodología SUM tiene como objetivo el desarrollar videojuegos de alta calidad en tiempo y costo, tomando en cuenta la mejora continua. Esto constituye una administración eficiente de los recursos asignados para el desarrollo y los riesgos del proyecto. Además, SUM se adapta a equipos de desarrollo multidisciplinarios donde intervienen participantes con diversas habilidades en diferentes campos. Es importante mencionar que esta metodología es adecuada para proyectos menores a un año, donde el equipo de desarrollo tenga entre tres a siete integrantes y donde la participación del cliente sea constante [2] [6] [9] [10].

Esta metodología se basa en los principios de las metodologías ágiles tal que no existe una metodología ágil especificada formalmente para el desarrollo de videojuegos [9]. Además, adapta la estructura y roles de Scrum ya que provee flexibilidad para plantear el ciclo de vida y una alta adaptabilidad a las diferentes realidades y contextos de desarrollo [6] [9] [11].

Fases y ciclo de vida

Las fases de la metodología SUM se las realiza de forma iterativa e incremental y se dividen en: concepto, planificación, elaboración, beta y cierre. Por otra parte, se define una fase de gestión de riesgos que, a diferencia de las anteriores, se la realiza a lo largo del ciclo de vida y cubre las demás fases. Como se muestra en la Figura 5, las fases de

concepto, planificación y cierre se las realiza en una iteración, mientras que las fases de elaboración y beta son realizadas en múltiples iteraciones según sea necesario. Cabe recalcar que estas fases se las realiza de forma secuencial [2] [6] [9].

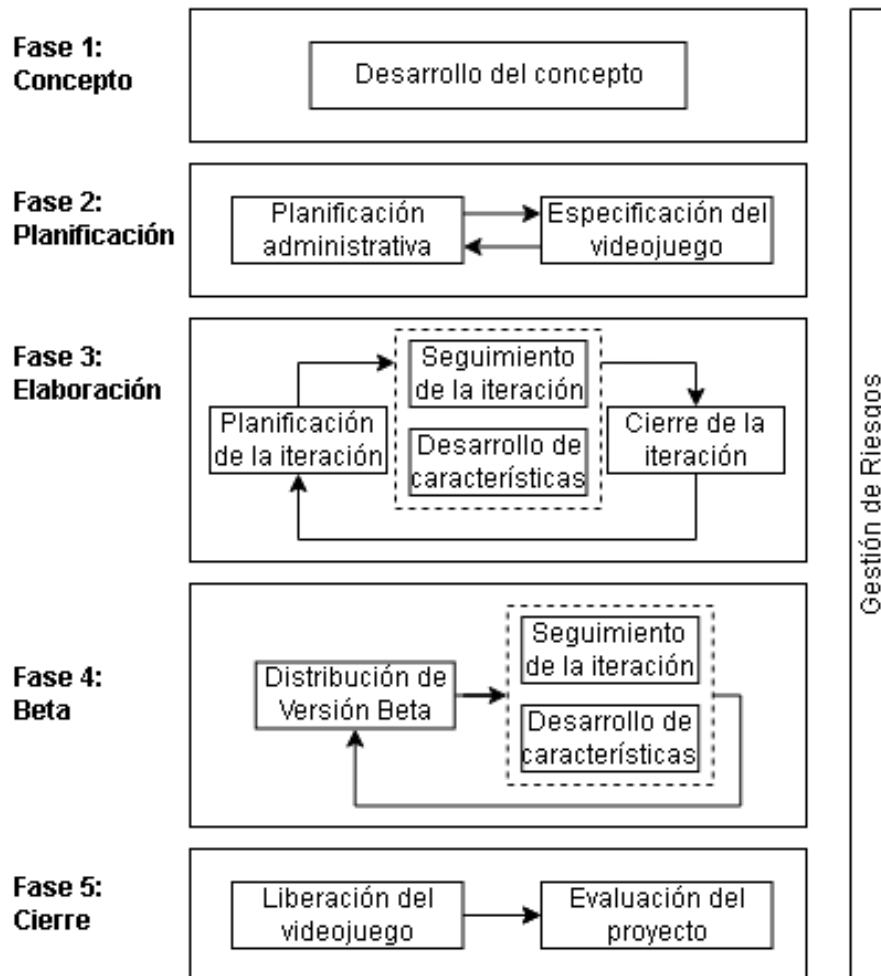


Figura 5. Diagrama del ciclo de vida de la metodología SUM

Las fases descritas en SUM son una adaptación de las definidas en Scrum donde las fases pre-game, game y post-game concuerdan con las fases de planificación, elaboración y beta-cierre [6] [9]. Cada una de las fases en SUM tienen sus objetivos a cumplir a lo largo del desarrollo del videojuego. En la Tabla 4 se listan los principales aspectos que se toman en cuenta en cada una de las fases:

Tabla 4. Descripción de las fases de la metodología SUM [2] [6] [9]

Fase	Aspectos analizados	Descripción
Concepto	<ul style="list-style-type: none"> • Definir el concepto del videojuego <ul style="list-style-type: none"> ○ Público objetivo ○ Modelo de negocio ○ Principales características ○ Gameplay ○ Personajes e historia • Definición del ambiente de desarrollo <ul style="list-style-type: none"> ○ Lenguaje de programación ○ Herramientas de desarrollo 	<p>Se construye a partir de ideas y propuestas por parte del equipo. Estos aspectos son refinados mediante reuniones hasta que todas las partes hayan validado los conceptos. Algunos aspectos pueden ser definidos posteriormente.</p>
Planificación	<ul style="list-style-type: none"> • Planificación de fases posteriores <ul style="list-style-type: none"> ○ Creación del cronograma ○ Conformación del equipo de desarrollo ○ Determinar tareas ○ Tercerizar tareas ○ Definición del presupuesto ○ Definir, estimar y priorizar características funcionales y no funcionales 	<p>Su construcción es de manera flexible ya que en cada iteración de la fase de elaboración se podrá realizar cambios como el agregar, modificar o eliminar alguna característica para satisfacer las necesidades del cliente.</p>
Elaboración	<ul style="list-style-type: none"> • Implementación del videojuego <ul style="list-style-type: none"> ○ Planificación de objetivos, métricas y tareas ○ Desarrollo de características ○ Seguimiento de los objetivos planteados ○ Evaluación del estado del videojuego 	<p>Se desarrolla el videojuego definiendo las tareas a realizar, y paralelamente se hace un seguimiento de los objetivos planteados en cada iteración para así evaluar el proyecto y conocer su estado.</p>
Beta	<ul style="list-style-type: none"> • Evaluar y ajustar aspectos del videojuego como: 	<p>Durante la ejecución de esta fase se evalúa y</p>

	<ul style="list-style-type: none"> ○ Gameplay ○ Diversión ○ Curva de aprendizaje ○ Curva de dificultad • Eliminación de errores <ul style="list-style-type: none"> ○ Verificación del videojuego ○ Reporte de errores y evaluaciones realizadas 	<p>verifica que lo desarrollado en la fase de elaboración concuerde con lo descrito en las dos primeras fases y de esta forma cumplir con los criterios de finalización establecidos. Además, se realiza la corrección de errores.</p>
Cierre	<ul style="list-style-type: none"> • Entrega de versión final del videojuego 	<p>Se realiza una evaluación de los problemas, soluciones, objetivos y estimaciones dentro del proyecto. Todos los resultados y lecciones aprendidas son anotadas para mejorar la metodología.</p>
Gestión de riesgos	<ul style="list-style-type: none"> • Minimización de ocurrencia e impacto de problemas <ul style="list-style-type: none"> ○ Probabilidad de ocurrencia ○ Impacto de ocurrencia ○ Mecanismos de monitoreo ○ Estrategia de mitigación ○ Plan de contingencia 	<p>Se realiza un seguimiento continuo de los riesgos que pueden ocurrir durante el ciclo de vida.</p>

Roles

En la metodología se reconocen 4 roles que son: equipo de desarrollo (Scrum Team con subroles), productor interno (Scrum Master), cliente (Product Owner) y verificador beta. El verificador beta verifica la funcionalidad del videojuego y comunica sus resultados [2] [6] [9] [8].

2.2.2 Metodología DAV para el desarrollo de videojuegos

La metodología de Desarrollo Ágil de Videojuegos denominada con sus siglas DAV es la unión entre el marco de trabajo SCRUM y el método ágil Extreme Game Development (XGD). Esta metodología tiene como objetivo la creación sencilla de videojuegos aplicando tecnología y técnicas para su desarrollo [2] [12]. Esta metodología toma en cuenta los elementos descritos en la Tabla 5:

Tabla 5. Descripción de los elementos para el desarrollo con DAV [2] [12]

Elemento	Descripción
Prácticas	Son estrategias y herramientas tomadas de SCRUM y XGD para el desarrollo del videojuego. Entre las prácticas se tiene: <ul style="list-style-type: none">• Diseño incremental• Historias de usuario• Ciclos semanales• Entregas pequeñas• Pruebas
Roles	Los roles que se definen en DAV son los definidos a su vez en SCRUM, añadiendo el rol de tester. Los roles establecidos en la metodología son adecuados para equipos pequeños. Los roles definidos son: <ul style="list-style-type: none">• Dueño del producto• DAV Master• Equipo DAV: conformado por diseñador, programador, artista, sonidista, tester.• Interesados – Clientes
Artefactos	Son, en su mayoría, artefactos seleccionados de SCRUM que permiten la planificación y revisión de cada iteración realizada durante el desarrollo. Los artefactos utilizados son: <ul style="list-style-type: none">• Historias de usuario• Pruebas de aceptación• Reserva del producto• Tarjeta de tarea

Además, al igual que otras metodologías, DAV define fases para el desarrollo del videojuego. Estas fases permiten distinguir en qué etapa de desarrollo se encuentra

el videojuego. Las fases son tomadas del marco de trabajo SCRUM y son descritas en la tabla 6.

Tabla 6. Fases de la metodología DAV [2] [12]

Fase	Descripción
Fase pre-juego	<p>Se realiza la concepción del videojuego describiendo las características que tendrá con la ayuda del dueño y miembros del equipo de desarrollo. Los aspectos esenciales que se describen son los siguientes:</p> <ul style="list-style-type: none"> • Género • Historia • Boceto • Interfaz de usuario • Objetivos • Reglas • Características • Jugabilidad • Diseño de niveles • Requerimientos técnicos • Marketing
Fase juego	<p>En esta fase se realizan iteraciones para desarrollar las funcionalidades del videojuego. Cada una de las iteraciones contiene las siguientes etapas:</p> <ul style="list-style-type: none"> • Elaborar • Integrar • Revisar • Ajustar
Fase post-juego	<p>En esta fase se han completado el desarrollo de todas las funcionalidades. Finalmente se cierra el proyecto realizando la liberación del videojuego, verificación de versiones, generación de documentación final y se realizan los lanzamientos</p>

3 METODOLOGÍA

3.1 Selección de la metodología

En base al estudio de diferentes metodologías tanto tradicionales como ágiles se optó por seleccionar la metodología SUM para el desarrollo del videojuego musical por las siguientes razones:

- La metodología SUM tiene como objetivo el desarrollo de videojuegos y por ello define procedimientos para alcanzar un desarrollo de adecuada de software de esta naturaleza.
- Al tener un equipo de desarrollo y asesoría multidisciplinario es necesaria una metodología que tenga en cuenta este aspecto, lo cual, es evidente en SUM.
- El tiempo estimado para el desarrollo es menor a un año con un equipo de desarrollo pequeño. Esto, por definición, se acopla mucho al contexto en el cual se puede aplicar la metodología SUM.
- Además, al no haber requisitos bien establecidos es necesario seleccionar una metodología ágil para que, durante el desarrollo, se pueda acoger al cambio de una forma adecuada.

3.2 Desarrollo de un videojuego musical utilizando la metodología SUM

En esta sección se describe cómo se elaboró el videojuego de gestos de mano tomando en cuenta la fase de conceptualización, planificación, elaboración, beta y cierre.

3.2.1 Conceptualización del videojuego

En esta fase se desarrolló la conceptualización del videojuego donde se describieron las características técnicas y elementos generales que deberá tener el producto desarrollado. Estos elementos fueron definidos en base a propuestas de aplicaciones que podrían utilizar los 11 gestos de la mano reconocidos por el modelo de inteligencia artificial. En la Tabla 7 se resumen los elementos de videojuegos obtenidos de la fase de conceptualización.

Tabla 7. Fase de conceptualización – elementos del videojuego de baile

Objetivos	Proveer una interfaz que se adecue con los 11 gestos de la mano para una interacción fácil y fluida entre el usuario y el videojuego.
	Proveer una retroalimentación al realizar gestos acertados por el usuario mediante una puntuación para determinar su rendimiento al realizar los gestos solicitados.
	Proveer un entorno atractivo para el usuario donde pueda interactuar mediante sus gestos con el modelo 3D (bailarín).
Género del videojuego	Videojuego musical.
Gameplay	
Acción	Gesto
<ul style="list-style-type: none"> • Desplazamiento entre canciones a la izquierda 	<ul style="list-style-type: none"> • Mano a la izquierda
<ul style="list-style-type: none"> • Desplazamiento entre canciones a la derecha 	<ul style="list-style-type: none"> • Mano a la derecha
<ul style="list-style-type: none"> • Volver al menú 	<ul style="list-style-type: none"> • Puño
<ul style="list-style-type: none"> • Seleccionar opción en menú • Seleccionar canción 	<ul style="list-style-type: none"> • Mano abierta • Apuntar al frente
<ul style="list-style-type: none"> • Seleccionar canción 	<ul style="list-style-type: none"> • Doble golpe de los dedos
<ul style="list-style-type: none"> • Desplazamiento en menú hacia arriba • Desplazamiento entre canciones para arriba • Subir volumen 	<ul style="list-style-type: none"> • Arriba
<ul style="list-style-type: none"> • Desplazamiento en menú hacia abajo • Desplazamiento entre canciones para abajo • Bajar volumen 	<ul style="list-style-type: none"> • Bajar
<ul style="list-style-type: none"> • Gesto de baile 	<ul style="list-style-type: none"> • Mano a la izquierda • Mano a la derecha • Mano abierta • Doble golpe de los dedos • Arriba

	<ul style="list-style-type: none"> • Abajo • Izquierda • Derecha • Apuntar al frente • Apuntar hacia atrás
Características	
Gráficos	<ul style="list-style-type: none"> • Entorno 3D <ul style="list-style-type: none"> ○ Escenario de baile ○ Modelo 3D animado de bailarín
Jugabilidad e interacción	<ul style="list-style-type: none"> • Uso de brazaletes para manipular el videojuego
Sonido	<ul style="list-style-type: none"> • Varias canciones para escoger • Reproducción de canciones a la hora de bailar
Roles	
Existe un modelo 3D de un bailarín que realiza pasos de baile según se requiera.	
Historia	
No existe.	
Ambientación	
Pista de baile con varias luces de colores.	
Público objetivo	
Hombre o mujer mayor a 18 años que disponga de un brazalete para el control de dispositivos y programas mediante gestos de la mano, le gusta la tecnología y su interacción con el ser humano; busca utilizar su brazalete para controlar algún programa de computador que le sirva como entretenimiento. Además, el usuario deberá tener la capacidad motriz total de su brazo derecho para realizar los gestos.	

Cabe recalcar que estos elementos del videojuego fueron idealizados para poder cambiarlos en posteriores fases y sirven como guía para el desarrollo del componente. En la Tabla 8 se resumen los aspectos técnicos tomados en cuenta en esta fase.

Tabla 8. Fase de conceptualización – elementos del videojuego de baile

Plataforma	
Sistema operativo	Windows 10
CPU	AMD 9 5900HS
Memoria	16 GB
GPU	RTX 3060
Tecnologías y herramientas	
Motor de videojuegos	Unity 3.1.0-beta.2
Lenguaje de programación	C#
Lenguaje del modelo de reconocimiento	Matlab
Dispositivo de entrada	Brazaletes de manipulación de dispositivos y programas con gestos de la mano.

Además, la Figura 1 muestra los gestos con los que el usuario podrá interactuar con el videojuego y de esta forma navegar entre los menús y realizar los gestos para que el modelo 3D del bailarín realice los pasos de baile. Por otra parte, en la Figura 2, se muestra una arquitectura de alto nivel donde se visualiza como interactuará el modelo de reconocimiento de gestos realizado en Matlab con el videojuego de baile desarrollado en Unity. Es necesario entender que la comunicación entre el modelo de reconocimiento y el videojuego se realizará mediante sockets. Como salida de esta fase se tiene un documento de conceptualización del videojuego musical con el cuál se da una vista general para así poder entrar a la fase de planeación. Para conocer más información sobre el desarrollo de esta fase véase el Anexo I.

3.2.2 Planificación

En esta fase se planifican varios aspectos administrativos y se plantean cronogramas para navegar a través de las diferentes fases restantes de la metodología. Para esto, se especifica un cronograma donde se detalla la semana referencial. La fecha de inicio y fin, las tareas específicas y el resultado esperado de ser necesario. En la Tabla 9 se muestra el cronograma elaborado como parte de la fase de planificación.

Tabla 9. Fase de planificación – Cronograma

Semana referencial / Etapas	Fecha inicio- fin (si aplica)	Tareas específicas	Resultado esperado (si aplica)
1		Desarrollo del concepto	Concepto del juego
2		Planificación Administrativa y especificación del videojuego	Plan del proyecto
3	02/06/22 al 08/06/22	Iteración de Elaboración 1	
4	09/06/22 al 15/06/22	Iteración de Elaboración 1	Plan de proyecto modificado, videojuego (versión ejecutable), plan de proyecto modificado
5	16/06/22 al 22/06/22	Iteración de Elaboración 2	
6	23/06/22 al 29/06/22	Iteración de Elaboración 2	Videojuego (versión ejecutable)
7	30/06/22 al 6/07/22	Iteración Beta 1	
8	7/07/22 al 13/07/22	Iteración Beta 1	Evaluación y errores encontrados, lista de cambios, videojuego (versión ejecutable)
9	14/07/22 al 20/07/22	Iteración Beta 2	
10	21/07/22 al 27/07/22	Iteración Beta 2	Evaluación y errores encontrados, lista de cambios, videojuego (versión ejecutable)
11	28/07/22	Liberación del videojuego	Videojuego musical

Además, se definen las personas que cumplirán los roles definidos por la metodología. Cabe mencionar que estos roles son acordes al contexto sobre el cual se encuentra desarrollado el videojuego y de tal forma se realizó una adaptación, es decir, el equipo

de desarrollo está conformado por una sola persona. En la Tabla 10 se muestran los roles y el nombre de la persona que lo desempeñará:

Tabla 10. Fase de planificación – Roles

Roles	Nombre
Equipo de desarrollo	César León
Productor interno	César León
Cliente	Ph.D. Lorena Barona Ph.D. Marco Benalcázar Ph.D. Ángel Valdivieso

Adicionalmente, se especificaron características funcionales utilizando la herramienta de historias de usuario. Esto permite definir las necesidades y características funcionales que desea el cliente desde su perspectiva. Además, es una herramienta con la cual los desarrolladores pueden definir diseños, entidades y otros aspectos. En la tabla 11 se muestran las historias de usuario con su nombre y funcionalidad:

Tabla 11. Fase de planificación – Historias de usuario

Historias de usuario			
ID	Nombre	Prioridad	Estimación
H07	Bailar en el videojuego	1	5
H06	Visualizar las canciones	2	3
H01	Visualizar menú principal	3	1
H05	Visualizar instrucciones	4	1
H04	Visualizar puntuaciones	5	1
H03	Configurar el volumen del videojuego	6	1
H02	Salir del videojuego	7	1

Una vez definidas las historias de usuario tomando en cuenta criterios de aceptación, prioridad y estimación se realizó la definición de los sprints en las fechas definidas para la fase de elaboración. Esto es descrito en la Tabla 12.

Tabla 12. Fase de planificación – Planificación de sprints

Sprint	Fecha de inicio	Fecha de fin	Identificador de historias
1	02/06/2022	15/06/2022	<ul style="list-style-type: none"> • H07 – Bailar en el videojuego • H06 – Visualizar las canciones
2	16/06/2022	29/06/2022	<ul style="list-style-type: none"> • H01 – Visualizar menú principal • H05 – Visualizar instrucciones • H04 – Visualizar puntuaciones • H03 – Configurar el volumen del videojuego • H02 – Salir del videojuego

Para profundizar en más detalle cada una de las historias de usuario véase el Anexo 2.

3.2.3 Elaboración

Se realizaron dos iteraciones donde se generaron varias versiones del videojuego. Se implementaron las funcionalidades descritas en el cronograma y se ajustaron aspectos visuales para mejorar la visibilidad de letras dentro de las interfaces. A continuación, en la Tabla 13 se describen las funcionalidades implementadas y cambios realizados durante las 4 semanas de desarrollo:

Tabla 13. Fase de elaboración – Tareas, cambios y ajustes realizados

Fase de desarrollo	Tareas realizadas	Cambios y ajustes realizados
Semana 1	<ul style="list-style-type: none"> • Elección y adecuación del modelo 3D en el entorno virtual. • Elección y adecuación de los pasos de baile para el modelo 3D. • Construcción del escenario (pista de baile). 	<ul style="list-style-type: none"> • Se corrigieron las animaciones para que el modelo no atravesara el escenario. • Se corrigió la estética visual donde el entorno visual no era una pista de baile. • Se ajustó los textos de la interfaz para mejorar su visualización.
Semana 2	<ul style="list-style-type: none"> • Se desarrolló la comunicación entre el videojuego y el aplicativo 	<ul style="list-style-type: none"> • Se corrigió el socket para recibir cualquier tipo de texto.

	<p>externo para recibir el gesto realizado.</p> <ul style="list-style-type: none"> • Se integró la comunicación para recibir el gesto con el modelo 3D del bailarín para realizar los pasos de baile. • Se desarrollo la lógica para la puntuación y reproducir la canción. • Se desarrollo la lógica para parar el baile, continuar el baile y volver a la pantalla de canciones. 	<ul style="list-style-type: none"> • Se corrigió la ejecución de los pasos de baile para que sean ejecutados fluidamente por el bailarín. • Se corrigió la lógica de puntuaciones para que aumente solo si el juego no estaba en pausa.
Semana 3	<ul style="list-style-type: none"> • Se desarrollo la pantalla de visualizaciones de canciones. • Se desarrollo la pantalla de menú principal. • Se implementó la navegación entre el menú principal, pantalla de canciones y la pantalla de baile. • Se incluyó las mejores puntuaciones en la pantalla de canciones por cada una de ellas. 	<ul style="list-style-type: none"> • Se corrigió el desplazamiento entre canciones para que se escuche la canción a seleccionar. • Se incluye una animación a los botones existentes en las pantallas desarrolladas para dar una retroalimentación si son presionados.
Semana 4	<ul style="list-style-type: none"> • Se implemento la configuración del volumen del videojuego. • Se desarrollo el cierre del videojuego. 	<ul style="list-style-type: none"> • Se incluyó música ambiental en las pantallas de menú principal y configuración del volumen. • Se incluyó la reproducción de la canción preseleccionada en la pantalla de canciones.

Durante cada semana se realizaba una presentación a los clientes los cuáles daban la aprobación y ajustes que se deberán realizar en la próxima semana. De esta forma en

la fase de desarrollo se cumplió con todas las historias de usuario a excepción de la H05 – Visualizar instrucciones. Esto se debe a que el tiempo estimado para realizar todas las historias de usuario definidas no fue suficiente, por lo cual se optó por realizar la implementación de esta funcionalidad al comienzo de la fase beta. En la fase beta se detalla más sobre este inconveniente y cuáles fueron los resultados.

Durante la fase de elaboración se desarrollaron artefactos que representan el diseño del videojuego musical. Por ello se crearon varias pantallas con componentes 3D y 2D los cuáles proveen una interfaz amigable y atractiva al usuario. A continuación, en la Figura 6 se muestra un diagrama de navegación entre las diferentes pantallas. Cabe recalcar que la navegación entre pantallas se la realiza mediante gestos de la mano definidos en la fase de conceptualización.

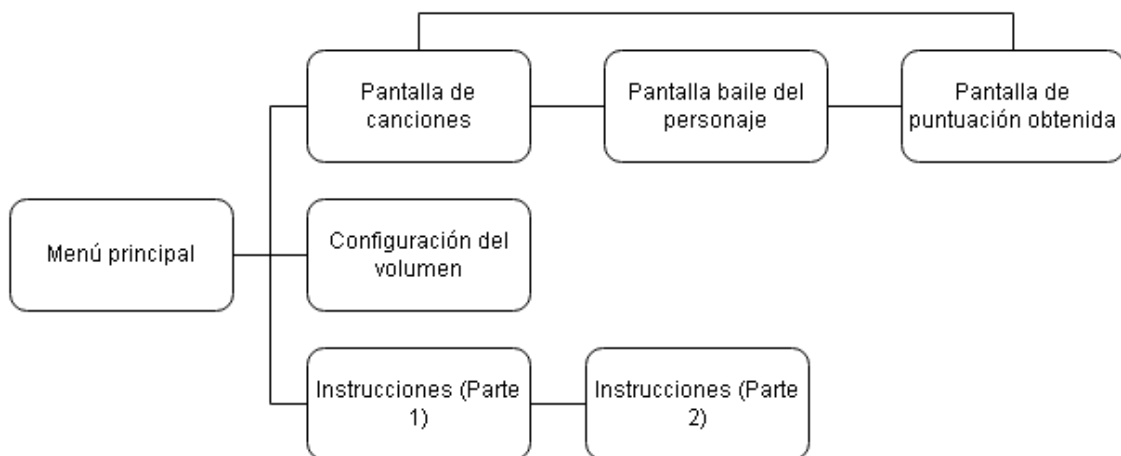


Figura 6. Diagrama de navegación entre pantallas del videojuego

Por otra parte, al usar Unity como motor de videojuegos, se realizaron varios scripts para establecer la lógica entre la navegación y los movimientos que debe realizar el modelo 3D del bailarín cuando el usuario se disponga a bailar. En la figura 7 se muestra un diagrama UML que representa los diferentes scripts realizados en C# y como cada uno de estos se relaciona entre sí.

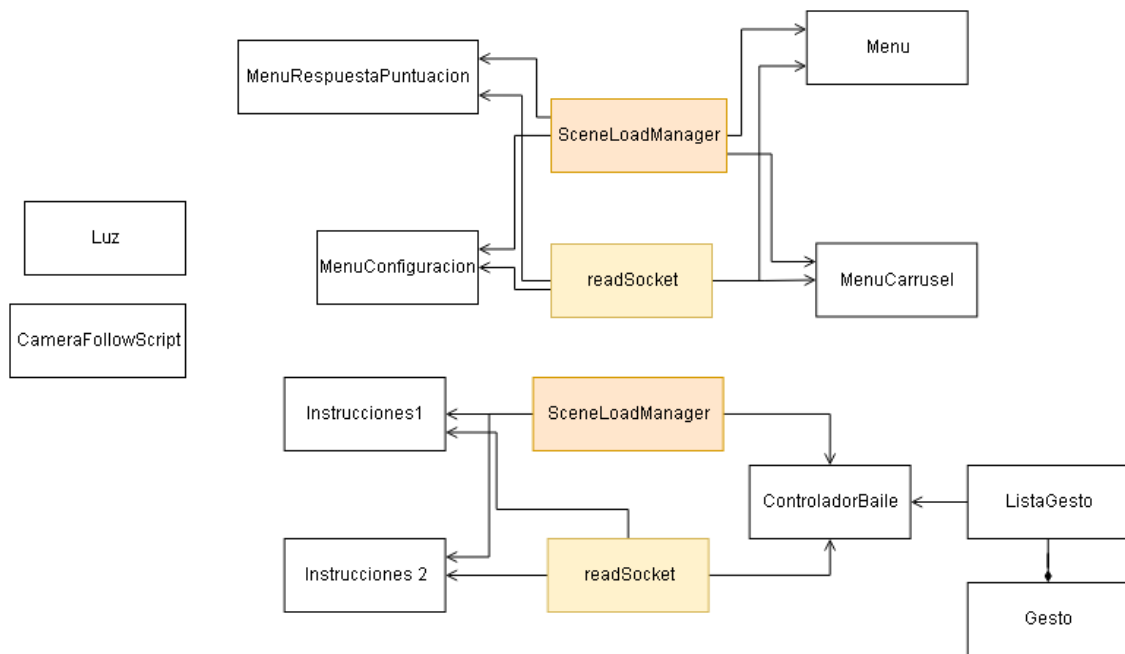


Figura 7. Diagrama de navegación entre pantallas del videojuego

Como se muestra en la figura 7, se definen 11 clases para la construcción del videojuego y cada una cumple un fin específico. Cada clase a excepción de Luz, CameraFollowScript, SceneLoadManager, readSocket, ListaGestos y Gesto controlan únicamente la lógica de una escena descrita en la figura 6. En la tabla 14 se da una descripción general de cada clase y a su vez se relaciona con la escena dentro del videojuego.

Tabla 14. Descripción de las clases en Unity

Clase	Descripción	Escena
Luz	Controla las luces colocadas en las diferentes escenas	Todas las escenas 3D
CameraFollowScript	Controla el movimiento de la cámara para que gire alrededor del modelo 3D	Pantalla baile del personaje
MenuRespuestaPuntuación	Controla la escena y los componentes gráficos de la escena	Pantalla de puntuación obtenida
MenuConfiguración	Controla la escena para la modificación del nivel del volumen	Configuración del volumen

Instrucciones1	Controla la escena y los componentes gráficos de la primera escena para instrucciones	Instrucciones (Parte 1)
Instrucciones2	Controla la escena y los componentes gráficos de la segunda escena para instrucciones	Instrucciones (Parte 2)
SceneLoadManager	Permite el cambio entre escenas	Todas las escenas
readSocket	Permite la comunicación entre el aplicativo de reconocimiento de gestos y el juego	Todas
Menu	Controla la escena principal y sus componentes gráficos	Menú principal
MenuCarrusel	Controla la escena donde se muestran todas las canciones y la lógica para su selección	Pantalla de canciones
ControladorBaile	Controla la escena donde se muestra el modelo 3D, sus componentes gráficos y las acciones que se pueden realizar usando gestos	Pantalla baile del personaje
ListaGesto	Conjunto de gestos	
Gesto	Permite establecer la instancia de un pase de baile de relacionado con el gesto con su nombre y duración	

Además, en la figura 7 se da indicios sobre la integración que se realiza para la comunicación entre el aplicativo de reconocimiento de gestos y el videojuego musical.

Esto se lo hablará en la sección 3.2.6 para evidenciar cómo se realizó la integración, cómo se transmitieron los datos y cuál es su procesamiento para que el videojuego utilice dicha información para realizar las diferentes acciones codificadas. Por otra parte, para conocer y visualizar el código y recursos desarrollados para la aplicación, véase el anexo IV.

3.2.4 Beta

La fase beta fue realizada con los clientes del componente a desarrollar. En esta fase se probaron aspectos técnicos, de comportamiento del videojuego y se verificaron aspectos visuales. Por otra parte, se desarrolló la historia faltante que se menciona en la sección anterior y la integración con el modelo de reconocimiento de gestos. Las tareas realizadas para la mejora e incremento de funcionalidad del videojuego se muestran en la Tabla 14:

Tabla 15. Fase beta– Tareas, cambios y ajustes realizados

Fase de desarrollo	Tareas realizadas	Cambios y ajustes realizados
Semana 1	<ul style="list-style-type: none"> • Implementación de la pantalla de instrucciones. • Integración final con el modelo de reconocimiento de gestos de la mano. 	<ul style="list-style-type: none"> • Se cambió el tamaño de textos en las pantallas de instrucciones y se redistribuyó la información en dos pantallas. • Se realizó un cambio en el socket del lado del videojuego para que reciba un número de gesto y lo mapee con el nombre del gesto realizado para así no cambiar todo el código del videojuego.
Semana 2	<ul style="list-style-type: none"> • Se realizó una presentación en vivo funcional del videojuego con la integración del modelo de reconocimiento de gestos. • Verificación de los criterios de aceptación descritos en las historias de usuario. 	<ul style="list-style-type: none"> • Se realizaron ajustes en el programa que envía los gestos para que no exista un desbordamiento al procesar los gestos en el videojuego. • Se corrigieron criterios de aceptación estéticos dentro del videojuego.

Con la finalización de esta fase se obtuvo la versión final del videojuego cumpliendo así las historias de usuario descritas en la fase de planificación. Además, se realizó un proceso informal de verificación del videojuego para ratificar que las soluciones a las necesidades y expectativas de los clientes estaban solventadas por el videojuego.

3.2.5 Cierre

El cierre fue realizado con la presentación final del videojuego al cliente donde se realizó la presentación con todos los cambios realizados en la fase beta. Al finalizar la presentación se dio la aceptación del componente por parte del cliente y por ende se confirmó el cumplimiento de los requerimientos que se describen en las fases de conceptualización y planificación.

3.2.6 Integración entre el aplicativo de reconocimiento de gestos de la mano y el videojuego musical desarrollado

Para la manipulación del videojuego se definió la utilización de un brazalete para recolectar gestos realizados por el jugador y mapearlo con diferentes acciones dentro del videojuego. Para ello, el videojuego hace uso de un aplicativo desarrollado en Matlab el cual utiliza redes neuronales LSTM para la creación de un modelo que pueda reconocer el gesto del usuario realizado en un tiempo determinado. Una vez que el modelo reconoce el gesto, envía mediante sockets el número del gesto que realizó el usuario. En la tabla 16 se muestra el mapeo entre el número enviado por el aplicativo de reconocimiento y el gesto relacionado.

Tabla 16. Relación entre el número de gesto enviado por el aplicativo de reconocimiento y el mapeo en el videojuego

Número de gesto	Nombre del gesto en el videojuego	Nombre en español
0	NoGesto	Sin gesto
1	WaveIn	Mano a la izquierda
2	WaveOut	Mano a la derecha
3	Fist	Puño
4	Open	Mano abierta
5	Pinch	Doble golpe de dedos
6	Up	Arriba
7	Down	Abajo
8	Left	Izquierda

9	Right	Derecha
10	Forward	Apuntar al frente
11	Backward	Apuntar hacia atrás

Una vez recibido el número del gesto, la clase readSocket del videojuego lo transforma el nombre en inglés del gesto para ser utilizado por las demás clases. Cabe recalcar que en cada una de las clases que hace uso de los gestos se define la lógica para realizar las acciones pertinentes en un momento determinado. Para profundizar más sobre el funcionamiento de la clase readSocket, véase el Anexo III.

4 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

4.1 Resultados

a) Menú principal

Al entrar al videojuego se muestra la pantalla principal con las opciones: JUGAR, VOLUMEN, INSTRUCCIONES y SALIR. El botón predeterminado es el de JUGAR por lo cual es resaltado con otro color. Para desplazarse entre los botones se utiliza el gesto “Arriba” y “Abajo”. Si se desea presionar un botón, el gesto a realizar debe ser “Mano Abierta” o “Apuntar al frente”. En la figura 8 se muestra la pantalla de Menú principal:



Figura 8. Pantalla de Menú principal

b) Instrucciones

Cuando se presiona el botón de instrucciones en el menú principal se despliega la primera pantalla de instrucciones donde se muestra cómo navegar a través de los menús. Si se realiza el gesto “Doble golpe de los dedos” se presentará la segunda pantalla de instrucciones que mostrará cómo realizar algunas acciones especiales dentro del baile. En la Figura 9 y 10 se muestran cada una de estas pantallas.



Figura 9. Pantalla de Instrucciones (Parte 1)



Figura 10. Pantalla de Instrucciones (Parte 2)

c) Configuración del volumen

Cuando se presione el botón de VOLUMEN en el menú principal se presentará la pantalla para configurar el volumen. Cabe recalcar que en esta pantalla se escucha una canción de ambiente para que la configuración del volumen sea en base al volumen de la canción que se escucha. Para subir o bajar el volumen se utilizan los gestos “Arriba” y “Abajo” y de esta forma el nivel de volumen dentro del juego cambiará. Para volver al menú principal se deberá realizar el gesto “Doble golpe de los dedos”. En la Figura 11 se muestra la pantalla de configuración del volumen del videojuego.

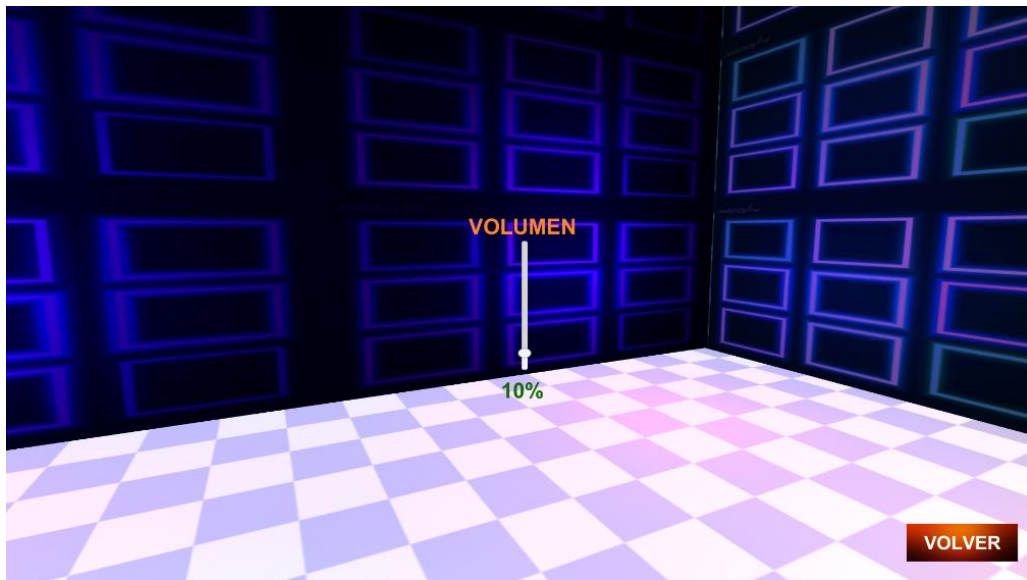


Figura 11. Pantalla de configuración del volumen

d) Pantalla de canciones

Cuando se presione el botón de “Jugar” en la pantalla de menú principal se desplegará la pantalla de canciones donde se muestra todas las canciones que se configuraron el videojuego. En cada recuadro de canción se muestra el nombre de la canción, dificultad y mejor puntuación obtenida. Además, si se realiza el gesto “Doble golpe de los dedos” se volverá a la pantalla de menú principal. En la figura 12 se muestra la pantalla de canciones.

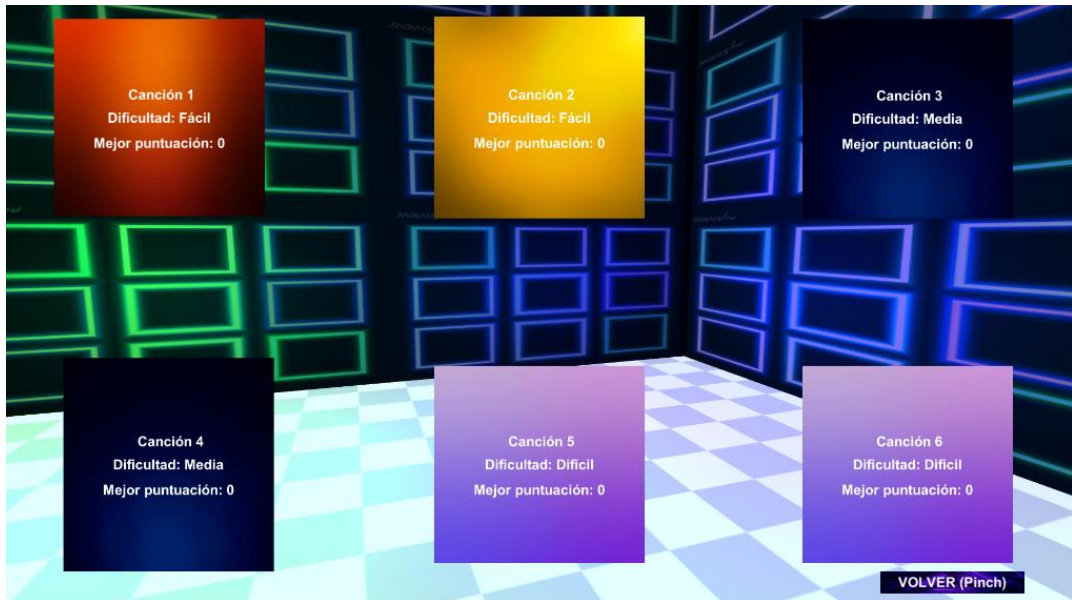


Figura 12. Pantalla de canciones

e) Pantalla baile del personaje

Después de haber seleccionado una canción en la pantalla de canciones se desplegará la pantalla donde se muestra el modelo 3D del bailarín, nombre de la canción, puntuación, video del gesto a realizar y el nombre del gesto. Además, existen dos textos ocultos, donde uno de ellos es la cuenta regresiva de 6 segundos para comenzar a bailar y otro mensaje informativo en caso de que se realice un gesto sea el indicado o no. Además, dentro de esta pantalla se podrá parar el juego y volver a la pantalla de canciones o continuar bailando. En la figura 13 se muestra la pantalla de baile del personaje con los componentes mencionados.



Figura 13. Pantalla de baile del personaje

f) Pantalla de puntuación obtenida

Una vez que el jugador haya terminado de bailar se mostrará la pantalla de puntuaciones obtenidas. Si el jugador obtuvo una puntuación mayor a la registrada como mejor puntuación, este valor se guarda y se presenta un mensaje con el texto “Nueva mejor puntuación”. Por otra parte, si el jugador no obtuvo una mejor puntuación se le presenta el texto “Bien hecho”. La figura 14 muestra la pantalla de puntuación obtenida:



Figura 14. Pantalla de puntuación obtenida

Para visualizar mejor el funcionamiento del videojuego, véase el video mediante el enlace en el Anexo V.

4.2 Conclusiones

Mediante la aplicación de la metodología SUM y siguiendo los procesos definidos en ella se ha obtenido un videojuego musical que contempla elementos técnicos propios del desarrollo de software como elementos del juego. Estos elementos del juego permiten que no solo se satisfaga los aspectos funcionales, sino también los no funcionales que permiten que el jugador tenga una experiencia buena a la hora de interactuar con el videojuego. Además, se evidencia que la relación entre los gestos reconocidos por el modelo y las acciones posibles a realizar en el videojuego concuerdan adecuadamente, lo cual brinda una interacción intuitiva al usuario.

Las funcionalidades implementadas en el videojuego toman en cuenta las características esenciales de los videojuegos como es la puntuación, retroalimentación de acciones, multimedia atractiva y otras formas en las que se atrae al jugador mediante el juego. Cabe recalcar que estas funcionalidades fueron diseñadas bajo la metodología SUM y por ello son tomadas en cuenta durante el desarrollo. De esta forma no se pierde el enfoque de la naturaleza del producto de desarrollado y permite que todos los miembros del equipo entiendan las particularidades que tiene el desarrollo de un videojuego.

Para la interacción entre el jugador y el videojuego se definió el uso de un modelo de reconocimiento de 11 gestos el cual recibe datos de los movimientos de la mano del jugador mediante un brazalete colocado en su antebrazo. Durante las pruebas realizadas se observó que el modelo de reconocimiento de gestos utilizado no reconocía todos los gestos realizados por el usuario. Por otra parte, cuando el modelo reconocía el gesto, este dato era enviado de forma efectiva al videojuego para que este lo procese y lo relacione con una acción dentro del aplicativo. Por ello, la interacción entre el usuario y el videojuego es deficiente.

4.3 Recomendaciones

Es esencial analizar y determinar la metodología más adecuada para el desarrollo de un videojuego. Esto beneficiará definir aspectos como el equipo de trabajo, características del videojuego, cronogramas, herramientas de desarrollo y otros aspectos esenciales para la obtención de un videojuego de calidad. Además, se debe tomar en cuenta que un videojuego puede ser desarrollado con diferentes metodologías y todo dependerá del contexto sobre el cual sea desarrollado. Un punto clave para la elección es la incertidumbre del producto, es decir, cuán bien se conoce al producto que se desea desarrollar. Si se conoce muy bien el producto, es adecuado considerar metodologías tradicionales, caso contrario el camino más efectivo siempre será seleccionar una metodología ágil.

Por otra parte, durante el desarrollo del videojuego es necesario definir qué elementos visuales se van a utilizar, en especial los modelos en 3D. Esto puede ser un factor crucial a la hora de desarrollar un aplicativo que haga uso de estos, ya que dichos modelos deben tener las características necesarias para realizar el movimiento que se necesite. Por ello, si bien no se describe una selección de elementos gráficos como son los modelos 3D es necesario tomar un tiempo para obtenerlos o diseñarlos.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] G. Morales, C. Nava, F. Martines y R. Mirsha, «Procesos de desarrollo para videojuegos,» *Culcyt*, pp. 25-39, 2010.
- [2] H. Rahman, *Videojuego educativo en 3D para dispositivos móviles Android, Enfocado al aprendizaje de la lógica de programación para usuarios entre los 5 a 18 años de edad*, Ecuador, 2017.
- [3] M. Torres, Aplicación de la Metodología Oohdm y Técnicas de Inteligencia Artificial en la Solución del Desarrollo de un videojuego, enfocado a niños de 6 a 10 años, utilizando la Tecnología dgi+ basado en C# y Wiimote, para su aplicación en la Empresa Virtual Learni, Escuela Politécnica del Ejercito, 2013.
- [4] F. Dille y J. Zuur, *The ultimate guide to video game writing and desing*, Lone Eagle Publishing Company, 2007.
- [5] J. Azorín, «El videojuego musical ¿Un recurso para la educación musical en educación primaria?,» *Revista de la Facultad de Educación de Albacete*, nº 29, pp. 19-36, 2014.
- [6] H. Lozano, *Implementación de un juego serio multiplataforma para el desarrollo de la orientación espacial en niños de 6 a 8 años*, Bogotá: Universidad Católica de Colombia, 2017.
- [7] R. Pressman, *Ingeniería de software: un enfoque práctico*, México: Mc Graw Hil, 2010.
- [8] I. Sommerville, *Ingeniería del software*, Madrid: PEARSON EDUCACIÓN, 2005.
- [9] N. Acerenza, A. Coppes, G. Mesa, A. Viera, E. Fernández, T. Laurenzo y D. Vallespir, «Una metodología para desarrollo de videojuegos-Versión extendida,» *Simposio Argentino de Ingeniería de Software (ASSE 2009)*, pp. 171-176, 2009.
- [10] F. Pabón, *Diseño y desarrollo de un video juego educativo con agentes inteligentes y speech recognition para la plataforma Windows aplicando la metodología sum. caso de estudio: tic-tac-toe en 3d*, Sangolquí: Universidad de las Fuerzas Armadas, 2015.
- [11] K. Schwaber y M. Beedle, *Agile Software Development with Scrum*, Prentice Hall PTR, 2001.
- [12] A. Barreno, G. Alexander, S. Guaraca y M. Gonzalo, Adaptación de las metodologías ágiles Scrum y Extreme Game Development en una metodología

para el desarrollo de videojuegos en Android. Caso práctico: Desarrollo de un videojuego, 2013.

- [13] SOFTENG, «Proceso y Roles de Scrum,» [En línea]. Available: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-scrum/proceso-roles-de-scrum.html>. [Último acceso: 29 Agosto 2022].

6 ANEXOS

ANEXO I

CONCEPTO DEL VIDEOJUEGO

Objetivos del juego

- Proveer una interfaz que se adecue con los gestos que puede distinguir el sistema de reconocimiento para permitir una interacción fácil y fluida entre el usuario y el videojuego
- Proveer una retroalimentación de los gestos acertados por el usuario mediante una puntuación para determinar su rendimiento al realizar los gestos indicados durante el juego.
- Proveer un entorno atractivo dónde el usuario pueda interactuar con sus gestos y los movimientos definidos que realizará el modelo 3D del personaje.

Género del videojuego

El videojuego por desarrollar pertenece al género de videojuego musical en el cual el usuario interactúa mediante el uso de dispositivos que reconocen sus movimientos o mediante el uso de periféricos como el ratón y teclado. El videojuego se basará en realizar movimientos indicados mediante pantalla para que el un modelo animado reaccione a dichos movimientos. Además, se incluirá música la cual es parte esencial para que el videojuego propuesto entre en dicha categoría. Un videojuego parecido y que es tomado como referencia es el reconocido título denominado JUSTDANCE, dónde el usuario sigue la coreografía que se le muestra en pantalla y de esta forma acumula un puntaje.

Características

Las características que el juego tendrá se detallan a continuación. Cabe mencionar que estas nos permitirán tener una mejor perspectiva sobre el producto que se desea obtener y de esta forma también identificar de mejor forma aspectos técnicos y artísticos:

- **Gráficos:** el videojuego se desarrollará en un entorno 3D dónde existiría un escenario y un modelo animado que reaccionará según los movimientos acertados que realice el jugador.
- **Jugabilidad e interacción:** la jugabilidad, es decir como el jugador interactúa con el videojuego, será mediante un brazalete el cual recopile información sobre el movimiento realizado por el usuario en un determinado momento. Esta información será procesada por el Sistema de Reconocimiento de Gestos y después se enviará el gesto reconocido al videojuego. Esta información será

procesada y en caso de que exista una acción disponible para dicho gesto se la realizará.

- **Sonido:** el videojuego proveerá la opción de escoger varias canciones para que el usuario “baile”. La canción será reproducida cuando se presente el modelo animado dentro del escenario.

Roles, historia y ambientación

Por la naturaleza del videojuego, no existe una historia como tal detrás del videojuego. Por otra parte, el rol que desempeña tanto el jugador como el modelo animado es de un bailarín el cuál debe seguir los movimientos (gestos) que se le presentan en pantalla y así poder obtener la mejor puntuación. Para obtener una buena puntuación, el jugador deberá realizar los gestos que se le indiquen, caso contrario la puntuación no aumentará. El espacio en el que el bailarín baila simula ser una pista de baile con luces de colores.

Tecnologías y herramientas

Para el desarrollo del videojuego se definió como motor de videojuego Unity en su versión más reciente 3.1.0-beta.2 el cual utiliza como lenguaje de programación C#. Mediante esta plataforma se desarrollará el videojuego tanto en su parte gráfica como lógica. Además, se utilizará el modelo de reconocimiento de gestos realizado utilizando Matlab. El videojuego se comunicará con una aplicación realizada en Matlab para recibir información sobre el gesto realizado por el usuario.

Público objetivo

Tomando en cuenta las características y objetivos del videojuego que se plantea, el público objetivo es definido de la siguiente forma: Hombre o mujer mayor a 18 años; que disponga de un brazalete para el control de dispositivos y programas mediante gestos de la mano; le gusta la tecnología y su interacción con el ser humano; busca utilizar su brazalete para controlar algún programa de computador que le sirva como entretenimiento. Además, el usuario deberá tener la capacidad motriz total de su brazo derecho para realizar los gestos.

ANEXO II

Especificación de características

Código: H01

Nombre: Visualizar menú de inicio

Cómo usuario del videojuego

Deseo visualizar un menú como primera pantalla que contenga las opciones de “Jugar”, “Configuraciones de volumen”, “Instrucciones” y “Salir”

Para poder comenzar a jugar

Y también configurar el volumen del videojuego.

Prioridad: 3

Estimación: 1

Criterios de aceptación:

- El menú constará de 4 botones en los cuales se deberá observar la descripción asignada en la historia de usuario.
- Los botones deberán estar ordenados de la siguiente manera:
 1. Jugar
 2. Instrucciones
 3. Configuraciones de volumen
 4. Salir
- Los botones deberán ser totalmente funcionales para ser capaces de redirigir a otra pantalla según sea el caso

Código: H02

Nombre: Salir del videojuego

Cómo usuario del videojuego

Deseo poder salir del videojuego presionando el botón del menú principal con la etiqueta “Salir”

Para terminar con la ejecución del videojuego y poder realizar otras actividades dentro del computador

Prioridad: 7

Estimación: 1

Criterios de aceptación:

- El botón “Salir” deberá encontrarse solamente en el menú principal
- Cuando se presione el botón el juego se cerrará

Código: H03

Nombre: Configurar el volumen del videojuego

Cómo usuario del videojuego

Deseo poder configurar el volumen del videojuego presionando el botón “Configuraciones de volumen” en el menú principal

Y modificar el nivel del volumen en la ventana emergente, el cual presentará una barra vertical con la escala del 0 al 100 donde el 0 representa la inexistencia de sonido dentro del videojuego y 100 el grado máximo de volumen

Para que el volumen del videojuego sea el de mi agrado

Prioridad: 6

Estimación: 1

Criterios de aceptación:

- Dentro de la barra se debe identificar el nivel de volumen actual del videojuego
- Al presionar los botones para subir y bajar el volumen, el volumen de la canción que se encuentre reproduciéndose deberá ser afectado por este cambio.
- Al presionar el botón “VOLVER” se visualizará completamente el menú principal
- Si el volumen está configurado en 0, no se escuchará ningún sonido dentro del videojuego

Código: H04

Nombre: Visualizar puntuaciones

Cómo usuario del videojuego

Deseo poder visualizar la mejor puntuación obtenida, nombre de la canción y dificultad de cada una de las canciones presentes dentro del videojuego en la ventana de canciones presionando el botón “Jugar” que se encuentra en el menú principal

Para así conocer mi habilidad en cada una de ellas

Prioridad: 5

Estimación: 1

Criterios de aceptación:

- Cuando se presione el botón “Jugar” del menú principal se presentará otra pantalla donde se muestre la información descrita anteriormente. Si no existe una puntuación para esa canción deberá aparecer con el valor de 0.
- La pantalla deberá contener el botón de “VOLVER” para volver al menú principal
- Ningún dato mostrado podrá ser modificado

Código: H05

Nombre: Visualizar instrucciones

Cómo usuario del videojuego

Deseo poder visualizar las instrucciones presionando el botón “Instrucciones” que se encuentra en el menú principal

Para conocer cómo se debe interactuar con el videojuego y que funciones puedo utilizar

Prioridad: 4

Estimación: 1

Criterios de aceptación:

- Cuando se presione el botón “Instrucciones” se deberá presentar una pantalla con la siguiente información del videojuego:
 1. Acciones que se pueden realizar dependiendo el gesto
 2. Pasos para seleccionar una canción y jugar
 3. Pasos para dejar de jugar
 4. Pasos para modificar el volumen del videojuego
- Esta información podrá ser presentada en una pantalla o en un tipo carrusel.
- Se deberá colocar el botón “Inicio” para regresar al menú principal
- Ningún dato presentado podrá ser modificado

Código: H06

Nombre: Visualizar las canciones

Cómo usuario del videojuego

Deseo poder visualizar el nombre de la canción, su duración, mejor puntuación y dificultad de todas las canciones como también escuchar un fragmento de la canción preseleccionada presionando el botón “Jugar” de la pantalla principal

Para poder elegir cual es la que deseo jugar

Prioridad: 2

Estimación: 3

Criterios de aceptación:

- Las canciones deberán ser presentadas en una tabla mostrando la información antes mencionada
- La canción preseleccionada por defecto a la primera que se muestre en la tabla
- Se podrá navegar por la tabla para preseleccionar una canción
- El fragmento de canción que se escuchará dependerá de la canción preseleccionada de ese momento

Código: H07

Nombre: Bailar en el videojuego

Cómo usuario del videojuego

Deseo poder visualizar al modelo 3D de la persona, la puntuación, los pasos de baile que debo realizar y escuchar la canción que he seleccionada previamente

Para poder realizar los movimientos que aparezcan en pantalla y así ir ganando puntos para sumar la mayor cantidad posible

Prioridad: 1

Estimación: 5

Criterios de aceptación:

- Los pasos serán obtenidos en tiempo real, por lo cual los pasos de una canción siempre serán diferentes
- Cuando el usuario realice el gesto pedido por el videojuego, el modelo 3D de la persona comenzará a bailar y la puntuación del usuario se incrementará.
- Si el usuario no realiza ningún gesto, el modelo 3D de la persona no realizará ningún movimiento y la puntuación del usuario no se incrementará.
- Se podrá pausar el videojuego, donde tanto el modelo 3D de la persona, los gestos requeridos por el videojuego y la canción se pondrán en "pausa".
- Se podrá reanudar el videojuego y tanto el movimiento del modelo 3D de la persona, los gestos requeridos por el videojuego y la canción se reanudarán.
- Al finalizar la canción completamente y si la puntuación es mayor de la que se tiene registro, se guardará como mejor nueva puntuación y se presentará un aviso al usuario mencionándolo. Si la puntuación no es mejor, se presentará la puntuación obtenida.
- Después de presentar toda la información sobre las puntuaciones se volverá a la pantalla de selección de canciones.

ANEXO III

Código de la clase readSocket

```
using UnityEngine;
using System.Collections;
using System.Net;
using System.Net.Sockets;
using System.Linq;
using System;
using System.IO;
using System.Text;

public class readSocket : MonoBehaviour {
    // Use this for initialization
    TcpListener listener;
    String msg = "";
    public void Start () {
        IPAddress localAddr = IPAddress.Parse("127.0.0.1");
        listener=new TcpListener (localAddr, 55001);
        listener.Start ();
        print ("is listening");
    }
    // Update is called once per frame
    public String Update () {
        if (!listener.Pending())
        {
            msg= "NoGesto";
            return msg;
        }
        else
        {
            print ("socket comes");
            TcpClient client = listener.AcceptTcpClient ();
            NetworkStream ns = client.GetStream ();
            StreamReader reader = new StreamReader (ns);
            msg = reader.ReadToEnd();
            print (msg);
        }
    }
}
```

```
switch(msg){
  case "1":
    msg = "WaveIn";
    break;
  case "2":
    msg = "WaveOut";
    break;
  case "3":
    msg = "Fist";
    break;
  case "4":
    msg = "Open";
    break;
  case "5":
    msg = "Pinch";
    break;
  case "6":
    msg = "Up";
    break;
  case "7":
    msg = "Down";
    break;
  case "8":
    msg = "Left";
    break;
  case "9":
    msg = "Right";
    break;
  case "10":
    msg = "Forward";
    break;
  case "11":
    msg = "Backward";
    break;
  case "0":
    msg = "NoGesto";
```

```
        break;
    }
    return msg;
}
}
```

ANEXO IV

Repositorio de recursos y códigos fuente del aplicativo

Enlace: <https://github.com/CesarIsraelLeonCifuentes/VideojuegoTesis>

ANEXO V

Video demostrativo del funcionamiento del videojuego

Enlace: <https://youtu.be/gCcrnLGinaQ>